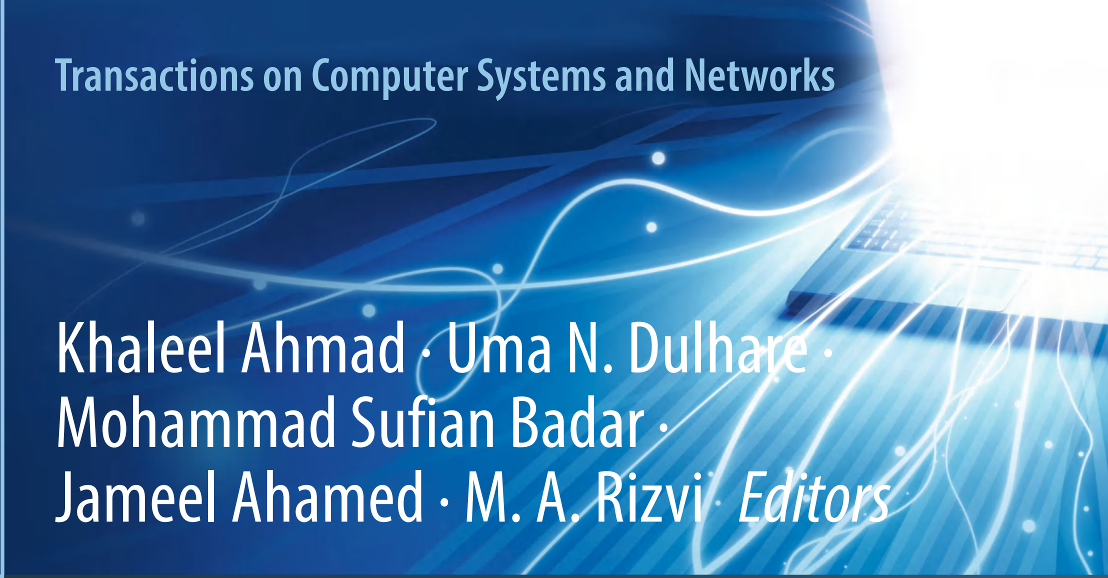
Transactions on Computer Systems and Networks



Khaleel Ahmad · Uma N. Dulhare ·
Mohammad Sufian Badar ·
Jameel Ahamed · M. A. Rizvi *Editors*

Fostering Machine Learning and IoT for Blockchain Technology

Smart Cities Applications, Volume 1

Springer

Transactions on Computer Systems and Networks

Transactions on Computer Systems and Networks is a unique series that aims to capture advances in evolution of computer hardware and software systems and progress in computer networks. Computing Systems in present world span from miniature IoT nodes and embedded computing systems to large-scale cloud infrastructures, which necessitates developing systems architecture, storage infrastructure and process management to work at various scales. Present day networking technologies provide pervasive global coverage on a scale and enable multitude of transformative technologies. The new landscape of computing comprises of self-aware autonomous systems, which are built upon a software-hardware collaborative framework. These systems are designed to execute critical and non-critical tasks involving a variety of processing resources like multi-core CPUs, reconfigurable hardware, GPUs and TPUs which are managed through virtualisation, real-time process management and fault-tolerance. While AI, Machine Learning and Deep Learning tasks are predominantly increasing in the application space the computing system research aim towards efficient means of data processing, memory management, real-time task scheduling, scalable, secured and energy aware computing. The paradigm of computer networks also extends it support to this evolving application scenario through various advanced protocols, architectures and services. This series aims to present leading works on advances in theory, design, behaviour and applications in computing systems and networks. The Series accepts research monographs, introductory and advanced textbooks, professional books, reference works, and select conference proceedings.

Khaleel Ahmad · Uma N. Dulhare ·
Mohammad Sufian Badar · Jameel Ahamed ·
M. A. Rizvi
Editors

Fostering Machine Learning and IoT for Blockchain Technology

Smart Cities Applications, Volume 1

Springer

Editors
Khaleel Ahmad
Maulana Azad National Urdu University
Hyderabad, Telangana, India

Mohammad Sufian Badar
Department of Bioengineering
University of California, Riverside
Riverside, CA, USA

M. A. Rizvi
National Institute of Technical Teachers
Training and Research
Bhopal, Madhya Pradesh, India

Uma N. Dulhare
Muffakham Jah College of Engineering
and Technology
Hyderabad, Telangana, India

Jameel Ahamed
Maulana Azad National Urdu University
Hyderabad, Telangana, India

ISSN 2730-7484 ISSN 2730-7492 (electronic)
Transactions on Computer Systems and Networks
ISBN 978-981-96-4077-5 ISBN 978-981-96-4078-2 (eBook)
https://doi.org/10.1007/978-981-96-4078-2

This Springer imprint is published by the registered company Springer Nature Singapore Pte Ltd.
The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721, Singapore

Preface

In the modern era, the integration of Machine Learning, Internet of Things (IoT), and Blockchain technology has the transformative potential to shape smart urban development. “Fostering Machine Learning and IoT for Blockchain Technology Smart Cities Applications” has taken the initiative, so that the readers are able to understand how these cutting-edge technologies can collaborate to enhance smart city applications.

This book is written to provide awareness of methods used for blockchain in the academic and professional community. While this book will discuss blockchain with IoT, big data and its applications like healthcare, police management, and cybersecurity, it will also focus on contemporary topics for research and development.

The book is organized into 13 chapters. It covers the pre-requisite concept of blockchain like cryptography, then it gives an introduction to blockchain technology, covering topics, viz., consensus mechanisms, architecture, and enterprise solutions for modern business applications, such as Hyperledger Fabric. It also puts an emphasis on private and consortium blockchains, along with practical applications such as lightweight encrypted police management system for enhancing public safety and data management. This is done on a blockchain-based serverless platform using the Blowfish algorithm to emphasize the importance of efficient data handling in smart city solutions.

The application of blockchain in healthcare explores how these technologies can improve patient care and streamline operations within the health sector, while also fortifying cybersecurity measures with the consideration of privacy and data safety.

Hyderabad, India — Khaleel Ahmad
Hyderabad, India — Uma N. Dulhare
Riverside, USA — Mohammad Sufian Badar
Hyderabad, India — Jameel Ahamed
Bhopal, India — M. A. Rizvi

Contents

Editors and Contributors

About the Editors

Dr. Khaleel Ahmad is currently an Assistant Professor in the Department of Computer Science and Information Technology at Maulana Azad National Urdu University, Hyderabad. He has more than nine years of teaching and research experience. He worked as a Visiting SERB International Research Fellow in the Department of Computer Science, University of Pisa, Italy, under the SERB International Research Experience (SIRE) Fellowship Programme funded by the Science and Engineering Research Board (SERB), Government of India for six months. He visited the National Defence University of Malaysia, Malaysia as a Visiting Faculty from 26 December 2018 to 6 January 2019. He completed one Research Project of 01.05 Lakh. He received a UGC-MANUU National Travel Grant in 2014. He has received the best paper award at two conferences in Malaysia and India. He coordinated two Faculty Development Programmes sponsored by DRDO and NITTTR, Bhopal. He has supervised one Ph.D. student, seven M.Tech. students, and several B.Tech. students, along with many MCA research projects. His research areas are Blockchain Technology, Cybersecurity, Cryptography, and Opportunistic Networks. He has filed two patents from Malaysia in collaboration with the National Defence University of Malaysia. He has published more than 50 papers in refereed Journals and conferences (viz. Nature, Elsevier, ACM, IEEE, and Springer) and 20 book chapters (Springer, Taylor & Francis, Wiley, IGI Global). He has edited four books published by Springer, Taylor & Francis, and Wiley. He has delivered lectures in India and abroad and chaired several sessions at National and International conferences. He is a life member of various international/national research societies viz. ISTE, CRSI, ISCA, IACSIT (Singapore), IAENG (Hong Kong), ISOC (USA). In addition, he is associated with many international research organizations as an editorial board member and reviewer.

Prof. Uma N. Dulhare is currently working as a Professor and Head of the Department of Computer Science and Artificial Intelligence, Muffakham Jah College of Engineering, Hyderabad, India. She has more than 20 years of teaching experience.

She received her Ph.D. from Osmania University, Hyderabad. Her research interests include Data Mining, Big Data Analytics and Machine Learning, IoT, Cloud Computing, and Biomedical Image Processing. She has published over 30 research papers in reputed national and international journals and book chapters. Also, she edited three books. She is a Member of the Computer Science Teachers Association (CSTA), USA, IEEE (SMC), ACM, Senior Member of the International Association of Engineers (IAENG), Senior Member of Universal Association of Computer and Electronics Engineers (UACEE) Life Member of ISTE.

Dr. Mohammad Sufian Badar Ph.D. has served as a Senior Teaching Faculty in the Department of Bioengineering at the University of California, Riverside, CA, USA. He served as an Analytics Architect in CenturyLink for over a year in Denver, CO, USA. Currently, he has been serving as a senior faculty (temporary) in the Department of Computer Engineering at Jamia Hamdard, New Delhi, India. He possesses an excellent academic record with an M.S. degree in Molecular Science and Nanotechnology and a Ph.D. in Engineering from Louisiana Tech University Ruston, LA, USA, respectively. Before joining the Ph.D. program at Louisiana Tech University, he graduated with an M.Sc. in Bioinformatics from Jamia Millia Islamia University, New Delhi, India. Dr. Sufian has over 14 years of teaching, research, and industry experience. He has published his research in conferences and highly reputed International journals. He has authored many chapters in the area of artificial intelligence/machine learning and blockchain/IoT. He has developed an algorithm for Face Detection, Recognition, and Emotion Recognition. He is currently in the process of developing a device that, using Biosensors, can correlate the physiology of the human body with the emotion recognition algorithm, giving us a clear measure of the amount of stress hormones in the body. Currently, he and his group have developed an ML model that predicts COVID-19 infection just from the patient's symptoms.

Dr. Jameel Ahamed has done his Doctor of Philosophy (Ph.D.) from the National Institute of Technology Srinagar, J&K (NIT Srinagar). He has completed a Bachelor of Technology (B.Tech.) from the Faculty of Engineering, Jamia Millia Islamia, New Delhi, and a Master of Technology (M.Tech.) from the National Institute of Technology, Srinagar, J&K (NIT Srinagar). Before joining the School of Technology, MANUU, Hyderabad, as an Assistant Professor, he worked as Guest Faculty in the Department of Electronics Engineering, Aligarh Muslim University, Aligarh, Uttar Pradesh for over a year. He coordinated two Faculty Development Programmes sponsored by DST and UGC. He also completed one Research Project of 01.05 Lakh. He has more than 15 research publications to his credit and delivered many invited talks. He has visited countries like the Czech Republic and the UK for conference paper presentations. He has supervised seven Master of Technology (M.Tech.) and several Bachelor of Technology (B.Tech.) and MCA research projects. His research areas include the Internet of Things, Computer Networks, Machine Learning, and Network Security.

Prof. M. A. Rizvi (Dean) obtained his Doctorate in Computer Science from Maulana Azad National Institute of Technology (MANIT), Bhopal. Dr. Rizvi has also achieved a Master's degree in Electronics and a Graduate Diploma in Computer Applications from Aligarh Muslim University, Aligarh, and a Master's of Business Administration (MBA HR) from Barkatullah University, Bhopal. Dr. Rizvi has more than 29 years of experience in the field of Computer Science and Applications as a faculty (Professor and Dean) at the National Institute of Technical Teachers' Training and Research, Bhopal (NITTTR) a Government of India Institute. He has published approximately 140 research papers in reputed international journals and international conferences across the globe. He has published three patents and two copyrights on facial recognition algorithms and computer vision-based automatic attendance calculation and updating systems. He has attended many international/national conferences in India and abroad to present his research papers. He was invited to many international conferences as a keynote speaker, session chair, and invited talk. A book named "Computer and Communication a Practical Manual for Internet Café" for NCERT, New Delhi, India written by him is another feather to his cap. He has authored 10 chapters in edited books published by international publishers.

Contributors

Jameel Ahamed Department of Computer Science and Information Technology, Maulana Azad National Urdu University, Hyderabad, India

Khaleel Ahmad Department of Computer Science and Information Technology, Maulana Azad National Urdu University, Hyderabad, India

Asmaa Mahfoud Hezam Alhakimi Faculty of Information Sciences and Engineering, Management and Science University, Shah Alam, Selangor, Malaysia

Orif Allanov Department of Cybersecurity and Forensics, Tashkent University of Information Technologies, Tashkent, Uzbekistan

Madhavi Satish Avhankar Indira College of Commerce and Science, Pune, India

Asif Ali Banka Department of Computer Science and Engineering, Islamic University of Science and Technology, Awantipora, Kashmir, India

Seerat Bashir Department of Computer Science and Engineering, Islamic University of Science and Technology, Awantipora, Kashmir, India

Ilkhom Boykuziev Applied Mathematics and Intelligent Technologies Faculty, National University of Uzbekistan, Tashkent, Uzbekistan;
Department of Cybersecurity and Forensics, Tashkent University of Information Technologies, Tashkent, Uzbekistan

Swatisipra Das Department of Computer Science, Fakir Mohan University, Balasore, India

Mubashir Farooq Department of Computer Science and Engineering, Islamic University of Science and Technology, Awantipora, Kashmir, India

Najma Farooq Department of Computer Science and Engineering, Islamic University of Science and Technology, Awantipora, Kashmir, India

Elham Ghanbari Department of Computer Engineering, YI, Islamic Azad University, Tehran, Iran

Asif Iqbal Hajamydeen Artificial Intelligence and Cyber Security Centre, Management and Science University, Shah Alam, Selangor, Malaysia

Ravindra S. Kamble Department of Computer Science & Engineering, D. Y. Patil Agriculture and Technical University, Kolhapur, Maharashtra, India

Meenakshi Kandpal School of Computer Science, Odisha University of Technology and Research, Bhubaneswar, India

Vinaya Keskar Department of Computer Science and Application, ATSS College of Business Studies and Computer Application, Pune, India

Halimjon Khujamatov Department of Computer Engineering, Gachon University, Seongnam, South Korea

Kishor Kolhe Department of Computer Science and Engineering, School of Computer Engineering and Technology, Dr. Vishwanath Karad MIT World Peace University, Pune, India

Deepika Kukreja Department of Information Technology, Netaji Subhas University of Technology, Delhi, India

Divyanshu Kumar Department of Information Technology, Netaji Subhas University of Technology, Delhi, India

Sunil A. Kumbhar Department of Computer Science & Engineering, D. Y. Patil Agriculture and Technical University, Kolhapur, Maharashtra, India

Vijaya Kumbhar School of Computer Studies, Sri Balaji University, Pune, India

Manisha Maddel Indira School of Business Studies, Pune, India

Adil Mudasir Malla Department of Computer Science and Engineering, Islamic University of Science and Technology, Awantipora, Kashmir, India

Nawaz Abdullah Malla University of Camerino, Camerino, Italy

Jyoti Mante Department of Computer Science and Engineering, School of Computer Engineering and Technology, Dr. Vishwanath Karad MIT World Peace University, Pune, India

Haris Manzoor Department of Computer Science and Engineering, University of Kashmir, Srinagar, India

Minati Mishra Department of Computer Science, Fakir Mohan University, Balasore, India

Pranati Mishra School of Computer Science, Odisha University of Technology and Research, Bhubaneswar, India

Sara Najafzadeh Department of Computer Engineering, YI, Islamic Azad University, Tehran, Iran

Fatemeh Nasiri Department of Computer Engineering, YI, Islamic Azad University, Tehran, Iran

Afeefa Noorain Department of Computer Science and Information Technology, Maulana Azad National Urdu University, Hyderabad, India;
Department of Computer Science, St. Francis College for Women, Hyderabad, India

Sneha Pandey Department of Information Technology, Netaji Subhas University of Technology, Delhi, India

Rojalina Priyadarshini Department of Computer Science and Engineering, C. V. Raman Global University, Bhubaneswar, India

Jyotirmayee Rautaray School of Computer Science, Odisha University of Technology and Research, Bhubaneswar, India

Laura Emilia Maria Ricci Department of Computer Science, University of Pisa, Pisa, Italy

Islambek Saymanov School of Mathematics and Natural Sciences, New Uzbekistan University, Tashkent, Uzbekistan;
Applied Mathematics and Intelligent Technologies Faculty, National University of Uzbekistan, Tashkent, Uzbekistan;
Information Technology and Computer Engineering Faculty, Andijan State University, Andijan, Uzbekistan

Deepak Kumar Sharma Department of Information Technology, Indira Gandhi Delhi Technological University for Women, Delhi, India

Iouliia Skliarova Department of Electronics, Telecommunications and Informatics, University of Aveiro, Aveiro, Portugal

Tushar Department of Information Technology, Netaji Subhas University of Technology, Delhi, India

Rasila Walhhekar Symbiosis Institute of Computer Studies and Research Centre, Pune, India

Pallavi Yarde School of Computer Studies, Sri Balaji University, Pune, India

Chapter 1
Introduction to Cryptography for Blockchain

Islambek Saymanov, Iouliia Skliarova, Ilkhom Boykuziev, and Orif Allanov

Abstract This chapter provides information about the major branches of the field of cryptography. In particular, public and private key cryptography and their applications are discussed. Public key cryptosystems are based on functions or algorithms whose inverses do not exist or whose inverses cannot be filled in by modern science and technology, at very high material cost and a lot of time in the future. Symmetric block cipher algorithms are an important component of all cryptographic systems. Symmetric block cipher algorithms provide data confidentiality by dividing data into blocks of a certain length and repeating operations on them. Symmetric block ciphers consist of various transformations and can be broadly classified into linear and nonlinear types. In recent years, elliptic curves have been widely used in cryptography. This section introduces general information about such elliptic curves, their position and properties in the coordinate system, and operations performed on points with rational coordinates in finite fields. These operations are operations such as adding points to an elliptic curve, and determining the order. The SHA256 hash algorithm and its sequence of steps are also explained. Information is also presented on hashing, one-way functions, methods for generating one-way functions, formulas used for hashing, and some properties of hash functions, including deterministic, fast

I. Saymanov (✉)
School of Mathematics and Natural Sciences, New Uzbekistan University, Tashkent, Uzbekistan
e-mail: islambeksaymanov@gmail.com

I. Saymanov · I. Boykuziev
Applied Mathematics and Intelligent Technologies Faculty, National University of Uzbekistan, Tashkent, Uzbekistan

I. Saymanov
Information Technology and Computer Engineering Faculty, Andijan State University, Andijan, Uzbekistan

I. Skliarova
Department of Electronics, Telecommunications and Informatics, University of Aveiro, Aveiro, Portugal

I. Boykuziev · O. Allanov
Department of Cybersecurity and Forensics, Tashkent University of Information Technologies, Tashkent, Uzbekistan

K. Ahmad et al. (eds.), *Fostering Machine Learning and IoT for Blockchain Technology*,
Transactions on Computer Systems and Networks,
https://doi.org/10.1007/978-981-96-4078-2_1

computation, avalanche effect, and must withstand collision properties. The characteristics of an electronic digital signature are different and depend on the bits of memory registers, which are determined by the characteristics of the binary number system. Copying or changing an electronic signature consisting of a specific sequence of memory bits is not difficult in computer communication systems. In today's highly developed world civilization, the use of documents, including confidential ones, in electronic form and their transmission in communication systems is widely used, which makes the issues of determining the authenticity of electronic documents and electronic signatures very important. No matter how convenient and secure public key cryptographic systems are, they cannot completely solve the authentication problem. Therefore, it is necessary to comprehensively use authentication methods and means together with cryptographic algorithms. Currently, electronic digital signatures based on RSA and El-Gamal are widely used. However, there are attack methods designed to create digital signatures. The signature can be forged by factoring and solving a discrete logarithm problem that produces the value of the signer's key using the signature verification key. As a result, digital signature algorithms for Edwards Curve and Elliptic Curve known as EdDSA (Edwards Digital Signature Algorithm) and ECDSA (Elliptic Curve Digital Signature Algorithm) were developed and are in use. This chapter also provides information about the DSA (Digital Signature Algorithm) GOST R 34.10–2001 DSA standards. The Diffie–Hellman key exchange algorithm used for key exchange is also discussed. Discusses the security of the Diffie–Hellman key exchange algorithm and shows that the algorithm can be analyzed using a man-in-the-middle attack. Cryptography uses authentication information that is provided to the other party, that is, known in advance by both parties. Authentication information is typically private or confidential information. It is also risky to provide the other party with complete authentication information when the time comes. In this context, providing part of the authentication information without providing the full information is called zero-knowledge proof. Using "zero-knowledge proofs", one party can demonstrate to another (the verifying party) that a statement is true, while hiding any information other than the truth of the statement. Protocols based on zero-knowledge evidence ensure the anonymity of the parties.

Keywords Public and private cryptography · Elliptic curve cryptography · SHA256 · Electronic digital signature · RSA · El-Gamal · DSS (Digital Signature Standard) · DSA (Digital Signature Algorithm) · Signature verification process · Diffie–Hellman key distribution algorithm · Man-in-the-middle attack · Probability · Zero-knowledge proofs

1.1 Introduction

This chapter provides information about the major branches of the field of cryptography. In particular, public and private key cryptography and their applications are discussed. It provides basic information about public key crypto systems and symmetric block cipher algorithms, which are important components of all cryptographic systems. It provides general information about elliptic curves, widely used in cryptography in recent years, their position and properties in the coordinate system, operations such as addition of points of elliptic curves performed in finite fields over points with rational coordinates, and determination of their meeting order. The SHA256 hash algorithm and its sequence of steps are also explained. Information is also presented on hashing, one-way functions, methods for generating one-way functions, formulas used for hashing, and some properties of hash functions, including determinism, fast computation, strict avalanche effect, and collision avoidance properties. The characteristics of an electronic digital signature are different and depend on the bits of memory registers, which are determined by the characteristics of the binary number system. The need for integrated use of authentication methods and means in conjunction with cryptographic algorithms is emphasized. This chapter also provides information about the DSA (Digital Signature Algorithm) GOST R 34.10-2001 DSA standards. The Diffie–Hellman key exchange algorithm used for key exchange is also discussed. Cryptography uses authentication information that is provided to the other party, that is, known in advance by both parties. This chapter also provides information about zero-knowledge proofs, which are known as providing part of authentication information without providing the full information.

A software model called public key cryptography illustrates the process of encoding data, in this example, the documents that are part of a file. The first step in the encoding procedure is to identify the master file or document that has to be encoded. Then, the encryption function will create the encoded document by using the combination method, shifting, and implementing the passphrase to the document's contents. After that, the decryption function uses the passphrase implementation technique to return the contents of the encrypted document, rearranging and combining them to create the original master document once more (Liestyowati 2020).

This paper presents an overview of public key cryptography based on the discrete logarithm problem for bounded fields and elliptic curves. We discuss the grouping law, which is one of the basic and important properties of elliptic curves, and show that the set of points on the curve forms an additive Abelian group (Muyinda 2009).

Hash functions in cryptography are used to guarantee the security of numbers. The significance of hash functions, their many structures, design techniques, attacks, and the most recent advancements in this area are all covered in this article (Sobti and Geetha 2012).

Many traditional and newer businesses and applications have recently been carrying out enormous amounts of electronic transactions, which have led to a critical need for protecting the information from being maliciously altered, ensuring authenticity, and supporting non-repudiation. Just as signatures facilitate validation and verification of the authenticity of paper documents, digital signatures serve the purpose of validation and authentication of electronic documents. This technology is rather new and emerging and is expected to experience growth and widespread use in the coming years (Subramanya and Yi 2006).

This paper extends the Diffie–Hellman algorithm by using the Diffie–Hellman algorithm to obtain a stronger secret key, and this secret key is exchanged between the sender and the receiver with a new secret for each message. A public key is generated. The second secret key is generated by taking the primitive root of the first secret key (Kumar et al. 2017).

This study proposes PipeZK, a two-subsystem pipeline accelerator, to efficiently and practically enable ZKP in real-world applications by handling the two computationally demanding tasks listed above. Large cores are divided into smaller cores in the first subsystem using a new data stream. These smaller cores are then operated by bandwidth-efficient hardware modules with optimized on-chip computational resources and off-chip memory access. In order to minimize resource usage, load imbalance, and heavy load-sharing processing units, the second subsystem uses a lightweight dynamic job dispatch method. PipeZK can achieve a 10 × speedup in conventional cryptographic benchmarks when measured at 28 nm, and a 5 × speedup in the well-known cryptocurrency application Zcash (Zhang et al. 2021).

1.2 Introduction to Public and Private Cryptography

Modern cryptography is the process of information exchange in the communication network:

- ensuring confidentiality;
- determination of completeness;
- authentication—determination of authenticity of parties and users;
- ensuring non-repudiation of authorship;
- providing keys creation, distribution, and management.

The main areas of use of cryptographic methods include: transfer of confidential data over a communication network (for example, via e-mail), ensuring the authenticity of transmitted data, storing information (documents, databases) in encrypted form in the memory of computer systems, and similar issues.

The growing use of encryption to protect data has led to an increase in the need for more user-friendly encryption methods, as well as their reliability.

The processes of protecting information using cryptographic methods are carried out by programming special cryptographically strong algorithms in algorithmic languages or using special technical devices (Schneier n.d.). At the same time, programming methods are easy to use. Although methods using technical devices require a large amount of material resources, they differ in efficiency, convenience, reliability, and other aspects.

The following general requirements are imposed on modern cryptographic methods of protecting information systems:

- the ability to read encrypted information is possible only when the decryption key is known;

- to determine the encryption key used for a certain part of the encrypted text or for the corresponding plaintext, the number of operations that must be performed must not be less than the number of all operations that must be performed to find the key, that is, not less than the number of elements of the set from which the key must be selected;
- knowledge of the encryption algorithm must not negatively affect the stability of the encryption algorithm;
- any level (more or less) of change in the key must lead to a sharp change in the encrypted information;
- elements of the encryption algorithm must be immutable;
- additional bits included in the data during the encryption process must be fully and reliably used in the encrypted text;
- there must be no simple and easily established dependencies between the keys used in the encryption process;
- an additional key obtained from a set of key contents must ensure reliable protection of information;
- the cryptographic algorithm is easy to use both in software and technical terms, and changing the key length should not lead to a failure of the encryption algorithm.

Shannon's paper entitled "A Theory of Communication in Secrecy Systems" in 1949 opened the era of scientifically based secret key cryptography. Based on his background in electrical engineering and mathematics, Shannon laid the foundations for the theory of a wide-area secret communication system in his 1948 paper on information theory. In his papers, Shannon discussed the strength of Vernam-style encryption, scientifically proved the impossibility of deciphering such cryptograms, and scientifically demonstrated a precise lower limit on the length of a secret key transmitted to a user over a secret communication network. Shannon's 1948 paper led to an increase in the number of scientific papers in the field of cryptology. Though noteworthy, his 1949 publication did not significantly boost the quantity of scientific papers published in the subject of cryptology. The reason for this is that Shannon's theory of communication in secret systems is based on a secret key, and this is due to the difficulty of solving the problem of communicating the secret key to the user (Babenko and Ischukova 2012).

In 1976, W. Diffie and M. E. Hellman's article "A New Direction in Cryptography" led to the development of open scientific work in this area rising to a very high level. In their work, they created the foundations of a scientific and practical method that does not require the transmission and reception of a secret key between users of the system through a specially protected communication network when encrypting and decrypting data in secret communication systems, creating the era of public (non-secret) key cryptography, which continues to develop today and is becoming increasingly relevant. It should be noted that R. K. Merkley W. Diffie and M. E. Hellman's article, which was submitted for publication to another scientific journal independently, but almost simultaneously with them, also outlined the basics of the idea of public key data encryption. But R. K. The fact that Merkley's article remained unpublished for so long almost deprived him of his copyright.

No matter how complex and reliable a cryptographic system based on an algorithm is, there remains an important delicate issue that arises in its practical application, namely, the issue of distributing keys among users of cryptographic systems. In fact, the key for interaction between users of a cryptographic system, which ensures secret communication in an information system, is created by one of them and must be secretly transferred to another. It follows that, in general, another cryptosystem must be used to deliver (transmit) the key (Federal Information Processing Standards Publication 198 2002; Chabaund and Joux 2000; Matusiewicz 2007; NESSIE Consortium 2002). To solve this problem, a tendency arose to create cryptosystems with an open key based on the achievements of classical and modern science and technology, in particular, the achievements of the science of higher algebra. The essence of cryptosystems with an open key is as follows:

1. Each user of the cryptographic algorithm of the information system creates (builds) two keys associated with a certain rule.
2. One of these generated keys is announced publicly, and the other is kept secret.
3. The original plain text is encrypted with the public key of the user receiving the information and transmitted to this user, while knowing this public key, there is no way to find the corresponding secret key—that is, there is no way to decrypt the encrypted text (cryptogram).
4. The transmitted (delivered) cryptogram is decrypted with a secret key known only to the real owner of the cryptogram.

Public key cryptosystems are based on functions or algorithms whose inverses do not exist or whose inverses cannot be compensated for by modern science and technology, with very large material costs and with a large reserve of time in the future. Such functions are called one-way functions and have the following property: for a given x-value f(x), the value of the function y = f(x) is calculated quite easily, but if the value of the function y for an unknown x-value is known, then calculating the x-value is quite impossible in terms of time and material.

Algorithms of public key cryptosystems are distinguished by the one-way functions that underlie them. But any one-way function does not allow for the convenient creation of public key cryptosystems and, on their basis, for constructing an algorithm for creating a secret communication service in an existing information system.

This gives rise to the following requirements that are important for public key cryptosystems that provide reliable information protection:

1. The transformation of the original plaintext into the form of encrypted text is a one-way process, and decryption using the encryption key cannot be decrypted, i.e., knowledge of the encryption key is not enough to decrypt the encrypted text.
2. Based on knowledge of the public key, the costs and time required to determine the secret key using modern science and technology are not feasible.

As mentioned above, in asymmetric cryptosystems, the public and private keys are related by such a mathematical organic law that the problem of finding the private key using the public key is based on the difficulties of solving mathematical problems that are difficult to solve or have no solution efficient algorithm (Abdurakhimov et al.

2021a, b, 2022). For example, expressing a given number in a sufficiently large digit as a product of prime numbers or solving discrete logarithmization problems in a finite field.

Cryptanalysis of existing asymmetric cryptosystems shows that they are based on the difficulties of solving the following problems for which there are no efficient solution methods in mathematics:

1. Dividing a given number by prime factors (factorization).
2. Discrete logarithmization in a finite field.
3. Solving a system of linear equations in a finite field.
4. The "Backpack" problem.
5. Performing operations on points of an elliptic curve defined in a finite field.

Examples of popular asymmetric cryptosystems include the following:

Based on the problem of dividing a given number by prime factors: RSA, Williams, Pohlig–Hellman, and other encryption algorithms.

1. Based on the complexity of the discrete logarithm problem in a finite field: DSA, EL-GAMAL, GOST 34.10-94, and other electronic digital signature (EDS) algorithms.
2. The McAleese encryption algorithm, based on the complexity of the problem of solving a system of linear equations in a finite field.
3. The Merkle–Hellman encryption algorithm, based on the complexity of the "backpack" problem.
4. The task of performing operations on points of an elliptic curve is based on complexity: we can cite ECDSA 2000, GOST R 34.10-2001, and existing standard DSA algorithms.

Symmetric block cipher algorithms are considered an important component of all cryptographic systems. Symmetric block cipher algorithms provide data confidentiality by dividing data into blocks of a certain length and repeating operations on them.

The main concepts associated with symmetric block cipher algorithms are as follows:

Block length. The length of the input data to be divided, usually 64 bit for Feistel network systems.

Key length. The length of the keys used to encrypt and decrypt data.

Number of rounds. A property of block cipher algorithms that specifies how many times the same operation is performed on the same block of data using different keys. For example, the number of rounds in the DES encryption algorithm is 16.

Round function. An important part of encryption and decryption algorithms is listed, consisting of a sequence of operations performed in each round. This function usually has two input values: a piece of data and a key.

Round key. The key used for encryption and decryption is calculated and generated by a specific algorithm based on the original key.

Encryption modes. This is a characteristic of symmetric block cipher algorithms based on the use of block cipher algorithms based on certain methods. In this case, the encryption algorithm does not change, but the system is built on this algorithm.

Initialization vector (IV). This is the required size when using block cipher algorithm modes, and it is not kept secret.

Encryption function. A function used to covertly transmit blocks of plain data.

Decryption function. A function used to convert ciphertext blocks to plaintext.

Modern symmetric encryption algorithms usually use the following mathematical operations:

XOR operation;

addition operation mod2^32;

cyclic operations of swiping left or right;

changing seats at tables;

logical operations AND, OR, NEGATION;

and so on.

According to the creation feature, modern block encryption algorithms are divided into symmetric encryption algorithms based on the following three main directions:

based on the Feistel network;

Based on the SPN (substitution-permutation network);

Based on the Lai-Messi architecture.

Symmetric block encryption algorithms based on the Feistel network. According to this approach, the plaintext blocks are divided into two equal parts, one half (usually the right part) is subject to key operations, and the other part is combined with the XOR operation (Saymanov 2024; Kabulov et al. 2021a, 2021b). After that, the sides of the two parts are swapped. According to the number of rounds specified in the algorithm (0, 1, ..., n), the round keys (k_0, k_1,…,k_n) are generated. The rounding function is a function f, which takes two parameters (Fig. 1.1).

The basic operations of the Feistel network are as follows:

the plaintext block is divided into two equal parts (L_0, R_0);

for each round the following is done: $L_{i+1} = R_i$, $R_{i+1} = L_i \oplus f(R_i, k_i)$; then the ciphertext is obtained: (R_{n+1}, L_{n+1}).

When decrypting using the keys from the ciphertext block (R_{n+1}, L_{n+1}) in reverse order ($i = n, n - 1, ..., 0$), the following operations are performed: $R_i = L_{i+1}$, $L_i = R_{i+1} \oplus f(L_{i+1}, k_i)$. Then the plaintext pair (L_0, R_0) is obtained. The difference between this architecture and others is that the inverse function of the encryption function is not required to decrypt the data, it is limited to using the keys in reverse order.

In general, the functional diagram of the m-round Feistel network is expressed as follows:

The application of the Feistel network can be found in many symmetric block cipher algorithms. Examples of such cryptographic algorithms include FEAL, LOCI, Khufu, Khafre Blowfish, Lucifer, CAST, as well as standard algorithms such as DES, GOST 28,147-89.

Block symmetric encryption algorithms based on SPN (Substitution–Permutation Networks). According to this architecture, a plaintext block and keys are accepted as

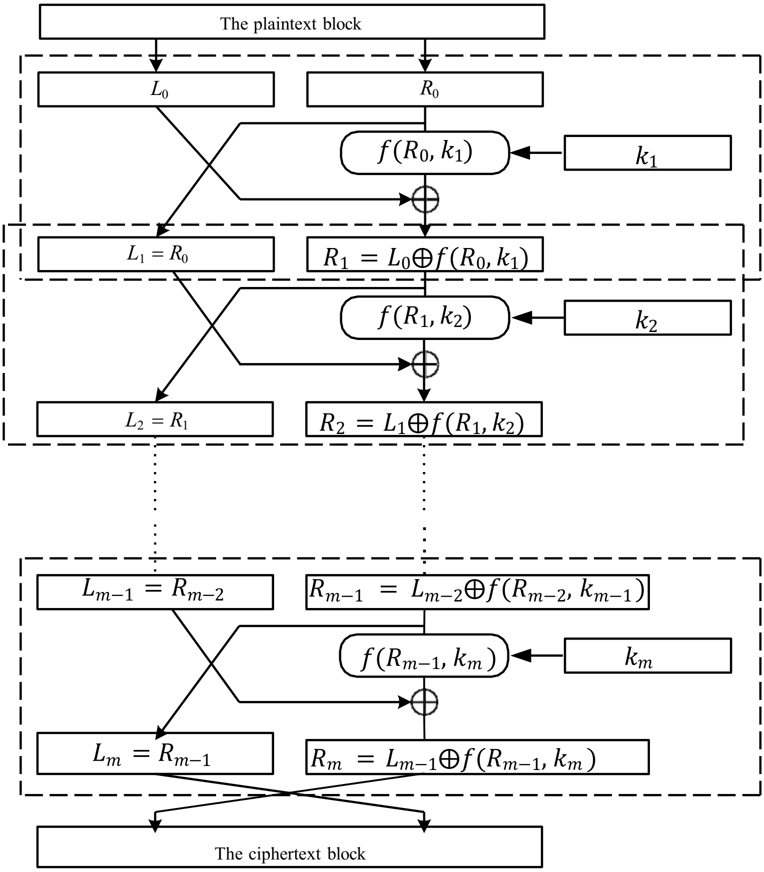


Fig. 1.1 Feistel network architecture

input parameters, and the ciphertext is obtained using substitution and substitution operations. It uses two types of tables: substitution tables (S-box) and permutation tables (P-box) (Fig. 1.2).

Each round uses separate keys to perform the swap and replace operations.

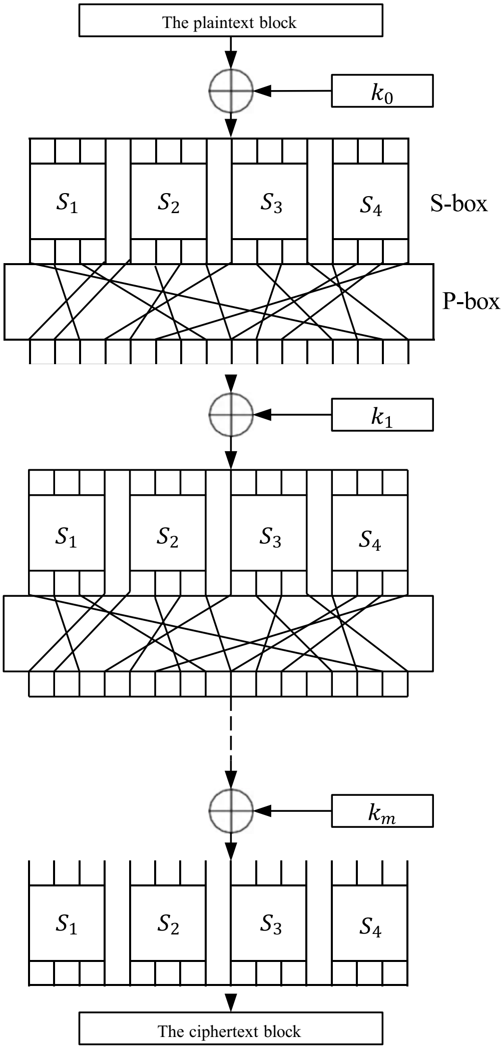
Unlike the Feistel network, decryption is performed using both the order of the keys and the tables inverted into tables.

Examples of algorithms based on this architecture are Rijndael, Kuznechik, Serpent, SQUARE, Own Dst 1105:2009, Bel-T, Kalina, CRYPTON, etc. which can be brought.

Lai–Massey ciphers (Lai–Massey ciphers). Cryptographic systems based on this method are similar to the Feistel network, except that the rounding function is irreversible. In this method, the input block is divided into two parts (Fig. 1.3).

Here F is a round function, H is a semi-round function, and k_0, k_1, ..k_n are the round keys corresponding to 0, 1,…,n.

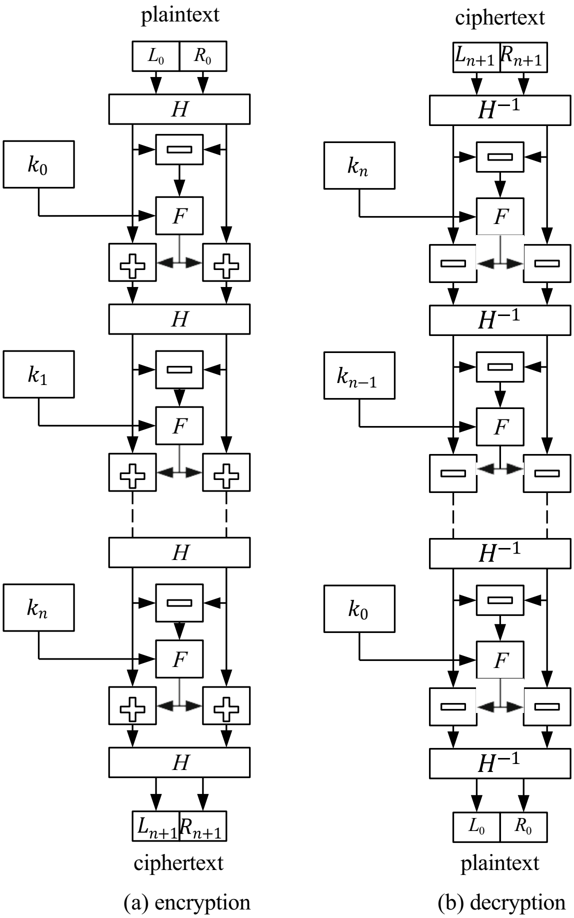
Fig. 1.2 SPN network architecture



Ciphers based on this method include IDEA, MESH, RIDEA, WIDEA-n, FOX/IDEA-NXT, REESSE3+, Bel-T, etc., giving examples.

Symmetric block ciphers consist of various transformations, and in general they can be divided into linear and nonlinear types.

Fig. 1.3 Lai–Massey network architecture



1.3 Elliptic Curve Cryptography

In recent years, elliptic curves have been widely used in cryptography. In this section, we will get acquainted with general information about elliptic curves, their place and properties in the coordinate system, and with operations performed on finite fields over points with rational coordinates lying on them.

Definition An elliptic curve obtained in a finite field, called the Weierstrass equation, is defined by the following equation:

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \tag{1.1}$$

where $a_1, a_2, a_3, a_4, a_5, a_6 \in K$.

An elliptic curve is usually denoted by E or E/K, and the points belonging to the elliptic curve, i.e., the solutions of Eq. (1.1) are called affine points of this elliptic curve.

Definition A point $\boldsymbol{P}(\boldsymbol{x}_0, \boldsymbol{y}_0) \in \boldsymbol{E}$ belonging to this elliptic curve $E{:}f(x, y) = y + a_1xy + a_3y - x^3 - a_2x^2 - a_4x - a_6 = 0$ is called smooth if one of the following conditions is true:

$$f_x'(x_0, y_0) \neq 0 \, \text{or} f_y'(x_0, y_0) \neq 0. \tag{1.2}$$

Definition An E (or E/K)-elliptic curve is called smooth if each of its affine points is smooth.

Example 1.1 For an elliptic curve $y^2 = x^3$, prove that the point (0, 0) is not smooth.

Solution

$$f(x, y) = y^2 - x^3, f_x' = -3x^2, f_x' = 2y.$$

Then we arrive at a contradiction with condition (2), as a result of which the point (0, 0) cannot be truly smooth.

Example 1.2 For an elliptic curve $y^2 = x^3 + x^2$, prove that the point (0; 0) is not smooth.

Indeed, $f(x, y) = y^2 - x^3 - x^2$, $f_x' = -3x^2 - 2x$, $f_y' = 2y$ becomes true, and we arrive at a contradiction with respect to condition (1.2). As a result, the point (0; 0) cannot be truly smooth. Its graph looks like the illustration shown in Fig. 1.4.

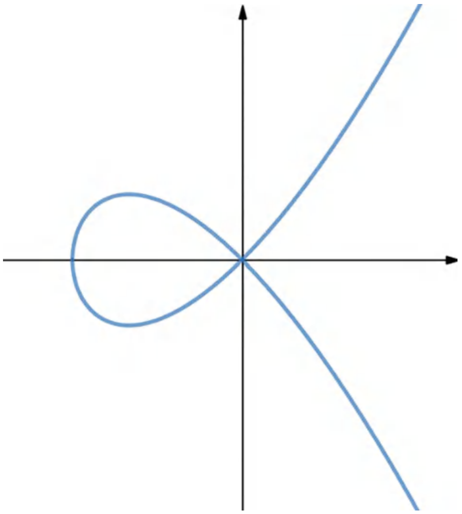


Fig. 1.4 Graph of an elliptic curve $y^2 = x^3 + x^2$

Methods for determining rational points belonging to an elliptic curve. Currently, the question of finding all rational solutions to the equation $y^2 = x^3 + ax^2 + bx + c$ remains unknown in mathematics. However, today the following two methods are used in cryptography science, although they are not very effective. We will get acquainted with these methods in detail below.

Method 1. By assigning values x_i to the chosen equation $y^2 = x^3 + ax + b$, it is checked that the right-hand side of the equation forms a perfect square. If some x_k form a perfect square, then the coordinates of the point corresponding to the equation

$$\left(x_k; y_k = \pm\sqrt{x_k^3 + ax_k + b}\right) \tag{1.3}$$

are fixed in pairs. This method gives good results if the coefficients of the equation are pre-conditioned. That is, it is a method of finding the point corresponding to this equation by creating an equation corresponding to the coefficients.

Example Let $a = 2$, $b = -3$, that is, the equation has the form: $y^2 = x^3 + 2x - 3$. According to formula (1.1), when $x = 2$, $y = \sqrt{2^3 + 2 * 2 - 3} = 3$.

Thus, the equation $y^2 = x^3 + 2x - 3$ has a rational point $P(2, 3)$.

Method 2. This method is used when a condition is set for the desired point. That is, having fixed the coordinates of the point (x, y) and one a-coefficient of the equation: $(a, x, y \in R)$,

$$b = y^2 - x^3 - ax. \tag{1.4}$$

The b-coefficient is calculated according to the formula and an equation is constructed based on this coefficient. This can be seen in the following example.

Example If $a = 2$, $P(x, y) = (1, 2)$, then according to formula (1.4): $b = 2^2 - 1^3 - 2 * 1 = -1$.

Thus, the point $P(1, 2)$ is a rational point corresponding to $y^2 = x^3 + 2x - 1$.

Adding rational points of elliptic curves. Let $E : y = x^3 + ax^2 + bx + c$ be the points$P(x_1, y_1)$, $Q(x_2, y_2)$ on an elliptic curve. Draw a straight line through these points. Then the drawn line intersects the E-curve at the third point. Move this $B(x_3, y_3)$ point symmetrically to the Ox–axis and the resulting:$B'(x_3, -y_3) = P(x_1, y_1) + Q(x_2, y_2)$ point. We declare the addition of points $P(x_1, y_1)$ and $Q(x_2, y_2)$ on an elliptic curve:

Figure 1.5 shows an example of a case where the equation $x^3 + ax^2 + bx + c = 0$ has one solution.

Explanation. Naturally, the E-curve does not always intersect the line that passes through points $P(x_1, y_1)$ and $Q(x_2, y_2)$. In this case, the intersection occurs at point 3. For instance, the line you wish to draw won't cross the curve at the third point (Fig. 1.7) if it is vertical to the Ox-axis through two points.

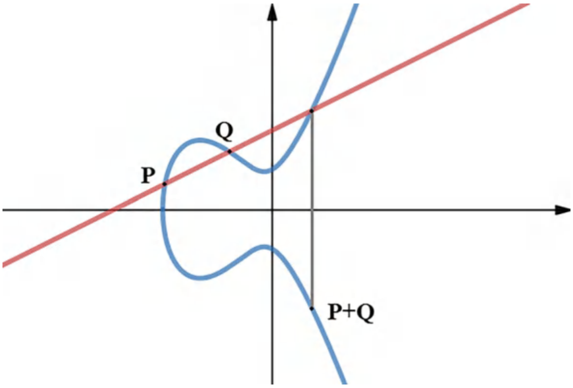


Fig. 1.5 Geometric representation of adding points

A line is taken into consideration individually if it passes through two points without intersecting the curve at the third point.

So, when $P(x_1, y_1) \neq Q(x_2, y_2) \neq 0$, we considered finding their sum $P(x_1, y_1) + Q(x_2, y_2)$. $P + P = ?$ How is this done? To do this, a trial line is drawn through the point P on the elliptic curve. This test line intersects the second part of the graph of the elliptic curve (part of the hyperbola) at some point. This intersection point moves symmetrically to the Ox-axis and is declared as $2P$ (Fig. 1.6).

Similarly, to find points $3P$, $3P = P + 2P$, etc., the search for points $4P = P + 3P$, $5P = 4P + P$ is also performed as above (Fig. 1.7).

Formulas for addition of points of an elliptic curve. According to what was discussed above, if $P, Q \in E$ are points. Passing through them, the cutting line intersects the E-curve at some third point $R(x_3, y_3)$.

Equations for adding elliptic curve points. Considering the previous discussion, if $P, Q \in E$ are points. The cutting line crosses the E-curve at a third point, $R(x_3, y_3)$, after passing through them.

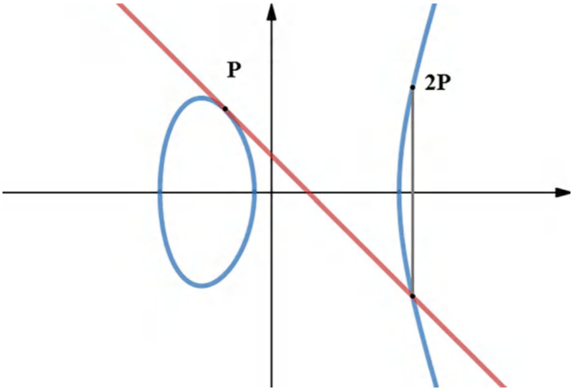


Fig. 1.6 Geometrical view of $2P$

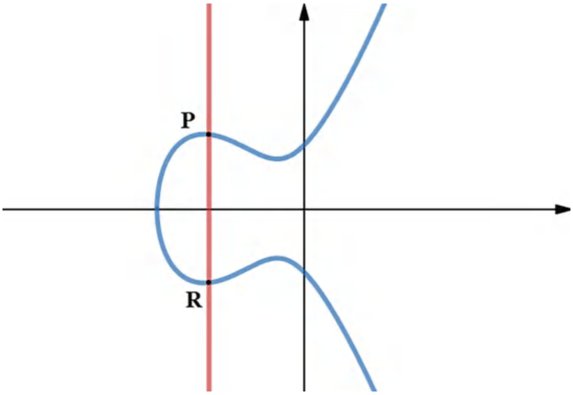


Fig. 1.7 Geometrical view of points vertical to the axis

Confirmation. If the points $P, Q \in E$ are rational points, then the point $R(x_3, y_3)$ is also a rational point.

From the proof of this statement, we can derive a formula for calculating the coordinates of the sum of points $P + Q$. Point $P + Q$ will be formed by moving point R—symmetrically to the Ox-axis. As a result, if we designate the coordinates of the desired point as (u, v), then these coordinates are found using the following formulas:

$$\begin{aligned} u &= k^2 - a - x_1 - x_2, \\ v &= -ku - d = -(k(u - x_1) + y_1), \end{aligned}$$

because $u = x_3, v = -y_3$. If the expression $k = \frac{y_1 - y_2}{x_1 - x_2}$ is substituted into these formulas we will have equality:

$$\begin{aligned} v &= \frac{y_1 - y_2}{x_1 - x_2}(-u + x_1) - y_1. \\ u &= \frac{(y_1 - y_2)^2}{(x_1 - x_2)^2} - (a + x_1 + x_2). \end{aligned} \tag{1.5}$$

These formulas make sense when $x_1 \neq x_2$.

If $x_1 = x_2$, then $y = kx + d$ is checked as a line and we arrive at the following formula:

$$\begin{aligned} u &= -2x_1 - a + \frac{(3x_1^2 + 2ax_1 + b)^2}{4y_1^2}, \\ v &= -y_1 - \frac{3x_1^2 + 2ax_1 + b}{2y_1}(u - x_1). \end{aligned} \tag{1.6}$$

Thus, if at least one P-rational point is a point of an elliptic curve, then using formulas (1.5) and (1.6) we can find $2P$, $3P$, $4P$, etc.

It should be noted that formulas (1.5) and (1.6) were derived with respect to equation $y^2 = x^3 + ax^2 + bx + c$. Now we present the formulas for adding rational points of an elliptic curve for the equation $y^2 = x^3 + ax + b$, which is widely used in cryptography:

$$\begin{aligned} u &= \frac{(y_1 - y_2)^2}{(x_1 - x_2)^2} - x_1 - x_2 \\ v &= -y_1 + \frac{y_1 - y_2}{x_1 - x_2}(x_1 - u), \end{aligned} \tag{1.7}$$

where $x_1 \neq x_2$.

If $x_1 = x_2$, then

$$\begin{aligned} u &= \frac{(3x_1^2 + a)^2}{4y_1^2} - 2 \\ v &= -y_1 - \frac{3x_1^2 + a}{2y_1}(x_1 - u). \end{aligned} \tag{1.8}$$

Example If the equation of an elliptic curve is $y^2 = x^3 - 2$, then the point $P(3, 5)$ is the following points: $2P = ?$, $3P = ?$, $4P = ?$, $5P = ?$ to be found.

Solution According to formula (1.6):

$$\begin{aligned} &y^2 = x^3 + ax^2 + bx + c, a = 0, b = 0; \\ &u = -2x_1 - a + \frac{(3x_1^2 + 2ax_1 + b)^2}{4y_1^2} = \frac{129}{100}; \\ &v = -y_1 - \frac{3x_1^2 + 2ax_1 + b}{2y_1}(u - x_1) = -\frac{383}{100}; \end{aligned}$$

therefore $2P = \left(\frac{129}{100}; -\frac{383}{100}\right)$.

As a result, using formula (1.6), we can calculate $3P$, $4P$, $5P$, i.e., if we take the first coordinate of the point $u_n = nP$, then:

$$\begin{aligned} &u_1 = 3, u_2 = \frac{129}{100}, u_3 = \frac{164323}{29241} \\ &u_4 = \frac{2340922881}{58675600}, u_5 = \frac{307326105747363}{160280942564521}. \end{aligned}$$

If we continue these calculations, then when we move to u_{11} we will encounter a 71-digit number.

The order of points of an elliptic curve.

In the previous section, finding nP if $P \in E$, i.e.,

$$nP = P + P + \cdots + P$$

was considered. However, the following two situations may occur during this addition process:

In any n-th step, the equality $nP = 0$ can be fulfilled;

$2P, 3P, 4P$, etc. nP-points can have different values.

Definition If $mP \neq 0$, all $m < n$ are satisfied and $nP = 0$, then we say that the P-point has n-finite order.

Example $y^2 = x^3 + 4$, show that the point $P(0, 2)$ has order $n = 3$.

Solution In fact, $a = b = 0$, according to formula (1.5):

$$\begin{aligned}
&u = 0\\
&v = -2\\
&2P = (0; -2), 3P = 2P + P = (0; -2) + (0; 2) = (0; 0)
\end{aligned}$$

and this coincides with the definition of the order of a point of an elliptic curve, i.e., $n = 3$. You can check this directly:

$y^2 = x^3 + 1, P(2, 3), n = 6.$

$y^2 = x^3 - 43x + 166, P(3, 8), n = 7.$

Explanation. In 1901, the French mathematician A. Poincaré (1854–1912) put forward the following conjecture.

Conjecture. It is always possible to find finite rational points of infinite order $P_1, P_2, ...P_k$ such that any P-rational point is represented by these rational points:

$$P = n_1 P_1 + \cdots + n_k P_k + Q.$$

Here $n_1, ...n_k$ are integers defined as single valued for the point P, and Q is a point of finite order. The number k here is called the color of the curve.

In 1922, the English mathematician L. Mordell proved the Poincaré conjecture. However, this proof did not provide a method for determining the color of the curve. Only in 1995 was it shown that the color of an elliptic curve can be found using a very complex analytical construction.

Definition The order of the point P belonging to E EC is the smallest natural number k satisfying the condition $[k]P = O$.

Add points has the following properties:

$$[0]P = O;$$

$$[1]P = P;$$

$$[n + m]P = [n]P + [m]P;$$

$$[-n]P = -([n]P).$$

It is known that if $z \neq 0$, the equation $y^2 = z(\mathrm{mod}\, p)$ can have two solutions. If these solutions are y_1 and y_2, then the equality $y_1 = -y_2$ is satisfied. If $y_2 < 0$, then $y_2 + p$ is also a solution to the above equation.

The orders of the points P and $(-P)$ are equal.

1.4 SHA256

A function that translates data of any length M to a fixed-length value $H(M)$ is called a hash function. Here, “fixed length” refers to 64, 128, 160, 256, or 512 bits.

The main purpose of practical use of hash function algorithms:

1. Control of its completeness during data transmission and storage.
2. Authentication of the data source.

Today, very general requirements are imposed on hash function algorithms:

1. Support for text of arbitrary length.
2. Outputs the assigned value.
3. Simple calculation of $H(x)$ for any given x.
4. Impossibility of calculating x from the equation $H(x) = h$ for any given function H (one-sidedness property).
5. $H(x) \neq H(y)$ for the received texts x and $y(y \neq x)$ (collision resistance property).

In general, hash function algorithms are divided into two types:

1. Keyed hash functions.
2. Keyless hash functions.

Keyed hash function is used in systems with a symmetric key. They guarantee the authenticity of the source, the completeness of information without additional tools in a system of users who trust each other.

Keyless hash functions guarantee the completeness of information using additional means (encryption or digital signature). These hash functions are used in a system of users who trust each other and do not trust each other.

In the United States, in 1993, the NSA and NIST jointly created the SHA-0 hash function algorithm in accordance with the FIPS PUB 180 standard. Soon, this algorithm was cracked by a group of cryptographers. Therefore, in 1995, in accordance with the FIPS PUB 180-1 standard, a certain part of the SHA-0 algorithm was modified and the SHA-1 algorithm was developed.

Later, a new encryption standard with key lengths of 128, 192, and 256 bits was developed in the USA, and due to the rapid development of technology, it became necessary to create new hash function algorithms with such tolerance. Therefore, in 2002, a new American hash function standard FIPS PUB 180-2 was adopted. The four hash functions defined by this standard are the $SHA-224$, $SHA-256$, $SHA-384$, and $SHA-512$ algorithms.

Below we will consider the SHA-256 hash function algorithm. In this algorithm, the input data length is less than 2^{64} bits, and the hash value length is 256 bits. This algorithm can be divided into two parts: a compression function and a data processing algorithm. The compression function consists of an encryption algorithm for an intermediate hash value of 256 bits, taking the next block of text as a key. In the compression function, in addition to the previous notations, the following notations are used: R^n-cyclic shift of the word by n bits up to ten, S^n-arithmetic (logical) shift of the word to the right by n bits. The word size is 32 bits, and addition is $(a+b)mod2^{32}$. Eight 32-bit words that represent the fractional portions of the square roots of the following prime numbers make up the original hash vector:

$$\{\ \text{6a09e667, bb67ae85, 3c6ef372, a54ff53a,}$$
$$\text{510e527f, 9b05688c, 1f83d9ab, 5be0cd19}\}\ .$$

Further calculations are carried out in the following steps:

Step 1. Primary processing. The hashed data is padded until its length is a multiple of 512, similar to SHA-1. When padding, a 1 is written after the data, and the remaining bits are filled with zeros. In this case, the data length is padded to a value comparable to 448 by 512 modules. Then, the 64-bit length of the data is written.

Step 2: Splitting data into 512-bit blocks. Extended data is divided into 512-bit blocks.

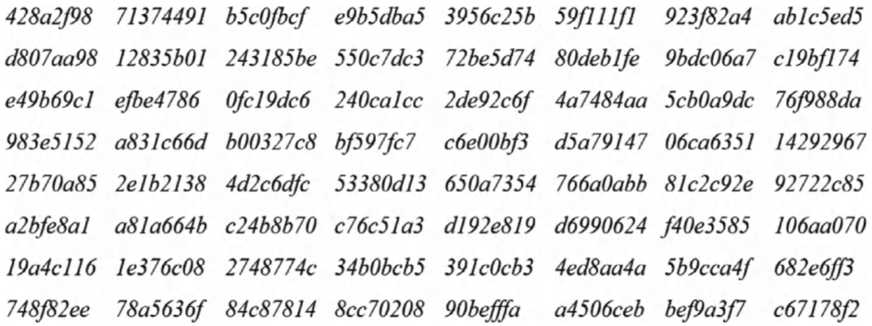
Stage 3. Main cycle. This loop uses six Boolean functions with 32-bit arguments and values:

$$\begin{aligned}
&Ch(x,y,z) = (x \wedge y) \oplus (\neg x \wedge z),\\
&Maj(x,y,z) = (x \wedge y) \oplus (x \wedge z) \oplus (y \wedge z),\\
&\Sigma_0(x) = S^2(x) \oplus S^{13}(x) \oplus S^{22}(x)\\
&\Sigma_l(x) = S^6(x) \oplus S^{11}(x) \oplus S^{25}(x),\\
&\sigma_0(x) = S^7(x) \oplus S^{18}(x) \oplus R^3(x),\\
&\sigma_l(x) = S^{17}(x) \oplus S^{19}(x) \oplus R^{10}(x).
\end{aligned}$$

The $M^{(i)}$ block is divided into 16 32-bit $M_0^{(i)}M_1^{(i)}...M_{15}^{(i)}$ words, and W_0, ..., W_{63} are defined as

$$\begin{aligned}
&W_j = M_j^{(i)}, j = 0, ..., 15\\
&\text{for } j = 16 \text{ to } 63\big\{W_j = \sigma_1\big(W_{j-2}\big) + W_{j-7} + \sigma_0\big(W_{j-15}\big) + W_{j-16}\big\}.
\end{aligned}$$

The first 32 bits of the fractional parts of the cube roots recovered from the following 64 16 values are taken as constants K_0, ..., K_{63}:



428a2f98	*71374491*	*b5c0fbcf*	*e9b5dba5*	*3956c25b*	*59f111f1*	*923f82a4*	*ab1c5ed5*
d807aa98	*12835b01*	*243185be*	*550c7dc3*	*72be5d74*	*80deb1fe*	*9bdc06a7*	*c19bf174*
e49b69c1	*efbe4786*	*0fc19dc6*	*240ca1cc*	*2de92c6f*	*4a7484aa*	*5cb0a9dc*	*76f988da*
983e5152	*a831c66d*	*b00327c8*	*bf597fc7*	*c6e00bf3*	*d5a79147*	*06ca6351*	*14292967*
27b70a85	*2e1b2138*	*4d2c6dfc*	*53380d13*	*650a7354*	*766a0abb*	*81c2c92e*	*92722c85*
a2bfe8a1	*a81a664b*	*c24b8b70*	*c76c51a3*	*d192e819*	*d6990624*	*f40e3585*	*106aa070*
19a4c116	*1e376c08*	*2748774c*	*34b0bcb5*	*391c0cb3*	*4ed8aa4a*	*5b9cca4f*	*682e6ff3*
748f82ee	*78a5636f*	*84c87814*	*8cc70208*	*90befffa*	*a4506ceb*	*bef9a3f7*	*c67178f2*

The first 32 bits of the fractional components are the cube roots that were taken from the numbers.

The main cycle is as follows:

```
for i=0 to N
{ // N is the number of blocks of extended data.
// registers a, b, c, d, e, f, g, h  with the intermediate value of the hash function (i-1)
// acquire (initialize) values.
  $a = H_1^{(i-1)}; b = H_2^{(i-1)}; \lesssim c = H_3^{(i-1)}; d = H_4^{(i-1)}; e = H_5^{(i-1)}; f = H_6^{(i-1)}; g = H_7^{(i-1)}; h = H_8^{(i-1)};$ // a, b, c, d, e, f, g and h should have the compression function applied.
for i=0 to 63
{   // Ch (e,f,g), Maj (a,b,c), $\Sigma_0(a)$, $\Sigma_1(e)$ va $W_j$ are calculated..
        $T_1 = h + \Sigma_1(e) + Ch(e, f, g) + K_j + W_j$
        $T_2 = \Sigma_0(a) + Maj(a, b, c)$
  $h = g; g = f; f = e; e = d + T_1; d = c; c = b; b = a; a = T_1 + T_2$
}
// Calculation of $i$ - intermediate hash value $H^{(i)}$.
    $H_1^{(i)} = a + H_1^{(i-1)}; H_2^{(i)} = b + H_2^{(i-1)}; H_3^{(i)} = c + H_3^{(i-1)}; H_4^{(i)} = d + H_4^{(i-1)};$
    $H_5^{(i)} = e + H_5^{(i-1)}; H_6^{(i)} = f + H_6^{(i-1)}; H_7^{(i)} = g + H_7^{(i-1)}; H_8^{(i)} = h + H_7^{(i-1)}$
}
```

The result is $H^{(N)} = H_1^{(N)}||H_2^{(N)}||H_3^{(N)}||H_4^{(N)}||H_5^{(N)}||H_6^{(N)}||H_7^{(N)}||H_8^{(N)}$ expression gives the hash value of M data.

Table 1.1 Lists some hash values computed with the SHA256 hash function algorithm

M value	H(M) hash value
0	5feceb66ffc86f38d952786c6d696c79c2dbc239dd4e91b46729d73a27fb57e9
1	6b86b273ff34fce19d6b804eff5a3f5747ada4eaa22f1d49c01e52ddb7875b4b
00	f1534392279bddbf9d43dde8701cb5be14b82f76ec6607bf8d6ad557f60f304e
11	4fc82b26aecb47d2868c4efbe3581732a3e7cbcc6c2efb32062c08170a05eeb8
a	ca978112ca1bbdcafac231b39a23dc4da786eff8147c4e72b9807785afee48bb
b	3e23e8160039594a33894f6564e1b1348bbd7a0088d42c4acb73eeaed59c009d
A	559aead08264d5795d3909718cdd05abd49572e84fe55590eef31a88a08fdffd
B	df7e70e5021544f4834bbee64a9e3789febc4be81470df629cad6ddb03320a5c

A selection of hash values computed with the SHA256 hash function algorithm is shown in Table 1.1.

1.5 Hashing: One-Way Function, Deterministic, Fast Computation, Avalanche Effect, Must Withstand Collisions

The hash function is calculated by successive reflections and iterative mechanisms over blocks of data. The reflections are performed over each block, they are called round reflections and are performed as round functions in the hash function. As a result of the reflections, the rounded hash value calculated at this step depends on all previous hash values and the data value in the current block. This results in the use of a cycle mechanism in the hash function.

In general, the round hash value of the block hash function H_i is calculated depending on two arguments of the hash function F, namely, the hash value of the previous round H_{i-1} and the value of the information in the current block Mi:

$$H_i = F(H_{i-1}, M_i), i = 1, 2, ...n, \tag{1.9}$$

where H_0 is a special constant (fixed) initial value.

Block ciphers or one-way functions can be used to increase the complexity of the F-function in block representations of the hash function. In addition, compression functions F are used, that is, functions that reflect the size (bit) of the information fed into the input block to a smaller size (bit) in the output block. And the variable H_i in each outgoing round is called a sliding variable.

In addition to round transpositions, XOR addition is also used to create hash functions. Figure 1.8 shows three different schemes for the general structure of such round reflections. Alternatively, a general scheme of the above variants can be used.

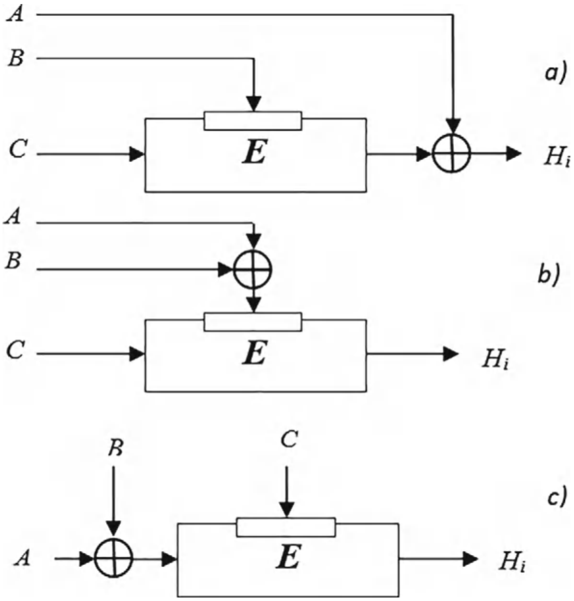


Fig. 1.8 Variants of the round hash function general construction scheme. (where: *a)*—is added to the output of the *E* function, *b)*—is added to the key of the *E* function, *c)*—is added to the input of the *E* function)

Usually, in the above schemes, H_{i-1}, M_{i-1}, or M_i values are used instead of the parameters A, B, C. Research shows that using a high-resistance encryption algorithm in a hash function is not a sufficient condition for constructing a strong hash function. Therefore, in order to increase the tolerance of hash function algorithms to existing cryptanalysis methods, firstly, the length of the input block $m \geq 128$, secondly, the F-function is selected as the hash function, in general, it is advisable to create a hash function algorithm in accordance with the formulas in Table 1.2.

Table 1.2 Possible formulas for constructing hash function schemes based on the durable block cipher algorithm *E*

№	Formula
1	$H_j = E_{H_{i-1}}(M_i) \oplus M_i$
2	$H_i = E_{H_{i-1}}(M_i \oplus H_{i-1}) \oplus M_i \oplus H_{i-1}$
3	$H_i = E_{H_{i-1}}(M_i) \oplus M_i \oplus H_{i-1}$
4	$H_i = E_{H_{i-1}}(M_i \oplus H_{i-1}) \oplus M_i$
5	$H_i = E_{M_i}(M_i \oplus H_{i-1}) \oplus H_{i-1}$
6	$H_i = E_{M_i}(M_i \oplus H_{i-1}) \oplus M_i \oplus H_{i-1}$
7	$H_i = E_{M_i}(H_{i-1}) \oplus M_i \oplus H_{i-1}$
8	$H_i = E_{M_i}(M_i \oplus H_{i-1}) \oplus H_{i-1}$
9	$H_i = E_{M_i \oplus H_{i-1}}(M_i) \oplus M_i$
10	$H_i = E_{M_i \oplus H_{i-1}}(H_i) \oplus H_{i-1}$
11	$H_i = E_{M_i \oplus H_{i-1}}(M_i) \oplus H_{i-1}$
12	$H_i = E_{M_i \oplus H_{i-1}}(H_{i-1}) \oplus M_i$

Round hash functions can also be built using these three parameters H_{i-1}, M_{i-1}, and M_i in several other variations. Such schemes use one backup register, but this situation allows for more secure schemes to be built.

In addition to block encryption, one-way functions are also used in building hash functions.

A one-way function is a function in which the input value cannot be calculated from the output value:

$$\begin{aligned} y &= F(x), \\ x' &= G(y), \\ x &\neq x, \end{aligned} \tag{1.10}$$

where f is a one-way function. For example, factorization, discrete logarithm functions are one-way functions.

A one-way function can also be constructed using the durable block cipher E-algorithm. For example:

$$F[x] = x \oplus E_k(x) \tag{1.11}$$

$$F[x, H] = H \oplus E_x(H) - SHA \tag{1.12}$$

$$F[x, H] = x \oplus E_H(x) - GOST28147 - 89, \tag{1.13}$$

where E is a symmetric block encryption algorithm in expression (1.11), a sequence of reflections known in expressions (1.12) and (1.13), k is a fixed key.

It was mentioned above that one-way functions can be built on the basis of the block cipher algorithm E. In practice, the argument of the one-way function is used as an input parameter for reflection several times. This does not always improve tolerance. For example, Fig. 1.11 shows a generalized diagram of *(b)* and *(c)* in Fig. 1.10.

The general formula for a hash function built using this scheme is as follows:$H_i = F(M_i \oplus H_{i-1}) \oplus M_i \oplus H_{i-1}$.

Deterministic

A hash function must be deterministic, meaning that it always produces the same hash value for a given input. In other words, the hash value must be a consistent result obtained only from the input after the mathematical function. This requirement rules out hashing functions that are influenced by external factors such as pseudo-random number generators and time of day. This also applies to cases where the behavior of the hash function is influenced by variables such as memory addresses, which can change during execution due to some entropy collection methods, but sometimes the addition of re-hashing can be applied. For example, in terms of function reuse, Python hash functions contain a random seed that is generated once at the beginning

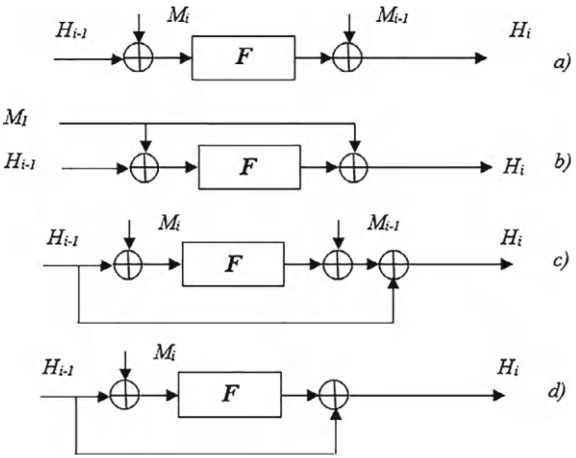


Fig. 1.10 A hash function scheme built on the basis of one-way block transformations

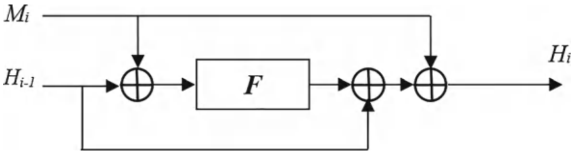


Fig. 1.11 F is a strong hash function created using a strong function

of the Python process. This random seed, combined with the input data, ensures that the Python hash function (SipHash) will behave like a true hash function during a single run. However, if the hash values are stored (e.g., written to disk), they will be invalid for further use, since the random seeds may be different the next time they are run.

Fast computation

Fast computation is essential for any hashing algorithm because many applications require real-time processing. The algorithm must be fast to meet these requirements. If the hashing algorithm is slow, it will take too long and reduce its usefulness in real-time scenarios, making it ineffective.

Avalanche Effect

The avalanche effect states that even a small change in the input should cause a large change in the output.

For example:

Text to hash: “SHA256”.

SHA256hash:

“b3abe5d8c69b38733ad57ea75e83bcae42bbbbac75e3a5445862ed2f8a2cd677”.

Text to hash: “SHA2560”.

SHA256hash:

“f52c3a68fa06510ce54be1bbf62a1f035b6ca2f84bd46fa0a98941a98e994cb0”.

The above example shows that although the changes in the input data are minimal, the two resulting output hash values are significantly different.

Must withstand collisions

Since the number of possible inputs is infinite and the number of possible outputs is finite, different inputs inevitably produce the same result, which is called a collision.

For example, with the SHA256 algorithm producing a 64-character output, although there are many possible combinations of those 64 characters, there are many more potential inputs. This means that collisions are possible when two different inputs produce the same hash value.

A good hashing algorithm should handle collisions effectively, ideally generating hash values in a way that minimizes or avoids collisions.

1.5.1 Digital Signature

At the end of any written letter or document, it is natural to have the signature of the author or the person responsible for its creation. This situation is usually caused by the following two reasons. First, the receiving party verifies the authenticity of the information by comparing the signature in the received information to the signature sample it has. Second, a personal signature legally guarantees authorship of the information document. Such a guarantee is especially important in sales, power of attorney, obligation, and similar transactions.

It is relatively difficult to forge personal signatures on documents, and the authors of personal signatures can be identified using modern advanced forensic methods. However, digital signature features are different and depend on the bits of memory registers, which are determined by the features of the binary number system. Copying or changing an electronic signature, which consists of a specific sequence of memory bits, is not difficult in computer-based communication systems.

In today’s highly developed global civilization, the use of documents, including confidential documents, in electronic form and their transmission in communication systems are widely used, which makes the issues of determining the authenticity of electronic documents and electronic signatures important.

Public key cryptographic systems, no matter how convenient and crypto-resistant they are, cannot fully solve the authentication problem. Therefore, it is necessary to use authentication methods and tools together with cryptographic algorithms in a complex manner.

In the following two cases (A) and (B) are considered, which show how the authentication system must protect against the adversary’s behavior toward its goal and mutual violations of the cryptosystem users’ usage protocol in the communication relationship.

Refusal. User (A) has actually sent information to User (B) and can reject the transmitted information. An electronic (digital) signature is used to prevent violations of such rules (disorder).

Modification. User (B) modifies received data and claims (claims) that user (A) sent the modified data.

Forgery. User (B) himself prepares information and claims that user (A) sent this fake information.

Active modification. A third user (V) illegally connects to the communication network of users (A) and (B) and transmits almost continuously, changing the information they are transmitting.

Masking. A third party (V) sends information to a user (B) on behalf of a user (A). In order to prevent violations of the rules of the communication system, such as: modification, falsification, active modification, masking, a digital signature is used—information consisting of a digital signature and a digital cipher that fully includes a part of the transmitted information.

Repeat. User (V) resends information sent by user (A) to user (B) to user (B). Such illegal behavior is used as a communication method to illegally use the electronic settlement system in the banking networks to rob other people's money. The following measures are taken to protect against such illegal methods.

- imitation tolerance - imitation tolerance;
- classification of information entering the cryptosystem based on protection purposes.

Electronic digital signature provides protection against several types of violations in communication systems, namely:

- if the secret key is known only to the user (A), then it cannot be denied that the information received by the user (B) was sent only by (A);
- does not allow the illegal (adversary) to modify, falsify, actively modify, mask and other similar rules of the communication system without knowing the secret key;
- eliminates many disagreements in connection with the interdependence of users of the communication system, and when such disagreements arise, it is possible to clarify them without an intermediary.

In most cases, there is no need to encrypt the transmitted data, it is necessary to confirm it with an electronic digital signature. In these situations, the recipient's private key is used to encrypt the plaintext, which is then delivered together with the ciphertext that was received. Using the sender's public key, the recipient can decrypt the ciphertext and compare it to the plaintext.

Asymmetric encryption algorithms are not widely used to encrypt large volumes of data, as demonstrated above. Asymmetric encryption algorithms are widely used in the field of cryptography, mainly in electronic digital signature systems.

Electronic digital signature algorithms perform the following tasks:

- the integrity of the signed information;

- ensures that the subject who digitally signs an electronic document does not deny authorship;
- determining the authenticity of the electronic document source.

The electronic digital signature system consists of the following two processes (Figs. 1.12 and 1.13):

- Creating an electronic digital signature;
- the electronic digital signature.

These days, RSA and El-Gamal-based electronic digital signatures are commonly utilized. However, there exist attack techniques designed to fabricate electronic digital signatures. The signature can be faked by factoring and solving the discrete logarithm problem, which yields the value of the signer's key using the signature verification key. As a result, digital signature algorithms for Edwards curve and Elliptic curve, known as EdDSA (Edwards curve Digital Signature Algorithm) and ECDSA (Elliptic Curve Digital Signature Algorithm), have been developed and are in use.

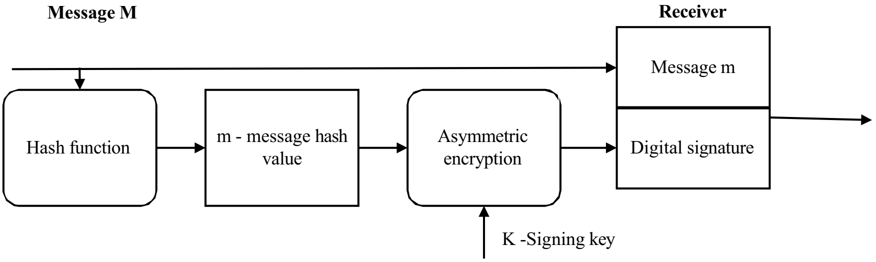


Fig. 1.12 The process of forming an electronic digital signature

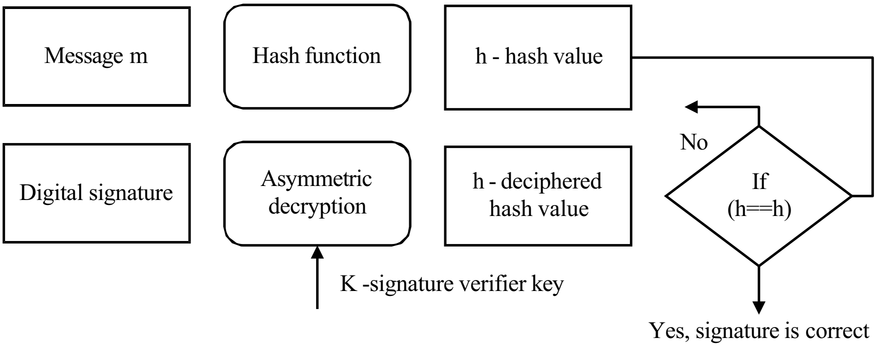


Fig. 1.13 Electronic digital signature verification process

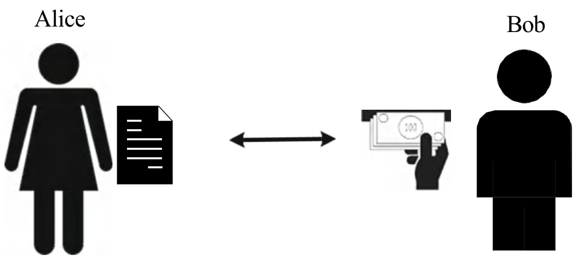


Fig. 1.14 Alice to Bob in himself the most good cookies recipe that there is cure is doing

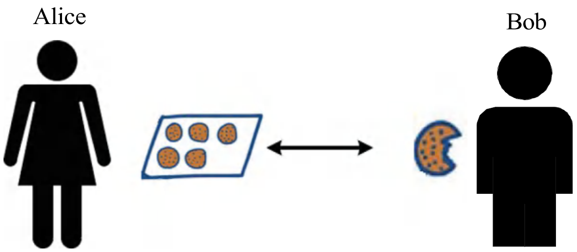


Fig. 1.15 Alice to Bob in himself the most good recipe that there is to prove for cookies prepared is giving

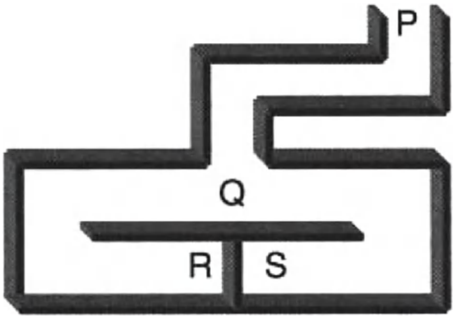


Fig. 1.16 Bob's Cave

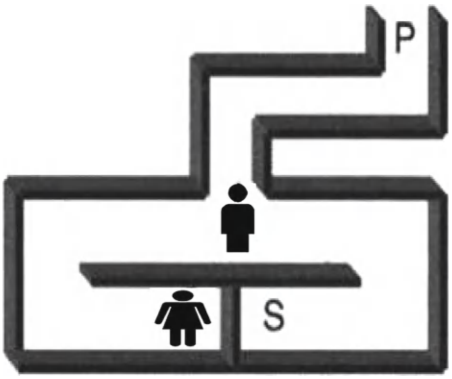
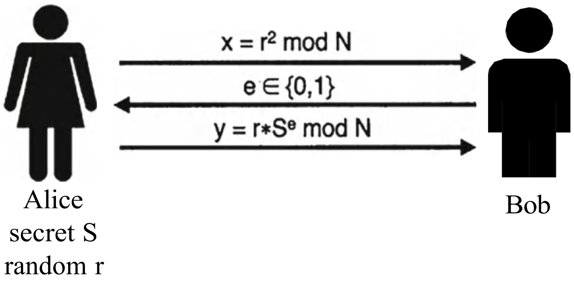


Fig. 1.17 Bob's Cave protocol

Fig. 1.18 Fiat–Shamir protocol



1.5.2 DSA Algorithm Based on RSA and El-Gamal

DSA based on RSA. An DSA algorithm based on the RSA algorithm can be implemented without too much difficulty. For this, it is enough to reverse the keys used for encryption and decryption and use its hash value instead of the data (Table 1.3).

DSA based on El-Gamal. In practice, DSA algorithms based on the El-Gamal encryption method are widely used. Key generation in DSA based on this method is done in the same way as in encryption. The signing and signature verification processes are as follows (Table 1.4).

In practice, many countries have their own DSA standards. They can get

- DSA standard based on El-Gamal (USA);
- GOST R 34.10–94 standard based on El-Gamal (Russia);
- ECA-based ECDSA 2000 standard (USA);
- ECA-based GOST R 34.10–2001 standard (Russia);

Table 1.3 DSA algorithm based on RSA

Formation of DSA	Checking the DSA
$H(M)^d$ mod $n = P$ Here: $H(M)$ is the hash value of information; d – signing key (secret key); P *is* a signature	P^e mod $n = H(M)$

Table 1.4 DSA based on El-Gamal

Formation of DSA	Checking the DSA
$a = g^k$ mod p $b = (H(M) - ax)k^{-1}$ mod $(p - 1)$ Here: $H(M)$ is the hash value of information; x—signing key (secret key); $GCD(k, p - 1) = 1$ is an integer (a, b)—signature	$y^a a^b$mod $= g^{H(M)}$ mod p the signature is valid if the equality holds, otherwise it is not Here: $y = g^x$ mod p— open key

- UzDSt 1092:2009 standard (Republic of Uzbekistan) based on parametric scaling and EC.

1.5.3 DSA Standard

NIST (National Institute of Standards and Technology) developed the DSS (Digital Signature Standard) DSA standard in 1991. It was based on the DSA (Digital Signature Algorithm) algorithm. The discrete logarithmization problem in a finite field serves as the foundation for this algorithm. The SHA-1 standard was used as the hash function.

Creating a signature:

1. Signatory M performs the following steps when signing information:
   a. p is a prime number ($2^{512} < p < 2^{1024}$ and 64 ga multiples of the bit length);
   b. q is a prime number ($2^{159} < q < 2^{160}$ and divisor of $p - 1$);
   c. $0 < h < p$ and $h^{(p-1)/q} modp > 1$ based on the quantity h satisfying the conditions $g = h^{(p-1)/q} modp$ is an integer;
   d. x—through the secret key, $y = q^x modp$ is the public key (here: $0 < x < q$);
   e. calculates the hash value of the data (H(M) is the hash value of the data in the interval [1; q]).
2. The data sender chooses a random number k (provided $0 < k < q$). This size will be removed after signing.
3. The signatures of M data are

$$r = g^k \bmod p \bmod q,$$
$$s = k^{-1}(xr + H(M)) \bmod q.$$

The resulting quantities (r, s) are added to the information M and sent to the signature verifier.

Signature Verification Process:

The process of signature verification is carried out based on the received information $M\prime$ and the signature attached to it $(r\prime, s\prime)$. It consists of two stages. If the signature fails the first-stage verification, it will not go to the second stage.

1. For received signatures, the condition $0 < s\prime < q$ or $0 < r\prime < q$ is checked. If this condition is met, the second stage will take place.
2. The second stage consists of
   a. $v = \left(s'\right)^{-1}(\mathrm{mod} q)$ is considered.
   b. $z_1 = H\left(M'\right)v \bmod q, z_2 = r'v \bmod q$ are values.
   c. then $u = g^{z_1} y^{z_2} modpmodq$ is the value.

i. If the equality $r\prime = u$ is fulfilled, then the electronic digital signature is valid $(M = M\prime)$. Otherwise, the signature will be considered a forgery.

1.5.4 GOST R 34.10-2001 DSA Standard

This algorithm is a modification of the GOST R 34.10-94 DSA algorithm based on an elliptic curve, a Russian standard. The processes of signature formation and verification in this algorithm are given below.

Signature formation process. As initial data: M the information to be signed, the ellipse parameters used, and the secret key for the signature. For this algorithm, the equation of the elliptic curve must be considered in the fundamental characteristic field satisfying the condition$F_p p > 2^{255}$. The signature is (r, s) equal to.

Steps to create a signature:

a. An arbitrary number k *in the interval* $1 \leq k \leq n - 1$ is chosen (where n is the order of points and G an integer satisfying the condition $2^{254} < n < 2^{256)}$.
b. $(x_1, y_1) = [k]$ is G.
c. $r = x_1 \text{mod}\, n$. If $r = 0$, return to step 1, and select the number k again.
d. The signer M calculates the hash value of the data, i.e., $e = H(M).IfH(M)modn = 0$, then the condition $H(M)$ mod $n = 1$ is obtained.
e. Based on the secret key d chosen from the interval $0 < d < n$, $s = (dr+ke)modn$ is calculated.
f. If$s = 0$, go back to step 1 and choose another number k.
g. The resulting pair of numbers (p, c) is an electronic digital signature for information M.

Signature verification process. Signature verification is performed on the basis of received public information $M\prime$ and signature $(r\prime, s\prime)$.

a. Verification is halted and the signature is deemed phony if $1 \leq r$', $s\prime \leq n - 1$ requirements are not satisfied.
b. $e = H\left(M'\right)$ is the hash value of the data.
c. $w = H\left(M'\right)(n - 2)$ mod n is the magnitude.
d. $u_1 = s'w$ mod q is a quantity.
e. $u_2 = \left(n - r'\right)w$ mod n is a quantity.
f. $X = \left[u_1\right]G + [u_2]Q = (x_1, y_1)$ quantity is calculated.
g. If x_1 mod is equal to $n = r$, then the signature is valid, otherwise it is considered to be forged.

In addition, many electronic digital signature algorithms are used in practice. Although the parameters and functions used in them vary, the mathematical problem used is based on one of the above.

a **Diffie–Hellman key exchange algorithm**

U. Diffie and M. Ye. This is what Hellman proposed as a one-sided function:

$$f(x) = \alpha^x(\text{mod}p). \tag{1.14}$$

We consider the discrete scaling function modulo$p$. As noted earlier, where x is an integer that can take values from 1 to $(p - 1)$; p—sufficiently large prime number; α is an integer that takes the values 1 to p and its powers $\alpha, \alpha^2, ..\alpha^{p-1}$ take the values *1,.. (p–1)* in some order. For example, if$p = 7, a = 3$, we have expressions, $p = 7, \alpha = 3$ if$\alpha = 3, \alpha^2 = 2, \alpha^3 = 6, \alpha^4 = 4, \alpha^5 = 5, \alpha^6 = 1$. In algebra, such a number is called a simple element of the finite field$GP(p)$, and it is known that such a number always exists.

If $y = f(x) = a^x$, then naturally it is an inverse function

$$x = f^{-1}(y) = \log_a y. \tag{1.15}$$

That is, finding x-values for given y's is called the problem of finding discrete logarithms. Even at sufficiently large values of p, for example, when $p = 2^{1000}$, it is possible to easily calculate the function $f(x)$ by carrying out operations of raising and multiplying by the square of no more than 1000.

If the discrete scaling function is truly one-sided, then it should be practically impossible to compute the expression $\log_a y$ for all values of y, that is, for all values of y that satisfy this inequality $1 \leq y \leq p$. M.Ye. Hellman and his student Polig, not only when p is a large prime, but also when $(p - 1)$ has a large prime multiplier q (or this q is multiplied by 2), according to the values of y of the function defined by expression (1.1) They showed that calculating the expression $\log_a y$ is complicated in practice. U. Diffie and M. Ye. For users of secret communication systems, Hellman created an algorithm for exchanging secret keys without a separate secret channel using discrete logarithms. According to this algorithm:

1. α and p numbers are known to all users.
2. Each user is, for example, i an integer between user 1 and $(p - 1)$ X_i chooses a number and keeps this number secret.
3. i–the user $Y_i = \alpha^{X_i}$(modp) calculates the value and enters this Y_i value into a public ledger that is verified by all users and is always available to them without keeping it confidential.
4. If is i the user of the secret communication system j wants to establish a secret communication with the user, i the user Y_i takes his private key from the public data book. X_i using

   $$Z_{ij} = (Y_i)^{X_i} = \left(\alpha^{X_j}\right)^{X_i} = (\alpha)^{X_i X_j}(\text{mod}p)$$

   calculates the value.
5. Likewise j– the user also Z_{ij} calculates. In this $Z_{ij} = Z_{ji}$ case, i and j users Z_{ij} can use the value as a secret key in a symmetric key cryptosystem that secures their

private communication. If an adversary could solve the problem of computing discrete logarithms, Y_i he would have a secret key i by taking $X_i = \log_\alpha Y_i$ and from the public ledger Y_j and $X_j = \log_\alpha Y_j$ calculating the values of and (Z_{ij} and j users).

It should be noted here that the open data book is open only to users of the secret information communication system.

As it can be seen from the above algorithm, although it is not fully proved theoretically, the adversary cannot calculate the value Z_{ij} by any other method. The given algorithm was developed by U. Diffie and M. Ye. It is called Hellman's public key distribution system. It was the first system to eliminate the need to transfer private keys over a private channel in a secure communication system, and it forms the basis of other robust and convenient public key cryptosystems today.

U. Diffie and M. Ye. Hellman's public key distribution system eliminates the need for the private key to be transmitted over a private channel like other public key cryptosystems, but does not solve the authentication problem (Table 1.5).

Here, the numbers p and g are known for both sides, and $p > g$ must satisfy the condition. The chosen numbers X_A and $X_B = 9$ are required to be in the interval$(2, ..., p - 2)$. Also, p is a large prime number.

This public key distribution protocol is not resistant to man-in-the-middle attacks (Table 1.6).

Security of Diffie–Hellman Key Distribution Algorithm. The security of this algorithm can be analyzed based on *a man-in-the-middle attack.* Because it knows the numbers a and p and tries to get the session key between Alice and Bob. The attacker also has open switches on both sides. The question is whether it is possible to calculate $k = \alpha^{ab}$ Here, $A = \alpha^a$ and $B = \alpha^b$. This problem is known as the Diffie–Hellman problem.

Table 1.5 An illustration of key exchange using the Diffie–Hellman algorithm

Alice	Evil Eve	Chapter
Alice and Bob generate two numbers $g, p(p > g).p = 11, g = 7$	The attacker also knows $p = 11, g = 7$.	Alice and Bob generate two numbers $g, p(p > g).p = 11, g = 7$
Alice generates her private key.$X_A = 6$		Bob generates his private key.$X_B = 9$
$Y_A = g^{X(A)}(\text{mod}p)$ $Y_A = 7^6 \text{ mod } 11 = 4$	→ ←	$Y_B = g^{X(B)}(\text{mod}p)$ $Y_A = 7^9 \text{ mod } 11 = 8$
Alice accepts $\text{Y}_\text{A} = 8$.	$Y_\text{V} = 4$, $\text{Y}_\text{A} = 8$ is also known to the intruder	Bob assumes $Y_V = 4$.
Private key = $Y_B{}^{X_A} modp$ Private key = $8^6 mod 11 = 3$		Private key = $Y_A{}^{X_B} modp$ Private key = $4^9 mod 11 = 3$

Table 1.6 Diffie–Hellman based on EC

Alice		Chapter
$n_A \in [1, n-1]$ a number satisfying the condition is selected $Q_A = n_A P$ calculates the public key BOB accepts from Q_B		$n_B \in [1, n-1]$ a number satisfying the condition is selected $Q_B = n_B P$ calculates the public key ALICE Receives from Q_A
Generating a public key		
$K = n_A Q_B$		$K = n_B Q_A$
$K = n_A Q_B = n_B Q_A = n_A n_B P$		
This method is immune to man-in-the-middle attacks just like the normal method		

An improved Diffie–Hellman problem. Given a finite cyclic group G. The numbers $\alpha \in G$ and $A = \alpha^a$ and $B = \alpha^b$ belong to G. Calculating α^{ab} is a Diffie–Hellman problem.

Discussions. The Diffie–Hellman key distribution scheme is standardized in ANSI X9.42 and is used in several protocols such as TLS. This algorithm is designed to exchange keys between two parties, and within a group, a combined Diffie–Hellman key distribution scheme can be used.

The El-Gamal public key encryption algorithm is used by GnuPG, OpenSSL, PGP, and other similar cryptographic programs.

b. Zero-Knowledge Proofs (ZKP)

Cryptography uses authentication information that is provided (known in advance to both parties) to prove its authenticity to the other party. Authenticating information is usually private or confidential information. It is also considered risky to provide complete authentication information to the other party when the time comes. In this context, providing a part of the authentication information without providing the complete information is called zero-knowledge proofs.

Through the use of “Zero-Knowledge Proofs”, one party can demonstrate to another (the verifier) that a statement is true while withholding any information other than the statement’s veracity.

Imagine that Alice has the best chocolate chip cookie recipe in the world and she wants to sell it to Bob. Bob needs to check if Alice really has the recipe, but if Alice shows it to him, he will find out about the secret recipe.

A ZKP proof is a way to guarantee that Alice actually has the recipe, but without revealing the actual recipe. How can this be done?

Alice in reality in it recipe that there is to prove for cookies cooks and Bob them taste sees. Bob them eats and they are good cookies such as to taste have that trust harvest does.

This is a coincidence that it is not trust harvest to do for Bob from Alice one how many times cookies cooking demand does and if they each trip the most good cookies such as to taste have If so, Bob is up level trust with Alice’s recipe there is said to the conclusion will come.

In zero-knowledge proofs, Alice tries to authenticate herself without telling Bob about the secret size even though she knows about it. In this case, Bob will be able to check Alice's secret message when he knows it and even if he doesn't have any information about it. At first glance, it doesn't seem possible. Bob confirms that Alice has a high likelihood of knowing the secret magnitude through an interactive probabilistic method. An interactive confirmation system looks like this.

Before analyzing the protocols of this category, let's first get acquainted with "Bob's Cave" (Fig. 1.3).

Assume Alice tells you she knows the secret word (the "open wire") that opens the door between R and S. Without uttering these words, is Alice able to persuade Bob?

Assume the subsequent protocol. Alice steps into Bob's Cave and flips a coin to choose if she should travel R or S. Bob then went into the cave and reached point Q. Assume Alice is located at position R in Fig. 1.4.

Alice is asked to be on the coin by Bob as he tosses it. Figure 1.7 shows that regardless of whether Alice is aware of the secret word or not, she is on side R if Bob selects it. However, Alice can only select side S if she is aware of the code word. Said another way, Alice has a $\frac{1}{2}$ chance of tricking Bob if she is unaware of the secret phrase. This might not seem very helpful, but this *probability* equals $(\frac{1}{2})^n$ if the protocol is repeated *n times*. As a result, Alice can consistently persuade Bob that she is aware of the secret phrase by having them perform this protocol n times.

Even if Alice (or Tridi) didn't know the secret phrase, she would still be able to convince Bob. However, Bob can reduce this probability sufficiently by choosing *n appropriately*. If *n* = *20*, in 1 out of 1,000,000 attempts, Alice will be able to prove the passphrase is valid without knowing it. Also, Bob can't learn anything in this protocol. Finally, one important aspect of this protocol is that it must be random in which direction Bob is asked to appear. If Alice or Tridi know in advance which way Bob will ask, the chances are even better.

Bob's Cave shows that zero-knowledge proofs exist. However, protocols of this type are not popular in practice. Can this be seen in action? Yes, the Fiat–Shamir protocol will be introduced below.

The Fiat–Shamir protocol aims to find the square root based on modulo *N in a factorization problem.* Assume p and q are prime numbers, such that $N = pq$. Alice knows the secret S and must keep it secret. N and $v = S^2 \mod N$ quantities are open. In this case, Alice needs to convince Bob without revealing knowledge about S.

Bob receives x from Alice. Bob $e \in \{0,1\}$ selects a random value and transmits it to Alice in the second message. Alice then determines the following: $y = rS^e \mod N$. Alice sends Bob a message in the third one. Bob then confirms that $y^2 = xv^e modN$.

$$\text{That is, } y^2 = r^2S^{2e} = r^2\left(S^2\right)^e = xv^e \mod N. \tag{1.16}$$

Bob transmits $e = 0$ or $e = 1$ in the second packet. Let's examine each instance in isolation. In the third message, Alice responds with $y = rS \mod N$ after Bob sends $e = 1$, thus equality (1.16) becomes

$$y^2 = r^2 S^2 = r^2 \left(S^2\right) = xv \bmod N.$$

In this case, Alice is required to know S.

In the alternative instance, equality (1.16) becomes: if Bob transmits $e = 0$ in the second message, Alice sends him $y = rmodN$ in the third message.

$$y^2 = r^2 = x \bmod N.$$

It is not necessary for Alice to know S in this instance. This is a serious problem, similar to the situation in Bob's Cave.

The random value e that is selected determines the protocol's security. Assume Tridi is aware that Bob is going to transmit the second message, $e = 0$. Next, in the first transmission, Tridi sends $x = r^2 modN$, and in the third, $y = rmodN$. In other words, Tridi does not need to be aware of the secret S.

In contrast, Tridi will transmit $x = r^2 v^{-1} modN$ in the first message and get $y = rmodN$ in the third message if he is aware that Bob would send $e = 1$ in the second message. Bob $y^2 = r^2$ and $xv^e = r^2 v^{-1} v = r^2$ computes and accepts the result as true in accordance with the protocol.

Consequently, Bob $e \in \{0,1\}$ has to be selected at random. If this is the case, Tridi has a probability of ½ to deceive Bob. After Bob completes n iterations, this probability decreases to $(1/2)^n$.

Using public key cryptographic systems to overcome these problems eliminates the possibility of anonymity for both parties. Protocols based on zero-knowledge proofs ensure anonymity of parties.

1.6 Conclusion

This chapter provides information about the main areas of cryptography. In particular, symmetric and asymmetric encryption algorithms, elliptic curve cryptographic algorithms, hash functions, digital signature algorithms, cryptographic key exchange algorithms, and zero-knowledge proof-based authentication methods were discussed. In general, the sections and methods of cryptography did not appear by themselves. Each of them was created based on a specific need. This idea is also applicable to the above sections. The need to ensure data confidentiality has existed since ancient times and encryption algorithms were created. Initially, symmetric encryption algorithms were created and asymmetric algorithms were created due to the problem of secure key sharing. Due to the low encryption speed of asymmetric algorithms, symmetric algorithms are still used today. While encryption algorithms are used to ensure data confidentiality, hashing algorithms are used to ensure data integrity. Algorithms belonging to the SHA family are mainly used for hashing data. After ensuring the confidentiality and integrity of the information, digital signature algorithms are used to authenticate it. Key exchange algorithms such as Diffie–Hellman are used to meet the need for secret transmission of encryption keys. Cryptography uses authentication information that is provided to the second party to confirm its authenticity. Since the authenticator's information is private or confidential information, it is considered dangerous to provide it entirely to the other party. In this context, zero-knowledge proofs are used for authentication by providing a portion of it without providing the full authentication information.

References

Abdurakhimov B, Boykuziev I, Khudoykulov Z, Allanov O (2021) Application of the algebraic cryptanalysis method to the Kuznyechik encryption algorithm. In: International conference on information science and communications technologies: applications, trends and opportunities, ICISCT 2021

Abdurakhimov B, Boykuziev I, Allanov O, Xidirov B (2021) Differential characteristics of reflections of Kuznyechik encryption algorithm. In: International conference on information science and communications technologies: applications, trends and opportunities, ICISCT 2021

Abdurakhimov B, Allanov O, Boykuziev I, Abdurazzokov J (2022) Application of artificial neural networks in the classification of classical encryption algorithms. In: 2022 International conference on information science and communications technologies, ICISCT 2022

Babenko L, Ischukova E (2012) Differential analysis GOST encription algorithm. In: Proccedings of the 3 international conference of security of information and networs (SIN 2012). ACM, New York, pp 149–157

Bruce Schneier, Applied Cryptography, Second Edition: Protocols, Algorthms, and Source Code in C (cloth), ISBN: 0471128457 Publication Date: 01/01/96

Chabaund F, Joux A (2000) Differential Collisions of SHA-0 [Электронный ресурс].-19 June, 2000. http://fchabaund.free.fr/English/Publications/, Свободный

Federal Information Processing Standards Publication 180-2 (2002a). Secure Hash Standard.

Federal Information Processing Standards Publication 198 (2002b). The Keyed-Hash Message Authentication Code (HMAC)

Kabulov A, Saymanov I, Yarashov I, Muxammadiev F (2021a) Algorithmic method of security of the Internet of Things based on steganographic coding. In: 2021 IEEE international IOT, electronics and mechatronics conference (IEMTRONICS). Toronto, ON, Canada, pp 1–5. https://doi.org/10.1109/IEMTRONICS52119.2021.9422588

Kabulov A, Saymanov I, Berdimurodov M (2021b) Minimum logical representation of microcommands of cryptographic algorithms (AES). In: 2021 international conference on information science and communications technologies (ICISCT). Tashkent, Uzbekistan, pp 1–5. https://doi.org/10.1109/ICISCT52966.2021.9670388

Kumar C et al (2017) Enhanced diffie-hellman algorithm for reliable key exchange. In: IOP conference series: materials science and engineering, vol 263, no 4. IOP Publishing, p 042015

Liestyowati D (2020) Public key cryptography. J Phys Conf Ser 1477(5):052062. IOP Publishing

Matusiewicz K (2007) Analysis of modern dedicated cryptographic hash functions. Macquarie University, p 153b

Muyinda N (2009) Elliptic curve cryptography. African Institute for Mathematical Sciences (AIMS)

NESSIE Consortium (2002) NESSIE Security report. Deliverable report D20–NESSIE, 2002. [Electronic resource]–Access mode: http://www.cryptonessie.org/

Saymanov I (2024) Logical automatic implementation of steganographic coding algorithms. J Math Mech Comput Sci 121(1):122–131. https://doi.org/10.26577/JMMCS2024121112

Sobti R, Geetha G (2012) Cryptographic hash functions: a review. Int J Comput Sci Issues (IJCSI) 9(2):461

Subramanya SR, Yi BK (2006) Digital signatures, vol 25, no 2. IEEE Potentials, pp 5–8

Zhang Y et al (2021) Pipezk: Accelerating zero-knowledge proof with a pipelined architecture. In: 2021 ACM/IEEE 48th annual international symposium on computer architecture (ISCA)

Chapter 2
Introduction to Blockchain Technology

Vijaya Kumbhar, Pallavi Yarde, Vinaya Keskar, Manisha Maddel, Sunil A. Kumbhar, and Ravindra S. Kamble

Abstract The book's philosophy is around using artificial intelligence-, IoT-, and blockchain-based technologies to improve smart city applications. This chapter covers blockchain technology, ensuring that readers grasp its ideas, components, and significance to smart cities. It lays the foundation for subsequent chapters that delve deeper into the connections between these technologies and how they can be used in practice. Blockchain systems are decentralized (distributed) ledger systems that guarantee that transactions across nodes in a network are safe, transparent, and final without the need for intermediaries. Its basic components include supply chain administration, financial services, healthcare, consensus algorithms, and smart contracts. All of these encourage accountability and authority through a wide range of fields. Blockchain innovation decreases dependence on an administrative authority and creates an adequate framework for growth security, eliminating illegal activities, and developing accuracy in digital transactions. As blockchain grows further, it has the potential to completely transform many sectors by providing cutting-edge answers to persistent problems with trust and data management. Finally, regulatory and ethical considerations are presented, emphasizing the importance of legal frameworks and ethical guidelines in deploying blockchain technology. By establishing a solid foundation in blockchain technology, this chapter prepares readers for

V. Kumbhar
School of Computer Studies, Sri Balaji University, Pune, India

P. Yarde (✉)
School of Computer Studies, Sri Balaji University, Pune, India
e-mail: pallavi.yarde@bitmpune.edu.in

V. Keskar
Department of Computer Science, ATSS College of Business Studies and Computer Application, Pune, India

M. Maddel
Indira School of Business Studies, Pune, India
e-mail: manishamaddel.official@gmail.com

S. A. Kumbhar · R. S. Kamble
Department of Computer Science & Engineering, D. Y. Patil Agriculture and Technical University, Kolhapur, Maharashtra, India

K. Ahmad et al. (eds.), *Fostering Machine Learning and IoT for Blockchain Technology*,
Transactions on Computer Systems and Networks,
https://doi.org/10.1007/978-981-96-4078-2_2

subsequent discussions on an overview of blockchain technology, fundamentals of concepts of blockchain, its architecture, versions, and its SWOC analysis.

Keywords Distributed ledger technology · Blocks · Transactions · Immutability · Blockchain architecture · Components · Smart contracts

2.1 Overview of Blockchain Technology

Blockchain technology is a revolutionary innovation, which misshapenned the way data is recorded, stored, and secured. Primarily it is considered the basic technology for Bitcoin, further blockchain has evolved with applications across various sectors, including finance, healthcare, supply chain, and, more recently, smart cities. This segment focuses on blockchain technology, covering its definition, basic principles, historical evolution, and relevance to modern applications, especially in the context of developing smart cities. At its core, blockchain is a decentralized and distributed digital ledger that records transactions transversely on various computers. Therefore, the post facto alteration of registered transactions cannot be possible. This preserves the safety and confidentiality of the data.

The primary foundations of blockchain technology are.

1. ***Decentralization***: Blockchain uses a network of peers to peers (P2P). Each participant maintains a copy of the carryout blockchain, ensuring no failures and mitigating the danger of central administration (Nakamoto 2008).
2. Blockchain transactions are ***transparent*** to all network members. All transactions are openly verifiable, thus builds trust among consumers (Yaga et al. 2018).
3. ***Immutability***: Once recorded on the blockchain, transactions cannot be edited or eliminated. This immutability is achieved by cryptographic hashing and the arrangement of blocks connected in a chain, which makes tampering very hard (Crosby et al. 2016).
4. Blockchain networks rely on processes of consensus for ***transaction verification*** and record. Common consensus methods include Proof of Work (PoW), Proof of Stake (PoS), and practical byzantine fault.

Satoshi Nakamoto, an individual or group of individuals, created the concept of blockchain technology in 2008 as the fundamental technology for the cryptocurrency Bitcoin. The issuance of the Bitcoin whitepaper marked the start of blockchain's rise from a fringe technological idea to a popular invention (Nakamoto 2008).

The release of Ethereum in 2015 was the second important milestone in blockchain's progress after Bitcoin. Ethereum expanded blockchain's capabilities beyond Bitcoin by allowing the development of smart contracts—self-executing contracts in which conditions are clearly encoded into code. This development paved the path to an array of distributed apps (DApps) and validated blockchain's promise in a variety of sectors (Swan 2015).

2.1.1 Fundamental Concepts of Blockchain

Blockchain technology is a decentralized and distributed ledger system that records transactions across many computers so that the record cannot be altered retroactively without altering all subsequent blocks and the network consensus.

1. **Decentralization**

   Unlike traditional centralized databases stored in a single location, a blockchain is decentralized and distributed across a network of computers (nodes). This decentralization improves security and transparency (Nakamoto 2008).

2. **Immutable Ledger**

   Once a transaction is recorded in a block and added to the blockchain, altering it is nearly impossible. Cryptographic hashes and consensus mechanisms ensure this immutability, making the blockchain highly secure (Buterin 2014).

3. **Consensus Mechanisms**

   Blockchain relies on consensus algorithms to agree on the validity of transactions. The most common mechanisms are Proof of Work (PoW) and Proof of Stake (PoS) (Narayanan et al. 2016).

4. **Cryptographic Security**

   Blockchain uses cryptographic techniques to secure transactions and control the creation of new blocks. Public and private keys are employed to ensure the authenticity and integrity of transactions (Drescher 2017).

5. **Transparency and Anonymity**

   Blockchain provides a transparent ledger that anyone can view, but it also offers anonymity through pseudonymous addresses, which makes it difficult to link transactions to specific individuals (Zohar 2015).

6. **Smart Contracts**

   Smart contracts are self-executing agreements with terms directly written into code. Once all requirements have been met, the agreement's provisions are instantly enacted and enforceable (Buterin 2014).

7. **Applications Beyond Cryptocurrency**

   While blockchain initially achieved importance with Bitcoin, its applications go beyond money. It may promote safe and transparent solutions for supply chain management, financial services, healthcare, and other sectors (Iansiti and Lakhani 2017).

8. **Distributed Ledger Technology (DLT)**

   Distributed Ledger Technology (DLT) is the set of technologies and protocols that allow for immutable simultaneous access, validation, and record updating over a network of different entities or areas. The main benefit of DLT is its decentralized governance, which eliminates the need for a central authority or middleman.

2.1.2 Cryptographic Hashing and Digital Signatures

Cryptographic hashing is fundamental in blockchain and Distributed Ledger Technology (DLT). It involves using a hash function to transform input data (of any size) into a fixed-size string of bytes. The resulting hash value is unique to the input data, making it a digital fingerprint of the data (Schneier 1996).

Critical Properties of Cryptographic Hash Functions (Schneier 1996):

1. **Deterministic**: The same input will always produce the same hash output.
2. **Fast Computation**: Hashing should quickly compute for any given input.
3. **Pre-image Resistance**: It should be computationally infeasible to reverse a hash back to its original input.
4. **Small Changes Affect Hash**: A slight change in the input drastically changes the output hash, a property known as the avalanche effect.
5. **Collision Resistance**: Finding two different inputs that produce the same hash output should be infeasible.
6. **Fixed Output Size**: Regardless of the input size, the hash output is always of a fixed size (e.g., SHA-256 always produces a 256-bit hash).

Example Hash Functions (Eastlake and Jones 2001):

- **SHA-256**: Commonly used in Bitcoin and other blockchain systems.
- **MD5**: An older hash function, less secure, and not recommended for cryptographic purposes.

Digital Signatures

Digital signatures are a cryptographic method for verifying the authenticity and integrity of digital messages or documents. They are a crucial component of blockchain technology, ensuring that transactions are valid and originate from legitimate sources (Rivest et al. 1978).

How Digital Signatures Work (Rivest et al. 1978):

1. **Key Pair Generation**: A user generates a pair of keys: a private key (kept secret) and a public key (shared with others).
2. **Signing**: The sender creates a hash of the message or transaction and then encrypts it with their private key to create a digital signature.
3. **Verification**: The receiver decrypts the digital signature using the sender's public key, obtaining the original hash. The receiver also hashes the received message and compares the two hashes. Once both signatures match, the message's integrity and authenticity are verified.

Key Properties of Digital Signatures (Diffie and Hellman 1976):

1. **Authentication** confirms the recipient's identification.
2. **Truthfulness** ensures the message wasn't altered during transit.

3. **Non-repudiation** means that the sender is prohibited from sending the communication.

Practical Application in Blockchain:

- **Transaction Validation**: Users sign deals using their private key. The additional nodes in the network verify the signature employing the public key of the sender before adding it to the database (Nakamoto 2008).
- **Block Integrity**: Each block in a blockchain includes a hash of the prior block, preserving chain integrity. Any change to a block's hash invalidates subsequent blocks (Nakamoto 2008).

Blockchain technology, which uses hash functions and digital signatures, assures secure, translucent, and impermeable transactions, and serves as the basis of autonomous systems.

2.2 Blockchain Architecture and Components

Blockchain architecture contains various components to create a secure, decentralized, and transparent system, as shown in Fig. 2.1.

The key elements and architecture of blockchain technology are

1. **Block Structure**

Each block in a blockchain contains multiple components (Nakamoto 2008):

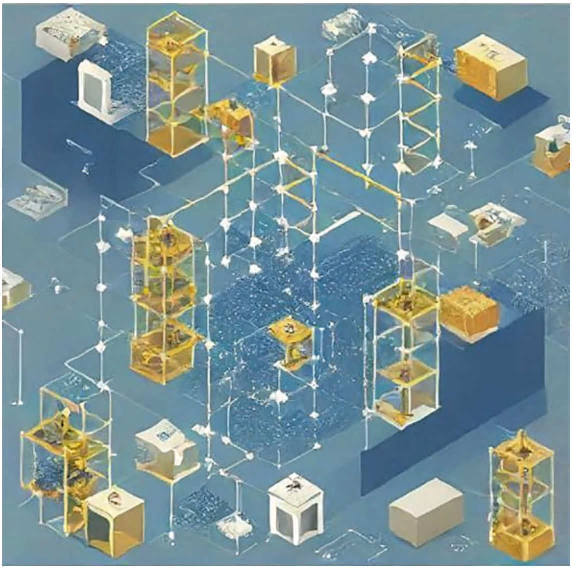


Fig. 2.1 Simulated blockchain

- **Header**: Contains metadata about the block, including
  - **Previous Block Hash**: A reference to the previous block's hash, ensuring the blockchain's integrity.
  - **Timestamp**: The time when the block was created.
  - **Merkle Root**: A hash of the root of the Merkle tree containing all transactions in the block.
  - **Nonce**: A value used in the proof-of-work consensus mechanism.
  - **Difficulty Target**: It specifies the difficulty level of the mining process.

- **Body**: It contains the actual transactions, each with
  - **Transaction Data**: Transaction Data details such as sender, receiver, and amount.
  - **Digital Signatures**: Verifications of the transaction's authenticity.
  - **Hash**: Unique identifier of the transaction.

2. **Nodes**

Nodes are the individual computers that participate in the blockchain network. They maintain copies of the blockchain and validate transactions and blocks (Antonopoulos 2017).

- **Full Nodes**: Store the entire blockchain and validate all transactions and blocks. They enforce the consensus rules.
- **Light Nodes (SPV Nodes)**: Store only the block headers and validate transactions using simplified payment verification (SPV).

3. **Consensus Mechanisms**

Consensus mechanisms ensure that all nodes in the network agree on the state of the blockchain. Common mechanisms include (Narayanan et al. 2016).

- **Proof of Work (PoW)**: Miners solve cryptographic puzzles to add new blocks.
- **Proof of Stake (PoS)**: Validators are chosen based on the number of tokens they hold and are willing to "stake".
- **Delegated Proof of Stake (DPoS)**: Token holders vote to elect delegates who validate transactions.
- **Practical Byzantine Fault Tolerance (PBFT)**: Nodes reach consensus through a process of message exchange.

4. **Cryptographic Techniques**

Blockchain relies heavily on cryptographic techniques to ensure security and integrity (Schneier 1996).

- **Hashing**: Converts input data into a fixed-size string of bytes. Common algorithms include SHA-256.

- **Digital Signatures**: Used to verify the authenticity of transactions. They involve public and private keys.

5. **Smart Contracts**

Smart contracts are self-executing contracts with the terms directly written into code. They automatically enforce and execute the terms of an agreement once the conditions are met (Buterin 2014).

6. **Peer-to-Peer Network**

The blockchain operates on a peer-to-peer (P2P) network where all nodes communicate directly with each other without a central authority. This ensures decentralization and robustness (Wattenhofer 2016).

7. **Ledger**

The ledger is a record of all transactions that have occurred in the blockchain network (Kosba et al. 2016). It is.

- **Distributed**: A copy is stored on each node in the network.
- **Immutable**: Once a transaction is recorded, it cannot be altered or deleted.

8. **Merkle Trees**

Merkle trees are data structures used to efficiently and securely verify the integrity of transactions. They allow nodes to verify specific transactions without downloading the entire blockchain (Merkle 1987).

9. **Wallets**

Wallets are tools used by users to interact with the blockchain. They store public and private keys and allow users to send, receive, and monitor their cryptocurrency holdings (Antonopoulos 2017).

- **Hot Wallets**: Connected to the Internet and more convenient but less secure.
- **Cold Wallets**: Offline and more secure but less convenient.

10. **Mining**

Mining is the process of validating transactions and adding them to the blockchain. Miners use computational power to solve complex mathematical problems, and in return, they are rewarded with cryptocurrency (Narayanan et al. 2016).

2.2.1 Blocks, Transactions, and Chains

Blockchain technology is made up of three fundamental elements: blocks of data, operations, and chains. Each of these components is essential for providing the distributed ledger network's security, transparency, and immutability.

1. **Blocks**

A block is a data structure of transactions.
Each block has two main parts: the header and the body (Nakamoto 2008).

- **Header**: The block header contains metadata about the block, including
  - **Previous Block Hash**: A hash of the previous block's header, linking the blocks together in a chain.
  - **Timestamp**: The time when the block was created.
  - **Merkle Root**: A hash of the root of the Merkle tree, which encapsulates all the transactions in the block.
  - **Nonce**: A value used in the proof-of-work consensus mechanism.
  - **Difficulty Target**: The difficulty level of the mining process.

- **Body**: The body of the block contains the actual transactions. Each transaction includes
  - **Transaction Data**: Transaction details, such as sender, receiver, and amount.
  - **Digital Signatures**: Verifications of the transaction's authenticity.
  - Transactions
  - **Hash**: A unique identifier for the transaction.

2. **Transctions**

Transactions are the fundamental operations that transfer value or information between participants in the blockchain network (Antonopoulos 2017). A transaction typically involves the following components:

- **Inputs**: References to previous transactions from which the funds or data are transferred.
- **Outputs**: The recipients of the funds or data, including the amounts being transferred.
- **Digital Signatures**: Proof that the sender authorized the transaction using their private key.
- **Transaction ID (TXID)**: A unique identifier generated by hashing the transaction details.

3. **Chains**

The chain in blockchain refers to the sequential linking of blocks. Each block contains a hash of the previous block's header, forming a continuous, immutable chain from the first (the genesis block) to the current block. This chaining mechanism ensures transaction integrity and chronological order (Wattenhofer 2016).

- **Genesis Block**: The first block in the blockchain serves as the foundation for all subsequent blocks.
- **Block Hashing**: Each block's header is hashed, and the resulting hash is used in the next block's header, creating a secure link.

- **Immutability**: Any change to a block would require altering all subsequent blocks, which is computationally infeasible in a large blockchain network.

  Practical Workflow in Blockchain (Narayanan et al. 2016):

1. **Creating a Transaction**:
   - A user initiates a transaction, signing it with their private key.
   - The transaction is broadcast to the network.
2. **Verifying Transactions**:
   - Nodes in the network validate the transaction, checking for authenticity and ensuring the sender has sufficient funds.
3. **Forming a Block**:
   - Validated transactions are grouped into a block by a miner or validator.
   - The block header is created, including the hash of the previous block, the Merkle root, and other metadata.
4. **Consensus Mechanism**:
   - Depending on the consensus mechanism (e.g., proof of work, proof of stake), nodes agree on the new block's validity.
   - In proof of work, miners compete to solve a cryptographic puzzle. The first to solve it adds the block to the blockchain.
5. **Adding to the Chain**:
   - Once consensus is reached, the new block is added to the blockchain.
   - The block's hash is included in the header of the next block, linking them together.
6. **Updating the Ledger**:
   - All nodes update their copies of the blockchain to reflect the new block.
   - The transaction is now confirmed and permanently recorded in the blockchain.

2.2.2 Nodes and Network Types

Nodes

- Nodes are tiny computers or servers that make up a blockchain network. They play a vital part in guaranteeing the blockchain's integrity and security by validating and relaying transactions (Nakamoto 2008).
- Nodes can have a variety of operations, including communication.

Types:

- **Full Nodes**: Store the entire blockchain and validate all transactions and blocks.

- **Light Nodes**: Store only a subset of the blockchain and rely on full nodes for transaction and block validation.
- **Mining Nodes**: In proof-of-work (PoW) systems, these nodes solve cryptographic challenges in order to produce new blocks while receiving rewards.
- **Validator Nodes**: In proof-of-stake (PoS) systems, nodes examine transactions and create new blocks based on their network stake.

2.2.3 Smart Contracts

Smart contracts are self-executing contracts in which the terms of the agreement are written directly into code. When certain requirements have been met, the contract's rules are automatically implemented and executed. These contracts function on blockchain structures, which render them tamper-proof and transparent.

Significance of Smart Contracts:

1. Smart contracts automate contract terms, eradicate human error, and improve efficiency (Buterin 2013).
2. Cost Reduction: Smart contracts remove the need for middlemen like attorneys or brokers, resulting in lower transaction costs and faster treatments (Szabo 1997).
3. Trust and Security: Blockchain's decentralized and permanent nature makes smart contracts secure and trustworthy, reducing the danger of fraud or manipulation (Christidis and Devetsikiotis 2016).
4. Transparency and Accuracy: Contract terms and execution are available to all parties, resulting in more transparency and fewer disputes (Zheng et al. 2017).
5. Smart contracts enable real-time execution of transactions and agreements, resulting in faster business operations (Kosba et al. 2016).
6. Smart contracts ensure agreement enforcement.

Use Cases of Smart Contracts:

Smart contracts represent a significant advancement in digital transactions and agreements. Their ability to automate, secure, and enforce agreements without intermediaries offers substantial benefits across various industries, enhancing efficiency, reducing costs, and increasing trust and transparency. By leveraging blockchain technology, smart contracts can revolutionize how agreements are executed and managed, providing a robust foundation for the future of decentralized applications and services.

2.3 Blockchain Revolution from 1.0 to 5.0

Blockchain 1.0: Cryptocurrency

The first version of blockchain technology, known as blockchain 1.0, primarily deals with Bitcoin. Blockchain 1.0 was introduced with the launch of Bitcoin in 2008 by a person or group known as Satoshi Nakamoto. This iteration focuses primarily on transactions between peers and distributed digital currencies (Nakamoto 2008).

Key Features:

- **Decentralization:** Rather than an individual authority, a network of nodes monitors transactions.
- **Consensus Mechanism:** PoW, or proof of work, is a process used for achieving acceptance on transaction records.
- **Immutability:** Transactions cannot be removed or updated while they are recorded.

Blockchain 2.0: Smart Contracts

Smart contracts, essentially self-executing contracts with the terms directly put into code, are a concept developed through Blockchain 2.0. Ethereum, proposed by Vitalik Buterin in late 2013 and launched in 2015, represents the second generation of blockchain technology (Buterin 2013).

Key Features:

- **Smart Contracts**: Enable automated, trustless transactions and agreements without intermediaries.
- **Decentralized Applications (DApps)**: Applications built on top of blockchain that utilize smart contracts.
- **Ethereum Virtual Machine (EVM)**: A runtime environment for executing smart contracts.

Blockchain 3.0: Scalability and Interoperability

Blockchain 3.0 focuses on addressing scalability issues and improving interoperability between different blockchain networks. This generation aims to support a wide range of applications beyond cryptocurrencies and smart contracts (Narayanan et al. 2016).

Key Features:

- **Scalability Solutions**: Implementations such as sharding, off-chain transactions, and layer 2 solutions.
- **Interoperability**: Protocols that allow different blockchains to communicate and interact with each other.
- **Governance Models**: Enhanced mechanisms for network governance and decision-making.

Blockchain 4.0: **Enterprise Solutions**

Blockchain 4.0 represents the application of blockchain technology in enterprise settings, focusing on improving business processes, supply chain management, and regulatory compliance. This iteration integrates blockchain with other technologies such as IoT and AI. (Drescher 2017).

Key Features:

- **Enterprise Use Cases**: Applications in supply chain management, financial services, and healthcare.
- **Private and Consortium Blockchains**: Blockchains that are restricted to certain participants for enhanced security and control.
- **Integration with Emerging Technologies**: Combining blockchain with IoT, AI, and big data for optimized solutions.

Blockchain 5.0: Decentralized Autonomous Organizations (DAOs) and Beyond

Blockchain 5.0 envisions a future where blockchain technology supports decentralized autonomous organizations (DAOs) and advanced applications that transform various sectors, including governance, digital identity, and decentralized finance (DeFi) (Zohar 2015).

Key Features:

- **DAOs**: Organizations run by smart contracts with no central authority, allowing for decentralized governance and decision-making.
- **Decentralized Finance (DeFi)**: Financial services built on blockchain, offering services like lending, borrowing, and trading without intermediaries.
- **Advanced Use Cases**: Integration with quantum computing, zero-knowledge proofs, and other cutting-edge technologies.

Comparative Analysis of Blockchain Revolution: From 1.0 to 5.0

The blockchain revolution has evolved through several stages, each bringing advancements and new capabilities to the technology. Here's a comparative analysis of Blockchain 1.0 through 5.0, covering key parameters, history, and technological developments.

Aspect	Blockchain 1.0	Blockchain 2.0	Blockchain 3.0	Blockchain 4.0	Blockchain 5.0
Era	2009–early 2010s	Early 2010s–mid-2010s	Mid-2010s–late 2010s	Late 2010s–early 2020s	Early 2020s–present
Key innovations	Bitcoin, proof of work (PoW)	Ethereum, smart contracts	Scalability solutions, interoperability	Enterprise blockchain, advanced consensus	Integration with AI, IoT, quantum computing
Core protocols	Bitcoin protocol (PoW)	Ethereum protocol (PoW, PoS)	Various protocols (e.g., Polkadot, Cosmos)	Hyperledger ,Corda, advanced protocols	Protocols integrating AI, IoT, quantum-secured systems
Consensus mechanisms	Proof of Work (PoW)	Proof of Work (PoW), Proof of Stake (PoS)	Proof of Stake (PoS), Delegated PoS (DPoS), PBFT	BFT variants, Byzantine Fault Tolerance (BFT)	Advanced consensus mechanisms integrating AI and quantum computing
Scalability	Limited scalability	Improved scalability with smart contracts	Layer 2 solutions (e.g., Plasma, Rollups)	Sidechains, sharding, and hybrid solutions	Enhanced scalability through AI- Driven optimization and quantum-resistant algorithms
Security	Basic cryptographic security	Enhanced with smart contract security	Improved security with advanced protocols	Enterprise-grade security, privacy features	Advanced cryptographic methods, quantum-resistant protocols
Primary use cases	Digital currency (Bitcoin)	Decentralized applications (DApps), smart contracts	Scalability and interoperability solutions	Enterprise solutions, digital identity	AI-driven applications, IoT integrations, advanced governance
Enterprise adoption	Limited to cryptocurrency	Growing with DApps and smart contracts	Significant, with enterprise blockchains like Hyperledger and Corda	Broad adoption in sectors such as supply chain, finance	Emerging use in AI-enhanced business processes and IoT

(continued)

(continued)

Aspect	Blockchain 1.0	Blockchain 2.0	Blockchain 3.0	Blockchain 4.0	Blockchain 5.0
Governance models	Minimal	Decentralized governance (DAOs)	Enhanced governance models, interoperability frameworks	Consortium-based governance models	Advanced governance with AI and automated systems
Impact	Established decentralized currency	Introduced smart contracts and DApps	Enhanced scalability and cross-chain interactions	Improved enterprise applications and governance	Revolutionized integration with emerging technologies
Future directions	Continued evolution of cryptocurrency	Expansion of DApps and DeFi applications	Further scalability improvements and cross-chain interoperability	Advanced enterprise solutions and governance models	Integration with AI, quantum computing, and IoT for comprehensive solutions
Key development	Introduction of Bitcoin by Satoshi Nakamoto in 2009, utilizing proof of work (PoW)	Launch of Ethereum in 2015 by Vitalik Buterin, introducing smart contracts and a Turing-complete virtual machine	Introduction of various blockchain platforms like Polkadot and Cosmos focusing on interoperability and scalability	Adoption of blockchain technology by enterprises through platforms like Hyperledger and Corda, focusing on private and permissioned blockchains	Exploration of blockchain integration with AI, IoT, and quantum computing for enhanced capabilities and security
Focus	Decentralized digital currency and payment systems	Decentralized applications (DApps) and programmable blockchain with smart contracts	Addressing scalability issues and enabling cross-chain interactions	Enterprise solutions, digital identity, and advanced governance models	Comprehensive solutions incorporating advanced technologies for broader applications

2.4 Introduction to Web 3.0

Web 3.0, often referred to as the “Semantic Web” or “Decentralized Web,” represents the next generation of the Internet, designed to enhance user experience through greater decentralization, smarter data handling, and increased user control. Unlike Web 1.0, which was static and read-only, and Web 2.0, which introduced dynamic content and social interaction, Web 3.0 aims to create a more autonomous, open, and interconnected web environment (Tapscott and Tapscott 2016; Kosba et al. 2016; Zhang and Zhang 2018; Atzori 2015).

Key Characteristics:

1. **Decentralization:**
   - **Blockchain Technology**: Web 3.0 heavily leverages blockchain technology to decentralize data storage and processing. This reduces reliance on centralized servers and intermediaries, enhancing security and trust.
   - **Decentralized Applications (DApps)**: Blockchain-based programs operate without an underlying authority, ensuring transparency and user control.
2. **Semantic Web:**
   - Web 3.0 relies on semantic technologies to enhance data interoperability and comprehension of context. This leads to enhanced data integration and more meaningful interactions.
   - Linked Data is the method of linking data from many sources to improve the web experience and provide analytics.
3. **Artificial Intelligence (AI) and Machine Learning:**
   - **Smart Algorithms**: AI and machine learning are used to process and analyze vast amounts of data, providing more personalized and relevant user experiences.
   - **Natural Language Processing (NLP)**: Enhancements in NLP enable more intuitive interactions between users and machines.
4. **User Empowerment:**
   - **Data Ownership**: Users have greater control over their personal data and can manage permissions and sharing preferences through decentralized platforms.
   - **Tokenization**: Digital tokens and cryptocurrencies can be used to incentivize and reward users, facilitating new economic models on the web.
5. **Interoperability and Integration:**
   - **Cross-Platform Integration**: Web 3.0 aims to seamlessly integrate various services and platforms, allowing for more fluid interactions and data exchanges.
   - **Smart Contracts**: Automated contracts that execute predefined conditions without human intervention, enabling trustless transactions.

Applications and Use Cases:

- **Decentralized Finance (DeFi)**: Decentralized networks provide monetary services such as borrowing, lending, and trading without the need for traditional intermediaries.
- **Digital Identity**: Systems that give users control over their digital identities, allowing for secure and private verification.
- **Supply Chain Management**: Enhanced transparency and traceability in supply chains through blockchain technology.
- **Content Creation and Distribution**: Platforms that allow creators to monetize content directly, bypassing traditional intermediaries.

2.4.1 Web 1.0

The World Wide Web's earliest opportunity edition, Web 1.0, debuted in the early 1990s and lasted until the early 2000s. Its primary characteristics include static web pages and little interaction from users. Web 1.0, sometimes known as the "read-only" web, was mostly concerned with giving knowledge in an easy-to-understand linear manner (Zhang and Zhang 2018; R3 2020; Kumar and Sharma 2019).

Key Characteristics:

1. **Static Webpages:**
    - **HTML-Only Pages:** In Web 1.0, static HTML pages were in all websites. Every single page was a distinct document that didn't alter or adapt despite user input.
    - **Limited Interactivity:** Users could only read content and navigate between pages through hyperlinks. There were no dynamic features or user-generated content.
2. **Basic Web Design:**
    - **Basic Layouts**: Text-based layouts with little visuals were the norm for websites. Simple text, picture, and hyperlink design components were used.
    - **Slow Load Times**: Websites typically loaded slowly as a result of inadequate speed and the usage of subpar graphics.
3. **Information Delivery:**
    - **Content Consumption**: Rather than focusing interaction with users, Web 1.0 focuses on content delivery. Websites functioned as online instructive guides or pamphlets.
    - **Static Content**: Web 1.0 sites had static content, which implied that the owner or administrator of the website had to manually change the HTML code of the site in order to make adjustments.

4. **No User Interaction**:
    - **Lack of Personalization**: There were no mechanisms for personalizing content or user interactions. Each visitor saw the same content in the same format.
    - **No User-Generated Content**: Users could not contribute content, post comments, or participate in discussions.
5. **Web Directories:**
    - **Search Engines**: Search engines like AltaVista, Lycos, and Yahoo! were used to navigate the growing web. Web directories organize links into categories to help users find relevant sites.
    - **Manual Listings**: Websites were often listed manually in directories and search engines, requiring site owners to submit their URLs for inclusion.

Historical Context:

- **Early Development**: The World Wide Web was invented by Tim Berners-Lee in 1989 and became publicly available in 1991. The early web was largely an academic and research tool.
- **Browser Technology**: Early web browsers like Mosaic and Netscape Navigator played a significant role in popularizing the web, providing a graphical interface for accessing web pages.

Examples of Web 1.0 Sites:

- **Personal Homepages**: Simple, individual websites created by users to share personal information, hobbies, and interests.
- **Company Websites**: Early corporate websites that provided basic company information, product descriptions, and contact details.

Impact and Legacy:

- **Foundation for Growth**: Web 1.0 laid the groundwork for the development of more interactive and dynamic web technologies that emerged with Web 2.0.
- **Limited Scope**: While Web 1.0 was revolutionary in its time, it was limited by its static nature and lack of user interactivity.

2.4.2 Web 2.0

Tim O'Reilly and Dale Dougherty first used the phrase "Web 2.0" in 2004 to refer to the second iteration of the World Wide Web. Web 2.0 stands out from Web 1.0 by its dynamic content, greater user interaction, and more collaborative online environment. To provide a more participating and interesting online experience, it focuses on user-generated content, social networking, and enhanced usability (Zhang and Zhang 2018).

Key Characteristics:

1. **Dynamic Content**:
   - Online 2.0 introduced dynamic web applications with configurable user interfaces. AJAX (Asynchronous JavaScript and XML) technology enables Internet pages to refresh their content without requiring a page reload.
   - Rich User Experiences: Enhanced online interfaces with functionality for dragging and dropping, live updates, including interactive forms.
2. **User-Generated Content**:
   - Blogs and Wikis, such as WordPress and Wikipedia, foster collaborative generation of knowledge through simple content creation, modifying and sharing possibilities.
   - Social media platforms like Instagram, Facebook, and Twitter enable users to engage, share happiness, and connect with others.
3. **Social Networking**:
   - Web 2.0 provides online communities and networks of friends, allowing users to join based on similar interests and activities.
   - Social sharing enables users to interact and engage with content sharing, debates, and social networking activities.
4. **Collaboration and Crowdsourcing**:
   - Collaborative Tools: Platforms like Wikipedia and Google Documents give real-time collaboration for projects and documents.
   - Crowdsourcing is a method utilized by businesses and organizations for collecting contributions, ideas, and regards from an array of people.
5. **Rich Media and Personalization**:
   - Web 2.0 offered multimedia content inclusion and display capabilities, including motion pictures, audio, and interactive graphics.
   - Web 2.0 technology can provide customized recommendations for material based on user habits and passions.
6. **Tagging and Folksonomies**: **Historical Context:**
   - **Technological Advancements**: The introduction of technologies like AJAX, better websites, and JavaScript frameworks (like jQuery) made transitioning from static to dynamic online applications easier.

- **User-Centric Focus**: Web 2.0 emphasizes the shift from content consumption to content creation and user interaction, reflecting broader trends in technology and society.

Examples of Web 2.0 Sites:

- **Social Media Platforms:** Facebook, Twitter, LinkedIn, Instagram.
- **Content Creation Platforms:** YouTube, Medium, WordPress.
- **Collaborative Tools:** Google Drive, Slack, GitHub.

2.4.3 Web 3.0

Building on the innovations of Web 2.0, Web 3.0, usually referred to as the "Semantic Web" or the "Centralized Web," is the next version of the World Wide Web. With more data linked and devices able to comprehend and process information with more context and relevance, Web 3.0 seeks to build a more intelligent, distributed. and user-centric Internet (Zhang and Zhang 2018; Wood 2014).

Key Characteristics:

1. **Decentralization:**
   - **Blockchain Technology:** To distribute data processing and storage, Web 3.0 makes use of blockchain technology. This improves security and transparency by lowering dependency on central servers and middlemen.
   - **Centralized Applications (DApps):** Programs built on anarchic blockchain technology give more accessibility and user autonomy.
2. **Semantic Web:**
   - **Enhanced Data Interoperability**: Web 3.0 places a high value on the application of semantic technologies to improve data integration and provide more useful and timely material to social media. This involves working with ontologies, metadata, and linked data mechanically.
   - **Machine Readability**: Content has been organized to make it easier for computers to comprehend and analyze, which improves search results and automates tasks.
3. **Artificial Intelligence (AI) and Machine Learning:**
   - **Smart Algorithms**: Artificial intelligence (AI) and other techniques are used to collect and analyze vast volumes of data to produce more relevant and custom user experiences.
   - **Enhanced natural language processing** (NLP) capabilities allow for a better comprehension of user inquiries and context. They also provide improved user interactions between apps on the Internet and customers.
4. **User Empowerment**:
   - **Data Ownership**: Users have greater influence over sharing and using their data. Distributed platforms provide permission and privacy management capabilities.

- **Tokenization** is using digital tokens and money to encourage user involvement and create new economic models.

5. **Interoperability and Integration**:

Web 3.0 aims to improve user satisfaction and information transfer by enabling seamless integration between browsers and services. **Smart contracts** enable autonomous and suspicious activities by incorporating criteria directly into code.

6. **Advanced Use Cases**:

Decentralized Finance (DeFi) bypasses middlemen and offers randomization lending, borrowing, and trading. Digital identity solutions provide buyers with private and secure access to retain and govern their digital identities. Blockchain and IoT solutions improve supply chain traceability and transparency.

Examples of Web 3.0 Technologies:

- **Blockchain Platforms:** Ethereum, Polkadot, and Cardano
- **Semantic Web Tools:** RDF (Resource Description Framework), OWL (Web Ontology Language), and SPARQL for querying linked data.
- **DeFi Platforms:** Uniswap, Compound, and MakerDAO, which offer decentralized financial services.

Impact and Future Directions:

- **Enhanced Web Intelligence:** Web 3.0 aspires to make the World Wide Web more intelligent and responsive, with data that is more merged and actionable.
- **User-Centric Innovations:** The focus on decentralization and user control represents a shift toward more equitable and user-friendly web experiences.

Comparative Analysis of Web 1.0, Web 2.0, and Web 3.0

Aspect	Web 1.0	Web 2.0	Web 3.0
Era	Early 1990s to early 2000s	Early 2000s to present	Emerging from the early 2020s
Description	The early web was characterized by static content and limited user interaction	The evolution of the web to a more dynamic, interactive, and social platform	The next generation of the web focuses on decentralization, semantic understanding, and enhanced user control
Content type	Static pages	Dynamic, user-generated content	Dynamic, decentralized, and semantically linked content
User interaction	Read-only	Read and write, social interaction	Read, write, and own; direct interaction with blockchain-based systems
Technologies	HTML, early browsers (e.g., Mosaic, Netscape)	AJAX, JavaScript frameworks, social media	Blockchain, semantic web technologies, AI

(continued)

(continued)

Aspect	Web 1.0	Web 2.0	Web 3.0
Data ownership	Minimal control over data	Limited control: data often collected by third parties	High control: users manage their data via decentralized platforms
Economic models	Traditional models, ad-supported	Ad-based revenue, freemium models	Token-based economies, decentralized finance (DeFi)
Core protocols	HTTP, HTML	HTTP, HTML, AJAX	HTTP, HTTPS, blockchain protocols (e.g., Ethereum, Polkadot)
Data storage	Centralized servers	Centralized servers, cloud storage	Decentralized storage, distributed ledgers
Interactivity	Static, hyperlinks only	Dynamic content, AJAX, REST APIs	Decentralized apps (DApps), smart contracts
Content creation	Manual updates by site owners	User-generated content (blogs, wikis)	User-generated and decentralized content; controlled via blockchain
Data storage	Centralized servers	Centralized servers, cloud storage	Decentralized storage, distributed ledgers
Security	Basic security measures, less emphasis	Improved security protocols, HTTPS	Advanced cryptographic security, decentralized consensus mechanisms
Privacy	Minimal user control over data	With limited control, data is often collected by third parties	Enhanced privacy with user control over personal data, pseudonymous transactions
Data ownership	Minimal to no control	Limited control	With high control, users manage their data via decentralized platforms
Economic models	Traditional models, ad-supported	Ad-based revenue, freemium models	Token-based economies, decentralized finance (DeFi)
Applications	Personal homepages, basic company sites	Social networking, blogs, wikis, multimedia sharing	Decentralized applications (DApps), smart contracts, digital identity management
Use cases	Basic information sharing	Collaborative platforms, social engagement	Decentralized finance (DeFi), secure digital identities, supply chain transparency
Impact	Established the web as an information medium	Shifted the web toward user interaction,social connectivity, and dynamic content	Aims to transform the web into a more decentralized, intelligent, and user- empowered environment
Future directions	Legacy of foundational web technologies	Continued growth in interactivity and social applications	Development of scalable decentralized technologies, greater AI integration, and semantic web advancements

2.5 SWOC Analysis of Blockchain Technology

Strengths: Blockchain technology offers several notable strengths:

- Blockchain centralized management removes the chance of one point of failure, resulting in greater security and trust.
- Transparency: Blockchain's unchangeable ledger simplifies tracking and verification, resulting in improved accountability.
- Cryptographic techniques secure blockchain data from tampering and fraud, maintaining its honesty and security.
- Blockchain removes intermediaries, lowering costs and accelerating transactions.
- Blockchain drives innovation by allowing smart contracts and decentralized apps (DApps) that result in novel enterprises and solutions.

Weaknesses: Despite its advantages, blockchain technology has several weaknesses:

- **Scalability**: Many blockchain systems struggle with transaction speed and network capacity limitations, which can hinder their ability to handle large volumes of transactions.
- **Energy Consumption**: Proof-of-Work (PoW) consensus mechanisms, used in many blockchain systems, require substantial computational power, leading to high energy consumption.
- **Complexity**: The technology can be complex to understand and implement, necessitating specialized knowledge and skills.
- **Regulatory Uncertainty**: The varying regulations and legal frameworks across different regions can create uncertainty and pose challenges for widespread adoption.
- **Data Privacy**: While blockchain is transparent, it can potentially expose sensitive information if not properly managed, raising privacy concerns.

Opportunities: Blockchain technology presents several opportunities for growth and development:

- **Enterprise Adoption**: Increasing interest from businesses and governments in leveraging blockchain for secure and transparent systems as well as for smart contracts.
- **Decentralized Finance (DeFi)**: Expanding the use of blockchain in financial services, including lending, trading, and payments, which offers new financial products and services.
- **Supply Chain Management**: Enhanced traceability and efficiency in monitoring goods and verifying transactions across supply chains, leading to better management and reduction of fraud.

- **Integration with Emerging Technologies**: Potential to combine blockchain with artificial intelligence (AI), the Internet of Things (IoT), and other technologies to create innovative solutions and applications.
- **Digital Identity**: Development of secure, decentralized digital identities for individuals and organizations, improving identity management and security.

Challenges:

- **Scalability Issues**: Ongoing efforts are required to overcome transaction volume and network growth constraints, assuring that blockchain systems are capable of meeting rising demand.
- **Interoperability**: Achieving seamless integration between different blockchain networks and platforms remains a challenge, affecting the ease of interaction and data exchange.
- **Regulatory Hurdles**: The evolving legal and regulatory landscapes may impact blockchain development and adoption, requiring adaptation to new rules and standards.
- **User Adoption**: There is resistance from traditional systems and reluctance to change established processes, which can slow down the adoption of blockchain technology.
- **Security Risks**: Potential vulnerabilities in smart contracts and blockchain implementations could be exploited, necessitating ongoing security improvements and monitoring.

2.6 Conclusion

The use of blockchain technology in smart towns marks a big step concerning integrating technological developments with concrete societal advantages. Blockchain provides a decentralized, transparent, and secure platform for addressing major difficulties in modern urban contexts, including data security, resource management, governance, and citizen involvement. Blockchain improves data confidence and privacy, making sure susceptible information is truthful and tamper-proof, fostering faith between citizens and authorities. The opportunity for enhancing resource management through real-time visibility and automation encourages sustainability and efficiency in municipal operations. Blockchain transparency promotes improved governance by creating transparent, unchangeable records of administrative activity, excluding corruption and enhancing accountability. Additionally, blockchain technology enables better traffic and transportation management, resulting in more efficient and less crowded urban transportation networks. It also improves citizen involvement by allowing for direct and transparent communication with local authorities and facilitating municipal services through automation.

References

Antonopoulos AM (2017) Mastering Bitcoin: unlocking digital cryptocurrencies. O'Reilly Media

Atzori M (2015) Blockchain-based smart contracts: opportunities and threats. In: 2015 IEEE international conference on social computing and networking (SocialCom), pp 122–128. https://doi.org/10.1109/SocialCom.2015.89

Buterin V (2013) Ethereum White Paper. https://ethereum.org/en/whitepaper/

Buterin V (2014) A next-generation smart contract and decentralized application platform. Ethereum Whitepaper. https://ethereum.org/en/whitepaper/

Christidis K, Devetsikiotis M (2016) Blockchains and smart contracts for the internet of things. IEEE Access 4 2292–2303. https://ieeexplore.ieee.org/document/7467408

Crosby M, Pattanayak P, Verma S, Kalyanaraman V (2016) Blockchain technology: beyond Bitcoin. Appl Innov Rev (2). https://j2-capital.com/wp-content/uploads/2017/11/AIR-2016-Blockchain.pdf

Diffie W, Hellman M (1976) New directions in cryptography. IEEE Trans Inf Theory 22(6):644–654. https://doi.org/10.1109/TIT.1976.1055638

Drescher D (2017) Blockchain basics: a non-technical introduction in 25 Steps. Apress

Eastlake D, Jones P (2001) US Secure Hash Algorithm 1 (SHA1). RFC 3174. https://tools.ietf.org/html/rfc3174

Iansiti M, Lakhani KR (2017) The Truth About Blockchain. Harv Bus Rev 95(1):118–127

Kosba A, Miller A, Shi E, Wen Z, Papamanthou C (2016). Hawk: the Blockchain model of cryptography and privacy-preserving smart contracts. In: 2016 IEEE symposium on security and Privacy (SP),pp 839–858. https://ieeexplore.ieee.org/document/7546538

Kumar P, Sharma R (2019) Identity management using Blockchain technology and machine learning. In: proceedings of the 2019 IEEE international conference on Blockchain and cryptocurrency (ICBC), pp 1–8. https://doi.org/10.1109/ICBC.2019.8794719

Merkle RC (1987) A digital signature based on a conventional encryption function. In Advances in Cryptology—CRYPTO '87. Springer, Berlin, Heidelberg, pp 369–378

Nakamoto S (2008) Bitcoin: a Peer-to-Peer electronic cash system. In: 2008. https://bitcoin.org/bitcoin.pdf

Narayanan A, Bonneau J, Felten E, Miller A, Goldfeder S (2016) Bitcoin and Cryptocurrency technologies: a comprehensive introduction. Princeton University Press

Pilkington M (2016) Blockchain Technology: Principles and Applications. In: Olleros FX, Zhegu M (eds) Research Handbook on Digital Transformations. Edward Elgar Publishing, pp 225–253

R3 (2020) Corda: a distributed ledger. https://www.r3.com/wp-content/uploads/2020/05/corda-technical-whitepaper.pdf

Rivest RL, Shamir A, Adleman L (1978) A method for obtaining digital signatures and public-key cryptosystems. Commun ACM 21(2):120–126. https://doi.org/10.1145/359340.359342

Schneier B (1996) Applied Cryptography: Protocols, Algorithms, and Source Code in C. John Wiley & Sons

Swan M (2015) Blockchain: blueprint for a new economy. O'Reilly Media. ISBN: 978-1491920497

Szabo N (1997) The Idea of Smart Contracts. http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart_contracts_idea.html

Tapscott D, Tapscott A (2016) Blockchain Revolution: How the Technology Behind Bitcoin Changes Money, Business, and the World. Penguin

Wattenhofer R (2016) The science of the Blockchain. Inverted Forest Publishing

Wood G (2014) Ethereum: a secure decentralized generalized transaction ledger (EIP-150). https://ethereum.github.io/yellowpaper/paper.pdf

Yaga D, Mell P, Roby N, Scarfone K (2018) Blockchain technology Overview. National Institute of Standards and Technology (NIST), U.S. Department of Commerce. https://nvlpubs.nist.gov/nistpubs/ir/2018/NIST.IR.8202.pdf

Zhang L, Zhang X (2018) Predictive analytics for environmental monitoring in smart cities. IEEE Trans Environ Eng 25(2):234–245. https://doi.org/10.1109/TENV.2018.2843287

Zheng Z, Xie S, Dai H, Chen X, Wang H (2017) An overview of Blockchain technology: architecture, Consensus, and Future Trends. In: 2017 IEEE international congress on big data (BigData Congress), pp 557–564. https://ieeexplore.ieee.org/document/8029379

Zohar A (2015) Bitcoin: under the hood. Commun ACM 58(9):104–113. https://doi.org/10.1145/2701411

Chapter 3
Reconnoitering Blockchain Consensus Mechanisms

Vijaya Kumbhar, Vinaya Keskar, Madhavi Satish Avhankar, Pallavi Yarde, Rasila Walhhekar, and Sunil A. Kumbhar

Abstract The integrity of Bitcoin is that blockchain facilitates decentralized transactions across sectors, but some challenges still exist. Initially linked with cryptocurrency, its core has reshaped perceptions. Blockchain ensures resistance to tampering, transactions, and the utilization of technology. Integrating blockchain with Machine Learning (ML) and the Internet of Things (IoT) encourages data security and privacy, providing a secure data platform. Consensus protocols used in blockchain ensure that the network nodes know about the transactions and the direction to extend to the ledger. This is achieved by defining rules and mechanisms to be implemented on nodes to validate and confirm the transaction. Hence, it ensures the reliability and security of the distributed ledger. Consensus protocols promise data validity and algorithms by integrating ML and IoT. It preserves the data's reliability and concealment by enabling protection and decentralized association. These algorithms create protocols for consensus and harmonization among distributed nodes. This enables reliable and efficient data exchange and also, investigation in ML and IoT networks. Practically, consensus protocols act as performers, among distributed players in

V. Kumbhar (✉)
School of Computer Studies, Sri Balaji University, Pune, India
e-mail: veejeya.kumbhar@gmail.com

V. Keskar
Department of Computer Science and Application, ATSS College of Business Studies and Computer Application, Pune, India

M. S. Avhankar
Indira College of Commerce and Science, Pune, India

P. Yarde
School of Computer Science, Sri Balaji University, Pune, India
e-mail: pallavi.yarde@bitmpune.edu.in

R. Walhhekar
Symbiosis Institute of Computer Studies and Research Centre, Pune, India

S. A. Kumbhar
Department of Computer Science & Engineering, D. Y. Patil Agriculture and Technical University, Maharashtra, India

K. Ahmad et al. (eds.), *Fostering Machine Learning and IoT for Blockchain Technology*,
Transactions on Computer Systems and Networks,
https://doi.org/10.1007/978-981-96-4078-2_3

Blockchain networks. This confirms the easy flow of transactions with perfection and integrity in the digital world. Consensus protocol transforms the ML and IoT applications with decentralized trust and harmony for data exchange and analysis. The scope of the consensus protocol is amalgamation with quantum-resistant algorithms to work upon scalability and security challenges. This may ensure the user privacy with legal and regularity considerations. Therefore, consensus mechanisms adapt to trust and accountability in decentralized systems. These protocols enable efficient data exchange and analysis, fostering innovation and scalability in ML and IoT ecosystems while maintaining integrity and privacy.

Keywords Consensus · Validation protocols · Byzantine fault tolerance · Proof-of-work · Proof-of-stake · Delegated proof-of-stake · Practical byzantine fault tolerance (PBFT) · Consensus algorithms

3.1 Introduction to Blockchain, Machine Learning, and IoT Integration

3.1.1 Overview of Blockchain Technology

Blockchain, the underlying technology behind Bitcoin, has recently garnered significant attention. Functioning as an unalterable ledger, it enables decentralized transactions. With the emergence of blockchain-based applications, various sectors like financial services, reputation systems, and IoT have been encompassed. Nevertheless, numerous challenges persist within blockchain technology, notably scalability and security issues necessitating resolution (Zheng et al. 2017).

Blockchain technology (BCT) has garnered global attention for about a decade, spurred by Nakamoto's introduction, initially focusing on digital currency or cryptocurrency. These currencies operate on decentralized systems, with blockchain as their underlying technology, hailed as one of today's most promising and disruptive technologies. Initially, many believed Bitcoin to be synonymous with blockchain, leading to misconceptions about its potential impact on business applications. However, the core technology behind Bitcoin has reshaped perceptions across various domains, including Banking, Healthcare, and Government, among others, among researchers, industrialists, and academicians (Komalavalli et al. 2020).

Blockchains are digital ledgers resilient to meddling, instigated in a distributed manner without a central repository, and often without centralized control from entities like banks, companies, or governments. They empower a community of users to record transactions in a shared ledger, certifying that transactions cannot be changed under normal network operation once published. This document provides a technical overview of blockchain technology to aid readers in understanding its functioning (Yaga et al. 2019).

In 2009, Bitcoin emerged as a peer-to-peer electronic cash system, inspiring the creation of numerous cryptocurrencies and paving the way for blockchain technology. This subset of Distributed Ledger Technology (DLT) saw the introduction of projects like Ethereum, Hyperledger Fabric, and R3 Corda by 2015. Blockchain has been developed with its diverse network architectures, consensus protocols, and intensive models during the last decades. These changes in blockchain have been adopted by startups, enterprises, and governments (Ramadoss 2022).

Blockchain is a distributed and decentralized database facilitating fast and reliable transactions without centralized monitoring. As compared to cryptocurrency, blockchain technology shows its significance across finance and non-finance sectors (Sheth and Dattani 2019).

3.1.1.1 Introduction to Machine Learning (ML) and Internet of Things (IoT)

Integrating AI technologies with blockchain network techniques improves the capabilities and functionalities of the essential system and can boost blockchain technology in several significant areas (Kumar et al. 2021).

Key areas are:

- **Implementation**: AI algorithms analyze the blockchain data in the following steps,
  - Extract insights
  - Detect patterns
  - Make predictions
  - Aids decision-making
  - Ensures risk assessment.
- **Optimization**: Helps to regulate smart contract execution, industrialize decision-making processes, and enable supporting dynamic contracts.
- **Security**: This integration boosts the security in blockchain networks by
  - identifying anomalies
  - Prevent fraudulent transactions by analyzing transaction patterns
  - user behavior in real time
- **Automation**: Helps in decision-making within Decentralized Autonomous Organizations (DAOs) on blockchain networks. This improves operational efficiency.
- **Safety & Privacy**: Some AI techniques like homomorphic encryption and multi-party computation improve privacy and confidentiality. This allows computations on encrypted data, maintaining privacy even though enabling data analysis and processing.

ML refers to applying techniques and algorithms to boost various features of blockchain technology. ML algorithms play an important role by:

- **Implementation**: It analyzes blockchain data
  - To reveal patterns, trends, and correlations,
  - Offering insights into transaction behaviors and market trends.
  - Predict future transaction volumes and detect anomalies
  - Aiding in decision-making.
- Optimization: It helps to optimize:
  - Smart contract execution by analyzing performance metrics and automating parameter tuning, enhancing efficiency and scalability.
  - Consensus mechanisms by analyzing network performance metrics and adjusting parameters for improved throughput and latency.
- **Security**: ML enhances security by detecting and preventing fraudulent activities on blockchain networks, bolstering network integrity.
- **Privacy and Confidentiality**: It applies differential privacy and federated learning to preserve user privacy. This enables data analysis & processing and employs encryption methods to protect sensitive blockchain data (Zheng et al. 2017; Kumar et al. 2021).

Integrating IoT nodes and blockchain technology networks enhances the overall efficiency of the system by:

- Enabling secure and decentralized data exchange and transactions.
- Helps in data integrity and security
- IoT devices generate huge sensed data also, securely record transactions on the blockchain. This confirms data integrity along with preventing unauthorized access or tampering.
- Blockchain's decentralized ledger facilitates direct data exchange among IoT devices. It eliminates intermediaries and promotes transparency and autonomy in data sharing.
- Blockchain ensures secure identity management for IoT devices. Every IoT device has a unique digital identity which is recorded on the blockchain. This ensures tamper-proof identity management and data security.
- Smart contracts deployed on blockchain networks automate the agreements and transactions between IoT devices. This helps in restructuring operations and reducing manual intervention.
- In the area of supply chain and logistics applications, blockchain-enabled IoT provides transparent and immutable tracking of goods and chattels all over the supply chain. It enhances visibility, traceability, and accountability (Zheng et al. 2017; Kumar et al. 2021).

3.1.1.2 Significance of Integrating Blockchain with ML and IoT

AI, ML, IoT, and blockchain are distinct technologies with their own set of characteristics and capabilities. However, combined, they can create synergies and unlock new opportunities in various domains.

Combining blockchain with ML, and IoT, collectively endorses data security, privacy, and trust in decentralized systems as shown in Fig. 3.1. Blockchain's core attributes, like immutability and cryptographic hashing, guarantee data integrity and security, making it an ideal solution for storing sensitive datasets in ML applications. Hence, it minimizes the risks of data fraud and illegal access. Also, in the IoT realm, blockchain proposes a protected environment for communicating and storing data from IoT devices. It addresses concerns around liabilities in centralized data repositories (Zheng et al. 2017).

3.2 The Heart of Blockchain *Consensus*: Synopsis and Core Principles

3.2.1 Overview of Consensus Protocols

Consensus within blockchain denotes the complicated algorithm by which a network of nodes collaboratively works and scopes to an agreed decision toward the authenticity of transactions. They followed a sequence in which they were recorded in the ledger. This process ensures the heftiness and scalability of blockchain systems (Bashar et al. 2019a). In blockchain technology, several consensus protocols, the well-known Proof-of-Work (PoW), and Proof-of-Stake (PoS) mechanisms, have

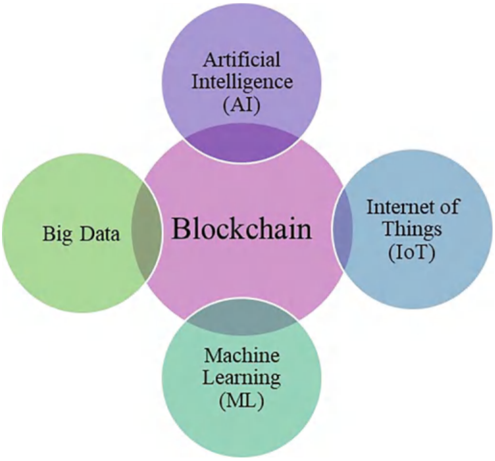


Fig. 3.1 Synergy of AI, ML, IoT, and big data with blockchain

emerged as Bedrock Principles to enable this crucial agreement. Hence, thereby serving as the spine of blockchain security (Sharma and Jain 2019). These continual advancements are mandatory to address lingering fears of security vulnerabilities and performance constraints (Sharma and Jain 2019). The concept of consensus in blockchain ecosystems covers agreement on transaction validity, embracing a broader criterion centered around application-specific predicates. In that way, enabling consensus in Byzantine processes, although concentrating on authenticating planned values instead of the nature of the recommending participant (Crain et al. 2019). Exploring deeper blockchain technology shows that the consensus algorithm makes a good impact on the operational efficiency and overall performance of blockchain applications.

Given that blockchain operates as a distributed ledger system, which is characterized by its domineering requirement for consensus mechanisms to carry the integrity of transactions. The selection of an appropriate Consensus algorithm develops as an essential of a blockchain system's efficacy (Yusoff et al. 2022). Thus, the serious consequence of prioritizing the performance and efficiency of blockchain applications underscores the importance of selecting an exact consensus algorithm (Yusoff et al. 2022).

Consensus, for blockchain, defines a communal bond between nodes in a decentralized network regarding the dynamism and order of transactions added to the distributed ledger. A formal definition can be provided as follows:

Formal Definition:

"Consensus in Blockchain systems is the process by which a distributed network of nodes achieves agreement on the state of the ledger, ensuring that all transactions are valid and consistently ordered. It involves the synchronization of independently maintained copies of the ledger across the network, such that every node converges to a shared and immutable record of transactions. Consensus protocols, such as proof of work or proof of stake, facilitate this agreement by providing mechanisms for nodes to collectively validate transactions and agree on their inclusion in the Blockchain" (Narayanan et al. 2016).

3.2.2 Historical Evolution of Consensus Mechanisms

The evolution of consensus mechanisms is an interesting journey that prop up the development of blockchain technology and decentralized networks. Consensus mechanisms ensure the contract among nodes in a network for the state of the ledger, its transactions, and the validity of new blocks being added to the chain. Here's an overview of the historical evolution of consensus mechanisms.

3.2.2.1 Proof-of-Work (PoW)

The concept of Proof-of-Work was introduced by Cynthia Dwork and Moni Naor in 1993 to discourage spam emails. However, PoW as the consensus mechanism for Bitcoin was implemented by Satoshi Nakamoto in 2008. PoW brings about network participants (miners) to make out complex mathematical puzzles to authenticate and add new blocks to the blockchain. The first successful sapper to crack the puzzle is satisfied with newly issued coins. PoW considered the longest chain with the most accumulated work is the valid chain that makes it secure but energy-intensive (Nakamoto 2008a).

3.2.2.2 Proof-of-Stake (PoS)

PoS was proposed as a substitute for PoW to address its energy consumption issues. In a PoS consensus mechanism, validators create new blocks and validate their transactions based on the number of coins they hold or stake. It was introduced in 2012 by Sunny King and Scott Nadal with a new coin Peer-Coin. PoS selects validators in the network based on their stake and incentivizes them to act honestly to defend their investment. Ethereum has been migrating to a PoS mechanism with its version Ethereum 2.0 to reduce energy consumption and improve scalability (King and Nadal 2012).

3.2.2.3 Delegated Proof-of-Stake (DPoS)

DPoS was introduced by Daniel Larimer with the launch of BitShares in 2014 and later popularized by the Semite and EOS platforms. In a DPoS system, coin containers poll for an inadequate number of delegates accountable for authenticating transactions and adding new blocks to the blockchain. DPoS proposes to attain scalability and energy efficiency by delegating consensus responsibilities to a smaller group of trusted nodes, albeit with some centralization concerns (Larimer 2014).

3.2.2.4 Proof-of-Authority (PoA)

PoA is a consensus contrivance where block validators are known and official by the network. Authenticators are typically reputable units or organizations rather than anonymous miners. PoA is used in permissioned or private blockchains where trust and identity verification are prioritized over decentralization. Ethereum's Clique PoA consensus mechanism is one instance of PoA used in private blockchain networks (Foundation 2024).

3.2.2.5 Proof-of-Space (PoSpace) and Proof-of-Capacity (PoC)

PoSpace and PoC leverage unused stowage space on a device as a resource for achieving consensus. PoSpace requires network participants to dedicate storage space, while PoC requires participants to prove they have pre-allocated storage space. Chia Network is a notable example of a cryptocurrency that uses PoSpace as its consensus mechanism (Network 2017).

3.2.2.6 Hybrid and Novel Consensus Mechanisms

There are numerous other consensus mechanisms and hybrid models being developed and experimented with, aiming to address the limitations of existing mechanisms. Some examples include Proof-of-Burn (PoB), Proof-of-Elapsed Time (PoET), Byzantine Fault Tolerance (BFT), and Directed Acyclic Graphs (DAGs) such as Tangle (used in IOTA).

Proof-of-Burn (PoB)

Early cryptocurrency developers proposed Proof-of-Burn as a consensus mechanism, but it gained more prominence in the cryptocurrency community around 2014-2015. However, it's significant to note that there isn't a specific single year associated with its development as it evolved gradually as a concept within the cryptocurrency space (Understanding Proof of Burnin Digital Ledger Systems 2023).

Proof-of-Elapsed Time (PoET)

PoET was developed by Intel and introduced as part of the Hyperledger Sawtooth blockchain platform. The development and introduction of PoET coincide with the launch of Hyperledger Sawtooth, which was announced in 2016 (Intel 2016).

Byzantine Fault Tolerance (BFT)

Byzantine Fault Tolerance is a concept that has been studied in distributed computing and computer science for several decades. The development and research surrounding BFT algorithms have been ongoing since the 1980s. However, its application to blockchain technology and cryptocurrency systems gained traction in the early to mid-2010s (Lamport et al. 1982).

Directed Acyclic Graphs (DAGs) Such as Tangle

Tangle, the consensus mechanism used in IOTA, was introduced with the launch of the IOTA project in 2015. While DAG-based structures and consensus mechanisms have been studied in academic research prior to this, the development of Tangle as a practical implementation occurred in 2015 with the founding of the IOTA Foundation (IOTAFoundation 2015).

The timeline of the evolution of consensus mechanisms is summarized in Table 3.1 as per the year of invention. These years provide a general timeline for the development and introduction of each consensus mechanism. Its value perceiving that the evolution and refinement of these mechanisms continue over time as they are implemented, tested, and adapted in various blockchain projects and systems. This historical evolution demonstrates the ongoing innovation and experimentation in the field of consensus mechanisms, obsessed by the essential to address scalability, energy efficiency, security, and decentralization in blockchain networks.

3.2.3 Importance of Consensus Protocols

Consensus protocols are fundamental for ensuring agreement and trust in distributed systems, which are systems where multiple computers or nodes work together. They play a critical role in several areas, particularly:

- **Blockchain Technology**: In blockchains, consensus protocols are essential for authenticating transactions and maintaining the honesty of the shared ledger. Without a consensus protocol, there would be no way to ensure that everyone on the network agrees on the order and validity of transactions, which would defeat the purpose of a decentralized system (Contributors 2023).
- **Distributed databases**: In distributed databases, manifold facsimiles of the records are deposited across dissimilar nodes. Consensus protocols ensure that all the copies of the data remain consistent, though more or less nodes fail or are converted inaccessible (Contributors 2023).
- **General distributed systems**: Consensus protocols are also used in various distributed systems where multiple nodes need to agree on a common state of information. This can include things like electing a leader node, coordinating tasks, or maintaining a consistent view of the system's health (Contributors 2023).

The importance of consensus protocols boils down to these key benefits:

- **Data consistency and integrity**: They ensure that everyone in the network has the same version of the truth, preventing conflicting data states (Lestienne 2019).
- **Security**: By building it difficult for malevolent performers to meddle with the data, consensus protocols contribute to the system's overall security (Lestienne 2019).

Table 3.1 Timeline of consensus mechanisms

Protocol	Year	Mechanism	Particulars	Introduced by
Byzantine Fault Tolerance (BFT)	1982	Byzantine Fault Tolerance (BFT)	Addresses consensus in the occurrence of faulty or nasty nodes	Lamport, L., Shostak, R., & Pease, M. (IOTAFoundation. 2015)
Bitcoin	2008	Proof-of-Work (PoW)	Needs miners to explain enigmas to legalize transactions. Energy-intensive	Nakamoto (King and Nadal 2012)
Peercoin	2012	Proof-of-Stake (PoS)	Validators are selected grounded on coin stake. Addresses PoW's energy consumption	King & Nadal (Larimer 2014)
BitShares	2014	Delegated Proof-of-Stake (DPoS)	Optimizes PoS by delegating consensus to trusted delegates. Centralization concerns	Larimer (Foundation 2024)
Proof-of-Burn (PoB)	2014–2015	Proof-of-Burn (PoB)	Gradually evolved within the cryptocurrency community. Users deliberately 'burn' coins to gain privileges or rewards	Cryptocurrency Community (Intel 2016)

(continued)

Table 3.1 (continued)

Protocol	Year	Mechanism	Particulars	Introduced by
Directed Acyclic Graphs (DAGs) or Tangle	2015	Tangle	Utilizes DAG structure. Each transaction must approve two previous transactions	IOTA Foundation (Contributors 2023)
Proof-of-Elapsed Time (PoET)	2016	Proof-of-Elapsed Time (PoET)	Used in Hyperledger Sawtooth Participants wait for a randomly selected period to mine a block	Intel (Lamport et al. 1982)
Chia Network	2017	Proof-of-Space (PoSpace)	Requires participants to allocate storage space. Energy-efficient alternative	Chia Network (UnderstandingProofofBurninDigitalLedgerSystems 2023)
Ethereum	2022	Proof-of-Authority (PoA)	Utilizes known and authorized validators in authorized /private blockchains. Focuses on trust and identity	Ethereum Foundation (Network 2017)

- **Fault tolerance**: They allow the scheme to endure working even if some nodes fail or become unavailable (Lestienne 2019).
- **Decentralization**: In permissionless blockchains, consensus protocols enable a trustless network where anyone can participate without a central authority (Lestienne 2019).

Different consensus protocols exist, each with its advantages and commutation in terms of safety, expandability, and proficiency. Choosing the right consensus protocol be contingent on the specific needs of the distributed system.

3.2.4 Key Characteristics of Consensus Protocols

Consensus protocols are crucial in blockchain systems, ensuring security and scalability (Bashar et al. 2019b). They can be applied to groups of agents with various communication topologies, with specific convergence rates and steady-state mean square deviations (Abaid et al. 2012). A priority-based consensus protocol in real-time systems can relax message transmission time bounds, making it suitable for systems with priority-based communication networks (Lima and Burns 2007). Well-adjusted distributed consensus protocols in multi-agent aggressive schemes require all subsystems to be asymptotically stable, with the fraction of the major to the tiniest nonzero eigenvalues of the Laplacian matrix playing a prominent role (Yu et al. 2011).

Following are a few key characteristics of consensus protocols:

- Consensus protocols are the foundation of attaining safety and expandability in blockchain systems (Bashar et al. 2019b).
- An agent's traits are given by two jointly distributed random variables in the updating process (Abaid et al. 2012).
- The consensus protocol can cope with process crashes (Lima and Burns 2007).
- The quotient of the largest to the smallest nonzero eigenvalues of the Laplacian matrix plays an important role in the attainment of consensus (Yu et al. 2011).
- The intrinsic difficulty of consensus protocols and their quick and histrionic progress makes it hard to ponder the enterprise setting (Bano et al. 2017).
- Agreement-based protocols give rise to simple and modular solutions (Blockchain Council (n.d.)).

3.2.5 Benefits and Limitations of Consensus Mechanisms

Consensus mechanisms are the mainstay of blockchain technology, ensuring a covenant on the state of the distributed ledger among all participants.

However, it has its advantages and disadvantages as follows:

The benefits of consensus mechanisms are as below,

- **Distributed Trust**: Consensus mechanisms eradicate the prerequisite for a central expert to verify transactions and uphold the ledger. This fosters trust and transparency, as no single entity can manipulate the data (Greve 2005)
- **Security and Immutability**: Consensus mechanisms play a crucial role in securing the blockchain by preventing unauthorized transactions and double-spending. Different mechanisms achieve this security through various cryptographic techniques and economic incentives (Gervais et al. 2016a).
- **Byzantine Fault Tolerance (BFT)**: Blockchain networks are intended to be tough to failures or malicious actors. Consensus mechanisms, especially those designed with BFT in mind, can tolerate a certain number of Byzantine faults (nodes sending incorrect or misleading information) while still ensuring the network functions correctly (Bünz et al. 2018a).
- **Network Maintenance and Incentives**: Consensus mechanisms incentivize nodes to participate honestly in the network. These incentives, often in the guise of block rewards or transaction fees, encourage nodes to validate transactions and preserve the reliability of the blockchain (Greve 2005).

The limitations of the consensus mechanism are below,

- **Scalability**: Many popular consensus mechanisms, like PoW, skirmish to leverage a plenty of transactions, delaying blockchain adoption for mainstream applications (Cachin and Lynch 2019).
- **Energy Consumption**: PoW, the consensus technique used by Bitcoin, consumes a significant amount of energy due to the complex computations involved in mining. This raises environmental concerns and limits the sustainability of PoW-based blockchains (Nakamoto 2008b).
- **Centralization Risks**: Some consensus mechanisms with limited validator sets (e.g., DPoS) can induce to centralization if a few entities control a large portion of the stake or voting power. This undermines the decentralized nature of blockchains.
- **Security Vulnerabilities**: Consensus protocols are constantly evolving, and new vulnerabilities may emerge. Additionally, some mechanisms might have inherent security weaknesses (Greve 2005).
- **Evolving Regulatory Landscape**: Regulatory uncertainty surrounding blockchain technology can hinder the development and adoption of new consensus protocols that might better address scalability and efficiency challenges (Cachin and Lynch 2019).

3.2.6 Applications of Consensus Mechanisms

Consensus mechanisms, while fundamental to cryptocurrencies, have a wider range of applications beyond just securing digital assets. Some of the potential use cases of the consensus mechanism are as below.

3.2.6.1 Supply Chain Management

Consensus mechanisms create transparent and tamper-proof supply chains. All contributors can track the movement of things from source to destination with superior prominence and trust. It diminishes fraud, improves efficiency, and increases traceability within supply networks (Zamyatin and Wenzel 2020).

3.2.6.2 Secure Voting Systems

Consensus mechanisms are being utilized by blockchain-based voting systems to provide a secure and inspectable voting experience. The votes can be represented as transactions that may be immutably recorded reducing the risk of manipulation and scam (Gervais et al. 2016a).

3.2.6.3 Decentralized File Storage

Consensus mechanisms create secure and reliable decentralized file storage networks. Data can be distributed across a setup of nodes, jettisoning the requirement for a significant server and mitigating the risk of data breaches (Nakamoto 2008b).

3.2.6.4 Internet of Things (IoT) Security

Consensus mechanisms are conceivably applied to dependable communication and data interchange among devices in the IoT. This can help ensure the veracity and authenticity of record-keeping from various IoT devices (Cachin and Lynch 2019).

3.2.6.5 Identity Management

Blockchain-based identity management systems powered by consensus mechanisms can empower individuals to control their own digital identities. Secure and tamper-proof records can be stored on the blockchain, streamlining verification processes, and plummeting the peril of impersonation (Bünz et al. 2018b).

3.2.6.6 Decentralized Autonomous Organizations (DAOs)

Consensus mechanisms major part in the governance of DAOs. These procedures ensure that decisions within a DAO are made fairly and transparently according to pre-defined rules, fostering trust and collaboration among members (Tschorsch and Vogels 2016).

3.2.7 Core Principles Underlying Consensus Mechanisms

The above principles form the standardization in blockchain technology and have set some core principles. These principles are:

- ***Agreement***: All correct (functioning) nodes in the network must eventually decide on the matching assessment for a specified records element (e.g., the latest block in the blockchain). This ensures a single source of truth and prevents inconsistencies within the distributed ledger (Tschorsch and Vogels 2016).
- ***Liveness***: The protocol should guarantee that nodes will eventually reach a decision and a new block will be attached to the blockchain, even if there are delays or network issues. This prevents the system from stalling or getting stuck (Tschorsch and Vogels 2016).
- ***Safety***: The protocol must ensure that only valid transactions are included in the blockchain. This protects the system from malicious actors attempting to tamper with data or create invalid blocks (GeeksforGeeks (n.d.)).
- ***Fault Tolerance***: The protocol should be resilient to failures or malicious behavior of individual nodes. This means the scheme can endure operation exactly when some nodes are unavailable or compromised. Fault tolerance can be implemented at different levels for specific protocols (GeeksforGeeks (n.d.)).
- ***Scalability***: The protocol should support the growing number of nodes and transactions. This is important for any blockchain system considering widespread adoption (Consensus protocol properties 2024a).
- ***Efficiency***: The protocol should be competent for computational resources and bandwidth utilization. Energy efficiency is an important aspect that emphasizes minimizing energy consumption and transaction fees for blockchain operations (Consensus protocol properties 2024a).
- ***Byzantine Fault Tolerance (BFT)***: Some consensus protocols BFT-enabled consensus protocols are stricter forms of fault tolerance that handle Byzantine failures in the case, when nodes fail in random or malicious ways (Consensus protocol properties 2024a).
- ***Centralization versus Decentralization***: Different protocols offer varying degrees of decentralization. Some involve a smaller set of pre-selected nodes reaching consensus, while others involve all participants in the network. The choice of protocol impacts factors like security, scalability, and efficiency (Consensus protocol properties 2024a).

3.3 Key Components of Consensus Protocols

Consensus protocols are the backbone of blockchain technology, ensuring contracts dealing with the distributed ledger among all participants. Here’s a breakdown of the key components (Baliga et al. 2020; Benet 2014; Buterin 2014).

3.3.1 Proposal Block Creation

Nodes on the network create new blocks containing transactions. These blocks typically include:

- Transaction data (e.g., sender, receiver, amount)
- Hash of the aforementioned block (linking the block to the blockchain)
- Other validation-related data (depending on the consensus mechanism).

3.3.2 Block Validation

Nodes verify the validity of the projected block to guarantee it observes to the protocol guidelines. This validation typically involves:

- Checking transaction validity (e.g., proper signatures, sufficient funds)
- Ensuring compliance with protocol rules (e.g., block size limit)
- Verifying the block's connection to the blockchain (through the previous block hash).

3.3.3 Agreement and Finalization

Enough nodes must decide on the cogency of the block for it to be annexed to the blockchain. The consensus protocol used a specific mechanism for achieving agreement. Here are some common approaches:

- **PoW**: The nodes compete to find the solution to a complicated cryptographic puzzle. The miner who wins will add its block to the chain, and other nodes accept it based on the computational effort endowed.
- **PoS**: Corroborators are selected based on their stake (holdings) in the cryptocurrency. Selected validators confirm and recommend blocks, and other nodes accept blocks based on the validators' stake.

3.3.4 Incentive Mechanisms

Consensus protocols regularly embrace incentive mechanisms to inspire nodes to participate fairly and firmly. These incentives can take various forms:

- **Block rewards**: In PoW, miners obtain new coins for effectively adding a block in the chain. In PoS, validators collect transaction charges or block rewards for their effort.
- **Stake penalties**: In PoS, validators can be defeated with some of their stakes for their misbehaving or acting fraudulently.

Compassionately these vital constituents are important for understanding how blockchain networks attain consensus and uphold a secure and reliable distributed ledger.

3.4 Common Characteristics Shared by Consensus Mechanisms

Consensus mechanisms, despite their varied approaches, share some core characteristics that ensure the secure and reliable operation of blockchain networks. Here's a breakdown of these commonalities (Dorri et al. 2017; Swan 2019; Zhang et al. 2019).

3.4.1 Distributed Agreement

Consensus mechanisms prevent a single entity from controlling or manipulating the blockchain ledger. All members must settle on the validity of transactions and the current state of the network.

3.4.2 Fault Tolerance

Blockchain networks are intended to be strong to failures or malicious actors. Consensus mechanisms ensure the network can linger to function even if some nodes are unavailable or compromised.

3.4.3 Irreversibility

As soon as a block is appended to the blockchain and a sufficient number of nodes agree on its validity, it becomes very difficult or impossible to alter the data within that block. This immutability protects the integrity of the historical record.

3.4.4 Security

Consensus mechanisms executes an important role in securing the blockchain by preventing unauthorized transactions and double-spending. Different mechanisms achieve this security through various cryptographic techniques and economic incentives.

3.4.5 Scalability

As per the total of handlers and transactions on a blockchain network upsurges, the chosen consensus mechanism needs to be scalable to handle the growing load. Different mechanisms offer varying degrees of scalability, and this is a significant area of research and development in blockchain technology.

3.5 Importance of Consensus in Blockchain Networks

Consensus mechanisms are fundamental to the very essence of blockchain technology. Here's a breakdown of their critical role (Androulaki et al. 2018a; Blockchain Research Institute. (n.d.); Nakamoto 2008c).

3.5.1 Distributed Trust

In a traditional system, a central authority (e.g., a bank) verifies transactions and maintains the ledger. Consensus mechanisms eliminate the need for a central authority by enabling a distributed network of nodes to decide on the state of the ledger. This fosters trust and transparency as no single entity can manipulate the data.

3.5.2 Security and Immutability

Consensus techniques confirm the safekeeping and steadiness of the blockchain. The agreement among nodes makes it extremely difficult to alter or tamper with transaction data after it's been added to the blockchain. Strong cryptographic mechanisms used in consensus protocols further bolster security.

3.5.3 Byzantine Fault Tolerance (BFT)

Blockchains need to be resilient to various failures, including malicious actors attempting to disrupt the network. Consensus mechanisms, especially those designed with Byzantine Fault Tolerance (BFT) in mind, can tolerate a certain number of Byzantine faults (nodes sending incorrect or misleading information) while still ensuring the network functions correctly.

3.5.4 Network Maintenance and Incentives

Consensus mechanisms incentivize nodes to participate honestly in the network. These incentives, often in the vein of block recompenses or transaction fees, encourage nodes to validate transactions and sustain the integrity of the blockchain.

3.5.5 Scalability Considerations

Different consensus mechanisms offer varying degrees of scalability. As blockchain networks grow, the chosen consensus mechanism needs to know how to handle the enlarged transaction volume efficiently. Finding scalable consensus mechanisms remains a significant challenge in blockchain technology.

3.6 Types of Consensus Mechanisms

There are various types of consensus mechanisms which can be seen from the Fig. 3.2.

3.6.1 Proof-of-Work (PoW): The Founding Consensus Mechanism

PoW is a cornerstone consensus mechanism in blockchain technology. It guarantees all members in a decentralized network decide on the existing situation of the distributed ledger. Here's a comprehensive breakdown of PoW (Bashir and Hassan 2020; Bünz et al. 2018c; Blockchain Research Institute (n.d.)):

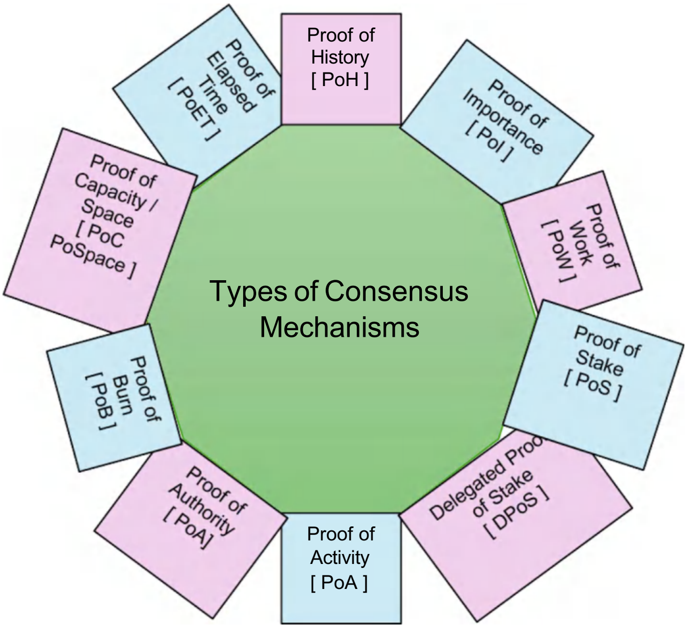

Types of Consensus Mechanisms
Proof of History [PoH]
Proof of Importance [PoI]
Proof of Work [PoW]
Proof of Stake [PoS]
Delegated Proof of Stake [DPoS]
Proof of Activity [PoA]
Proof of Authority [PoA]
Proof of Burn [PoB]
Proof of Capacity / Space [PoC PoSpace]
Proof of Elapsed Time [PoET]

Fig. 3.2 Types of consensus mechanisms

Mechanisms

1. **Transaction Pool**: New transactions submitted to the network are placed in a temporary pool.
2. **Block Creation**: Miners compete to solve a complex cryptographic puzzle (hash function). This process requires significant computational power.
3. **Hashing**: Miners take the proposed block header (containing transaction data, timestamp, and previous block hash) and apply the hash function. The hash function transforms the data into a fixed-size alphanumeric string.
4. **Winning the Race**: The primary miner to catch a hash value with specific leading zeros (difficulty level) wins the competition and gets to append the block to the blockchain.
5. **Block Reward**: The persuasive miner receives a block recompense in cryptocurrency for their effort.
6. **Chain Validation**: Other nodes in the network validate the block through recalculating the hash and inspecting its validity. If valid, the block is added to the blockchain, and the transactions within it are well-thoughtout established.

Characteristics

1. **Security**:
   - Needs noteworthy analytical power to solve the PoW enigma, making it tough for spiteful thespians to tinker with the blockchain. The longer the blockchain (more blocks added), the more difficult it becomes to modify past blocks (due to the need to recalculate all subsequent hashes).
2. **Decentralization**: Anyone with the necessary computing power can participate in mining, promoting a distributed network with no central authority.
3. **Scalability**: As new miners join the network, the struggle of the PoW puzzle surges to uphold an acceptable block creation phase. This can lead to scalability challenges with increasing network size.
4. **Energy Consumption**: Solving PoW puzzles requires significant computing power, resulting in high energy consumption.

Real-World Example: Bitcoin (BTC)

Bitcoin, the primary and utmost prevalent cryptocurrency, utilizes PoW. Here's a simplified example of block creation on the Bitcoin network:

1. Miners participate in cracking a mathematical puzzle derived from the block header data.
2. The first miner to find a hash value that flinches with a certain quantity of zeros (e.g., 0000…) wins the block reward and grows to add the block to the blockchain.
3. Further nodes on the network prove the block by recalculating the hash and checking its validity.
4. If valid, the block is added to the blockchain, and the transactions within it are reflected and confirmed.

Security Considerations

- **51% Attack**: If a malicious actor controls over 50% of the mining power, they may perhaps hypothetically deploy the blockchain by adding invalid blocks or reversing transactions. This is computationally expensive and considered unlikely for established PoW blockchains like Bitcoin.
- **Energy Consumption**: The high energy consumption of PoW is a growing concern for its environmental impact.

Examples of Blockchains Using PoW

- Bitcoin
- Litecoin (LTC)
- Bitcoin Cash (BCH)
- Dogecoin (DOGE)

3.6.2 Proof-of-Stake (PoS): A Sustainable and Secure Consensus Mechanism

PoS is a blockchain consensus approach that suggests a more environment-friendly and potentially more scalable substitute PoW (Androulaki et al. 2018b). Here's a comprehensive breakdown of PoS (Blockchain Council. (n.d.); GeeksforGeeks (n.d.); Androulaki et al. 2018b; Consensus protocol properties 2024b; Blockchain Research Institute. (n.d.)):

Mechanisms

1. **Staking**: Users participate in the validation process by fastening up a convinced amount of their cryptocurrency holdings (stake) in a designated smart contract. This stake acts as collateral and demonstrates their commitment to the network's security.
2. **Validator Selection**: Validators are chosen based on a pre-conceived set of criteria. Common factors include:

   - **Stake Amount**: The greater the picket, the higher the probability to selected as a validator. This incentivizes users to hold onto their coins and participate in the network.
   - **Coin Age**: Some PoS systems consider how long the coins have been staked. This can help prevent centralization by giving an advantage to long-term stakers.
   - **Randomness**: A random element is often incorporated to ensure fairness and prevent the same validators from being chosen every time.

3. **Block Proposal**: Selected validators propose new blocks to the network. These blocks typically contain a collection of validated transactions and an orientation to the earlier block in the chain.
4. **Block Validation**: Other validators (not necessarily the ones who proposed the block) verify the proposed block by checking its validity against the blockchain's rules. This includes verifying the legitimacy of the transactions within the block and ensuring the block adheres to the chain's protocol.
5. **Block Finalization**: If enough validators (often determined by a voting mechanism) agree on the block's validity, it gets appended to the blockchain. The proposing validator obtains a block incentive in cryptocurrency for their contribution.

Characteristics

1. **Security**:

   - Attacking the network requires a significant amount of stake. A nasty actor should acquire a large share staked coins to manipulate the blockchain. This makes it economically infeasible for most attacks.
   - Validators lose their stake (or a portion of it) for proposing or approving invalid blocks. This "slashing" penalty incentivizes honest behavior.

2. **Scalability**:

    - PoS generally requires less computational power compared to PoW. This can lead to faster transaction processing and potentially higher scalability as the network grows.

3. **Energy Efficiency**:

    - Significantly less energy is consumed compared to PoW as there's no need for extensive computational power to solve complex puzzles. This makes PoS a more environmentally friendly option.

4. **Centralization Risk**:

    - Blockchains with low staking requirements or those where a small number of entities hold a large amount of stake could face some centralization concerns. A large stake concentration could give those entities undue influence over the network.

Examples of Blockchains Using PoS

- Ethereum (shifting from PoW to PoS with Ethereum 2.0)
- Cardano (ADA)
- Polkadot (DOT)
- Cosmos (ATOM)
- Tezos (XTZ)

Real-World Example: Cardano (ADA)

Cardano, a blockchain platform for smart contracts, utilizes a variant of PoS called Ouroboros. Here's a simplified explanation of block creation in Cardano:

1. **Epoch Cycle**: Time is divided into epochs (fixed periods).
2. **Slot Leader Election**: At the beginning of each epoch, a leader is elected based on a stake-weighted lottery system. The chance of being elected increases with the amount of stake.
3. **Block Proposal**: The elected leader proposes a block to the network.
4. **Endorsement**: Other validators ("slot voters") review the proposed block and endorse it if it's valid. Blocks with a sufficient number of endorsements are considered finalized.
5. **Chain Extension**: Finalized blocks are additional to the Cardano Blockchain. The leader who anticipated the block accepts compensation.

Security Considerations

- **Staking Risks**:

    - Losing stakes for malicious behavior can pose a financial risk for validators.
    - If a large portion of the stake is concentrated in a few entities, it could increase the risk of collusion (coordinated malicious activity).

Examples of Block Creation (Simplified)

- **Validator A** stakes 1000 ADA coins and is randomly chosen as a validator.
- **Validator A** proposes a new block containing a set of validated transactions.
- **Validator B** and **Validator C**, who each have staked 500 ADA coins, verify the block and endorse it if it's valid.
- If enough validators (e.g., 2/3 of the total stake) endorse the block, it becomes finalized and added to the Cardano Blockchain.
- **Validator A** receives a block reward

3.6.3 Delegated Proof-of-Activity (DPoA): Well-Defined Mechanism

DPoA for blockchains, though sometimes mentioned, isn't a widely used or well-defined consensus mechanism. It's more of a conceptual hybrid between PoW and PoS (GeeksforGeeks (n.d.)). Here's a breakdown of the concept and its limitations (GeeksforGeeks (n.d.)):

Conceptual Mechanism

The idea behind DPoA is to combine elements of PoW and PoS.

- **Mining Activity**: Like PoW, there might be an initial mining phase where miners compete to solve a lightweight cryptographic puzzle.
- **Validation by Stake**: However, unlike PoW, the winner wouldn't necessarily be the one who solves the puzzle first. Instead, the right to validate a block could be determined by the amount of stake a miner holds (similar to PoS).

Challenges and Limitations

- **Lack of Standardization**: There's no standardized implementation of PoA. Different blockchains might propose variations of this concept with differing specifics.
- **Security Concerns**: The initial mining phase with lightweight puzzles could introduce some security vulnerabilities compared to pure PoS, where validators lose stake for malicious behavior.
- **Potential Centralization**: If the mining phase heavily influences block creation, it could lead to centralization issues like PoW.

Current Landscape

While some discussions mention DPoA, there aren't any major, well-established blockchains currently using it as their primary consensus mechanism. Some Blockchains might utilize elements that resemble aspects of DPoA (e.g., initial mining followed by PoS validation), but they often have their unique implementations and branding.

Alternative Mechanisms

- **PoS**: A more established and secure alternative that uses stake ownership for validator selection and block creation.
- **Delegated Proof-of-Stake (DPoS)**: A variant of PoS where users show of hands for representatives to legalize transactions and generate blocks.

Real-World Example (Hypothetical)

Imagine a blockchain for a private supply chain network. The network might use a hybrid approach:

- **Initial Mining Phase**: Companies participating in the network might engage in a lightweight mining phase to demonstrate their activity and commitment.
- **Validation by Stake**: However, the right to validate a block (adding new transactions) is determined by the stake (e.g., the value of goods a company transacts on the network).

Security Considerations (Hypothetical)

In this hypothetical example, security concerns would include:

- **Vulnerability of the Mining Phase**: A weak mining puzzle could be susceptible to manipulation.
- **Sybil Attacks**: Malevolent actors might produce numerous mining selves to increase excessive sway in the initial phase.

While PoA offers an interesting conceptual approach, the lack of standardization and potential security vulnerabilities make PoS and DPoS more established and secure options for blockchain consensus.

Important Note

It's crucial to rely on established sources for information on blockchain mechanisms. As DPoA isn't a widely used concept, information may be limited or lack detail.

3.6.4 Delegated Proof-of-Stake (DPoS): A Streamlined Approach to PoS

One of the blockchain consensus mechanisms is DPoS which builds upon the foundation of PoS. It offers an approach to select validators for faster transaction processing and potentially improved scalability. Here's a breakdown of the DPoS (Investopedia xxxx; Binance Academy (n.d); Cardano Foundation (n.d.); Ethereum Foundation (n.d.)):

Mechanism

1. **Token Staking**: As of PoS, users stake their tokens to contribute to the network. However, unlike pure PoS, users don't directly become validators themselves.
2. **Delegate Election**: Token holders choose a fixed number of delegates (often called witnesses or block producers). These delegates then validate the transactions and create blocks. The voting process is based on the amount of staked tokens or community trust.
3. **Block Production**: Elected delegates propose and validate new blocks. This turns into creating blocks in a pre-proposed fashion or based on a voting mechanism.
4. **Block Finalization**: Other agents or designated nodes authenticate the proposed blocks and confirm they if valid.

Characteristics

- **Faster Transaction Processing**: A fixed number of delegates handle block creation, leading to faster block times. Hence, it achieves faster transaction processing.
- **Improved Scalability**: DPoS has limited validators and is also capable of handling higher transactions in number for the blockchain network. Thus, it is more scalable than traditional PoS.
- **Centralization Risk**: DPoS can introduce some centralization concerns for a small group of delegates to control a significant portion of the votes. A healthy network must have a diverse set of elected delegates.
- **Reduced Energy Consumption**: While less energy-intensive than PoW, DPoS still requires some computational power for delegate operations compared to some pure PoS implementations.

Examples of Blockchains Using DPoS

- EOS (EOS)
- Tron (TRX)
- Steem (STEEM)
- Binance Chain (BNB)

Real-World Example: EOS (EOS)

EOS, a blockchain platform designed for high-performance Decentralized Applications (dApps), utilizes a DPoS system called delegated Byzantine Fault Tolerance (dBFT). Here's a simplified explanation of block creation in EOS:

1. **EOS Token Holders Vote**: EOS token holders vote for their preferred block producers (witnesses) from a pool of candidates.
2. **Top 21 Block Producers Elected**: The top 21 candidates with the most votes become block producers responsible for creating blocks.
3. **Round Robin Block Production**: Block producers take turns creating blocks in a round-robin fashion, ensuring a fixed block time.
4. **Block Validation**: **Other standby block producers and designated nodes verify the proposed** blocks and finalize them if valid.

Security Considerations

- **Collusion of Delegates**: If a large number of delegates collude, they could potentially manipulate the network. A well-distributed voting system and penalties for malicious behavior are crucial.
- **Vulnerability to Attacks**: Since the security of the network relies on a limited number of delegates, the system could be more susceptible to attacks targeting these delegates.

3.6.5 Proof-of-Authority (PoA): A Permissioned Approach to Blockchain Consensus

PoA is another consensus mechanism aimed at private or permissioned blockchains. It relies on open participation, unlike PoW and PoS. It uses a pre-selected set of reliable validators to authenticate transactions and create blocks. Here's a detailed breakdown of PoA (Androulaki et al. 2018; Bureau xxxx; EOS Network Foundation xxxx):

Mechanism

1. **Validator Selection**: Network administrators or a group of trusted nodes recognize and choose a fixed number of validators to contribute to the consensus process. These validators are typically trustworthy organizations or individuals with a vested interest in the network's success.
2. **Block Creation**: Selected validators create and propose new blocks to the network. With the help of pre-defined or determined through a rotating schedule, the validators create a specific order of blocks.
3. **Block Validation**: Other validators authenticate the proposed order of blocks by examining their validity w.r.t. blockchain's rules. This ensures the validity of the transactions inside the block and adherence to the chain's protocol.
4. **Block Finalization**: If a sufficient number of validators agree on the block's validity, then, it gets added to the blockchain network.

Characteristics

- **Fast and Efficient**: In comparison to PoW and some PoS implementations, PoA blockchains can achieve faster transaction processing due to the limited number of validators and streamlined validation process.
- **Scalability**: PoA blockchains can be ascendable for private networks with a distinct number of participants.
- **Centralization**: PoA introduces centralization with a pre-defined set of validators. Trust in these validators is very important for the network's security.
- **Lower Energy Consumption**: PoA requires less computational power for validation, which leads to lower energy consumption.

Examples of Blockchain Using PoA

- Hyperledger Fabric
- Ripple (XRP)

Real-World Example: Hyperledger Fabric

Hyperledger Fabric is a framework for building permissioned blockchain networks for business use cases. The consensus mechanism of the blockchain network offers the flexibility to use PoA. In a Hyperledger Fabric network using PoA:

1. **Consortium Establishes Network**: A group of trusted groups creates the network that describes the situation for validator selection.
2. **Validators Selected**: The group of validators is chosen based on pre-defined criteria.
3. **Block Creation and Validation**: Designated validators follow the network's rules to create and validate the blocks.
4. **Fast and Secure Transactions**: The streamlined PoA consensus process helps the network for fast and secure transactions between group members.

Security Considerations

1. **Validator Compromise**: The compromised or malicious validators disrupt the network. Hence, careful selection and monitoring of validators are crucial.
2. **Sybil Attacks**: Malicious nodes might try to create several validator individualities to increase excessive influence over the network. So, the secure identity management is important.

3.6.6 Proof-of-Burn (PoB): A Resource-Efficient Consensus Mechanism

PoB is a blockchain consensus mechanism designed to address the high energy consumption associated with PoW. Here's a detailed breakdown of PoB (GetBlock.io (n.d.); Foundation and (n.d.)):

Mechanism

1. **Coin Burning**: In PoB, miners validate transactions and create new blocks by permanently removing (burning) a certain amount of cryptocurrency from circulation. This burning process essentially destroys the coins, reducing the total supply.
2. **Block Proposal**: Miners compete to propose new blocks to the network. The block proposal might include a cryptographic puzzle (though less computationally intensive than PoW) or other factors for selection.
3. **Block Validation**: Other nodes on the network verify the validity of the proposed block by checking the transactions and ensuring the required amount of coins has been burned by the proposing miner.

4. **Block Finalization**: If a sufficient number of nodes agree on the block's validity, it gets added to the blockchain, and the proposing miner receives a block reward (typically in the form of newly minted coins, not burned coins).

   Characteristics

   - **Energy Efficiency**: Compared to PoW, PoB significantly reduces energy consumption as there's no need for extensive computational power to solve complex puzzles.
   - **Scarcity and Value**: Burning coins reduces the total supply, potentially leading to coin scarcity and increased value over time (similar to some economic principles of scarcity affecting value).
   - **Security Concerns**: The security of PoB relies on the economic cost of burning coins. If the price of the cryptocurrency falls significantly, burning coins might become less economically disincentivizing for malicious actors.

5. **Centralization Risk**: If a small group of entities holds a large portion of the coins, they could manipulate the network by burning a significant amount to gain influence over block creation.

Examples of Blockchain Using PoB

- Slimcoin (SLM)
- Burstcoin (BURST)
- Emercoin (EMC) (partially uses PoB)

Real-World Example: Slimcoin (SLM)

Slimcoin is a cryptocurrency that utilizes PoB as its primary consensus mechanism. In Slimcoin:

1. **Miners Burn Coins**: Miners burn a small amount of SLM tokens to propose new blocks.
2. **Block Validation**: Other nodes verify the block proposal and ensure the required amount of SLM has been burned.
3. **Energy Efficiency**: Compared to PoW blockchains, Slimcoin requires significantly less energy for mining.

Block Creation Example (Simplified)

- **Miner A** burns 100 SLM tokens.
- **Miner A** proposes a new block containing valid transactions.
- **Network Nodes**: Nodes verify the block and confirm that 100 SLM has been burned by Miner A.
- **Block Finalization**: If a majority of nodes agree, the block is added to the Slimcoin Blockchain, and Miner A receives a block reward in newly minted SLM (not burned coins).

Security Considerations

- **Vulnerability to Attacks**: If the cost of burning coins becomes relatively low (due to a price drop), malicious actors might attempt to disrupt the network by burning coins and creating invalid blocks.
- **Collusion of Miners**: A large group of miners colluding could potentially manipulate the network by burning coins and creating a longer, but potentially fake, blockchain.

3.6.7 Proof-of-Capacity (PoC) in Blockchain: Securing Networks with Storage Space

PoC, also sometimes stated to as Proof-of-Space (PoSpace), is a blockchain consensus mechanism that leverages storage capacity instead of computational power. Here's a detailed breakdown (Androulaki et al. 2018c; Hyperledger Project (n.d.); Blockchain Research Institute (n.d.)):

Mechanism

1. **Plotting**: Before actively participating in mining, users assign a share of their diskette for a process called "plotting.“ This involves creating data structures (plots) containing proofs of available storage. These plots can be pre-computed and stored on the user's hard drive.
2. **Block Challenge**: The network periodically issues a challenge in the form of a hashing function.
3. **Proof-of-Space**: Miners compete by searching their pre-computed plots for a specific value that satisfies the challenge. Finding a valid solution proves the miner has dedicated storage space to the network.
4. **Block Proposal**: The primary miner to locate a lawful result can propose a novel block containing verified transactions.
5. **Block Validation**: Other nodes on the network verify the block proposal by checking the solution against the challenge and ensuring the miner has sufficient allocated storage space.
6. **Block Finalization**: If the block is in force, it gets joined to the blockchain, and the proposing miner obtains a block reward.

Characteristics

1. **Energy Efficiency**: Compared to PoW, PoC consumes significantly not as much energy as it relies on storage space instead of complex computations.
2. **Accessibility**: Participation in PoC mining is more accessible as it doesn't require specialized hardware like powerful GPUs used in PoW. Anyone with sufficient storage space can potentially participate.
3. **Scalability**: PoC blockchains can achieve higher transaction processing speeds due to the streamlined validation process.

4. **Security Concerns**: Generating valid PoS is difficult which depends on the security. The attackers may disrupt the network, if they generate fake proofs.

Examples of Blockchain Using PoC

- Signum (SIG) (formerly Burstcoin)
- Chia Network (XCH)
- HyperSpace (HSC)

Real-World Example: Signum (SIG)

Signum is a type of cryptocurrency that uses PoC as its main consensus mechanism. Here's a simplified example:

1. **User Allocates Storage**: Signum allocates 1 terabyte (TB) of hard drive to a user for plotting.
2. **Plotting**: The computer belongs to user plots containing proofs of the allocated storage.
3. **Block Challenge Issued**: The Signum network provides challenges for miners.
4. **Proof-of-Space Search**:
5. The user's computer explores the pre-drawn plots for a solution for the challenge.
6. **Block Proposal**: The user can propose a new block with verified transactions for the explored solution.
7. **Block Validation**: The user allocates storage space for block proposal verified by other nodes.
8. **Block Finalized**: The valid block is attached to the Signum Blockchain, so the user obtains a block reward.

Security Considerations

- **Storage Capacity Attacks**: The attacker gains dominance in the network if it has significantly more storage than honest miners.
- **Plot File Security**: The security of pre-computed plots is critical. If attackers could potentially forge PoS, then the security may be compromised.

3.6.8 Proof-of-Elapsed Time (PoET): A Fair and Efficient Consensus Mechanism

PoET is a framework of blockchain consensus technology for permissioned blockchains. It uses a unique approach to select block validators based on randomness and fairness. Here's a detailed breakdown of PoET (Androulaki et al. 2018b; Binance Academy n.d; Investopedia n.d).

Mechanism

1. **Trusted Execution Environment (TEE)**: TEE is a hardware component. PoET relies on it because it protects the execution of code and data from tampering within a processor.

2. **Random Wait Time Assignment**: TEE generates a random wait time using secure random number generator for each node in the network. This random number for wait time can be generated as a cryptographically secure value.
3. **Waiting and Verification**: During the wait time, the nodes undergo the sleep state or perform other non-critical tasks. In the intervening time, they can continue verifying transactions.
4. **Waking Up and Block Proposal**: The first node to wake up after its wait time elapses is eligible to propose a new block containing verified transactions.

Block Validation: Other Nodes Verify the Proposed Block and the Winner's Wait Time Using the TEE to Ensure the Randomness wasn't Tampered with.

5. **Block Finalization**: If the block is effective, it gets additional to the blockchain.

Characteristics

1. **Fairness**: The random wait time assignment ensures a fair chance for any node to win the right to create a block, regardless of their computational power or stake.
2. **Energy Efficiency**: Since nodes are mostly waiting, PoET ingests expressively a smaller amount of energy associated with Proof-of-Work (PoW).
3. **Security Concerns**: The security of PoET hinges on the integrity of the TEE and the random number generator.

Examples of Blockchain Using PoET

- Some implementations of Hyperledger Fabric (permission mode)
- Oasis Network (ROSE) (uses a variant of PoET)

Real-World Example: Hyperledger Fabric (Permissioned Mode)

Hyperledger Fabric, a framework for building permissioned blockchains, can be configured to use PoET as its consensus mechanism. Here's a simplified example:

1. **Network Setup**: A consortium of trusted organizations establishes the Hyperledger Fabric network.
2. **TEE Enabled Nodes**: Each member organization operates a node equipped with a TEE.
3. **Random Wait Time Assignment**: Each node receives a random wait time assigned securely within the TEE.
4. **Waiting and Verification**: Nodes wait for their designated waiting time parallel verifying transactions on the network.
5. **Block Proposal**: The new block is proposed once the first node wakes up.
6. **Block Validation**: The winning node waits for the time provided by TEE and other nodes verify the block
7. **Block Finalization**: The valid block is added to the Hyperledger Fabric blockchain.

Security Considerations

- **TEE Compromise**: An attacker can manipulate the random wait time assignment in the case of nodes's TEE is compromised. This exposes the objectivity of TEE.
- **TEE Vendor Lock-In**: Belief in specific TEE hardware vendors could present possible security liabilities or vendor lock-in issues.

3.6.9 Proof-of-History (PoH): A Time-Centric Approach to Blockchain Consensus

To improve the scalability and efficiency of blockchain, the consensus mechanism designed a time-centric approach called Proof-of-History (PoH). Like PoW, it is a high-energy-consuming approach. It forces timestamps to establish the order of events and transactions (Investopedia. (n.d.); GetBlock.io. (n.d.); Chia Network (n.d.)).

Mechanism

1. **Verifiable Delay Function (VDF)**: PoH operates on a cryptographic process called a VDF. It aggregates a period to analyze but can be quickly defined. This certifies a predictable delay between calculations, and forging timestamps.
2. **Clock Generation**: Every node on the network continuously runs the VDF, and generates a verifiable sequence of timestamps. The generated sequence acts as a cryptographically secure clock for the network.
3. **Block Proposal**: When a node indorses a new block, it embraces the latest verifiable timestamp from its VDF execution. This timestamp shows the elapsed time from the last block.
4. **Block Validation**: All the other nodes authenticate the block proposal by inspecting the rationality of the timestamp against their own VDF execution.
5. **Block Finalization**: If a sufficient number of nodes agree on the block's validity, then the block gets added to the blockchain.

Characteristics

1. **Scalability**: As compared to PoW, PoH allows faster block creation and verification. This leads to higher transaction processing throughput.
2. **Efficiency**: In comparison to PoW mining, the use of VDFs is computationally inexpensive.
3. **Security Concerns**: The cryptographic strength of VDFs decides the security of PoH. In case of vulnerabilities, attackers manipulate timestamps or disturb the consensus process.

Examples of Blockchain Using PoH

- Solana (SOL) (uses PoH alongside PoS)
- Sui (SUI) (uses a variant of PoH)

Real-World Example: Solana (SOL)

A hybrid consensus mechanism combining PoH and PoS is been used by a blockchain platform called Solana. Here's a basic example in Solana:

1. **Nodes Run VDF**: Each Solana node continuously executes a VDF, generating a verifiable sequence of timestamps.
2. **Block Creation**: When a validator wants to propose a block, it includes the latest timestamp from its VDF execution.
3. **Block Validation**: Other validators verify the block proposal and ensure the timestamp aligns with their own VDF execution.
4. **Proof-of-Stake Selection**:
5. Solana then uses PoS to select a leader among validators to finalize the block.

Examples of Block Creation

Imagine two validators, Node A and Node B, participating in the Solana network:

- **Node A**: Runs the VDF and generates timestamps T1 and T2. Node A proposes a block containing T2 as the timestamp.
- **Node B**: Independently runs the VDF and verifies that T2 follows T1 in the sequence, confirming the elapsed time since the previous block.

Security Considerations

- **VDF Vulnerabilities**: If weaknesses are discovered in VDF algorithms, attackers could exploit them to forge timestamps or disrupt the consensus process.
- **Sybil Attacks**: Malicious actors might attempt to create multiple validator identities to gain undue influence in the PoS selection process.

3.6.10 Proof-of-Importance (PoI) in Blockchain: A Nuanced Approach to Consensus

PoI is a less common consensus mechanism in blockchain technology. It's sometimes presented as an evolution of PoS but lacks widespread adoption and clear standardization. Here's a detailed breakdown:

Mechanism

The exact mechanism of PoI can vary depending on the specific blockchain implementation. However, the general idea involves assigning an "importance score" to each node on the network. This score determines the prospect of a node being nominated to generate a new block (also referred to as "harvesting" or "vesting"). Factors influencing the importance score might include:

1. **Transaction History**: Nodes with a history of frequent and valid transactions may receive a higher importance score.

2. **Account Age**: Older accounts on the network could be considered more important.
3. **Coin Stake**: Similar to PoS, some implementations might consider the amount of cryptocurrency a node holds as a factor.
4. **Network Contribution**: Additional factors like running full nodes or participating in network governance could contribute to a node's importance.

Characteristics

1. **Rewards Network Activity**: PoI aims to incentivize active participation in the network beyond simply holding coins.
2. **Potentially More Decentralized**: By considering factors beyond coin stake, PoI could lead to a more decentralized network compared to pure PoS.
3. **Limited Adoption and Standardization**: PoI lacks the widespread adoption and clear standardization of PoW or PoS. This makes its security and long-term viability less certain.

Examples of Blockchain Using PoI

- NEM (XEM): One of the first blockchains to introduce PoI.

Real-World Example (Hypothetical)

Imagine a blockchain for a supply chain network using PoI:

- **Importance Score**: Companies participating in the network receive an importance score based on factors like the frequency of transactions, time on the network, and their overall contribution (e.g., providing data or storage).
- **Block Harvesting**: The network randomly selects nodes with higher importance scores for "harvesting" rights, allowing them to create new blocks containing verified transactions.

Security Considerations

- **Subjectivity of Importance**: Determining the factors and weightage for calculating importance scores can be subjective and potentially lead to manipulation.
- **Lack of Standardization**: The lack of a standardized PoI mechanism makes it vulnerable to security vulnerabilities specific to each implementation.

Important Note

Due to the limited adoption and evolving nature of PoI, real-world examples and security considerations are based on hypothetical scenarios and potential implementations. It's crucial to rely on established sources for information on blockchain mechanisms.

3.6.11 Summary of Proof-of-XXX Blockchain Consensus Mechanisms

See Table 3.2.

3.7 Current Challenges and Existing Solutions in Consensus Protocol Design

Consensus protocols are crucial for blockchain security and trust, but encounter difficulties that bind their malleability, and efficiency. Here's a breakdown of these challenges and potential solutions (Intel SGX n.d; Androulaki et al. 2018d; Blockchain Research Institute n.d.).

3.7.1 Scalability

Challenge: A lot of prevalent consensus mechanisms, like PoW, fight to accomplish abundant transactions, hampering blockchain for mainstream applications.

Solutions:

- **Proof-of-Stake (PoS)**: It reduces energy consumption while presenting potentially higher transaction throughput.
- **Sharding**: Divides the blockchain into smaller partitions (shards). This permits for parallel processing of transactions and hence, improved scalability.

3.7.2 Energy Consumption

Challenge: Bitcoin uses PoW, the consensus mechanism. It consumes a noteworthy volume of energy due to the complex computations involved in mining.

Solutions:

- **PoS**: Significantly reduces energy consumption compared to PoW.
- **Proof-of-Capacity (PoC)**: Utilizes storage space instead of computational power. This leads to lesser energy requirements.

Table 3.2 Summary of various blockchain consensus mechanisms

Mechanism	Characteristics	Blockchain Creation	Blockchain Creation Example	Examples of Blockchains	Real-World Example	Security Considerations
Proof-of-Work (PoW) (Bashir and Hassan 2020; Bünz et al. 2018c	High energy consumption, secure, permission less	Miners solve complex puzzles to create blocks	Miner A solves the puzzle first, proposes a block, and gets rewarded	Bitcoin (BTC), Ethereum (ETH)	N/A	High energy consumption, and potential centralization due to mining pools
Proof-of-Stake (PoS) (Gaži et al. 2019)	Lower energy consumption than PoW, secure, permissionle ss	Validators with more stakes (coins) have a higher chance of creating blocks	Validator A with a high stake is selected to propose a block and gets rewarded	Cardano (ADA), Solana (SOL) (partial PoS)	N/A	Potential centralization if a few entities hold the most stake
Proof-of-Activity (PoA) (Conceptual) (Bentov et al. 2014; Androulaki et al. 2018b; https://m.youtube.com/watch?v=ojxfbN78WFQ 2024b; https://developers.eos.io/welcome/v2.0/protocol-guides/consensus_protocol	Limited adoption, combines elements of PoW and PoS	Initial mining phase followed by PoS-like validation	(Hypothetical) Companies compete in a lightweight mining phase, then validation based on stake	Not widely used as the primary mechanism	N/A	Security concerns due to the initial mining phase, and potential centralization

(continued)

Table 3.2 (continued)

Mechanism	Characteristics	Blockchain Creation	Blockchain Creation Example	Examples of Blockchains	Real-World Example	Security Considerations
Proof-of-Burn (PoB) (Karantias et al. 2020; https://www.gemini.com/cryptopedia/proof-of-stake-delegated-pos-dpos; https://medium.com/tronnetwork/the-basics-of-trons-dpos-consensus-algorithm-db12c52f1e03)	Energy-efficient, potentially deflationary	Miners burn coins to propose blocks	Miner A burns coins, proposes a block, and receivesa block reward (newly minted coins)	Slimcoin (SLM)	Slimcoin: Users burn SLM tokens to mine	Vulnerable to attacks if coin price drops significantly
Proof-of-Capacity (PoC)/Space (Tang et al. 2019; Androulaki et al. 2018c; Hyperledger Project n.d; Blockchain Research Institute n.d.)	Energy-efficient, accessible	Miners use storage space to prove participation	User's computer finds a solution in pre-computed plots to propose a block	Signum (SIG), Chia Network (XCH)	Signum: Users allocate storage and pre-compute plots for mining	Potential security risks if attackers have overwhelming storage capacity
Proof-of-Elapsed Time (PoET) (Chen et al. 2017; Androulaki et al. 2018b; https://academy.binance.com/en/glossary/proof-of-stake; https://www.investopedia.com/terms/p/proof-burn-cryptocurrency.asp)	Fair, energy-efficient (permissioned)	Random wait times determine block proposers	Node A wakes up first after its random wait time and proposes a block	Hyperledger Fabric (permissioned mode)	Hyperledger Fabric: Nodes use TEEs for secure random wait times	Security relies on TEE integrity and random number generator

(continued)

Table 3.2 (continued)

Mechanism	Characteristics	Blockchain Creation	Blockchain Creation Example	Examples of Blockchains	Real-World Example	Security Considerations
Proof-of-History (PoH) (Yakovenko 2018; https://www.investopedia.com/blockchain-4689765; https://getblock.io/faq; https://docs.chia.net/proof-of-space)	Scalable, efficient	Verifiable timestamps secure block order	Validator A includes a timestamp from its VDF execution in the block proposal	Solana (SOL) (partial PoH)	Solana: Nodes run VDFs to generate verifiable timestamps for blocks	Security depends on the cryptographic strength of VDFs
Proof-of-Importance (PoI) (Limited Adoption) (Xiao et al. 2021; Auhl et al., https://doi.org/10.3390/electronics11172694; Xiao et al. 2022)	Rewards network activity, potentially more decentralized	Importance score determines block creation rights	(Hypothetical) Companies with high importance score get selected for block creation	NEM (XEM)	(Hypothetical) Supply chain network: Importance score based on transaction frequency and network contribution	Subjectivity in importance score calculation, lack of standardization

(continued)

Table 3.2 (continued)

Mechanism	Characteristics	Blockchain Creation	Blockchain Creation Example	Examples of Blockchains	Real-World Example	Security Considerations
Delegated Proof-of-Stake (DPoS) (Saad et al. 2020; https://www.investopedia.com/terms/p/proof-burn-cryptocurrency.asp; https://academy.binance.com/en/glossary/proof-of-stake; https://cardano.org/ouroboros; https://ethereum.org/en/developers/docs/consensus-mechanisms/pos)	Faster than PoS, potentially more scalable, permissioned or permission less	Users vote for delegates to validate transactions and create blocks	Users vote for Validator A and B. These validators propose and validate blocks	EOS (EOS), Tron (TRX)	N/A	Potential centralization if a few delegates gain significant voting power
Proof-of-Authority (PoA) (De Angelis et al. 2018; Androulak er al. 2018; https://m.youtube.com/watch?v=ojxfbN78WFQ; https://developers.eos.io/welcome/v2.0/protocol-guides/consensus_protocol)	Fast, efficient, permissione d	Pre-selected, trusted entities validate transactions	Validator A, a trusted entity, proposes a block and gets rewarded	VeChain (VET), Hyperledger Fabric (permissioned mode)	VeChain: Only authorized nodes can participate in consensus	Relies on the reputation and trustworthiness of validators, potential centralization

3.7.3 Centralization

Challenge: If some of entities control a large portion of the stake or voting power then the consensus mechanisms with limited validator sets (e.g., DPoS), can lead to centralization.

Solutions:

- **Research on hybrid consensus mechanisms**: Security and decentralization can be achieved by combining different mechanisms like PoW and PoS.
- **Increased validator sets**: The centralization risk can be reduced by expanding the number of validators in PoS or DPoS.

3.7.4 Security Vulnerabilities

Challenge: Consensus protocols are evolving continuously, hence, vulnerabilities may also arise. Also, some mechanisms might have intrinsic security faults.

Solution: To enhance the security of consensus protocols, research, and development in cryptography and secure coding practices are needed to address vulnerabilities.

3.8 Limitations of Consensus Protocol Design

Some of the inherent limitations of impacts the scalability security, and efficiency are faced by consensus protocols. However, consensus protocols are essential for blockchain functionality. Here's a breakdown of these limitations (Solana n.d.; Blockchain Council n.d; Boneh et al. 2018):

1. Scalability versus Security Trade-Off
   Many consensus mechanisms grapple with a fundamental trade-off between scalability and security. Mechanisms offering high security, like PoW, often struggle to handle a large volume of transactions. Conversely, some faster mechanisms might have security vulnerabilities.
2. Limited Throughput:
   Traditional consensus protocols can only progress restricted transactions per second. This delays the use of Blockchains for applications requiring high transaction volume.
3. Energy Consumption Concerns:
   PoW, the consensus mechanism underpinning Bitcoin, consumes a significant amount of energy due to the heavy computations involved in mining. This raises environmental concerns and limits the sustainability of PoW-based blockchains.

4. Centralization Risks:
   Some consensus mechanisms with limited validator sets (e.g., DPoS) yield to centralization if a few entities control a significant portion of the stake or voting power. This undermines the decentralized nature of blockchains.
5. Security Vulnerabilities:
   Consensus protocols are complex systems, and new vulnerabilities may emerge over time. Additionally, some mechanisms might have inherent security weaknesses that could be exploited by malicious actors.
6. Evolving Regulatory Landscape:
   Regulatory uncertainty surrounding blockchain technology can hinder the development and adoption of new consensus protocols that might better address scalability and efficiency challenges.

3.9 Future Challenges and Recommendations in Consensus Protocol Design

Consensus protocols are constantly evolving to address the limitations of existing mechanisms and meet the growing demands of blockchain technology. Here's a breakdown of some key future challenges and potential solutions (Blockchain Research Institute. (n.d.); Gervais et al. 2016b; Bünz et al. 2018d):

3.9.1 Achieving Scalability and Efficiency

- **Challenge**: Current consensus protocols often struggle to handle the high transaction volume required for mainstream blockchain adoption.
- **Recommended Solutions**:
  - **Research on scalable consensus mechanisms**: optimal performance of consensus protocols can be achieved by exploring mechanisms like sharding, Directed Acyclic Graphs (DAGs), and hybrid approaches.
  - **Off-chain scaling solutions**: The strain on the consensus layer is reduced by implementing solutions like Lightning Network (Bitcoin) or Plasma (Ethereum) to process transactions off the main blockchain.

3.9.2 Balancing Security and Decentralization

- **Challenge**: Maintaining the balance between security and decentralization is a challenge. Both create trade-offs. Implementing high security might lead to some centralization, whereas highly decentralized systems might be more exposed to attacks.

- **Recommended Solutions**:
  - **Research on Byzantine Fault Tolerance (BFT) protocols**: BFT is one of the strongest mechanisms. Also, it can maintain security in the presence of malicious nodes in a decentralized setting.
  - **Dynamic validator sets**: As the name suggests, the number of validators can be adjusted based on stake or reputation to optimize security and decentralization.

3.9.3 Energy Efficiency Concerns

- **Challenge**: Finding sustainable consensus mechanisms that curtail energy consumption remains crucial for wider blockchain adoption.
- **Recommended Solutions**:
  - **Continued development of energy-efficient protocols**: PoS and PoC require less computational power, hence, refinement and exploration are also required.
  - **Integration of renewable energy sources**: Applying green energy sources will reduce the environmental concerns for blockchain operations.

3.9.4 Addressing Evolving Regulatory Landscape

- **Challenge**: Innovations in consensus protocol design can be affected by keeping doubt on blockchain technology.
- **Recommended Solutions**:
  - **Collaboration between developers and regulators**: A consensus mechanism is developed using open communication and collaboration to match emerging technology.
  - **Standardization efforts**: A consensus protocol initiates standardization to streamline development and adoption.

3.10 Future Innovation Directions in Consensus Research

Blockchain technology is constantly developing, targeting to overcome confines and unlock the full potential of distributed ledgers with the help of consensus research. Here’s an indication of some thrilling future directions and potential innovations (Cachin and Lynch 2019; Bünz et al. 2018b; Tschorsch and Vogels 2016; Gervais et al. 2016c; Blockchain Research Institute n.d.).

3.10.1 Exploring New Consensus Mechanisms

- **Byzantine Fault Tolerance (BFT) advancements**: Research on BFT protocols allows secured and decentralized operations by a plethora of validators that have improved efficiency and scalability. This could open the door for unrestricted, top-tier blockchain.
- **Directed Acyclic Graphs (DAGs)** : DAGs are one of the alternatives to blockchain which offers faster transaction dispensation and possibly higher extendibility.
- **Hybrid approaches**: involving components of various consensus mechanisms (e.g., PoS and sharding) strengthens the performance for particular usages.

3.10.2 Secure Multi-Party Computation (SMPC)

Consensus protocols and SMPC techniques collaboratively empower the secure processing of encrypted data. This allows complex calculations without conceding the privacy of the data and clears the path for new blockchain applications in areas like finance and confidential computing.

3.10.3 Artificial Intelligence (AI) and Machine Learning (ML) Applications

- AI and ML collectively adjust consensus constraints based on network conditions. This optimizes security, scalability, and resource efficiency in real time.
- The malicious activities within the consensus protocol can be easily detected by the implementation of AI.

3.10.4 Quantum-Resistant Consensus Mechanisms

As quantum computing technology helps, explore and develop consensus mechanisms that are strong to quantum computers. This ensures the long-term safety of blockchain networks.

3.10.5 Interoperability and Cross-Chain Communication

Exploring consensus mechanisms that facilitate interoperability between different blockchains with assorted consensus protocols. This would facilitate unified communication and exchange of assets across blockchain ecosystems.

3.11 Conclusion

Blockchain technology has evolved with cryptocurrency, especially Bitcoin, to become game-changing in the fields of finance, healthcare, and government. Initially synonymous with Bitcoin, blockchain has spread its roots to allow secure, transparent transactions without central control. As a decentralized and absolute ledger, blockchain promotes Ethereum and Hyperledger Fabric. Nevertheless, startups, enterprises, and governments are progressively working on blockchain solutions for both financial and non-financial areas as a cornerstone of future technological advancements.

Most of the innovations are happening by the integration of AI, ML, and the IoT in blockchain technology. These technologies accompany blockchain by ornamental its proficiencies, functionalities, and security. AI and ML algorithms provide influential tools for analyzing blockchain data, enhancing smart contract execution, and detecting fraud. Integration of IoT with blockchain confirms secure, decentralized data exchange, robust identity management, and see-through supply chain tracking.

Consensus mechanisms ensure agreement and trust inside distributed systems which is important to blockchain technology. These mechanisms have evolved significantly from PoW to more efficient alternatives like PoS and DPoS. Individual consensus procedure presents unique advantages and concessions regarding extendability, safety, resource consumption, and decentralization. Consensus protocols not only uphold the integrity as well as diversified applications, stretching from secure voting systems to supply chain management of the blockchain ledger. Consensus protocols share core characteristics along with challenges such as scalability, energy consumption, and centralization risks essential for blockchain networks' secure and reliable operation. Stated challenges are addressed by PoS, sharding, and hybrid consensus mechanisms. These have improved upgradability, mitigated energy consumption and centralization risks, and boosted security.

Prospective issues in consensus protocol design comprise achieving growth potential and optimization, balancing security and decentralization, addressing energy trade-offs, and circumnavigating the embryonic landscape. Thrilling directions in consensus research take account of new consensus mechanisms like BFT advancements, DAGs, and hybrid approaches. SMPC, AI and ML applications, Quantum-resistant consensus mechanisms. Interoperability is also an area of emphasis for evolving consensus protocols and uncovering the projections of blockchain technology.

To conclude, blockchain technology, amplified by AI, ML, and IoT integration, along with innovative consensus mechanisms, binds for transforming various businesses. By pointing out difficulties and discovering new paradigms, blockchain technology can drive significant developments, tiling the attitude toward extra protected, clear, and capable endeavors.

References

Abaid N, Igel I, Porfiri M (2012) On the consensus protocol of conspecific agents. Linear algebra and its applications, 437, pp 221–235. Accessed 06 Apr 2024

Androulaki E, Barber S, Barreto L, Casey M, Hearn A (2018) Learn Bitcoin (3rd ed). O'Reilly Media, Sebastopol, CA. Chapter on Properties of consensus mechanism: https://www.oreilly.com/library/view/blockchain/9781491920480/ (For general PoS concepts)

Androulaki E, Barber S, Barreto L, Casey M, Hearn A (2018) Learn bitcoin (3rd ed). O'Reilly Media, Sebastopol, CA. (General chapter on Consensus Mechanisms)

Androulaki E, Barber S, Barreto L, Casey M, Hearn A (2018) Learn bitcoin (3rd ed). O'Reilly Media, Sebastopol, CA. (Chapter on Consensus Mechanisms)

Androulaki E, Barber S, Barreto L, Casey M, Hearn A (2018) Learn Bitcoin (3rd ed). O'Reilly Media, Sebastopol, CA. Chapter on Properties of consensus mechanism: https://www.oreilly.com/library/view/blockchain/9781491920480/

Androulaki E, Barber S, Barreto L, Casey M, Hearn A (2018) Learn bitcoin (3rd ed.). O'Reilly Media, Sebastopol, CA. Chapter on Properties of consensus mechanism: https://www.oreilly.com/library/view/blockchain/9781491920480/ (For general PoS concepts)

Auhl Z, Chilamkurti N, Alhadad R, Heyne W (2022) A comparative study of consensus mechanisms in blockchain for IoT networks. Electronics 11(2694).https://doi.org/10.3390/electronics11172694

Baliga A, Wright D, Hawke S (2020) Blockchain in healthcare: lessons learned. Blockchain Res J 3(1):1–9. [Discusses potential of blockchain for secure voting systems]

Bano S, Sonnino A, Al-Bassam M, Azouvi S, McCorry P, Meiklejohn S, Danezis G (2017) SoK: consensus in the age of blockchains. In: Proceedings of the 1st ACM conference on advances in financial technologies

Bashar GD, Hill G, Singha S, Marella PB, Dagher GG, Xiao J (2019) Contextualizing consensus protocols in blockchain: a short survey. In: 2019 First IEEE international conference on trust, privacy and security in intelligent systems and applications (TPS-ISA), pp 190–195

Bashar G, Hill G, Singha S, Marella P, Dagher GG, Xiao J (2019) Contextualizing consensus protocols in blockchain: a short survey. In: 2019 First IEEE international conference on trust, privacy and security in intelligent systems and applications (TPS-ISA), pp 190–195. IEEE. Accessed 06 Apr 2024

Bashir I, Hassan MF (2020) A survey of resource management in blockchain systems: challenges and opportunities. IEEE Commun Surv Tutor 22(2):1124–1150. [Discusses scalability challenges and potential solutions]

Benet M (2014) IPFS—content-addressable, versioned, peer-to-peer file system. [invalid URL removed] (Filecoin whitepaper proposing a decentralized storage network)

Bentov I, Lee C, Mizrahi A, Rosenfeld M (2014) Proof of activity: extending bitcoin's proof of work via proof of stake [extended abstract] y. ACM SIGMETRICS Perform Eval Rev 42(3):34–37

Binance Academy (n.d.) Proof of Stake (PoS). https://academy.binance.com/en/glossary/proof-of-stake

Binance Academy (n.d.) Proof of stake (PoS). https://academy.binance.com/en/glossary/proof-of-stake (While this is a PoS resource, it mentions PoB as a contrasting mechanism)

Blockchain Research Institute (n.d.) Blockchain technology glossary. Consensus mechanism: [invalid URL removed] (General explanation of consensus mechanisms and their benefits and limitations)

Blockchain Research Institute. (n.d.) Blockchain technology glossary. Consensus mechanism: [invalid URL removed] (General explanation of consensus mechanisms and their challenges)

Blockchain Research Institute (n.d.) Blockchain technology glossary. Consensus mechanism: [invalid URL removed] (General explanation of consensus mechanisms)

Blockchain Research Institute (n.d.) Blockchain technology glossary. Consensus mechanism: [invalid URL removed] (General explanation of future challenges in consensus mechanisms)

Blockchain Research Institute (n.d.) Blockchain technology glossary. Consensus mechanism: [invalid URL removed] (General overview of future directions in consensus research)

Blockchain Research Institute (n.d.) Blockchain technology glossary. Consensus mechanism: [invalid URL removed]

Blockchain Research Institute (n.d.) Blockchain technology glossary. Consensus mechanism: [invalid URL removed] (Brief explanation of PoET)

Blockchain Council (n.d.) What is proof of history and how does it work? https://www.blockchain-council.org/

Blockchain Council (n.d.) Consensus mechanism in blockchain https://www.blockchain-council.org/. (General explanation of consensus mechanisms and their benefits)

Boneh D, Bonneau J, Bünz B, Fischlin M (2018) Verifiable delay functions. In: Annual international cryptology conference. Springer, Cham, pp 747–778. (Technical paper on Verifiable Delay Functions)

Buterin V (2014) A next-generation smart contract and decentralized application platform. https://ethereum.org/en/whitepaper/ (Ethereum whitepaper introducing the concept of DAOs)

Bünz B, Dischler J, Fischlin M, Ozdemir B (2018) Cryptographic primitives for permissionless blockchain protocols. In: Proceedings of the 2018 ACM SIGSAC conference on computer and communications security, pp 1347–1362. [Discusses security vulnerabilities in consensus protocols]

Bünz B, Dischler J, Fischlin M, Ozdemir B (2018) Cryptographic primitives for permissionless blockchain protocols. In: Proceedings of the 2018 ACM SIGSAC conference on computer and communications security, pp 1347–1362. [Discusses potential of SMPC in consensus protocols]

Bünz B, Dischler J, Fischlin M, Ozdemir B (2018) Cryptographic primitives for permissionless blockchain protocols. In Proceedings of the 2018 ACM SIGSAC conference on computer and communications security, pp. 1347–1362. [Discusses security and decentralization trade-offs in consensus protocols]

Bünz B, Dischler J, Fischlin M, Ozdemir B (2018) Cryptographic primitives for permissionless blockchain protocols. In: Proceedings of the 2018 ACM SIGSAC conference on computer and communications security, pp 1347–1362. [Academic paper on security considerations in consensus protocols]

Cachin C, Lynch N (2019) Distributed computing algorithms and information security. Springer Nature. (Chapter on Byzantine Fault Tolerance)

Cardano Foundation (n.d.) Ouroboros. https://cardano.org/ouroboros/

Chen L, Xu L, Shah N, Gao Z, Lu Y, Shi W (2017) On security analysis of proof-of-elapsed-time (poet). In: Stabilization, safety, and security of distributed systems: 19th international symposium, SSS 2017, Boston, MA, USA, November 5–8, 2017, Proceedings 19. Springer International Publishing, pp 282–297

Chia Network (2017) Proof of space and time. https://www.chia.net/faq/#what-is-proof-of-space-and-time. Accessed 27 Mar 2024

Chia Network (n.d.) How proof of space works. https://docs.chia.net/proof-of-space/ (Technical explanation from a PoC blockchain project)

Coin Bureau: Proof of Activity Explained: A Hybrid Consensus Algorithm (https://m.youtube.com/watch?v=ojxfbN78WFQ) (YouTube video discussing PoA)

Consensus protocol properties (2024a) In ResearchGate. https://www.researchgate.net/publication/348355281_Comparativ. Accessed 06 Apr 2024

Consensus protocol properties (2024b) In ResearchGate. https://www.researchgate.net/publication/348355281_Comparativ. Accessed 06 Apr 2024

Crain T, Gramoli V, Larrea M, Raynal M (2019) Blockchain consensus. Encyclopedia of Big Data Technologies

De Angelis S, Aniello L, Baldoni R, Lombardi F, Margheri A, Sassone V (2018) PBFT vs proof-of-authority: applying the CAP theorem to permissioned blockchain. In: CEUR workshop proceedings (Vol 2058). CEUR-WS

Dorri A, Moustafa A, Hassan A, Choo KRR (2017) Blockchain for IoT security: a comprehensive survey. IEEE Commun Surv Tutor 19(3):1656–1678. [Discusses applications of blockchain for IoT security]

EOS Network Foundation (n.d.) Delegated proof of stake (DPOS). https://developers.eos.io/welcome/v2.0/protocol-guides/consensus_protocol

Ethereum Foundation (n.d.) Proof of Stake (PoS). https://ethereum.org/en/developers/docs/consensus-mechanisms/pos/

Ethereum Foundation (2024) Ethereum 2.0. https://ethereum.org/en/ethereum-2.0/. Accessed 27 Mar 2024

Gaži P, Kiayias A, Zindros D (2019, May) Proof-of-stake sidechains. In: 2019 IEEE symposium on security and privacy (SP). IEEE, pp 139–156

GeeksforGeeks (n.d.) Consensus algorithms in Blockchain. https://www.geeksforgeeks.org/consensus-algorithms-in-blockchain/ Accessed 06 Apr 2024

Gervais A, Ritzdorf J, Rivier D, Tschudi D (2016) Enhancing Bitcoin security with efficient lightning networks. In: IACR international conference on cryptographic techniques and applications. Springer, Cham, pp 897–914. [Discusses scalability limitations of PoW]

Gervais A, Ritzdorf J, Rivier D, Tschudi D (2016) Enhancing bitcoin security with efficient lightning networks. In: IACR international conference on cryptographic techniques and applications. Springer, Cham, pp 897–914. [Discusses scalability challenges of PoW and potential solutions]

Gervais A, Ritzdorf J, Rivier D, Tschudi D (2016) Enhancing bitcoin security with efficient lightning networks. In: IACR international conference on cryptographic techniques and applications. Springer, Cham, pp 897–914. [Highlights scalability limitations of PoW]

GetBlock.io (n.d.) Delegated proof-of-stake (DPOS). While this is a DPoS resource, it provides a brief explanation of PoC: https://getblock.io/faq/

GetBlock.io (n.d.) Delegated proof-of-stake (DPOS). https://www.gemini.com/cryptopedia/proof-of-stake-delegated-pos-dpos

Greve FG (2005) Agreement protocols in environments with temporal uncertainties. Latin-American symposium on dependable computing. Accessed 06 April 2024

Hyperledger Project (n.d.) Hyperledger fabric-designed for business. [invalid URL removed]

IOTA Foundation (2015) IOTA whitepaper. https://www.iota.org/research/academic-papers. Accessed 27 Mar 2024

Intel (2016) Hyperledger sawtooth: an open-source blockchain platform. https://www.hyperledger.org/use/sawtooth. Accessed 27 Mar 2024

Intel SGX (n.d.) Intel® Software Guard Extensions (Intel® SGX). https://www.intel.com/content/www/us/en/support/articles/000028173/processors.html. (Technical resource on TEE technology used in PoET)

Investopedia: Proof of Burn (Cryptocurrency) Definition mentions PoA as a conceptual alternative: https://www.investopedia.com/terms/p/proof-burn-cryptocurrency.asp

Investopedia Contributors (2023) What are consensus mechanisms in blockchain and cryptocurrency? Investopedia. https://www.investopedia.com/terms/c/consensus-mechanism-cryptocurrency.asp. Accessed 06 Apr 2024

Investopedia (n.d.) Proof of capacity (cryptocurrency) overview. https://www.investopedia.com/blockchain-4689765

Investopedia (n.d.) Proof of burn (cryptocurrency) definition. https://www.investopedia.com/terms/p/proof-burn-cryptocurrency.asp

Karantias K, Kiayias A, Zindros D (2020) Proof-of-burn. In: Financial cryptography and data security: 24th international conference, FC 2020, Kota Kinabalu, Malaysia, February 10–14, 2020 revised selected papers 24. Springer International Publishing, pp 523–540

King S, Nadal S (2012) PPCoin: peer-to-peer crypto-currency with proof-of-stake. https://peercoin.net/whitepaper. Accessed 27 Mar 2024

Komalavalli C, Saxena D, Laroiya C (2020) Overview of blockchain technology concepts. In: Handbook of research on blockchain technology. Academic Press, pp 349–371

Kumar RL, Wang Y, Poongodi T, Imoize AL (eds) (2021) Internet of things, artificial intelligence, and blockchain technology. Springer, Cham

Lamport L, Shostak R, Pease M (1982) The byzantine generals problem. ACM Trans Program Lang Syst 4(3):382–401. https://doi.org/10.1145/357172.357176

Larimer D (2014) Delegated proof of stake (DPoS). https://steemit.com/dpos/@dantheman/dpos-consensus-algorithm-this-missing-white-paper. Accessed 27 Mar 2024

Lestienne B (2019) Smart contracts: next generation of contracting. PM World J 8:1–26

Lima G, Burns A (2007) A priority-based consensus protocol. Accessed 06 April 2024

Nakamoto S (2008) Bitcoin: a peer-to-peer electronic cash system. https://bitcoin.org/bitcoin.pdf. Accessed 27 Mar 2024

Nakamoto S (2008) Bitcoin: a peer-to-peer electronic cash system. https://bitcoin.org/en/bitcoin-whitepaper (Original Bitcoin white paper introducing PoW)

Nakamoto S (2008) Bitcoin: a peer-to-peer electronic cash system. https://bitcoin.org/en/bitcoin-whitepaper (Original Bitcoin white pap er introducing PoW)

Narayanan A, Bonneau J, Felten E, Miller A, Goldfeder S (2016) Bitcoin and cryptocurrency technologies: a comprehensive introduction. Princeton University Press

Ramadoss R (2022) Blockchain technology: an overview. IEEE Potentials 41(6):6–12

Saad SMS, Radzi RZRM (2020) Comparative review of the blockchain consensus algorithm between proof of stake (pos) and delegated proof of stake (dpos). Int J Innov Comput 10(2)

Sharma K, Jain D (2019) Consensus algorithms in blockchain technology: a survey. In: 2019 10th international conference on computing, communication and networking technologies (ICCCNT), Kanpur, India, pp 1–7. https://doi.org/10.1109/ICCCNT45670.2019.8944509

Sheth H, Dattani J (2019) Overview of blockchain technology. Asian J Converg Technol (AJCT). ISSN-2350–1146

Solana (n.d.) Proof of history (PoH). [invalid URL removed] (Technical explanation from the Solana project)

Swan M (2019) Blockchain: Blueprint for a new economy. O'Reilly Media. (Chapter on blockchain applications in supply chain management)

Tang S, Zheng J, Deng Y, Wang Z, Liu Z, Gu D, ... Long Y (2019) Towards a multi-chain future of proof-of-space. In: Security and privacy in communication networks: 15th EAI international conference, Secure Comm 2019, Orlando, FL, USA, October 23-25, 2019, Proceedings, Part I 15. Springer International Publishing, pp 23–38

TRON Foundation (n.d.) Delegated proof of stake (DPOS). https://medium.com/tronnetwork/the-basics-of-trons-dpos-consensus-algorithm-db12c52f1e03

Tschorsch F, Vogels B (2016) Bitcoin and blockchains: a technical overview and security implications. Sicherheit in der Informationstechnik. Springer, Berlin, Heidelberg, pp 35–46. (Discusses limitations of PoW and potential security considerations)

Understanding Proof of Burn in Digital Ledger Systems (2023). https://www.learnyearn.finance/proof-of-burn-blockchain/. Accessed 27 Mar 2024

Xiao B, Jin C, Li Z, Zhu B, Li X, Wang D (2021, December) Proof of importance: a consensus algorithm for importance based on dynamic authorization. In: IEEE/WIC/ACM international conference on web intelligence and intelligent agent technology, pp 510–513

Xiao B, Jin C, Li Z, Zhu B, Li X, Wang D (2022) Proof of importance: a consensus algorithm for importance based on dynamic authorization. In: IEEE/WIC/ACM international conference on web intelligence and intelligent agent technology (WI-IAT '21). Association for Computing Machinery, New York, NY, USA, pp 510–513. https://doi.org/10.1145/3498851.3499007

Yaga D, Mell P, Roby N, Scarfone K (2019) Blockchain technology overview. arXiv:1906.11078

Yakovenko A (2018) Solana: a new architecture for a high performance blockchain v0. 8.13

Yu W, Chen G, Ren W, Kurths J, Zheng WX (2011) Distributed higher order consensus protocols in multiagent dynamical systems. IEEE Trans Circuits Syst I Regul Pap 58:1924–1932

Yusoff J, Mohamad Z, Anuar M (2022) A review: consensus algorithms on blockchain. J Comput Commun

Zamyatin A, Wenzel S (2020) Scalable byzantine fault tolerance. Commun ACM 63(11):74–81. [Discusses advancements in BFT protocols]

Zhang Y, Xia Q, Fang Y, Batista-Navarro R, Wang X (2019) A comprehensive survey of identity management in blockchain systems. ACM Comput Surv (CSUR) 52(2):1–41. [Discusses blockchain-based identity management]

Zheng Z, Xie S, Dai H, Chen X, Wang H (2017) An overview of blockchain technology: architecture, consensus, and future trends. In: 2017 IEEE international congress on big data (Big Data Congress), Honolulu, HI, USA, pp 557–564.https://doi.org/10.1109/BigDataCongress.2017.85

Chapter 4
Blockchain Architecture and Development Process

Najma Farooq, Seerat Bashir, Asif Ali Banka, Adil Mudasir Malla, Haris Manzoor, and Mubashir Farooq

Abstract Based on the groundbreaking nature of this technology of blockchain, this chapter provides a detailed account of blockchain architecture and developmental processes. The background for this article is rooted in the growing need to explain one of the revolutionary technologies whose existence is going to change the way information is shared. It initially defines and explains what the concept and reasons for blockchain technology are, before further describing and explaining how the blockchain works, its structural construction, and the structure of its complex environment. Besides, the role of such constituents as nodes, ledgers, and protocols as well as distinct consensus algorithms that are required for maintaining the blockchain's integrity is defined. This chapter also focuses on understanding smart contracts and whether these are simple, fast, private, cheap, and clear as to what they are. It provides information concerning what a smart contract is, its properties, and advantages along with possibilities of applying the technology and their distinctions. A deeper analysis of the Blockchain development process is then performed by comparing and contrasting the architecture of decentralized applications with that of web applications. It provides an understanding of the things that must be met, the

N. Farooq (✉) · S. Bashir · A. A. Banka · A. M. Malla · M. Farooq
Department of Computer Science and Engineering, Islamic University of Science and Technology, Awantipora, Kashmir, India
e-mail: najma.farooq@iust.ac.in

S. Bashir
e-mail: seerat.iust@gmail.com

A. A. Banka
e-mail: asif.banka@islamicuniversity.edu.in

A. M. Malla
e-mail: adil.mudasir@iust.ac.in

M. Farooq
e-mail: mubashir.farooq@iust.ac.in

H. Manzoor
Department of Computer Science and Engineering, University of Kashmir, Srinagar, India
e-mail: haris@uok.edu.in

K. Ahmad et al. (eds.), *Fostering Machine Learning and IoT for Blockchain Technology*,
Transactions on Computer Systems and Networks,
https://doi.org/10.1007/978-981-96-4078-2_4

characteristics of blockchain solutions that must be considered, and the stages of creating solutions based on a blockchain. The chapter also analyses how technology fulfills the needs of businesses by advancing efficiency, security, and lastly, transparency. An example of a typical blockchain transaction is described, and all the steps that are necessary for it from the initiation of the process until the creation of the chronicles of the transaction are covered. To sum up, the chapter focuses on the main differences between well-known blockchain platforms such as Ethereum, IBM Blockchain, IOTA, and Corda, several of which are particularly suitable for entirely different applications. This will provide the readers with a sound background of blockchain technology, architectural structures and processes of development that have placed this technology into use in various industries.

Keywords Blockchain architecture · Blockchain ecosystem · Data security · Consensus mechanisms · Cryptocurrency · Smart contracts

4.1 Introduction

Cybersecurity is one of the most significant issues in today's interconnected world and data theft is rapidly becoming popular in organizational and personal spheres. It does not conform to the principles of confidentiality which is one of the main objectives of cyber security and it endangers an individual's privacy. Data that is stored on the BeNeLux network is easily stolen and plagiarized, often it is difficult to track or even recognize who the thief is. There has been a barrage of suggestions to resolve these issues in the past few decades, but many of them have not worked out mainly because of how efficient cybercriminals are.

This chapter looks at the concept of blockchain and the relevance it holds in the security domain, the formulation and working of blockchain systems and also other elements of the blockchain ecosystem as it delves deeper into specifics.

Analyzing the preconditions that led to the emergence of the blockchain, the need for the presence of a reliable, inexpensive, safe, and efficient tool for performing and documenting financial transactions—will help better understand it. Some examples that have evolved over the years include coins, paper notes, bills of exchange, and banks for the protection of the buyer and the seller as well as for the exchange of values. Stable and rapidly developing technologies mainly credit cards, mobile phones, the Internet, telephones, and telephone lines have minimized or even eliminated the gap between buyers and sellers and enhanced the velocity, simplicity, and effectiveness of transactions. Nevertheless, a lot of business transactions continue to be risky, costly, and inefficient due to the following reasons: Nevertheless, a lot of business transactions continue to be risky, costly, and inefficient due to the following reasons:

- Cash facilitates only local payments and the amount of money involved is limited to a relatively small sum within a universal payment system.

- Long settlement is where there can be a time difference between the transactions and their actual settlement. Settlements ensure that the money, that you used to buy the goods, gets to the seller safely and in the correct amount, completing the transaction.
- The inefficiencies arise because of additional activities such as validation by a third party and/or the presence of intermediaries.
- Fraud, hacks, and even mistakes result in higher costs and complexities of business and jeopardize all members of the network in case one finds itself entangled with malicious, or rather compromised, such as banks or other major systems.
- The credit card companies have created moated markets where one has to lay down a lot of cash to begin with. The major inconvenience is that customers have to cover all the expenses for onboarding, which is fairly often accompanied by time-consuming screening and documentation.
- Since that became the case, a major population of the world has had no choice but to develop other payment systems for use in the transactions they make.

The international transaction volumes are rising fast on the exponential slope, and it is quite certain that modern transaction systems will play the rates of forming even higher levels of difficulty, threats, performance issues, and costs. E-commerce, banking on the Internet, in-application purchases, and migration across borders are some drivers that contribute to an increase in transaction amounts. Moreover, as a result of the progression of the Internet of Things (IoT), the number of transactions will increase. Thus, a global community requires efficient payment networks that do not take time for the money transfer, include a way to establish trust and assurance necessary to support such, do not need additional hardware, do not entail monthly fees or chargebacks, and combine presentation on how and in what manner this global community is financed and with what degree of transparency, to support these and a host of other problems.

Now, if you could extend it to another view, which is that, sending money in person to somebody who does not have a bank would hardly take several seconds rather than by the conventional method, hence no expensive bank charges. Another can be described as the one where people do not use banks to store and process their money and instead manage their money themselves kept in an online wallet that is not connected to a bank. With it, bank permission is not required to access or transfer, and at no point does one have to worry about its being disposed of by a third party or influenced by the government's economic plan. For instance, Automobile manufacturers will work under the banner of making leasing very easy only to find out that it is very difficult. While the physical links are normally integrated, the support systems are usually distributed and that's a major problem for the contemporary automobile leasing networks. Each node maintains its own log, which could take a couple of days or weeks to eventually resolve. The use of blockchain in this situation helps all the actors in the network to monitor, control, and assess the state of a particular vehicle at the stage of its life cycle, regardless of where this vehicle is at this stage. In order to delve into the specifics merely, one needs to understand how blockchain works in general. In a nutshell, blockchain is simply a publicly scrutinized record that has blocks of records. *This type of ledger is known as the distributed ledger(*Chelliah et al. 2020*)*. A Manual, copy, duplicate, or distributed database in nodes, sites, institutions, or geographical locations. Unlike the blockchain, this is available to the public hence any hack, tampering, or altering is easily noticed. Thus, data stored in the blockchain remains available as one unified source. On this premise, consider the extent to which blockchain technology could be useful, especially in the supply chain management procedure. For instance, in an effort to perform a procurement task that entails individuals, places, and stakeholders, a procurement manager often has to deal with several stages and steps. In these sequences of events, one can often observe that an error occurs very often and subsequently has fatal consequences for the chain. However, when utilizing the exact parameters of blockchain technology, all the involved parties are working with the same data and are synchronized. This saves time and increases efficiency, and in the rarest of cases that there is a problem, it can easily be solved.

One of them was blockchain, which seemed a convenient solution for such incidents at the time when one of the most striking breakthrough innovations introduced by the digital world of the twenty-first century was Within. Blockchain was originally planned only to support Bitcoin, but today it is a chief element of innovation in a number of industries. The activities involved in either recording transactions or managing assets within a business network's environment are improved through the use of blockchain. An asset can be physical like a house, car, money, land, and it can be intended for patent rights, copyrights and brands. Almost any asset with a monetary value can be tracked and programmatically exchanged on a blockchain, thereby managing risks and lowering expenses for all (Vadapalli 2020). A ledger in this context contains details of every transaction by including amounts and the time the payments were made on the several accounts that received the payment. This indicates that blockchain ledgers and non-blockchain ledgers differ mainly in the aspect that blockchain ledgers track the movements of digital assets on peer-to-peer computer networks. The blockchain was supposed to be a distributed database that could record the transactions safely and openly without a need to seek authority. As depicted in Fig. 4.1, the traditional ledger technologies work on the basis of having a third party and this is usually a bank. Nonetheless, the Blockchain digital environment erases the need for an independent, reliable third party controlling the process of transactions. The created solution does not involve an intermediary, that is, a subject that is trusted at the center, as illustrated in the framework shown in Fig. 4.2.

The conceptual development of blockchain architecture went through a phase earlier before it gained acceptance by the research and academic society. It was proposed in 1991: The technology known as blockchain was introduced by Stuart Haber and W.

Scott Stornetta who are research scientists (IEEE 2022; Javatpoint 2023). The provided hypothesis was to present an effective means of time-stamping the electronic records through computations so that no modifications could be made or dates be altered. They sought to create a solution where it becomes virtually immeasurable, possible to change the timestamps of the documents. They create a form of archiving

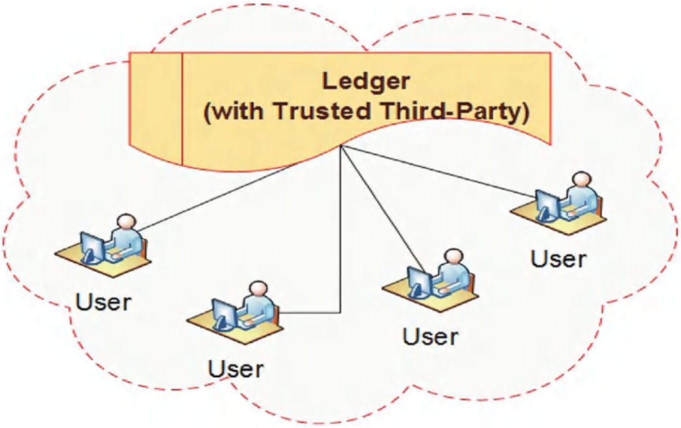


Fig. 4.1 Ledger with trusted third-party

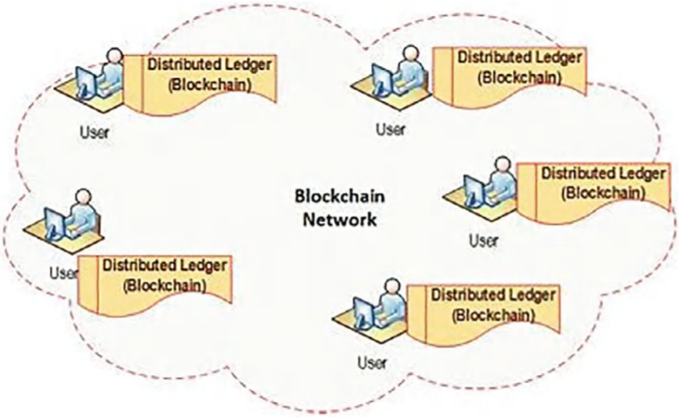


Fig. 4.2 Blockchain network

stamped papers by the use of a blockchain which can be highly secured through the use of cryptographic means. On this idea, the framework of the blockchain was set and built into what it is today. However, the groundwork of the architecture presenting it was not established until 2008. This technology underwent a major revolution when a person or a team of people on the Internet handle Satoshi Nakamoto created bitcoin, which is a digital currency that employs the blockchain concept. With this important development, the design of the new generation blockchain which began as a theoretical idea has moved to the position of practicality that is redesigning electronic commerce. Understanding prior to that is beneficial and leads to a clear understanding of the nature of blocks and their functioning and the described specifics of block structures (Nakamoto 2008).

4.2 Blockchain Architecture

Blockchain architecture has been described as the parts and characteristics that make up any Blockchain system. These are composed of several parts, all of which together comprise the superior sovereign system along with nodes, chains, blocks, miners, and other levels. Briefly, the architecture of a blockchain makes use of what is known as Distributed Ledger Technology (Strategists 2024). This system can have many nodes where data can be stored such that it provides more security and transparency. The information is stored in the connection of blocks that are called chains, and every block contains the details of the transaction, the time it reflects, and the hash of the previous block.

Blockchain simplistically and extensively describes everything that constitutes the creation of Blockchain technology. It is considered the ground formation of blockchain technology. For example, in the newly emerged server-client relationship, the design of the WWW presupposes the concentration of all the information in the server. However, in the case of blockchain, there is no central body that maintains, updates, and approves new entries in the blockchain. In other words, the blockchain structure provides the certainty and reliability of every document and data. However, the participants, even if they do not trust each other, can always end up making a deal/ arrive at a decision.

The core components of the blockchain technology are:

- **The Block**

This data structure referred to as a block contains all the transactions that are distributed to all the nodes in the blockchain network. A block has two hashes (a hexadecimal number which serves as a unique identifier to the block.), the hash of the previous block and its own hash, along with some contents such as (as shown in Fig. 4.3): A block has two hashes (a hexadecimal number which serves as a unique identifier to the block.), the hash of the previous block and its own hash, along with some contents such as (as shown in Fig. 4.3): Transaction Specifics: Data on each and every transaction that is supposed to happen. Time Stamp: Also, each block carries a timestamp which determines when the precise data was put into the block and saved. This is typical for supply chain or transactional data where it is essential to determine the specific time when a specific payment or delivery was made. Nonce: In cryptography, an abbreviation for a number used only once; nonce is a random number used to differentiate a block. For the sending of the above-mentioned data, there is the use of the hashing algorithm which includes nonce, transaction detail, and the former hash. This results in the known output referred to as the "hash address" with an overall length of 64 characters [256 bits]. It is unique to a block and is based on the idea that errors made in one block may result in good outcomes in the next block, leading to the block being profitable in the end. Let us revise, hashing makes it easier for one to spot changes that may have occurred in any block. Hash is the central factor that determines the security of the overall design of any blockchain architecture.

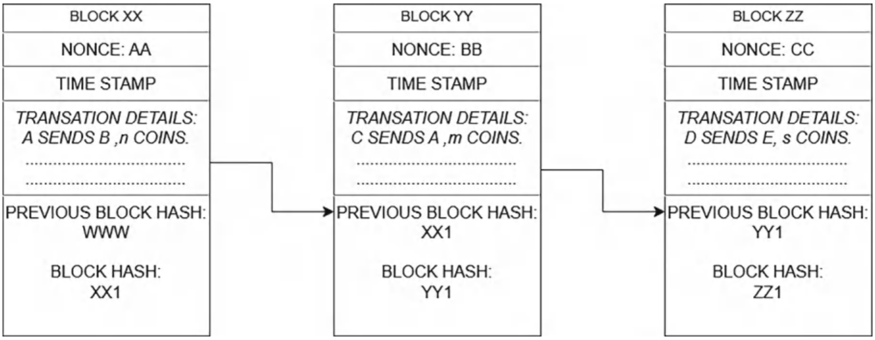


Fig. 4.3 Structure of a block

- **Node**

A node is a crucial component of the network of a blockchain system of architecture. It is either any device that is an element of the blockchain network with the most common being a computer. Nodes are important for the blockchain business and its maintenance since they operate in various capacities. It is essential in the maintenance of the blockchain mainly because they have several uses. In a blockchain structure, a node is a junction that may or may not retrieve and/or deliver information within the chain. They are involved in communication processes where every node intends to synchronize and distribute data in the blockchain. By sharing the list of transactions stored in the blocks among all users, the use of blockchain in a network is efficient and secure because the network is not controlled by a single authority. It can interact with other devices on the network because it already has all the core programs in-house and is connected to the Internet. Nodes also engage in data relaying where it involves other nodes on the same network, ledger storage as well as the updating of the shared ledger.

- **Chain**

This is a series of blocks linked to the current block going up to the first block of all times / also known as the genesis block. The whole intellectual chain is then made safe by using different metadata that links every block to every other block. Actually, the term ‘blockchain’ is derived from the operational concept of the blocks being chained together.

- **Transaction**

Transaction is the smallest unit of the blockchain system that also encompasses data and records among others; they form the blockchain’s shared ledger. To explicate using an example when during a transaction bitcoin could be transferred from one address to another. A simple definition of blockchain can be given as the process and system that aims at the successful and safe completion of a transaction. A transaction formed needs to be validated and digitally signed before it can be incorporated into a blockchain. Also, the nodes of the system substantiate it.

- **Miners/Stakers**

There are two significant things associated with the term "Miners". First, it represents a machine or a node that performs services to the network such as data transfers, transaction authorizations, block validation, as well as blockchain storage. Second, it signifies the person who buys, sets up, and sustains a node for the purpose of receiving the protocol's offered reward.

In the context of blockchain technology the term "Mining" is used to depict the procedure of incorporating new transactional data into the current public/digital record. When a block transaction is hashed, it is nearly impossible to falsify it; thus, mining acts as a security mechanism for the entire Blockchain without requiring centralization.

- **Consensus Protocol**

They are protocols by which the computers in the peer-to-peer network of the node communicate and, more importantly, reach a consensus regarding the process of smart contracts or how transactions are to be affected and documented in the network. All the computers that form a part of the network must download and execute primary software that supports and enforces the particular protocol.

4.2.1 Working of a Blockchain

Essentially, blockchains allow users to enter data-related agreements with people unknown to him/her through the web. Blocks tend to offer settings in which data is made accessible for *storing in a platform that will not belong to anyone, can be populated by anyone, and can never be edited by anyone*. Unlike in the other networks, everybody in the network has the responsibility of overseeing everything aching towards the network. I just wanted to emphasize that the properties I have described in this paper are basic and they fit under the 'technical' definitions such as *distributed ledger, peer-to-peer network, and cryptographically hashed.*

Blocks are simply a list, and to start with, a blockchain could contain no data whatsoever. After that, the creators will create the first chunk of the chain known as the Genesis or the Root Block. This block, instead of referring to anything like what is observed in the other blocks, does not refer to anything at all. In the course of time, the users are allowed to append the information to that list; the format of the given data depends on the function of the given blockchain for instance, if it is a bitcoin blockchain, then you will get several of the transactions. But in all probability, it will look slightly different if the blockchain is envisaged to be used in the supply chain. The way how blockchain actually looks is a collection of receipts arranged into boxes and connected with ribbons. The receipts gathered starting from the time the preceding box was placed up to the chain are accumulated in a new box that is incorporated into the chain. The receipts are transactions in this analogy while the boxes are analogous to blocks.

Every operation within the blockchain structure is a virtual operation, often a monetary or information exchange action. They are coordinated by a network of computers known as nodes from the time they are created to the moment when they are included in the blockchain. These nodes operate with specific software that enables them to interact with each other as well as perform transactional activities on a real-time basis. Each time that a transaction or a change is made by users of the system, the messages that include information about the change are sent to the entire network; where all nodes are always focused. For instance, following the example of cryptocurrency, particularly the bitcoin, if A desires to send B a bitcoin, A would broadcast out the bitcoin to B. The nodes perform several actions on the received messages or convert certain formats into others. Particularly, they ensure the message was signed by the user who broadcasted it, and the result is a message that can be indisputably attributed to a specific person and hard to forge (unlike in the case of authentic handwriting). This ensures that those people who are fake cannot come in and say that the messages were sent by them.

The nodes also play a critical role in confirming it. This procedure is critical in the design structure of blockchain due to its provision of authenticity and consistency in every transaction. This validation is often done using the mining model on the blockchain structure, especially on public blockchains as in bitcoin's case. To enable a transaction to be incorporated into blocks in the blockchain, the miners have to solve immensely complicated mathematical problems. On the completion of the validation process and incorporation into a block in the format of a transaction, a transaction becomes an unchangeable feature of the blockchain. It will also have a valid date and time stamp and a unique communicating identification number to further prevent the altering of the record. Then, it will be connected to the preceding block and so on Thus, the structure of the block will be connected successively from one block to another. After that, this block will be connected to another block, and thus on end to end of the structure. That way it forms a chain of blocks, this term is where the word blockchain originated from, whether they are public or private, you must use Application Programming Interfaces (APIs) to make a connection to the blockchain database (Fig. 4.4).

4.2.2 Layered Architecture of the Blockchain Ecosystem

Blockchain architecture goes wider the software and networks and enters the hardware domain and structure A: Blockchain can have different structures based on the kind of blockchain in function; there is no right way to organize blockchain structure. Still, the given configuration should be generally inherent in most of the blockchains. The current section is connected with the discussion of the five most typical layers in the systems of the Blockchain type and these are the Protocol layer, the Network, the Consensus layer, the Data layer, and the Application layer. It is very important for

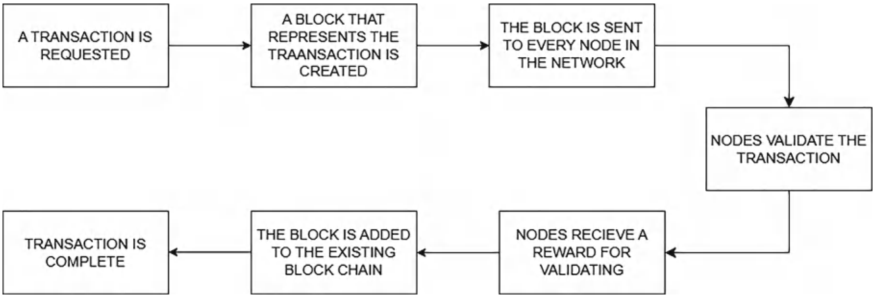

A TRANSACTION IS REQUESTED
A BLOCK THAT REPRESENTS THE TRAANSACTION IS CREATED
THE BLOCK IS SENT TO EVERY NODE IN THE NETWORK
NODES VALIDATE THE TRANSACTION
NODES RECIEVE A REWARD FOR VALIDATING
THE BLOCK IS ADDED TO THE EXISTING BLOCK CHAIN
TRANSACTION IS COMPLETE

Fig. 4.4 Working of a blockchain

the evaluation of blockchains, the detection of specific applications, and the attempt at establishing interaction between blocks in this hierarchy.

- **Hardware/Infrastructure Layer:**

The final of the four layers of the blockchain structure is finally known as the infrastructure layer. The former works in conjunction with the latter for the purpose of dedicating the features necessary to run the nodes and for them to be able to transmit information. Its purpose is to present in order the necessary capacities to host a blockchain (Strategists 2024). This layer cutout offers the fundamental layout of the blockchain and includes everything that wants to execute any kind of transaction, any cryptographic computation, etc. It has mining machinery, computers, servers, etc. Thus, Smart contracts VMs are another component of the infrastructure layer that includes smart contracts and do similar operations to Operating systems.

- **Data Layer**

The function of this layer is to ensure secure and confident messaging transfer (Chiat 2024). The two primary tasks of the data layer in blockchain technology are the management and storage of data. This layer stores the ledger data and the history of the transactions for the chain in a structural and impenetrable manner. Blocks are understood as the specific data organization, which is used. Joining each block with the next by a cryptographic is known as a blockchain, so evidently blockchain is all about. This chain gives an irreversible and secure method of keeping a record of multiple transactions. Despite the fact that the details of hashing as well as the process of transaction addition do not seem to be of utmost importance as they do not seem to be painted out boldly or are hidden beneath some other topics of major significance, one must then pause to reflect on the significance of the layer that has been recognized as the data layer and the functions it serves in the shell of particular importance to the achievement of privacy and integrity. Almost all of the cryptographic tools including the public and private keys, the cryptography libraries, and even the elliptic curve digital signing method are set up in this layer and used in protocols-signatures. For security, a digital signature—that will be a mixture of a set of operations performed on the data and a cryptographic primitive; to prove that only a specific party has the right private key without explaining it to others—is given to each transaction. The creation of digital signatures is more complicated and needs methods like the Rivest-Shamir-Adleman or popular as RSA and the Elliptic Curve Digital Signature Algorithm; or shortly the ECDSA (Hacken 2024).

- **Network Layer**

As it was mentioned, Layer 1 also known as the network layer is to create a peer-to-peer network that connects the blockchain nodes. Whereas to make sure that more than one node approves the transaction the layer makes sure that the exact transaction is fan out across the network.

- **Consensus/Protocol Layer**

This layer consults all its nodes to agree on the validity of transactions. Also referred to as the layer of protocol, it spells out the guidelines for the participation of players in the blockchain, the most crucial of which is the consensus mechanism (Strategists 2024).

For identifying how transactions were included in the blockchain, the consensus layers are relevant. There are numerous consensus methods that can be employed, such as: As to the question of what consensus methods can be used, it is possible to list the following:

- Proof of Work (PoW): New blocks and transactions are verified by the nodes and the only way they can do it is by solving mathematical problems. (used by Bitcoin).
- Proof of Stake (PoS): In this process of validating transactions and building new blocks, nodes are selected based on the amount of stake, they own on the Internet. (used by Ethereum).
- Delegated Proof of Stake (DPoS): This method is used by the community participants to select nodes to approve transactions and provide construction of new blocks.
- Byzantine Fault Tolerance (BFT): This way, a consensus process that includes malfunctioning nodes is employed to arrive at a general consensus.

The presented basic consensus mechanisms are explained in Sect. 4.3.

- **Application Layer**

The first part of the application layer is the smart contract or, in other words, programmable code that defines state transitions. It has different names in other ecosystems, it is known as 'programs' in Solana and 'chain code' in Hyper ledger, and can work as escrow, payment channel, or vault. Hacking is common among smart contracts due to the fact that any simple deficit in the code can be utilized to gain unauthorized traction.

Candidates do not directly interact with the smart contract much of the time. They rely on an API or the front end of a Web3 application. An example of a decentralized application that incorporates both the UI and the smart contracts is the Uniswap website (Fig. 4.5).

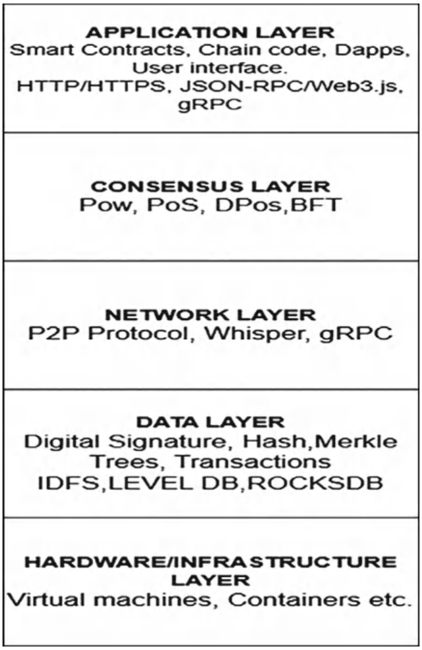

APPLICATION LAYER
Smart Contracts, Chain code, Dapps, User interface.
HTTP/HTTPS, JSON-RPC/Web3.js, gRPC
CONSENSUS LAYER
Pow, PoS, DPos,BFT
NETWORK LAYER
P2P Protocol, Whisper, gRPC
DATA LAYER
Digital Signature, Hash,Merkle Trees, Transactions
IDFS,LEVEL DB,ROCKSDB
HARDWARE/INFRASTRUCTURE LAYER
Virtual machines, Containers etc.

Fig. 4.5 Layered architecture of blockchain

4.2.3 Types of Blockchain Architecture

- **Public Blockchain**

It means that anyone can become a node and join the P2P network within the structure of the public blockchain without anyone's approval. Besides, many of the new members of the network still maintain the use of their anonymous IDs. The architecture of the public blockchain, however, offers incentives such as opportunity costs and mining or staking rewards to ensure the stakeholders do not cheat. Public Blockchain architectures are largely decentralized and hence they feature security as their main strength. The disadvantage is that updates experience adversarial governance procedures regularly and are sometimes rigid.

- **Private Blockchain**

In this case, therefore, all the nodes that are realized in the P2P network are owned by a single person, company, or organization. The only person or entity that may permit someone else to be added to the network in such a scenario is the owner of the blockchain. Another advantage is that being a private data structure, a private blockchain can be modified in every way by its owner. Also, unlike in other environments with many people to persuade, improvements are easy to carry out. Private blockchains on the other hand are the least secure and are easily susceptible to censorship. This is due to the disadvantage where there is a likelihood of the central authority being dethroned. They are also not rigid because the owner can just change the regulations.

- **Hybrid Blockchain**

There are at present, two tiers forming the architecture of this blockchain. The basic structure is a P2P network accessible by the public. An entity creates its own blockchain, which occupies the second level in this regard. The high level of decentralization that is secure to the public blockchain network also benefits the private blockchain. At the same time, it becomes possible for a company or another organization where the use of blockchain is necessary to initially develop a blockchain for its specific needs, as well as change its attributes if necessary.

- **Consortium Blockchain**

A consortium blockchain architecture is a system realized by the cooperation of a large number of participants. All of them can own stock in the firm overseeing its operations, occupy a node on the network, and contribute to decision-making. The participants, in the consortium blockchain designs, often have to provide their identity. Also, applicants for the membership of the consortium and network have some set of conditions that need to be met, for instance, the payment for the services (Fig. 4.6).

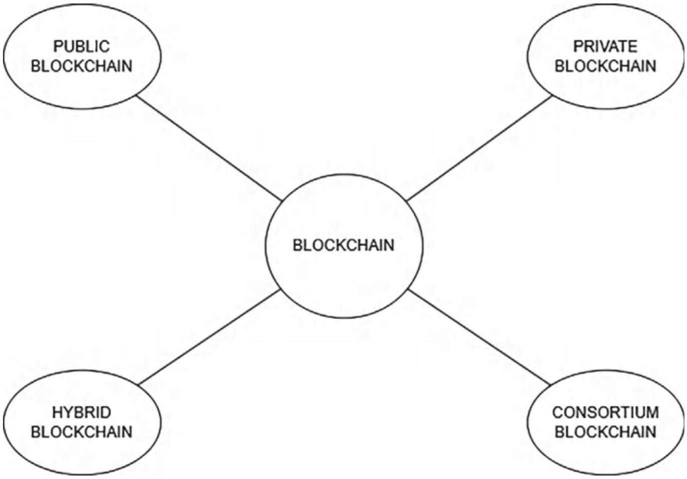


Fig. 4.6 Types of Blockchain architecture

4.2.4 Components of Blockchain Ecosystem

An ecosystem of using the blockchain can be described as the framework that is used by the majority in a certain application in terms of protocol and governance. In terms of protocol structure, there is a concept of the set of principles defining the behavior of the participants, ownership of data, funding schemes, entrance and exit conditions, and the conditions according to which data is shared between the participants (Blockchains 2024). The blockchain protocol is another vital aspect of every blockchain system as it entails the rules and frameworks to be followed in the running of the network. Some of the well-known protocols are Ethereum, bitcoin, and others. The various necessities within these ecosystems work in concert to increase the effectiveness and utility of these ecosystems.

- **Distributed Ledger Technology (DLT)**

The main branch of technology in blockchain is known as distributed ledger technology or DLT. It is digital data that has actively been copied and shared and is synchronized across multiple sites or nodes but it does not have a single data center and it also lacks management tools.

Offline databases stored separately on centralized servers also have a different concept after the invention of DLT. It prescribes an environment where all the users of the given network are able to view all the transactions registered in the ledger and are also able to engage in the consensus and validation method in order to establish new transactions.

Because of such a structure, prime focal points are not consolidated, which further helps in increasing the dependability of the system of faults. Moreover, it forces members of the network to act and agree on something, which in turn strengthens beliefs in reconciliations of the ledger.

- **Smart Contracts**

These are largely computerized legal agreements that can actualize all the provisions pertaining to the contractual terms of this code language. They might assist in the negotiation or affirmation or even fortification of a deal written in a contract after they are activated as communicated in the document. Smart contracts are the tools for executing reliable transactions without an intermediary; tendencies are observed that make transactions transparent, especially in terms of conflicts when shareholders exchange assets for money or shares and so on. If properly applied they could revolutionarily lessen the troubles, expense, and superfluous of conventional contract law.

- **Consensus Mechanisms**

Sophisticated and diverse structures of blockchains exist and more to that, each of the structures uses a different consensus. There are many consensus mechanisms available but some of the most famous are Delegated Proof of Stake (DPoS), Proof of Work (PoW), and Proof of Stake (PoS). Some focus on making it very secure and 'not centralized', while others care more about how fast it can go and how big it can expand. Both have their advantages and disadvantages.

The decision-making mechanism varying blockchain from being an efficient solution to a centralized application is the consensus algorithm. In relation to centralized applications, all the users depend on the central power to execute transactions in a trustworthy way. But it means that in the absence of a supervisor or a leader, the network's nodes bear responsibility for this. Allocation methods such as PoW and PoS mentioned above are used by blockchains to reach consensus in an unreliable environment. In a move to arrive at the consensus, the consensus algorithm transmits the miner's transaction to the validator nodes. Depending on the result of consensus, the transaction is recorded in the ledger or removed from it (Talks n.d.).

- **Asset**

Anything that is valued by the nodes of the network, as well as those that are indifferent to their physical and non-physical characteristics, can be considered assets. Here are a few instances of assets.

- Code blocks: Smart contract—Based on code blocks are a kind of pattern used by some of blockchains similar to Ethereum to store code in the type of smart contract which lays a fitting for the concept of decentralization applications.
- Financial transactions: Dogecoin or bitcoin, for instance, operates on blockchain technology, and their transactions are recorded as a ledger.
- Business transactions: To ensure that the product is good and that some conditions are met, businesses can use private blockchains in trading with other businesses. This enhances security and ensures accountability is met as required in the course of organizational operations.

  Medical records: As a result of the instability in the environment, medical records of patients are usually stored in private blockchains to retain accuracy.

  - **Cryptography**

Cryptographic engineering is employed in the construction of blockchain due to the fact that it facilitates the management of the generation of coins besides offering safety to the transactions. It ensures that all the nodes in the network have the same status of the distributed ledger by linking the contents of each new block to every other block of the distributed ledger that had been created before. Public-key cryptography provides a way to authenticate access rights and to control the access to the objects, while cryptographic hashes ensure the objects' integrity and their legitimate state. When summing up, it is possible to state that these cryptographic methods provide blockchain technology rather than a solid foundation in terms of security.

Applications of blockchain are not limited to bitcoins that are often associated with the technology. It plays a role in several fields because the promise technology holds for reducing costs and increasing the efficiency of companies, as well as improving trust and equality among them, cannot be ignored.

Almost every industry is using blockchain technology, including:

- The cryptocurrency
- Medical care

- Banking and finance
- Real estate
- Shop
- Transportation and distribution
- Governance and voting
- The Internet of Things
- Public relations and advertising

4.2.5 Use Cases of Blockchain

- **Blockchain in Money Transfer**

The applications for the transfer of cryptocurrencies—initially stimulated by Bitcoin—have gained immense popularity. Blockchain is widely popular in the banking sector since it can benefit financial organizations of various sizes and save them money and time. Third-party fees can be cut to a very minimum, papers can be avoided and ledgers can be made in real time through the use of blockchain. They can be most valuable to large banks as the following lists the benefits of LBP. Many of these companies conduct efficient transfers of funds through the use of blockchain. Blockchain technology can accommodate the management of bank payments effectively and at a low cost, facilitate the identification of possible money laundering processes, and produce alternatives to the regular means of creditworthiness assessment.

The future potential of blockchain's application is gradually being acknowledged within the sphere of banking. Blockchain is decentralized such as through removal of control by institutions like banks and facilitates trade of currencies and secures them. Taking of security for loans, the handling of payments, and other facilities which make it a worthy contender and a close substitute to current industrial applications. In fact, it is possible to state that blockchain can solve several of the most challenging problems for the banking industry. It was initially used as the secure distributed ledger for Bitcoin and was known for enhancing security and operations speeds, especially in finance.

- **Blockchain Applications in Health Care**

Blockchain technology can be beneficial for healthcare providers to improve patient outcomes, maintain data accuracy, and update the delivery of services. It is impossible to attract shady characters to obscure Maltese addresses and hack through the outer layer of encryption, as has been the problem with previous years and older systems. Besides this, it also allows the secure exchange of data in real time which eliminates many sources of big administrative waste.

- **Secure Electronic Health Records (EHRs)**

It has been seen that when the patient records are split, the medical practitioners are locked out from the whole patient record. This problem is addressed by a blockchain integrated with the current EHR software. The decentralized system provides the consolidation of a patient's record and safely stores and preserves EHRs. Also, it provides the capability to choose who has access to EHRs, management of amendment to EHRs, and control over the sharing of EHRs among healthcare givers.

- **Management of the Pharmaceutical Supply Chain**

Ensuring the legitimacy of drugs and medications is among the tasks that the healthcare industry has to solve continuously. In such a way, medical goods can be fully viewed and their path can be tracked through the means of blockchain, where all stages of the supply chain are visible. Blockchain implementation is one of the safest and the most legitimate approaches to addressing the lives lost due to fakes in this industry that records thousands of deaths annually.

- **Blockchain in Money Transfer**

For example, the financial service industry realized that they had to solve the Overseas Payments problem. Because of countless intermediaries, traditional cross-border payments are relatively expensive and several days long.

Solution using blockchain approach: Cross-border payment processing for purchase and travel that amounts to online, real-time, fee-less money transfer possible with blockchain technology. For instance, Ripple (To facilitate the foreign as a payment processing network, ripple uses blockchain technology for electronification of money transfers. Based on its native coin XRP, consumers may make cross-border payments in several fiat and currencies with on-demand liquidity. It allows for the carrying out of international transactions within the blink of an eye by using the ($) symbol; XRP Ledger. Correspondence is reduced and hence the costs are slashed because the bank does not need to serve many outside institutions.

Working: The basic form of a transaction is an agreed amount of currency local to the sender. The original amount of money is converted to the sender's local currency in XRP to the blockchain network that in turn relays it to the recipient's country and converts it back into the local monetary unit of the recipient. The transaction is completed within one clear and readily identifiable ava endpoint that is a distinct culmination of the transaction or process as a whole that has been described by the web service of seconds, and once this information is inputted into the blocks, it cannot be removed or changed.

4.3 Consensus Mechanisms

Consensus mechanisms, in blockchain, are a bunch of protocols or algorithms that allow all the nodes present in a distributed blockchain network to reach an agreement among each other on the validity of transactions or about the ledger's state. It seals the nature of integrity and security of distributed ledger by preventing activity such as double-spend and by maintaining network harmony throughout the systems. Consensus mechanisms are considered to be the essential foundations of blockchain technology since they offer the primary framework required to make decisions in a peer-to-peer system. As with any consensus mechanism, they are all varied and are adopted based on various types of applications and network demands Biswas (n.d.).

4.3.1 Structure of Consensus Mechanisms

The diagram encapsulates the following (Fig. 4.7).

- Nodes (Participants): The parties or units of a network that are involved in the consensus-building process.
- Transaction Pool: A number of transactions that are to be incorporated as a block of transactions at a given time.
- Block Proposals: Nodes send blocks of transactions into the network with a view of having them incorporated into the blockchain.
- Consensus Process: It refers to the way that nodes establish an agreement with each other on the validity and chronological sequence of the transactions' entries and includes Proof of Work (PoW), Proof of Stake (PoS), Delegated Proof of Stake (DPoS), Practical Byzantine Fault Tolerance (PBFT), Proof of Authority (PoA).
- Block Validation: Network approval occurs as a result of the proposed block.
- Blockchain: The validated block is appended onto the chain hence making the record set rigid.

4.3.2 Types of Consensus Mechanisms

- **Proof of Work (PoW)**: Proof of Work consensus mechanism is famous because. In PoW, miners work hard to solve complicated algorithms which are mathematically oriented. The former is to place a block on the chain, and the latter, along with it, earns cryptocurrency as a reward. The main advantage of PoW is great security but on the other hand, it consumes a lot of energy and needs great computational resources.

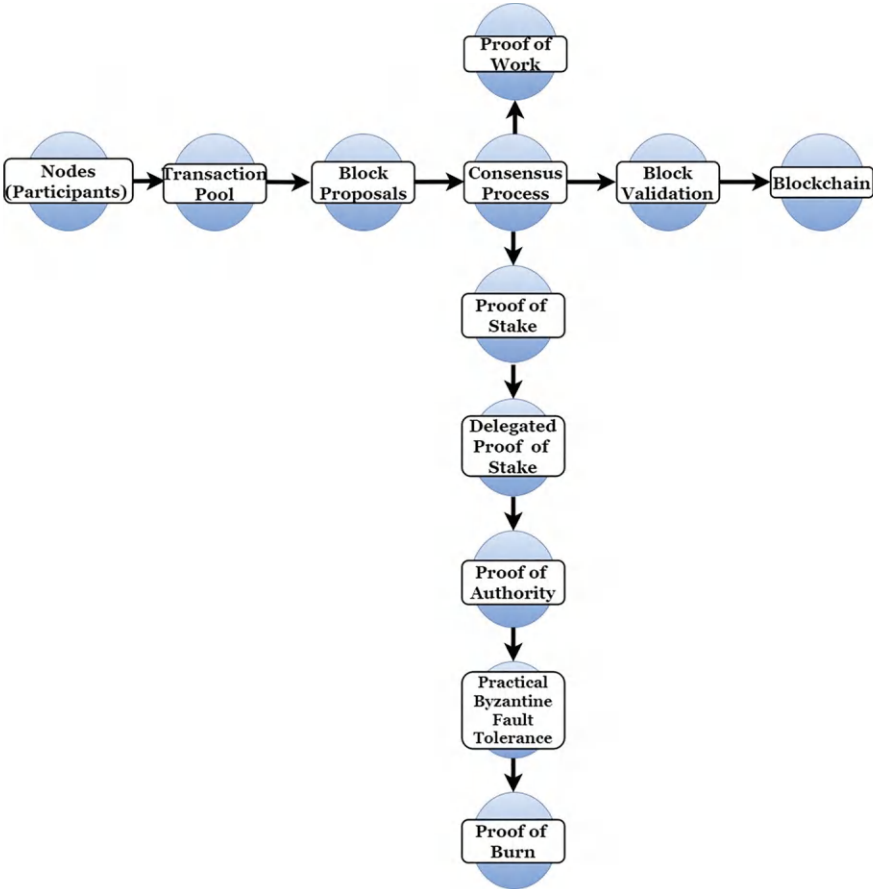


Fig. 4.7 Structure of consensus mechanism

- **Proof of Stake (PoS)**: Another consensus algorithm is Proof of Stake which unlike PoW is more friendly to the environment. In validators' cases, the roles are assigned with reference to the number of tokens and the desire to stake them for receiving proof obligations. Tokens staked on the staking solution are locked for such a duration; thus, the longer the period, the more tokens, the higher the chances of being selected for the validation of transactions. Specifically, PoS requires much less energy in comparison to PoW, but it can successfully protect the network from possible threats.
- **Delegated Proof of Stake (DPoS)**: DPoS for that reason advances PoS any more taking it to becoming a voting system to be precise. Owners contribute a few tokens which are a few representatives who do the actual validation of transactions and

running of the blockchain. This method is preferable and less difficult for the network because the nodes needed for the consensus are reduced here.

- **Proof of Authority (PoA)**: Special consideration is given to the Proof of Authority, validators are a few selected individuals who are vested with the responsibility of validating the transaction and generation of blocks. These are valued and validated, and as such, are already familiar with the network hence, makes PoA suitable for a private or a consortium blockchain environment whereby all individuals are well known. It is very efficient and helps to develop more swifter or faster transaction movement.
- **Practical Byzantine Fault Tolerance (PBFT)**: The full meaning of the abbreviation PBFT is Practical Byzantine Fault Tolerance and the success of this approach is tied to the nodes that may fail or act dishonestly. PBFT is a primary node that informs the other nodes that it has created a block and wants them to approve it. For the block to be added to the blockchain, a majority of nodes or peers (which are normally two or three) have to approve the block. PBFT works well for permissioned blockchain systems with a smaller number of nodes for making the fault tolerance high.
- **Proof of Burn (PoB)**: Proof of Burn is another consensus model that is rather distinctive; participants have to make their coins 'burn', that is, effectively destroy them by transferring them to an address that can no longer be spent. In return, they get the right to mine new blocks in the proportion of coins that they burnt. The concept of PoB is that it integrates features of PoW and PoS, as it guarantees security due to economic loss without necessitating physical GPUs/CPUs Patel. (n.d.).

4.4 Smart Contracts

Smart contracts are indeed the normal parties' initial agreements that are programmed in the form of automated, self-executing, and self-enforcing computer codes. Unlike traditional contracts on servitude, which necessitate the input of lawyers, brokers, or other intermediaries to enforce, smart contracts operate on blockchain, therefore making the process open, secure, and unlinked. Indeed, smart contracts can be considered a great step in the sphere of digital deals, and digital deals have a lot of advantages in front of traditional contracts. These are based on open blockchain platforms; of which, the most popular is Ethereum, which is an open-source decentralized platform for building smart contracts and decentralized applications IBM (n.d.).

Here's a step-wise guide on how they work:

- Coding the Contract: The provision of the agreement is then codified. This code contains provisions in terms of when the contract will be implemented, for instance, the passing of money for a good or a service to be delivered.

- Deploying to the Blockchain: The smart contract is then deployed on the blockchain once the code has been written down. This makes it non-alterable and shared among all nodes of the network, thus, transparent.
- Triggering Execution: Thus, when the predefined conditions are met the next actions are done according to the smart contract decisions.
- Verification and Validation: The implementation of the contract is confirmed and checked by the nodes of the network. This decentralized verification makes certain that all members of the contract perform their duties as depicted in the contract.

4.4.1 Characteristics of Smart Contracts

- **Self-Executing Nature**: The other feature that has been attributed to smart contracts is that they are self-executing in execution. Smart contracts are different from conventional legal contracts that require third-party enforcement by executing the terms and conditions of the agreements. This form of automated execution removes the factor of reliance on the third party and it can guarantee that everyone honors his/her contractual obligations with objective and without reference to prejudice.
- **Immutable and Tamper-Proof**: Smart contracts are built on the blockchain as they offer the decentralized and unalterable record of the transaction. Smart contracts cannot be changed once deployed because those influences that are recorded in the blockchain are transparent and fixed. This guarantee forecloses the changes in the contractual terms and conditions to enhance the authenticity of the contracts and the reliability of the parties to the contracts.
- **Decentralization and Transparency**: Decentralization is a major characteristic of smart contracts because the contracts run on blockchain networks that are spread across many nodes. This seems to decentralize the execution and control of the contract making it difficult for any single entity to control or censor the process. Moreover, since all participants in the contract can see the terms of such an agreement and the functioning of smart contracts, its operation increases impartiality.
- **Conditional Logic and Flexibility**: Smart contracts are very flexible and in case one wants to change part of a contract, it is very easy to do so since smart contracts include programming and invoke conditional statements. The legal nature of contractual relations is relatively very open, which implies that any situation and condition is régulable in the contracts. This versatility enables smart contracts to introduce several uses across different industries, including finance and supply, the contracts might have intricate relations, and integration, within the fulfillment.
- **Cost-Efficiency and Time Savings**: In its application, smart contracts save time and therefore reduce the cost, as there are no middlemen in the conclusion of the contract and contract execution. Smart contracts reduce the fees and time involved in the manual and centralized processing of transactions and thus are

more efficient in carrying out the actual financial transactions. Due to their cost-efficient nature, smart contracts are appealing to organizations that aim to cut their expenses and improve efficiency.

4.4.2 Benefits of Smart Contracts

- Automation: It removes the need for middlemen or third parties since smart contracts do the execution of the contracts automatically. This minimizes the possibility of people making mistakes and also enhances the rate of transactions IBM (n.d.).
- Security: Smart contracts are saved on blockchain technology, which entails that they are well-secured. The key feature of blockchain is that once a contract has been written it cannot be changed, this means that the contract is safeguarded against attempts to modify or sever it.
- Transparency: Smart contracts have been reported to be transparent in the blockchain network whereby all the participants can see the terms and the implementation of the smart contracts.
- Cost-Effectiveness: In this manner, smart contracts pose the potential of drastically cutting the transaction costs derived from the involvement of so many intermediaries and multiple processes. This makes them to be wanted among businesses that are interested in enhancing their operations.

4.4.3 Applications of Smart Contracts

Smart contracts have a wide range of applications across various industries.

- Finance: Several procedures can also be carried out through smart contacts; these include loaning, claiming insurance products, and trade financing. This also helps in cutting down on the time needed to perform a certain task and improves reception.
- Supply Chain Management: Smart contracts automatically record the movement of goods, thus, eradicating the cases of fraud in the supply chain. For example, in supply chain management a smart contract can call for payment upon delivery verification.
- Real Estate: Real estate is perhaps one of the most intricate disciplines of business, which involves the transfer of property or assets from one owner to another, and may contain numerous participants besides a great amount of paperwork. Smart contracts can solve the buying and selling of property issues by automating the exchange of property titles and funds.
- Healthcare: Towards the patient's record, smart contracts can work with patient information to promote data security and protect the data while only giving authorized teams access to the file.

- Legal Industry: Clearly smart contracts will benefit all sorts of contracts from a simple non-disclosure agreement to a large-scale multi-contract agreement since they guarantee that the next action will be for the agreed terms to be fulfilled.

4.4.4 Inter-Relationship Between Characteristics, Benefits, and Applications of Smart Contracts

Interrelationships between these elements are considered to be basic to the understanding of how smart contracts improve different spheres of the economy and open up new horizons for decentralized approaches. Smart contracts' features of integration of automated actions, open accession, inalterability, decentralization, self-implementation, conditional collaboration, cryptographic safeguard, and programmability afford smart contracts numerous advantages including approachability, credibility, safety, fraudulence diminution, economical, reliability, preciseness, data assurance, versatility, and personalization. Smart contracts benefit from these in multiple diverse uses in areas of financial services, supply chains, health records, property records, electronic voting, insurance, patents, identification/authentication, games, trading cards, and collectibles. Thus, the decentralization of smart contracts' inherent potential defines their global applicability and feasibility for implementation across industries (Fig. 4.8).

Here is a table summarizing the relation between the smart contract characteristics, benefits, and application domain:

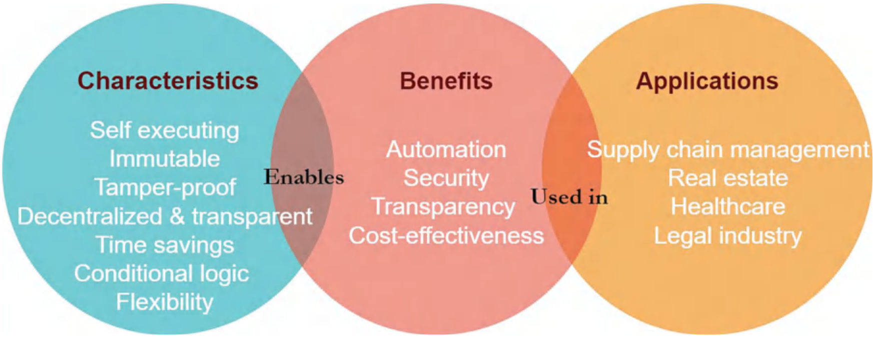


Fig. 4.8 Venn diagram of inter-relationship between characteristics, benefits and applications of smart contracts

4.4.5 Real-Life Examples of Smart Contracts

In a practical application, let us consider when a landlord has rented a property to his tenant. Earlier they would set up a legal contract that involved elements such as the amount of rent to be paid monthly, payment deadlines, and penalties in case the payment was made after the said date and both parties would sign it. Next, let's introduce a smart contract into this process. These outline the parties, parties' addresses, rent amount (Ethereum Currency (ETH) defined in the smart contract), due date, and penalties to the tenant. The smart contract implemented by the wallet can check whether the rent in Ether is paid in full before the due date every time the tenant's wallet makes the transfer for the monthly rent. If so, it quickly spends the payment on the landlord as soon as possible as the landlord looks forward to its speedy disbursement. However, if the payment is made later than the agreed time, the smart contract incorporates the specific amount of the late fee to be added to the amount of money required for the next month's rent. This eliminates the need to engage a middleman and makes rent payments to be clear and effective. Naqvi (n.d.).

Here is the real-time working guide for the above example.

1. Coding the contract

A. Contract definitions

See Fig. 4.9

B. Modifier

See Fig. 4.10

C. Pay rent on time

See Fig. 4.11

D. Pay rent with late fee

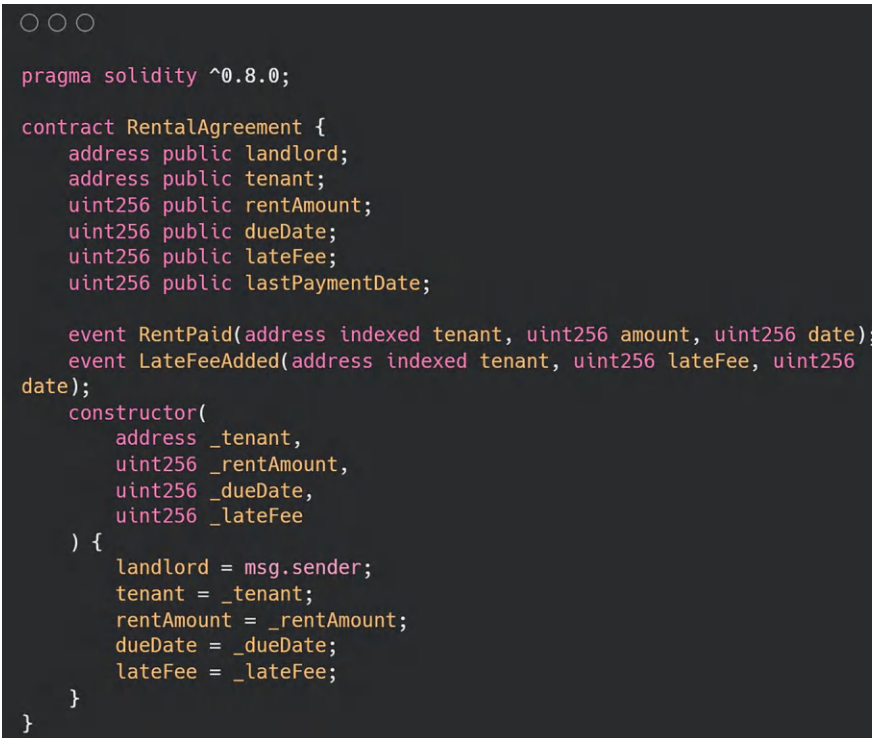
See Fig. 4.12

2. Deploying the contract

To create the contract, go to the 'Deploy & Run Transactions' part. Supply with the tenants address the amount of Ether in wei which is owed on the agreed rent date and time via a timestamp, as well as the ether valued late fee in wei. Complete these portions in the constructor and then click on Deploy.

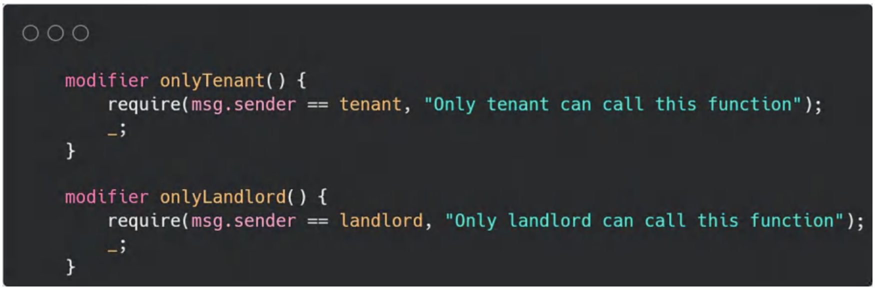
3. Triggering execution

- Paying rent on time: In essence, the person renting is required to call the "payRent" function with the exact amount of rent before the due date.
- Paying rent late: Ensure that you can see both events named "RentPaid" and "LateFeeAdded". This means the payments were affected right.



```
pragma solidity ^0.8.0;

contract RentalAgreement {
    address public landlord;
    address public tenant;
    uint256 public rentAmount;
    uint256 public dueDate;
    uint256 public lateFee;
    uint256 public lastPaymentDate;

    event RentPaid(address indexed tenant, uint256 amount, uint256 date);
    event LateFeeAdded(address indexed tenant, uint256 lateFee, uint256
date);
    constructor(
        address _tenant,
        uint256 _rentAmount,
        uint256 _dueDate,
        uint256 _lateFee
    ) {
        landlord = msg.sender;
        tenant = _tenant;
        rentAmount = _rentAmount;
        dueDate = _dueDate;
        lateFee = _lateFee;
    }
}
```

Fig. 4.9 Defining the contract

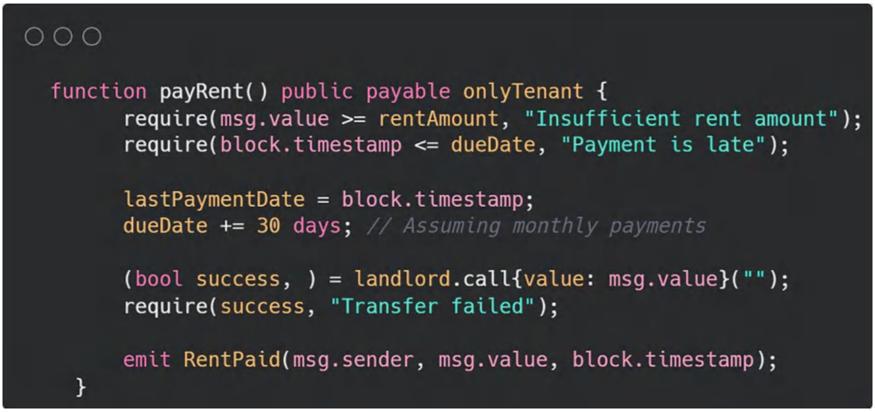


```
    modifier onlyTenant() {
        require(msg.sender == tenant, "Only tenant can call this function");
        _;
    }

    modifier onlyLandlord() {
        require(msg.sender == landlord, "Only landlord can call this function");
        _;
    }
```

Fig. 4.10 Modifiers of smart contract

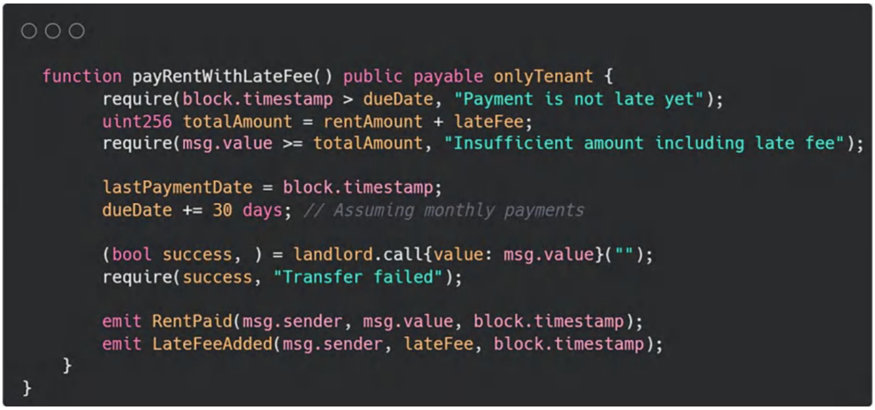
4. Verification and validation

- Watch for the ‘RentPaid’ and ‘LateFeeAdded’ that happens here. This assists you to understand if everything passed through alright.
- Ensure that the total balance for the landlord increases by the amount of rent while the tenant’s balance decreases in accordance with the due date of the rent.



```
    function payRent() public payable onlyTenant {
          require(msg.value >= rentAmount, "Insufficient rent amount");
          require(block.timestamp <= dueDate, "Payment is late");

          lastPaymentDate = block.timestamp;
          dueDate += 30 days; // Assuming monthly payments

          (bool success, ) = landlord.call{value: msg.value}("");
          require(success, "Transfer failed");

          emit RentPaid(msg.sender, msg.value, block.timestamp);
      }
```

Fig. 4.11 Code depicting paying rent on time



```
    function payRentWithLateFee() public payable onlyTenant {
          require(block.timestamp > dueDate, "Payment is not late yet");
          uint256 totalAmount = rentAmount + lateFee;
          require(msg.value >= totalAmount, "Insufficient amount including late fee");

          lastPaymentDate = block.timestamp;
          dueDate += 30 days; // Assuming monthly payments

          (bool success, ) = landlord.call{value: msg.value}("");
          require(success, "Transfer failed");

          emit RentPaid(msg.sender, msg.value, block.timestamp);
          emit LateFeeAdded(msg.sender, lateFee, block.timestamp);
      }
  }
```

Fig. 4.12 Code depicting paying rent with late fee

- Cross-check that there is a due date after which the payment for next month, for example, is expected.
- In general, if you adhere to all of these steps, you are able to deploy the blockchain a Smart Contract code written in Solidity immediately.

4.4.6 Hands-On Experience for Understanding Smart Contracts

Making practical experiences for smart contracts means exposing students to actual activities to experience how smart contracts function, and ways of programming, deploying, and utilizing the smart contracts. Here's what this might entail.

- Learning Basics of Blockchain

This knowledge would encompass the basic understanding of how blockchains work, varieties of blockchains (public, private, consortium), and various concepts as the consensus mechanisms, nodes, and ledger.

- Programming Languages

Learn some programming languages for creating smart contracts such as Solidity for Ethereum, and Vyper. In this age of technology, Solidity language is the most popular and commonly utilized in the formation of smart contracts.

- Development Tools and Environments
  - Truffle Suite: An architecture that was designed for the development of the Ethereum platform, which includes a smart contract development kit including a testing framework and asset pipelining.
  - Remix IDE: An online IDE specifically designed for writing, testing, and deployment of Solidity smart contracts.
  - Ganache: Ethereum local-based environment for testing purposes of smart contracts.

- Writing Smart Contracts

The ultra beginners should start with basic contracts (for example, Hello World contract). And continuations of the simple functions, compound functions, conditions, and complex events governing contracts.

- Deploying Smart Contracts
  - Learn how to create and publish smart contracts to various Ethereum networks (e.g., Ropsten and Rinkeby for use during testing, and Ethereum for the actual use).
  - Develop techniques for handling deployment transactions such as the MetaMask wallet.

- Interacting with Smart Contracts:
  - Use Web3.js or Ethers.js libraries for calling the deployed smart contracts from the web application.
  - Develop users' interfaces of smart contracts by creating applications that let the user send messages to such contracts, e.g., executing a particular transaction or Contract details inquiry where applicable.

- Testing and Debugging
  - Write unit tests for smart contracts using the testing frameworks of Truffle or Hard Hat.
  - Remix and Truffle Debugger should be used for debugging of contracts.

- Security Best Practices
  - Gather information on the typical smart contract weaknesses like re-entering, integer overflow/underflow.
  - Ensure that security measures are good practices that are followed, and use static analyzers such as MythX or Slither.
- Participating in Hackathons

Some of the blockchain hackathons where one can work on actual problems, interact with other developers, but principally get ideas on smart contract projects include;

- Deploying on Testnets

Use the Rinkeby, Kovan, and Ropsten test nets to run your smart contracts in a real-life environment, but without the real Ether.

4.5 Blockchain Development Process

4.5.1 Web APP Architecture

Unlike Web 2.0 applications, Web 3.0 thus removes the centralized database that holds the state of the application and there is no central web server where most of the back-end operations are done. Rather, it is employed to create applications on a state machine that is distributed and managed by unknown people from the Internet. There is no authority that possesses control over this state machine, which is owned by all participants of the global network Qutub (n.d.).

Rather than the highly regulated backend of Web 2.0, in Web 3.0, developers can write smart contracts, which specify actions of the decentralized applications and disseminate them in the decentralized state machine. What this implies is that anyone, who desires to create a blockchain application, uploads their code to this shared state machine. And the front end? It remains pretty much the same, focussing on finding, appraising, and managing assets that would generate income and increase the company's worth.

Here's what the architecture looks like.

- **Front-end**: The front end prescribes the logic of the user interface but also interacts with the application logic included in smart contracts. The front end is able to send data to smart contracts and call functions to communicate with any of the nodes in the distributed Ethereum network. Each node in the Ethereum network has another full record of every state in the Ethereum state machines: raw code and Data of each smart contract. When using the blockchain to convey data, and the code inside the blockchain, you have to convey with one of these nodes because a particular node can request to fire a transaction on EVM. A miner will then sign the transaction and broadcast the change that the transaction made on the rest of the network.

There are two ways to broadcast a new transaction:

- Establish your own node that comprises the Ethereum blockchain application.
- Choose nodes from third-party services such as Infura, Alchemy, and QuickNode.

No need to run a full node reducing the time needed for its setup which can take days; may require more bandwidth and disk storage than a laptop sociably provides. Additionally, as a DApp grows, the number of megabytes needed to store the entire Ethereum blockchain gets larger and as such, consumes more resources requiring more nodes and possibly, full-time DevOps specialists for management.

- **Wallet integration**: To avoid such complications, most of the DApps rely on third parties such as Infura or Alchemy for node services. While this creates a new point of failure, it improves node management because there is only one to manage. These nodes, when self-provided or deployed by third parties, are commonly called "providers." Thus, when associating with a provider, one can only read the state stored on the blockchain. However, to write to the state, transactions must be "signed" using a private key more commonly referred to as Wallet integration via MetaMask.
- **Storage on the Blockchain**: This architecture only makes sense if the smart contracts and data of an app are located exclusively on the Ethereum blockchain. Nevertheless, when all the data is stored on the blockchain, the costs can rise rapidly quite shortly. Here, users are charged for writing to the Ethereum blockchain since writing to the decentralized state machine expands the costs that the nodes have to incur to maintain the state machine. To not allow these costs to impact the application's users, one can integrate decentralized off-chain storage like IPFS or Swarm. IPFS can be explained as a distributed file system, which is used for data storage and data retrieving. Unlike a normal database where all data is stored in a centralized location, data in IPFS is shared worldwide with peers hence it can easily be retrieved when required. Moreover, IPFS also offers an incentives layer called Filecoin, that provides motivation to nodes for delivering the data. To store files and generate IPFS hashes to be stored on the blockchain, organizations such as Infura and Pinata offer IPFS nodes and services to the providers. Swarm is another example of a distributed storage network that has a reward system as well as a punishment system integrated into a decentralized application on the Ethereum platform. So now, with IPFS or Swarm, the application architecture looks like this: So now, with IPFS or Swarm, the application architecture looks like this:
- **Smart Contracts**: A smart contract is an actual digital code that is applied to the Ethereum blockchain environment and defines conditions of changes in the state of the blockchain. Programs used in smart contracts are developed in higher-level languages like Solidity or Vyper. The code of the smart contract is located on the

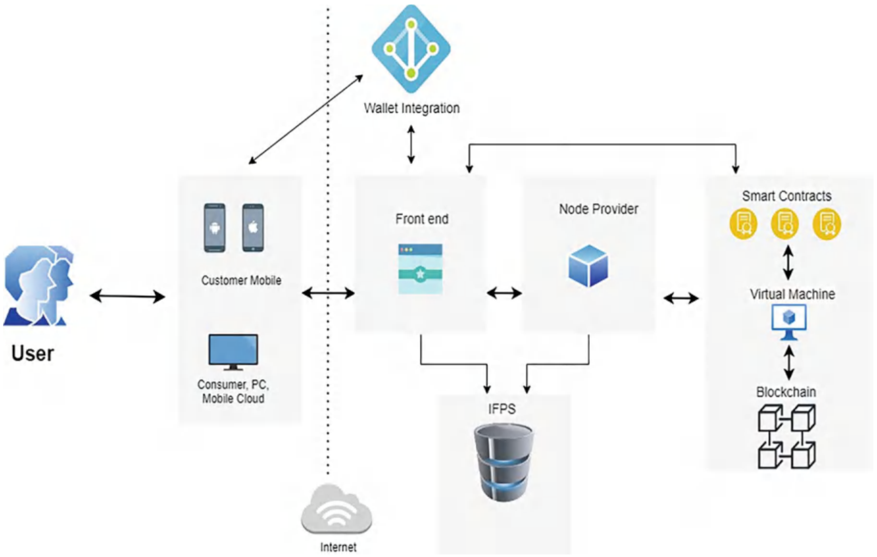


Fig. 4.13 Architecture of web app

Ethereum Blockchain; thus, all people can examine the application logic of each smart contract in the network.

- **Ethereum Virtual Machine (EVM)**: The Ethereum Virtual Machine implements the logic specified in smart contracts while the processing of the state transitions happens on this globally accessible program. The EVM does not support programming languages at the higher level such as Solidity and Vyper in which smart contracts are penned. However, these high-level languages must be compiled into bytecode that EVM is capable of running.
- **Blockchain**: Ethereum is also described as a 'world computer' due to its global availability as well as a deterministic state machine that is managed by the P2P network of nodes. Transitions on this state machine are determined by the rules of consensus that the peers of the network adhere to. This design enables any person in the world to read from this machine and write into it; thus, it belongs to the world with no central owner. Ethereum records can only be written to and never to be deleted (Fig. 4.13).

4.5.2 DApp Architecture

Decentralized applications are more commonly referred to as DApps, which are a change in software applications. As for the difference from traditional Web 2.0 applications, which in fact are web applications, they are based on centralized servers and use intermediaries for work. In Web 2.0, applications are operated by users yet applications themselves are run by applications residing in a central server owned by a single proprietor, which could be an issue of confrontation to the user's privacy, his/her data security, and probably his/her control. However, DApps are developed from Web 3. 0 Principles as concepts where users own and govern the data and assets, and the applications operate independently on the P2P network with no middleman involved. Below is the model of the architecture of DApp that has been proposed (Geeks for Geeks (n.d.).

- Front End: The front end of a Dapp, therefore, covers aspects such as application interfaces, which are the screens through which the users interface with a given application. This applies to websites' HTML pages, form controls, buttons, and many other user-interface components. Technologies that are applied in the front end include HTML, CSS, JavaScript, modern frameworks like React.js or Angular are used to design the front-end terminal of DApps.
- Wallet Integration: Users are expected to engage their digital wallet to sign transactions and handle resources and smart contracts in DApps. Wallet integration ensures that control and avail of the users' digital assets within or from the DApp securely. Mobile wallets and decentralized wallets such as MetaMask, Trust Wallet, and Coinbase Wallet provide application programming interfaces (APIs or SDKs) for incorporating wallet services into decentralized applications Blocktorch (n.d.).
- Decentralized Storage: DApps can use DSS for storing and accessing user data and information securely without a central point of failure. Applications such as IPFS, Swarm, or Storj use peer-to-peer distributed storage networks capable of providing availability, redundancy, and anti-censorship. As has been established, DApps that adopt decentralized storage benefit from the privacy and security of data and reduced dependency on centralized servers.
- Back End (Optional): Despite the fact that the DApps are, by their design, decentralized applications, they could also contain the back-end part responsible for the tasks that can't be performed directly on the blockchain, including data processing, authentication, or integration with the systems outside the blockchain. Back-end services can be located on usual servers or on such P2P structures as IPFS or Swarm. The back end engages the blockchain network through APIs or other interfaces.
- Smart Contracts: Smart contracts are digital contracts into which the terms of the agreement can be programmatically enforced. These exist on the blockchain network and are self-executing contracts that trigger certain operations at specified conditions. Smart contracts are at the core of DApps because they define how the application should conduct business and manage the interaction. Smart contracts are coded using programming languages, for example, Solidity for Ethereum or chaincode for Hyperledger fabric.
- Oracles (Optional): At times, DApps may need some extra information for processing /runtime data, or may need to trigger external events that are outside the view of the blockchain. There are certain preconditions that are necessary to be fulfilled by the developers so as to enhance the successful development of a

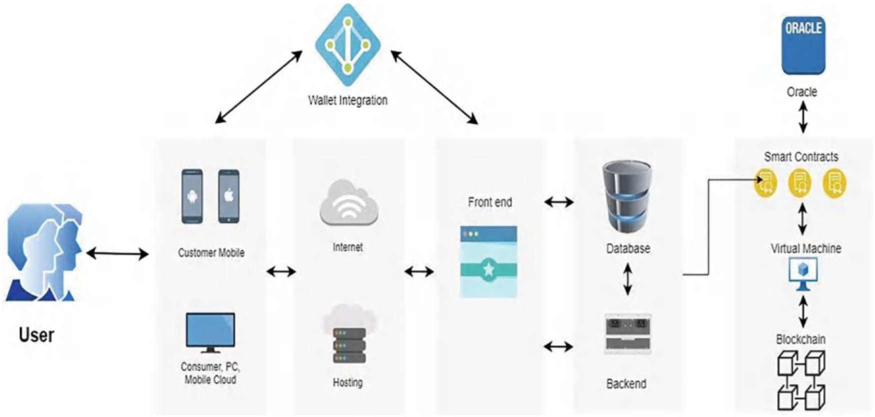


Fig. 4.14 Architecture of DApp

project. It involves general knowledge and technical knowledge as well as any other prerequisite that may be necessary.

- Blockchain: The blockchain is the decentralized platform on which the Decentralized Application or short DApp is situated and functions. It contains the distributed computerized record of all completed transactions and has a decentralized consensus system that makes the network more secure. DApps can be created on any blockchain platform of the developer's choice; these include Ethereum, EOS, Tron, as well as others; It's noteworthy that the discussed platforms are different and have different performances as well as their relative advantages and disadvantages (Fig. 4.14).

4.5.3 Prerequisites of Blockchain Development Process

There are several prerequisites that developers should consider to ensure a smooth and successful development process. These prerequisites comprise both technical skills and foundational knowledge.

- **Understanding of Blockchain Fundamentals**: With the objective of developing an effective decentralized platform good knowledge including the basics of blockchain such as decentralized networks, consensus, cryptography, and DLT should be possessed by developers. Thus, it provides a preliminary understanding of the idea of the blockchain and potential opportunities for using it and implementing it into various areas (Simplilearn n.d.).

Resources: Blockchain basics by Coursera (Bahga and Madisetti 2020), Mastering Blockchain by Imran Bashir (Kandaswamy 2024).

- **Hold in Programming Languages**: While working with blockchain, knowing programming languages is quite crucial since it helps the developers write smart contracts, execute, create decentralized applications and engage with the blockchain platform.

Common programming languages used in blockchain development include: Solidity, JavaScript, Python, and Go.

- **Familiarity with Blockchain Platforms and Tools**: Knowledgeable of the varieties of blockchain applications platforms, frameworks, and development tools could be important when doing development work on blockchain applications. This comprises understanding various features of the solutions such as Ethereum, Hyperledger Fabric, Corda, and Binance Smart Chain and the limits of each together with tools such as Truffle, Remix, Ganache, and MetaMask Dam L.7 (n.d.).

  Resources: Truffle Suite, Remix IDE, Ganache, MetaMask.

- **Knowledge of Smart Contracts and Solidity**: Knowledge of smart contracts and Solidity: The candidates should also have a good understanding of the best practices when developing smart contracts/ smart contract development principles and have prior coding knowledge in solidity as it is the language used when developing smart contracts for Ethereum or any other Ethereum virtual machine compatible blockchain. These are areas such as contracts, simple and complex data types, functions and their types, events, and inheritance.

  Resources: Solidity Documentation, CryptoZombies.

- Experience with Web Development: As many blockchain applications entail the development of user interface and integration to web technology, developers must have proficiency in web development technologies like HTML, CSS, JavaScript, and novel front-end frameworks like React.js, Angular, or Vue.js. This leads to the creation of great interfaces that users can use to access decentralized applications and blockchain networks.

  Tools: HTML, CSS, JavaScript, React, Angular, Vue.js.

- Knowledge of Cryptography and Security: Blockchain is a software development with a focus on aspects of cryptography and security measures aimed at guaranteeing the privacy and security of information and transactions. They should know about cryptographic algorithms, hash functions, digital signatures, encryption techniques, and security guidelines for smart contracts, wallets, and the blockchain network against kinds of threats and risks.

Tools/Resources: Crypto101 (Houtven 2020), OWASP Blockchain Security, Mastering Bitcoin by Andreas M. Antonopoulos (2023).

4.5.4 Design and Development of Blockchain Development Process

In the nascent stage, the practice of designing and developing a blockchain project entails going through several steps and factors that need to be taken into consideration in order to effectively and efficiently realize blockchain solutions ScienceSoft (n.d.).

- **Define Requirements and Use Cases**: The first step is to identify actual needs and desirable outcomes for the project that will be solved through the use of blockchain. The first of these steps is to define the problem space, define and set business outcomes, and then explore opportunities within the said problem space where blockchain solutions could add value, all within the context of supply chain management, decentralized finance (DeFi), identity, or assets management.
- **Choose the Right Blockchain Platform**: In consultation with the project requirements, need, and different use cases the blockchain platform most suitable for the need is then chosen. Others include issues of scalability, consensus, time of transaction processing, and many others Xilinx (n.d.).
- **Design the Architecture**: subsequently, the architecture of the blockchain solution is defined followed by the structure of the blockchain network, the part played by participants or nodes, consensus algorithm, and ways of data storage and access. Depending on the nature of the project whether it will be a public, private, or permissioned blockchain, the architecture design of the project is then determined.
- **Develop Smart Contracts**: In cases where there is a requirement for the operation of business-related operations on the blockchain smart contracts are used.
- **Implement Front-End and Back-End Components**: To improve and work out the case it is essential to create the front-end and back-end parts of the blockchain application. Front-end and back-end interfaces and applications such as HTML, CSS, JavaScript, and others; frameworks such as React. For the front end, some of the languages used are js, or Angular, while on the back end, some of the languages that are widely used are JavaScript Node JS, Python, or Java among others.
- **Integrate with External Systems**: The essential integration techniques are applied if the blockchain app requires interaction with other systems, for example, the regulatory systems, databases, APIs or the IoT nodes, and so on. During the integration of the blockchain technology with other systems common standards and API are applied in order to allow for smooth interaction between the network and the other systems while ensuring that data is synchronized.
- **Test and Debug**: The blockchain application undergoes a rigorous test in order to determine invalid states and to remove any form of disappointment or insecurity. The latter comprises unit tests, integration tests as well as end-to-end tests in order to confirm the correct functionality, efficiency, and security of the application.
- **Deploy and Monitor**: After the application is compiled and run, it is released for use in the actual setting or for the consumers. The efficiency, accessibility,

and safety of blockchain networks and the applications that are being created are being tracked with the help of monitoring tools and analytics.

4.5.5 Blockchain Benefit Enterprises Across Industries

Blockchain goes beyond conveying benefits to enterprises across industries. Blockchain is thus depicted as a solution with several benefits for enterprises of all forms and kinds of industries to revamp conventional business models and discover prospects for new added values. [Besides, it is necessary to identify other sources of revenue and to improve collection and management of payments from existing ones]s: [Besides, it is necessary to identify other sources of revenue and to improve collection and management of payments from existing ones] Pratt (n.d.).

- Enhanced Security: Blockchain technology utilizes cryptographic features and decentralized consensus to enhance the security of any data. In essence, all the transactions that occur on the blockchain are well encrypted, and/or are coded in a way that cannot be changed or altered.
- Increased Transparency: Blockchain has properties that are highly useful for creating a single source of truth within a network of information and/or transactions. Every transaction is stored in a block which is shared with all the users who have consented to the exchange of services thus avoiding third-party interference and possible fraud.
- Improved Efficiency: Blockchain integrates the various activities of a business where it automates manual activities, cuts down the paperwork, and elimination of middlemen. Smart contracts are self-executing contracts on the blockchain that run an agreed series of operations that have been pre-programmed when certain conditions are met minimizing the need for human interaction.
- Cost Reduction: In many cases, it expels intermediaries, optimizes various processes, and decreases the probability of fraud and mistakes, which can become major revenue-saving opportunities for enterprises. Contract policies are self-executing and there is no need to involve third parties, which reduces expenses, and due to the implementation of supply chain solutions that are based on blockchain, the amount of paperwork is minimized and work productivity is increased.
- Decentralization and Resilience: Blockchain solutions work together with decentralized systems, which involve the dissemination of business data over a network of nodes. This decentralization also removes the one-stop centres and thus enhances the robustness and reliability of the network. In case one node fails or becomes unavailable because of a network problem, others will go on and people will still find access to data and services.
- Innovation and New Business Models: Hereby, blockchain brings opportunities to construct new types of the company's business and revenue sources through P2P dealings, decentralized management, and tokenization of objects. DeFi enables the utilization of financial services on the blockchain without the need for brokers,

while on the other hand, NFTs transform the ownership and management of digital assets.

4.5.6 Life Cycle of Blockchain Transaction

The process of a Bitcoin transaction is carried out in different phases apart from the basic phases of signing, initiation, confirmation, and post-transaction phase. All of them have their functions in the defense of the blockchain network, non-interference with parties' transactions, and the provision of their authenticity, security, and relevancy. Ayoub et al. (n.d.).

- Transaction Initiation: The life cycle starts when a user wants to start a transaction and creates a transaction request to transfer tokens/ assets from one address to another address. This initiation may happen through a user interface, which is a wallet application or a decentralized exchange, or through contract interaction.
- Transaction Signing: After entering the transaction, the user authenticates the transaction using his/her unique identification code called the key. This step is rather aimed at providing the certainty of the transaction schedule and the identity of the participant making the payment for goods or services, as only this participant can create a digital signature containing the private key. The digital signature also serves the function of proving that the transaction is coming from the owner of the private key and has not been tampered with in any way when passing through other facilities.
- Transaction Propagation: Following the execution of a transaction, it is sent interconnective to the other nodes of the blockchain network. Every node in the network gets the transaction request and examines its authenticity with the help of set rules and consensus algorithms. The proposed transaction is forwarded across the network and received by several nodes in the network.
- Transaction Validation: Once there is a transaction, nodes in that network then work to confirm the authenticity of the transaction. This validation process includes verifying the signatures of the sender, ensuring that the sender has enough credits or enough permission from the network and of course, the transaction should not infringe the protocol of the system. The consensus algorithms employed are of a conclusive nature and may include such forms as Proof of Work (PoW) or Proof of Stake (PoS).
- Block Inclusion: In turn, the verified transactions are collected in blocks, which in turn are joined in the simplest linear chain—a blockchain. In PoW-based blockchains, miners vie to solve complex mathematical problems; whereas, validators stake their coins to generate more blocks in the case of PoS-based blockchains. Once a block has been successfully created it needs to be transmitted to the network and then attached to the blockchain which consists of the validated transactions.

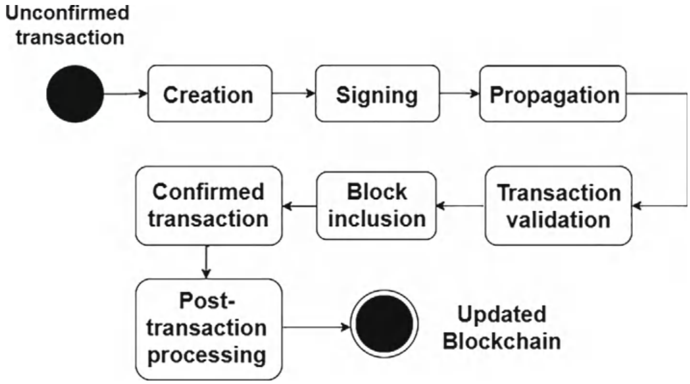


Fig. 4.15 Life cycle of blockchain transaction

- Confirmation: Once a transaction is included in a block and goes through the process of forming a string of blocks known as the block chain the said transaction is considered confirmed. This shows that a transaction cannot be reversed as subsequent blocks come to lay on top of the existing block, making the data irreversible in the blockchain database. The number of confirmations may also differ depending on the platform of microblockchain together with the corresponding level of the security of the given transaction.
- Post-Transaction Processing: With some more confirmations, the transaction initiates other processes or actions of the application as controlled by the smart contract features. This could include simply changing the account balances as well as the occurrence of certain events, the performance of conditional actions, or the calling of other transactions. Substantiation guarantees that changes resulting from the transaction implemented are also reflected as changes across the blockchain network and linked application systems platform (Fig. 4.15).

4.6 Platforms: Ethereum, IBM Blockchain IOTA, Corda

Actual applications in the spectrum of blockchain are platforms consisting of decentralized applications and smart contracts. Looking at the available options of the Blockchain platforms, Ethereum, IBM Blockchain IOTA (Trust over IP), and Corda can be considered as the three most preferred ones, based on their operational abilities and features (Ayoub et al. n.d.).

4.6.1 Ethereum

Ethereum as of the beginning of 2015 created by Vitalik Buterin at the end of 2013 is now one of the central pieces of the blockchain area. While blockchain technology is best known for serving as the foundation of digital currencies, this turns on it supporting the nucleation of decentralized applications, specifically DApps and smart contracts. The solutions are created based on Ethereum as it is an open-source blockchain that has served as a base for decentralization in many spheres starting from finance and ending with art.

4.6.1.1 Ethereum's Core Components

- **Smart Contracts**: However, the actual mechanism that brings into light Ethereum functions is smart contracts. These smart contracts are automated contracts that contain the provisions of the agreement in the form of codes. They directly put into practice the details of the contract and also regulate the execution of the contractual terms without the interference of third parties which reduces the chances of a contract containing incorrect and fraudulent provisions being produced. Smart contracts define a vast number of applications starting from the basic exchanging parties of value and reaching contractual relations with several participants.
- **Ethereum Virtual Machine (EVM)**: EVM is the initials of Ethereum Virtual Machine and it interfaces a virtual computing setting for executing smart contracts. It enables the developers to write programs for the applications in Solidity which is a language used in writing smart contracts for Ethereum. The EVM just ensures that the contracts run the former agreed terms and it will not cease or be changing, to be adjusted, fixed, or managed by any other person.
- **Ether (ETH)**: Ether is the internal virtual currency in Ethereum. It acts as the energy that powers smart contracts and DApps, Ethereum is an operating system that supports DApps. Customers use 'gas' which is in the local cryptocurrency Ether and every transaction requires fees to be paid to miners. It guarantees the network's protection and functionality at the same time.
- **Decentralized Applications (DApps)**: DApps are applications built on Ethereum that utilize this unique propensity of decentralized and trustless platforms. Obviously, as opposed to the conventional application, DApps do not have the concept of a single verifier, making the system more protected and clear. Some of the samples of DApps are Uniswap, CryptoKitties, and MakerDAO.

4.6.1.2 Impact on Industries

- Finance: Ethereum is a digital money requiring no middleman through decentralized finance (DeFi). DeFi platforms enable users to buy credits, borrowing tools, and trade without involving the services of a middleman such as a bank. This has led to the democratization of finance thereby making it possible for a larger number of people to access finance and other related services at cheaper rates.

- Supply Chain: Ethereum's blockchain which is capable of changing the way business is done can facilitate supply chain management by rendering transactions that cannot be altered. It eliminates fraud and increases efficiency in the tracking and tracing of goods to their point of origin. Businesses can trace their products right from the source, thus increasing the chances of a product being of good quality and meeting the set standard as well as other legal requirements.
- Real Estate: Within the real estate industry, Ethereum can help in property transactions since it eliminates paperwork and automates operations. Some of the activities of smart contracts include authentication, title deed transfer, and even rental agreements, which overall make transactions faster as well as secure.
- Healthcare: The concept of Ethereum can improve the issues of security and privacy of patient records in health care. Storing the medical records in a blockchain allows the patients and the healthcare providers to have an unchangeable ledger of the records and control who can access the information. The data may not be accessed and shared by any person, which decreases the possibility of leakage of records.

4.6.1.3 Innovations and Future Prospects

Ethereum always changes to fit the demands that the blockchain society has demanded. This is seen especially in the token standards that it supports, which include the ERC-20 token standard that has been widely used in ICOs and later in decentralized finance. Also, the ERC-721 that came later created a basis for unique tokens, or non-fungible tokens (NFTs), which changed the landscape of owning digital assets and art.

4.6.1.4 Scalability and Ethereum 2. 0

The proof-of-work consensus mechanism of the current system restricts the allowance of the number of transactions on the network hence congestion and high fees of transacting. In response to this, Ethereum is currently migrating to Ethereum 2. 0, a significant version that focuses on the enhancement of factors such as scalability, security, and sustainability. Ethereum 2. 0 will begin to adopt a proof-of-stake consensus model to help it cut back on energy which in turn means more transactions per second. This transition will be a rollup of rollups first by upgrading the beacon chain, then by adding new shard chains, and finally by merging the current Ethereum Mainnet into the new system.

4.6.1.5 Installation and Deployment

A. Install Node.js, npm, and Git.
B. Install Ethereum Client. Download and use one of the Ethereum clients (for example, Geth).
C. Like any other social network, to get started you need to set up a private network. Create a custom genesis.json file. Bootstrap the network with 'geth init'.
D. Start the Network. Start the private network using the 'geth–networkid' command.
E. Deploy Smart Contracts. Perform programmer-level compiling smart contracts in Solidity. Utilize with Truffle or Web3.js.
F. Interact with Smart Contracts. Use Web3.js for the various interactions for the web interface.

Ethereum is one of the revolutionary platforms in the new landscape of blockchain technology serving purposes beyond cryptocurrencies. It also has brought new capabilities in its smart contract as well as the support of DApps, which has provided new possibilities in different fields of industries for creation and decentralization. Today it faces these problems, but constant progress and updates make Ethereum more scalable and efficient in the future, so it is important to become a base layer in the blockchain industry. In the future, Ethereum is going to remain one of the main directs of development of decentralized technologies, becoming the initiator of the next wave of the digital transformation of the world (Zawalnyski n.d.).

4.6.2 IBM Blockchain IOTA

IBM Blockchain IOTA is a newly emerging next-generation solution that has been developed for the enterprise-level blockchain that has the inherent potential to define the standard for digital verification as well as secure business interactions in the digital world. As derived from the Hyperledger Fabric framework, IBM Blockchain IOTA focuses on identity, credential, and data sharing security and privacy as well as interconnectivity. With the use of DID, VC, and CP Intersect, IBM Blockchain IOTA seeks to enhance self-sovereign identity, particularly in digital interactions.

4.6.2.1 Distinctions of IBM Blockchain IOTA

- **Hyperledger Fabric Framework**: IBM Blockchain IOTA is based on the Hyperledger Material Platform, the blockchain platform which has a modular design for distributed letter solutions. Some of the features that users can take advantage of include permissioned networks, pluggability of consensus algorithms, and extensive privacy features to mention but a few, this makes it a perfect example for use for enterprise applications.

- **Decentralized Identifiers (DIDs)**: DIDs are integral to IBM Blockchain IOTA as they ensure that there are no authorities that manage the subjects' identities. DIDs allow one to control their personal/organizational identity thus enabling the enhancement of privacy and security. All DIDs are peculiar and can serve to attest to different transactions on the blockchain.
- **Verifiable Credentials**: Provable records are the credentials making use of blockchains for the generation, validation, and storage of credentials. These are documents whose authenticity is used in a situation to show qualification, or any characteristic that the holder has without further disclosure. For instance, a verifiable credential can demonstrate a user's age while avoiding the necessity to reveal the date of birth. Thus, there is selective disclosure which guarantees privacy but at the same time, trust is preserved.
- **Cryptographic Proofs**: Techniques like zero-knowledge proofs are employed to affirm interactions and transactions with the help of the blockchain without disclosing certain information. The above proofs make certain that the analyzed data is not only accurate but also secure from unauthorized access. One of the most important factors in IBM Blockchain IOTA's operations is cryptographic proofs as they help to provide its decent and trustworthy nature.

4.6.2.2 Impact on Industries

- Finance: IBM Blockchain IOTA can change the finance industry by offering fast and trustworthy ways for identification and credential checks. The platform can be of great advantage to financial institutions for improvised KYC as well as AML features, which lowers fraud and compliance expenses. They can also improve the aspects of cross-border operations and financial services based on the trusty sharing of the verified credentials provided by the platform.
- Supply Chain: In supply chain management, the IBM Blockchain IOTA provides improvement in the traceability function. When every supply chain process is documented on the blockchain platform, the businesses solve the problem of fake products and violations of the law. It lowers the incidences of fake and fraudulent products that may find their way into the market and thus enhances the flow of a genuine supply chain.
- Healthcare: IBM Blockchain IOTA offers numerous benefits in securing health data and protecting patient information. For instance, by adopting DIDs and verifiable credentials, the information about a patient can be exchanged with the proper authorization of other healthcare facilities without compromising data security. This act embraces the aspects of record accessibility, content, and confidentiality so that the patient record is complete, updated, and available only to authorized personnel.
- Education and Employment: Employers and educational institutions can help students use IBM Blockchain IOTA for secure credentialing. One way through which human ability can be incorporated into blockchain technology, especially

by students and employees is by having their qualifications, skills, and achievements engraved in the blockchain to avoid forgery. This makes the process of the verification of the academic degrees and other professional certifications more seamless thereby minimizing the instances of fraud in the production of fake certificates.

- Innovations and Future Prospects: IBM Blockchain IOTA brings new, more efficient solutions to the problem field of digital trust and identity. It focuses on their self-sovereign identity, verifiable credentials, as well as cryptographic proofs which offer the culture of the modern society that privacy and security are essential. Blockchain identity solutions are on track to be nearly universal, and IBM Blockchain IOTA looks ready to become the foundation of all interactions.

4.6.2.3 Scalability and Integration

A major advantage of the IBM Blockchain IOTA is it is adaptable and can be implemented effortlessly with other systems. The elements of modularity of the platform enable the businesses that implement the platform to develop their solutions in line with blockchain technology. Also, as a result of IBM's long-standing experience that has worked in the enterprise technology field, it guarantees that the IBM Blockchain IOTA being new is compatible with the former's system hence enabling the easy integration of the blockchain technology.

4.6.2.4 Installation and Deployment

A. Install Docker and Docker Compose.
B. Install Hyperledger Fabric. Get fabric binaries and Docker images.
C. Set up Fabric network. Duplicate the Fabric samples repository.
D. Launch the Network. To create the network, launch it with Docker Compose.
E. Deploy Smart Contracts (Chaincode). Limit the ability to write Chaincode in Go, Node.js, or Java. Deploy using Fabric CLI.
F. Interact with the Network. Utilize the product called Fabric CLI or application programming interfaces issued by the company.
G. Integrate DIDs, VC, and cryptographic proofs.

IBM Blockchain IOTA is a very strong tool that satisfies the need for protected, credible, and personally identifiable information interaction. Hence, through the use of the Hyperledger Fabric framework, decentralized identifiers, verifiable credentials, and cryptographic proofs, IBM Blockchain IOTA has a strong ground on which trust in the outcome of machine learning and artificial intelligence in the digital age can be established. Examples of its uses are found in finance, supply chain management, healthcare industry, and education to mention a few of them. While people are unknowingly engaged in billion transfers per day in electronic transactions; and as the complexity of the engagement tends to rise, a solution like the IBM Blockchain

IOTA offers this principle greater control of their data and hence, confidence and security in the digital world.

4.6.3 Corda

Corda is another type of blockchain that has been built to suit the needs of enterprises and it has been designed by R3. It mainly contrasts with the conventional public blockchains since it has a permissioned design, which means all members' identities are clear and legitimate. Given this focus on identity, security, and regulation, one can say that the key industries that are likely to benefit from Corda are finance, health care, and supply chains. The solutions' architecture at Corda allows for easy integration with the company's existing systems, which makes it scalable to handle the complexity of business transactions.

4.6.3.1 Core Components of Corda

- Permissioned Network: For Corda, the participants of the blockchain must obtain permission to be able to join and engage with the others in the network. This setup makes the environment more secure and less likely to be penetrated by scammers because all the parties are well-articulated. It also helps in meeting the compliance standards hence making Corda suitable to be used in organizations that operate within highly regulated environments.
- Smart Contracts & CorDapps: Smart contracts are also in Corda known as CorDapps (Corda Distributed Applications). These are applications that are executable on the Corda platform and they enable automatic and secure business flows. CorDapps are written in Kotlin, a language that is interchanged with Java, thus, it means that irrespective of the level of skill, one can develop CorDapps. Here are several applications enabling the business rules to be run on the blockchain, thus diminishing the need for intermediaries.
- Notary Service: Notary provided by Corda is vital to Integral's security as it guards against double-spending while finality provides assurance that the transaction it will be complete. Notaries affirm transactions, to confirm they are fresh and have not been already used. This particular service discussed here can be arranged in such a way that it will offer other degrees of privacy and security based on the business network's requirements.
- Data Privacy and Confidentiality: Security and data confidentiality are core parameters of Corda and form an essential component of the platform's design. While public blockchains work under a system where all the data is open for all the participants, Corda enables the transactions to be seen only by a selected number of participants. This selective disclosure makes it possible to maintain a strong security measure, where businesses' sensitive data do not appear on the public

network yet one can enjoy the transparency feature where everyone on the public network can see the clipboard data.

4.6.3.2 Impact on Industries

- Finance: Through associated Corda, Corda influences the financial industry and ensures the safety and efficiency of financial transactions between financial institutions. In particular, it encourages the development of new products, listed in global trading, including derivatives and syndicated loans, based on blockchain. It is a platform that offers chances to implement the ideas of the upgrades and remain secret and meet the needed requirements of the laws for the financial institutions and then reduce the costs and become a preferred choice for the banks.
- Supply Chain: In the supply chain industry, Corda improves the components' visibility and track record. With the record of each transaction in the supply chain on blockchain, counterfeiting, and quality compromise becomes frustrating to businesses. This also helps in cutting the instances of fraud and errors in the supply chain making the whole process efficient. Corda's selective data sharing also enables the business to ensure that certain information is not accessible while at the same time allowing the right parties to have visibility of the necessary details.
- Health care: Hence, Corda can be considered to be the best option since it focuses on permissioned networks and data privacy features that will be useful in the healthcare segment. It can maintain the records of the patients and also share the records with other personnel with the access rights only to touch the records. The above boosts the security of the data and adherence to regulatory rules like HIPAA. Furthermore, there is an opportunity to use Corda to preserve the confidentiality and authenticity of transactions regarding the pharmaceutical supply chain.
- Legal and Regulatory Compliance: The organization's structure of Corda complies with the legal and regulatory industries' strict needs. This makes it possible to fulfill the following regulations since it provides verifiable and immutable records of the transactions made. CorDapps can enable the generation of legal contracts and agreements where the formalities of legal processes can be significantly accelerated and made cheaper when implemented compared to normal procedures.

4.6.3.3 Innovations and Future Prospects

These features called Corda are an ever-growing product for the firm that remains in apposition to being developed with more features adding to the proposition of making it easier and easier to use and scale. Confidential identity and additionally, such features as the use of modern cryptographic methods contribute to the fact that this platform is constantly among modern blockchain solutions. Currently, R3, the organization that stands behind Corda, seeks to grow its partners and developers list, which creates a solid ground for the project's development.

4.6.3.4 Scalability and Integration

The organization of Corda is highly scalable, and it is capable of integrating with other systems of enterprises. This has the advantage of allowing one to implement change in the blockchain solution based on business hierarchy. Due to integration with existing systems, the use of Corda does not disrupt existing processes and easily adapts organizations to blockchain technology.

4.6.3.5 Installation and Deployment

A. Download and install JDK along with IntelliJ IDEA to your system.
B. Download Corda. First of all, open the GitHub website and save the Corda repository to your local machine.
C. Build Corda. It is noticeable that Gradle can be used to build Corda.
D. Manage nodes and modify the configuration of nodes.
E. Start the network. Look at node executables and start nodes.
F. Call CorDapps and issue CorDapps and build as well as package CorDapps. This implies that to successfully deploy the CorDapps, Copy JAR files to the directory of each node created as ‘CorDapps’.
G. Interact with the network. One should use the Flows in Corda via the RPC.
H. Develop an application that would encompass all financial operations with a client.

Corda is one of the most essential forms of the distributed ledger and blockchain that primarily fits the corporate world. Some of the areas that find its application ideal include permissioned networks, smart contract capabilities, and emphasis on data privacy make it ideal for industries that demand security and compliance with the law. Thus, Corda’s effectiveness in finance, supply chain, healthcare, and legal fields proves its applicability. On this note, as Corda grows into the next phase, it is bound to enhance even more the transformation of this sector of the enterprise blockchain making it the leading platform for a range of applications.

The blockchain platforms such as Ethereum, IBM Blockchain IOTA, and Corda are different in their concerns about the new technology but they all on the same goal to create an open and transparent environment as well as cement trust the digital interactions whether it is providing the financial freedom to its user through Ethereum, results on secure exchange network of data through IOTA or providing an enterprise level solution to the global market of businesses through Corda.

4.7 Future Research on Blockchain

The following are some of the potential domains of blockchain architecture and development process enhancements or future research directions:

1. Performance and Scalability

To raise transaction speed and reduce latency, Layer 2 solutions like state that means besides the main channel, other affiliated channels and rollups can be optimized. Thus, it is recommended that more research is conducted on sharding strategies for the efficiency of splitting blockchain networks and managing processing demands in relation to information. Exploring new consensus algorithms that offer greater scalability while at the same time sacrificing decentralization or security.

2. Cross-Chain Communication:

Develop policies and proscriptions to facilitate friendly interface and transfer of information between various blockchain networks. Enhance the efficiency and also the security of the blockchain bridges to enable the transfer of data and assets from one chain to another.

3. Security:

Design and propose algorithms for cryptography to be implemented and be resistant to quantum computing. He has secured vulnerabilities in smart contracts to detect and solve and to offer formal verification procedures and automation too. Analyze the state of the art in protecting privacy including Homomorphic encryption and zero-knowledge proofs.

4. Governance:

Decentralized Governance Models: Traditional blockchain governance mechanisms often face challenges in ensuring meaningful community participation and effective decision-making. Future research should focus on designing and evaluating more robust, transparent, and inclusive decentralized governance frameworks. This includes exploring voting mechanisms, proposal systems, and conflict resolution protocols. Additionally, attention should be given to developing incentive and reward structures that align the interests of various stakeholders and promote long-term network sustainability.

Acknowledgements This work was supported by the Islamic University of Science and Technology.

References

Antonopoulos AM, Harding DA (2023) Mastering bitcoin: programming the open Blockchain, 3rd edn. O'Reilly Media, Sebastopol. https://www.oreilly.com/library/view/mastering-bitcoin-3rd/9781098150082/

Ayoub M, Algarni F, Quasim MT Decentralised Internet of Things. ResearchGate. https://www.researchgate.net/publication/339242145_Decentralised_Internet_of_Things

Bahga A, Madisetti V (2020) Mastering Blockchain: A deep dive into distributed ledgers, consensus protocols, smart contracts, DApps, cryptocurrencies, Ethereum, and more, 3rd edn. Packt Publishing, Birmingham

Biswas S (n.d.) Consensus in Blockchain. Clear Tax. https://cleartax.in/s/consensus-in-blockchain

101 Blockchains. Blockchain Ecosystem. https://101blockchains.com/blockchain-ecosystem/. Accessed 23 Jul 2024

Blocktorch (n.d.) Architecture of Decentralized Applications (dApps). Medium. https://medium.com/@blocktorch/architecture-of-decentralized-applications-dapps-d583db198a6f

Chelliah P, Saini K, Surianarayanan C (2020). Blockchain Technol Appl. https://doi.org/10.1201/9781003081487

Chiat (2024) 5 layers of blockchain architecture. Available via Chiat. https://chiatribe.com/5-layers-of-blockchain-architecture/

Dam L.7 Steps in the Blockchain development process. Ekotek. https://ekotek.vn/blockchain-development-process

DX Talks (n.d.) Understanding Blockchain ecosystems: a complete guide. https://www.dxtalks.com/blog/news-2/understanding-blockchain-ecosystems-a-complete-guide-515

Finance Strategists (2024) Blockchain architecture. Available via Finance Strategists. https://www.financestrategists.com/wealth-management/blockchain/blockchain-architecture/

Geeks for Geeks (n.d.) Architecture of a dApp. https://www.geeksforgeeks.org/architecture-of-a-dapp/

Hacken (2023) Top Blockchain Platforms. https://hacken.io/discover/blockchain-platforms/

Hacken (2024) Blockchain architecture layers: a comprehensive guide. Available via https://www.hacken.io/blockchain-architecture-layers-guide

IBM (n.d.) Smart Contracts. https://www.ibm.com/topics/smart-contracts

IEEE (2022) A sophisticated analysis of the blockchain technology and its applications domain. In: Proceedings of the IEEE

Javatpoint (2023) History of Blockchain. In: Javatpoint. Available via Javatpoint. https://www.javatpoint.com/history-of-blockchain. Accessed 12 July 2024

Kandaswamy R (2024) Blockchain Basics. Coursera. https://www.coursera.org/learn/blockchain-basics

Laurens Van Houtven ‘lvh’ (2020) Crypto 101: The introductory book on cryptography. GitHub, San Francisco. https://github.com/crypto101/book

Nakamoto S (2008) Bitcoin: a peer-to-peer electronic cash system. Available via bitcoin.org. https://www.bitcoin.org

Naqvi SJ Converting a property rental paper contract into a smart contract. Medium. https://medium.com/@naqvi.jafar91/converting-a-property-rental-paper-contract-into-a-smart-contract-daa054fdf8a7

Patel (n.d.) Consensus algorithms in Blockchain. Geeks for Geeks. https://www.geeksforgeeks.org/consensus-algorithms-in-blockchain/

Pratt MK Top 10 benefits of Blockchain for business. TechTarget. https://www.techtarget.com/searchcio/feature/Top-10-benefits-of-blockchain-technology-for-business

Qutub A Demystifying web application architecture: a comprehensive analysis of web app components. Enterprise Monkey. https://enterprisemonkey.com.au/web-application-architecture/

Rawat DB, Chaudhary V, Doku R (2020) Blockchain technology: emerging applications and use cases for secure and trustworthy smart systems. J Cyber Secur Priv. https://doi.org/10.3390/jcsp1040021

Science Soft Blockchain Development Guide. https://www.scnsoft.com/blockchain/blockchain-development-guide#:~:text=Blockchain%20Development%20Guide%20Highlights&text=Conduct% 20a%20feasibility%20study.,Deploy%20the%20solution

Simplilearn How to Become a Blockchain Developer | Must Have Skills. https://www.simplilearn.com/tutorials/blockchain-tutorial/how-to-become-a-blockchain-developer

Vadapalli R (2020) Fundamentals of Blockchain, 1st edn. Blockchainprep, UAE, ISBN: 301.345.908

Xilinx (n.d.) The Developer’s Guide to Blockchain Development. https://www.xilinx.com/products/design-tools/resources/the-developers-guide-to-blockchain-development.html

Zawalnyski A. The Top 8 Blockchain Platforms. Expert Insights. Available at: https://expertinsights.com/insights/the-top-blockchain-platforms/.

Chapter 5
Hyperledger Fabric: The Enterprise-Grade Blockchain for Modern Business Applications

Swatisipra Das, Minati Mishra, and Rojalina Priyadarshini

Abstract One among the rising innovations, blockchain technology is popular for its attractive features like consensus mechanisms, Merkle tree, distributed ledger, smart contracts, and peer-to-peer networks. As a decentralized technology, it ensures transparency, and immutability of transaction data, which can sometimes lead to privacy concerns for certain business applications in permissionless or public blockchains. In addition to privacy, there are other issues like low transaction throughput, high transaction time, and high cost of computing resources, etc. However, these issues are effectively addressed by permissioned or private Blockchain networks. Hyperledger Fabric is an open-source project of the Linux Foundation, designed for modern enterprise-level applications and industrial solutions. The Fabric has advanced privacy control over the participant's data. Transactions are traceable, and irreversible which helps to build trust between organizations. This chapter explores and analyzes the architecture, components, transaction flow, endorsement flow, endorsement policies, data distribution protocol, chaincode, certificate authority, membership service provider, and the step-by-step procedure of creating a Fabric network. This also examines the application of Hyperledger Fabric in various businesses such as health care, supply chain, and finance.

Keywords Hyperledger Fabric · Permissioned network · Hyperledger tools · Certificate authority · Membership service

S. Das · M. Mishra (✉)
Department of Computer Science, Fakir Mohan University, Balasore, India
e-mail: minatiminu@gmail.com

R. Priyadarshini
Department of Computer Science and Engineering, C. V. Raman Global University, Bhubaneswar, India

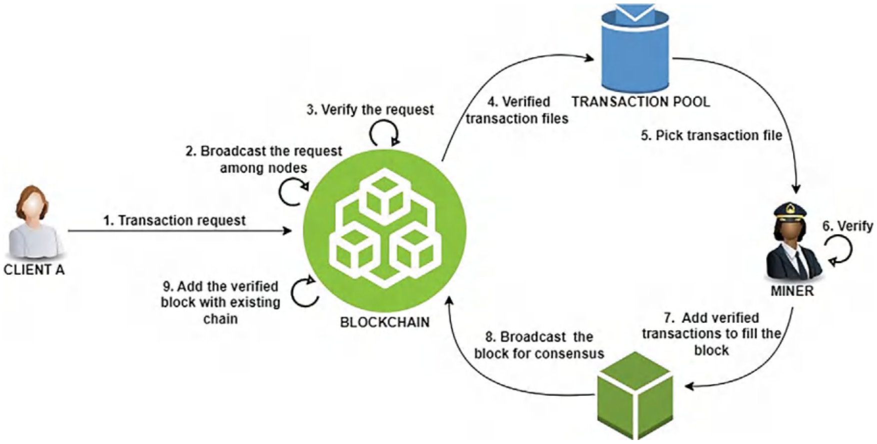
K. Ahmad et al. (eds.), *Fostering Machine Learning and IoT for Blockchain Technology*,
Transactions on Computer Systems and Networks,
https://doi.org/10.1007/978-981-96-4078-2_5

5.1 Introduction

To eliminate third-party interference in online transactions, decentralized blockchain technology emerged alongside the groundbreaking innovation of Bitcoin (Dabbagh et al. 2019). Blockchain stores transaction data in the form of interconnected blocks. Each block can accommodate multiple transactions, depending upon the size of the block. These interconnected blocks store the hash of previous ones to ensure immutability (Leng et al. 2020). For achieving consensus in a distributed system like blockchain, consensus algorithms are employed to establish agreement among all peers in the network. These algorithms are crucial for transaction validation, maintaining system state, synchronizing the distributed ledger, and providing incentives (Touloupou et al. 2022). Blockchain also utilizes smart contracts for the automatic execution of terms and conditions between untrusted parties. These self-executing contracts, introduced by Nick Szabo ensure seamless transaction processing (Kushwaha et al. 2022). To maintain the integrity of transaction data within a block, Blockchain uses the Merkle tree that concatenates hash values of data items in chronological order. This ensures data integrity and security (Monrat et al. 2019). The sequence of transaction processing in Blockchain is given in “Fig. 5.1”.

5.1.1 *Types of Blockchain*

Depending upon the requirement and usage, blockchain can be public, private, or consortium (federated).



3. Verify the request
4. Verified transaction files
TRANSACTION POOL
5. Pick transaction file
2. Broadcast the request among nodes
1. Transaction request
CLIENT A
6. Verify
MINER
9. Add the verified block with existing chain
BLOCKCHAIN
8. Broadcast the block for consensus
7. Add verified transactions to fill the block

Fig. 5.1 Transaction flow in blockchain

i. Public Blockchain: These kinds of blockchains are open to everyone and every transaction record is viewable to all. Everyone can take part in the consensus process maintaining a replica of the ledger (Khettry et al. 2021; Sarmah 2018).

ii. Private Blockchain: Also referred to as permissioned blockchain. In this case, the nodes need permission or invitation from the organizer to get admission. It is a fully centralized network where restrictions are there on who can take part in the consensus process. This type of network is suitable for enterprise use cases where privacy, control, and security are of paramount importance (Khettry et al. 2021; Sarmah 2018).

iii. Consortium Blockchain: Also known as federated blockchain which follows a partially decentralized architecture. Unlike the public blockchain, these blockchain networks are private and permissioned in which only predefined entities are allowed to participate. The consensus process is controlled by a group of pre-selected nodes (Khettry et al. 2021; Sarmah 2018).

5.1.2 Hyperledger Project

To advance blockchain technologies across various industries, the Hyperledger project was launched in 2015. It is a permissioned, distributed ledger, free and open-source platform created by Linux foundation (Satapathy et al. 2019). In this global open-source collaboration forum, numerous industry leaders are involved. Some of the industry benefits from Hyperledger and its family projects are as follows (Baset et al. 2018):

i. Sustainable development: Hyperledger project provides sustainable development which implies that active community members review each other's contribution to ensure quality and security.

ii. Innovation and Extensibility: By using open-source blockchain technology, enterprises need not follow the vendor locked-in process (depending on the service of a single vendor). It gives the flexibility to choose vendors from the most innovative and active communities. Active communities always contribute new and innovative business ideas for improvement.

iii. Quick response: Hyperledger projects are driven by an active and large community of blockchain specialists who provide rapid response in vulnerability identification and software solutions.

This project provides a number of frameworks and tools that are shown in "Fig. 5.2" (Sah et al. 2021; Li et al. 2019; Aggarwal and Kumar 2021; Dhillon et al. 2017). The details of these tools and frameworks are discussed in the following subsections.

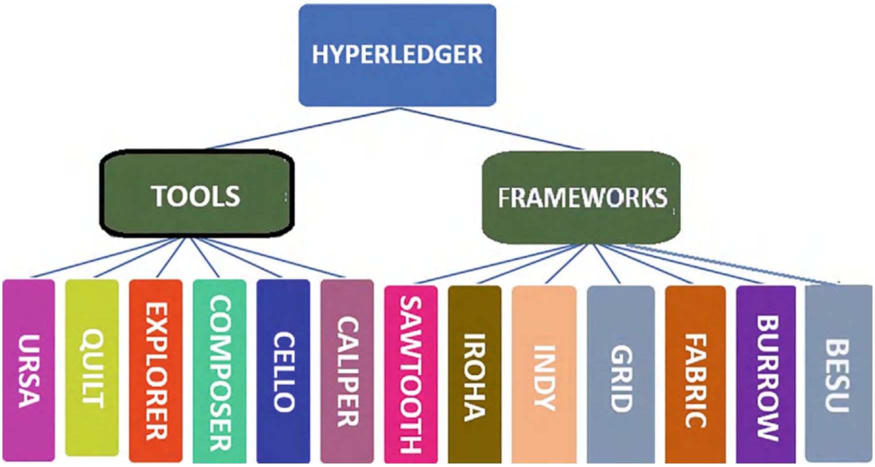


Fig. 5.2 Hyperledger umbrella

5.1.2.1 Hyperledger Frameworks

Since its development in 2015, the hyperledger project has gone through several major releases including, Sawtooth, Iroha, Burrow, Fabric, Grid, Indy, and Besu (Sah et al. 2021; Li et al. 2019).

1. Hyperledger Sawtooth: Sawtooth was developed by Intel with the intention to design and deploy a distributed ledger. Sawtooth introduced the Proof of Elapsed Time (PoET) consensus process, that accommodated a bigger group of distributed validators with a negligible resource requirement (Sah et al. 2021; Dhillon et al. 2017). It is an integrated blockchain platform designed to build, deploy, and execute distributed ledgers. It is popular for its scalability, flexibility, and support for various consensus algorithms. The key features of Hyperledger Sawtooth include modular architecture, support for multiple consensus algorithms such as PoET, Practical Byzantine Fault Tolerance (PBFT), and Raft. Parallel transaction execution, deployment and execution of smart contracts using various languages including Python, JavaScript, and Go, permissioned and permissionless deployment, dynamic on-chain settings, flexibility and adaptability to various business needs, scalability, interoperability and Integration, collaboration and community support are some of the additional features of Sawtooth.
2. Hyperledger Iroha: Iroha was the result of a joint collaboration of Nippon Telegraph and Telephone (NTT) Data (a Japanese system integration company), Sora Mitsu (a Japanese fintech company), Hitachi (global conglomerate headquartered in Japan), and Colu (an Israeli blockchain technology company). They have designed and developed Iroha with the intention of easy integration of distributed ledger technology in the required projects. The major focus of Iroha is to design and develop mobile applications for both Android and iOS with client libraries (Li

et al. 2019). It is a versatile and user-friendly distributed ledger platform designed for the creation and management of robust, reliable, and secure blockchain applications. Its easy to use interface and seamless integration make Iroha a popular choice for both enterprise and personal use cases. The key features of Hyperledger Iroha include simplicity, modular design, strong role-based permissions, rich query capabilities, fault tolerance, customizability and scalability. It uses Yet Another Consensus (YAC)–a pluggable consensus algorithm. Iroha also offers a user friendly interface, high Performance and robust cryptographic protections to ensure the security of data and transactions.

3. Hyperledger Indy: The Hyperledger Development Kit (HDK) of Indy provides a set of tools, libraries, and reusable components that developers can use to design digital identity management solutions (Sah et al. 2021). It is a distributed ledger platform specifically designed to manage distributed identity. The tools, libraries, and reusable mechanisms of Indy help in the process of creation and management of identities in blockchain or other decentralized ledgers. The key features of Hyperledger Indy include decentralized identity management, verifiable credentials, privacy and security, interoperability, Plenum consensus protocol for efficient and scalable identity-related transactions. Users have full control over their identity data, reduced identity fraud by use of strong cryptographic methods enhanced privacy with selective disclosure techniques, interoperability.
4. Hyperledger Grid: With the help of data models borrowed from existing open standards, smart contracts, and distributed ledgers, Grid provides a platform for industries to design supply chain solutions with ease (Sah et al. 2021). This open-source framework provides a set of tools, libraries, and data models that developers can use to build and deploy supply chain applications on Distributed Ledger Technology (DLT). Hyperledger Grid helps organizations in restructuring their supply chain operations to improve transparency and efficiency. The flexible and modular design of Grid makes it a favorable candidate for industries working on supply chain solutions.
5. Hyperledger Fabric: This widely used, robust, and flexible, IBM framework provides dedicated channels to achieve privacy in the distributed blockchain networks. This helps organizations in creating dedicated channels between them for secret communications. There are certification authorities who provide participants with identity certificates to take part in various transaction processing activities.

   It is a scalable and integrated open-source framework for the development of enterprise-grade applications for industries operating in the fields of manufacturing, health care, finance, and supply chain with ease and efficiency. Due to its robust security features, flexible architecture, and ability to handle complex and high volumes of transactions, Hyperledger Fabric is a popular choice in several industries.
6. Hyperledger Burrow: This has been specifically designed for a multichain environment. There are three main components in Burrow such as consensus engine, the permissioned Ethereum Virtual Machine (EVM), and a remote call gateway to execute contracts (Dhillon et al. 2017). It is an open-source framework that

offers a modular and permissioned blockchain client with a built-in EVM. It facilitates the development of smart contract applications with a focus on performance, simplicity, and interoperability within enterprise settings. This creates a platform for the private blockchains to adopt the best features of the Ethereum smart contract. Its focus on performance, modularity, and interoperability makes it an attractive alternative for enterprises to implement blockchain to fit their predefined requirements.

7. Hyperledger Besu: This enterprise-friendly, open-source Ethereum client has been designed to support both permissioned and permissionless blockchains. Besu is written in Java and provides a comprehensive suite of features for deploying and managing blockchain networks. The key features of Hyperledger Besu include Ethereum compatibility, secure and confidential transactions within a consortium and account-level permissioning, flexible deployment options, advanced monitoring and management, interoperability, enterprise-grade performance, scalability, strong community support.

5.1.2.2 Hyperledger Tools

Hyperledger Tools (HTs) are used to maintain and execute blockchain. With the help of HTs, it is easier to monitor and discover information from distributed ledgers. These are also helpful in designing and modifications in the blockchain network (Sah et al. 2021, Li et al. 2020). Listed below are some of the popular HTs.

1. Hyperledger Caliper: It is a performance monitoring and benchmarking tool that offers reliable matrices to measure network throughput, transaction executed per second, transaction volume, and so on. By providing a standardized framework for evaluating blockchain performance, this helps organizations make informed decisions about their blockchain technology investments and improvements.
2. Hyperledger Cello: Cello is an on-demand blockchain that minimizes the efforts to create, manage and abort blockchains. It aims to reduce the effort required to manage, operate, and deploy blockchain networks, making blockchain technology more accessible and manageable for organizations thereby accelerating the adoption and implementation of blockchain technology across various industries.
3. Hyperledger Composer: This PoC (Proof of Concept) based tool provides the organization with a user-friendly and easy to use interface for developing blockchain-based business models. The high-level abstraction, provided by this tool for defining business networks, makes it easier for developers and business users to create and manage blockchain solutions without bothering about the complexities of the blockchain stack.
4. Hyperledger Explorer: It is a user-friendly web application that allows users to view, monitor, and manage blockchain networks by exploring data, transactions, network information, ledger, and smart contracts.
5. Hyperledger Quilt: This simple and integrated business blockchain tool helps interoperability among different blockchain networks and provides a platform

for transactions across different DLTs and payment systems. The Inter-Ledger Protocol (ILP) suit of Quilt promotes integrated financial and technological setups through interoperable and linked networks that overcome the isolation between DLTs.

6. Hyperledger Ursa: Ursa encompasses a modular and flexible cryptographic library with multiple signature schemes that help the developers integrate cryptography into their blockchain applications. This tool helps developers in designing secure and interoperable blockchain solutions without requiring them to design their own cryptographic algorithms.

5.2 Hyperledger Fabric Model

Fabric follows an Execute-Order-Validate (EOV) transaction architecture (Shalaby et al. 2020). This open-source, permissioned blockchain framework comprises a set of member organizations and entities such as membership services, certificate authorities, nodes, and peers (Aggarwal and Kumar 2021; Gaur et al. 2020). A diagrammatic representation of Fabric architecture is given in "Fig. 5.3".

1. Membership Service Provider (MSP): MSP maintains the identities, roles, and permissions available with the participant nodes of the Fabric network (Gaur et al. 2020). The identities are provided in the form of digital certificates and these certificates ensure participation of authenticated nodes in the Fabric network (Aggarwal and Kumar 2021).

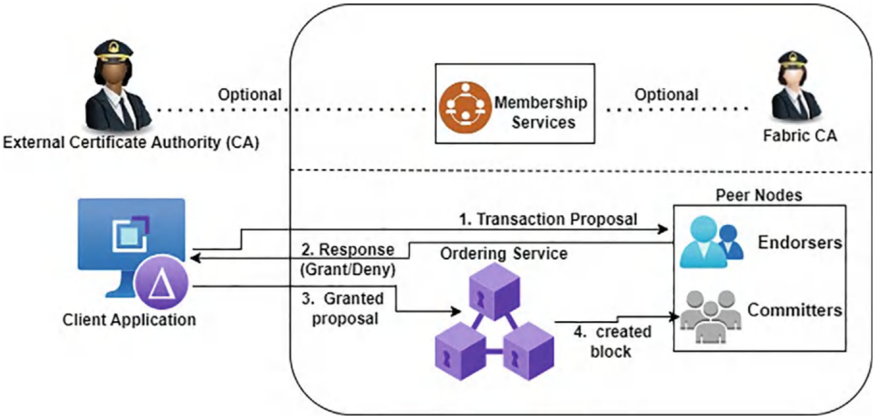


Fig. 5.3 Hyperledger Fabric architecture

2. Certificate Authority (CA): CA is responsible for providing certificates to all the participants. A CA can be an insider or outsider and ensures public and private key pair to each node of the network. The nodes use the public–private key pair to perform a transaction in a Fabric network (Aggarwal and Kumar 2021).
3. Nodes: Orderer, peer, and client are the three kinds of nodes present in a Fabric network. Each of these nodes has equal admission, access, and validation rights (Aggarwal and Kumar 2021). Client nodes are responsible to submit the transaction request. Orderers are the creator of blocks and they guarantee the atomic delivery of blocks to all the peer nodes. The consistency of the distributed ledger is also ensured by these Orderers of the network (Foschini et al. 2020). There are three types of peers, each with different responsibilities. Peer nodes mainly act as transaction validators in the network (Gaur et al. 2020).
4. Peers: Peers can be endorsing or committing peers. Endorsers receive transaction proposals from client nodes and send their decisions after verification. Committing peers are responsible to commit the transaction proposal and maintain the ledgers (Aggarwal and Kumar 2021). There are three types of committing peers as given below:

   i. Regular peer: A peer that performs the basic functions of maintaining the ledger and chaincode state but does not perform specialized roles like endorsing, leading, or anchoring unless specifically configured to do so is a regular peer. This division of roles helps in optimizing the performance and scalability of the blockchain network.
   ii. Anchor peers: A member organization can have one or more discoverable anchor peers. These peers are responsible for cross-organization communication. The connection between an anchor peer of one organization is not possible with the regular peer of another organization. Anchor peers are defined at the time of channel configuration.
   iii. Leader peers: The leader peers of the Fabric network distribute blocks received from the orderer to other nodes of the member organization using the GOSSIP data dissemination protocol. Leader peer selection can be static or dynamic. In static selection, any number of peers are selected as leaders while in dynamic selection, only one peer is selected as leader of an organization.

The diagrammatic representation of Fabric architecture is shown in “Fig. 5.3”.

In this architecture, the MSP provides identities to the admin, orderers, and peers of the network. These identities are digital certificates issued to the actors from the Fabric CA and prove the legitimacy of the actors. The certificate authorities can be internal as well as external. There are one or more admins in the Fabric network, who are responsible to manage the peer nodes within the organization. Clients send transaction requests to endorsing peers for verification and after getting approval, it forwards the request along with the endorsing peer’s approval to the orderers. Orderers are responsible for ordering services. They create blocks and send the blocks to committing peers for verification and ledger update. The different types

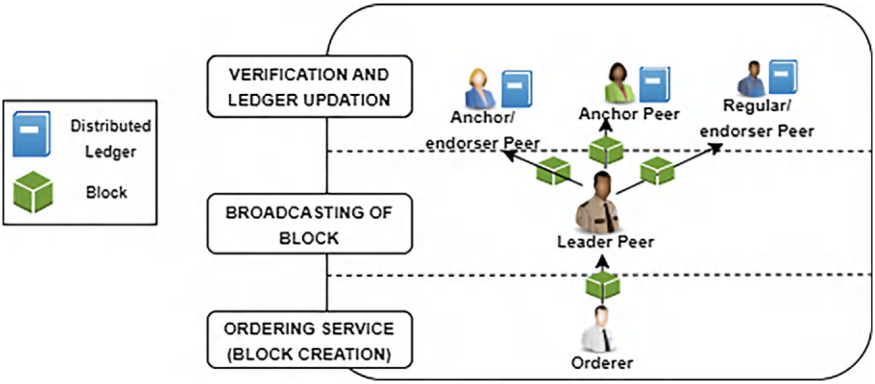


Fig. 5.4 Peer nodes involved in a Fabric network and block passing between them for verification

of nodes involved in Fabric network along with the passing of data blocks between them are shown in “Fig. 5.4”.

5.3 Hyperledger Fabric Components

The Hyperledger Fabric framework comprises several key components that work together to create, deploy, and perform other operations of permissioned blockchain networks. Some of the major components of Fabric are listed below.

1. Ordering Service: Orderers are specific nodes in the Fabric network who, oversee the transaction requests and add those to blocks. “Raft”, “kafka”, and “solo” are three popular ordering services. “Solo” is used for development with only one orderer node (Nguyen et al. 2019). “Kafka” uses a leader–follower model. The leader directs transactions to the follower orderers. It follows the dynamic approach for leader selection. “Raft” is based on the raft protocol and like “Kafka” it also follows the leader–follower model. The only difference between “Raft” and “Kafka” is, as compared to “Kafka” “Raft” is easier to set up (Shalaby et al. 2020).
2. Channel: It is a private communication pathway for conducting transactions between specific network members. It allows multiple parties to transact privately without sharing data with unauthorized parties. Every channel maintains its own ledger and smart contracts. A channel consists of members like member organizations, anchor peers of each participant organization, shared ledger, chaincode application, and orderers (Gaba et al. 2022).
3. Smart Contracts (Chaincode): These are automated contracts embedded into the code. The business rules and regulations between participant organizations are written in the form of code. The smart contracts are packed into a chaincode and the chaincode is then installed on peers (Gaba et al. 2022).

4. Ledger: It is a record of every transaction occurring on the network. Ledger maintains a complete and immutable history of transactions, providing transparency and auditability. Each peer maintains the ledger at its end. The ledger state records details regarding a business object. The Peer Transaction Manager (PTM) maintains the transaction in a key-value-version pair. Key is a business object, value is the set of details regarding the object, and version is a unique number comprising of a block sequence number and the in-block transaction sequence (Androulaki et al. 2018).
5. Consensus Mechanism: Hyperledger Fabric supports pluggable consensus mechanisms. Organizations choose appropriate consensus algorithms as per their requirements. The consensus algorithms supported by Hyperledger Fabric include Practical Byzantine Fault Tolerance (PBFT), Raft, and many others.

These components work together to provide a secure, scalable, and flexible blockchain platform suitable for a wide range of enterprise use cases. The modular architecture of Hyperledger Fabric allows customization and integration with existing enterprise systems and this makes the Hyperledger Fabric a standard for private blockchain networks. The diagrammatic representation of different organization's participation in channels is shown in "Fig. 5.5".

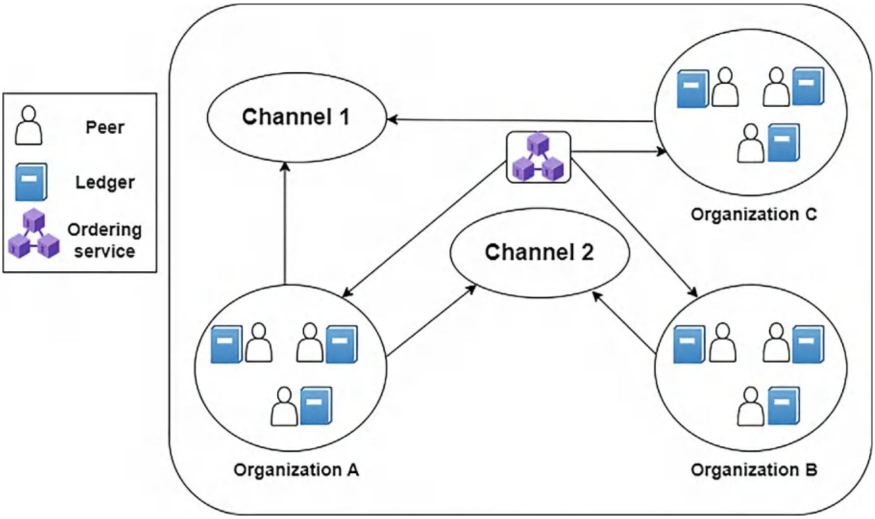


Fig. 5.5 Structure of different organization's participation in different channels

5.4 Hyperledger Fabric Transaction Flow

From the initiation of a transaction proposal to the final commitment of the transaction to the ledger, the transaction flow in Hyperledger Fabric involves several steps to ensure that the transactions are processed securely, consistently, and efficiently, maintaining the integrity and consistency of the network. A detailed overview of the process is discussed below.

Hyperledger Fabric follows an EOV process for the execution of a transaction request (Shalaby et al. 2020; Nguyen et al. 2019; Androulaki et al. 2018).

5.4.1 Execute Phase

This is the first phase of Fabric transaction flow and the main goal of this phase is endorsing the transaction requests. Following are the responsibilities of this phase:

1. Submission of transaction proposal from client to endorsing nodes: The client creates a transaction request and submits the request to a group of chosen endorsing peers of the network. This requires the client to know the number of endorsing peers present and the number of endorsing nodes required to achieve consensus.
2. Verification of transaction by endorsing peers: Upon receiving a transaction request from the client, each endorsing peer checks the following:

   i. Structure of the transaction proposal.
   ii. Whether it is a duplication of an existing transaction?
   iii. Validity of the requester's signature.

Each endorsing peer invokes the chaincode at their end to generate the result (grant/deny) and sign on the result.

3. The endorsing peer forwards the signed transaction to the client. To move to the next step client must have a threshold number of responses from endorsing peers.

5.4.2 Order Phase

This phase is responsible for the verification of responses and block creation.

1. Verification of responses received from endorsers: The client verifies the signature of the endorsing nodes and compares all the received responses. If the client node receives a threshold amount of grant responses, then it sends the proposal along with its responses to the orderers.
2. Block creation: The orderers order the received requests and add the verified transaction requests in a block.

5.4.3 *Validate Phase*

This phase is responsible for the validation of transactions of a block and committing the block to the ledger.

1. Broadcasting of created block: A member organization can have more than one leader peers. The orderer node forwards the created block to every leader peer of the network. The leader peers of an organization receive the block and send it to the regular and anchor peers of the organization using the GOSSIP protocol.
2. Updating the distributed ledger: Each peer validates the transactions in a deterministic way. After the transactions are validated by committers (anchor and regular), the peers add the verified block to its channel's chain. Every channel in the network has a blockchain and each peer can be a participant of more than one channel. The ledger is updated by the verified transaction data.

The diagrammatic representation of Fabric transaction phases and transaction flow are presented in "Figs. 5.6", and "5.7".

5.5 Hyperledger Fabric Endorsement Flow

In Hyperledger Fabric, the endorsement phase is one of the important phases, where transaction proposals get validated before being recorded on the blockchain. After the generation of the transaction request, the client sends the request along with its signature to the endorsing peers of the network. After receiving the transaction request, each endorsing peer verifies the request at their end (Kwon and Yu 2019). They all will perform the below-listed operations:

1. Identity verification of the client: To verify whether the requested client is authentic or not, endorsers request the MSP for the client's identity verification.

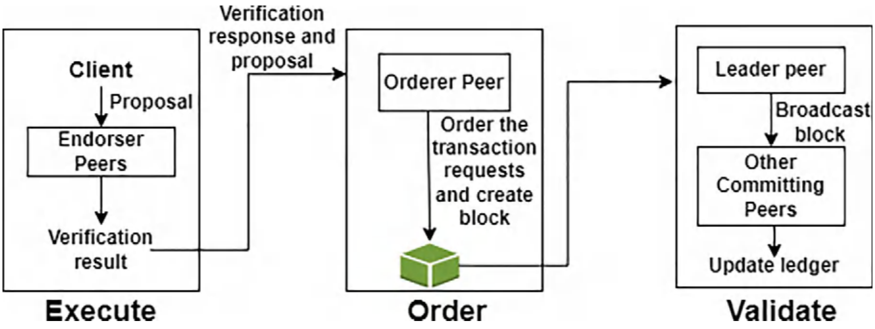


Fig. 5.6 Working of execute-order-validate of Hyperledger Fabric

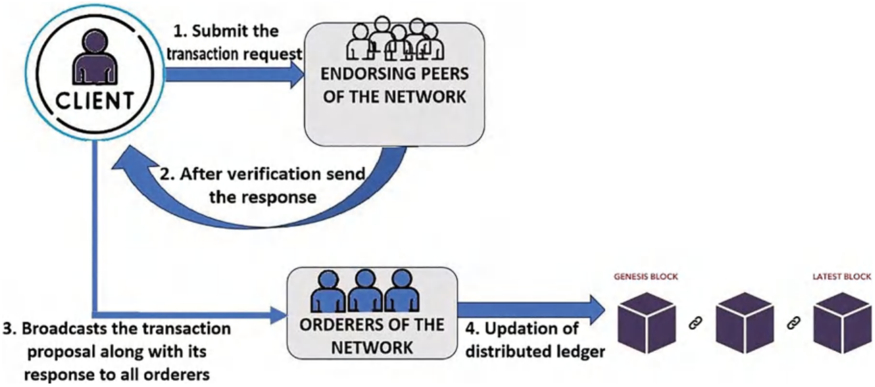

CLIENT
1. Submit the transaction request
ENDORSING PEERS OF THE NETWORK
2. After verification send the response
3. Broadcasts the transaction proposal along with its response to all orderers
ORDERERS OF THE NETWORK
4. Updation of distributed ledger
GENESIS BLOCK
LATEST BLOCK

Fig. 5.7 Flow of transactions in Hyperledger Fabric

2. Execution of chaincode: Every endorser executes the chaincode within the docker container in an isolated manner. Through this execution, they get a set of read–write transactions. The read set constitutes input values that the transaction needs to form the distributed ledger. The write set consists of the outputs after processing the inputs. Transactions update the ledger with this output.

After the simulation of the read–write set, if the endorsers do not find any error, then they send their responses (grant/deny) by attaching their signatures to the respective client. Once the client receives the threshold amount of grant responses, it moves toward the ordering phase of the transaction (Piao et al. 2023). The endorsement flow of the Fabric network is shown in "Fig. 5.8".

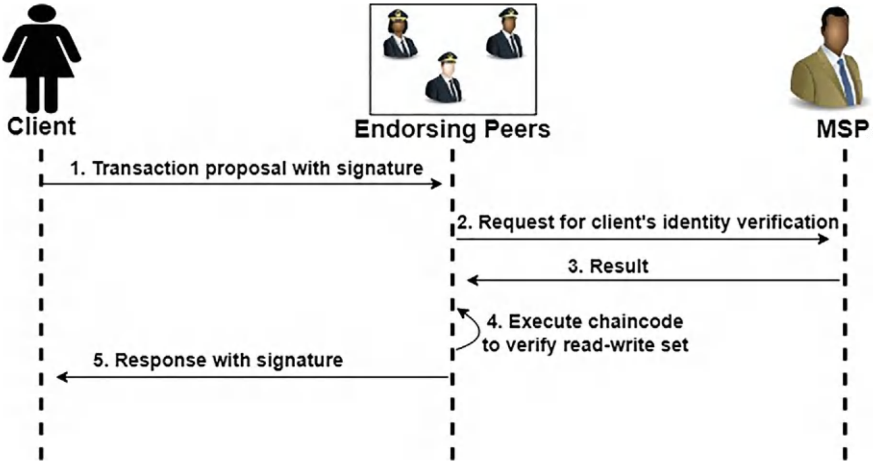

Client
Endorsing Peers
MSP
1. Transaction proposal with signature
2. Request for client's identity verification
3. Result
4. Execute chaincode to verify read-write set
5. Response with signature

Fig. 5.8 Endorsement flow in Hyperledger Fabric

5.6 Hyperledger Fabric Endorsement Policies

Endorsement policies in the Hyperledger Fabric, control validation and endorsement of transactions by the participating peers. Only the transactions proved to be valid, as per the policy, are endorsed ensuring security, integrity, and consistency of the network.

As per the policy, a transaction must be validated and endorsed by a threshold number of peers before it is added to the ledger and for this, the transaction is processed against the smart contract (chaincode) producing a digital signature to indicate that the transaction has been validated by the endorsing peer.

After getting endorsements from a threshold number of peers, as per the policy statement, the transaction is presented to the ordering service for inclusion in a block and ledger update. Peers must verify the endorsements and validate the transaction prior to committing it to the corresponding copies of the local ledgers.

These policies are defined at the time of smart contract instantiation or during transaction proposal and specify the required number and types of endorsements necessary for the validation of a transaction. These policies are generally expressed using logical operators such as “AND”, “OR”, and “M out of N”.

Hyperledger Fabric allows organizations to modify these policies according to their suitability. Endorsement policies generally involve multiple parties or organizations to ensure that transactions are validated by a consensus of authorized entities. This improves reliability preventing single points of failure in a blockchain network. These policies are updated periodically to accommodate changing network requirements or business needs. This flexibility enables organizations to advance their endorsement policies as their business needs evolve without disrupting the network’s operation.

5.7 Hyperledger Fabric Data Distribution Protocol

The data distribution protocol manages how data is disseminated and stored across the network. The data distribution protocol is designed to provide scalability, privacy, and resilience while guaranteeing the integrity and reliability of data stored on the blockchain. By leveraging channels, private data collections, and distributed ledger technology, Fabric enables secure and efficient data exchange across enterprise blockchain networks. Some of the key aspects of the data distribution protocol in Fabric are:

i. Private Data Collection: Fabric supports the concept of private data collection. This allows the confidential exchange of data between specified parties. When a transaction involves private data, it is shared only with the designated participants, ensuring privacy and confidentiality.

ii. Channels: Fabric implements a channel-based architecture, where transactions are confined to specific channels shared among a subset of network participants. Each channel has its ledger, maintaining a separate history of transactions, and employs its endorsement policies and access controls.
iii. State and History Database (DB): Fabric separates the ledger into two distinct components: the state DB and the history DB. State DB maintains the current state of the ledger, representing the newest of all data whereas, history DB maintains the complete history of all transactions, enabling auditability and traceability.
iv. Distributed Ledger: The ledger in Fabric is distributed across all peers, each peer preserving a copy of it. Consensus algorithms ensure that all peers have a consistent view of the ledger, preventing tampering or unauthorized modifications.
v. Endorsement and Ordering: Before transactions are added to a block and saved in the ledger, they must be endorsed by a sufficient number of endorsing peers and ordered by the ordering service. This ensures that transactions are valid and agreed upon by the network before they are permanently recorded.
vi. CouchDB as State Database: Hyperledger Fabric supports CouchDB as the state database, providing a flexible and efficient storage solution for rich query capabilities and complex data models. CouchDB enables the storage of JSON documents, allowing for the representation of diverse data structures.
vii. Data Distribution and Replication: Fabric employs a Gossip protocol for data distribution and peer-to-peer communication. Peers exchange blocks and transactions with their neighboring peers in a decentralized manner, ensuring high availability and resilience against network failures.

5.8 Hyperledger Fabric Chaincode

Chaincode is a container of correlated smart contracts and can contain multiple smart contacts. Smart contracts are the specifications of the transactions defined on a blockchain network and are coded in the chaincode. The specifications include business rules, workflow logic, etc. The business experts of member organization group approve the smart contracts. This self-executing contract implements business logic for the creation and modification of logical assets in the ledger. It runs in an isolated and secure docker container. Applications submit transactions to initialize and manage the ledger states. Fabric supports the development of chaincode in three programming languages like Go, Java, and Node.

Chaincodes run as a separate process within docker container. The container isolates chaincodes from each other to avoid collision. When a chaincode is installed on a network, the smart contracts defined in that chaincode become accessible to the participating organizations of that network. Smart contracts of a chaincode can access all endorsement policies of the chaincode. Smart contracts run on a peer node owned by a member organization and the endorsement policy defines, a transaction produced by the smart, needs to be endorsed by how many participating organizations to be considered authentic. The peers can communicate with chaincode using remote procedure calls (gRPC) messages. The chaincode is invoked during the Execute phase (the first phase of Fabric transaction flow). Fabric can be customized by two things such as channel configuration and special chaincodes or system chaincodes (Androulaki et al. 2018).

5.8.1 Structure of Chaincode

The channel members of a Fabric network follow the following four phases of chaincode life cycle such as development of the chaincode, installing it on peers, approving it for an organization, and committing the approved chaincode to the channel.

1. Chaincode Development: The chaincode development phase involves the creation of business transaction specifications, coding, and packaging of the chaincode. The chaincode is packed in a tar file (extension `tar.gz`) and installed on peers. To generate the tar file `Package` command is used. The tar file has two parts such as chaincode (a `code.tar.gz` file which is also a tar file containing the source code for the chaincode), and metadata (`metadata.json`) file which contains the parameter and configuration information for the package chaincode). Chaincode packages are uniquely identified by label mechanism. The syntax of packaging the chaincode is as follows:

```
Peer lifecycle chaincode package package_name -p
path_to_the_chaincode --label chaincode_label
```

After packaging the chaincode the developer organization put the package in a shared repository. Member organizations that have decided to use the shared code can access the repository.

2. Chaincode installation: Admins of different organizations install packaged chaincode on their respective peers by using the `install` command. Successful installation of the package leads to the creation of a package ID, which is then used for the approval of the chaincode. By executing the `queryinstalled` command, the admins of the respective organization can check the list of packages installed on their peers. The syntax of the `install` and `queryinstalled` commands are as given below:

```
Peer lifecycle chaincode install package_name
Peer lifecycle chaincode queryinstalled
```

All organizations belonging to a channel need to run the same chaincode.

3. Approving a chaincode for an organization: Before using a chaincode on a channel, the chaincode definition needs to be approved by channel members.

This is carried out by the admin of different organizations by executing the `approveformyorg` command. The syntax of `approveformyorg` command is:

```
Peer lifecycle chaincode approveformyorg -n
name_of_chaincode -v version_of_chaincode -c channel-id -
sequence sequence_no.(default=1) --init-required --package-
id id_of_package
```

After approval, the approved chaincode package is then installed on peers of the respective organization. The chaincode definition includes a set of parameters that specify how the chaincode operates within the blockchain network. Here are some major parameters of a chaincode definition:

i. Name: The name of the chaincode that the applications will use to invoke it.
ii. Version: The chaincode version is given with the chaincode package. With each upgrade of chaincode binaries, the chaincode version changes.
iii. Sequence: The sequence number is an integer that tells how many times the chaincode is upgraded.
iv. Endorsement Policy: This can be a policy referred to in the channel configuration or as a string that can be passed to the Command Line Interface (CLI). By default, it can be set as:

```
Channel/Application/Endorsement
```

4. Committing chaincode to channel: A committed transaction from an admin of any member organization is submitted for verification and endorsement. If the endorsement rules are satisfied, then the commit is successful otherwise it fails. To commit a chaincode, "`commit`" command is used and the syntax of this is:

```
Peer lifecycle chaincode commit -n name_of_chaincode -v
version_of_chaincode -c channel-id -sequence
sequence_no.(default=1) --init-required
```

Whenever the commit transaction gets successful after that only a chaincode can invoked or queried. The command to check the approvals from various organizations the `checkcommitreadiness` command is used. The syntax to execute the `checkcommitreadiness` command is as follows:

```
Peer lifecycle chaincode checkcommitreadiness -n
name_of_chaincode -v version_of_chaincode -c channel-id -
sequence sequence_no.(default=1) --init-required
```

To check a specific chaincode is committed or not, the `querycommitted.` command is used and the syntax is as below:

```
Peer lifecycle chaincode querycommitted -n name_of_chaincode
-v version_of_chaincode -c channel-id
```

The structure of organizations associated with the same channel have the same chaincode as shown in "Fig. 5.9".

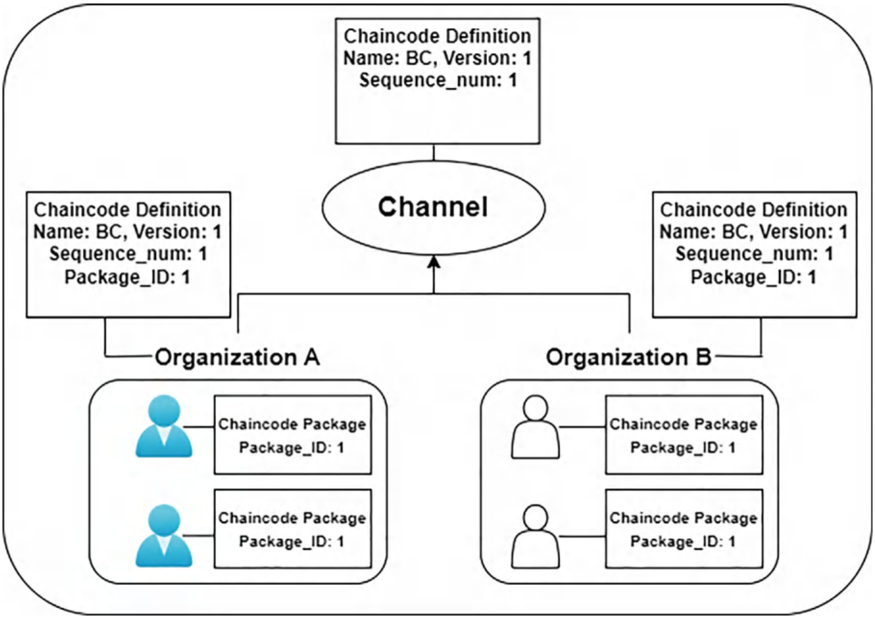


Fig. 5.9 Structure of organizations associated with the same channel have the same chaincode

5.8.2 Writing of a Sample Chaincode

There are three basic peer chaincode commands used to initialize, query, and invoke a chaincode. The initialization command is used to initialize the chaincode for which the input parameters such as chaincode name, channel name, and initialization argument are needed. The command is as given below:

```
peer chaincode invoke --isInit -n name_of_the chaincode -
C channel_name -c initialization_argument
```

The query command reads the states from the local peer and there is no change in the state due to the invocation of this query command. To run this query command the channel name, chaincode name, and query argument are required. The command is as given below:

```
peer chaincode query -C channel_name -n
name_of_the_chaincode -c query_argument
```

The invoke command is used to initiate a transaction. It leads to the change of state and is recorded in the ledger. The invoke command needs some input parameters such as channel name, chaincode name, and transaction argument as below:

```
peer chaincode invoke -C channel_name -n
name_of_the_chaincode-c transaction_arguments
```

If there are two peers, A and B with an initial balance of 500, 400, respectively, then to perform one transaction of 100 rupees from A to B, the chaincode is as follows: (the channel name is channel1 and chaincode name is BC.)

```
peer chaincode invoke --isInit -n BC -C channel1 -c
'{"Args": ["init", "A", 500, "B", 400]}'
peer chaincode invoke -C channel1 -n BC -c '{"Args":
["invoke", "A", "B", 100]}'
peer chaincode query -C channel1 -n BC -c '{"Args":
["query", "A"]}'
```

5.9 Hyperledger Fabric CA

Before accessing the Fabric network, all the entities need to be authenticated. For this purpose, digital certificates are issued to the users. The certificates are not only used as the identity of the user but also it has the information about the role assigned to the specific user which determines the privileges the user has. The certificates are generated at the time of crypto material generation. The digital certificates are issued, validated, and revoked by the Fabric CA. The implementation of Fabric CA has two parts such as Fabric CA server and Fabric CA client (Honar Pajooh et al. 2021). Fabric allows its member organizations to deploy their own certificate authorities to manage the identities and certificates within the organization.

Listed below are the important functions of the CA:

1. Identity Registration: The CA registers participants within the network by issuing unique digital identities. These identities are typically in the form of cryptographic key pairs (public and private keys), along with metadata such as user roles and affiliations.
2. Issue of Certificate: Once registered, the CA issues digital certificates to the participants. These certificates contain the participant’s public key and other relevant identity information, digitally signed by the CA. Certificates are used for verification and encryption purposes.
3. Revocation of Certification: In case a participant’s private key is compromised or the participant no longer has permission to access the network, the CA can revoke their certificate. This ensures that compromised or unauthorized entities cannot participate in network transactions.
4. Certificate Renewal: Certificates have a finite lifespan. The CA manages the renewal process, issuing new certificates to participants before the old ones expire, thus ensuring uninterrupted access to the network.
5. Enrollment: This process typically involves proving the identity of the entity requesting enrollment and obtaining the necessary cryptographic materials securely. Through this process, the participants obtain their cryptographic constituents such as keys and certificates from the CA.

6. Authentication: During network interactions, participants use their digital certificates to authenticate themselves to other network components, such as peers or ordering services. This guarantees the participation of only authorized participants in transaction processing.

5.10 Hyperledger Fabric MSP

The Fabric certificate authorities are in charge of issuing identities and the MSP is in charge of verification, and maintenance of identities, roles, and permissions of participants within the Fabric network (Androulaki et al. 2018). Based on their scope and level of administration, a MSP can be a local or a channel MSP.

1. Local MSP: Local MSPs are accountable for managing the identities and roles of nodes (peers and orderers), and clients. The local MSP of a node defines the permissions a node has and the local MSP of a client allows for authentication of itself when it requests for a transaction.
2. Channel MSP: Channel MSPs manage the identities, roles, and permissions at the channel level within the Fabric network. This MSP is a part of the channel configuration and is distributed to all participants of that channel. This ensures that all nodes have the same rules and information for verification of the identities of participants. The channel configuration is updated with the MSP of that organization, every time a new organization joins it.

The structure of the coexistence of local and channel MSPs in the Fabric network is shown in “Fig. 5.10”.

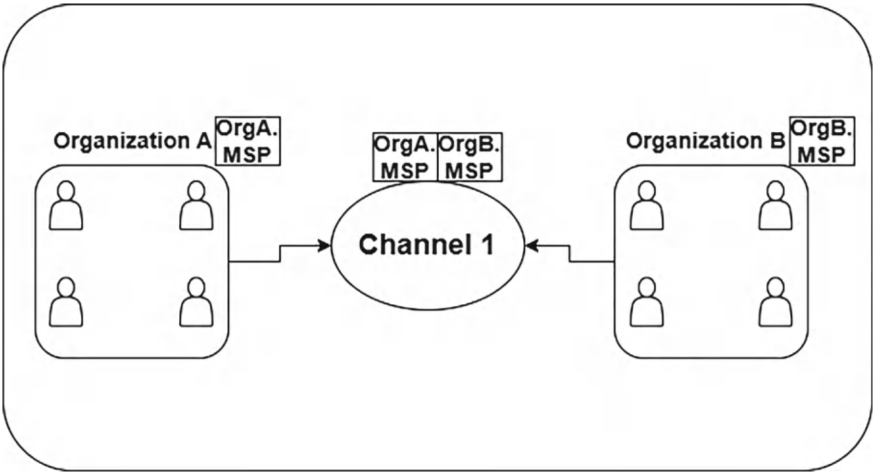


Fig. 5.10 Local and channel MSPs in the Fabric network

In an Hyperledger Fabric network, each channel has its own MSP that manages the channel-specific identity and access control policies that enable the creation of private communication channels between specific participants while ensuring data privacy and integrity.

The MSP in an Hyperledger Fabric is responsible for managing identities, issuing, and revoking certificates, enforcing access control policies. It follows a pluggable design that allows organizations to seamlessly integrate their existing identity management systems while sticking to their preferred authentication mechanisms and regulatory requirements.

Identity Management: The MSP manages the identities of network participants, including organizations, users, and client applications. It assigns a unique identifier to each participant and ensures that identities are properly authenticated and authorized within the network.

Certificate Issue: One of the primary functions of the MSP is to issue digital certificates to network participants. These certificates contain the participant's public key and other identity-related information, digitally signed by the CA.

Certificate Revocation: The MSP can revoke certificates in case a participant's private key is compromised or if any participant no longer has permission to access the network. Revocation ensures that compromised or unauthorized entities cannot participate in network transactions.

Access Control Policies: MSP enforces access control policies to govern which participants are allowed to perform specific actions within the network. It defines roles, permissions, and other security policies to regulate interactions between network entities and ensure data confidentiality, integrity, and availability.

Pluggable Design: Hyperledger Fabric's MSP architecture is designed to be pluggable, allowing organizations to integrate their existing identity management systems seamlessly. This flexibility enables organizations to leverage their preferred authentication mechanisms and comply with regulatory requirements.

5.11 Create a Basic Hyperledger Fabric Network

Building of a Hyperledger Fabric framework constitutes two phases such as network architecture and an application layer.

1. Network architecture: A Fabric network consists of one or more channels that connect two or more organizations, member organizations, each member organization has peer and orderer nodes. The Fabric channel configuration is defined by two important parameters such as batch-timeout, and batch size.

    i. Batch-timeout: This refers to the time elapsed in waiting before the creation of a block.

ii. Batch size: This defines the number of transactions per block (tbp). The parameters used in this definition are: max trans count (the maximum number of tpb), absolute max bytes (maximum number of bytes per block), and preferred max bytes (chosen number of bytes per block).

Selection of the above parameters plays an important role in determining the performance of the Fabric network (Shalaby et al. 2020).

2. Application layer: The Hyperledger composer is used to implement the Fabric application layer. Composer helps in modeling the business network which is packaged in an archive file (.bna). A business network is defined using, model (which defines all assets, participants, and transactions), script (nothing but the smart contract file which implements the transaction logic), and access control file (access control rules are defined here in the form of CRUD: create, read, update, and delete) (Shalaby et al. 2020).

The building of a Fabric network follows five steps:

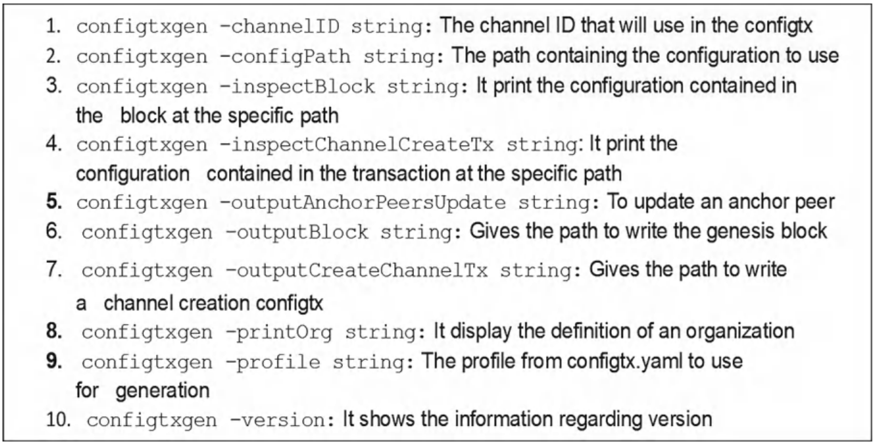
I. Installation of Pre-requisites: Some of the pre-requisites need to be installed to build a Fabric network such as

   (a) cURL (It sets up the environment to run the Fabric test network)
   (b) Docker and Docker compose (After installation the version of these two can be checked by running the commands such as `docker -- version`, and `docker-compose --version`)
   (c) Go programming language (require to write Fabric chaincode)
   (d) Fabric samples, and Binaries.

II. Crypto Generator: To generate cryptographic tools such as certificates and signing keys, cryptogen tool is used. To ensure that only authenticated entities are communicating, the identity certificates are issued. The cryptogen file (crypto-config.yaml) contains the network topology and allows the developer to generate certificates and keys for the participant member organizations and their peers. Each member organization has a CA. The cryptogen tool is also responsible for generating certificates for the orderers. All the identity certificates and keys are saved to a folder crypto- config.

III. Configuration Transaction Generator: The configtxgen tool is used to create the channel configuration artifacts such as orderer, channel, and anchor peer. The configtxgen (configtx.yaml) file contains network definitions. Let's suppose a channel C1 consists of two member organizations (G1 and G2), and these two are maintaining two peer nodes. Initially, the certificates/keys are generated for the associate organizations (G1, and G2) of C1 and their peer nodes. The usages of configtxgen are shown in "Fig. 5.11".

   (a) Genesis block creation: It is the configuration block that initializes the ordering service, or serves as the first block on a chain. To run the genesis block creation command it needs the channel and profile name. The syntax is as below:



1. `configtxgen -channelID string`: The channel ID that will use in the configtx
2. `configtxgen -configPath string`: The path containing the configuration to use
3. `configtxgen -inspectBlock string`: It print the configuration contained in the block at the specific path
4. `configtxgen -inspectChannelCreateTx string`: It print the configuration contained in the transaction at the specific path
5. `configtxgen -outputAnchorPeersUpdate string`: To update an anchor peer
6. `configtxgen -outputBlock string`: Gives the path to write the genesis block
7. `configtxgen -outputCreateChannelTx string`: Gives the path to write a channel creation configtx
8. `configtxgen -printOrg string`: It display the definition of an organization
9. `configtxgen -profile string`: The profile from configtx.yaml to use for generation
10. `configtxgen -version`: It shows the information regarding version

Fig. 5.11 Usages of configtxgen tool

```
configtxgen -outputBlock genesis_block_name -profile
profile_name -channelID channel_ID
```

To print the contents of the created genesis: `configtxgen - inspectBlock genesis_block_name.`

(b) Channel creation transaction: The syntax of channel creation transaction is as follows:

```
configtxgen -outputCreateChannelTx channel_tx_name -
profile profile_name -channelID channel_ID
```

To print the contents of the channel creation transaction: `configtxgen.`

```
-inspectChannelCreateTx channel_tx_name
```

(c) Anchor peer updation transaction: To update the anchor peer list of a specific organization the following syntax is used:

```
configtxgen -outputAnchorPeersUpdate
file_name -profile profile_name -asOrg
organization_name
```

To print the definition of an organization: `configtxgen.`

```
-printOrg organization_name
```

IV. Channel Transaction Artifacts: After the configuration of a channel next step is the creation of channel transaction artifacts. The channel transaction artifact contains the definition of a channel. The command is as mentioned below:

```
export CHANNEL\_NAME= channel_name && ../bin/configtxgen -
profile profile_name -outputCreateChannelTx./channel-
artifacts/ channel.tx -channelID channel_ID
```

After the successful creation of the channel transaction artifact, the next step is to define the anchor peers for the member organizations (let G1 and G2). The command is as below:

```
../bin/configtxgen -profile profile_name -
outputAnchorPeersUpdate ./channel-artifacts/
organization_name anchors.tx -channelID channel_name -asOrg
organization_name
```

iii. Start the Network: The network to be started running the command given below.

```
docker-compose -f docker-compose-cli.yaml up -d
```

5.12 Conclusion and Future Scope

Blockchain technology has changed the way of sharing and managing data from centralized to decentralized. To solve the low transaction throughput, long transaction time, and privacy issues with public blockchain the Hyperledger Fabric has been introduced. Which follows the modular blockchain architecture standard for enterprise blockchain platforms. Four aspects of Hyperledger Fabric that make it the most preferable one for business applications are permissioned networks, confidential transactions, no cryptocurrency, and programmability. To promote the Hyperledger Fabric-based business applications, this chapter discusses the various aspects of Hyperledger Fabric such as architecture, components, transaction flow, endorsement flow, endorsement policies, data distribution protocol, chaincode, CA, MSP, and step by step procedure to create a Fabric network. There are many application areas of Hyperledger Fabric such as health care, food supply chain, identity management, real estate, and decentralized cloud data storage. The pharmaceutical industry is an important application area of Hyperledger Fabric blockchain. Where counterfeit drugs and drug journey tracing are severe problems. A Fabric-based secure drug supply chain system can solve the traceability and counterfeit drugs problems in pharmaceutical industries.

References

Aggarwal S, Kumar N (2021) Hyperledger. In: Advances in computers, vol 121. Elsevier, pp 323–343

Androulaki E, Barger A, Bortnikov V, Cachin C, Christidis K, De Caro A, Enyeart D, Ferris C, Laventman G, Manevich Y, et al (2018) Hyperledger fabric: a distributed operating system for permissioned blockchains. In: Proceedings of the thirteenth EuroSys conference, pp 1–15

Baset SA, Desrosiers L, Gaur N, Novotny P, O'Dowd A, Ramakrishna V (2018) Hands-on blockchain with Hyperledger: building decentralized applications with Hyperledger Fabric and composer. Packt Publishing Ltd

Dabbagh M, Sookhak M, Safa NS (2019) The evolution of blockchain: a biblio- metric study. IEEE Access 7:19212–19221

Dhillon V, Metcalf D, Hooper M (2017) The hyperledger project. In: Blockchain enabled applications: understand the Blockchain ecosystem and how to make it work for you, pp 139–149

Foschini L, Gavagna A, Martuscelli G, Montanari R (2020) Hyperledger fabric blockchain: chaincode performance analysis. In: ICC 2020-2020 IEEE international conference on communications (ICC). IEEE, pp 1–6

Gaba P, Raw RS, Mohammed MA, Nedoma J, Martinek R (2022) Impact of block data components on the performance of blockchain-based vanet implemented on hyperledger fabric. IEEE Access 10:71003–71018

Gaur N, O'Dowd A, Novotny P, Desrosiers L, Ramakrishna V, Baset SA (2020) Blockchain with hyperledger fabric: build decentralized applications using hyper- ledger fabric 2. Packt Publishing Ltd

Honar Pajooh H, Rashid M, Alam F, Demidenko S (2021) Hyperledger fabric blockchain for securing the edge internet of things. Sensors 21(2):359

Khettry R, Patil KR, Basavaraju AC (2021) A detailed review on blockchain and its applications. SN Comput Sci 2(1):30

Kushwaha SS, Joshi S, Singh D, Kaur M, Lee H-N (2022) Systematic review of security vulnerabilities in ethereum blockchain smart contract. IEEE Access 10:6605–6621

Kwon M, Yu H (2019) Performance improvement of ordering and endorsement phase in hyperledger fabric. In: 2019 sixth international conference on internet of things: systems, management and security (IOTSMS). IEEE, pp 428–432

Leng J, Zhou M, Zhao JL, Huang Y, Bian Y (2020) Blockchain security: a survey of techniques and research directions. IEEE Trans Serv Comput 15(4):2490–2510

Li D, Wong WE, Guo J (2020) A survey on blockchain for enterprise using hyperledger fabric and composer. In: 2019 6th International conference on dependable systems and their applications (DSA). IEEE, pp 71–80

Monrat AA, Schelén O, Andersson K (2019) A survey of blockchain from the perspectives of applications, challenges, and opportunities. IEEE Access 7: 117134–117151

Nguyen TSL, Jourjon G, Potop-Butucaru M, Thai KL (2019) Impact of network delays on hyperledger fabric. In: IEEE INFOCOM 2019-IEEE conference on computer communications workshops (INFOCOM WKSHPS). IEEE, pp 222–227

Piao X, Ding H, Song H (2023) Performance analysis of endorsement in hyperledger fabric concerning endorsement policies. Electronics 12(20):4322

Sah S, Surendiran B, Dhanalakshmi R, Arulmurugaselvi N (2021) A survey on hyperledger frameworks, tools, and applications. In: Internet of things, artificial intelligence and blockchain technology. Springer, pp 25–43

Sarmah SS (2018) Understanding blockchain technology. Comput Sci Eng 8(2):23–29

Satapathy U, Mohanta BK, Panda SS, Sobhanayak S, Jena D (2019) A secure framework for communication in internet of things application using Hyperledger based blockchain. In: 2019 10th international conference on computing, communication and networking technologies (ICCCNT). IEEE, pp 1–7

Shalaby S, Abdellatif AA, Al-Ali A, Mohamed A, Erbad A, Guizani M (2020) Performance evaluation of hyperledger fabric. In: 2020 IEEE international conference on informatics, IoT, and enabling technologies (ICIoT). IEEE, pp 608–613

Touloupou M, Themistocleous M, Iosif E, Christodoulou K (2022) A systematic literature review toward a blockchain benchmarking framework. IEEE Access 10:70630–70644

Chapter 6
Private and Consortium Blockchain

Afeefa Noorain, Khaleel Ahmad, and Laura Emilia Maria Ricci

Abstract Blockchain is an implementation of Distributed Ledger Technology. It is categorized as a public, private, and consortium blockchain. Blockchain is public and permissionless by default as it supports transparency, immutability, and security. Private blockchains are permissioned blockchain networks with a restricted number of nodes, governed by an organization or its entities. Consortium blockchain is a coalesced permissioned network that allows multiple organizations to collaborate where predetermined nodes will govern the network. Private and consortium blockchain can be created and deployed with the help of Hyperledger Fabric. Hyperledger Fabric offers pluggable modules for consensus, identity management, and privacy. The chaincode functions as a smart contract that defines the business logic and is executed on the ledger after achieving the consensus. It has different tools and libraries for developing, testing, and deploying enterprise-grade applications.

Keywords Blockchain · Chaincode · Hyperledger fabric · Private · Permissioned · Consortium · Configuration · Performance · Visualization

Introduction

Blockchain is a decentralized peer-to-peer network with a disseminated ledger that maintains a record of transactions committed on the database. It allows append-only

A. Noorain (✉) · K. Ahmad
Department of Computer Science and Information Technology, Maulana Azad National Urdu University, Hyderabad, India
e-mail: afeefaafn@gmail.com

K. Ahmad
e-mail: khaleelahmad@manuu.edu.in

A. Noorain
Department of Computer Science, St. Francis College for Women, Hyderabad, India

L. E. M. Ricci
Department of Computer Science, University of Pisa, Pisa, Italy
e-mail: laura.ricci@unipi.it

K. Ahmad et al. (eds.), *Fostering Machine Learning and IoT for Blockchain Technology*,
Transactions on Computer Systems and Networks,
https://doi.org/10.1007/978-981-96-4078-2_6

transactions that ensure immutability. Transactions are submitted in the mempool, and miners compete to select and validate the transactions with the maximum reward for performing mining that commits the transactions in the block. The mining is done by calculating the hash value that generates the nonce value for all the transactions to be committed in the block. Once the block is added to the blockchain the copy of the ledger is shared with all nodes in the network. Because of the decentralization characteristic, blockchain is being adopted in the development of different industry applications (Zhu et al. 2023; Liang et al. 2019; Francesco Maesa and Mori 2020; Javaid et al. 2021; Verma et al. 2022).

Private and consortium blockchains are suitable for most enterprise-grade applications. Enterprise-grade application is a software product designed for large organizations to address diverse requirements of all processes that operate or govern the organizations. Supply Chain Management (Ghode et al. 2020), Healthcare (Khezr et al. 2019), and Human resources management (Salah et al. 2020) are a few examples of blockchain applications. Several studies have been carried out to analyze the performance of the private blockchain (Pongnumkul et al. 2017; Geyer et al. 2019; Al-Sumaidaee et al. 2023) and have shown remarkable performance in terms of data accessibility and privacy.

Hyperledger Fabric (Al-Sumaidaee et al. 2023) platform provides support for building a permissioned blockchain network. In these blockchains, the nodes belong to one or different organizations depending on the type of blockchain i.e., private or consortium. They perform the task based on the identity and privileges assigned to them. When a transaction is submitted it should be verified and endorsed by the endorsing peer (node). This endorsing peer performs the verification employing a copy of the current ledger. If the transaction is legitimate then the transaction is distributed via the ordering peers (nodes) to the nodes that can commit the transaction in the block. Once committed the transaction cannot be reversed. Nodes in this network hold a copy of the ledger and take different responsibilities in the network to initiate, endorse, and commit the transactions. This defines the concept of permissioned in blockchains.

Hyperledger was formed by the Linux Foundation in 2015 collaborating with different companies and organizations interested in blockchain technology to develop open-source blockchain technology easily available for researchers and organizations to build their solutions(Li et al. 2019). As part of the collaborative initiative, Fig. 6.1 shows Hyperledger Greenhouse which brings all the developers, organizations, and vendors to actively participate in developing the enterprise blockchain network. Hyperledger encourages the development of several Blockchain tools that facilitate enterprise-grade application development. Hyperledger Explorer and Caliper benchmark are tools designed for visualizing and analyzing performance, respectively (Li et al. 2019).

The following sections demonstrate the creation of private and consortium blockchains. These chains can be visualized with the help of Hyperledger Fabric Explorer and performance analysis using the Caliper tool.

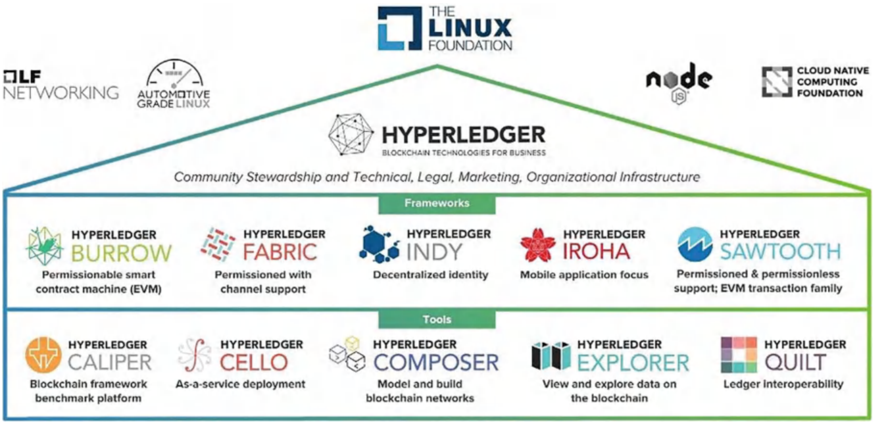

THE LINUX FOUNDATION
LF NETWORKING
AUTOMOTIVE GRADE LINUX
node JS
CLOUD NATIVE COMPUTING FOUNDATION
HYPERLEDGER
BLOCKCHAIN TECHNOLOGIES FOR BUSINESS
Community Stewardship and Technical, Legal, Marketing, Organizational Infrastructure
Frameworks
HYPERLEDGER BURROW
Permissionable smart contract machine (EVM)
HYPERLEDGER FABRIC
Permissioned with channel support
HYPERLEDGER INDY
Decentralized identity
HYPERLEDGER IROHA
Mobile application focus
HYPERLEDGER SAWTOOTH
Permissioned & permissionless support; EVM transaction family
Tools
HYPERLEDGER CALIPER
Blockchain framework benchmark platform
HYPERLEDGER CELLO
As-a-service deployment
HYPERLEDGER COMPOSER
Model and build blockchain networks
HYPERLEDGER EXPLORER
View and explore data on the blockchain
HYPERLEDGER QUILT
Ledger interoperability

Fig. 6.1 The hyperledger greenhouse structure (Li et al. 2019)

6.1 Create a Private Blockchain Network Using Hyperledger Fabric

A private blockchain is a centralized network conventionally managed by a single organization. A single organization entirely controls this blockchain and the authorized nodes of this organization have permission to be a part of the network (About Hyperledger n.d.). This ensures the privacy and security of the data shared in the network. This network is described as a centralized network because of its permissioned characteristic. As shown in Fig. 6.2 the authorized nodes of an organization are identified as peers in the blockchain network that perform various operations. Every member is allotted an identity which is an x.509 certificate issued by a trusted Certificate Authority. These identities define the privileges and pair of public and private keys. Privileges define a member's access rights and operations in the network. The Membership Service Provider (MSP) plays the role of an authenticator who verifies the identity and permits access according to the assigned role.

A channel (Khettry et al. 2021) is created for communication between the peers and applications to ensure secure data transfer among the members. The channel promotes transparency (Surjandari et al. 2021) among the members by allowing all the members to maintain the same ledger copy. These ledgers are updated by a specific peer called an Orderer. Orderer's task is to keep the ledgers consistent and updated. The ledger stores the factual information about the business objects. It is an assemblage of two components: the current state of an object and a blockchain. The blockchain contains the transactional logs that lead to the current state. The blockchain is immutable (Ravi et al. 2022) as every block in the blockchain is linked with the hash of the previous block which is committed to the chain with a set of legitimate transactions. Any new transaction request that does not validate with the current state of the ledger will be rejected and only valid transactions are processed.

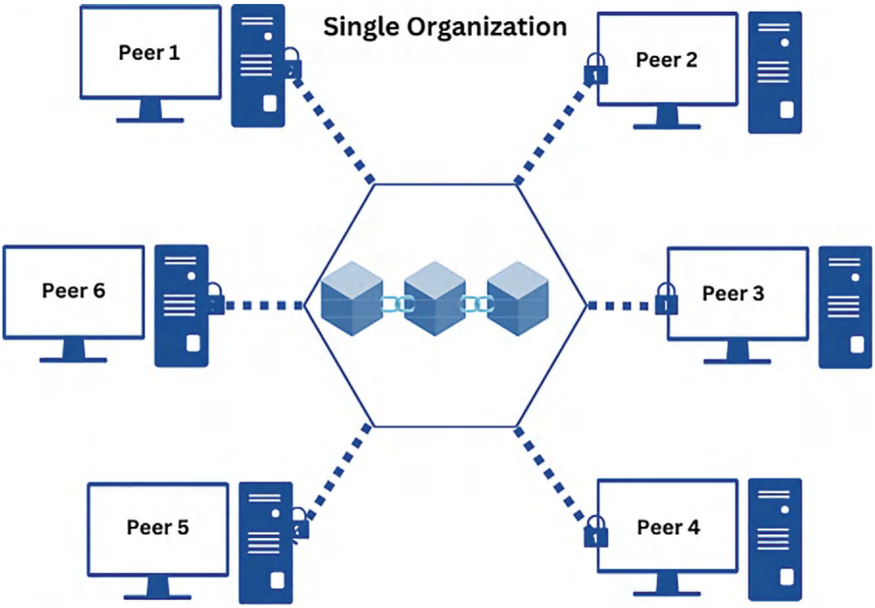


Fig. 6.2 Private blockchain for a single organization

6.1.1 Prerequisites for Creating a Blockchain

This section describes the requirements for building a blockchain on Hyperledger Fabric. Hyperledger Fabric is platform-independent and can be installed on Windows, Mac OS, or Linux. To install the Hyperledger Fabric, it is best to set up a Linux-based virtual machine. Hyperledger Fabric blockchain can also be set up on AWS a web-based console for creating and deploying the network. This section focuses on the creation of blockchain using a Linux-based virtual machine.

The prerequisites that need to be installed on the machine are:

- Nodejs
- NPM
- Docker
- Docker Compose
- Go Language
- Hyperledger Fabric binaries.

Node.js and NPM. Node.js and NPM offer support to design applications that can interface with the Fabric blockchain network.

Docker and Docker Compose. Docker is an open-source project that accelerates the development and deployment of the Fabric network as it containerizes code, dependencies, and configurations. Docker Compose allows configuring the Fabric network artifacts to be deployed on the Docker.

The Fabric binaries are the CLI tools that allow interaction with the sample test network and help in configuring the environment for the intended Fabric Network. These samples will have chaincode written in Go and JavaScript and installing Go language packages is favored. A complete installation guide is available in Sutradhar et al. (2023).

These Fabric samples contain the configuration files written in YAML and can be easily edited as they are a set of key-value pairs. Docker compose configuration is written in YAML as it is strongly recommended for Docker. All these configurations can be modified to adapt to the blockchain network being designed.

6.1.2 Network Setup

The network components recognized in Hyperledger Fabric for a private Blockchain are:

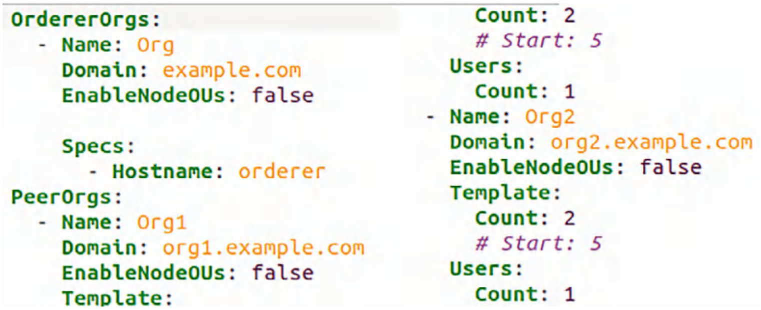
1. An organization
2. A set of peers
3. Certificate Authority to authorize
4. At least one channel for communication
5. an Orderer to maintain the channel.

To begin the creation of a private blockchain, Fabric sample binaries can be downloaded or a new one can be built. The above-listed components are to be up dated in the configuration files of the Fabric.

The configuration files crypto-config.yaml as shown in Fig. 6.3 and configtx.yaml shown in Fig. 6.4 facilitates the generation of the channel artifacts for the network. Once these artifacts are generated activate the Fabric network using the docker-images which can be containerized using the docker-compose.yaml as shown in Fig. 6.5. On this Fabric network, create a channel for the peers to join and deploy the chaincode (Hyperledger-fabricdocs Documentation Release master hyperledger 2021). These chaincodes can be defined in Go, JavaScript, and others. Chaincode helps design the business logic which all the members of the channel should approve. The following section shows the deployment of the Hyperledger Fabric network for consortium blockchain with figures.

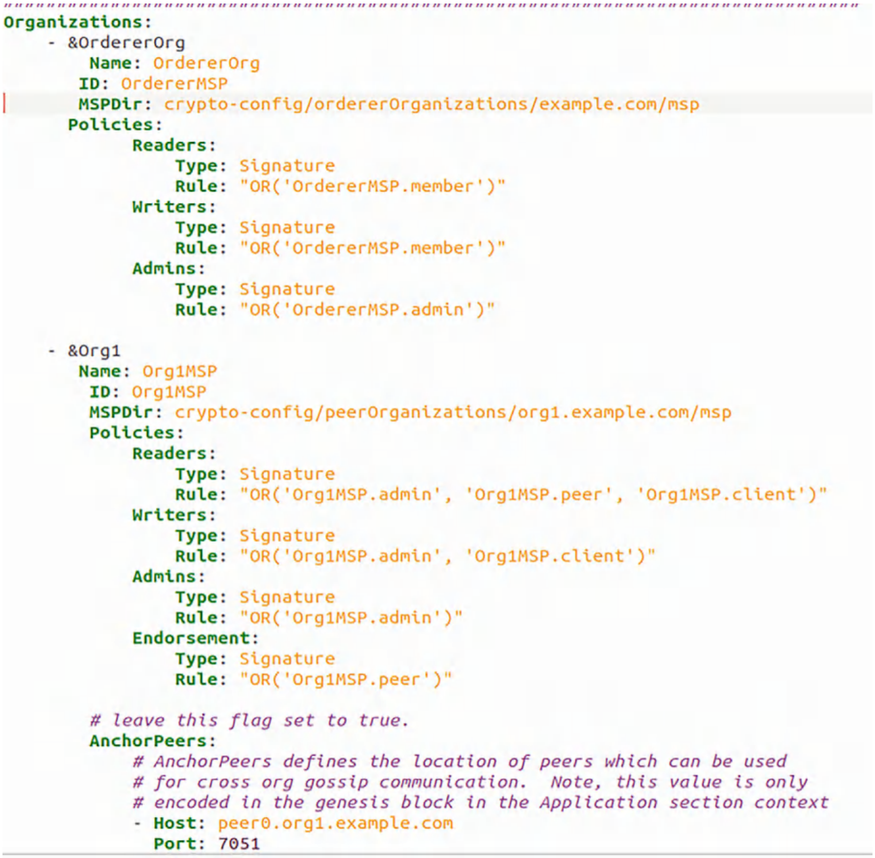
6.2 Create Consortium Blockchain Network Using Hyperledger Fabric

An enterprise-grade application is not always confined to a single organization. It encompasses different entities, collaborators, vendors, and organizations. In this context, a private blockchain requires scaling to include other organizations. For instance, in the health care industry, all medical records owned by a patient should



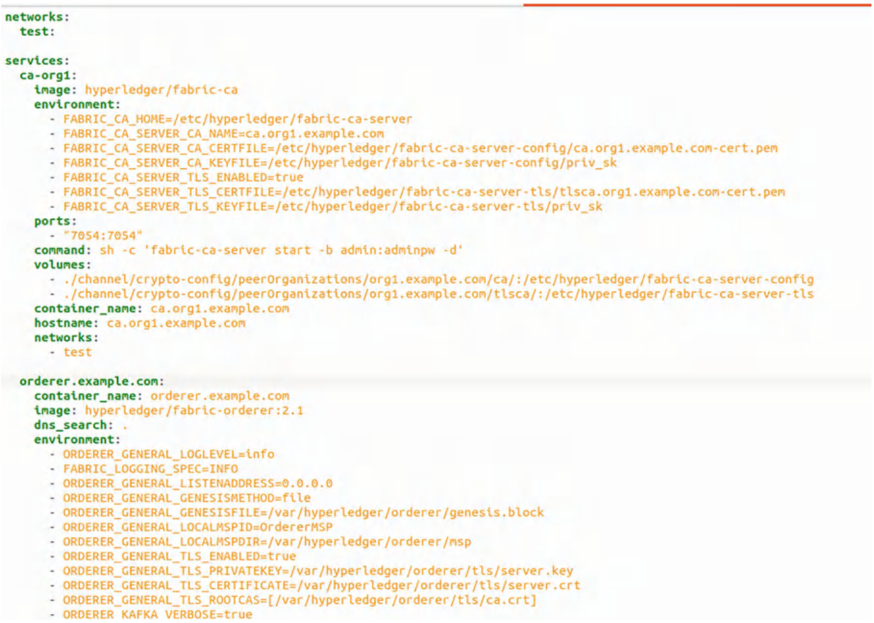
```
OrdererOrgs:
  - Name: Org
    Domain: example.com
    EnableNodeOUs: false

    Specs:
      - Hostname: orderer
PeerOrgs:
  - Name: Org1
    Domain: org1.example.com
    EnableNodeOUs: false
    Template:
      Count: 2
      # Start: 5
    Users:
      Count: 1
  - Name: Org2
    Domain: org2.example.com
    EnableNodeOUs: false
    Template:
      Count: 2
      # Start: 5
    Users:
      Count: 1
```

Fig. 6.3 Code snippet from crypto-config.yaml file



```
Organizations:
    - &OrdererOrg
        Name: OrdererOrg
       ID: OrdererMSP
       MSPDir: crypto-config/ordererOrganizations/example.com/msp
      Policies:
            Readers:
                Type: Signature
                Rule: "OR('OrdererMSP.member')"
            Writers:
                Type: Signature
                Rule: "OR('OrdererMSP.member')"
            Admins:
                Type: Signature
                Rule: "OR('OrdererMSP.admin')"

    - &Org1
       Name: Org1MSP
        ID: Org1MSP
        MSPDir: crypto-config/peerOrganizations/org1.example.com/msp
        Policies:
            Readers:
                Type: Signature
                Rule: "OR('Org1MSP.admin', 'Org1MSP.peer', 'Org1MSP.client')"
            Writers:
                Type: Signature
                Rule: "OR('Org1MSP.admin', 'Org1MSP.client')"
            Admins:
                Type: Signature
                Rule: "OR('Org1MSP.admin')"
            Endorsement:
                Type: Signature
                Rule: "OR('Org1MSP.peer')"

        # leave this flag set to true.
        AnchorPeers:
            # AnchorPeers defines the location of peers which can be used
            # for cross org gossip communication.  Note, this value is only
            # encoded in the genesis block in the Application section context
            - Host: peer0.org1.example.com
              Port: 7051
```

Fig. 6.4 Code snippet from configtx.yaml file



```
networks:
  test:

services:
  ca-org1:
    image: hyperledger/fabric-ca
    environment:
      - FABRIC_CA_HOME=/etc/hyperledger/fabric-ca-server
      - FABRIC_CA_SERVER_CA_NAME=ca.org1.example.com
      - FABRIC_CA_SERVER_CA_CERTFILE=/etc/hyperledger/fabric-ca-server-config/ca.org1.example.com-cert.pem
      - FABRIC_CA_SERVER_CA_KEYFILE=/etc/hyperledger/fabric-ca-server-config/priv_sk
      - FABRIC_CA_SERVER_TLS_ENABLED=true
      - FABRIC_CA_SERVER_TLS_CERTFILE=/etc/hyperledger/fabric-ca-server-tls/tlsca.org1.example.com-cert.pem
      - FABRIC_CA_SERVER_TLS_KEYFILE=/etc/hyperledger/fabric-ca-server-tls/priv_sk
    ports:
      - "7054:7054"
    command: sh -c 'fabric-ca-server start -b admin:adminpw -d'
    volumes:
      - ./channel/crypto-config/peerOrganizations/org1.example.com/ca/:/etc/hyperledger/fabric-ca-server-config
      - ./channel/crypto-config/peerOrganizations/org1.example.com/tlsca/:/etc/hyperledger/fabric-ca-server-tls
    container_name: ca.org1.example.com
    hostname: ca.org1.example.com
    networks:
      - test

  orderer.example.com:
    container_name: orderer.example.com
    image: hyperledger/fabric-orderer:2.1
    dns_search: .
    environment:
      - ORDERER_GENERAL_LOGLEVEL=info
      - FABRIC_LOGGING_SPEC=INFO
      - ORDERER_GENERAL_LISTENADDRESS=0.0.0.0
      - ORDERER_GENERAL_GENESISMETHOD=file
      - ORDERER_GENERAL_GENESISFILE=/var/hyperledger/orderer/genesis.block
      - ORDERER_GENERAL_LOCALMSPID=OrdererMSP
      - ORDERER_GENERAL_LOCALMSPDIR=/var/hyperledger/orderer/msp
      - ORDERER_GENERAL_TLS_ENABLED=true
      - ORDERER_GENERAL_TLS_PRIVATEKEY=/var/hyperledger/orderer/tls/server.key
      - ORDERER_GENERAL_TLS_CERTIFICATE=/var/hyperledger/orderer/tls/server.crt
      - ORDERER_GENERAL_TLS_ROOTCAS=[/var/hyperledger/orderer/tls/ca.crt]
      - ORDERER KAFKA VERBOSE=true
```

Fig. 6.5 Code snippet from docker-compose.yaml file

be accessible to hospitals, doctors, and medical practitioners. Similarly in every industry, there is always partial data sharing among the organizations to build trust and promote transparency.

Consortium blockchain (Foschini et al. 2020) leverages support for trustworthy and transparent data transactions by allowing different organizations to be a part of the network. It is ideal for developing enterprise-grade applications because of its adaptability. Hyperledger Fabric provides pluggable modules for good stewardship of the application. Figure 6.6 depicts the consortium blockchain with two organizations.

In the creation of a consortium blockchain, there is no restriction on the number of organizations. All the peers of these organizations communicate with the help of a channel they belong to. Hyperledger Fabric allows the peers to be part of multiple channels. Multiple channels are required when two organizations want to share confidential information. These organizations will create a separate channel open to specific members of these two organizations. This channel maintains its ledger discretely termed as private data collection. As the organizations are more than one, the number of orderers also will increase consequently leading to the establishment of an ordering service (Zhong et al. Dec. 2020). Ordering services use different consensus mechanisms like RAFT (Khettry et al. 2021) to govern the blockchain and its channels.

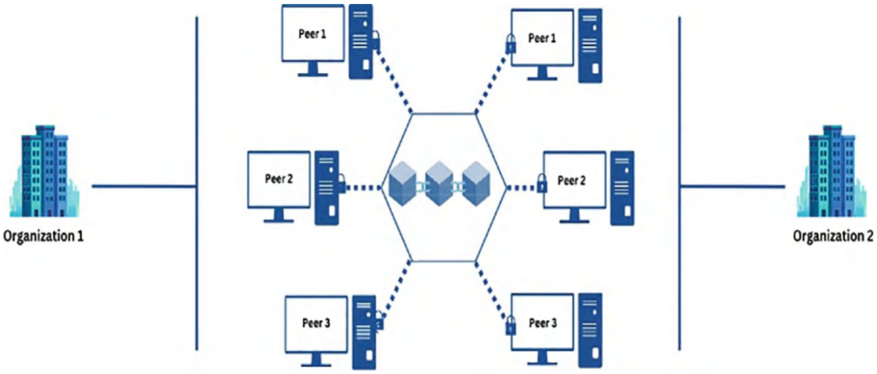


Fig. 6.6 Consortium blockchain with two organizations

The network components recognized in Hyperledger Fabric for a consortium Blockchain are:

1. Organizations
2. A set of peers for each organization
3. Certificate Authority for each organization
4. Channels for communication
5. Orderers to maintain the channel.

The following steps show the creation of a consortium blockchain.

Step 1: Creation of artifacts for the network

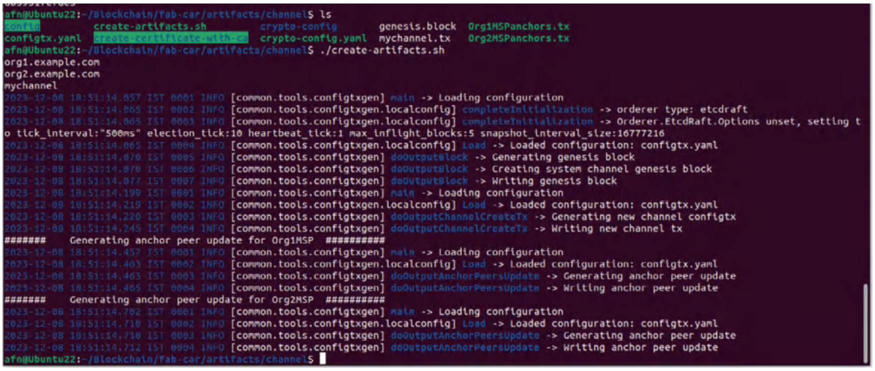
The artifacts are the entities and policies that participate in the network. Configuration files contain all the artifact details of the organizations, peers, orderers, and channels. The capabilities defined with version numbers in these files help to enable compatibility and consistency with fabric binaries.

As Hyperledger Fabric is a permissioned network the policies define all mem bers' access rights and govern the same. These files additionally configure the Certificate Authority, which is used to verify network members. On execution of these configuration files, the genesis block, channel details with orderers, and participating organizations are generated. Figure 6.7 depicts the successful creation of the artifacts for two organizations, Org1 and Org2.

Step 2: Setup the docker services with the Fabric images

Deploying Hyperledger Fabric configuration files and other dependencies in a virtual machine might be difficult and time-consuming. Docker provide better service and convenience compared to virtual machines. The artifacts defined above are individually packaged as images. These images are deployed in the docker and are labeled with container IDs.

Figure 6.8 displays the generation of the docker images for 2 peers and 1 CA for each organization. Also, there are 4 orderers for the ordering service with the database. Following is the output of the docker-ps command which lists all the containers running with images created. The status and port number are essential for



```
afn@Ubuntu22:~/Blockchain/fab-car/artifacts/channel$ ls
config          create-artifacts.sh             crypto-config         genesis.block   Org1MSPanchors.tx
configtx.yaml   create-certificate-with-ca      crypto-config.yaml    mychannel.tx    Org2MSPanchors.tx
afn@Ubuntu22:~/Blockchain/fab-car/artifacts/channel$ ./create-artifacts.sh
org1.example.com
org2.example.com
mychannel
2023-12-08 18:51:14.057 IST 0001 INFO [common.tools.configtxgen] main -> Loading configuration
2023-12-08 18:51:14.065 IST 0002 INFO [common.tools.configtxgen.localconfig] completeInitialization -> orderer type: etcdraft
2023-12-08 18:51:14.065 IST 0003 INFO [common.tools.configtxgen.localconfig] completeInitialization -> Orderer.EtcdRaft.Options unset, setting t
o tick_interval:"500ms" election_tick:10 heartbeat_tick:1 max_inflight_blocks:5 snapshot_interval_size:16777216
2023-12-08 18:51:14.065 IST 0004 INFO [common.tools.configtxgen.localconfig] Load -> Loaded configuration: configtx.yaml
2023-12-08 18:51:14.070 IST 0005 INFO [common.tools.configtxgen] doOutputBlock -> Generating genesis block
2023-12-08 18:51:14.070 IST 0006 INFO [common.tools.configtxgen] doOutputBlock -> Creating system channel genesis block
2023-12-08 18:51:14.077 IST 0007 INFO [common.tools.configtxgen] doOutputBlock -> Writing genesis block
2023-12-08 18:51:14.199 IST 0001 INFO [common.tools.configtxgen] main -> Loading configuration
2023-12-08 18:51:14.219 IST 0002 INFO [common.tools.configtxgen.localconfig] Load -> Loaded configuration: configtx.yaml
2023-12-08 18:51:14.220 IST 0003 INFO [common.tools.configtxgen] doOutputChannelCreateTx -> Generating new channel configtx
2023-12-08 18:51:14.245 IST 0004 INFO [common.tools.configtxgen] doOutputChannelCreateTx -> Writing new channel tx
#######    Generating anchor peer update for Org1MSP  ##########
2023-12-08 18:51:14.452 IST 0001 INFO [common.tools.configtxgen] main -> Loading configuration
2023-12-08 18:51:14.463 IST 0002 INFO [common.tools.configtxgen.localconfig] Load -> Loaded configuration: configtx.yaml
2023-12-08 18:51:14.463 IST 0003 INFO [common.tools.configtxgen] doOutputAnchorPeersUpdate -> Generating anchor peer update
2023-12-08 18:51:14.465 IST 0004 INFO [common.tools.configtxgen] doOutputAnchorPeersUpdate -> Writing anchor peer update
#######    Generating anchor peer update for Org2MSP  ##########
2023-12-08 18:51:14.702 IST 0001 INFO [common.tools.configtxgen] main -> Loading configuration
2023-12-08 18:51:14.710 IST 0002 INFO [common.tools.configtxgen.localconfig] Load -> Loaded configuration: configtx.yaml
2023-12-08 18:51:14.710 IST 0003 INFO [common.tools.configtxgen] doOutputAnchorPeersUpdate -> Generating anchor peer update
2023-12-08 18:51:14.712 IST 0004 INFO [common.tools.configtxgen] doOutputAnchorPeersUpdate -> Writing anchor peer update
afn@Ubuntu22:~/Blockchain/fab-car/artifacts/channel$
```

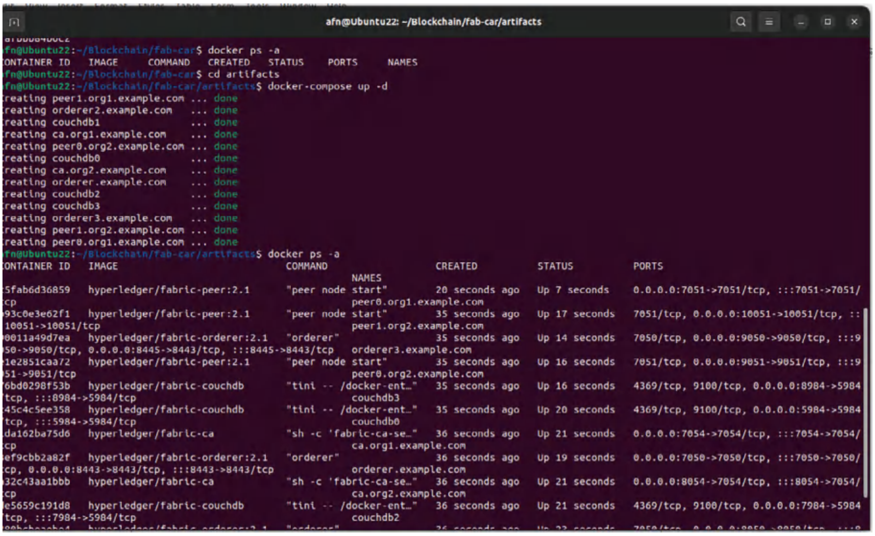
Fig. 6.7 Artifact Creation as defined in the configuration file

accessing the container and should be checked before deploying the chaincodes on the channel.

Step 3: Channel creation

The genesis block is used in channel creation with a channel creation transaction that includes the initial configuration of the channel. Subsequently, the genesis block is passed to an ordering service node in a join request as shown in Fig. 9a. Figure 9b confirms the successful channel creation with the orderer defined in the configuration file.

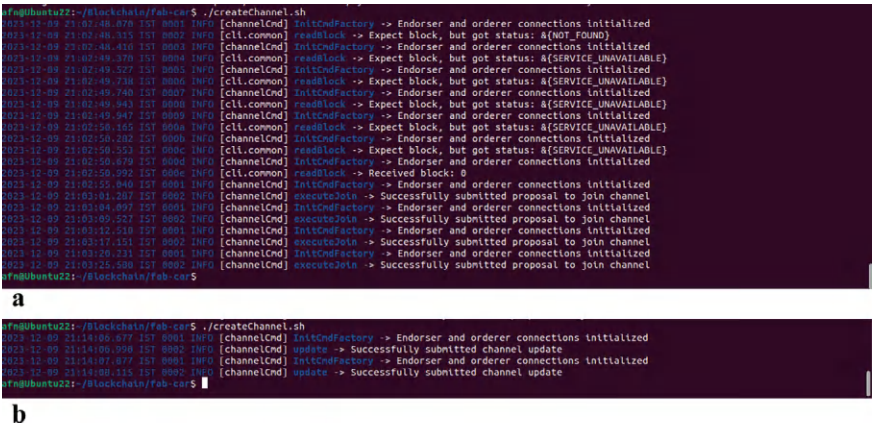
With the above three steps, the consortium network is successfully created and is active. The client applications can access and query the ledger with the help of



afn@Ubuntu22: ~/Blockchain/fab-car/artifacts

```
fn@Ubuntu22:~/Blockchain/fab-car$ docker ps -a
ONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
fn@Ubuntu22:~/Blockchain/fab-car$ cd artifacts
fn@Ubuntu22:~/Blockchain/fab-car/artifacts$ docker-compose up -d
reating peer1.org1.example.com ... done
reating orderer2.example.com   ... done
reating couchdb1               ... done
reating ca.org1.example.com    ... done
reating peer0.org2.example.com ... done
reating couchdb0               ... done
reating ca.org2.example.com    ... done
reating orderer.example.com    ... done
reating couchdb2               ... done
reating couchdb3               ... done
reating orderer3.example.com   ... done
reating peer1.org2.example.com ... done
reating peer0.org1.example.com ... done
fn@Ubuntu22:~/Blockchain/fab-car/artifacts$ docker ps -a
ONTAINER ID   IMAGE                                COMMAND                  CREATED          STATUS          PORTS
                                                                NAMES
5fab6d36859   hyperledger/fabric-peer:2.1          "peer node start"        20 seconds ago   Up 7 seconds    0.0.0.0:7051->7051/tcp, :::7051->7051/
cp                                                              peer0.org1.example.com
93c0e3e62f1   hyperledger/fabric-peer:2.1          "peer node start"        35 seconds ago   Up 17 seconds   7051/tcp, 0.0.0.0:10051->10051/tcp, ::
10051->10051/tcp                                                peer1.org2.example.com
0011a49d7ea   hyperledger/fabric-orderer:2.1       "orderer"                35 seconds ago   Up 14 seconds   7050/tcp, 0.0.0.0:9050->9050/tcp, :::9
50->9050/tcp, 0.0.0.0:8445->8443/tcp, :::8445->8443/tcp         orderer3.example.com
1e2851caa72   hyperledger/fabric-peer:2.1          "peer node start"        35 seconds ago   Up 16 seconds   7051/tcp, 0.0.0.0:9051->9051/tcp, :::9
51->9051/tcp                                                    peer0.org2.example.com
6bd0298f53b   hyperledger/fabric-couchdb           "tini -- /docker-ent…"   35 seconds ago   Up 16 seconds   4369/tcp, 9100/tcp, 0.0.0.0:8984->5984
tcp, :::8984->5984/tcp                                          couchdb3
45c4c5ee358   hyperledger/fabric-couchdb           "tini -- /docker-ent…"   35 seconds ago   Up 20 seconds   4369/tcp, 9100/tcp, 0.0.0.0:5984->5984
tcp, :::5984->5984/tcp                                          couchdb0
da162ba75d6   hyperledger/fabric-ca                "sh -c 'fabric-ca-se…"   36 seconds ago   Up 21 seconds   0.0.0.0:7054->7054/tcp, :::7054->7054/
cp                                                              ca.org1.example.com
ef9cbb2a82f   hyperledger/fabric-orderer:2.1       "orderer"                36 seconds ago   Up 19 seconds   0.0.0.0:7050->7050/tcp, :::7050->7050/
cp, 0.0.0.0:8443->8443/tcp, :::8443->8443/tcp                   orderer.example.com
32c43aa1bbb   hyperledger/fabric-ca                "sh -c 'fabric-ca-se…"   36 seconds ago   Up 21 seconds   0.0.0.0:8054->7054/tcp, :::8054->7054/
cp                                                              ca.org2.example.com
e5659c191d8   hyperledger/fabric-couchdb           "tini -- /docker-ent…"   36 seconds ago   Up 21 seconds   4369/tcp, 9100/tcp, 0.0.0.0:7984->5984
tcp, :::7984->5984/tcp                                          couchdb2
```

Fig. 6.8 Docker images for the network



```
afn@Ubuntu22:~/Blockchain/fab-car$ ./createChannel.sh
2023-12-09 21:02:48.070 IST 0001 INFO [channelCmd] InitCmdFactory -> Endorser and orderer connections initialized
2023-12-09 21:02:48.315 IST 0002 INFO [cli.common] readBlock -> Expect block, but got status: &{NOT_FOUND}
2023-12-09 21:02:48.410 IST 0003 INFO [channelCmd] InitCmdFactory -> Endorser and orderer connections initialized
2023-12-09 21:02:49.370 IST 0004 INFO [cli.common] readBlock -> Expect block, but got status: &{SERVICE_UNAVAILABLE}
2023-12-09 21:02:49.527 IST 0005 INFO [channelCmd] InitCmdFactory -> Endorser and orderer connections initialized
2023-12-09 21:02:49.738 IST 0006 INFO [cli.common] readBlock -> Expect block, but got status: &{SERVICE_UNAVAILABLE}
2023-12-09 21:02:49.740 IST 0007 INFO [channelCmd] InitCmdFactory -> Endorser and orderer connections initialized
2023-12-09 21:02:49.943 IST 0008 INFO [cli.common] readBlock -> Expect block, but got status: &{SERVICE_UNAVAILABLE}
2023-12-09 21:02:49.947 IST 0009 INFO [channelCmd] InitCmdFactory -> Endorser and orderer connections initialized
2023-12-09 21:02:50.165 IST 000a INFO [cli.common] readBlock -> Expect block, but got status: &{SERVICE_UNAVAILABLE}
2023-12-09 21:02:50.282 IST 000b INFO [channelCmd] InitCmdFactory -> Endorser and orderer connections initialized
2023-12-09 21:02:50.553 IST 000c INFO [cli.common] readBlock -> Expect block, but got status: &{SERVICE_UNAVAILABLE}
2023-12-09 21:02:50.679 IST 000d INFO [channelCmd] InitCmdFactory -> Endorser and orderer connections initialized
2023-12-09 21:02:50.992 IST 000e INFO [cli.common] readBlock -> Received block: 0
2023-12-09 21:02:55.040 IST 0001 INFO [channelCmd] InitCmdFactory -> Endorser and orderer connections initialized
2023-12-09 21:03:01.287 IST 0002 INFO [channelCmd] executeJoin -> Successfully submitted proposal to join channel
2023-12-09 21:03:04.097 IST 0001 INFO [channelCmd] InitCmdFactory -> Endorser and orderer connections initialized
2023-12-09 21:03:09.527 IST 0002 INFO [channelCmd] executeJoin -> Successfully submitted proposal to join channel
2023-12-09 21:03:12.510 IST 0001 INFO [channelCmd] InitCmdFactory -> Endorser and orderer connections initialized
2023-12-09 21:03:17.151 IST 0002 INFO [channelCmd] executeJoin -> Successfully submitted proposal to join channel
2023-12-09 21:03:20.231 IST 0001 INFO [channelCmd] InitCmdFactory -> Endorser and orderer connections initialized
2023-12-09 21:03:25.580 IST 0002 INFO [channelCmd] executeJoin -> Successfully submitted proposal to join channel
afn@Ubuntu22:~/Blockchain/fab-car$
```

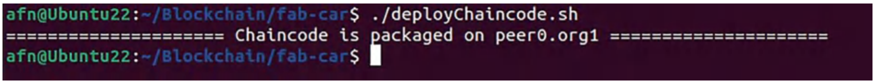
```
afn@Ubuntu22:~/Blockchain/fab-car$ ./createChannel.sh
2023-12-09 21:14:06.677 IST 0001 INFO [channelCmd] InitCmdFactory -> Endorser and orderer connections initialized
2023-12-09 21:14:06.998 IST 0002 INFO [channelCmd] update -> Successfully submitted channel update
2023-12-09 21:14:07.877 IST 0001 INFO [channelCmd] InitCmdFactory -> Endorser and orderer connections initialized
2023-12-09 21:14:08.115 IST 0002 INFO [channelCmd] update -> Successfully submitted channel update
afn@Ubuntu22:~/Blockchain/fab-car$
```

Fig. 6.9 **a** Channel creation. **b** Successful confirmation of joining the channel

smart contracts defined in the Fabric Chaincode. The Fabric Chaincode should be first packaged and installed on the channel peer members. The installed chaincode must be approved by the channel members and on getting a valid endorsement from the required number of peers the chaincode is committed. The Fabric chaincode can be deployed to perform various transactions on the ledger. The following screen demonstrates the lifecycle of a chaincode.

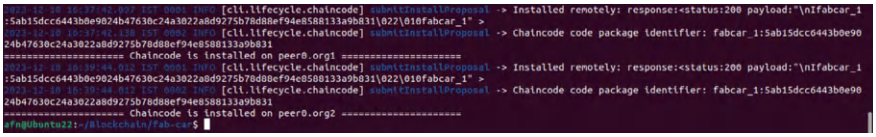
1. Package as shown in Fig. 6.10
2. Install as shown in Fig. 6.11
3. Approve as shown in Fig. 6.12
4. Commit as shown in Figs. 6.13 and 6.14.

Now the committed chaincode can easily be invoked by the client applications. Each invoke request must have the endorsement of the required number of peers following the chaincode endorsement policy to be validated against the current ledger



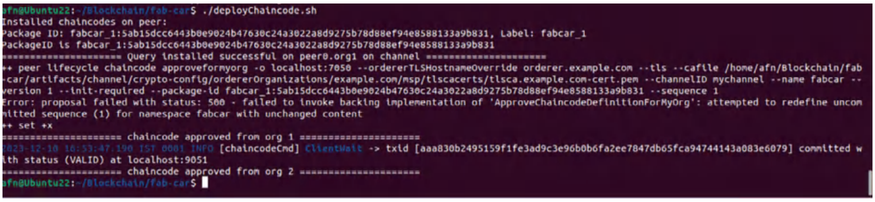
```
afn@Ubuntu22:~/Blockchain/fab-car$ ./deployChaincode.sh
===================== Chaincode is packaged on peer0.org1 =====================
afn@Ubuntu22:~/Blockchain/fab-car$
```

Fig. 6.10 Packaging the chaincode on the channel



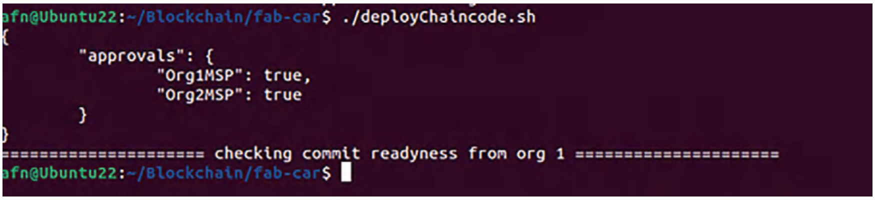
```
2023-12-10 16:37:42.097 IST 0001 INFO [cli.lifecycle.chaincode] submitInstallProposal -> Installed remotely: response:<status:200 payload:"\nIfabcar_1
:5ab15dcc6443b0e9024b47630c24a3022a8d9275b78d88ef94e8588133a9b831\022\010fabcar_1" >
2023-12-10 16:37:42.138 IST 0002 INFO [cli.lifecycle.chaincode] submitInstallProposal -> Chaincode code package identifier: fabcar_1:5ab15dcc6443b0e90
24b47630c24a3022a8d9275b78d88ef94e8588133a9b831
===================== Chaincode is installed on peer0.org1 =====================
2023-12-10 16:39:44.012 IST 0001 INFO [cli.lifecycle.chaincode] submitInstallProposal -> Installed remotely: response:<status:200 payload:"\nIfabcar_1
:5ab15dcc6443b0e9024b47630c24a3022a8d9275b78d88ef94e8588133a9b831\022\010fabcar_1" >
2023-12-10 16:39:44.012 IST 0002 INFO [cli.lifecycle.chaincode] submitInstallProposal -> Chaincode code package identifier: fabcar_1:5ab15dcc6443b0e90
24b47630c24a3022a8d9275b78d88ef94e8588133a9b831
===================== Chaincode is installed on peer0.org2 =====================
afn@Ubuntu22:~/Blockchain/fab-car$
```

Fig. 6.11 Installing the chaincode on the channel



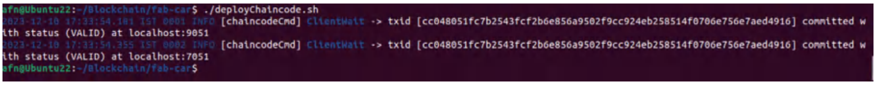
```
afn@Ubuntu22:~/Blockchain/fab-car$ ./deployChaincode.sh
Installed chaincodes on peer:
Package ID: fabcar_1:5ab15dcc6443b0e9024b47630c24a3022a8d9275b78d88ef94e8588133a9b831, Label: fabcar_1
PackageID is fabcar_1:5ab15dcc6443b0e9024b47630c24a3022a8d9275b78d88ef94e8588133a9b831
==================== Query installed successful on peer0.org1 on channel ====================
++ peer lifecycle chaincode approveformyorg -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com --tls --cafile /home/afn/Blockchain/fab
-car/artifacts/channel/crypto-config/ordererOrganizations/example.com/msp/tlscacerts/tlsca.example.com-cert.pem --channelID mychannel --name fabcar --
version 1 --init-required --package-id fabcar_1:5ab15dcc6443b0e9024b47630c24a3022a8d9275b78d88ef94e8588133a9b831 --sequence 1
Error: proposal failed with status: 500 - failed to invoke backing implementation of 'ApproveChaincodeDefinitionForMyOrg': attempted to redefine uncom
mitted sequence (1) for namespace fabcar with unchanged content
++ set +x
==================== chaincode approved from org 1 ====================
2023-12-10 16:53:47.190 IST 0001 INFO [chaincodeCmd] ClientWait -> txid [aaa830b2495159f1fe3ad9c3e96b0b6fa2ee7847db65fca94744143a083e6079] committed w
ith status (VALID) at localhost:9051
==================== chaincode approved from org 2 ====================
afn@Ubuntu22:~/Blockchain/fab-car$
```

Fig. 6.12 Validating Chaincode



```
afn@Ubuntu22:~/Blockchain/fab-car$ ./deployChaincode.sh
{
        "approvals": {
                "Org1MSP": true,
                "Org2MSP": true
        }
}
==================== checking commit readyness from org 1 ====================
afn@Ubuntu22:~/Blockchain/fab-car$
```

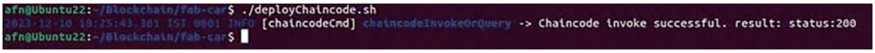
Fig. 6.13 Checking the endorsing peer organizations for the commit



```
afn@Ubuntu22:~/Blockchain/fab-car$ ./deployChaincode.sh
2023-12-10 17:33:54.181 IST 0001 INFO [chaincodeCmd] ClientWait -> txid [cc048051fc7b2543fcf2b6e856a9502f9cc924eb258514f0706e756e7aed4916] committed w
ith status (VALID) at localhost:9051
2023-12-10 17:33:54.355 IST 0002 INFO [chaincodeCmd] ClientWait -> txid [cc048051fc7b2543fcf2b6e856a9502f9cc924eb258514f0706e756e7aed4916] committed w
ith status (VALID) at localhost:7051
afn@Ubuntu22:~/Blockchain/fab-car$
```

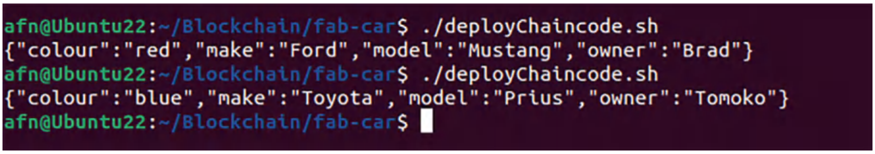
Fig. 6.14 Commit the chaincode

state as illustrated in Figs. 6.15 and 6.16. Any changes or upgrades made in the chaincode require the execution of the complete chaincode lifecycle and each deployment is saved with different version numbers in the logs.



```
afn@Ubuntu22:~/Blockchain/fab-car$ ./deployChaincode.sh
2023-12-10 18:25:43.361 IST 0001 INFO [chaincodeCmd] chaincodeInvokeOrQuery -> Chaincode invoke successful. result: status:200
afn@Ubuntu22:~/Blockchain/fab-car$
```

Fig. 6.15 Chaincode invocation



```
afn@Ubuntu22:~/Blockchain/fab-car$ ./deployChaincode.sh
{"colour":"red","make":"Ford","model":"Mustang","owner":"Brad"}
afn@Ubuntu22:~/Blockchain/fab-car$ ./deployChaincode.sh
{"colour":"blue","make":"Toyota","model":"Prius","owner":"Tomoko"}
afn@Ubuntu22:~/Blockchain/fab-car$
```

Fig. 6.16 Query specific records by using the specific smart contract of the chaincode

6.3 Integrate Hyperledger Explorer Tool

Hyperledger Fabric blockchain network as shown in the previous section is completely deployed and executed in CLI. Any transaction initiated or committed is also displayed in the shell environment. To make these activities human-understandable and create an abstract visual view of the Hyperledger Fabric blockchain network DTCC, Intel, and IBM developed a tool called Hyperledger Explorer (Manevich et al. 2018). This project is currently moved to End of Life status by the maintainers but the code repository is still available on Git Hub to be cloned and implemented.

Hyperledger Explorer can be implemented through containerization in Docker or by installing individual dependencies. The dashboard of Hyperledger Explorer gives a complete overview of the Fabric network currently running. The interface can be utilized to visualize the entire execution of the transactions in the network and analyze the network traffic patterns. The display includes the particulars of the number of blocks, transactions, nodes, and chaincodes as card visuals. It gives the specifics of the participating peers and overall transaction analysis of the transactions per hour/min in the other supported graphical formats.

6.3.1 Hyperledger Explorer Architecture

The Architecture of Hyperledger Explorer comprised of:

- Presentation Layer: The presentation layer is designed using ReactJS as it is simple and offers better performance and support for rendering interactive UI.
- Backend Layer: Multiple services are deployed in the backend for provisioning access to blockchain running in the background like Node.js, REST API, swagger, gateway, and synchronizer. All these services reinforce the connection with the Hyperledger Fabric network and enable a real-time view of the channel, blocks, and transactions.

Synchronizer in the backend layer accesses the Fabric through the gateway and continuously identifies the organizations with their MSPs in the channel. Recognize the peers and get concurrent updates from the network regarding the transactions, and peers joining or leaving the network. The following section highlights the configuration required for consortium blockchain.

6.4 Integrate Hyperledger Explorer Tool with Consortium Blockchain

Integrating the Hyperledger Explorer tool empowers efficient monitoring of the Hyperledger Fabric network activities. It does not assist in the creation of the Fabric blockchain network or invocation of queries on the ledger. To assimilate this tool the Fabric should be up and running in the background as explained in Section 6.3. The main components that need to be installed for the integration are as follows:

1. Docker and Docker Compose
2. Node.js
3. PostgreSQL.

After the successful installation or upgradation of the above components, it is necessary to clone the blockchain Explorer repository. This repository contains three pivotal configuration files to establish the connection with the Hyperledger Fabric network. The first file is *config.json.* The following is the content of the same.

```
{
"network-configs": {
      "first-network": {
            "name": "first-network",
            "profile":"./connection-profile/first.json"
            }
      },
      "license": "Apache-2.0"
}
```

This code specifies the user-defined name of the network and the connection profile to be used for the network configuration. The second file is first.json which defines the other network configuration parameters.

```
{
      "name": "first-network",
      "version": "1.0.0",
      "license": "Apache-2.0",
      "client": {
            "tlsEnable": true,
            "adminUser": "admin",
            "adminPassword": "adminpw",
            "enableAuthentication": false,
            "organization": "Org1MSP",
            "connection": {
                  "timeout": {
                        "peer": {
                              "endorser": "300"
                        },
                        "orderer": "300"
                  }
            }
      },
      "channels": {
            "mychannel": {
                  "peers": {
                        "peer0.org1.example.com": {}
                  },
                  "connection": {
                        "timeout": {
                              "peer": {
                                    "endorser": "6000",
                                    "eventHub": "6000",
                                    "eventReg": "6000"
                              }
                        }
                  }
            }
      },
      "organizations": {
            "Org1MSP": {
                  "mspid": "Org1MSP",
                  "adminPrivateKey": {
                        "path":
"/etc/data/peerOrganizations/org1.example.com/users/Adm
in@org1.example.com/msp/keystore/priv_sk"
                  },
                  "signedCert": {
                        "path":
"/etc/data/peerOrganizations/org1.example.com/users/Adm
in@org1.example.com/msp/signcerts/Admin@org1.example.co
m-cert.pem"
                  }
            }
      },
      "peers": {
            "peer0.org1.example.com": {
                  "tlsCACerts": {
                        "path":
"/etc/data/peerOrganizations/org1.example.com/peers/pee
r0.org1.example.com/tls/ca.crt"
                  },
                  "url":
"grpcs://peer0.org1.example.com:7051",
                  "eventUrl":
"grpcs://peer0.org1.example.com:7053",
                  "grpcOptions": {
                        "ssl-target-name-override":
"peer0.org1.example.com"
                  }
            }
      }
}
```

The code has details of the organization and peers along with the private keys and certificates. These authentication details are required to validate the access to the network. It also contains the credentials to be used for accessing the Hyperledger Explorer interface. The third file is docker-compose.yaml as shown below.

```
version: "2.1"
volumes:
  data:
  walletstore:
networks:
  test:
services:
  explorerdb.mynetwork.com:
    image: hyperledger/explorer-db:latest
    container_name: explorerdb.mynetwork.com
    hostname: explorerdb.mynetwork.com
    environment:
      - DATABASE_DATABASE=fabricexplorer
      - DATABASE_USERNAME=hppoc
      - DATABASE_PASSWORD=password
    healthcheck:
      test:  "pg_isready  -h  localhost  -p  5432  -q  -U
postgres"
      interval: 30s
      timeout: 10s
      retries: 5
    volumes:
      - data:/var/lib/postgresql/data
    networks:
      - test

  explorer.mynetwork.com:
    image: hyperledger/explorer:1.1.1
    container_name: explorer.mynetwork.com
    hostname: explorer.mynetwork.com
    environment:
      - DATABASE_HOST=explorerdb.mynetwork.com
      - DATABASE_DATABASE=fabricexplorer
      - DATABASE_USERNAME=hppoc
      - DATABASE_PASSWD=password
      - LOG_LEVEL_APP=debug
      - LOG_LEVEL_DB=debug
      - LOG_LEVEL_CONSOLE=info
      - LOG_CONSOLE_STDOUT=true
      - DISCOVERY_AS_LOCALHOST=false
    volumes:
      -
./config.json:/opt/explorer/app/platform/fabric/config.
json
```

```
      -                                           ./connection-
profile:/opt/explorer/app/platform/fabric/connection-
profile
      - ./examples/net1/crypto:/tmp/crypto
      - walletstore:/opt/wallet
      - ./crypto-config/:/etc/data
    command: sh -c "node /opt/explorer/main.js && tail
-f /dev/null"
    ports:
      - 8080:8080
    depends_on:
      explorerdb.mynetwork.com:
        condition: service_healthy
    networks:
      - test
```

The code above encompasses the configuration required to containerize Hyperledger Explorer and Explorer database in Docker. This file should be executed to start the Hyperledger Explorer on the local host with port number 8080 as demonstrated in Manevich et al. (2018). In this consortium blockchain, the Explorer will show the two peers of each organization Org1 and Org2. The Explorer graphically displays the number of blocks and transactions committed on the blockchain. It allows a user to see the network, channel, chaincode, block, and transaction details by switching to the respective tabs.

6.5 Integrate Hyperledger Caliper Tool with Private Blockchain

The previous sections enlightened the procedure to create a blockchain network in Hyperledger Fabric and envisage the blockchain network with abstraction using the Hyperledger Explorer tool. These networks can be designed for various use cases of enterprise-grade applications. When instances are compared individually for a use case these networks may perform differently for each case. Some of the major performance metrics are the duration it takes to execute a query and render an output within a defined time window. Another one could be the number of transactions that the network can process in a given period of assessment usually seconds, minutes, or hours. The other aspect that can also affect the performance is resource utilization. The efficient use of resources is always preferred. Hy perledger Greenhouse provides a performance benchmark tool called Caliper (Hyperledger Explorer Documentation Explorer 2021; Hyperledger Caliper Project 2024) to test these metrics for different Distributed Ledger Technologies like Ethereum (Alom et al. 2022), Besu (Dabbagh et al. 2020), and Fabric. Nguyen et al. (Mostarda et al. n.d.) proposed an extended model of the Hyperledger Caliper tool.

Ucbas et al. (Nguyen et al. 2021) performed a study to evaluate and compare the performance of private blockchain networks on Ethereum and Hyperledger Fabric focusing on the performance parameters throughput and latency to assess the capability of the network to support scalability. The Hyperledger performed better when assessed for throughput but failed to surpass Ethereum in latency. Throughput is generally calculated in TPS (Transactions per second) to measure the performance of a blockchain in terms of handling the number of transactions in a timeframe. On the other hand, latency evaluates the time the network takes to validate a transaction.

Geyer et al. (2019) employed the Hyperledger Caliper tool to appraise the performance of the Hyperledger Fabric private network for communication networks and emphasize the parameters that impact the performance. Baliga et al. (Ucbas et al. 2023) presented an approach to evaluate the performance and scalability of the Fabric 1.0 network using the Hyperledger Caliper tool. The study shows that the throughput is crucial as it increases the system latency. It also highlights the inefficient utilization of resources by the network members. The subsequent section discusses how the Hyperledger Caliper benchmark tool can be integrated with the network.

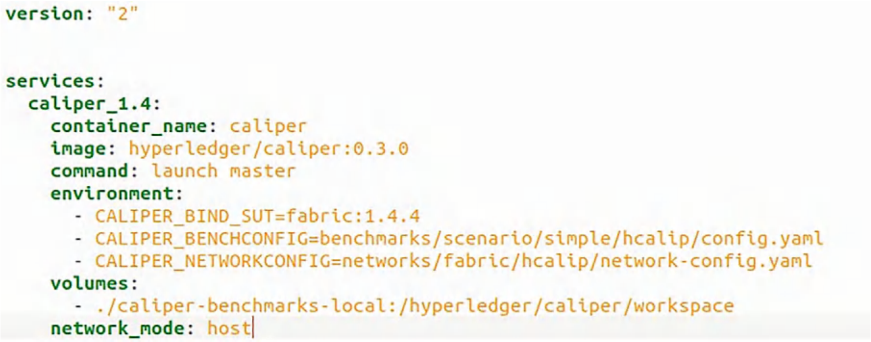
6.6 Integrate Hyperledger Caliper Tool with Consortium Blockchain

To integrate Hyperledger Caliper with the network created in Sect. 6.3, ensure the network is active and running. The Hyperledger Caliper benchmark tool can be utilized by installing the npm packages of Caliper or by containerizing the docker images. The repositories for both methods are easily available. For in stalling and embedding the Caliper in the application, three steps are required:

1. Installing the latest Caliper CLI
2. Bind the Fabric network
3. Launch the Caliper Manager to perform the tests.

Following are the installation commands required to do the same:

```
user@ubuntu:~/caliper-benchmarks$    npm   install   --
only=prod @hyperledger/caliper-cli@0.6.0
user@ubuntu:~/caliper-benchmarks$  npx  caliper  launch
manager \
    --caliper-bind-sut fabric:fabric-gateway \
    --caliper-workspace . \
    --caliper-benchconfig benchmarks/scenario/simple/
config.yaml \
    --caliper-networkconfig networks/fabric/test-
network.yaml
```



```
version: "2"


services:
  caliper_1.4:
    container_name: caliper
    image: hyperledger/caliper:0.3.0
    command: launch master
    environment:
      - CALIPER_BIND_SUT=fabric:1.4.4
      - CALIPER_BENCHCONFIG=benchmarks/scenario/simple/hcalip/config.yaml
      - CALIPER_NETWORKCONFIG=networks/fabric/hcalip/network-config.yaml
    volumes:
      - ./caliper-benchmarks-local:/hyperledger/caliper/workspace
    network_mode: host
```

Fig. 6.17 Code snippet of docker-compose.yaml

Following the successful execution of the above commands, the test results can be seen on the console.

Caliper containerization is simple as the docker images can be cloned easily and deployed in the docker. For demonstration purposes in this section, this method is assumed. After cloning the docker-images the files that required to be modified are docker-compose.yaml, network-config.yaml, and config.yaml.

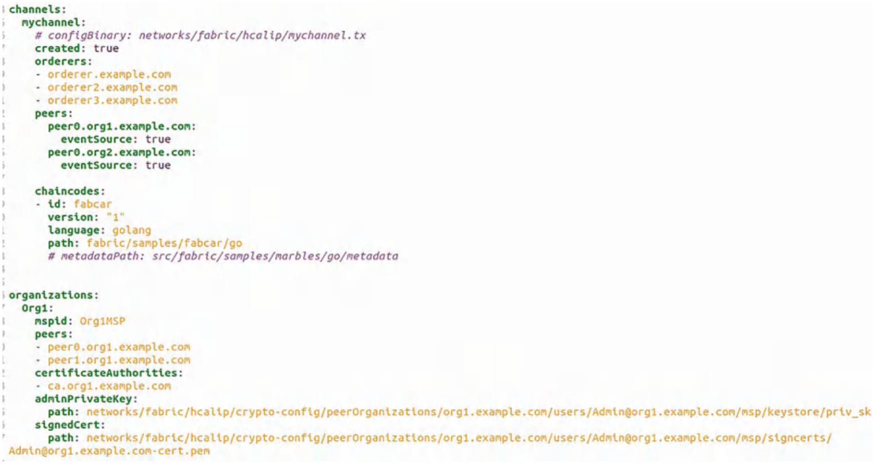
In docker-compose.yaml file as shown in Fig. 6.17, the Caliper image details with version number 0.3.0 and the container name should be mentioned. The configuration of environment variables is essential as it accepts the details of the Fabric version to bind with the Caliper. It also includes the network configuration as shown in Fig. 6.18 which contains the details of the network artifacts of the blockchain like organizations, peers, orderers, Certificate Authorities, and its crypto material. These crypto materials support authenticating and executing the test benchmark parameters set in the config.yaml file is shown in Fig. 6.19.

The above file contains the benchmarks to be followed for evaluating the performance of the consortium blockchain network. After defining the name and de scription for the test, the number of processes has to be specified in workers for the test execution. These numbers are essential for the generation of workload for the Caliper. The rounds sections specify the initialization of the ledger from the chaincode with the label open. This label contains the other details as the number of transactions and transactions per second for the first round. The label query also contains similar details for the execution.

The following screenshots show the execution of the network (Figs. 6.20 and 6.21).

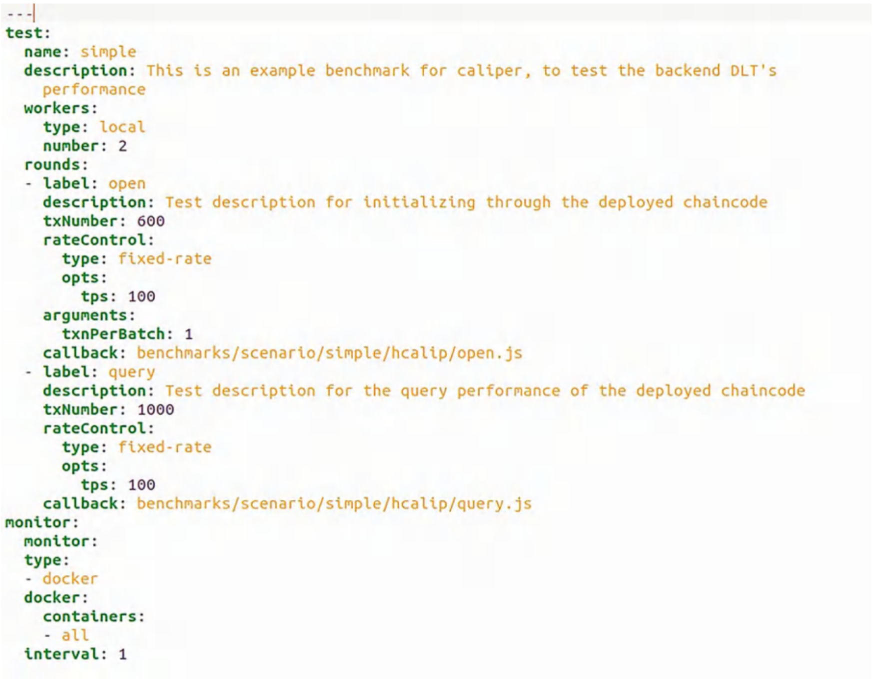
After the Caliper is created the report in HTML format will be generated in the Caliper benchmark directory as shown in Fig. 6.22.

The report shows Basic and system details as mentioned in the configuration files. In the summary of the performance metrics, the success and failure rates show the number of transactions executed without any error, which confirms the stability



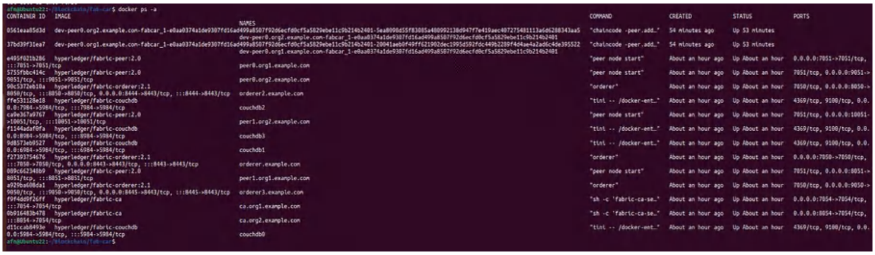
```yaml
channels:
  mychannel:
    # configBinary: networks/fabric/hcalip/mychannel.tx
    created: true
    orderers:
    - orderer.example.com
    - orderer2.example.com
    - orderer3.example.com
    peers:
      peer0.org1.example.com:
        eventSource: true
      peer0.org2.example.com:
        eventSource: true

    chaincodes:
    - id: fabcar
      version: "1"
      language: golang
      path: fabric/samples/fabcar/go
      # metadataPath: src/fabric/samples/marbles/go/metadata


organizations:
  Org1:
    mspid: Org1MSP
    peers:
    - peer0.org1.example.com
    - peer1.org1.example.com
    certificateAuthorities:
    - ca.org1.example.com
    adminPrivateKey:
      path: networks/fabric/hcalip/crypto-config/peerOrganizations/org1.example.com/users/Admin@org1.example.com/msp/keystore/priv_sk
    signedCert:
      path: networks/fabric/hcalip/crypto-config/peerOrganizations/org1.example.com/users/Admin@org1.example.com/msp/signcerts/
Admin@org1.example.com-cert.pem
```

Fig. 6.18 Code snippet of network-config.yaml



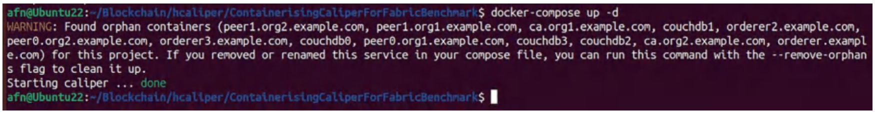
```yaml
---
test:
  name: simple
  description: This is an example benchmark for caliper, to test the backend DLT's
    performance
  workers:
    type: local
    number: 2
  rounds:
  - label: open
    description: Test description for initializing through the deployed chaincode
    txNumber: 600
    rateControl:
      type: fixed-rate
      opts:
        tps: 100
    arguments:
      txnPerBatch: 1
    callback: benchmarks/scenario/simple/hcalip/open.js
  - label: query
    description: Test description for the query performance of the deployed chaincode
    txNumber: 1000
    rateControl:
      type: fixed-rate
      opts:
        tps: 100
    callback: benchmarks/scenario/simple/hcalip/query.js
monitor:
  monitor:
  type:
  - docker
  docker:
    containers:
    - all
  interval: 1
```

Fig. 6.19 Test parameters in config.yaml



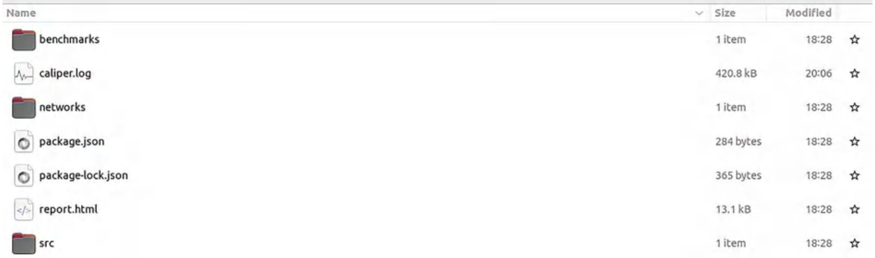
```
afn@Ubuntu22:~/Blockchain/fab-car$ docker ps -a
CONTAINER ID   IMAGE                                                                                                                                                                           COMMAND                  CREATED             STATUS              PORTS
                                                                                                       NAMES
0561eaa85d3d   dev-peer0.org2.example.com-fabcar_1-e0aa0374a1de9307fd16ad499a8507f92d6ecfd0cf5a5829ebe11c9b214b2401-5ea8098d55f83085a480992138d947f7e419aec407275481113a6d6288343aa5   "chaincode -peer.add…"   54 minutes ago      Up 53 minutes
                                                                                                       dev-peer0.org2.example.com-fabcar_1-e0aa0374a1de9307fd16ad499a8507f92d6ecfd0cf5a5829ebe11c9b214b2401
37bd39f31ea7   dev-peer0.org1.example.com-fabcar_1-e0aa0374a1de9307fd16ad499a8507f92d6ecfd0cf5a5829ebe11c9b214b2401-20041aeb0f49ff621902dec1995d592fdc449b2289f4d4ae4a2ad6c4de395522   "chaincode -peer.add…"   54 minutes ago      Up 53 minutes
                                                                                                       dev-peer0.org1.example.com-fabcar_1-e0aa0374a1de9307fd16ad499a8507f92d6ecfd0cf5a5829ebe11c9b214b2401
e495f021b286   hyperledger/fabric-peer:2.0                                                                                                                                                     "peer node start"        About an hour ago   Up About an hour    0.0.0.0:7051->7051/tcp,
:::7051->7051/tcp                                                                                      peer0.org1.example.com
5755fbbc414c   hyperledger/fabric-peer:2.0                                                                                                                                                     "peer node start"        About an hour ago   Up About an hour    7051/tcp, 0.0.0.0:9051->
9051/tcp, :::9051->9051/tcp                                                                            peer0.org2.example.com
90c5372eb10a   hyperledger/fabric-orderer:2.1                                                                                                                                                  "orderer"                About an hour ago   Up About an hour    7050/tcp, 0.0.0.0:8050->
8050/tcp, :::8050->8050/tcp, 0.0.0.0:8444->8443/tcp, :::8444->8443/tcp                                 orderer2.example.com
ffe531128e18   hyperledger/fabric-couchdb                                                                                                                                                      "tini -- /docker-ent…"   About an hour ago   Up About an hour    4369/tcp, 9100/tcp, 0.0.
0.0:7984->5984/tcp, :::7984->5984/tcp                                                                  couchdb2
ca9e367a9767   hyperledger/fabric-peer:2.0                                                                                                                                                     "peer node start"        About an hour ago   Up About an hour    7051/tcp, 0.0.0.0:10051-
>10051/tcp, :::10051->10051/tcp                                                                        peer1.org2.example.com
f1144adaf0fa   hyperledger/fabric-couchdb                                                                                                                                                      "tini -- /docker-ent…"   About an hour ago   Up About an hour    4369/tcp, 9100/tcp, 0.0.
0.0:8984->5984/tcp, :::8984->5984/tcp                                                                  couchdb3
9d8573eb9527   hyperledger/fabric-couchdb                                                                                                                                                      "tini -- /docker-ent…"   About an hour ago   Up About an hour    4369/tcp, 9100/tcp, 0.0.
0.0:6984->5984/tcp, :::6984->5984/tcp                                                                  couchdb1
f27393754676   hyperledger/fabric-orderer:2.1                                                                                                                                                  "orderer"                About an hour ago   Up About an hour    0.0.0.0:7050->7050/tcp,
:::7050->7050/tcp, 0.0.0.0:8443->8443/tcp, :::8443->8443/tcp                                           orderer.example.com
089c662348b9   hyperledger/fabric-peer:2.0                                                                                                                                                     "peer node start"        About an hour ago   Up About an hour    7051/tcp, 0.0.0.0:8051->
8051/tcp, :::8051->8051/tcp                                                                            peer1.org1.example.com
a929ba608da1   hyperledger/fabric-orderer:2.1                                                                                                                                                  "orderer"                About an hour ago   Up About an hour    7050/tcp, 0.0.0.0:9050->
9050/tcp, :::9050->9050/tcp, 0.0.0.0:8445->8443/tcp, :::8445->8443/tcp                                 orderer3.example.com
f9f4dd9f26ff   hyperledger/fabric-ca                                                                                                                                                           "sh -c 'fabric-ca-se…"   About an hour ago   Up About an hour    0.0.0.0:7054->7054/tcp,
:::7054->7054/tcp                                                                                      ca.org1.example.com
0b916483b478   hyperledger/fabric-ca                                                                                                                                                           "sh -c 'fabric-ca-se…"   About an hour ago   Up About an hour    0.0.0.0:8054->7054/tcp,
:::8054->7054/tcp                                                                                      ca.org2.example.com
d11ccab8493e   hyperledger/fabric-couchdb                                                                                                                                                      "tini -- /docker-ent…"   About an hour ago   Up About an hour    4369/tcp, 9100/tcp, 0.0.
0.0:5984->5984/tcp, :::5984->5984/tcp                                                                  couchdb0
afn@Ubuntu22:~/Blockchain/fab-car$
```

Fig. 6.20 Activating the Fabric network for the test



```
afn@Ubuntu22:~/Blockchain/hcaliper/ContainerisingCaliperForFabricBenchmark$ docker-compose up -d
WARNING: Found orphan containers (peer1.org2.example.com, peer1.org1.example.com, ca.org1.example.com, couchdb1, orderer2.example.com,
peer0.org2.example.com, orderer3.example.com, couchdb0, peer0.org1.example.com, couchdb3, couchdb2, ca.org2.example.com, orderer.exampl
e.com) for this project. If you removed or renamed this service in your compose file, you can run this command with the --remove-orphan
s flag to clean it up.
Starting caliper ... done
afn@Ubuntu22:~/Blockchain/hcaliper/ContainerisingCaliperForFabricBenchmark$ 
```

Fig. 6.21 Containerizing the Caliper images with the above configuration



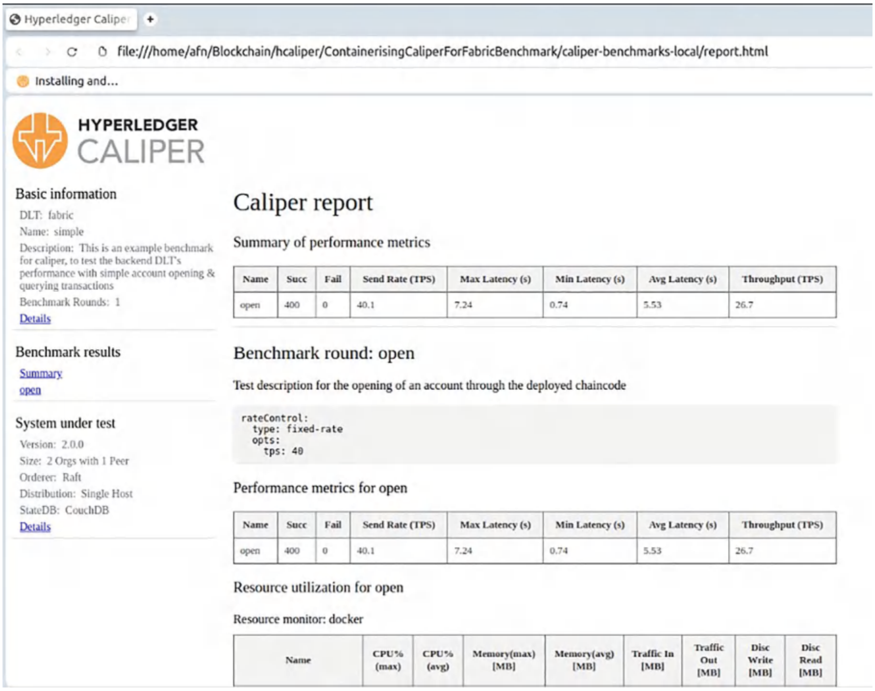
Name	Size	Modified	
benchmarks	1 item	18:28	☆
caliper.log	420.8 kB	20:06	☆
networks	1 item	18:28	☆
package.json	284 bytes	18:28	☆
package-lock.json	365 bytes	18:28	☆
report.html	13.1 kB	18:28	☆
src	1 item	18:28	☆

Fig. 6.22 Directory structure with report.html

of the blockchain network. Transaction send rate in TPS presents the number of transactions sent in the network during the benchmark test (Fig. 6.23).

Latency is defined as the time taken to process and commit a transaction on the ledger and throughput on the other hand focuses on the number of transactions processed per second. The report shows a maximum latency of 7.24 s, a minimum latency of 0.74 s, and an average latency of 5.53 s. The throughput of the test is 26.7. Similarly, the execution of the other labels gives the performance status which can be graphically represented for reporting. From this report, it can be concluded that the network performance can be improved by maintaining the latency and improving the throughput by increasing the workload and transactions (Fig. 6.24).

With the increase in the advocacy of blockchain for different enterprise-grade applications and new technologies like IoT, Artificial Intelligence, and others efficient resource utilization is an obligation (Baliga et al. 2018). The resource utilization report pro vides information regarding the resources used in building, deploying, and exploiting the Fabric blockchain network.


Hyperledger Caliper
file:///home/afn/Blockchain/hcaliper/ContainerisingCaliperForFabricBenchmark/caliper-benchmarks-local/report.html
Installing and...
HYPERLEDGER
CALIPER

Basic information

DLT: fabric
Name: simple
Description: This is an example benchmark for caliper, to test the backend DLT's performance with simple account opening & querying transactions
Benchmark Rounds: 1
Details

Benchmark results

Summary
open

System under test

Version: 2.0.0
Size: 2 Orgs with 1 Peer
Orderer: Raft
Distribution: Single Host
StateDB: CouchDB
Details

Caliper report

Summary of performance metrics

Name	Succ	Fail	Send Rate (TPS)	Max Latency (s)	Min Latency (s)	Avg Latency (s)	Throughput (TPS)
open	400	0	40.1	7.24	0.74	5.53	26.7

Benchmark round: open

Test description for the opening of an account through the deployed chaincode

```
rateControl:
  type: fixed-rate
  opts:
    tps: 40
```

Performance metrics for open

Name	Succ	Fail	Send Rate (TPS)	Max Latency (s)	Min Latency (s)	Avg Latency (s)	Throughput (TPS)
open	400	0	40.1	7.24	0.74	5.53	26.7

Resource utilization for open

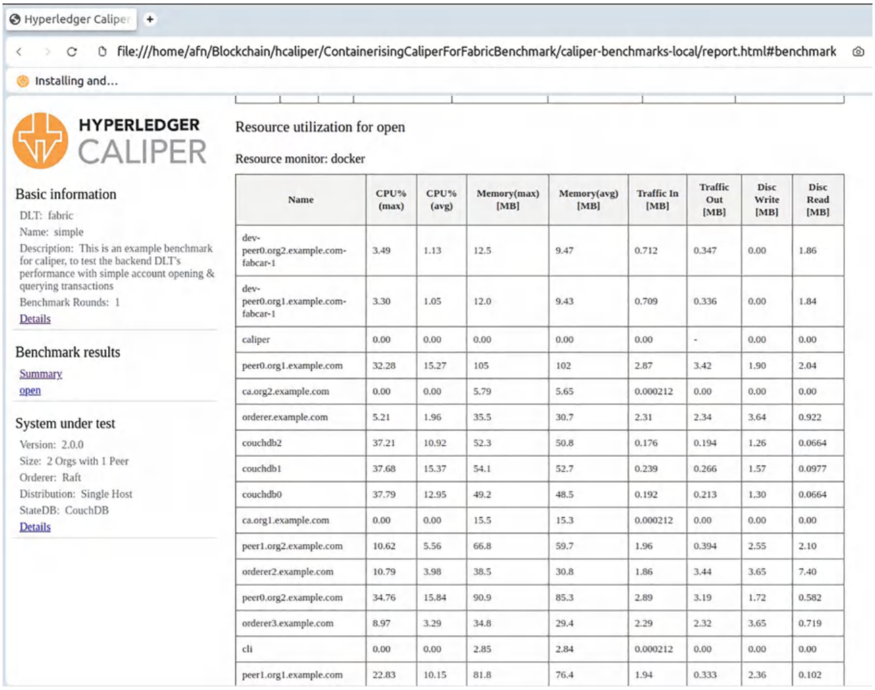
Resource monitor: docker

Name	CPU% (max)	CPU% (avg)	Memory(max) [MB]	Memory(avg) [MB]	Traffic In [MB]	Traffic Out [MB]	Disc Write [MB]	Disc Read [MB]

Fig. 6.23 Performance report for open-label

This report is essential as it contributes to resource planning and understanding the areas where the resources have been underutilized or overutilized. This report also helps to comprehend whether the network can be scaled with the available re sources. Further, how the capacity of resources should be efficiently increased to accommodate the future requirements of the network under study. The resources that are discussed in this report are distributed among the clients, peers, Certificate Authorities, Orderers, and Database including:

- CPU utilization is essential for understanding the computational power re quired for processing the transactions on the network. The report shows the maximum percentage of CPU resources available and how much it is used.
- Storage Space is critical to ensure persistence in the ledgers across all the peers. Storage capacity should be efficiently utilized for storing the encrypted transaction data, increasing the performance of the blockchain network in terms of accessibility and maintaining auditable records to ensure transparency.
- Traffic analysis shows the scalability power of the blockchain network. It shows the channel performance in various loads and the rate at which the data is transferred within the network. The disc read and write refers to storing and retrieving the data from the storage devices that maintain the ledger.


Hyperledger Caliper
file:///home/afn/Blockchain/hcaliper/ContainerisingCaliperForFabricBenchmark/caliper-benchmarks-local/report.html#benchmark
Installing and...
HYPERLEDGER
CALIPER

Basic information

DLT: fabric
Name: simple
Description: This is an example benchmark for caliper, to test the backend DLT's performance with simple account opening & querying transactions
Benchmark Rounds: 1
Details

Benchmark results

Summary
open

System under test

Version: 2.0.0
Size: 2 Orgs with 1 Peer
Orderer: Raft
Distribution: Single Host
StateDB: CouchDB
Details

Resource utilization for open

Resource monitor: docker

Name	CPU% (max)	CPU% (avg)	Memory(max) [MB]	Memory(avg) [MB]	Traffic In [MB]	Traffic Out [MB]	Disc Write [MB]	Disc Read [MB]
dev-peer0.org2.example.com-fabcar-1	3.49	1.13	12.5	9.47	0.712	0.347	0.00	1.86
dev-peer0.org1.example.com-fabcar-1	3.30	1.05	12.0	9.43	0.709	0.336	0.00	1.84
caliper	0.00	0.00	0.00	0.00	0.00	-	0.00	0.00
peer0.org1.example.com	32.28	15.27	105	102	2.87	3.42	1.90	2.04
ca.org2.example.com	0.00	0.00	5.79	5.65	0.000212	0.00	0.00	0.00
orderer.example.com	5.21	1.96	35.5	30.7	2.31	2.34	3.64	0.922
couchdb2	37.21	10.92	52.3	50.8	0.176	0.194	1.26	0.0664
couchdb1	37.68	15.37	54.1	52.7	0.239	0.266	1.57	0.0977
couchdb0	37.79	12.95	49.2	48.5	0.192	0.213	1.30	0.0664
ca.org1.example.com	0.00	0.00	15.5	15.3	0.000212	0.00	0.00	0.00
peer1.org2.example.com	10.62	5.56	66.8	59.7	1.96	0.394	2.55	2.10
orderer2.example.com	10.79	3.98	38.5	30.8	1.86	3.44	3.65	7.40
peer0.org2.example.com	34.76	15.84	90.9	85.3	2.89	3.19	1.72	0.582
orderer3.example.com	8.97	3.29	34.8	29.4	2.29	2.32	3.65	0.719
cli	0.00	0.00	2.85	2.84	0.000212	0.00	0.00	0.00
peer1.org1.example.com	22.83	10.15	81.8	76.4	1.94	0.333	2.36	0.102

Fig. 6.24 Resource utilization report

6.7 Conclusion

Hyperledger, being an open-source and collaborative approach, has gained significant advancement in cross-industry blockchain technology like health care, finance, supply chain, governance, and many more. The support for modularity and the ability to build private and permissioned networks for enterprise-grade applications has shown remarkable demand globally. Many libraries, tools, and frameworks are developed under Hyperledger to address the specific use cases and requirements. In this Chapter, the overall implementation of the private and consortium blockchain networks is demonstrated with the help of Hyperledger Fabric, Hyperledger Explorer, and Hyperledger Caliper. Each has a role in deploying a private and consortium blockchain network. Using the Hyperledger Fabric framework, a private and consortium blockchain network is efficiently designed and deployed. The different stages of the lifecycle of the chaincode on the channel are observed and presented. The Hyperledger Explorer, though it is deprecated, efficiently visualized the network and provided an overall presentable view of the network. With the help of the Hyperledger

Caliper benchmark tool, test benchmarks are set that effectively evaluate the performance of the blockchain network designed and deployed. Code samples covering the full blockchain network implementation are used to illustrate the configurations needed for the framework and tools.

References

About Hyperledger. Accessed 23 May 2024. [Online]. Available: https://www.hyperledger.org/about

Alom I, Ferdous MS, Chowdhury MJM (2022) BlockMeter: an application agnostic performance measurement framework for private blockchain platforms [Online]. Available: http://arxiv.org/abs/2202.05629

Al-Sumaidaee G, Alkhudary R, Zilic Z, Swidan A (2023) Performance analysis of a private blockchain network built on Hyperledger Fabric for healthcare. Inf Process Manag 60(2). https://doi.org/10.1016/j.ipm.2022.103160

Baliga A, Solanki N, Verekar S, Pednekar A, Kamat P, Chatterjee S (2018) Performance characterization of hyperledger fabric. In: Proceedings—2018 crypto valley conference on blockchain technology, CVCBT 2018. Institute of Electrical and Electronics Engineers Inc., pp 65–74. https://doi.org/10.1109/CVCBT.2018.00013

Dabbagh M, Kakavand M, Tahir M, Amphawan A (2020) Performance analysis of blockchain platforms: empirical evaluation of hyperledger fabric and ethereum. In: IEEE international conference on artificial intelligence in engineering and technology, IICAIET 2020. Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/IICAIET49801.2020.9257811

Di Francesco Maesa D, Mori P (2020) Blockchain 3.0 applications survey. J Parallel Distrib Comput 138:99–114. https://doi.org/10.1016/j.jpdc.2019.12.019

Foschini L, Gavagna A, Martuscelli G, Montanari R (2020) Hyperledger fabric blockchain: chaincode performance analysis. In: IEEE international conference on communications. Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/ICC40277.2020.9149080

Geyer F et al. (2019) Performance perspective on private distributed ledger technologies for industrial networks. In: 2019 international conference on networked systems (NetSys), pp 1–8. https://doi.org/10.1109/NetSys.2019.8854512

Ghode DJ, Jain R, Soni G, Singh SK, Yadav V (2020) Architecture to enhance transparency in supply chain management using blockchain technology. In: Procedia Manufacturing. Elsevier B.V., pp 1614–1620. https://doi.org/10.1016/j.promfg.2020.10.225

Hyperledger Caliper Project. Accessed 02 June 2024. [Online]. Available: https://github.com/hyperledger/caliper

Hyperledger Explorer Documentation Explorer, 2021. Accessed 01 June 2024. [Online]. Available: https://www.hyperledger.org/projects/explorer

Hyperledger-fabricdocs Documentation Release master hyperledger, 2021. Accessed: 01 June 2024. [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/latest/index.html

Iftikhar MZ et al (2020) Efficient resource utilization using blockchain network for IoT devices in smart city. In: Lecture notes in networks and systems, vol 97. Springer, pp 521–534. https://doi.org/10.1007/978-3-030-33506-9_47

Javaid M, Haleem A, Pratap Singh R, Khan S, Suman R (2021) Block-chain technology applications for Industry 4.0: a literature-based review. Elsevier Ltd. https://doi.org/10.1016/j.bcra.2021.100027

Khettry AR, Patil KR, Basavaraju AC (2021) A detailed review on blockchain and its applications. Springer. https://doi.org/10.1007/s42979-020-00366-x

Khezr S, Moniruzzaman M, Yassine A, Benlamri R (2019) Blockchain technology in healthcare: a comprehensive review and directions for future research. Appl Sci (Switzerland) 9(9). https://doi.org/10.3390/app9091736

Li D, Wong WE, Guo J (2020) A survey on blockchain for enterprise using hyperledger fabric and composer. In: Proceedings—2019 6th international conference on dependable systems and their applications. DSA 2019. Institute of Electrical and Electronics Engineers Inc., pp 71–80. https://doi.org/10.1109/DSA.2019.00017

Liang W, Tang M, Long J, Peng X, Xu J, Li KC (2019) A secure Fabric blockchain-based data transmission technique for industrial internet-of-things. IEEE Trans Industr Inform 15(6):3582–3592. https://doi.org/10.1109/TII.2019.2907092

Manevich Y, Barger A, Tock Y (2018) Poster: Service discovery for hyperledger fabric. In: DEBS 2018—Proceedings of the 12th ACM international conference on distributed and event-based systems. Association for Computing Machinery, Inc., pp 226–229. https://doi.org/10.1145/3210284.3219766

Mostarda L, Pinna A, Sestili D, Tonelli R, Performance analysis of a BESU permissioned blockchain

Nguyen MQ, Loghin D, Dinh TTA (2021) Understanding the scalability of hyperledger fabric. [Online]. Available: http://arxiv.org/abs/2107.09886

Pongnumkul S, Siripanpornchana C, Thajchayapong S (2017) Performance analysis of private blockchain platforms in varying workloads. In: 2017 26th international conference on computer communication and networks (ICCCN). Vancouver, BC, Canada, pp 1–6. https://doi.org/10.1109/ICCCN.2017.8038517

Ravi D, Ramachandran S, Vignesh R, Falmari VR, Brindha M (2022) Privacy preserving transparent supply chain management through hyperledger fabric. Blockchain: Res Appl 3(2):100072. https://doi.org/10.1016/J.BCRA.2022.100072

Salah D, Ahmed MH, Eldahshan K (2020) Blockchain applications in human resources management: opportunities and challenges. In: ACM international conference proceeding series, pp 383–389. https://doi.org/10.1145/3383219.3383274

Surjandari I, Yusuf H, Laoh E, Maulida R (2021) Designing a permissioned blockchain network for the halal industry using hyperledger fabric with multiple channels and the raft consensus mechanism. J Big Data 8(1). https://doi.org/10.1186/s40537-020-00405-7

Sutradhar S, Karforma S, Bose R, Roy S, Djebali S, Bhattacharyya D (2023) Enhancing identity and access management using hyperledger fabric and OAuth 2.0: A Blockchain-Based Approach for Security and Scalability for Healthcare Industry. [Online]. Available: https://ssrn.com/abstract=4445894

Ucbas Y, Eleyan A, Hammoudeh M, Alohaly M (2023) Performance and scalability analysis of ethereum and hyperledger fabric. IEEE Access 11:67156–67167. https://doi.org/10.1109/ACCESS.2023.3291618

Verma A et al (2022) Blockchain for industry 5.0: vision, opportunities, key enablers, and future directions. IEEE Access 10:69160–69199. https://doi.org/10.1109/ACCESS.2022.3186892

Zhong B, Wu H, Ding L, Luo H, Luo Y, Pan X (2020) Hyperledger fab- ric-based consortium blockchain for construction quality information management. Front Eng Manag 7(4):512–527. https://doi.org/10.1007/s42524-020-0128-y

Zhu C, Li J, Zhong Z, Yue C, Zhang M (2023) A survey on the integration of blockchains and databases. Springer. https://doi.org/10.1007/s41019-023-00212-z

Chapter 7
Implementing Flexible Business Processes with Algorand Blockchain

Nawaz Abdullah Malla, Khaleel Ahmad, Jameel Ahamed, and Halimjon Khujamatov

Abstract Blockchain technology has emerged as a promising solution for implementing business processes in a decentralized and transparent manner. Algorand, a new-generation blockchain platform, offers innovative features such as updating and deleting smart contracts making it well-suited for deploying business processes on a distributed network, enabling the adaptation of evolving conditions and requirements. This chapter aims to provide readers with an overview of business processes and the key characteristics of the Algorand blockchain that enable the effective implementation of these processes. By exploring a use case, of a pizza delivery smart contract, we demonstrate how Algorand's unique properties, such as smart contract updation and deletions, scalability, security, and smart contract capabilities, can be leveraged to create efficient and reliable business process solutions. The chapter serves as an introduction to the potential of Algorand blockchain in revolutionizing the way organizations collaborate and execute their business processes in a decentralized environment.

Keywords Algorand · Smart contracts · Business processes · Blockchain · DApps · Flexibility

N. A. Malla (✉)
University of Camerino, Camerino, Italy
e-mail: nawaz.malla@unicam.it

K. Ahmad · J. Ahamed
Department of Computer Science and Information Technology, Maulana Azad National Urdu University, Hyderabad, India
e-mail: khaleelahmad@manuu.edu.in

J. Ahamed
e-mail: jameel@manuu.edu.in

H. Khujamatov
Department of Computer Engineering, Gachon University, Seongnam, South Korea
e-mail: kh.khujamatov@gmail.com

K. Ahmad et al. (eds.), *Fostering Machine Learning and IoT for Blockchain Technology*, Transactions on Computer Systems and Networks,
https://doi.org/10.1007/978-981-96-4078-2_7

7.1 Introduction

In today's evolving business landscape, organizations are increasingly seeking innovative ways to streamline their operations and collaborate with partners across geographical boundaries.

Conventional business practices typically rely on intermediaries or third-parties and centralized systems, which can lead to inefficiencies, a lack of transparency, and issues with stakeholder confidence. The introduction of blockchain technology (Nakamoto 2008) has come up with new features for addressing these problems and changing how businesses operate. The next-generation blockchain platform Algorand (Gilad et al. 2017) has garnered a lot of attention because of its unique features, which include scalability and transaction finality, making it perfect for carrying out business processes in a decentralized environment (Sánchez Cano 2020). The scalability, security, and smart contract capabilities of Algorand allow businesses to design dependable and effective solutions for carrying out their business operations across a dispersed network.

The objective of the chapter is to provide readers with a general understanding of business processes and the key characteristics of the Algorand blockchain that facilitate the implementation of these processes. By exploring a use case, we demonstrate how Algorand's innovative features can be leveraged to create transparent, tamper-proof, and collaborative business process solutions.

The chapter is organized as follows: Sect. 7.2 introduces the concept of business processes and their importance in the modern business landscape. Section 7.3 delves into the key features of the Algorand blockchain and its suitability for implementing business processes. Section 7.4 presents the related works about smart contract upgradeability. Section 7.5 presents a use case that showcases the application of the Algorand blockchain in a real-world business scenario. Finally, Sect. 7.6 concludes the chapter and highlights future research directions in the field of blockchain-based business process management.

7.2 Introduction to Business Processes in the Modern Business Landscape

Nowadays, in the modern fast-paced, and interconnected world, businesses operate in complex environments where efficiency and adaptability are crucial for success. At the core of every organization's operations are business processes, which define how tasks are executed, information flows, and decisions are made to achieve specific goals and deliver value to customers (Online available n.d.). Understanding the concept of business processes and their importance is essential for organizations looking to optimize their operations, enhance productivity, and stay competitive in the modern business landscape.

7.2.1 What Are Business Processes?

Business processes are defined as a set of interconnected activities or tasks that are performed in a coordinated manner to achieve a specific objective or defined business outcome. These processes may involve people as well as various functions, departments, and systems within an organization, including technology, and resources to deliver products or services to customers (Fettke et al. 2006). Business processes can be divided into three broad categories. Operational processes which include manufacturing, sales, and customer service. Management processes include strategic planning, budgeting, and performance management and support processes provide the human resources, IT support, and procurement.

7.2.2 Importance of Business Processes

- Productivity and Efficiency: Streamlining operations, getting rid of duplication, and cutting inefficiencies all contribute to increased productivity and cost savings when business processes are well-defined and optimized.
- Uniformity and Quality: Standardized processes improve customer satisfaction and brand reputation by ensuring uniformity in operations, product/service quality, and customer experience.
- Risk Management and Compliance: Robust corporate procedures support the identification and management of risks, guarantee adherence to regulations, and improve internal governance and transparency.
- Innovation and Agility: In a dynamic corporate environment, agile business processes help firms react swiftly to shifts in the market, customer needs, and competitive challenges.
- Data-Driven Decision-Making: Information and insights from business processes are leveraged to promote continuous improvement, make well-informed decisions, and maximize performance throughout the company.
- Customer Focus: Organizations can offer tailored, value-added solutions that increase customer loyalty and retention by aligning business operations with customer requirements and expectations through customer-centric procedures.

Organizations that prioritize effective, customer-centric business processes are better positioned to navigate challenges, seize opportunities, and achieve sustainable growth (Sustainability Online n.d.) and success in the modern business landscape, where digital transformation, globalization, and disruptive technologies are reshaping industries. Through a comprehensive comprehension of the importance of business processes and their successful utilization, enterprises may augment operational excellence, stimulate innovation, and generate value for all parties concerned.

7.2.3 Flexibility in Business Process

Flexibility in business processes refers to the ability to adapt to changes without losing its identity. This means that a business process is considered flexible if it can be modified without being completely replaced. Business process flexibility is the capacity to implement changes in the process type and instances by altering only the necessary parts while maintaining stability in other areas (Regev et al. 2006).

Flexibility in business processes can be understood in various ways. Since flexibility entails the ability to change, it can be categorized based on the types of changes it allows. The business process flexibility taxonomy, shown in Fig. 7.1, includes three orthogonal dimensions: the abstraction change, the perspective of change, and the properties of change, which encompass extent, duration, swiftness, and anticipation. Each of these dimensions is explained in the following sections.

Abstraction Level: Flexibility can be categorized based on the level of abstraction at which changes are made. This includes changes at the process type, instance, or specific task levels.

Changes at the process type level involve modifying the overall process model. This includes changes to the process definition, such as adding or removing tasks,

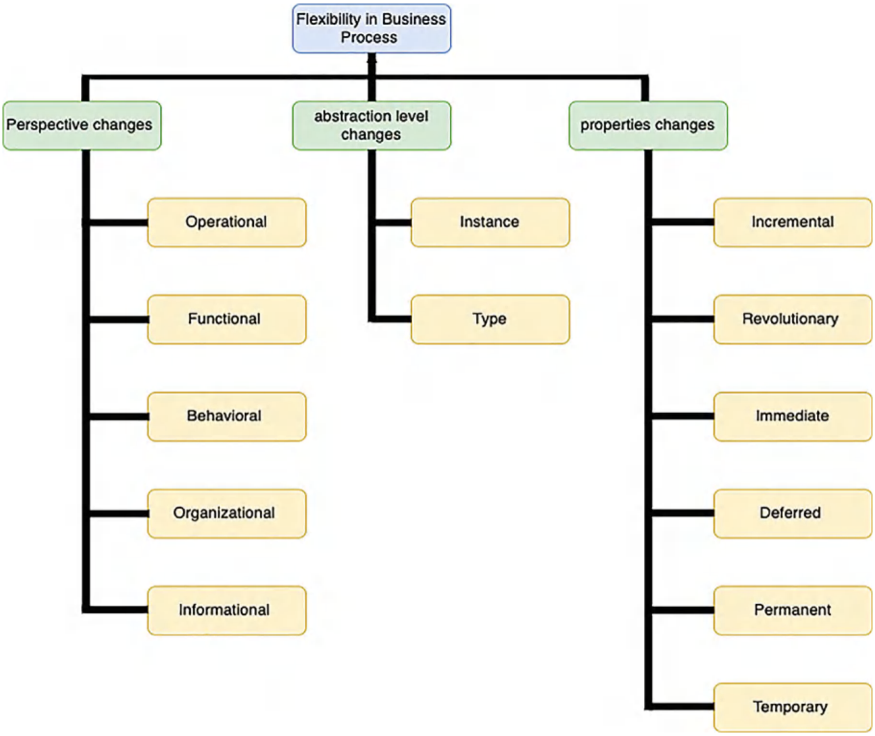


Fig. 7.1 Type of changes in business process

adjusting the workflow, or altering the process structure. These changes are typically made at design time and affect all instances of the process (Schonenberg et al. 2008a).

Changes at the instance level involve modifying a specific process instance. This includes changes to the execution of a particular process instance, such as adjusting the order of tasks or adding new tasks. These changes are typically made at runtime and affect only the specific process instance being executed (Schonenberg et al. 2008b).

Perspective of Change: Flexibility can be classified based on the perspective of change, which includes changes to the operational, behavioral, organizational, informational, and functional structure of the business rules.

Properties of Change: Flexibility can be characterized by the properties of the change, such as Extent: The scope of the change, including the number of tasks or activities affected. Duration: The time frame within which the change is implemented. Swiftness: The speed at which the change is executed. Anticipation: The ability to anticipate and prepare for changes before they occur.

Moreover, Business process flexibility is crucial for organizations to remain competitive and adapt to changing market conditions. It enables them to respond quickly to customer demands, changes in the supply chain, and other environmental variations. By understanding the different dimensions of flexibility, organizations can design and manage their processes to be more adaptable and resilient.

7.2.4 Benefits of Integrating Blockchain Technology into BPM Systems

Trusted relationship without a third party: Blockchain technology (BCT) plays a crucial role in establishing trusted relations among services in Industry 4.0 by leveraging Quality of Service (QoS) values. This is achieved without the need for trusted intermediaries, which simplifies processes and reduces potential points of failure.

Cost-effective: BCT also offers significant cost reductions by eliminating manual operations and fees traditionally paid to intermediaries. Additionally, there is no need for central authorities to maintain QoS values, and the management of attribute-based digital certificates can be omitted, leading to operational and economic savings.

Fast transactions: The automation of service selection and composition through BCT enables real-time updates and verifications of QoS values, accelerating transactions and making business processes more agile and responsive to changes. These benefits collectively enhance the interoperability of cross-organizational business processes in Industry 4.0, improving efficiency, agility, and compliance while reducing the need for manual intervention and the risks associated with centralized systems.

7.3 Algorand and Its Features

In this section, we present the basic concepts behind blockchain technology and then we move to the Algorand blockchain characteristics, with a focus on its delete and update features of smart contracts.

The Algorand blockchain is a decentralized record-keeping system that uses blocks linked together through cryptography to prevent data tampering (Nakamoto 2008). Each block includes a cryptographic hash of the preceding block, timestamp, and transaction details. Due to its decentralized and distributed nature, modifying any block would impact all subsequent blocks (Rajasekaran et al. 2022).

Algorand (Chen et al. 1607), an open and permissionless blockchain, employs the Byzantine agreement protocol to handle malicious nodes. It uses a proof-of-stake consensus protocol, which requires minimal computation power, making it fast and efficient in finalizing transactions. Unlike other blockchain platforms, forking is almost impossible in Algorand, as it leverages a voting mechanism to validate blocks and uses a pure proof of stake consensus mechanism (Gilad et al. 2017).

Algorand's unique consensus mechanism guarantees transaction finality thus making forking impossible, as it uses a voting process to validate blocks. This feature allows Algorand to produce blocks every 3.3 s, with each block capable of holding up to 25,000 transactions. Finally results in a throughput of 7500 transactions per second (7500TPS) (Algorand 2023). Moreover, like other blockchains, Algorand has its own native cryptocurrency called Algo. To engage in consensus, develop applications, pay transaction fees, or store data on the Algorand blockchain, users must hold enough Algorand tokens to pay the transaction fee and meet the minimum balance requirement.

Algorand supports two types of smart contracts: stateless smart contracts (also known as smart signatures) and stateful smart contracts. Stateless smart contracts are used for signing transactions and provide a mechanism for delegating signature authority to another account. Stateful smart contracts, on the other hand, refer to applications that reside on the blockchain with a specific application ID.

Data on the Algorand blockchain is stored in accounts using an account-based model (Algorand smart contracts n.d.). Further, these accounts also keep the local state of an account, Algo balances, and asset balances. To use local state, an account must opt-in by signing and submitting a transaction Application Call of type Opt-In. The local state is only visible once an account has opted into the application. Algorand accounts are classified into three categories.

- **Application Accounts**: Application accounts are associated with the Algorand addresses and the application ID which are controlled by the stateful smart contracts. These application accounts are like Ethereum smart contracts. The spending from the application account depends on the logic of the smart contracts rather than the private keys.
- **Signature Accounts**: Signature accounts are represented by a unique Algorand address which is obtained by hashing the smart signature control using the logic signature for controlling the spending behavior (Algorand smart contracts n.d.).

The signature accounts are used for conditional spending scenarios and enhanced privacy.

- **External Accounts**: The external accounts are associated with public keys and private keys that can be transformed into Algorand addresses to sign the transactions. External accounts are the same as externally owned accounts in Ethereum.

Moreover, Algorand allows the submission of a group of different types of transactions together at one time, called atomic transactions. It succeeds only if all transactions in the group are successful (Atomic: atomic transfer n.d.). By leveraging these key features, organizations can implement efficient, secure, and transparent business processes on the Algorand blockchain. The platform's scalability, smart contract capabilities, and asset management functionalities enable the creation of innovative solutions that address the challenges faced by traditional centralized systems.

7.4 Key Features of Algorand Blockchain for Implementing Business Processes

The Algorand blockchain offers several innovative features that make it well-suited for implementing business processes in a decentralized environment. These features include:

Scalability and Performance: Algorand uses a unique consensus mechanism called Pure Proof of Stake (PPoS) that enables high transaction throughput and low latency, making it suitable for real-time business applications that require fast and efficient processing of transactions. The platform can handle thousands of transactions per second with near-instant finality, ensuring that business processes can be executed in an efficient manner.

Smart Contract Capabilities: Algorand offers a robust framework for smart contracts that enables developers to put intricate business logic on the blockchain. Algorand's low-level language, TEAL (Transaction Execution Approval Language), or PyTeal (PyTeal: Algorand Smart Contracts in Python n.d.), a domain-specific language based on Python, can be used to write smart contracts. These smart contracts can be used to automate a number of business process activities, including asset transfers, contract execution, and decision-making.

Smart Contract Upgradeability: While smart contracts are immutable by design to ensure trustlessness, there are techniques to enable upgrades in a controlled manner. Some approaches include: Using a proxy contract (Bodell et al. 2023) that forwards a call to an upgradeable implementation contract Storing contract logic in external libraries that can be swapped out Allowing the contract owner to selectively enable/disable certain functions Upgradeability comes with trade-offs around security and trust assumptions (Salehi et al. 2022). Careful design is needed to preserve the benefits of immutability while enabling necessary changes.

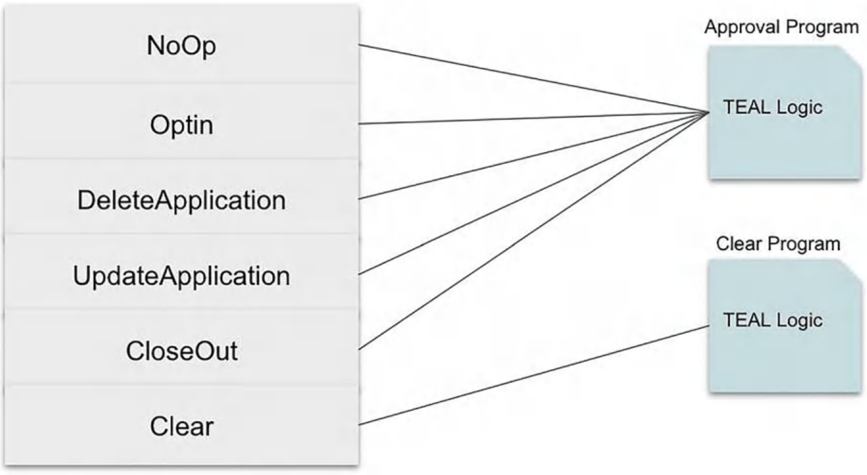


Fig. 7.2 Overview of Algorand smart contract

Figure 7.2 illustrates the components and transactions involved in Algorand smart contracts.[1] These contracts are implemented using two programs: the *ApprovalProgram* and the *ClearStateProgram*. The *ApprovalProgram* handles all application calls to the contract, whereas the *ClearStateProgram* is used to remove the smart contract from an account's balance record when a clear call is made.

The *UpdateApplication* transaction type is used to update TEAL programs for a contract, which can be created using either the goal command or the SDKs.

Interoperability: Algorand's interoperability with legacy systems and other blockchain platforms allows for a smooth integration with the current infrastructure of businesses. This feature allows organizations to leverage the benefits of blockchain technology while maintaining compatibility with their current systems and processes.

Security and Immutability: The Algorand blockchain is designed to be secure and tamper-resistant, ensuring that business processes executed on the platform are protected from unauthorized modifications or interference. The platform uses advanced cryptographic techniques, such as verifiable random functions (VRFs) and digital signatures, to ensure the integrity and confidentiality of transactions and data stored on the blockchain.

Decentralization and Transparency: Algorand is a decentralized platform that functions without a central authority like other decentralized networks, enabling cooperation and the transparent execution of business activities by several parties. Algorand blockchain's immutable nature guarantees that all transactions and process stages are recorded and subject to audits by authorized parties, thus enhancing transparency and accountability.

Asset Creation and Management: Algorand facilitates the development and administration of personalized digital assets, referred to as Algorand Standard Assets

[1] https://developer.Algorand.org/docs/get-details/dapps/smart-contracts/apps/.

(ASAs) (Algorand standard Assests n.d.). These assets can be incorporated into business processes to enable transactions and asset transfers. They can represent several kinds of value, including digital rights, loyalty points, and fiat currency.

7.5 Related Works

The scientific community is delving deeply into Algorand's ability to solve issues that other blockchain platforms are facing. This includes in-depth analyses of its underlying protocol and architecture (Chen et al. 2018), as well as a study on its potential to resolve the blockchain trilemma, an issue that has garnered a lot of attention from researchers. Numerous research works have been carried out on the security features of Algorand smart contracts, emphasizing their semantics and pinpointing different kinds of vulnerabilities. For example, Bartoletti et al. demonstrated the fundamental features of the Algorand blockchain through rigorous proofs (Bartoletti et al. 2021), while Sun et al. developed the Panda framework (Sun et al. 2023) to automatically find weaknesses.

Alturki developed a comprehensive model, providing formal proof of its asynchronous safety for the consensus protocol (Alturki et al. 2019). Benhamouda et al. analyzed Al-Gorand's protocol using a custom network model designed to address concerns of protocol designers, particularly focusing on recovery from intermittent network disruptions (Sánchez Cano 2020). D'Onfro conducted a critical evaluation of the viability of smart contracts in consumer protection laws, noting issues with post-execution adjustments, industry resistance, and coding hurdles for compliance (D'Onfro 2020).

Abbasi et al. highlighted the limitations of the current transaction fee model of Algorand Blockchain and proposed a competitive market for Algorand block space, thereby determining the optimal transaction fee and block size (Abbasi et al. 2022). Šljukić et al. developed a model for conducting auctions on the Algorand blockchain to rent real estate properties owned by municipalities (Šljukić et al. 2022).

The implementation of business processes on the Algorand blockchain presents challenges that have not been thoroughly explored, despite extensive research on smart contract security and performance metrics within the Algorand ecosystem. This study aims to introduce and explore the complexities involved in integrating business processes into the Algorand blockchain, with the objective of gaining a comprehensive understanding and contributing valuable insights to this under-examined area of research.

7.6 Use Case: Pizza Delivery on the Algorand Blockchain

To showcase the application of the Algorand blockchain in a real-world business scenario, let's consider a simple pizza delivery process involving a customer, a restaurant, and a delivery boy. We can illustrate how the characteristics of the platform may be utilized to develop a transparent, safe, and effective solution by putting this approach into practice on the Algorand blockchain.

Using the Beaker library, the Pizza Delivery State smart contract offers a complete management solution for the pizza delivery process. Customers can place orders through this contract, which also designates a delivery boy and keeps track of the order's progress. The contract ensures that only authorized parties can access and update the order information. The place order function enables the customer to place orders by specifying their address and pizza details. The handover pizza function assigns a delivery boy to the order, and the deliver pizza function updates the order status to"delivered" once the pizza is delivered to the customer. The update restaurant function allows the restaurant to update the order status and reset the order details. This smart contract ensures a secure and efficient way to manage pizza deliveries, providing a seamless experience for both customers and restaurants.

7.6.1 Process Flow

The sequence of activities between the different components of the process is shown in Fig. 7.3, it presents the choreography diagram of the pizza delivery process. The choreography diagram displays the interactions between the system components without exposing the internal structure.

Places an order: The customer places an order, selects a pizza from the available options, and chooses the type of delivery either providing the address for delivery or picking up the order from a restaurant.

Restaurant receives the order: The restaurant takes the order details from the blockchain and prepares the pizza accordingly. Once ready, the customer either picks up the order from the restaurant, or the restaurant hands over the pizza to the delivery boy by updating the order status on the blockchain.

Delivering the pizza: The delivery boy can view the order details and the restaurant's confirmation on the blockchain. They then pick up the pizza and deliver it to the customer's address.

Customer receives the pizza: Upon delivery, the customer confirms receipt of the pizza, and the delivery boy updates the order status on the blockchain.

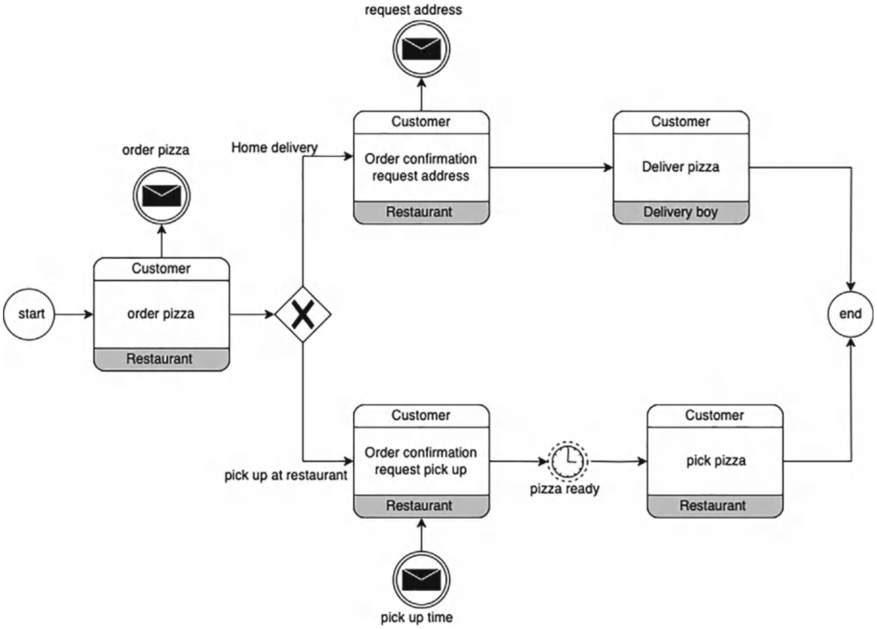


Fig. 7.3 Overview of pizza delivery process

7.7 Conclusion

The Pizza Delivery smart contract concludes by showcasing the transformative power of blockchain technology in the food delivery industry. By leveraging the yTeal library with the Beaker framework and Algorand's smart contract capabilities, this contract provides a robust and secure solution for managing the pizza delivery process. The key features of this smart contract include the smart contract is upgradable to make changes after post-deployment of the smart contract to instances like order placement, delivery boy assignment, order status tracking, and restaurant update capabilities, all of which work together to streamline operations, improve customer satisfaction, and reduce the risk of fraud or mishandling of orders according to changing conditions. As the food delivery industry continues to evolve, the Pizza Delivery State smart contract serves as a compelling example of how blockchain-based solutions can transform traditional business processes, paving the way for a more efficient, secure, and customer- centric future.

Acknowledgements Malla is a student of the national PhD in Blockchain and DLT (University of Camerino, Italy) supported by a PNRR scholarship D.M. 351/2022.

References

Nakamoto S(2008) Bitcoin: a peer-to-peer electronic cash system. Decentralized business review

Gilad Y, Hemo R, Micali S, Vlachos G, Zeldovich N (2022) Algorand: Scaling byzantine agreements for cryptocurrencies. In: Proceedings of the 26th symposium on operating systems principles. Springer, Heidelberg, pp 51–68

Sánchez Cano JE (2020) The technological innovation of the blockchain and its impact on the energy sector. Panorama economico (Ciudad de Me´xico) 16(31):157–178.

Online available. https://www.villanovau.com/articles/bpm/what-is-a-business-process/

Fettke P, Loos P, Zwicker J (2006) Business process reference models: survey and classification. In: Bussler CJ, Haller A (eds) Business process management workshops. BPM 2005. Lecture notes in computer science, vol 3812. Springer, Berlin, P. 44. https://doi.org/10.1007/11678564

Sustainability, Online. https://algorandtechnologies.com/news/algorands-leadership-in-blockchain-sustainability

Regev G, Soffer P, Schmidt R (2006) Taxonomy of flexibility in business processes. BPMDS, p 236

Schonenberg H, Mans R, Russell N, Mulyar N, van der Aalst WM (2008) Towards a taxonomy of process flexibility. In: CAiSE forum, vol 344, pp 81–84

Schonenberg H, Mans R, Russell N, Mulyar N, van der Aalst W (2008) Process flexibility: a survey of contemporary approaches. In: International workshop on co-operation and interoperability, architecture and ontology. Springer, Berlin, pp 16–30

Rajasekaran AS, Azees M, Al-Turjman F (2022) A comprehensive survey on blockchain technology. Sustain Energy Technol Assess 52:102039. https://www.sciencedirect.com/science/article/pii/S2213138822000911, https://doi.org/10.1016/j.seta.2022.102039

Chen J, Micali S (2016) Algorand. arXiv:1607.01341

Algorand (2023) Why Algorand?—Algorand Developer Portal. https://developer.algorand.org/docs/get-started/basics/whyalgorand/

Algorand smart contracts.” [Online]. Available: https://developer.algorand.org/docs/get-details/dapps/smart-contracts/smart-contracts

Atomic: atomic transfer. [online]. Available: https://developer.algorand.org/docs/get-details/atomic transfers/

PyTeal: Algorand Smart Contracts in Python. [Online]. Available: https://pyteal.readthedocs.io/en/stable/

Bodell III, William E, Meisami S, Duan Y (2023) Proxy hunting: understanding and characterizing proxy-based upgradeable smart contracts in blockchains. In: 32nd USENIX security symposium (USENIX Security 23)

Salehi M, Clark J, Mannan M (2022) Not so immutable: upgradeability of smart contracts on ethereum. arXiv:2206.00716

Algorand standard Assests. [online] available: https://developer.algorand.org/docs/get-details/asa/

Chen J, Gorbunov S, Micali S, Vlachos G (2018) Algorand agreement: superfast and partition resilient byzantine agreement. Cryptology ePrint Archive

Bartoletti M, Bracciali A, Lepore C, Scalas A, Zunino R (2021) A formal model of algorand smart contracts. In: Financial cryptography and data security: 25th international conference, FC 2021, Virtual Event, Revised Selected Papers, Part I 25. Springer, pp 93–114

Sun Z, Luo X, Zhang Y (2023) Panda: Security analysis of algorand smart contracts. In: 32nd USENIX security symposium (USENIX Security 23), pp 1811–1828

Alturki MA, Chen J, Luchangco V, Moore B, Palmskog K, Peña L, Roşu G (2020) Towards a verified model of the algorand consensus protocol in coq. In: Formal methods. FM 2019 international workshops. Porto, Portugal, Revised Selected Papers, Part I 3, Springer, pp 362–367

D’Onfro D (2020) Smart contracts and the illusion of automated enforcement. Wash. UJL Pol’y 61:173

Abbasi M, Manshaei MH, Rahman MA, Akkaya K, Jadliwala M (2022) On algorand transaction fees: Challenges and mechanism design. In: ICC 2022-IEEE international conference on communications. IEEE, pp 5403–5408

Šljukić M, Labus A, Despotović-Zrakić M, Naumović T, Bogdanović Z (2023) A model for municipality buildings renting auction on algorand blockchain. In: Marketing and smart technologies: proceedings of ICMarkTech 2022, vol. 2. Springer, pp 207–218

Chapter 8
Blockchain-Based Lightweight Encrypted Police Management System

Sneha Pandey, Tushar, Divyanshu Kumar, Deepika Kukreja, and Deepak Kumar Sharma

Abstract The number of criminal activities has been rising each year. This issue is not peculiar to any state or country but, rather, has become a global challenge. The reason behind the spike includes many factors such as poverty, unemployment, illiteracy, etc., but the issues present in the existing police management system are also a dominant factor. With advancing technology and digital governance, the Electronic First Information Report (e-FIR) has been introduced in different parts of the world. e-FIR is stored in a centralised system which has various challenges and limitations such as data tampering, lack of accountability, corruption and unreported FIRs. This paper proposes a blockchain-based lightweight encrypted police management system. It covers all aspects of RDBMS right from complaint lodging, FIR registration, distributed database management system and above all a lightweight system. This allows immutable complaint filing by the citizens, which is stored in the blockchain, and can only be accessed by authorised personnel. A decentralised database, InterPlanetary File System (IPFS), is used to store FIR, evidence, chargesheet, etc. Also, to ensure transparency and data integrity of IPFS, logs of all transactions on IPFS are stored in the blockchain. The proposed work compares the time taken by different encryption algorithms to encrypt data and store it on IPFS. In addition, the time trends for various encryption algorithms as file sizes increase have also been examined and presented.

Keywords Blockchain · Police management system · Lightweight system · IPFS · Decentralisation · Encryption · Immutable complaint filing

S. Pandey · Tushar · D. Kumar · D. Kukreja (✉)
Department of Information Technology, Netaji Subhas University of Technology, Delhi, India
e-mail: deepikakukreja18@gmail.com

Tushar
e-mail: tusharvishi2006@gmail.com

D. K. Sharma
Department of Information Technology, Indira Gandhi Delhi Technological University for Women, Delhi, India

K. Ahmad et al. (eds.), *Fostering Machine Learning and IoT for Blockchain Technology*,
Transactions on Computer Systems and Networks,
https://doi.org/10.1007/978-981-96-4078-2_8

8.1 Introduction

The world has seen a lot of advancements in various fields ranging from education, healthcare, transportation and communication to space exploration. But the field of policing has not witnessed any breakthrough advancements. For a very long time, the police department maintained paperwork and documented manually in registers and files. This system had various drawbacks such as lack of storage space, security issues, prone to damage, transportation issues, difficulty in editing, high cost and environmental damage. To address these problems, e-FIR system was introduced. e-FIR is an online document filed to the police stations by a victim or someone on his/her behalf by logging into specified websites when an offence such as murder, kidnapping and theft is committed. It is centralised in nature which means that all the access and power lies completely in the hands of few people. Due to which the following challenges come up:

1. Data Tampering: The current system is vulnerable to data tampering and unlawful changes as police personnel have tremendous power over what goes into the system and can easily change or modify it.
2. Lack of Accountability: The police personnel can't be held accountable as they are in power and can also forge evidence to protect themselves. This system also lacks necessary accountability and many e-FIRs are oftentimes not taken seriously by police and are ignored until someone goes and files an FIR physically.
3. Corruption: Police corruption such as bargaining with criminals for bribes, accepting small gratuities for not registering cases or modifying the complaint details also remains a major issue in the existing centralised infrastructure.
4. Denial to lodge an FIR: The police officers refuse to lodge an FIR against influential people under socio-political pressure.
5. Security Issues: The current system is a centralised system which is prone to various attacks that may damage the integrity and privacy of data such as SQL injection and Distributed Denial-of-service (DDoS) attacks.

This paper proposes a blockchain-based lightweight police management system. It uses a blockchain decentralised ledger and IPFS distributed database. The system is lightweight since only selective data is stored in the blockchain which minimises the computation cost of storing and accessing data. The remaining data is stored on the IPFS. The system also encrypts only those data items that need encryption security as a result the cost of encrypting and decrypting data is also minimised. Our system maintains two entities: Complaint and FIR. A complaint is the first-hand information of an offence that the victim or victim's family members or any witness of the crime submit to the police station in his own words. On the other hand, FIR is a written document prepared by the police when they receive information/ complaints about the commission of any offence. This ensures that no complaint goes unreported. The system achieves data integrity, security, trust and transparency leading to constitutional policing.

8.2 Related Work

Researchers have been working to improve police management architecture by proposing advancements in e-FIR, criminal data storage systems and ways to bridge the gap between citizens and police.

Tasnim et al. (Tasnim et al. 2018) proposes a blockchain-based system for securing criminal data from any lawful changes and unauthorised access. The system stores the criminal data on cloud storage. The read, write and update transactions on cloud storage are stored in the blockchain. The users are pre-registered into the system that allows only government authorities to add and maintain criminal data. The system lacks in addressing the integrity of user data stored on cloud databases.

A system known as the ‘Third Eye’ that connects the home ministry to each police station that exists in a city in Bangladesh (Mollah et al. 2012), which monitors the actions taken by police and related data. Since there exists a central database that is only handled by the home ministry, it is easily vulnerable to manipulation. Another online centralised police complaint management system for Saudi Arabia has been proposed (Tabassum et al. 2018). The system has registered users who can file a complaint. The officers verify the details and register the complaint in the system. They have also maintained criminal data to track the most wanted or top criminals in the country. But as the system is centralised it is prone to data tampering, lack of accountability, corruption and unreported FIRs.

A consensus-based distributed blockchain system that addresses e-FIR data integrity (Khan et al. 2020). Specifically, smart contracts have been used to provide integrity to e-FIR data stored in the database. A blockchain-based smart policing system to boost transparency and accountability has been developed (Mukherjee and Halder 2020). Additionally, it made it possible to exchange information on justice, criminal cases and evidence related to investigations. The system offers all fundamental functions, including FIR filing, investigation tracking, forensic report addition and justice delivery. (Sing et al. 2022) develops a blockchain-based, permissionless, Flutter app to aid in the remote access to criminal records. It is built on Ethereum, Ganache (for testing smart contracts), IPFS (for simple file tracking and protected data storage) and MetaMask (for serving as a digital wallet). Jain et al. (Jain et al. 2021) is a blockchain-based approach to minimise corruption by enabling third-party monitoring of tamper-evident transactions, thereby increasing transparency and accountability. However, because they are not using transaction logs, any unlawful transaction in the system can’t be audited and finding the culprit also becomes a major challenge. A web-based system to manage criminal records in the city of Mangalore is presented (Hingorani et al. 2020). According to the author, the web programme will not only manage criminal records but also assist the police department in identifying regional issues without physically visiting them. However, because criminal records are stored in a central database, the system is vulnerable to data corruption, tampering and unreported FIRs.

8.3 Motivation

The majority of the current systems for managing police complaints are centralised, making them more susceptible to attacks, data manipulation, corruption, lack of accountability and single point of failure. Many systems only keep FIRs and NCRs and not the complaint. This is a problem since complainants won't have evidence that they have filed a complaint if a police officer declines to file an FIR or NCR against powerful individuals. The shortcomings of current methods highlight the need for a transparent system where a user is not required to blindly believe the police department.

8.4 Proposed Work

A blockchain-based lightweight police management system has been devised that accomplishes all the functionalities to assist the police. It stores complaints by the citizens, FIRs registered by the police, criminal records, chargesheets, pieces of evidence, forensic reports and judicial proceedings of ongoing cases. The system is decentralised with the help of blockchain and IPFS. It is made lightweight to reduce the computational cost of blockchain and the space–time complexity of encryption–decryption.

The system can be comprehended in three phases:

1. Blockchain Architecture

The system stores two major entities on the blockchain. First, the complaint was registered by the complainant. Second, it stores the logs of all read, write and update operations on the IPFS.

a. Complaint—The victim, the victim's family members, or any witness can fill out the complaint form. It is first-hand information about the offence and the police file the FIR based on these details. It contains fields such as Complainant Name, Type of Offence, Place, Date, Time, Offence and Suspect Details. Since the complaint is stored on the blockchain, it cannot tamper and the police have to register the FIR corresponding to it without any delay or conditions. This develops accountability and trust between citizens and police.
b. IPFS operation logs—There would be constant read, write and update operations on IPFS, and its log is recorded on the blockchain. The log contains the author, time and commit ID of the operation. The commit ID can be used to get a detailed description of all the changes made to IPFS. This ensures that an immutable change log is made for IPFS, and the reporting officers can be questioned for any unlawful changes. This also allows one to go back in time and check the progression of the case at each change, as well as revert to an earlier state if an error or file corruption occurs. This increases accountability in the police management system.

2. IPFS Architecture

IPFS is used to store FIRs registered by the police, criminal records, chargesheets, pieces of evidence, forensic reports and judicial proceedings of ongoing cases since storing them on the blockchain is a very costly operation due to their large size. This results in a lightweight system.

3. Lightweight Architecture

The system has been designed to be lightweight using two methodologies:

c. Storing selective data on blockchain—The blockchain only stores complaints registered by the user and the logs of all read, write and update operations on the IPFS to reduce the computational cost of the blockchain. (Stoll et al. 2019) claims that, as of November 2018, Bitcoin consumed 45.8 TWh of electricity annually and estimates the range of carbon emissions between 22.0 and 22.9 MtCO2. This indicates that the emissions created by Bitcoin are comparable to those produced by Sri Lanka and Jordan.
d. Encrypting selective data on IPFS—The information on the IPFS is encrypted to protect the privacy and security of the complainant. The encrypted entities are complaint names and complaint details in FIR, pieces of evidence and forensic reports. Furthermore, in some countries like India and Canada, criminal records would also be encrypted. Since the system encrypts only selected items, it helps to reduce the overall time and computation power for encryption and decryption.

There are two major stakeholders in the system—the complainant and the police. The system serves the following functionalities to both stakeholders:

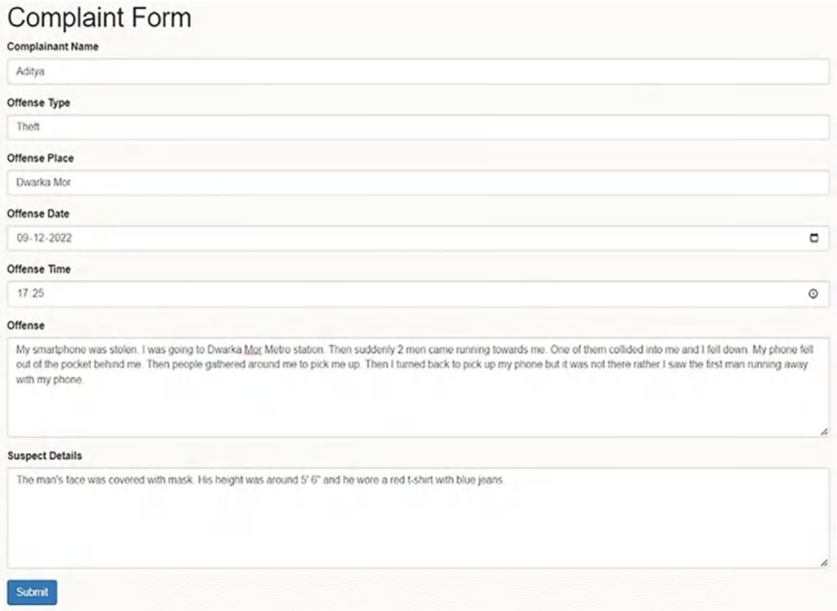
1. *Complainant-side functionality*

   a. Every citizen is registered on the complaint portal using a government-issued unique identification number as in India by Aadhar Card Number and in the U.S. by Social Security Number and can use it to log in to the portal.
   b. The victim, the victim's family members, any witness to the crime or any other person can register a complaint by filling out the complaint form on the portal. (Fig. 8.1)
   c. When the complainant submits the form, the relay captures the form data and sends it to be stored on the blockchain.
   d. The complainant can also upload any pieces of evidence using the same portal.
   e. The portal has a dashboard to view the active complaints, chargesheets and judicial proceedings. The complainant can also track their status.

Figure 8.2 demonstrates the complainant side functionality in detail.

2. *Police-side functionality*

   a. Police officials are registered on the FIR portal using their unique biometric ID and can use it to log in. They remain logged in to the portal so that they can be notified about any updates.



Complaint Form

Complainant Name

Aditya

Offense Type

Theft

Offense Place

Dwarka Mor

Offense Date

09-12-2022

Offense Time

17:25

Offense

My smartphone was stolen. I was going to Dwarka Mor Metro station. Then suddenly 2 men came running towards me. One of them collided into me and I fell down. My phone fell out of the pocket behind me. Then people gathered around me to pick me up. Then I turned back to pick up my phone but it was not there rather I saw the first man running away with my phone.

Suspect Details

The man's face was covered with mask. His height was around 5' 6" and he wore a red t-shirt with blue jeans.

Fig. 8.1 Complaint form

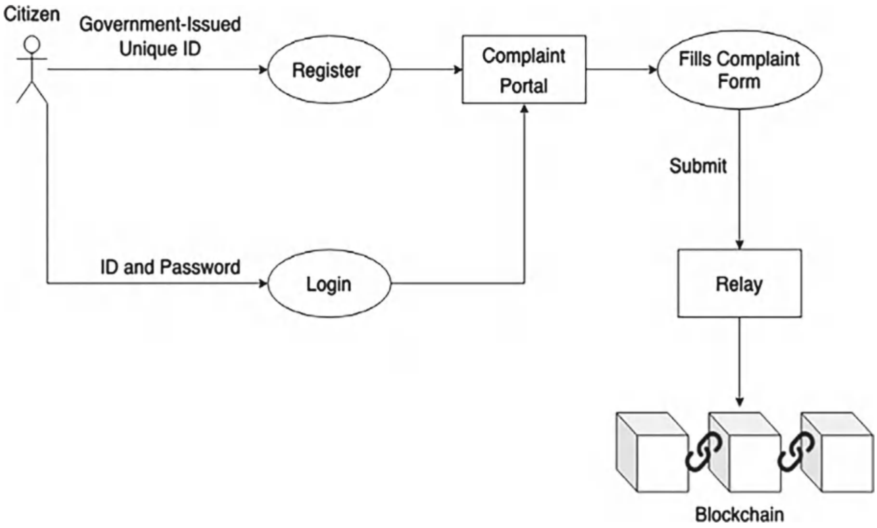
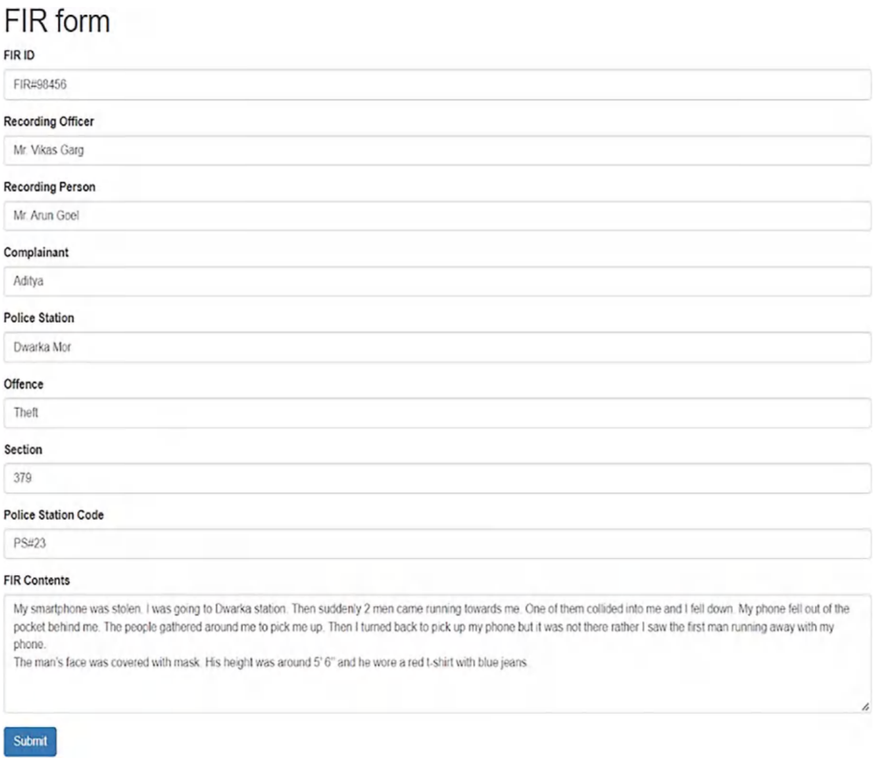


Fig. 8.2 Complainant-side functionality

b. Whenever a new complaint is submitted, the concerned police officer is notified through the FIR portal. The FIR portal and the complaint portal are connected through the relay. So, the new complaint details are shown to the police on this portal.
c. Now, the police officer registers an FIR using the form available on the portal. The officer fills out the FIR form according to the complaint and submits it. (Fig. 8.3)
d. The submitted FIR data is stored on IPFS. Furthermore, the officer can read and update the FIR through the same portal whenever needed.
e. As the investigation proceeds, the police can also store the evidence and forensic reports.
f. Once the criminal is found, his or her details are also stored in the system.

Figure 8.4 demonstrates the police side functionality in detail.



FIR form

FIR ID

FIR#98456

Recording Officer

Mr. Vikas Garg

Recording Person

Mr. Arun Goel

Complainant

Aditya

Police Station

Dwarka Mor

Offence

Theft

Section

379

Police Station Code

PS#23

FIR Contents

My smartphone was stolen. I was going to Dwarka station. Then suddenly 2 men came running towards me. One of them collided into me and I fell down. My phone fell out of the pocket behind me. The people gathered around me to pick me up. Then I turned back to pick up my phone but it was not there rather I saw the first man running away with my phone.
The man's face was covered with mask. His height was around 5' 6" and he wore a red t-shirt with blue jeans.

Fig. 8.3 FIR form

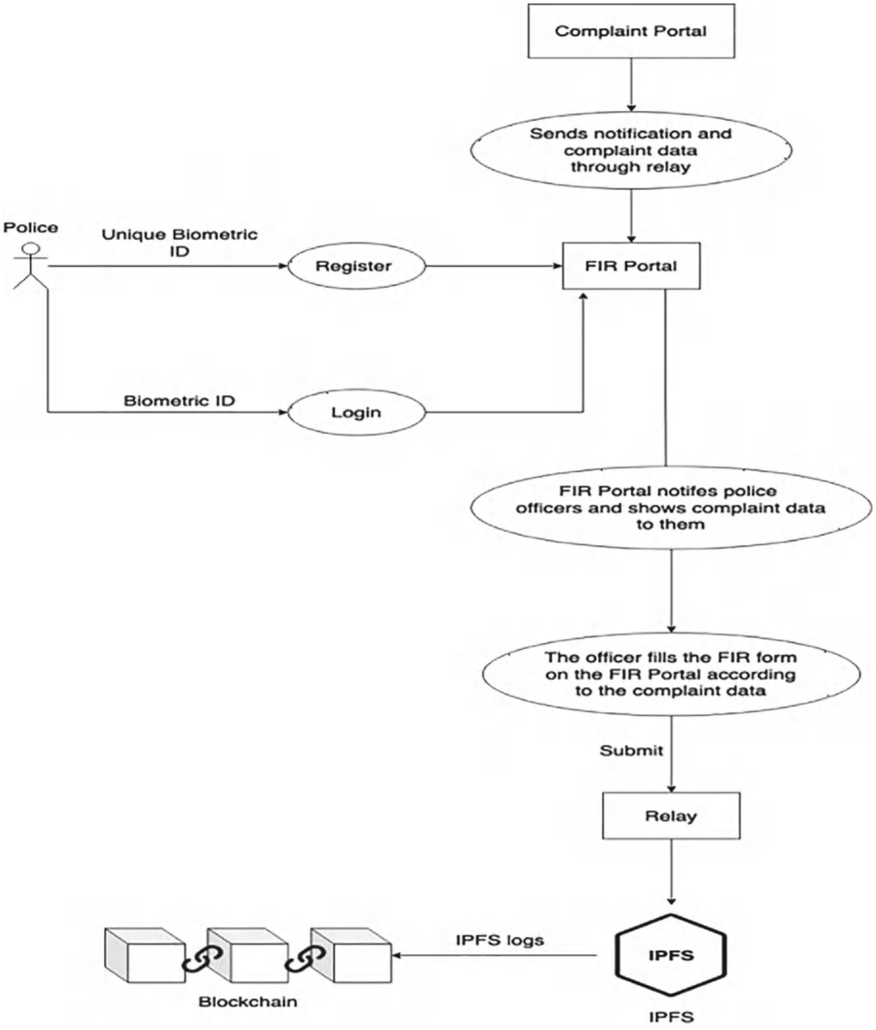

Complaint Portal
Sends notification and complaint data through relay
Police
Unique Biometric ID
Register
FIR Portal
Biometric ID
Login
FIR Portal notifes police officers and shows complaint data to them
The officer fills the FIR form on the FIR Portal according to the complaint data
Submit
Relay
IPFS logs
IPFS
Blockchain
IPFS

Fig. 8.4 Police-side functionality

8.5 Implementation and Results

There were two possibilities for implementing a blockchain-based system: building a new blockchain network or adopt an existing open-source one like Ethereum and Bitcoin. The work made advantage of the existing blockchain network rather than recreating the wheel. Further research determined that, although being the biggest blockchain network available, Bitcoin has relatively limited functionality and would not have been useful for the purpose. Therefore, Ethereum was adopted, the second most popular blockchain frequently used for decentralised apps. Among other advantages, it enabled to create Smart Contracts, impose validations and maintain security.

Working on an actual blockchain requires real money. The system uses Ganache, a personal blockchain because the objective was just to demonstrate the viability of the concept and conduct testing to determine trends. For testing reasons, it enabled immediate mining with a difficulty setting of 0. The advantage of utilising Ganache is that it offers ten distinct accounts, each with 100 ethers. They are only being used for development, where we may construct numerous distinct personal blockchain addresses for each node.

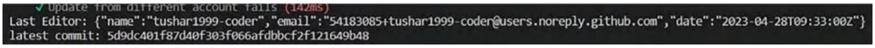
The method by which Ethereum is mined utilises the notion of Proof of Work (PoW), but defines operations in a smart contract and permitting addresses for operations can deliver the advantages of Proof of Authority (PoA). Smart Contracts are nothing but digital contracts with features to enforce contract requirements. They form the basis of any decentralised application. Smart Contracts empowered to decide the structure of each block and set authentication as well as predefined constraints that each block must adhere to.

Solidity is the programming language we used to write the Smart Contracts. Solidity compiler converts Smart Contract code into bytecode and Application Binary Interface (ABI), which is then utilised by Ethereum Virtual Machine (EVM) for deployment. Remix provides online tools for deploying Smart Contracts on local or test networks. It also provides debugging options along with easier ways to interact with the code to ensure correct working. Web3.js: a widely used library to enable JavaScript to interact with EVM and the Smart Contract is used.

The data fields can also be updated. This leads to the creation of a new block linked to the previous block, which gives the required flexibility to the policeman to add more data or change some data. To ensure even further security, sensitive data like sections levied on criminals can only be changed by certain accounts (accounts (Singh et al. 2023) in our case). If we try to access such a function from a different account, authorization checks of smart contracts are not satisfied and the transaction is rejected.

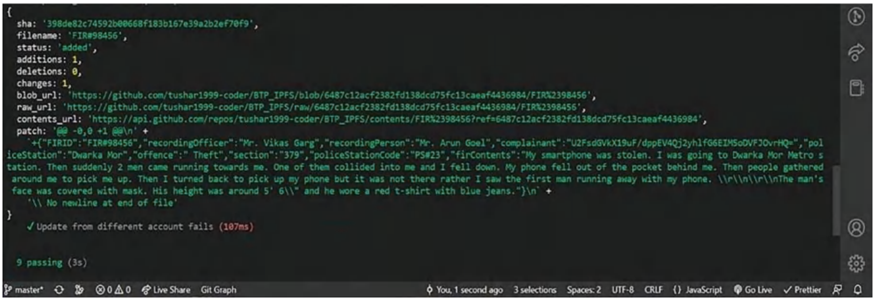
In the system, IPFS is used for storing multiple entities like FIR (Liao 2023), evidence, chargesheet, forensic reports, criminal records and judicial proceedings of ongoing cases. For lodging FIR, we designed and developed an FIR form using Node JS and JavaScript as shown in Fig. 8.3. The police officer fills out this form and submits it. This data is captured in the relay and parsed to separate different fields. Then selective entities are encrypted. This encrypted data (Aggarwal et al. 2024) along with other data is sent to the IPFS for storage. This constitutes an IPFS transaction.

tored on the blockchain. These IPFS logs contain the author, time and commit ID of the operation. The commit ID can be used to get a detailed description of all the changes made to IPFS, as shown in Figs. 8.5 and 8.6. Furthermore, there are various encryption algorithms like AES, DES and TripleDES available. They have been examined based on time taken and file sizes to encrypt and store data on IPFS and selected the most suitable algorithm that has a good trade-off between security and speed for our implementation. This can be analysed using Figs. 8.7 and 8.8.



```
    ✓ Update from different account fails (142ms)
Last Editor: {"name":"tushar1999-coder","email":"54183085+tushar1999-coder@users.noreply.github.com","date":"2023-04-28T09:33:00Z"}
latest commit: 5d9dc401f87d40f303f066afdbbcf2f121649b48
```

Fig. 8.5 IPFS last edit and last commit



```
{
  sha: '398de82c74592b00668f183b167e39a2b2ef70f9',
  filename: 'FIR#98456',
  status: 'added',
  additions: 1,
  deletions: 0,
  changes: 1,
  blob_url: 'https://github.com/tushar1999-coder/BTP_IPFS/blob/6487c12acf2382fd138dcd75fc13caeaf4436984/FIR%2398456',
  raw_url: 'https://github.com/tushar1999-coder/BTP_IPFS/raw/6487c12acf2382fd138dcd75fc13caeaf4436984/FIR%2398456',
  contents_url: 'https://api.github.com/repos/tushar1999-coder/BTP_IPFS/contents/FIR%2398456?ref=6487c12acf2382fd138dcd75fc13caeaf4436984',
  patch: '@@ -0,0 +1 @@\n' +
    `+{"FIRID":"FIR#98456","recordingOfficer":"Mr. Vikas Garg","recordingPerson":"Mr. Arun Goel","complainant":"U2FsdGVkX19uF/dppEV4Qj2yhlfG6EIMSoDVFJOvrHQ=","pol
iceStation":"Dwarka Mor","offence":" Theft","section":"379","policeStationCode":"PS#23","firContents":"My smartphone was stolen. I was going to Dwarka Mor Metro s
tation. Then suddenly 2 men came running towards me. One of them collided into me and I fell down. My phone fell out of the pocket behind me. Then people gathered
 around me to pick me up. Then I turned back to pick up my phone but it was not there rather I saw the first man running away with my phone. \\r\\n\\r\\nThe man's
 face was covered with mask. His height was around 5' 6\\" and he wore a red t-shirt with blue jeans."}\n` +
    '\\ No newline at end of file'
}
    ✓ Update from different account fails (107ms)


  9 passing (3s)
```

Fig. 8.6 IPFS log

Fig. 8.7 Time taken by various encryption algorithms to encrypt a file and store it on IPFS

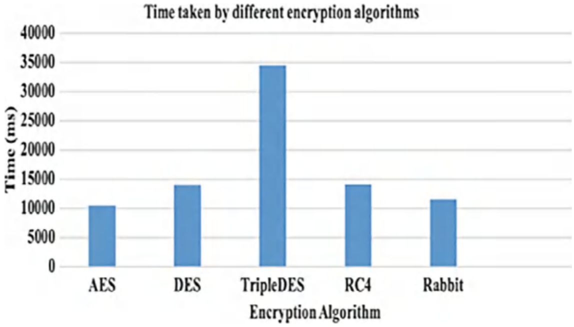


Fig. 8.8 Time taken by various encryption algorithms to encrypt files with increasing sizes

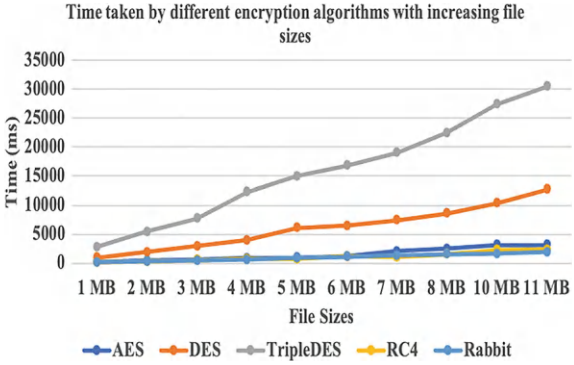


Table 8.1 Gas used for different tasks

Task	Execution Cost (approx)
Deploy the Contract	1,039,617 wei
Update Complaint	46,623 wei
Update Crime	46,777 wei
Update Police Station Code	46,693 wei
Update Section	46,803 wei

Table 8.2 System specifications

System RAM	8 GB
Hard Drive	1 TB HDD
System Core	Intel core i5 7th gen
OS	Windows 10
Core	2

As depicted in Fig. 8.7, AES algorithm took the least time among others. Furthermore, as shown in Fig. 8.8, it can observed that as file size increases, time taken by DES and TripleDES increases drastically rendering them impractical for the proposed system's use. As the performance of AES is comparatively better than others so AES algorithm has been used for data encryption.

For every operation on the blockchain, there is an associated execution cost (gas fee) with it as shown in Table 8.1. The cost is expressed in 'wei' which is the smallest denomination of ether. For reference purposes, 1 ether = 10^18 wei. All the findings in this paper are executed on a machine with the specifications shown in Table 8.2.

The proposed system showcases a fully developed functionality of the police management system. It stores new complaints, handles new FIR, makes different chains for different FIRs and stores the progress of a case in the form of more blocks linked to the chain. Only authorised personnel can edit the field and whatever addition/updation/deletion will occur on IPFS, its log will also be stored, making it impossible to tamper with anything without being caught. Thus increasing transparency, accountability and security of the entire management system.

8.6 Conclusion and Future Scope

The paper concludes with the working model of blockchain-based lightweight (Dhingra et al. 2024) police management system. The model uses the Ethereum blockchain (Singh and Kukreja 2023; Li et al. 2021; Hassija et al. 2019; Arabsorkhi and Ebrahimi 2022; Patil and Pise 2022; Patil et al. 2021; Pawade et al. 2020) and IPFS database for data storage. We examined various encryption algorithms on the basis of time taken and file sizes to encrypt and store data on IPFS and then selected the most suitable algorithm that has a good trade-off between security and speed for our implementation. Also, since we are only storing selective data items on the blockchain, our model is lightweight as compared to the existing models.

The proposed blockchain system can be extended to include ML algorithms and insights for a much more advanced police management system. The blockchain and IPFS database can be used to train an ML model that will help the system understand the relationship between different cases solved and their corresponding officers. And then assign cases to the police officers based on their areas of expertise. It can also suggest possible subordinates to work with experts and learn new skills. The database can also be used to plot the heatmap of crimes that would help to distribute the workforce according to the crime severity and urgency.

References

Aggarwal A, Awasthi E, Kukreja D, Kedia J, Bala I (2024) A novel framework for image encryption by integrating modified moth flame optimization and logistic chaotic map for enhanced security

Arabsorkhi A, Ebrahimi S (2022) Blockchain applications for the police task force of IRI: a conceptual framework using Fuzzy Delphi Method. Special issue: the business value of Blockchain, challenges, and perspectives. J Inf Technol Management 14: 36–61

Dhingra B, Jain V, Sharma DK, Gupta KD, Kukreja D (2024) RLET: a lightweight model for ubiquitous multi-class intrusion detection in sustainable and secured smart environment. Int J Inf Secur 23(1):315–330

Hassija V, Patel A, Chamola V (2021) Police fir registration and tracking using consortium blockchain. In: advances in machine learning and computational intelligence: proceedings of ICMLCI 2019. Springer Singapore, pp 785–794

Hingorani I, Khara R, Pomendkar D, Raul N (2020) Police complaint management system using blockchain technology. In: 2020 3rd international conference on intelligent sustainable systems (ICISS). IEEE, pp 1214–1219

Jain A, Das S, Kushwah AS, Rajora T, Saboo S (2021) Blockchain-based criminal record database management. In: 2021 Asian conference on innovation in technology (ASIANCON). IEEE, pp 1–5

Khan ND, Chrysostomou C, Nazir B (2020) Smart fir: securing e-fir data through blockchain within smart cities. In: 2020 IEEE 91st vehicular technology conference (VTC2020-Spring). IEEE, pp 1–5

Li M, Lal C, Conti M, Hu D (2021) LEChain: a blockchain-based lawful evidence management scheme for digital forensics. Futur Gener Comput Syst 115:406–420

Liao T (2023) Blockchain-enabled police management framework for securing police data. Soft Comput 27(23):18061–18075

Mollah MB, Islam SS, Ullah MA (2012) Proposed e- police system for enhancement of e-government services of Bangladesh. In: 2012 international conference on informatics, electronics and vision (ICIEV). IEEE, pp 881–886

Mukherjee A, Halder R (2020) Policechain: Blockchain-based smart policing system for smart cities. In: 13th international conference on security of information and networks, pp 1–5

Patil S, Pise R (2022) Blockchain-based secure evidence-management police assistance system. Blockchain for Smart Systems 155–165

Patil, S, Kadam S, Katti J (2021) Security enhancement of forensic evidences using blockchain. In: 2021 third international conference on intelligent communication technologies and virtual mobile networks (ICICV). IEEE, pp 263–268

Pawade D, Sakhapara A, Shah R, Thampi S, Vaid V (2020) Blockchain based secure traffic police assistant system. Int J Educ Manag Eng 10(6):34

Singh R, Kukreja D (2023) Blockchain for IoT security and Privacy: Applications, challenges and future directions. In: proceedings of the 5th international conference on information management and machine intelligence, pp 1–5

Singh AV, Tiwari AO, Singh SS, Lobo VB (2022) A criminal record keeper system using Blockchain. In: 2022 6th international conference on trends in electronics and informatics (ICOEI). IEEE, pp 840–845

Singh R, Kukreja D, Sharma DK (2023) Blockchain-enabled access control to prevent cyber attacks in IoT: systematic literature review. Front Big Data 5:1081770

Stoll C, Klaaßen L, Gallersdörfer U (2019) The carbon footprint of bitcoin. Joule 3(7):1647–1661

Tabassum K, Shaiba H, Shamrani S, Otaibi S (2018) E-Cops: an online crime reporting and management system for Riyadh city. In: 2018 1st international conference on computer applications and information security (ICCAIS). IEEE, pp 1–8

Tasnim MA, Omar AA, Rahman MS, Bhuiyan MZA (2018) Crab: Blockchain based criminal record management system. In: 11th international conference and satellite workshops. In security, privacy, and anonymity in computation, communication, and storage: SpaCCS, Melbourne, NSW, Australia, Proceedings. Springer International Publishing, vol 11, pp 294–303

Chapter 9
Data Management in Blockchain Based Serverless Platform Using Blowfish Algorithm

Meenakshi Kandpal, Pranati Mishra, and Jyotirmayee Rautaray

Abstract Data storage and uploading require privacy and security. In today's world, data is what we are in search of everywhere. The emerging technology known as blockchain has completely changed the security landscape across all major computer science fields. The confidentiality and integrity of data is maintained by ensuring robust protection from unauthorized access. By reducing the risk of a data breach, blockchain technology enhances data immutability and transparency. Serverless offers scalable and secure storage solutions allowing efficient handling of large datasets. There are several algorithms such as RSA (Rivest-Shamir-Adleman), McEliece, Knapsack, DES (Data Encryption Standard), and Triple DES, however, these provide longer encryption and decryption times. Specific data may contain sensitive information from multiple individuals or organizations. Such type of models may involve security issues of privacy and integrity. A safe processing platform is required by the network to maintain huge amounts of IOT (Internet of Things) data that are collected through gadgets and wearables. Many managing systems, protocols, algorithms, and hardware components are used to preserve and safeguard these data. In this paper, we introduce a representation, which offers a blockchain-based data storage architecture that uses the Blowfish encryption and decryption technique to securely store data in blockchain based serverless platform and also helps in promoting scalability with an efficient time for encryption and decryption. This innovative combination of blockchain and blowfish algorithm establishes a secure and decentralized framework, which offers the solution for cryptographic data protection. Further, a rigorous comparative analysis of RSA, McEliece, Knapsack, DES, and Triple DES with the blowfish algorithm is done. We assessed six encryption algorithms for data storage in blockchain networks to guarantee the maximum level

M. Kandpal (✉) · P. Mishra · J. Rautaray
School of Computer Science, Odisha University of Technology and Research, Bhubaneswar, India
e-mail: meenakshikandpal14@gmail.com

P. Mishra
e-mail: pranatimishracse@outr.ac.in

J. Rautaray
e-mail: jrautraycse@outr.ac.in

K. Ahmad et al. (eds.), *Fostering Machine Learning and IoT for Blockchain Technology*,
Transactions on Computer Systems and Networks,
https://doi.org/10.1007/978-981-96-4078-2_9

of security. We discovered that the most effective way to create a decentralized, immutable system is to use blowfish encryption and decryption methods.

Keywords Blockchain · IoT · Blowfish algorithm · Serverless framework · Encryption · Decryption · Data storage · Privacy · Security · RSA · McEliece · Knapsack · DES · Triple DES

9.1 Introduction

Data management systems have been completely transformed by the quick development of blockchain technology, which provides previously unheard-of levels of decentralization, security, and openness. These advantages are increased by using a serverless platform, which offers a scalable and effective way to manage enormous volumes of data without requiring conventional server infrastructure. Through the integration of the Blowfish algorithm, a powerful symmetric key encryption technique renowned for its efficiency and speed, this method guarantees that data stays extremely safe throughout processing and storage throughout the blockchain network (Jyoti and Chauhan 2022; Ekblaw et al. 2016). This combination offers a robust and effective data management solution appropriate for a range of applications (Li et al. 2018; Banerjee et al. 2018; Dorri et al. 2017a). It also improves data security and integrity by utilizing the decentralized nature of blockchain to eliminate single points of failure. This study highlights how blockchain technology can revolutionize data management techniques in the digital era by examining its synergistic integration with the Blowfish algorithm within a serverless architecture (Salahuddin et al. 1805; Brambilla et al. 2016; Ferrer 2018).

In this epoch of rapid technological advancement, the amalgamation of blockchain technology and serverless computing has emerged as a groundbreaking paradigm, promising unparalleled security, efficiency, and scalability (Benisi et al. 2020). As businesses and individuals continue to navigate the intricate landscape of decentralized systems, the critical challenge of security sensitive data becomes more pronounced than ever (Dorri et al. 2017b; Ren et al. 2019; Kshetri 2017). In response to this demand for robust data protection, our chapter delves into the dynamic fusion of blockchain and serverless architecture, fortified by the formidable blowfish algorithm (Javed et al. 2020 Jan; Phansalkar et al. 2019; Puthal et al. 2018). Blockchain technology usages cryptography to preserve the consistency of a system as this is a peer-to-peer blockchain network, which helps the unauthorized access from a third party (Ali et al. 2018). The mechanism of blockchain technology can be well understood from the reference of Fig. 9.1. As is well known, a blockchain block consists of data as well as the hash of the previous block and the current block. The specific type of blockchain dictates the type of information that is stored in each block.

Every block has a hash that resembles a piece of DNA. Similar to DNA, a hash uniquely identifies a block and all of its contents, and each bit of information within a hash is always distinct. After a block is produced, its hash is calculated (Xu et al.

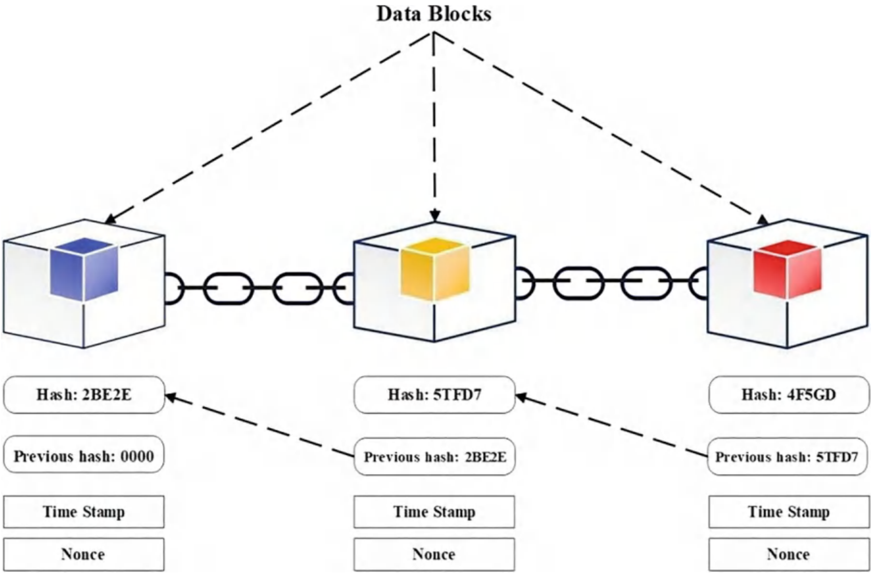


Fig. 9.1 Generalized architecture of Blockchain

2018). The hash will totally alter if there are any changes made to the block. Stated differently, hashes are extremely useful for identifying modifications to blocks. A block is no longer the older block when its hash is modified. The third element in every block is the hash of the block before it. This essentially creates a chain of blocks, and it is how the blockchain's security is maintained (Kurt Peker et al. 2020).

Data stored on blockchains is inherently public, raising concerns about the privacy and confidentiality. Additionally, serverless functions are ephemeral, posing challenges for the data persistence and access control. The blowfish algorithm used in the paper, renowned for its speed and flexibility, presents a compelling solution for securing data in the context (Liang et al. 2020).

- Variable key length: Adapts to various security requirements.
- Feistel Network Structure: Offers strong resistance to cryptanalysis.
- Data-Dependent Key Scheduling: Enhances encryption strength.

Through the meticulous analysis and empirical evidences, we aim to demonstrate how the marriage of blockchain and serverless computing, fortified by the blowfish algorithm, can create an impregnable fortress for sensitive data, ensuring confidentiality. Integrity and availability in an increasing digital world.

The popularity of blockchain and serverless technologies may serve as the foundation for a fusion of the two technologies. Since, IoT faces security concerns, and blockchain has grown in popularity for delivering security across different domains, the two technologies may prove to be complementary (Chen et al. 2019). As a result, blockchain can help secure IoT sensory data and establish itself as a vital ledger technology. Blockchain can greatly improve the security of IoT sensory data by utilizing distributed storage mechanisms and encryption algorithms. Furthermore, without the intervention of humans, the data's integrity may be well kept with this merger method.

9.1.1 Organisations

The following sections of this chapter are organized as follows: Section 9.2 provides Objective and Motivation behind this work. The Related work incorporating Blowfish into serverless blockchain systems, illustrated in Sect. 9.3, which provides a reliable option for safe and effective data administration Sect. 9.4 outlines the proposed framework, detailing the methodology of Encryption and Decryption using Blowfish algorithm. Section 9.5 gives light on Results and Discussion; Further Sect. 9.6 gives comparative analysis of RSA, McEliece, Knapsack, DES, Triple DES with blowfish algorithm. Section 9.7 summarizes the paper by outlining the main findings and contributions, and proposing pathways for future research.

9.2 Objective and Motivation

The primary objective of this research is to examine and demonstrate the usefulness of integrating the blowfish algorithm into Blockchain based serverless platform for enhanced data management. The study aims to explore cryptographic strengths of blowfish within the context of decentralized systems, specifically focusing on its application in securing the data transactions, storage and communication within a serverless computing environment. The motivation behind the research stems from the pressing need for advanced data security solutions in the ever-evolving landscape of decentralized computing.

9.3 Related Work

Vasantha et al. (2019a) presented a thorough review of privacy and security utilizing blockchain and the blowfish algorithm. The core security features supported are essential building blocks for Bitcoin-like crypto currency systems, the supplementary threat protection properties that are required in many blockchain application are introduced later. It is particularly suggested by the author that the system examines the security and protection approaches for achieving these security qualities in blockchain-based frameworks with blowfish algorithms, including respective consensus algorithms, hash tied storage, mixing protocol, anonymous signature, and additional components. This study provides in-depth examination of security and privacy with threat protection using block chain and the blowfish algorithm (Vasantha and Prasad 2019a). The well-known concept of Blockchain and its use in the context of bitcoin transactions, such as online email transactions. After that, the system outlines the crucial security aspects that serve as essential requirements and building blocks for Bitcoin-like cryptocurrency systems. It then covers the additional security and protection properties needed in various blockchain applications (Vasantha and Prasad 2019a). The goal of the study is to offer a secure informative outcome based on that square chain innovation (Vasantha et al. 2019b). The blowfish algorithm is combined with blockchain technology to make communications more secure presenting a model design for block chain with blowfish-based messaging that maintains the performance and security of data recorded on the block chain is discussed (Vasantha and Prasad 2019b). Despite being recognized as an unbreakable method, due to limited block size support the application of Blowfish algorithm has been unsuitable (Reyes et al. 2018). Further, the author provided an improved version of the Blowfish cryptographic algorithm that handles 128-bit block size input through dynamic selection encryption reducing cypher function runtime by the convenient use of randomly selected rounds (Reyes et al. 2018). Kalpanadevi et al. (2022) focused on the sensor's device, which when connected to the network or service from any location, will receive signals indicating changes that have occurred in the bodies of patients, and then transmit these signals to the IoT middleware. The data received by the IoT middleware is sent to the internet cloud, where it is stored for analysis. An IPv6 addressing scheme controls the middleware that uses the Runge-Kutta (RK) Blowfish algorithm. This algorithm modifies the Feistel cypher of Blowfish through the integration of Blowfish and the Runge-Kutta method (Kalpanadevi et al. 2022). Individuals and data owners can connect to the network via block chain and cloud technologies using BDBC-block- chain data transmission which employs blowfish security is optimized in the cloud network system (Bandaru and Visalakshi 2024). To protect the data from unauthorized access, the data owner encrypts it using the blowfish technique before transmitting it via blockchain to the cloud (Bandaru and Visalakshi 2024). The author's study also demonstrated a crypto stego technique using Blowfish and the LSB(Least Significant Bit) algorithm protects sensitive server data from unauthorized access (Singhal et al. 2022). Consequently, retaining privacy and securing sensitive data are stored on the cloud (Singhal et al. 2022).

According to recent studies, incorporating Blowfish into serverless blockchain systems provides a reliable option for safe and effective data administration. Modern data management systems have several issues that match Blowfish's speed and agility, along with the decentralized and scalable nature of blockchain and serverless computing, address. This integration is a major step forward, laying the groundwork for more study and innovation in safe, expandable, and effective data management technology.

9.4 Proposed Model

Blockchain framework has the potency to discourse the issue of secure data storage on the serverless platform. A data block is a collection transaction that requires strong security. This narrative unfolds as the data block undergoes a gentle fragmentation, transforming into shards—tiny yet significant fragments of information. To preserve data decentralization, each shard will be routed to a different serverless framework. After safely delivering shared data to the serverless framework, these data blocks are individually encrypted using the Blowfish technique. The data that was extracted is now saved in the serverless framework's blockchain technology, which will be now managed and scaled. When a request is made to access these data objects, the data is decrypted using the Blowfish technique and then returned to the data user. Figure 9.2 illustrates the steps how data is stored in blockchain based serverless platform using the blowfish algorithm.

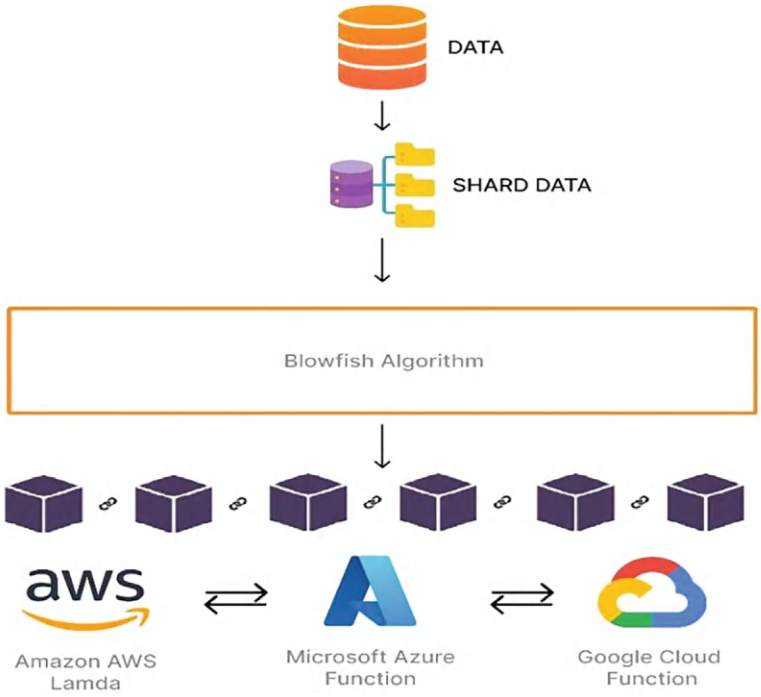


Fig. 9.2 Secure data storage using Blowfish algorithm in Blockchain based serverless platform

9.4.1 Blowfish Algorithm for Data Storage

Bruce Schneier developed the encryption technique known as Blowfish in 1993 to replace the DES encryption system. The topic has a high encryption frequency and is significantly faster than DES. No effective cryptanalytics methods have yet been found for it. It is primarily secure and safe block cyphers that is non-patentable and hence freely available to anyone. Symmetric encryption refers to the use of the identical key to encode and decode data. The encryption key and decryption key in the Blowfish algorithm convert secret data into ciphertext. Blowfish is the godfather of the Twofish and AES encryption algorithms.

The Blowfish algorithm positions as a stalwart sentinel in the zone of data storage, providing a great solution for encrypting and shielding the data. Its inheritance of security is joined with its adaptability to varying computational settings, positions it as an esteemed optimal choice for confirming the privacy and integrity of data stored in present-day storage arrangements. Blowfish additionally employs a block size of 64, which creates tremendous complexity and ensures the key's complete security. Twofish's upgraded and significantly larger 128 block size implementation did address several concerns. However, it lacks the speed that some users seek in encryption. Figure 9.5 shows the encryption process of the blowfish algorithm.

- 64-bit block
- 32-bit to 448-bit variable key
- 18-bit subkeys
- Total rounds: 16
- Substitution boxes: 4 [individually with 512, 32-bit elements].

9.4.2 Sub Key Generation

1. 18 SubKeys SubKeys[0]…SubKeys (Xu et al. 2018) are required for both encryption and decryption operations using the same sub keys in both.
2. The SubKeys-array element containing these 18 SubKeys is a 32-bit item.
3. SubKeys[i] = SubKeys[i] XOR (i + 1)th 32-bits of input key (based on the key length, it may roll over to first 32-bits) (Fig. 9.3).

9.4.3 Initialization Substitution Boxes

Both the encryption and decryption operations necessitate the employment of four substitution boxes shown in Fig. 9.4 (Sboxes), every one having 256 entries (S[i][0] – S[i][253], 32 bits, and 0&lei & le4).

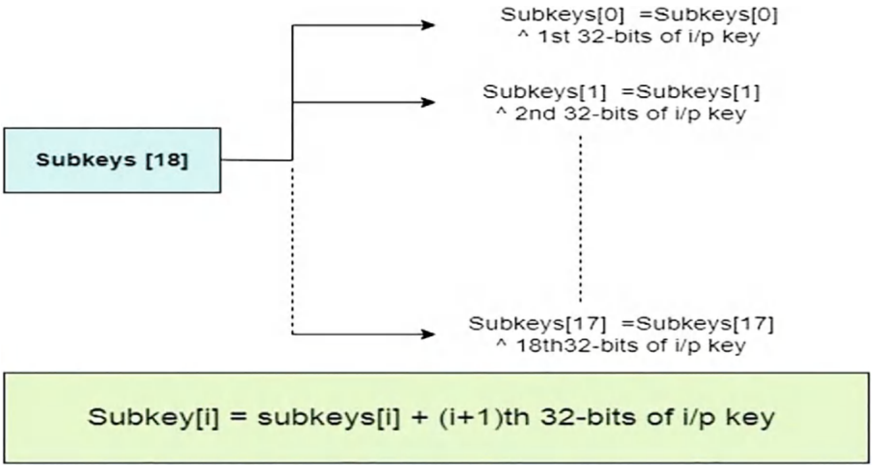

Subkeys [18]
Subkeys[0] =Subkeys[0] ^ 1st 32-bits of i/p key
Subkeys[1] =Subkeys[1] ^ 2nd 32-bits of i/p key
Subkeys[17] =Subkeys[17] ^ 18th32-bits of i/p key
Subkey[i] = subkeys[i] + (i+1)th 32-bits of i/p key

Fig. 9.3 Subkey array generation phase

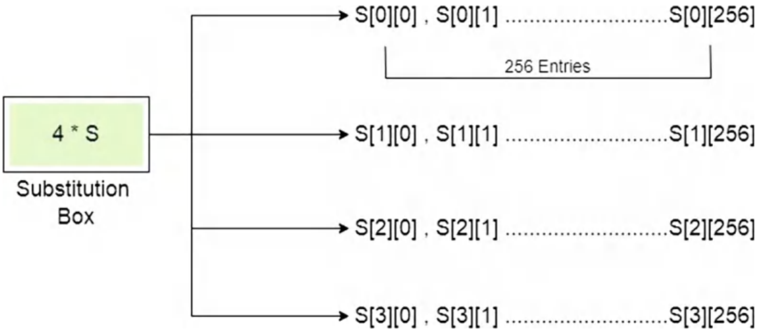

4 * S
Substitution Box
S[0][0] , S[0][1] ....S[0][256]
256 Entries
S[1][0] , S[1][1] ....S[1][256]
S[2][0] , S[2][1] ....S[2][256]
S[3][0] , S[3][1] ....S[3][256]

Fig. 9.4 Substitution box Initialization

9.4.4 Blowfish for Encryption

- **Input:** Plain Text 64-bits
- **Output:** Cipher Text 64- bits

Start

1. The Plain Text (64 bits) is broken into two 32-bit sections.
2. The LHS 32-bits are XORed with the first Subkey, and the output is sent to the Substitution boxes.
3. 32 bits of data are divided into four portions and sent to four Substitution boxes.
4. The first and second substitution boxes are XOR, and the result is XOR with the third, and the output from the third box is XOR with the fourth to create a 32-bit output from the function.

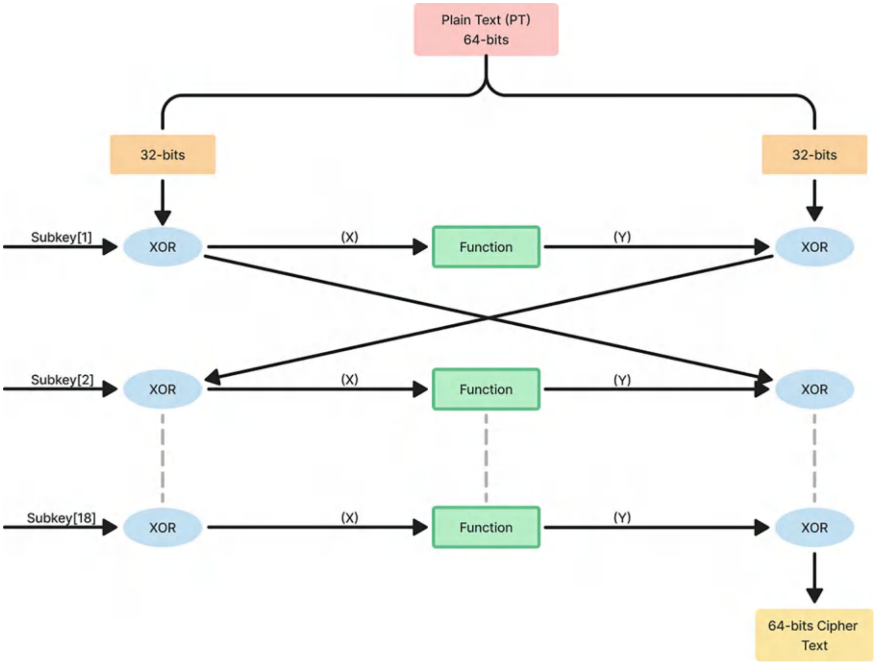


Fig. 9.5 Blowfish based encryption flow

5. The function's output is XOR with the RHS 32-bits once more.
6. In the next phase, the LHS and RHS outputs are crisscrossed and XORed with the 2nd subkey.
7. The XOR and crossing operation is repeated till 18 subkeys are expended.
8. Finally, we have a 32-bit Cypher text.

End (Fig. 9.5)

9.4.5 Blowfish for Decryption

Fig 9.6 shows the steps for the decryption process using blowfish which is reverse of encryption. In input the file is an encrypted text then it is transformed to a plain text. The procedure of decryption is undistinguishable to that of encryption. The same subkeys and substitution boxes are employed. The subkeys are employed in inverse direction during the decryption procedure, from the 18th to the first subkey.

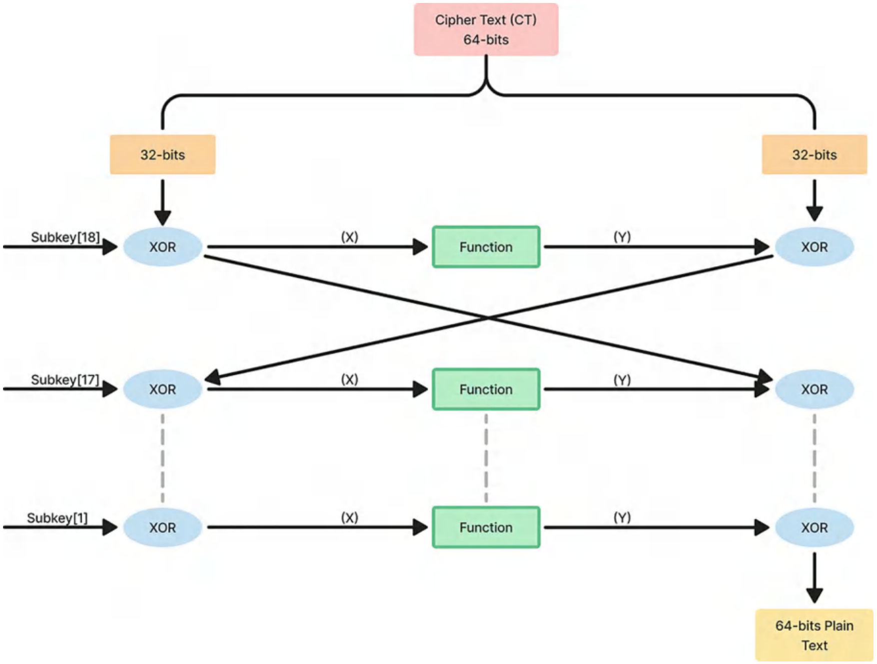


Fig. 9.6 Blowfish based decryption flow

9.5 Results and Discussion

In this chapter, we projected usage of Blowfish algorithm to certify additional security to the data stored in blockchain framework. The proposed model recommends the amalgamation of the blockchain and serverless environment intended for safe and secure data storage using blowfish encryption decryption algorithm for excellent security. Blowfish algorithm provide various benefits which makes it better from different algorithms. This algorithm provides a facility of variable length of the key which helps the algorithm to be adapted by a different system according to their requirements. Its adaptability ranges from 32 bits–448 bits. In this paper we used the blowfish algorithm for the encryption and decryption process and it is very time-efficient for the conversion of the text from the encrypted format to the decrypted format or the vice -versa. Further it has a feistel network helping in a simplified algorithm and uses minimal resources for operating efficiently. Hence it is a method which is an open source and transparent which is a key point for the ideal cryptography. It has a great solution towards memory efficiency, as minimizing the memory is also a criterion for a better program.

9.6 Comparative Analysis

To guarantee security, effectiveness, and scalability in the field of data management for blockchain-based serverless platforms, selecting the appropriate encryption method is essential. Symmetric encryption algorithm Blowfish is a standout option for a number of reasons. First off, Blowfish provides a high degree of security since its changeable key size (32-448 bits) makes it immune to brute force assaults. Because of its versatility, it can be used to meet a variety of security requirements. Blowfish offers a more reliable solution without sacrificing speed as compared to DES, which is now regarded as insecure because of its short key length, and Triple DES, which offers mediocre security with performance concerns.

Another important consideration is performance. Blowfish is well-known for its remarkable speed and effectiveness, which are essential in a serverless setting where it's necessary to maximize computational resources. Although very safe, asymmetric algorithms like RSA and McEliece typically require more resources and run longer, which makes them unsuitable for high-throughput situations like those seen in blockchain platforms. \Despite its historical significance, Knapsack is no longer practical because of its susceptibility to contemporary cryptographic assaults. Blowfish, on the other hand, has endured and is still reliable and effective for a range of uses. Since data is dispersed among several nodes in a blockchain-based serverless network, effective and secure encryption is required to preserve data integrity and privacy. The ability to securely handle data without adding substantial latency is made possible by Blowfish's quick encryption and decryption operations, which is essential for preserving the platform's performance and scalability. Therefore, using Blowfish instead of other encryption algorithms like RSA, McEliece, Knapsack, DES, and Triple DES for data management in a blockchain-based serverless platform gives a balanced approach, solid security, and high performance (Table 9.1).

9.7 Conclusion

The world's digitalization shows no signs of slowing. Two years into the pandemic, fraudsters are still preying on customers who live their lives online. Data that has previously been compromised is being utilized to tap into a rich vein of more personal information stored on websites such as online shopping, social media, and healthcare. In 2021, two billion data including usernames and passwords were compromised, a 35% increase over the previous year. In 2021, unauthorized access was the most common vector, accounting for 50% of all breaches.

In a Blowfish-based largely secure data storage state, a key of variable size is primarily produced, and the data is translated by means of the Blowfish process rule to keep it safe from hackers. This model recommends the Blowfish algorithm because it facilitates faster data storage transactions, i.e. the process of encryption and decryption is significantly faster while maintaining data security.

Table 9.1 Comparative study of RSA, McEliece, Knapsack, DES, and Triple DES with Blowfish

Algorithm	Category	Size(key)	Level of security	Performance	Uses
RSA	Asymmetric	1024–4096 bits	High	Moderate	Digital signatures, protected exchange of keys, encryption
McEliece	Asymmetric	50,000 + bits	High	Moderate	Protected communications, post- quantum cryptography
Knapsack	Asymmetric (obsolete)	Varies	Low	Fast	Significant interest, not used in practice
DES	Symmetric	56 bits	Low	Fast	Momentous interest, substituted by AES and Triple DES
Triple DES	Symmetric	168 bits	Moderate	Moderate	Data encryption, used in inheritance systems
Blowfish	Symmetric	32–448 bits	High	Very Fast	Data encryption (suitable for applications like file and database management encryption)

References

Ali S, Wang G, White B, Cottrell RL (2018) A blockchain-based decentralized data storage and access framework for pinger. In: 2018 17th IEEE international conference on trust, security and privacy in computing and communications/12th IEEE international conference on big data science and engineering (TrustCom/BigDataSE), pp 1303–1308

Bandaru VNR, Visalakshi P (2024) BDBC-block-chain data transmission using blowfish security with optimization in cloud network. Int J Intell Syst Appl Eng 12(5s):370–378

Banerjee M, Lee J, Choo KK (2018) A blockchain future for internet of things security: a position paper. Digit Commun Netw 4(3):149–60

Benisi NZ, Aminian M, Javadi B (2020) Blockchain-based decentralized storagenetworks: a survey. J Netw Comput Appl 102656

Brambilla G, Amoretti M, Zanichelli F (2016) Using blockchain for peer-to-peerproof-of-location. arXiv-1607

Chen Y, Ding S, Xu Z, Zheng H, Yang S (2019) Blockchain-based medicalrecords secure storage and medical service framework. J Med Syst

Dorri A, Kanhere SS, Jurdak R (2017) Towards an optimized blockchain for IoT. In: 2017 IEEE/ACM second international conference on internet-of-thingsdesign and implementation (IoTDI) 2017 Apr 18. IEEE, pp 173–178

Dorri A, Kanhere SS, Jurdak R, Gauravaram P (2017) Blockchain for IoTsecurity and privacy: the case study of a smart home. In: 2017 IEEE international conference on pervasive computing and communicationsworkshops (PerCom workshops). IEEE, pp 618–623

Ekblaw A, Azaria A, Halamka JD, Lippman A (2016) A case study for blockchain in healthcare: "MedRec" prototype for electronic health records and medical research data. In: Proceedings of IEEE open & big data conference, vol 13, p 13

Ferrer EC (2018) The blockchain: a new framework for robotic swarmsystems. In: Proceedings of the future technologies conference. Springer, Cham, pp 1037–1058

Javed MU, Rehman M, Javaid N, Aldegheishem A, Alrajeh N (2020) Tahir blockchain-based secure data storage for distributed vehicular networks. Appl Sci 10(6):2011

Jyoti A, Chauhan RK (2022) A blockchain and smart contract-based data provenance collection and storing in cloud environment. Wireless Netw 28(4):1541–1562

Kalpanadevi D, Rani MJ, Karuppasamy M (2022) Enhancement of RK-blowfish algorithm for data encryption through block chain in healthcare system. Math Stat Eng Appl 71(3s2):70–80

Kshetri N (2017) Blockchain's roles in strengthening cybersecurity andprotecting privacy. Telecommun Policy 41(10):1027–1038

Kurt Peker Y, Rodriguez X, Ericsson J, Lee SJ, Perez AJ (2020) A cost analysis of internet of things sensor data storage on blockchain via smart contracts. Electronics 9(2):244

Li R, Song T, Mei B, Li H, Cheng X, Sun L (2018) Blockchain for large-scale internet of things data storage and protection. IEEE Trans Serv Comput 12(5):762–771

Liang W, Fan Y, Li KC, Zhang D, Gaudiot JL (2020) Secure data storage andrecovery in industrial blockchain network environments. IEEE transactions on industrial informatics

Phansalkar S, Kamat P, Ahirrao S, Pawar A (2019) Decentralizing AI Applications With Block Chain.

Puthal D, Malik N, Mohanty SP, Kougianos E, Yang C (2018) The blockchainas a decentralized security framework [future directions]. IEEE Consum Electron Mag 7(2):18–21

Ren Y, Leng Y, Cheng Y, Wang J (2019) Secure data storage based on blockchain and coding in edge computing. Math Biosci Eng 16(4):1874–92

Reyes ARL, Festijo ED, Medina RP (2018) Blowfish-128: a modified blowfish algorithm that supports 128-bit block size. In: 8th international workshop on computer science and engineering, Bangkok, Thailand, pp 578–584

Salahuddin MA, Al-Fuqaha A, Guizani M, Shuaib K, Sallabi F (2018) Softwarization of internet of things infrastructure for secure and smarthealthcare. arXiv:1805.11011

Singhal V, Singh D, Gupta SK (2022) Crypto STEGO techniques to secure data storage using DES, DCT, blowfish and LSB encryption algorithms. J Algebr Stat 13(3):1162–1171

Vasantha R, Prasad RS, Guntur AJSTD (2019) Secured email data based on blowfish with blockchain technology. Sci Technol Dev 8:456–464

Vasantha R, Prasad RS (2019) An exploration on blowfish algorithm and security of block chain systems. JASC J Appl Sci Comput ISSN 1076–5131

Vasantha R, Prasad RRS (2019) A secured blockchain technology by using blowfish algorithm for new broadcast proxy provisional re- encryption & its application to cloud e-mail

Vasantha R, Prasad RS, Guntur AP (2019) An efficient secured system by using blowfish with block chain technology

Xu Q, Aung KM, Zhu Y, Yong KL (2018) A blockchain-based storage systemfor data analytics in the internet of things. In: New advances in the internet of things. Springer, Cham, pp 119–138

Chapter 10
Utilizing Blockchain in Cybersecurity: An Emphasis on Privacy and Data Safety

Asif Iqbal Hajamydeen and Asmaa Mahfoud Hezam Alhakimi

Abstract The distributed and immutable ledger system of blockchain technology and cybersecurity converge for addressing critical data security and privacy issues. In Cybersecurity, Blockchain Technology incorporation offers a promising solution to strengthen privacy while making sure that data is secure. This chapter provides a thorough analysis of how blockchain technology can be integrated into the field of cybersecurity specifically focusing on aspects like privacy and data security. It explores fundamental elements of blockchain including decentralized ledgers, consensus mechanisms as well as smart contracts. Additionally, it looks at the security implications of blockchain which includes its advantages, challenges, typical attack vectors, and defensive approaches. The paper also presents some practical applications such as identity management, data privacy and IoT security in the realm of cybersecurity. The chapter further illustrates the important role played by blockchains in encryption, data integrity and access controls with examples from finance and healthcare sectors. Besides bringing to the fore future issues and developments in regard to blockchain cybersecurity, it also offers knowledge on the areas that are emerging. The final part of this chapter recaps all the main points and highlights how life-changing blockchain technology can be with respect to enhanced secrecy as well as data safety in vulnerable sectors.

Keywords Cyber security · Block chain · Smart contract · Distributed ledger · Consensus mechanism

A. I. Hajamydeen (✉)
Artificial Intelligence and Cyber Security Centre, Management and Science University, Shah Alam, Selangor, Malaysia
e-mail: asif@msu.edu.my

A. M. H. Alhakimi
Faculty of Information Sciences and Engineering, Management and Science University, Shah Alam, Selangor, Malaysia
e-mail: asmaa@msu.edu.my

K. Ahmad et al. (eds.), *Fostering Machine Learning and IoT for Blockchain Technology*,
Transactions on Computer Systems and Networks,
https://doi.org/10.1007/978-981-96-4078-2_10

10.1 Introduction

Blockchain is a style of record keeping that is virtually very hard or impossible to alter, falsify, or corrupt. It is a public record that copies and provides transactions in a contributing network of computers. Blockchain is a formation that contains the records of transactions that are called blocks in several databases interconnected by a network with multiple nodes. This storage space is commonly referred to as a 'digital ledger'. All transactions in the ledger are sanctioned by the digital signature of the owner, which validates the transaction and protects it from tampering. Therefore, the information contained in the digital ledger is substantially safe. In a straightforward way, the digital ledger resembles a Google spreadsheet distributed among multiple devices in a network, where transactional data are accumulated based on definite purchases. The stimulating part is that everyone can view the data, however, cannot corrupt it.

All performed transactions are logged in a list of blocks on the blockchain, which may be deemed as a public ledger. As new blocks are constantly inserted into it, this sequence gets longer. Transactions that are integrated into the blockchain are unchangeable and cannot be altered. The essential attributes of blockchain technology are largely decentralization, persistence, anonymity, and auditability. Blockchain can be used by corporations that insist on a high level of integrity and dependability to draw in clients. In a blockchain, a block is the fundamental unit which is a grouping of data or information. By joining it with other blocks in chronological sequence, the data is added to the block in the blockchain, and a chain of linked blocks is formed. Beforehand, a new block is appended to the chain, the block must be verified and accepted by most of the participating nodes in the network through a process called agreement. After a block is added to the chain, it cannot be altered or eliminated without adjusting all the subsequent blocks, making the blockchain virtually tamper-proof. As a result, it creates a shared sequential database of transactions among several network nodes, computers, or servers. Blockchain is a chain of blocks, with three components in a block: data, its own hash, and the previous block's hash. The final block in the chain that needs persistent connection is included in every block's cryptographic hash.

Blockchain is a distributed database that preserves a journal of all transactions in a protected and append-only method. Blockchain swiftly became dominant among numerous industries because of its distributed nature (Nakamoto 2008). Especially for establishments that cannot allow a single point of failure, the blockchain database makes it nearly impossible for sensitive data or credentials to be compromised by hackers (Christidis and Devetsikiotis 2016). Moreover, blockchain can be managed not only by trusted developers or administrators but also by anyone who is a known party (Imran 2018). Each computer connected to the internet needs to have blockchain node software installed and run an application specific to the blockchain ecosystem (Zheng et al. 2018). The extent of participation of these computers can be restricted depending on the use cases (Pilkington 2016). For example, manufacturing companies are utilizing blockchain technology to monitor the real-time movement of goods within their supply chains (Xu et al. 2018).

Every device that participates in a blockchain network, often a computer or a computing device is called a blockchain node. It executes the program for the blockchain protocol, aiding in transaction validation and network security. Blockchain nodes exchange messages between them and the network is progressively decentralised as the nodes increase. Each blockchain node keeps a copy of the blockchain's primary protocol and its complete transaction history. Due to decentralization, individuals can operate a node from anywhere in the world, as long as they are connected to a decentralized blockchain network and have the required resources. The fundamental responsibilities of a blockchain node are transaction validation and broadcasting. A transaction submitted by a user to a node is broadcast to the whole network. Each node in the network validates the transaction to confirm that the sender is authorized to transmit the funds and has the necessary amounts on hand.

Decentralization is the distribution of authority, information, and functions instead of centralization to a single organization. It refers to the allocation of authority and decision-making responsibilities from a central entity to several entities scattered in the framework of the blockchain. Decentralized networks are envisioned to reduce the amount of confidence that users have in one another, and to prevent users from exploiting their power or controlling one another in a way that weakens the network's efficiency. This is because every node in the network has a duplicate or identical copy of the data contained in a distributed record. However, the majority of people in the network disregard a member's record that is changed or corrupted in any way. Once data is entered into a decentralized blockchain, it cannot be removed or altered, but new data can be appended to it.

10.1.1 Importance of Cybersecurity in Blockchain

Maintaining the integrity and security of blockchain networks is dependent on cybersecurity. Because blockchain operates in a distributed fashion, it is characteristically resistant to numerous cyberattacks. To protect against any vulnerabilities and take extensive advantage of its security benefits, further procedures are necessary. Blockchain shields data stored in blocks by using encryption algorithms. To establish a chain that is impossible to tamper with, each transaction is cryptographically hashed and connected to the previous blocks. Digital signatures reinforce the legitimacy of network users, adding more protection.

The incorporation of blockchain technology into cybersecurity frameworks has in the recent past become increasingly important in today's world. Through its distributed and hack-proof storage system, blockchain brings significantly positive changes to the world of information systems security (Nakamoto 2008). Since records contained in blockchain are permanent and immutable, it is very difficult for adversaries to modify or manipulate data, which enhances system security (Christidis and Devetsikiotis 2016). Blockchain technologies avoid susceptibilities inherent in conventional centralized systems, reducing the likelihood of a single point of failure that can be exploited by a hacker (Imran 2018). Security measures against unauthorized access and data leakage are made possible by cryptographic protocols in blockchain technology (Zheng et al. 2018).

Furthermore, greatly enhanced audit trails and incident response facilities are due to the inherent and unalterable transparency of blockchain, which is crucial to maintaining robust cybersecurity over the long term (Pilkington 2016). Through the automation of tasks and the implementation of security policies on blockchain platforms, the probability of errors is reduced, and system stability is promoted (Xu et al. 2018). The propagation of data in blockchain also reduces its susceptibility to common cybersecurity threats such as Distributed Denial of Service (DDoS) attacks (Liang et al. 2017a). In a connected world, technology that facilitates the seamless sharing of information across different parties securely and successfully is central for cooperative cybersecurity (Zhang and Wen 2017). Blockchain integration enhances access control as well as user authentication within the field of identity management (Crosby et al. 2016).

Its traceability and transparency features make it highly attractive as an incident response tool and for the making of efficient audit trails. This expertise is needed to maintain strong cybersecurity measures with growing complexity in the digital environment. Another cybersecurity concern is solved by decentralization in blockchain: blockchain faces a common problem much better than traditional approaches to defending against Distributed Denial of Service (DDoS) attacks.

10.1.2 Organization of the Chapter

The next subtopic of this chapter gives an overview of the fundamentals of blockchain such as distributed ledger technology, smart contracts, consensus mechanism, and the type of blockchain. Following this, it reviews the relationship between blockchain and cybersecurity, examining features, shortcomings, and security features of blockchain, common types of attacks, and protection measures. It then looks into several blockchain cyber-security use cases such as supply chain security, identity and access management, data privacy and protection, and the Internet of Things security. The following presents a more comprehensive description of how blockchain works in the context of data protection and privacy, as well as the encryption of data, its integrity, accessibility control, and other privacy protection techniques like private contracts and zero-knowledge proofs. Conclusion is presented in this chapter which consists of a brief overview of the key points and their implications for cybersecurity professionals, real-life examples of healthcare and financial services cybersecurity, and future developments, trends, and directions.

10.2 Blockchain Basics

Blockchain is a decentralized bookkeeping structure that employs a number of computers to store transactions with the assurance of their security, openness, and immutability. Through cryptographic methods, blocks are linked in such a way that they form a continuous chain of data. A blockchain's consensus mechanism is a network of nodes that discusses the authenticity of a particular transaction before adding it to the blockchain.

10.2.1 Distributed Ledger Technology

The foundation of blockchain systems is the Distributed Ledger Technology (DLT), which establishes the contemporary approach to data organization and protection (Imran 2018; Buterin 2014). Fundamentally, through decentralization of an appreciated database among many nodes, DLT removes any necessity of a central authority as mutually accepted and able to be synchronized across nodes (Nakamoto 2008; Swan 2015). Due to the absence of concentrated weaknesses that miscreants can exploit, this decentralized approach enhances the credibility of the data as well as creates fabulous protection against cybercriminal activities (Zheng et al. 2018). The ability to identify tampering from the tamper-evident environment that accompanies the records architecture underpinned by DLT systems is crucial for preserving the original identification and chain of custody of data in cybersecurity applications (Narayanan et al. 2016).

DLT consensus mechanisms, such as Proof of Work or Proof of Stake, help ensure that all participants have a common view of the current state of the ledger, thus discouraging fraud and unauthorized alterations (Bano et al. 2019). Real-time auditing and transaction monitoring can be carried out to improve the general security of the system, as demonstrated by evaluations of DLT implementations (Xu et al. 2017a). Cybersecurity and data protection benefit from DLT, as it guarantees data accuracy and privacy and encourages secure information exchange (Liang et al. 2017a). Known as DLT or distributed ledger technology, it tends to be efficient in the protection of sensitive information since its core elements provide strong tenets in the cryptography field to prevent data leakage incidents and unauthorized accesses to such data (Zheng et al. 2017).

Additionally, the flexibility coupled with scalability inherent in DLT systems makes them adaptable for responding to evolving cybersecurity threats, contributing to their prolonged viability amidst contemporary challenges (Christidis and Devetsikiotis 2016). With businesses progressively discerning the potential of DLT to bolster their cybersecurity infrastructures, a marked increase in its utilization is anticipated in forthcoming years (Radziwill 2018). Such a shift is poised to fundamentally alter the approaches to data protection and privacy in the digital age (Pilkington 2016). Provisions for real-time auditing and transaction monitoring are provided by the inherent accessibility within DLT. The ability to promptly detect suspicious actions or discrepancies significantly enhances overall system security. From a cybersecurity standpoint, DLT supplies a framework for maintaining data integrity and confidentiality while facilitating secure information sharing. This proves particularly advantageous in scenarios necessitating the exchange of sensitive data among multiple parties.

10.2.2 Consensus Mechanisms

To assure congruence concerning the extant condition of the dispersed ledger and sustain data impeccability and safeguarding, consensus techniques stand as pivotal within blockchain technology (Bano et al. 2019). These techniques, like Proof of Work (PoW) and Proof of Stake (PoS), abolish the necessity for a centralized authority to authenticate and chronicle transactions (Wang et al. 2019). PoW, renowned through Bitcoin, amplifies the difficulty and energy consumption required for modifying the blockchain by compelling users to tackle intricate mathematical conundrums, thereby bolstering security against nefarious incursions (Nakamoto 2008; Vukolić 2016). On the other hand, PoS appoints validators predicated on their possession within the network, presenting augmented energy efficiency while upholding stringent security protocols (Xiao et al. 2020).

Delegated Proof of Stake (DPoS) and Practical Byzantine Fault Tolerance (PBFT) are two new consensus algorithms assigned to improve on the previous approaches in terms of security, scalability, and energy consumption (King and Nadal 2012). These dynamic consensus techniques greatly improve the stability of blockchain networks against many forms of cyber threats, such as 51% attacks and double spending attempts (Wang et al. 2019). In addition, a consensus system decentralizes data, which enhances the protection of data because there are no centralized points that can be easily hacked or modified (Zheng et al. 2018).

In the case of cybersecurity, consensus techniques allow for secure as well as fully transparent decision- making, which is essential in decentralized systems (Underwood 2016). Continued exploration of new consensus algorithms such as Proof of Authority (PoA) and Proof of History (PoH) may pave the way for the improvements of blockchain networks' security, privacy, and efficiency in cybersecurity practices as blockchain technology progresses (Yang et al. 2018). It finally shows that the constant evolution and advancement of consensus mechanisms play an essential role in determining the future of blockchain-based cybersecurity solutions, which would mean better protection of valuable data and important infrastructure (Christidis and Devetsikiotis 2016; Castro and Liskov 1999).

The consensus mechanisms are being created and enhanced day by day which indicates that it will be one of the key roles in delivering the necessary framework of blockchain cybersecurity solutions in the future. These solutions may afford better security for critical infrastructures and data that are sensitive in nature.

10.2.3 Smart Contracts

Smart contracts, which are self-executing codes employed on blockchain networks, have been deemed useful in strengthening cybersecurity measures and preserving individuals' privacy (Christidis and Devetsikiotis 2016; Szabo 1996). These automated agreements reduce the impact of outside interference and the human factor by performing predefined operations when certain conditions are met (Buterin 2014). In the area of cybersecurity, smart contracts can help to implement data management policies, security measures, and access controls directly and consistently across large systems (Xu et al. 2017a). Smart contracts on the blockchain are immutable, which create a firm and less susceptible layer for the cybersecurity industry to build upon during the design and assessment of security measures (Wood 2014).

In addition, smart contracts can facilitate simultaneous data sharing and confidentiality by strictly adhering to multiple-party access control and data protection policies (Zhang and Wen 2017). The interaction of smart contracts with external data sources through oracles improves the real-time identification of threats and the integration of automatic response procedures into the cybersecurity frameworks' incident response components (Haber and Stornetta 1991). It is necessary to admit that smart contracts can be buggy or can be exploited by people who know how it works (Atzei et al. 2017). Hence, security audits and formal verification are needed for smart contracts (Bhargavan et al. 2016). The current investigation is aimed at designing better and safer smart contracts that will improve the technology's capacity to implement cybersecurity and data protection policies in numerous industries and uses (Bartoletti and Pompianu 2017).

In the aspect of cybersecurity, smart contracts are found to be advantageous and operational tools that offer programmed as well as secure means of enforcing access control, protection policies as well as security measures. As it is capable of performing predetermined operations independently without intermediaries, it greatly enhances the general security of blockchain-based systems in terms of minimizing the risks of adversarial interference or human screw-ups, while enabling real-time threat detection and mitigation. But this shows that this is still an emerging technology in tackling advanced cyber threats in various industries, the flaws in smart contract code exhibit the imperative necessity of security checks and constant examination to construct fit-for-purpose architectures.

10.2.4 Blockchain Types

There are several forms of blockchain technology, each with unique characteristics and security concerns for applications in the field of cybersecurity (Imran 2018; Zheng et al. 2018). Platforms such as Ethereum and Bitcoin are highly open and decentralized yet bring numerous issues for data sensitivity in sensitive cybersecurity cases (Nakamoto 2008; Swan 2015). On the other hand, private blockchains are more flexible with data and participants' access, which means that they will be relevant for companies that are obliged to follow data protection legislation or have strict security measures (Zheng et al. 2017). Consortium or federated blockchains give the authority to manage the network to a set of predetermined members, combining the features of public and private ones. This can be beneficial in inter-organizational cybersecurity projects.

Public, as well as private chains, are integrated with hybrid blockchains to create flexible options that can vary according to the needed cybersecurity levels in various businesses (Samaniego et al. 2016). The blockchain ecosystem has been extended with the use of layer 2 and sidechains that address the issue of scalability while maintaining the security factor given by the base blockchain architecture (Wang et al. 2019). Additionally, developing blockchains that have quantum-safe mechanisms implies the ability to counter threats that result from development in quantum computing (Chen et al. 2016). Selecting the right type of blockchain for the implementation of cybersecurity solutions also rests more and more on the progress in the field or legal restrictions or even the balance between performance, privacy, and transparency (Christidis and Devetsikiotis 2016; Pilkington 2016).

The public blockchains offer high decentralization and transparency on the other hand private and consortium blockchains help to govern the data access and themselves due to which, it can be effective for the demands of multifaceted organizations and requirements of law. All these capture the nature of this field and why it is important to be very careful in choosing the right blockchain type given certain cybersecurity needs, legal issues, and the trade-off between throughput, anonymity, and openness given that these technologies are constantly evolving.

10.3 Blockchain and Cybersecurity

Blockchain security is a robust risk management system designed specifically for blockchain networks. By utilizing cybersecurity frameworks, assurance services, and industry best practices, it effectively minimizes the vulnerabilities to attacks and fraud.

10.3.1 Blockchain Security Features

Cybersecurity is an interesting area where blockchain is quite valuable because it has a wide range of security features (Zheng et al. 2018, 2017). Due to the distributed structure of blockchain, it is less vulnerable to acts of terrorism and technical faults and has increased reliability against attacks and failures (Buterin 2014; Swan 2015). Each block is calculated through cryptographic hashing, which results in a distinct value that shows evidence of any modifications made to the stored information (Nakamoto 2008; Boneh and Naor 2000). In a blockchain, public-key cryptography is applied to the digital signatures and identity confirmation for cybersecurity processes to be genuine (Narayanan et al. 2016).

Proof of work or proof of stake assures all nodes are in a consensus on the current state of the ledger, mitigating fraud and unauthorized alteration (Bano et al. 2019; Wang et al. 2019). Since blockchains are resistant to alteration, they also maintain a clear paper trail of all transactions, thus increasing accountability and transparency in the cybersecurity interface (Imran 2018; Kosba et al. 2016). Smart contracts help in the management and implementation of security policies since they are less likely to produce mistakes in their execution (Christidis and Devetsikiotis 2016; Szabo 1996). Also, some blockchain models have permissioned blockchains, which enable fine-grained access control on data and address privacy and other security needs in some cybersecurity applications (Peck 2017).

In cybersecurity frameworks, the fact that data stored in the blocks cannot be altered and smart contracts can operate autonomously bolsters the principles of responsibility and policy compliance. However, it is valid to notice strong security protections and to have several cybersecurity risks in mind. They are implementation issues, problems of scale, and potential issues with smart contracts. Similarly, choosing between public, private, or hybrid blockchain structures has its pros and cons in terms of control, privacy, and transparency that should be measured against organizational needs and legal frameworks.

10.3.2 Blockchain Vulnerabilities

Though blockchain technology offers numerous benefits, it has not escaped vulnerability issues that can hinder cybersecurity (Zheng et al. 2018, 2017). One of these risks is a 51% attack, an action when an adversary hijacks the majority of the network's hashing power in order to alter transactions (Vukolić 2016; Heilman et al. 2015). This is particularly dangerous, mainly because there is potential for smaller blockchains to be affected. They can also be susceptible to other forms of attacks such as integer overflows, which if exploited, can lead to gigantic losses and lessen the credibility of blockchain-based security solutions (Atzei et al. 2017). Furthermore, quantum computers in the future can potentially crack the algorithms used by blockchain to ensure its security (Chen et al. 2016).

Another critical imperative is that there may be violations of privacy (Zhang and Wen 2017). For this reason, the privacy of individuals can be compromised due to public blockchains as a result of their open access. Thus, it is imperative that privacy-preserving steps are not only integrated but done so meticulously (Kosba et al. 2016). In blockchain network systems, Sybil attacks are possible when an attacker controls multiple fake identities in order to wield more power in consensus processes (Douceur 2002). As seen in previous cases in different blockchain projects, the manifestation of errors made in the code or in the protocols and implementations of blockchains have led to attacks (Bhargavan et al. 2016).

In addition, although blockchain's decentralization and immutability are often a bonus for security, they are also its downside since they complicate the process of removing any content that may be unlawful or malicious (Peck 2017). This poses both an ethical and a technical challenge to cybersecurity practitioners in their endeavors. Such privacy issues in public blockchains and Sybil attack risks mean that there is a need to address the design of privacy-preserving practices, and also identity verification procedures appropriately (Samaniego et al. 2016). While blockchain's attribute of recording information in an immutable and cryptographically secure manner offers several benefits, it becomes a conundrum for cybersecurity practitioners seeking to eradicate nefarious information (Douceur 2002; Zhang et al. 2019).

10.3.3 Blockchain Attacks and Countermeasures

Reactively, the built-in security mechanisms in implementing blockchain systems can be exposed to different classes of attacks (Zheng et al. 2018; Atzei et al. 2017)). Preventing the leakage of information and anonymity requires protection, which is vital for such systems (Zhang and Wen 2017; Zhang et al. 2019). The general idea is to decentralize the network by making the architecture less centralized while adopting more secure consensus algorithms, which can reduce the chances of being subject to 51% attacks (Vukolić 2016; Heilman et al. 2015). To counter Sybil attacks which are created using many fake accounts, limits may be placed on participation when staking is based on the number of accounts, or reputation systems may be employed (Douceur 2002; Kiayias et al. 2017).

There are several protocols that have been created for heuristic proper detection and prevention of such exploits before decoding smart contracts (Bhargavan et al. 2016; Kosba et al. 2016). Greater confirmation times and checkpoints also act as measures against double-spending attacks, though they are more useful for small underlying networks (Karame et al. 2012). For instance, one should diversify the node connections and adopt measures for eclipse attack detection, as an eclipse attack is a situation where a single and malicious node disrupts an honest network (Heilman et al. 2015). Cryptographic algorithms that are immune to quantum attacks are another anticipatory measure, even though they are still at the abstract level presently (Chen et al. 2016). Furthermore, as potential privacy violations may occur with the implementation of public blockchains, techniques like secure multi-party computation and zero-knowledge proofs are used (Samaniego et al. 2016; Sasson et al. 2014). This helps to maintain the data privacy features while at the same time retaining the transparency benefits that arise from blockchain applications (Boneh and Naor 2000; Peck 2017).

There is no doubt that the blockchain community has proved its working model in its pre-emptive approach to dealing with established risks as seen by its development of formal verification standards as well as automated audit tools for smart contracts. Further, impeccable measures such as checkpoints and longer confirmation times to prevent double-spending have been orchestrated. On the same note, the advancement of blockchain security features is also evidenced by the shifting efforts of the leading scholars to develop quantum-resistant cryptographic functions while also incorporating safe multi-party computation and zero-knowledge proofs.

10.4 Blockchain Applications in Cybersecurity

Various use cases of the blockchain system have been found to improve the cybersecurity system in various industries (Zheng et al. 2018, 2017). Blockchain-based identity management solutions minimize the vulnerability to ID theft and other unauthorized access since users' identity data are safer and decentralized (Zhang and Wen 2017; Othman and Callahan 2018). For the reason of controlling the information exchange and access in secure systems and to create immutable records, blockchain is also applied in secure communication and access management systems (Boneh and Naor 2000). Additionally, threat intelligence platforms using blockchain enable timely and secure sharing of threat information between organizations to collectively enhance their protection against continually emerging cyber threats (Zhang et al. 2019; Liang et al. 2017b).

10.4.1 Identity and Access Management

With blockchain interoperability, Identity and Access Management (IAM) efficiency in terms of security, user control, and privacy has greatly improved (Zhang and Wen 2017; Othman and Callahan 2018). IAM solutions leveraging the blockchain approach allow for avoiding a centralized system and its inherent vulnerabilities and massive data loss due to the mentioned shortcomings (Shrier et al. 2016). Self-accountable digital identification (SI) solutions based on blockchains are designed to provide users with full control over their identity, improving privacy and mitigating overdependence on centralised power (Boneh and Naor 2000; Zhang et al. 2019). The permanency of records within the blockchain guarantees that identity-related transactions and access logs cannot be altered, thereby offering a credible certification trail for purposes of compliance and forensics (Zheng et al. 2018, 2017).

Current IAM solutions use smart contracts for implementing and managing access control policies at the blockchain layer for uniform and trustworthy permissions over sophisticated systems (Christidis and Devetsikiotis 2016; Szabo 1996). They also minimize the possibilities of fraud and identity theft and enhance secure and efficient identity assurance practices (Atzei et al. 2017; Elsden et al. 2018). Zero-knowledge proofs help to establish identity or characteristic proof without exposing extra information in IAM systems implemented through blockchain (Sasson et al. 2014). As organizations continue to integrate blockchain for IAM, there is the emergence of standardization for blockchain identities to facilitate interoperability with other blockchain platforms and enterprise systems (Liang et al. 2017b).

Besides removing single confinement and reducing vulnerabilities to hack attacks, the use of IAM with blockchain presents efficient identity confirmation processes, smart contract-based access control, and secure audit trails (Kosba et al. 2016; Kshetri 2017). Better, more secure, and less infringing identities of users interconnected within numerous systems and networks are being enabled by the ongoing standardization of the interoperability layers and new cryptographic technologies like zero-knowledge proofs.

10.4.2 Data Privacy and Protection

In the realm of cybersecurity, blockchain technology plays a role, in safeguarding data privacy and security. By employing a framework blockchain effectively minimizes the vulnerability to data breaches by eliminating points of failure. It utilizes methods such as hash functions and public key cryptography to ensure secure data encryption and integrity verification. The implementation of zero-knowledge proofs within blockchain systems enables verification of information without disclosing data thereby enhancing confidentiality in transactions and data sharing. Additionally, smart contracts automate data access. Uphold privacy standards. The unchangeable nature of records facilitates compliance with data protection regulations like the General Data Protection Regulation (GDPR) by offering a history of data access and modifications. Private and permissioned blockchains provide organizations with control over data visibility and access rights enabling them to strike a balance between confidentiality and transparency. Ongoing exploration into privacy-enhancing techniques such as party computing and homomorphic encryption aims to enhance the efficacy of data protection measures as blockchain technology advances.

10.4.3 IoT Security

When it comes to security blockchain technology presents an opportunity to address security challenges associated with Internet of Things (IoT) devices. By enhancing IoT device authentication mechanisms and ensuring data integrity the decentralized structure of blockchain can help mitigate risks linked to control systems. The immutable ledger feature of blockchain allows IoT networks to maintain tamper records of device operations and data exchanges making tasks, like analysis and anomaly detection more achievable. Smart contracts, on blockchain platforms, handle access management. Enforce security policies for devices ensuring that security controls are consistently applied across various device types. Blockchain based identity management systems provide a solution, for managing the identities of billions of devices thereby mitigating risks.

The technology, including the blockchain, is used in IoT data markets to enable secure and private sharing of data amongst devices and people for the purpose of boosting new technologies while still protecting personal data. Lightweight blockchain protocols that are intended for IOT devices, which have limited resources deal with performance and energy issues in implementing IOT security. This holds hope for another generation of adaptive security models that are more resilient to changes in the landscape of the Internet of Things due to its convergence with other cutting-edge technologies like edge computing and artificial intelligence. In order to improve the overall safety position of IoT systems, smart contracts can allow control over device IDs so as to scale up or down depending on the number of nodes and automated reinforcement of other security policies using blockchains. Unlike lightweight blockchain protocols that take care of resource constraints in IoT devices, there is a possibility for combining it with emerging fields such as edge computing and artificial intelligence thus creating flexible security frameworks that are more robust as the internet ecosystem evolves further.

10.5 Blockchain and Data Security

In today's digital world, keeping data safe and accurate has become a big worry. This is true since companies and organizations still depend a lot on digital information. But this dependence brings its own problems. As more and more digital data piles up smart cyber-attacks are also becoming more common. These threats put the three main parts of information security at serious risk: keeping data secret, making sure it stays correct, and being able to access it when needed.

10.5.1 Data Encryption in Blockchain

Data encryption plays a key role in boosting the security and privacy of blockchain systems in cybersecurity applications (Zheng et al. 2018, 2017). Blockchain technology uses cryptographic hash functions as a standard to keep data intact, but people often add extra encryption methods to safeguard sensitive data on the network (Boneh and Naor 2000; Tschorsch and Scheuermann 2016). Blockchain networks often use symmetric encryption algorithms like Advanced Encryption Standard (AES) because they can encrypt large amounts of data before adding it to the blockchain (Zhang and Wen 2017). Asymmetric encryption, which uses public-key cryptography, makes sure data stays private and unchanged in blockchain networks by offering secure key exchange and digital signatures (Nakamoto 2008; Narayanan et al. 2016).

Cutting-edge encryption methods, like homomorphic encryption, have exciting uses in blockchain systems that want to protect user privacy (Sasson et al. 2014; Gentry 2009). Mixing encryption with zero-knowledge proofs lets people check encrypted data without showing the actual info, which boosts the privacy of blockchain transactions (Samaniego et al. 2016; Zhang et al. 2019). Blockchain systems are starting to use quantum-proof encryption to get ready for possible threats from quantum computers in the future (Chen et al. 2016). As blockchain tech moves forward, ongoing studies on fresh encryption methods and how they fit into blockchain designs are making blockchain-based cybersecurity tools even better at protecting data (Liang et al. 2017b; Kshetri 2017).

10.5.2 Data Integrity and Immutability

Blockchain technology's use of an immutable ledger significantly enhances data security (Zheng et al. 2018, 2017). To manipulate any information on a blockchain, a consensus from the majority of network participants would be required, effectively altering the entire chain (Bano et al. 2019; Wang et al. 2019). Cryptographic hashing, a specialized mathematical function, assigns each data block a unique identifier (hash), making the data highly resistant to tampering (Boneh and Naor 2000; Tschorsch and Scheuermann 2016). Any attempt to modify the data results in a change of the hash, which is immediately detectable across the entire network (Zhang and Wen 2017; Underwood 2016). The implementation of immutable records promotes data transparency and trustworthiness by creating a verifiable audit trail (Kosba et al. 2016; Kshetri 2017). This feature facilitates the easy detection of unauthorized data access or modifications, thereby enhancing the overall security of data within the blockchain network (Zhang et al. 2019; Samamiego et al. 2016).

It is imperative to note that an indisputable record is maintained through consensus and cryptographic hash functions in the digital ledger of blockchain technology, implying a significant potential for data accuracy and reliable record-keeping. Inbuilt, it gives the functionality of an audit trail that not only attests to transparency and credibility of the source of information but also enhances the security of data as it is easier to detect unauthorized alteration or breach of the blockchain network.

10.5.3 Data Access Control

The openness of blockchain technology facilitates enhanced user data ownership, marking a significant shift in data management paradigms (Zhang and Wen 2017; Othman and Callahan 2018). This user-oriented approach enables the implementation of fine-grained access control through smart contracts, which are self-executing codes written on the blockchain (Christidis and Devetsikiotis 2016; Szabo 1996). Smart contracts can be utilized to define access rights for specific data and under what conditions, eliminating the need for centralized authorities (Xu et al. 2017b). The absence of a centralized point vulnerable to hacking minimizes the risk of unauthorized access and data theft (Zheng et al. 2018, 2017). Furthermore, blockchain-based access management allows users to selectively disclose specific data attributes while preserving the overall privacy (Samaniego et al. 2016; Do et al. 2017). This granular control over data exposure represents a significant advancement in balancing data utility and user privacy in decentralized systems (Zhang et al. 2019; Liang et al. 2017b). A significant deviation from centralized systems, decentralized management of data access is done through smart contracts, and it offers better security and individual control over personal information. This decentralized approach can revolutionize data privacy and governance across multiple sectors by putting the power of deciding data sharing directly in the hands of the users and reducing the vulnerability of data leaks and thefts.

10.6 Blockchain and Privacy

The integration of blockchain technology is revolutionizing cybersecurity as it offers protection in handling of transactions and data storage. However, data privacy is still an issue to consider while dealing with this open ecosystem. All data is stored in public ledgers which is not suitable for sensitive data. Some of the proposed solutions to this issue include permissioned blockchains and other cryptographic techniques such as zero- knowledge proofs to find a balance between anonymity or privacy, security, and transparency. In an age where privacy is a major issue, this area of ongoing investigation is crucial to realizing the full potential of the blockchain technology.

10.6.1 Privacy-Preserving Techniques

This is because all the data on a public blockchain is posted and can be seen by anyone which could compromise privacy (Zhang and Wen 2017; Zhang et al. 2019). Several solutions are being developed to address this problem (Samaniego et al. 2016; Sasson et al. 2014). Zero- knowledge proofs allow the verifier to prove information to the prover without revealing the actual information (Sasson et al. 2014; Gentry 2009). Similarly, methods such as homomorphic encryption enable computations to be made on data without disclosing the data to other parties (Gentry 2009; Acar et al. 2018). In addition, permissioned blockchains enhance privacy in specific applications by restricting access to only those with permission to do so. Also, users can share data with blockchain apps only when necessary and manage it with SSI frameworks in place. These methods demonstrate potential solutions to privacy issues and the potential for increased blockchain adoption.

10.6.2 Zero-Knowledge Proofs

The particularities of blockchain ecosystems have made the use of Zero-Knowledge Proofs, or ZKPs, a powerful cryptographic instrument for enhancing privacy

(Samaniego et al. 2016; Sasson et al. 2014). Thus, these approaches can show ownership of specific data without revealing the data (Gentry 2009). This enables users to ascertain their suitability for transactions or resources in a blockchain without exposing their details (Zhang et al. 2019; Jawurek et al. 2013). More recent architectures for ZKPs are Zero-Knowledge Scalable Transparent Argument of Knowledge (zk-STARKs) and Zero-Knowledge Succinct Non- Interactive Argument of Knowledge (zk-SNARKs), which have better scalability and efficiency than early ZKPs and thus are more applicable to real blockchain systems (Acar et al. 2018; Ben-Sasson et al. 2019). While ZKP has a lot of applications for private blockchains, exploration continues to overcome such challenges as high-proof generation and verification (Liang et al. 2017b; Tomescu et al. 2020).

The overall security of the blockchain system is enhanced by the ability to make authorization for transactions or resource use without revealing the identity. This could lead to increased adoption of blockchain technology in fields like finance and healthcare which deal with client information. Such privacy-enhancing technologies might be more useful for large-scale blockchain applications as lighter variants like zk-STARKs and zk-SNARKs are devised. This scalability might contribute to the increased usage of blockchains that ensure privacy further. If ZKPs are incorporated into more blockchain use cases, standardization and formulation of best practices might be required. This could reduce possible weaknesses and ensure that work is done uniformly across the organization.

Thus, the applicability and usefulness of ZKP techniques have increased with the development of newer and more efficient techniques like zk-STARK and zk-SNARK. ZKPs possess the potential for private blockchains for the industries dealing with the privacy of data such as the financial and health sector. However, the chapter is assembled on tackling challenges such as proof generation and proof checking. It has great potential to be adopted broadly that solutions to improve privacy on blockchain still have a lot of scope, but it is necessary for them to be standardized and follow best practices to ensure that they are not susceptible to certain common risks. ZKPs have the potential to increase the usage of privacy-based blockchain solutions in a number of sectors as the technology advances and is implemented in the application of blockchain.

10.6.3 Confidential Smart Contracts

In blockchains, there is a growing trend in the usage of anonymous programming execution known as confidential smart contracts which are seen as a method to bolster data privacy (Szabo 1996; Samaniego et al. 2016). These specialized contracts encode the data contained within the contract using certain cryptographic methods (Gentry 2009; Zhang et al. 2016). Apart from this, all the rest of the smart contract logic remains transparent for the purposes of controlling its auditability while the data remaining sensitive ensures that only those having a decryption key are able to retrieve the information (Zhang et al. 2019). As compared to smart contracts fully private, which can cause issues of opacity and extremely cumbersome enforcement procedures, this entails a number of advantages (Liang et al. 2017b; Steffen et al. 2019). If there are many use cases that require finding a balance between two factors–audited and private—then for such an intermediate supply chain to protect privacy or safe data trading, smart contracts, which are confidential, have a tremendous amount of potential (Kshetri 2017; Kosba et al. 2016). A prospective solution to the lack of both privacy and openness on blockchains is crypto contracts, which provide secure data processing as well. It is crucial for private-preserving processes, such as supply chain management or data marketplaces that demand responsibility along with privacy-preserving capability.

10.7 Case Studies

Blockchain technology has been seen to hold huge potential in disrupting several industries especially for businesses that rely on quality, security and credibility of data. This introduction lays the groundwork for two case studies that examine how blockchain technology is being applied to cybersecurity in two distinct industries: This positioning strategy fits well with two specific industry categories of operations: financial services and healthcare. The following case studies will aim to demonstrate how blockchain technology works in an appropriate cybersecurity issue within each sector. In the following, we will address the specific advantages, challenges, and potential future implications of using blockchain in security solutions based on real-life applications.

10.7.1 Blockchain in Healthcare Cybersecurity

Case Study 1: Medicalchain's MyClinic.Com

In 2018, Medicalchain, a company registered in the United Kingdom, launched MyClinic. com), a healthcare startup that uses blockchain technology to safely store and exchange patient records. Its use allows patients to share their health data securely but also to receive consultations from remote physicians. Undefined com makes use of two different blockchain structures: This approach includes an EHR blockchain for storing electronic health records and a permission blockchain for rights management. Security is assured in this system in so far as the data integrity, patient data confidentiality, and secure transfer of patients' data between healthcare providers is concerned. There are positive impacts noted with regard to patient data control and the ease of telemedicine services on the platform. However, there are challenges that have limited adoption and implementation into current mainstream healthcare platforms.

MyClinic from Medicalchain is an application that employs blockchain technology. com offers a convenient approach to telemedicine and decentralized healthcare record management. The ability of the platform's dual blockchain to allow for secure information sharing while preserving data integrity and confidentiality was permitted. Preliminary evidence suggests an increase in patient self-governance and the possibility of remote health care services. Nevertheless, the need for broader recognition of the proposed regulations and integration issues with the existing healthcare platforms may limit the platform's popularity.

Case Study 2: IRYO Network

In 2018, IRYO Network was launched as a global blockchain-based platform for healthcare that aims to provide full ownership of medical records for the patient while ensuring the information to be non-disclosable. To ensure patient data security the system utilizes encryption and maintains zero knowledge about the data. To.

control the access limits and maintain an unalterable audit trail, the technology of blockchain is incorporated. However, to implement this concept, IRYO has partnered with healthcare providers in several countries. Better patient trust and data sharing have been highlighted by early users of the systems. It must however, meet challenges concerning legal restraints in a number of countries and the need for massive changes in current healthcare IT systems.

A detailed and expansive plan for the application of blockchain in handling patient-oriented health care data is provided by IRYO Network. Challenging problems of the healthcare sector are solved by the platform, focusing on data ownership, protection and harmonization. Nevertheless, the amount and complexity of the regulation is too high, and a massive restructuring of the IT framework is necessary if it is to be put into practice. However, early adoption indicates that IRYO faces some challenges that it needs to overcome in order to be profitable and sustainable over the long term.

10.7.2 Blockchain in Financial Services Cybersecurity

Case Study 1: Mastercard's Blockchain-Based Cross-Border Payments

The aim for Mastercard's latest expansion of its blockchain-enabled cross-border payment system is to increase transaction speed and reduce risk. The permissioned blockchain technology used in the system also facilitates real-time settlements of payment providers and banks. It offers enhanced identity management, tamper-proof transaction histories, and encryption from end to end to address the cybersecurity issue. The blockchain eliminated the possibility of fraud and cyberattacks that may be associated with traditional cross-border payment systems. Mastercard tested this technology in 2022 with several major banks with results of faster processing times and reduced cost. Further, smart contracts are employed to ensure Anti-Money Laundering (AML) compliance and automatically check customers against the blacklist. Preliminary findings indicate a noteworthy decline in fraudulent payments and enhanced adherence to regulations.

This is a significant milestone for Mastercard which has embarked on implementing a cross-border payment system based on the blockchain. Thus, by integrating the enhanced methods of identification of customers and end-to-end encryption with permissioned blockchain system for real-time settlements, the comprehensive approach to minimize the security and effectiveness challenges of cross-border payments is illustrated. The system holds the potential to revolutionize global financial transactions, by the stated elimination of the fraud risk and cyberattacks, faster transaction, and low cost.

Case Study 2: Bank of America's Patent for Blockchain-Based Security System

The patent for the blockchain-based security system that Bank of America just implemented expires in the year 2021. This solution employs the use of blockchain technology to effectively secure and regulate access to some of the most sensitive data within the networks of the bank. The blockchain works as decentralized and unalterable list of transactions that records all changes and attempts to enter personal information. They also employ smart contracts to ensure adherence to the security protocols and manage access control mechanisms. The system has therefore enhanced the bank's capability in detecting insider threats and thwarting illicit access. As for the security problems involving data access, Bank of America registered a dramatic decrease in the first year after adopting the concept. In addition to the real-time auditing capabilities, the blockchain solution allows for immediate detection and prevention of potentially compromising scenarios.

Applying blockchain technology to such actual cybersecurity issues of the financial industry advanced when Bank of America implemented a blockchain-based security system for managing access to sensitive data. The applicability of blockchain in the improvement of transparency, accountability, and security in managing sensitive data is illustrated by the deployment of smart contracts for implementing access control measures and blockchain as a distributed, audit-proof data storage for documenting data access attempts. The system's accomplishment is exhibited by the observed decline in security incidents pertaining to data access, highlighting the definite advantages of blockchain in cybersecurity applications.

10.8 Future Directions

Cybersecurity is one of the realms that demonstrates blockchain possibilities of the area, encompassing AI incorporation, privacy-preserving smart contracts, scaling solutions, and interconnectivity. By addressing current gaps and enhancing the efficiency, reliability, and adaptiveness of the blockchain-based technology for various cybersecurity contexts, the recent innovations may even revolutionize the notion of cybersecurity.

10.8.1 Emerging Trends and Challenges

There is a growing focus to explore how the application of the blockchain technology can promote cybersecurity in different sectors. It is distributed and once written is impossible to alter for it is appropriate for or use like tamper-proof audit trails, identity management, and archival storage.

Scalability solutions: It is a continuous challenge to handle blockchain scalability difficulties without sacrificing security by implementing layer-2 solutions and sharding strategies (Wang et al. 2019, 2018). This pattern reflects the increasing need in enterprise-level cybersecurity applications for high-performance blockchain systems.

Interoperability: A growing trend is improving data interchange and cross-chain connectivity while preserving security among various blockchain networks (Liang et al. 2017b; Belchior et al. 2021). The objective of this development is to establish a more efficient and linked blockchain ecosystem for all-encompassing cybersecurity solutions.

Smart contracts that protect privacy: Work is being done to create private smart contracts that can process encrypted data without disclosing private information (Steffen et al. 2019). This trend responds to the increased demand for anonymity in blockchain-based cybersecurity applications, particularly in industries managing sensitive data.

AI integration: A growing trend in cybersecurity is combining blockchain technology with artificial intelligence to improve threat detection and automate security responses (Liu et al. 2020). With this connection, blockchain systems should have more flexible and effective security mechanisms.

Energy efficiency: One of the main challenges facing proof-of-work blockchains is the development of more energy-efficient consensus processes (Vukolić 2016). This pattern indicates the growing attention that cybersecurity is paying to sustainable blockchain solutions.

Decentralised identity: There's a growing movement to advance self-sovereign identification solutions on blockchain to improve security and privacy in digital interactions (Mühle et al. 2018; Othman and Callahan 2018). The goal of this development is to increase overall cybersecurity while giving people more control over their digital identities.

Integrating blockchain and AI for improved threat detection is a good approach to learn how the two modern technologies can benefit from each other. However, because of technological considerations and possible security implications therein, use a more sceptical approach when considering any negative side effects or challenging to implement regarding these trends. Besides, energy efficiency is an important issue, however, a comprehensive review of the possible impact on the security and decentralisation characteristic of blockchain systems would provide a more objective perspective.

10.8.2 Future Prospects

The future investigation on cybersecurity employing blockchain technologies should focus on new problems and new opportunities. For sustainability against upcoming threats via quantum computing, blockchain systems have to devise quantum computing cryptographic techniques (Aggarwal et al. 2017). As a result, future work for applying blockchain in more cybersecurity usages should focus on examining more scalable mechanisms that retain security while augmenting the transaction rate. This is vital to understand how to enhance crosstalk blockchain connection without compromising its security since it is critical in developing comprehensive, cross-chain cybersecurity solutions. To address data privacy issues in a wide range of industries, it will be crucial to develop more novel privacy-preserving solutions for the Blockchain ecosystem, including Homomorphic Encryption (HE) and Zero-Knowledge Proofs (ZKP) (Hassan et al. 2019).

One emerging direction is to explore how blockchain technology can enhance Artificial Intelligence (AI) and Machine Learning (ML) to advance threat intelligence and automate responses (Liu et al. 2020). We are now starting to see the importance of focusing on the efficiency of consensus protocols to consider the environmental cost of blockchain. Even academic investigation on decentralized identity management systems based on blockchain technology may change the understanding and approach to identifying users and controlling access in cybersecurity (Yang et al. 2018). For blockchain-based security systems to be futureproofed, future studies will involve understanding how things such as post-quantum cryptography will impact existing blockchain paradigms and making migration plans. To keep firm cybersecurity, uphold, it is necessary to shift focus on governance structures for the decentralized blockchain networks which can in operation provide persistent security upgrades and protocol improvement.

It is essential to solve the scalability and interoperability issues for blockchain to be utilised more frequently in cybersecurity systems; It is also critical to understand the system's intricacy when discussing security enhancements. Some of suggested directions in enhancing cybersecurity capacities are analysing the effectiveness of privacy-preserving approaches for enhancing cybersecurity and integrating the blockchain with AI and ML for better identification of threats. It would have been useful to provide a critique of the above-discussed limitations of these strategies, and a critical discourse for the ethical implications of blockchain for cybersecurity solutions. Moreover, while dedicated energy-efficient consensus procedures, even though being designed for the purpose, may be reviewed here as more 'partial' in their approach, a more extensive analysis of decentralisation-security- efficiency balance could provide a less biased perspective on this issue.

10.9 Summary and Conclusion

An introductory knowledge of the system such as distributed ledger, Smart Contract and Consensus process has been explained in this chapter. It also examines the use of blockchain along with cybersecurity and spends time going over attacks, weaknesses, and protection measures. Of the blockchain cybersecurity applications, it discusses data privacy, supply chain cybersecurity, identity and access management cybersecurity, and cybersecurity of things on the Internet. The chapter describes the details on how blockchain technology provides data security and privacy through methods like private smart contract and zero-knowledge proof, access control, and data encryption and integrity. The real-life examples are discussed for cybersecurity in healthcare and the financial sector on how branding leaders–Mastercard and Bank of America–have integrated blockchain.

New developments and challenges in cybersecurity, application of blockchain and AI in synergy, energy efficient consensus algorithms, management of decentralized identity, post-quantum cryptographic solutions and architectural models of block chain networks are the topics discussed in this chapter. Lastly, using different examples mentioned in the given chapter, it is explained how possibly blockchain technology can enhance the cybersecurity standards since it provides a comprehensive information on the usage, benefits and the possible future trends concerning the protection of valuable assets and personal rights in the digital environment.

Thus, introducing the element of blockchain technology into cybersecurity has been a significant advancement toward enhancing privacy and preventing identity theft. Organisations are also capable of developing adequate security mechanisms against cyber threats through the structure of a decentralised ledger, smart contracts, and cryptography mechanisms of blockchain. By discussing examples of the specific sectors, such as banking and healthcare, the article establishes the value of the blockchain system for maintaining the data's integrity and controlling their access. The new competencies that will be necessary and the distributed security models which will have to be adopted will be other of the key challenges that will face cybersecurity professionals while embracing this new model fully and exploit the potential of blockchain technology.

References

Acar A, Aksu H, Uluagac AS, Conti M (2018) A survey on homomorphic encryption schemes: theory and implementation. ACM Comput Surv (Csur) 51(4):1–35

Aggarwal D, Brennen GK, Lee T, Santha M, Tomamichel M (2017) Quantum attacks on Bitcoin, and how to protect against them. arXiv:1710.10377

Atzei N, Bartoletti M, Cimoli T (2017) A survey of attacks on ethereum smart contracts (sok). In: Principles of security and tTrust: 6th international conference, POST (2017) Held as part of the European joint conferences on theory and practice of software, ETAPS 2017, Uppsala, Sweden, 22–29 Apr 2017, Proceedings, vol 6. Springer, Berlin Heidelberg, pp 164–186

Bano S, Sonnino A, Al-Bassam M, Azouvi S, McCorry P, Meiklejohn S, Danezis G (2019). SoK: consensus in the age of blockchains. In: Proceedings of the 1st ACM conference on advances in financial technologies, pp 183–198. (Oct 2019)

Bartoletti M, Pompianu L (2017) An empirical analysis of smart contracts: platforms, applications, and design patterns. In: Financial cryptography and data security: FC 2017 international workshops, WAHC, BITCOIN, VOTING, WTSC, and TA, Sliema, Malta, April 7, 2017, Revised Selected Papers 21. Springer International Publishing, pp 494–509

Belchior R, Vasconcelos A, Guerreiro S, Correia M (2021) A survey on blockchain interoperability: past, present, and future trends. ACM Comput Surv (CSUR) 54(8):1–41

Ben-Sasson E, Bentov I, Horesh Y, Riabzev M (2019) Scalable zero knowledge with no trusted setup. In: Advances in cryptology–CRYPTO 2019: 39th annual international cryptology conference, Santa Barbara, CA, USA, 18–22 Aug 2019, Proceedings, Part III 39. Springer International Publishing, pp 701–732

Bhargavan K, Delignat-Lavaud A, Fournet C, Gollamudi A, Gonthier G, Kobeissi N, Rastogi A, Sibut-Pinote T, Swamy N, Zanella-Béguelin S (2016) Formal verification of smart contracts: Short paper. In: Proceedings of the 2016 ACM workshop on programming languages and analysis for security, pp 91–96. (Oct 2016)

Boneh D, Naor M (2000) Timed commitments. In: Annual international cryptology conference. Springer, Berlin, Heidelberg, pp 236–254. (Aug 2000)

Buterin V (2014) A next-generation smart contract and decentralized application platform. White Paper 3(37):2–1

Castro M, Liskov B (1999) Practical byzantine fault tolerance. In: OsDI, vol 99, no 1999, pp 173–186. (Feb 1999)

Chen L, Chen L, Jordan S, Liu YK, Moody D, Peralta R, Perlner RA, Smith-Tone D (2016) Report on post-quantum cryptography, vol 12. US Department of Commerce, National Institute of Standards and Technology, Gaithersburg, MD, USA

Christidis K, Devetsikiotis M (2016) Blockchains and smart contracts for the internet of things. IEEE Access 4:2292–2303

Crosby M, Pattanayak P, Verma S, Kalyanaraman V (2016) Blockchain technology: beyond Bitcoin. Appl Innov 2(6–10):71

Do HG, Ng WK (2017) Blockchain-based system for secure data storage with private keyword search. In: 2017 IEEE world congress on services (SERVICES). IEEE, pp 90–93. (June 2017)

Douceur JR (2002) The sybil attack. In: International workshop on peer-to-peer systems. Springer, Berlin, Heidelberg, pp 251–260 (Mar 2002)

Elsden C, Manohar A, Briggs J, Harding M, Speed C, Vines J (2018) Making sense of blockchain applications: a typology for HCI. In: Proceedings of the 2018 chi conference on human factors in computing systems, pp 1–14. (Apr 2018)

Gentry C (2009) Fully homomorphic encryption using ideal lattices. In: Proceedings of the forty-first annual ACM symposium on theory of computing, pp 169–178. (2009, May)

Haber S, Stornetta WS (1991) How to time-stamp a digital document. Springer, Berlin Heidelberg, pp 437–455

Hassan MU, Rehmani MH, Chen J (2019) Privacy preservation in blockchain based IoT systems: Integration issues, prospects, challenges, and future research directions. Futur Gener Comput Syst 97:512–529

Heilman E, Kendler A, Zohar A, Goldberg S (2015) Eclipse attacks on {Bitcoin's}{peer-to-peer} network. In: 24th USENIX security symposium (USENIX security 15), pp 129–144

Imran B (2018) MASTERING BLOCKCHAIN: distributed ledger technology, decentralization, and smart contracts explained; distributed ledger. PACKT Publishing

Jawurek M, Kerschbaum F, Orlandi C (2013) Zero-knowledge using garbled circuits: How to prove non- algebraic statements efficiently. In: Proceedings of the 2013 ACM SIGSAC conference on computer & communications security, pp 955–966. (Nov 2013)

Karame GO, Androulaki E, Capkun S (2012) Double-spending fast payments in bitcoin. In: Proceedings of the 2012 ACM conference on computer and communications security, pp 906–917. (Oct 2012)

Kiayias A, Russell A, David B, Oliynykov R (2017). Ouroboros: a provably secure proof-of-stake blockchain protocol. In: Annual international cryptology conference. Springer International Publishing, Cham, pp 357–388. (July 2017)

King S, Nadal S (2012) Ppcoin: Peer-to-peer crypto-currency with proof-of-stake. Self-published paper, August, vol 19, no 1

Kosba A, Miller A, Shi E, Wen Z, Papamanthou C (2016) Hawk: the blockchain model of cryptography and privacy-preserving smart contracts. In: 2016 IEEE symposium on security and privacy (SP). IEEE, pp 839–858. (May 2016)

Kshetri N (2017) Blockchain's roles in strengthening cybersecurity and protecting privacy. Telecommun Policy 41(10):1027–1038

Liang X, Zhao J, Shetty S, Liu J, Li D (2017a) Integrating blockchain for data sharing and collaboration in mobile healthcare applications. In: 2017 IEEE 28th annual international symposium on personal, indoor, and mobile radio communications (PIMRC). IEEE, pp 1–5. (Oct 2017)

Liang X, Shetty S, Tosh D, Kamhoua C, Kwiat K, Njilla L (2017b) Provchain: a blockchain-based data provenance architecture in cloud environment with enhanced privacy and availability. In: 2017 17th IEEE/ACM international symposium on cluster, cloud and grid computing (CCGRID). IEEE, pp 468–477. (May 2017)

Liu Y, Yu FR, Li X, Ji H, Leung VC (2020) Blockchain and machine learning for communications and networking systems. IEEE Commun Surv Tutor 22(2):1392–1431

Mühle A, Grüner A, Gayvoronskaya T, Meinel C (2018) A survey on essential components of a self-sovereign identity. Comput Sci Rev 30:80–86

Nakamoto S (2008) Bitcoin: a peer-to-peer electronic cash system

Narayanan A, Bonneau J, Felten E, Miller A, Goldfeder S (2016) Bitcoin and cryptocurrency technologies: a comprehensive introduction. Princeton University Press

Othman A, Callahan J (2018) The horcrux protocol: a method for decentralized biometric-based self-sovereign identity. In: 2018 international joint conference on neural networks (IJCNN). IEEE, pp 1–7. (July 2018)

Peck ME (2017) Blockchain world-Do you need a blockchain? This chart will tell you if the technology can solve your problem. IEEE Spectr 54(10):38–60

Pilkington M (2016) Blockchain technology: principles and applications. In: Research handbook on digital transformations. Edward Elgar Publishing, pp 225–253

Radziwill N (2018) Blockchain revolution: How the technology behind bitcoin is changing money, business, and the world. 2016. Dan Tapscott and Alex Tapscott. Penguin Random House, New York, pp 348

Samaniego M, Jamsrandorj U, Deters R (2016) Blockchain as a service for IoT. In: 2016 IEEE international conference on internet of things (iThings) and IEEE green computing and communications (GreenCom) and IEEE cyber, physical and social computing (CPSCom) and IEEE smart data (SmartData). IEEE, pp 433–436. (Dec 2016)

Sasson EB, Chiesa A, Garman C, Green M, Miers I, Tromer E, Virza M (2014) Zerocash: decentralized anonymous payments from bitcoin. In: 2014 IEEE symposium on security and privacy. IEEE, pp 459–474. (May 2014)

Shrier D, Wu W, Pentland A (2016) Blockchain & infrastructure (identity, data security). Mass Inst Technol Connect Sci 1(3):1–19

Steffen S, Bichsel B, Gersbach M, Melchior N, Tsankov P, Vechev M (2019) zkay: specifying and enforcing data privacy in smart contracts. In: Proceedings of the 2019 ACM SIGSAC conference on computer and communications security, pp 1759–1776. (Nov 2019)

Swan M (2015) Blockchain: blueprint for a new economy. O'Reilly Media, Inc.

Szabo N (1996) Smart contracts: building blocks for digital markets. EXTROPY: J Transhumanist Thought (16), 18(2):28

Tomescu A, Abraham I, Buterin V, Drake J, Feist D, Khovratovich D (2020) Aggregatable subvector commitments for stateless cryptocurrencies. In: International conference on security and cryptography for networks. Springer International Publishing, Cham, pp 45–64 (Sept 2020)

Tschorsch F, Scheuermann B (2016) Bitcoin and beyond: a technical survey on decentralized digital currencies. IEEE Commun Surv Tutor 18(3):2084–2123

Underwood S (2016) Blockchain beyond bitcoin. Commun ACM 59(11):15–17

Vukolić, M. (2016). The quest for scalable blockchain fabric: Proof-of-work vs. BFT replication. In *Open Problems in Network Security: IFIP WG 11.4 International Workshop, iNetSec 2015, Zurich, Switzerland, October 29, 2015, Revised Selected Papers* (pp. 112–125). Springer International Publishing.

Wang S, Zhang Y, Zhang Y (2018) A blockchain-based framework for data sharing with fine-grained access control in decentralized storage systems. IEEE Access 6:38437–38450

Wang W, Hoang DT, Hu P, Xiong Z, Niyato D, Wang P, Wen Y, Kim DI (2019) A survey on consensus mechanisms and mining strategy management in blockchain networks. IEEE Access 7:22328–22370

Wood G (2014) Ethereum: a secure decentralised generalised transaction ledger. Ethereum Project Yellow Paper 151(2014):1–32

Xiao Y, Zhang N, Lou W, Hou YT (2020) A survey of distributed consensus protocols for blockchain networks. IEEE Commun Surv Tutor 22(2):1432–1465

Xu C, Wang K, Guo M (2017b) Intelligent resource management in blockchain-based cloud datacenters. IEEE Cloud Comput 4(6):50–59

Xu X, Weber I, Staples M, Zhu L, Bosch J, Bass L, Pautasso C, Rimba P (2017) A taxonomy of blockchain-based systems for architecture design. In: 2017 IEEE international conference on software architecture (ICSA). IEEE, pp 243–252. (Apr 2017)

Xu LD, Xu EL, Li L (2018) Industry 4.0: state of the art and future trends. Int J Prod Res 56(8):2941–2962

Yang Z, Yang K, Lei L, Zheng K, Leung VC (2018) Blockchain-based decentralized trust management in vehicular networks. IEEE Internet Things J 6(2):1495–1505

Zhang F, Cecchetti E, Croman K, Juels A, Shi E (2016) Town crier: an authenticated data feed for smart contracts. In: Proceedings of the 2016 aCM sIGSAC conference on computer and communications security, pp 270–282. (Oct 2016)

Zhang R, Xue R, Liu L (2019) Security and privacy on blockchain. ACM Comput Surv (CSUR) 52(3):1–34

Zhang Y, Wen J (2017) The IoT electric business model: using blockchain technology for the internet of things. Peer- to-Peer Netw Appl 10:983–994

Zheng Z, Xie S, Dai HN, Chen X, Wang H (2018) Blockchain challenges and opportunities: a survey. Int J Web Grid Serv 14(4):352–375

Zheng Z, Xie S, Dai H, Chen X, Wang H (2017). An overview of blockchain technology: architecture, consensus, and future trends. In: 2017 IEEE international congress on big data (BigData congress). IEEE, pp 557–564. (June 2017)

Chapter 11
Introduction to IoT and Security Issues

Jyoti Mante and Kishor Kolhe

Abstract The Internet of Things increases the facility by which data from digital devices can be gathered, analyzed, and perfected in numerous disparate industries. IoT, however, exposes several types of security violations, including compromise of trust, confidentiality, integrity, and resource availability. Among these, Distributed Denial of Service (DDoS) attacks are the most prevalent. They are the flooding of communication channels with networks under subjection to attacks indistinguishable from multiple IoT devices, these attacks yield the network resource availability. This chapter provides a deep insight into IoT, the security vulnerabilities of IoT, DDoS attacks on IoT systems, the types of these attacks, the vulnerabilities existing regarding where they can be conducted, and the usage of these IoT devices to create botnets. Also, it discusses in depth how AI and blockchain can be used as controls for preventing attacks. It also offers a comparative analysis and looks at the latest DDoS defense strategies for assistance in understanding. This chapter also includes an in-depth overview of IoT, including its applications, architecture, and componentry, with a focus on the projected increase of 30 billion connected devices by 2020.

Keywords Internet of Things (IoT) · DDoS · Botnets · Architecture · Application · Artificial Intelligence · Blockchain

J. Mante (✉) · K. Kolhe
Department of Computer Science and Engineering, School of Computer Engineering and Technology, Dr. Vishwanath Karad MIT World Peace University, Pune, India
e-mail: jyoti.khurpade@mitwpu.edu.in

K. Kolhe
e-mail: Kishor.kolhe@mitwpu.edu.in

K. Ahmad et al. (eds.), *Fostering Machine Learning and IoT for Blockchain Technology*,
Transactions on Computer Systems and Networks,
https://doi.org/10.1007/978-981-96-4078-2_11

11.1 Introduction

11.1.1 Definition of IoT

IoT has gained significant relevance due to the rapid growth in mobile devices, communication technologies, cloud computing, and data analytics (Aqeel 2020). Today, over seven billion people utilize the Internet for various activities such as emailing, social media interactions, reading, gaming, and online shopping. Due to the widespread usage of the internet, new trends have emerged that allow machines to operate independently within a global network, communicating and making decisions. Billions of devices can connect and exchange data over Internet Protocol (IP) in the world of the Internet of Things (IoT). Large volumes of data are produced by these networked things, which are then gathered, examined, and used to carry out operations and offer information for making decisions (Aqeel 2020). Add sensors and actuators, and the Internet of Things joins the larger class of cyber-physical systems. Smart grids, virtual power plants, smart homes, intelligent transportation systems, and smart cities are some examples of the inventions that fall under this category. The "things" in IoT encompass a wide array of devices, ranging from heart monitoring implants and biochip transponders on farm animals to live-streaming cameras for wildlife observation, sensor-equipped automobiles, DNA analysis tools for environmental monitoring, and devices aiding firefighters in rescue operations (Tiwary et al. 2018; Balaji et al. 2019). Legal perspectives often describe these "things" as an inseparable blend of hardware, software, data, and services. Given this foundation, the future of IoT spans numerous applications, from smart grids and smart cities to intelligent vehicles and smart electricity meters. This chapter explores the myriad applications of IoT in science and technology, offering a comprehensive literature review. It goes into several applications, ranging from smart urban infrastructure and agriculture to life-saving technology, and explores the architecture and components of the Internet of Things, emphasizing its salient aspects. Additionally, it compares IoT with machine-to-machine (M2M) communication, identifies some limitations, and examines existing protocols and security issues. The chapter concludes with potential future research directions, identifying open areas and challenges within the IoT landscape (Tiwary et al. 2018; Balaji et al. 2019).

11.1.2 Evolution of IoT

11.1.2.1 M2M Communication

Aspects of machine-to-machine (M2M) communication include interconnection and efficient communication between machines. As seen in Fig. 11.1, backend servers and wireless networks are typically what enable M2M communication (Wang et al. 2015). Device-integrated sensors gather sensory data, which is subsequently processed in

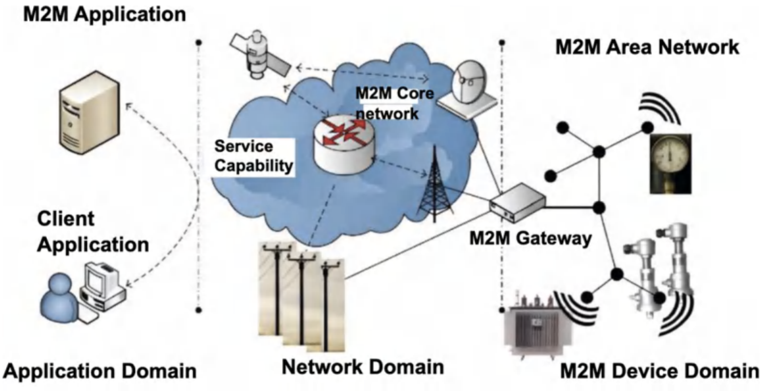


Fig. 11.1 Machine-to-machine (M2M) communications system

M2M applications and sent over a variety of network types, as shown by the data flow in Fig. 11.1 from right to left. Without human assistance, M2M communication enables robots to finish the communication process on their own. The Third Generation Partnership Project (3GPP), one of several groups working on M2M standards development, has made headway in defining M2M concepts, service needs, and functional structures (Wang et al. 2015). A radio access network designed exclusively for M2M communication was introduced by 3GPP in 2010. In many different applications, the integration of M2M communication with heterogeneous networks is becoming more and more prevalent. Upcoming advancements in 3G and 4G wireless technology should improve data transfer speeds, enabling a greater variety of M2M services. Mobile cellular networks are one type of wireless network where M2M communication can take place (Wang et al. 2015).

11.1.2.2 Wireless Sensor Networks

As shown in Fig. 11.2, wireless sensor networks (WSNs) are made up of many self-organizing sensor nodes scattered over open spaces in a certain configuration. The combination of these sensors monitors several environmental parameters. Enhancing our comprehension of the environment and enabling applications to make automated judgments based on specified criteria are the main objectives of gathering this data (Wang et al. 2015).

11.1.2.3 Sensor Web Enablement and Sensor Web

The Sensor Web Enablement was developed by the Open GIS Consortium. architecture in 2001. It is a service-based architecture and set of standards for a range of sensors. Languages for defining sensors, measurements, and other environmental

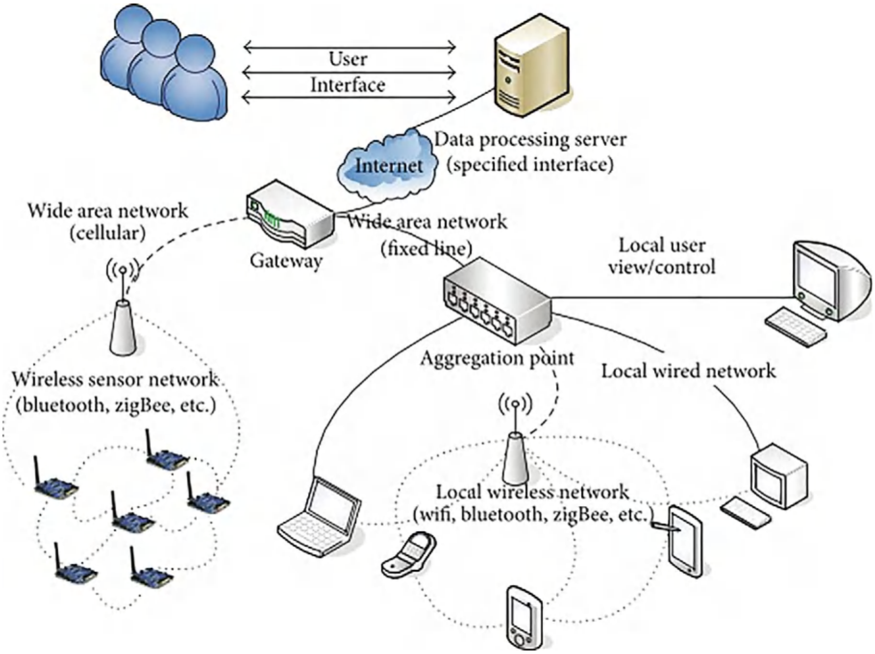
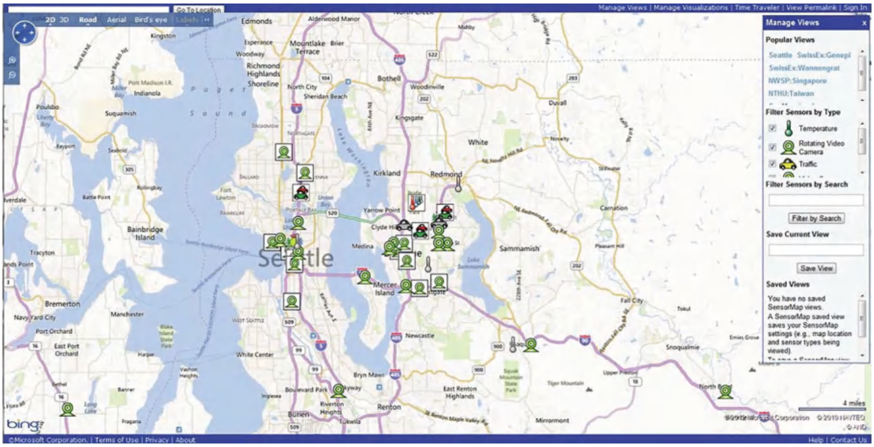


Fig. 11.2 Wireless sensor network (WSN)

elements are included in this architecture. In SWE, geographic data is an essential component (Wang et al. 2015). Developers can access and use different network sensors over the web or other networks to construct a variety of applications to fulfill user demands by combining sensor data with geographic information and making it available online. This integrated approach to managing sensor data, which is an extension of conventional sensor networks (SNs), is referred to as the "sensor web" (SW). Usually, SW is used in large-scale distributed systems composed of mobile sensor platforms called pods that are wirelessly broadcasting. Every pod in the same measurement cycle is aware of the data that other pods have measured. As SNs have developed, a new field of study called SW has arisen, reflecting spatially dimensional real-time information systems. Real-time environmental monitoring platforms are being developed through projects like Microsoft's "Sensor Web" and the University of California's "JPL Sensor Web." To further demonstrate the possibilities of this technology, Microsoft has developed an online platform called "Sensor Map" that gives users access to real-time data from several sensors as shown in Fig. 11.3 (Wang et al. 2015).



You have no saved SensorMap views.
A SensorMap saved view saves your SensorMap settings (e.g., map location and sensor types being viewed).

Fig. 11.3 An instance of Microsoft's "Sensor Web" initiative

11.1.2.4 WoT-Web of Things

As the Web has evolved, so has traditional Web 2.0 technology to handle the diversity of data, networks, and devices. The online of Things (WoT), a concept that promotes communication and information exchange between smart devices and the online, is the result of this progress (Wang et al. 2015). By incorporating actual sensors into the traditional web foundation, the Web of Things (WoT) closes the gap between the virtual and physical worlds. It links sensors to the Internet using common application protocols like HTTP. Smart gateways with web servers installed that provide RESTful access to a range of devices were the first to be used in the WoT. Using this method, programmers might design web services that interface with real-world data streams in apps. Devices capable of functioning as IP-enabled devices with embedded web servers are those that can be integrated directly. For devices without the resources to run web servers, indirect integration is employed. Here, an intermediary proxy serves as a gateway for the web server, enabling communication between the devices and the internet. Additionally, this proxy makes it possible to integrate heterogeneous data as web services, including RFID or sensor data (Wang et al. 2015). The notion of the "E-Skin of Earth" has surfaced, with the objective of establishing an all-encompassing network of up-to-date geographic data via sensor webs. This concept facilitates large-scale distributed systems in which wireless data transmission and measurement sharing occur between movable sensor platforms, or pods, inside a network.

11.1.2.5 Semantic Sensor Networks

The composition of wireless sensor networks (WSNs) is changing more often as they grow, and a wider range of sensors are being installed. Semantic sensor networks (SSNs) improve semantic interoperability and integration by employing ontologies and semantic technologies to abstract and explain sensor data. Ontologies abstract and characterize sensors, allowing high-level specifications for data management, organization, querying, comprehension, and control. The W3C Semantic Sensor Network Incubator Group developed ontologies that specify the functionalities of sensors and sensor networks between 2009 and 2011 (Wang et al. 2015).

As seen in Fig. 11.4, (Wang et al. 2015) which displays semantic sensor network ontology classes and their use cases, the essential components of SSNs are use cases and ontology modules. In the future, the ontology inside a linked sensor data environment needs to be standardized in order to connect the Internet of Things and the Internet of Services. Semantic web technologies require comprehensive ontology definitions, yet creating a single ontology that is suitable for all scenarios remains a challenging undertaking. A key component of the internet is the Internet of Things (IoT), which is a virtually connected network as shown in Fig. 11.5 (Wang et al. 2015). It is possible to link, query, and combine virtual representations of objects by utilizing web services and semantic web technologies. The semantic vision tackles data management problems brought on by the massive volumes of data that smart items and resources accessible via web interfaces exchange (Wang et al. 2015).

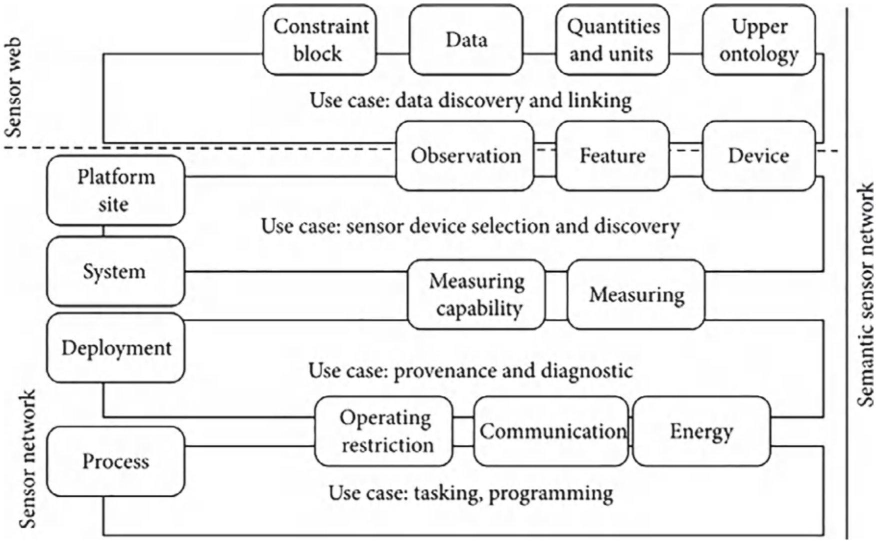


Fig. 11.4 Distinctions between the semantic network, sensor network, and SW

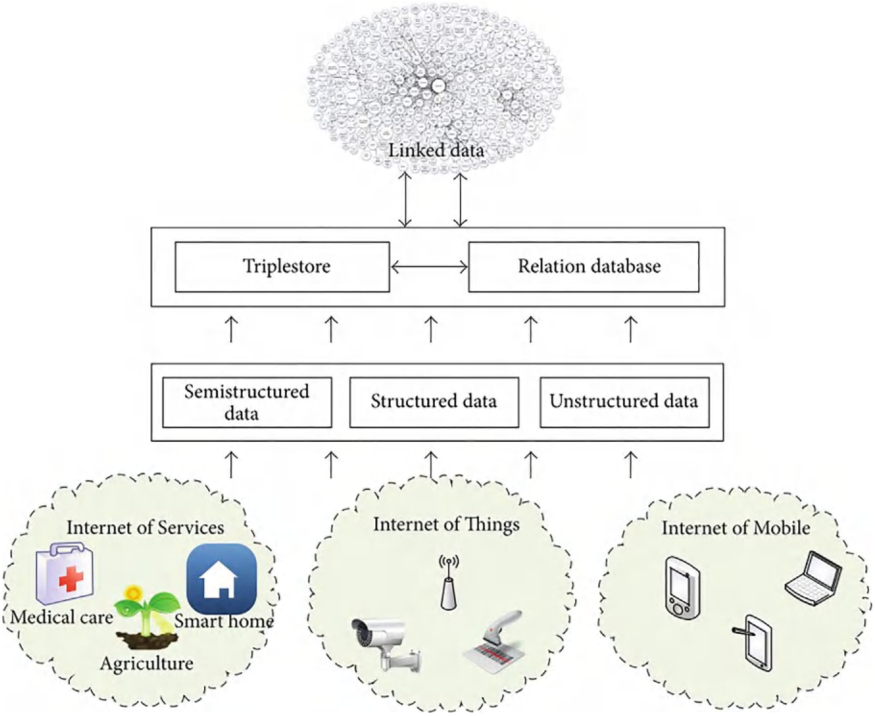


Fig. 11.5 Semantic vision data management using IoT

11.1.2.6 Evolution Among Technologies

Figure 11.6 (Wang et al. 2015) shows the evolution of sensors, semantic sensor networks (SSNs), sensor networks (SNs), sensor webs (SWs), and the Web of Things (WoT). These technologies come from many fields and are over time being included in the Internet of Things (IoT). The relationships and evolutionary processes of these interrelated technologies are depicted in Fig. 11.6. The figure's first row lists the pertinent criteria for each technology, and the second row describes how each one has evolved over time. The primary areas of concentration at each stage are outlined in the third row (Wang et al. 2015).

Different standardizing bodies push the standards, switching from information technology to communication technology. Regarding more precise data processing and efficient data use, the main issues at each level vary. A deeper comprehension of the data is shown in the third row of Fig. 11.6, which illustrates the shift in key concerns. Figure 11.7 (Wang et al. 2015) shows the development trajectory from the standpoint of data leverage, with the current stage at the top and the preliminary stage at the bottom. The first two columns of Fig. 11.6 (sensor devices and sensor networks) indicate this stage, where the primary focus is on encoding raw sensory data, or "data" in Fig. 11.7. The next step is to annotate raw sensory data with labels and tags to make

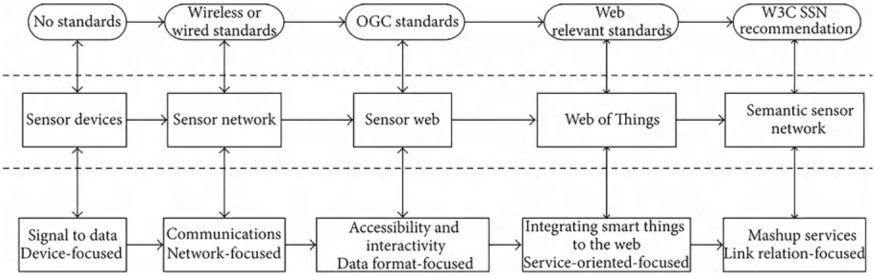
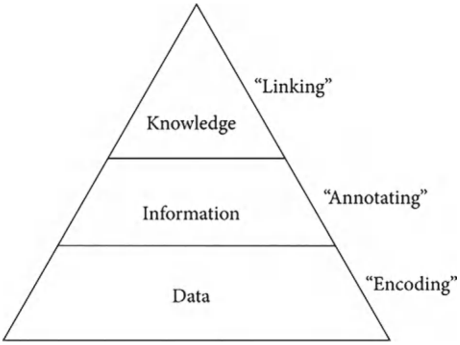

No standards
Wireless or wired standards
OGC standards
Web relevant standards
W3C SSN recommendation
Sensor devices
Sensor network
Sensor web
Web of Things
Semantic sensor network
Signal to data Device-focused
Communications Network-focused
Accessibility and interactivity Data format-focused
Integrating smart things to the web Service-oriented-focused
Mashup services Link relation-focused

Fig. 11.6 The development of sensors, SNs, SWs, the WoT, and SSNs

Fig. 11.7 Development pattern of fundamental issues at each level from a data perspective



it interactive and self-explanatory. This is illustrated by the third and fourth columns, which stand for the sensor web and the Web of Things, respectively. Figure 11.7 refers to this phase as "information". The fifth column (Semantic Sensor Network) refers to the current stage, which entails building deep and wide relationships with enormous volumes of heterogeneous data, or "knowledge" as it is called in Fig. 11.7.

The Internet of Things is based on substantial technological developments and the concept of pervasive networking; it is not science fiction or commercial hype. Since no one technology can meet all of the requirements for the Internet of Things (IoT), it is thought of as the convergence of at least six new technologies. The relationships between machine-to-machine (M2M) communication, SSNs, the WoT, the IoT, SNs, SW, and SWs—which are crucial components of the IoT—are depicted in Fig. 11.8 (Wang et al. 2015).

11.1.3 Evolution of IoT in Security

Before delving into the specifics of how IoT security has evolved, it is crucial to understand the changing priorities within the field from 2012 to the present. Figure 11.9, (Roman et al. 2018) which compiles and analyses articles from the database Scopus

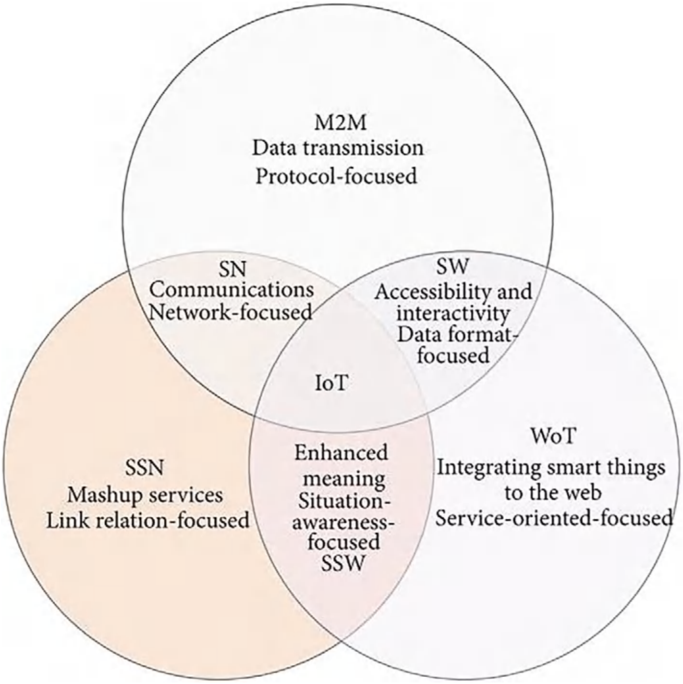


Fig. 11.8 M2M communication, SSNs, the Internet of Things, the WoT, SNs, SSW, and SW relationships

that specifically describes IoT security measures, provides a visual representation of this evolution. Through this data, we can discern which scientific community has always given security topics priority. For instance, the significance of fundamental domains like confidentiality, identity verification, reliability, safe correspondence, breach identification, and authorization have mainly been steady. However, some areas like physical unclonable functions (PUF) and security engineering have seen increased attention, while others, such as IoT forensics, remain underexplored (Roman et al. 2018).

11.1.3.1 Linear Evolution

Several IoT security areas have experienced clear, linear progress. Improving the effectiveness of current algorithms and protocols and adapting them to the Internet of Things setting have been the main objectives in these disciplines. For instance, much progress has been made in cryptographic primitives like elliptic curves, which were once thought to be too complicated for limited devices. Protocols like digital signatures (ECDSA) and key agreement (ECDH) can now be implemented at the

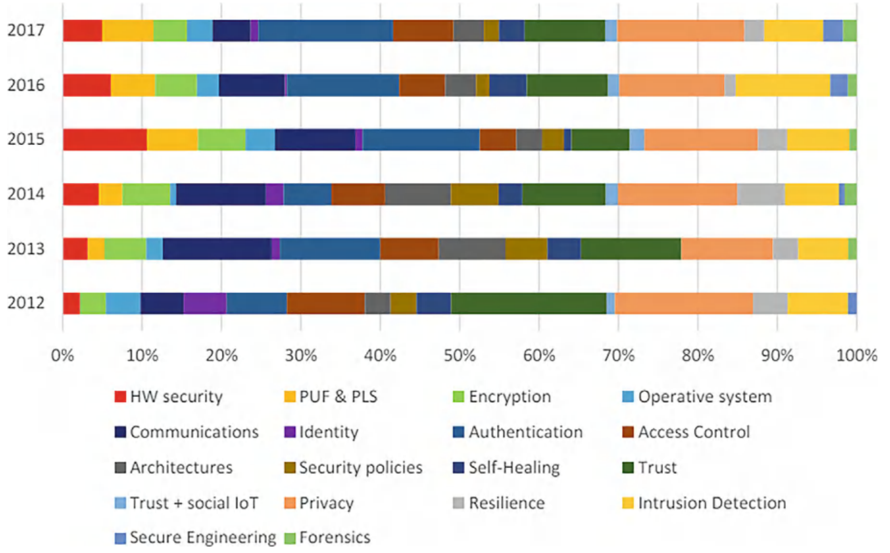


Fig. 11.9 Evolution of IoT

sensor level with less memory overhead and energy consumption, owing to developments in lightweight curves and efficient algorithms. Moreover, even more complex protocols, for instance, those based on bilinear pairings, their way into potent hardware. The first focus of work in the authorization and domains was on altering the existing user-to-service and device-to-service authentication methods. Improved cryptographic primitives have allowed for a shift from RBAC towards more sophisticated, token-based systems like ABAC as well (Roman et al. 2018). The mechanisms of trust and privacy have also evolved from early mechanisms in sensor networks to more advanced solutions tailored for the needs of IoT. Models and architectures of trustworthy cloud-enabled IoT infrastructures have been developed, paying attention to aspects such as data collection, usability, intrusion detection systems, and access control. Research in privacy has also probed techniques of data security such as homomorphic encryption, besides other concepts like location privacy, group anonymity, and plausible deniability (Roman et al. 2018).

11.1.3.2 Understudied Subjects

While there is huge progress in this domain, there are many underdeveloped areas of IoT security. One of the major problems in this regard is identity management. Working groups are currently underway to marry protocols such as OpenID into IoT frameworks; however, deeper aspects of identity, like core identity, association identity, and location identity, are not yet investigated for enabling IoT applications. Secure management and self-healing mechanisms that would provide situational awareness and resilience have only recently gathered momentum. One direction that research is taking includes predictive systems with machine learning and reactive systems based on functionality replication (Roman et al. 2018). Also related, yet mostly unexplored, fields are security and usability, forensics, and secure software engineering. Secure software engineering is taking off, emphasizing the simulation of risks and requirements related to security and privacy. Little research has gone into usability, with some surveys underlining its importance for security and privacy perception improvement. Forensics, key to analyzing attacks on IoT elements, has only recently received attention (Roman et al. 2018).

11.1.3.3 Bold Approaches

New strategies are explored for tackling the security issues of IoT. Most of the techniques in the IoT can use the physical layer for securing information. Physical-layer security and physical-unclonable functions are key methods used in the Internet of Things. Whereas PLS approaches secure the communication channels through the physical properties of wireless transmission, the PUFs provide hardware fingerprints that are easy to evaluate but difficult to predict. Prototypes of these technologies have already been constructed for device authentication and key distribution using both off-the-shelf hardware and hardware extensions (Roman et al. 2018). To minimize the uncertainty emanating from interactions, the concept of social IoT incorporates social networking into the IoT. From social links, trust factors, and context information, researchers have developed social IoT trust models and applied it in applications like trustworthy crowdsourcing and service composition (Roman et al. 2018). These new technologies underlying distributed ledgers, such as blockchains and smart contracts, make it possible to build secure firmware updates, decentralized access control management, monitoring digital and physical goods services, and so on. However, a number of important obstacles still lie on the way: scalability problems, high costs, and low throughput of transactions, possible attacks, and cost-effectiveness. Therefore, this work describes how smart contracts enable both decentralized and trusted operations of IoT networks. Such technologies can safely be used to implement services like secure firmware update or decentralized access control management and monitoring of both digitally and physically represented goods. Important challenges to be faced may include scalability, low transaction rate, high fees, potential attacks, and cost-effectiveness. Conclusion: In conclusion, an evaluation of the IoT security domain reveals that it is dynamic and changing: great development occurs in one area, whereas new difficulties arise in other areas. So only by further research and innovation, these obstacles can be defeated, and all the potential of the Internet of Things can be realized (Roman et al. 2018).

11.1.4 Benefits and Applications of IoT

11.1.4.1 The Importance of IoT

The Internet of Things is a new force that constantly invades all aspects of household life and business tenure, which is more and more central to the needs of people everywhere. Here are some benefits which stress the need for IoT:

1. **Customer Engagement**: Old-fashioned methods of analytics have the problem of introducing inaccuracies and are not engaging, so the customer has been passively watching. IoT has taken up a different approach to replace passive and lower engagement rates by introducing to audiences that functional objects are now capable of understanding how to interact with customers through engaging and active experiences (Tiwary et al. 2018).
2. **Optimization of Technology**: Data and Technologies that augment the customer's experience also help us make devices work in a better manner (Tiwary et al. 2018). IoT generates not just the most important functional and field data necessary for successful technological improvements but it also makes sure that the devices are working at their best.
3. **Reduction of Waste**: Rather than just IoT analytics, the IoT links gaps and their reflection is far from a skin-deep insight. Given that the IoT supplies the real-world data, it is a key factor that greatly helps in resource efficiency and thus waste reduction is ensued along with efficient operations (Tiwary et al. 2018).
4. **Advanced Data Collection**: Most of the traditional methods of data collection are limited to passive usage and use without sufficient data. IoT surpasses them in every way. These, in turn, the data will be transmitted to where the data is needed (Tiwary et al. 2018). This is a way to develop more detailed and specific knowledge about the environment, and consequently someone will be able to make an informed decision.

Thus, it is obvious that machine-to-machine communication (IoT) is so relevant—this is due to its assistance in the field of data collection which may lead to software optimization, less waste of resources, and drive customer engagement. These pros and cons exemplify how IoT enhances global communication and the efficiency of the global world.

11.1.4.2 IoT Applications

The Internet of Things (IoT) is built to provide services that make life better and facilitate the business world in several ways. The areas of sensor applications always come on the list of such use cases that involve a great number of sensors to process the data. For example, in the medical domain, the different applications such as bio-sensors, cloud, and fog computing, signal processing, and machine learning, are proposed and given examples. The term 'occupant safety' means exactly what it reads as it is the safety of the occupant of a vehicle, whether it is an autonomous or a non-autonomous vehicle. Additionally, IoT applications are prominently seen in transportation and logistics, healthcare, smart environments (like smart homes and offices), and personal and social domains (Chanal and Kakkasageri 2020).

1. **Medical Applications**: IoT applications in the medical industry are rapidly growing, ranging from remote monitoring systems to smart sensors and integrated medical equipment. IoT facilitates better patient-doctor contact and speeds up the scheduling process. Keeping security and privacy while managing the massive amount of data generated by IoT devices is still a challenge. Some well-known examples of IoT medical applications are the Open Artificial Pancreas System (OpenAPS), Continuous Glucose Monitoring (CGM) devices, linked inhalers, smart contact lenses, wearable apps for tracking depression, and devices made with Apple's Research Kit for conditions like arthritis (Chanal and Kakkasageri 2020). The primary challenges involve data security, protocol adherence, data aggregation, and the cost of IoT app development in healthcare.
2. **Military Applications**: In the military sector, IoT is used to create a comprehensive, real-time situational awareness and enhance decision-making. Modern military equipment, equipped with processing and communication capabilities, integrates seamlessly into the military data network, acting as sensors or actuators. Disaster relief, communications on the battlefield, mission-critical voice services, equipment tracking, intrusion detection, and threat analysis are some of the major military IoT uses. Ensuring security and reliability is crucial, as insecure IoT networks can be susceptible to data manipulation or disruption (Chanal and Kakkasageri 2020).
3. **Industrial Applications**: The term Industrial Internet of Things (IIoT) refers to the use of data aggregation and analysis from sensors by IoT applications in industry to enhance machine effectiveness and operational throughput. Predictive maintenance, motion control, automation, smart meters, factory control, and smart grids are a few examples of applications (Chanal and Kakkasageri 2020). Interoperability, security, effective data analysis and transfer, and the merging of information technology and operational technology are among the challenges.
4. **Automotive Applications**: IoT plays a significant role in managing daily traffic, reducing congestion, and preventing accidents through intelligent transportation systems. Sensors like GPS, accelerometers, RFIDs, and infrared sensors are used to manage traffic, monitor vehicle conditions, and enhance driver response times. Infotainment systems, integrated navigation, automobile diagnostics, parking management, and crash response are a few prominent automotive Internet of things applications. One of the main security concerns is the personal data that is being kept safe and unauthorized access prevention as the vehicle systems may be accessed without the user's intention (Chanal and Kakkasageri 2020).
5. **Environmental Applications**: Tracking animals, managing trash and water, keeping an eye on the weather, and protecting the environment are IoT applications for environmental monitoring. Smart environment sensors help keep track and value the data gathered for potential solutions such as air and water

quality, pollution, natural disaster prediction, and endangered species protection (Chanal and Kakkasageri 2020). These uses, in turn, leads to contribute toward a lifestyle of sustainability due to the better recognition and problem-solving of environmental issues.

6. **Agricultural Applications**: Through the use of a variety of agricultural drones, precision farming, soil analysis, smart greenhouses, livestock monitoring, and agricultural drones; farmers may experience a significant increase in productivity and, at the same time, waste reduction costs in their processes. While crop growth and quality are subject to climate indicator monitoring, there are also inquisition concerns like security and privacy. To add further intelligence to the agriculture sector, IoT devices are now being designed with security systems and monitoring capabilities (Chanal and Kakkasageri 2020).
7. **Retail Applications**: IoT is used by retailers for offering a thrilling customer experience versus online shopping and it is characterized by rich data and performance analytics. Implementation of the smart supply chain, customer behavioral analysis, and the use of smart store tech are the promising applications. The problems in this sphere are assuring hardware base, user data confidentiality, and only authorized user access.
8. **Consumer Applications**: IoT makes it possible to remotely control items via the Internet, creating new opportunities for both consumers and enterprises. Connected autos, automated houses, and building health monitoring are examples of consumer applications. Information security, machine interoperability, machine responses in unanticipated settings, and the sluggish adoption of new technology are some of the challenges. Solutions involve secure protocols for data transmission, real-time data analysis, and systems that offer modular, scalable, and secure IoT applications (Chanal and Kakkasageri 2020). In conclusion, IoT applications span across various domains, enhancing efficiency, security, and quality of life while presenting unique challenges that require innovative solutions (Fig. 11.10).

11.2 IoT Devices Trends and Anticipated Growth

11.2.1 Elements of IoT

An efficient Internet of Things system requires a number of essential parts. These consist of the following: (i) middleware components, such as databases for storage and data analysis tools; (ii) hardware components, like sensors and actuators; and (iii) visualization through a variety of apps (Tiwary et al. 2018). As shown in Fig. 11.11, (Tiwary et al. 2018) this section explores the fundamental components of IoT that are necessary to construct IoT.

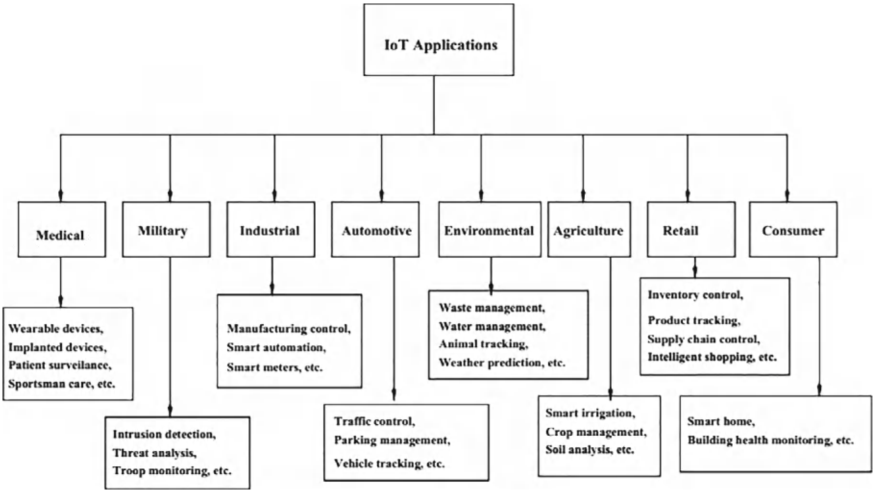


Fig. 11.10 Applications of IoT

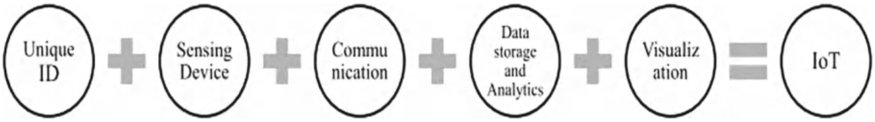


Fig. 11.11 Elements of IoT

1. **Unique Identification for Each Smart Device**: In order for the Internet of Things ecosystem's many smart gadgets to interact and be managed remotely, each one needs a distinct identity. IPv6 addressing greatly increases this capability, supporting a bigger range of unique addresses, whereas IPv4 addressing only offers a restricted number of unique addresses. Every device also receives an object ID, which is utilized to identify it within the communication network (Tiwary et al. 2018).
2. **Sensing Devices**: Every smart object has sensors in it that continuously collect information and that information is based on certain conditions such as the temperature, humidity, sound intensity, air pollution, or motion (Tiwary et al. 2018).
3. **Communication**: The smart devices are carrying out communication through a variety of technologies that connect RFID, Bluetooth, Wi-Fi, NFC, UWB, Z-wave, 3G, 4G, and LTE-A with databases (Tiwary et al. 2018). Just Drugs. Ranging the highest and lowest levels in drugs in organisms and the environment ideas have popped through my brain about xenobiotics in the environment, and now I am recording it. The material also usually scales up the topics covered and adds latest chapters to keep the book cur (Tiwary et al. 2018).

4. **Statistics and storage of data**: The Internet of Things (IoT)'s smart gadgets generate enormous amounts of data that need to be stored and analysed. Intelligent algorithms and sophisticated analytical tools are necessary to glean meaningful insights from this raw data (Tiwary et al. 2018). These tools need to work in different platforms. Middleware in the Internet of Things architecture comprises analytical and storage tools that need a centralized infrastructure to support them.
5. **Visualization**: In today's smart world, users can use smartphones, laptops, or IoT systems to interact. By installing some applications, users can access the libraries of centralized databases and get relevant information about their environment. This framework covers the key components needed in order to build high-quality IoT systems within the given range that will gather and process the information for efficient communication, data storage, and analysis, as well as visualization (Tiwary et al. 2018).

11.2.2 Impact of IoT on Various Industries

The possibility of the Internet of Things (IoT) bringing massive change in the economy is enormous. A considerable number of the big IT companies are the giants of the scale of the development of the IoT along the lines of Google, Samsung, and IBM. The positive aspect of the Internet of things is only one side of the coin, it also means new problems which should be treated (Kour et al. 2021). The Impact of IoT in different sectors is shown in Fig. 11.12 (Kour et al. 2021).

1. **Privacy Invasion**: IoT can lead to privacy breaches by exposing sensitive personal and financial information through data leaks (Kour et al. 2021).
2. **Unauthorized Access and Misuse**: The enormous amount of data generated by IoT devices endangers unauthorized access and misuse due to the sheer volume and variety of Big Data.
3. **Changes in Workforce and Organizational Structure**: The automation brought by IoT may lead to significant changes in staffing and organizational structures, which can have negative social implications (Kour et al. 2021). Despite these concerns, IoT offers substantial opportunities for revenue generation and can significantly enhance living standards.

11.3 Architecture and Architectural Elements of IoT

11.3.1 IoT Architecture Layers

There are various sorts of Internet of Things architectures, including three-layer, middleware-based, SOA-based, five-layer, cloud and fog-based, and social IoT (SIoT). The ensuing sections (Chanal and Kakkasageri 2020; Said and Masud 2013) will provide a detailed discussion of these structures.

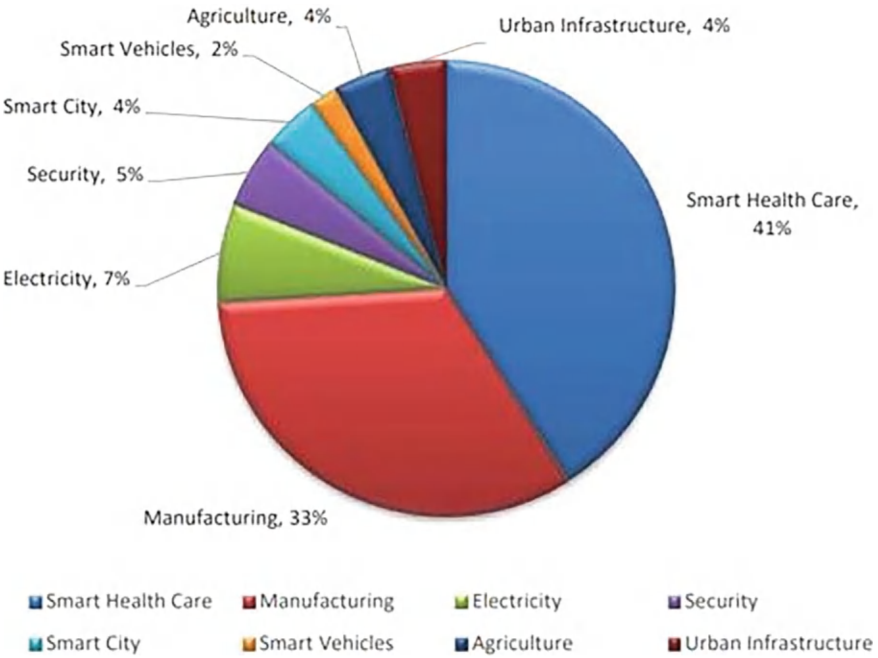


Fig. 11.12 Impact of IoT on various industries

11.3.1.1 Three Layer Architecture

When the Internet of Things first emerged, the three-layer paradigm (perception, network, and application layers) shown in Fig. 11.13 (Said and Masud 2013) was the most common architecture. The perception layer's job is to detect and gather information on each object in the Internet of Things system using RFID tags, sensors, cameras, and other devices. The network layer, which serves as the foundation of the Internet of Things, is in charge of transmitting the data collected by the perception layer. It consists of administration and information centers in addition to the hardware and software elements of the internet network. Finally, the application layer seeks to act as a mediator between industrial technologies and their application to human needs by bridging IoT technology with social demands (Said and Masud 2013).

11.3.1.2 Cloud and Fog Based Architecture

Cloud computing infrastructure, which offers platform, software, and infrastructure services, enables large-scale data storage. The next step forward in computing technologies is fog computing, where information processing and analysis are handled by sensors and gateways. Monitoring, pre-processing, storage, and security layers are situated in between the perception and application layers (Chanal and Kakkasageri

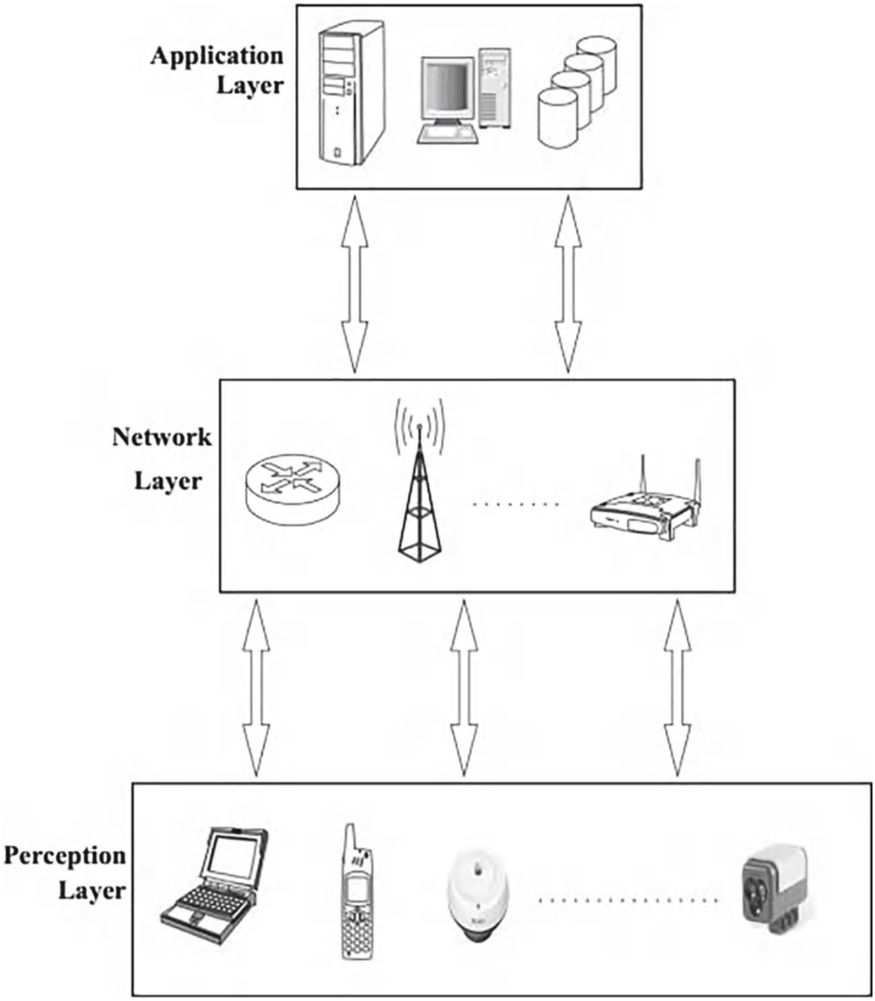


Fig. 11.13 Three layer structure

2020). Before data is sent to the cloud, pre-processing takes place at the network edge. Power, resources, responses, and services are managed by the monitoring layer. Data storage, distribution, and replication are managed by the temporary storage layer. Through a variety of encryption and decryption techniques, the security layer guarantees the privacy and confidentiality of data. Fog computing and edge computing have become more popular recently. Although edge computing improves physical devices with intelligent data pre-processing capabilities, fog computing, a term first used by Cisco, refers to smart gateways and smart sensors (Chanal and Kakkasageri 2020). Figure 11.14 (Said and Masud 2013) shows a general block diagram of cloud and fog-based architecture.

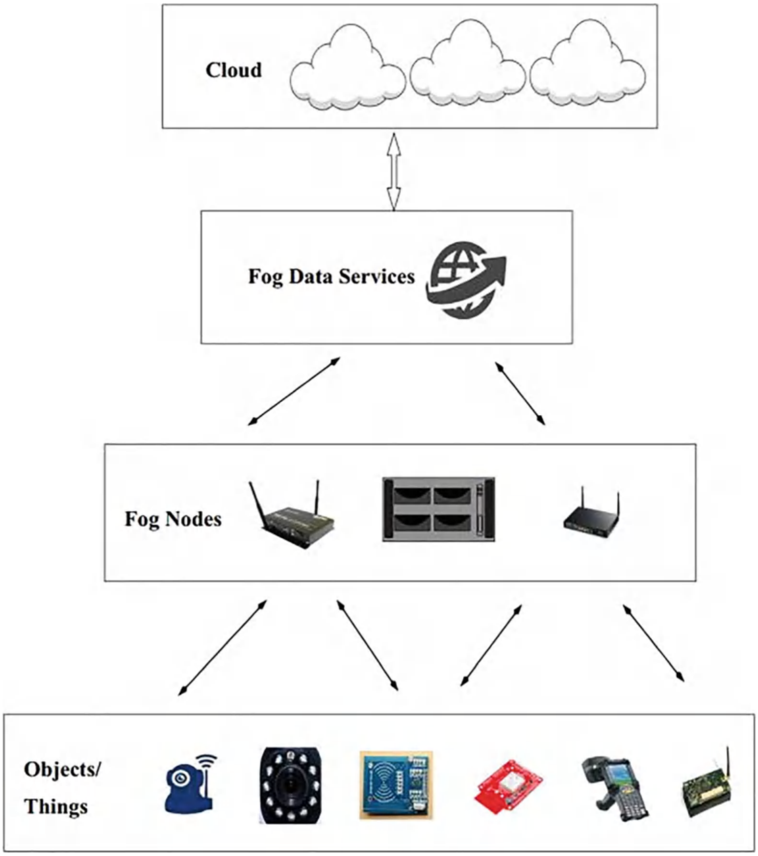


Fig. 11.14 Structure based on cloud and fog

11.4 Issues and Challenges in IoT

IoT is a groundbreaking technology, but it's not without its difficulties. Data transfer between networked devices is a complicated procedure that causes problems for Internet of Things developers (Khan and Singh 2022).

1. **Scalability Challenges**: Scalability is the capacity of a system to grow or shrink without compromising its functionality. The primary obstacle of handling a wide set of devices is that they have different cache, processing, bandwidth, and storage capacities (Khan and Singh 2022). On the other hand, one of the methods for the solution is the use of cloud-based IoT systems that can provide support to strong scalability. However, one of the most crucial issues is people who are always kept in different conditions and circumstances to gain access to all the services and resources without exceptions (Balaji et al. 2019). If indeed resources are to

be perpetually available, a well-functioning means of data transfer should prevail (Khan and Singh 2022).

2. **Introperability Issues**: Interoperability of the system implies the ability to interact and exchange information among the devices or nodes in the network. For the complete realization of the IoT, this is a must. Compatibility issues arose in Internet of Things systems due to the different settings and characteristics of a variety of technologies. Interoperability can be categorized into four levels: technical, semantic, syntactic, and organizational (Khan and Singh 2022). To address these issues, researchers have proposed several countermeasures, known as interoperability handling approaches, which include adapter/gateway-based methods and service-oriented architecture-based methods. Connecting diverse devices poses a significant challenge, as it challenges the current communication models that rely on a centralized server/client paradigm for network connectivity. This issue can be addressed by partially decentralizing IoT networks using fog computing models.
3. **Security and Privacy Concerns**: Building confidence in IoT devices requires a focus on security and privacy. IoT security issues and dangers are a big worry for developers because of all the risks, vulnerabilities, cyberattacks, and threats (Khan and Singh 2022). These problems are caused by unsecured software, firmware, and network interfaces; inadequate authorization and authentication; and inadequate transport layer encryption. IoT security falls into three primary categories:

   (a) **System Security**: This focuses on identifying various security challenges across the entire IoT system, developing distinct security frameworks, and providing appropriate security guidelines.
   (b) **Application Security**: This addresses security issues specific to application requirements.
   (c) **Network Security**: By doing this, the network that different IoT devices utilize to communicate with one another is protected.

   IoT systems are susceptible to a range of attacks that try to undermine the security of the device or network. Physical, encryption-based, DoS, firmware hijacking, ransomware, botnets, and man-in-the-middle assaults are some examples of these active and passive attacks (Khan and Singh 2022). The attacks fall into the following other categories:

   (1) **Protocol-Based Attacks**: These target communication channels and dispatch mechanisms by taking advantage of the internal architecture of Internet of Things components that are based on protocols. Among them are:
      (a) **Communication Protocol Attacks**: These include sniffer, flood, key, and pre-shared attacks, which take advantage of node transitions.
      (b) **Network Protocol Attacks**: These include wormhole, selective forward, and sniffer attacks that take advantage of vulnerabilities in the connection establishment process.

(2) **Data-Based Attacks**: These aim to verify the legitimacy of data memos and packets sent between nodes. Data disclosure, malicious node VM formation, hash collisions, and denial of service are a few examples.

These aim to verify the legitimacy of data memos and packets sent between nodes. Data disclosure, malicious node VM formation, hash collisions, and denial of service are a few examples.

(1) **Low-Level Attack**: An ineffective attempt to breach a network.
(2) **Medium-Level Attack**: Without compromising data integrity, an attacker listens in on the medium.
(3) **High-Level Attack**: An attack that affects or alters data integrity.
(4) **Extreme High-Level Attack**: An attacker breaches the network, carries out illegal activities, sends unsolicited messages, blocks it, or renders it inoperable (Khan and Singh 2022).

Every layer of the IoT architecture needs security procedures to thwart security risks and attacks in order to guarantee IoT network security. In order to ensure security and data protection in Internet of Things-based systems, a number of protocols are created and put into place at each communication channel layer. To offer security protections, cryptographic protocols like Datagram Transport Layer Security (DTLS) and Secure Socket Layer (SSL) are incorporated between the application and transport layers. Ensuring users feel safe when using IoT-based applications is another important concern: privacy. To enable communication between trusted parties, a network must allow secure authentication and warranty. The different privacy policies for every item in the Internet of Things is another problem. Therefore, prior to data transfer, privacy policies for every deployed object should be confirmed (Khan and Singh 2022)

4. **Ethical, Law and Regulatory Issues**: As IoT develops, it solves a lot of practical issues but also brings with it a lot of moral and legal quandaries, like privacy protection, data security, and trust and safety. Certain rules must be followed in order to uphold moral standards and avoid infractions. Additionally, many IoT users advocate for government regulations on data privacy, protection, and security due to growing concerns about the technology (Khan and Singh 2022).

11.5 Security Issues in IoT

The Internet of Things is made up of a tiered architecture with different technologies and functions used at each layer. Security threats are increasing along with the quick spread of IoT devices. In order to strengthen the security of IoT technologies, this section explores potential security risks across IoT layers, highlighting important characteristics including Confidentiality, Integrity, Availability, Authentication, Data Freshness, and Self-Organization (Aqeel 2020).

There are several layers in the Internet of Things architecture, and each one is susceptible to various security threats. There are many security criteria and challenges that must be met. Authentication and access control mechanisms are the main topics of recent Internet of Things research (Said and Masud 2013). To fulfill future IoT security concerns, it is imperative to embrace new networking protocols like IPv6 and 5G, though, given how quickly technology is advancing. The layers of any Internet of Things application are the middleware layer, application layer, network layer, and sensor layer. These layers are all made up of different technologies, each of which brings with it new problems and security risks. The different DDoS security risks in IoT applications across these layers are examined in this section (Hassija et al. 2019).

11.5.1 Security Issues at Sensing Layer

At the sensor level, handling IoT sensors and actuators is the most important function. The actuator is a device that serves the 'information' obtained by the sensors, one of the elements whose purpose is to sense a variety of 'surroundings' data. Typically, there are different types of sensors used to collect various data, such as temperature and humidity sensors, some types of smoke detectors, ultrasonic sensors, and the camera sensors. These sensors can be mechanical, electrical, electronic, or chemical. Diverse technology, a sensing layer like RFID, GPS, Wireless Sensor Networks (WSNs), and Remote Sensing Networks (RSNs), has been designed in different IoT applications (Hassija et al. 2019). The leading security threats at the sensing layer include and are not limited to:

1. **Node Capturing**: The basic concept of the IoT system is to connect many low-power nodes such as sensors or actuators to the network. However, these nodes are elements of the system that are easy to attack suddenly. These attackers may take over or replace a node with a malicious one, thus compromising the entire system.
2. **Malicious Code Injection Attack**: The means normally used by attackers to realize a Code Injection Attack is by injecting a malicious code into the memory of the nodes mostly a result of over-the-air software or firmware updates. The code explicitly defines the mode of operation of the nodes—whether the nodes do not behave in the intended way, or the nodes have an unexpected entry in the IoT system.
3. **False Data Injection Attack**: The attacker after the capturing of the node may push false data into the IoT system, resulting in wrong logical activities and even its malfunctioning. This method can be employed to carry out Distributed Denial of Service (DDoS) attacks as well.
4. **Side-Channel Attacks (SCA)**: Attacks of these types take advantage of microarchitecture and processors, those that emit electromagnetic waves, and power consumption to disclose information. Side-channel attacks can be energy consumption attacks, laser attacks, timing attacks, or electromagnetic attacks. Numerous alternatives are the latest technologies of the chips in performing

various protective measures which make them very efficient in avoiding these attacks while cryptographic operations are performed.

5. **Eavesdropping and Interference**: Devices are more prone to spying if they are put in open areas or other spaces. Eavesdroppers can capture the data during the transmission or authentication phases.
6. **Sleep Deprivation Attacks**: Adversaries may attempt to drain the battery of low-powered IoT edge devices, causing a denial of service. This can be achieved by running infinite loops on edge devices using malicious code or artificially increasing their power consumption.
7. **Booting Attacks**: Edge devices are vulnerable during the boot process when inbuilt security measures are not yet enabled. Attackers may exploit this vulnerability to attack nodes during their restart phase. Since edge devices often undergo sleep–wake cycles, securing the boot process is crucial.

11.5.2 Security Issues at Network Layer

The data from the sensing layer to the compute unit for processing is mainly transmitted by the network layer in the Internet of Things. There are some of the main network layer security concerns that surround it:

1. **Phishing Site Attack**: Phishing attacks are trying a lot of IoT devices all at once with little effort from the attacker, the aim is to attack some devices. Swindling web pages may also occur by means of phishing. After a person's account and password are compromised, all of the IoT environments they use are open to cyber-attacks. Tempting fake websites can be easily created and they can reach the web where IoT devices are operating.
2. **Access Attack**: More commonly known as Advanced Persistent Threat (APT), this attack sees a person that does not have permission to be on the IoT network expertly gaining access but remaining undetected for a long period of time. The primary goal is actually to steal important information rather than to make an immediate impact. IoT applications that transmit and receive crucial data permanently are very open to attacks like this.
3. **DDoS/DoS Attack**: As a result of DDoS/DoS attacks, a number of the badges of the target servers are possessed by the unusually large amount of unwanted appeals from the server, which debar the server and disturb the services for their legal users. A situation in which the target server is flooded by numerous sources is called a Distributed Denial of Service (DDoS) attack. These attacks are making the under or improperly protected network layers easier to be attacked, especially the network layer which is known as the most vulnerable to the attack. The DDoS attacks that are distributed among servers are made possible because most of the IoT devices do not possess the necessary strong configurations; poorly implemented systems are the ones that are particularly petrified. The exploit of the Mirai botnet that involves the weakness in devices is still active and it is

recognized as the main reason for the blockage of some servers by the continually sending of requests to weakly configured IoT devices.

4. **Data Transit Attacks**: IoT applications have to manage the immense data storage as well as that in between the devices. Thus the traffic is the main target for cyber-attacks. As data is transmitted from sensor to actuator to the cloud, in some cases it can be intercepted and thus security can be compromised. The technologies of connections are diversified and they bring about the risks that the IoT applications are subjected to.
5. **Routing Attacks**: Noises nodes created by attackers in the IoT network may execute the manipulation of the routing paths that the nodes currently opt for. Sinkhole attacks are a type of routing attack in which an attacker advertises the shortest route that is actually a fraud and therefore channels the traffic through that route. Wormhole attacks of that type are the most risky of all attacks and even of clear-sequence types when other attacks like sinkhole attacks are combined. An attacker might set up a wormhole centering on a compromised node, and subsequently directly to an internet device, thereby irreversibly degrading the built-in security measures in an IoT network.

11.5.3 Security Issues at Middleware Layer

Between the network layer and the application layer in the Internet of Things, the middleware layer acts as an abstraction layer. It has strong computational and storage capabilities as well. Brokers, permanent data storage, queuing systems, and machine learning, among others, also should be included in this layer. The middleware layer, besides the above-mentioned, also offers APIs to the application layer to satisfy the needs of the application layer. The middleware layer, despite playing an essential role in approving a reliable and robust IoT program, is, however, prone to several kinds of attacks as well as the IoT application will be in great risk of the attack that not only affects middleware but also the entire application. In fact, this level is fraught with issues related to cloud and database security as well. There are actually different types of middleware layer attacks, such as:

1. **Man-in-the-Middle Attack**: The MQTT protocol utilizes a publish-subscribe model for communication between clients and subscribers, through a broker, which acts as a proxy, that is, this is a middleman performing the communication tasks. This breaks the agencies doing the operation, which means the messages can be sent, but the places to which they are sent are unknown. If a hacker can operate the broker as the middleman, they are then able to ‘eavesdrop’ on all the communication overlooked by the clients alone.
2. **SQL Injection Attack**: Middleware is prone to SQL Injection (SQLi) attacks, whereby attackers implant malware-laden SQL command strings into ongoing processes. This actually facilitates the unauthorized access to private data of any user as well as flexing web pages in the database. The SQL Injection is the top

problem faced by Web security as per the Open Web Application Security Project (OWASP) that has brought out OWASP Top 10 2018 document.

3. **Signature Wrapping Attack**: Middle-tier web services are built using XML signatures. In a signature wrapping attack, the attacker is able to disrupt the signature algorithm by exploiting the weakness in the SOAP (Simple Object Access Protocol) which operates precariously, thus resulting in them being able to run operations or alter the intercepted video.
4. **Cloud Malware Injection**: In this case, the attacker will grow to full control by introducing the bad code or virtual machine in the cloud. The attacker is acting as a righteous supplier, therefore, they are creating a virtual machine instance, or, a malicious service module. This operation makes the attacker to access service requests the target service, and then captures and optionally modifies sensitive data.
5. **Flooding Attack in Cloud**: As is the case in a DoS attack, this attack brings congestions to cloud systems by sending numerous requests to a single service, thereby impacting the service quality (QoS). These assaults are capable of creating additional loads on the cloud servers and in result, the cloud systems are affected to a great extent.

11.5.4 Security Issues at Gateways

One of the most important layers in an IoT ecosystem is the gateway layer which bridges objects, devices, people, and cloud services. It handles operations such as data encryption and decryption and layer-by-layer protocol translation. IoT systems utilize technologies such as LoRaWAN, ZigBee, Z-Wave and TCP/IP, and gateways are intermediaries that connect them. The following are the main security issues with IoT gateways:

1. **Secure On-boarding**: The transmission of the encryption key by the link between the gateways and the devices is the very point where hackers can breach the system. They can easily snoop the signal and intercept the communication with a device or the cloud.
2. **Extra Interfaces**: To ensure being safe is to keep the targets minimum. Look at reducing the use of the more commonly used interfaces and protocols of the systems. Only the necessary functionality should be what is enabled to fight against unauthorized access.
3. **End-to-End Encryption**: Absolutely secure end-to-end encryption is the requirement for confidentiality.
4. **Firmware Updates**: Most of the IoT devices do not support direct handling of firmware updates besides the devices with advanced technology. These updates are managed by the gateway that is responsible for the process details and also ensures the safety of the system.

11.5.5 Security Issues at Application Layer

The application layer is the most frontline layer that connects the IoT and the end-users. The layer provides services for smart homes, smart meters, smart cities, smart grids, and more. It demonstrates various challenges of security concerning data leaks and privacy which is a sensitive topic depending on the type of the application. An application support or middleware layer that bridges the network layer and the application layer is commonly present as well and the main function is to support resource allocation and computation. At the application layer, the top security issues are:

1. **Data Theft**: Since the IoT (Internet of Things) involves transfer of the data from one location to another, it is of greater concern to the security breaches happening in the process. Thus, in an environment where applications hold the chance for the theft of data, users will likely be more reluctant to share any personal information. Such security measures as data encryption, data isolation, and strong authentication are required to protect against data theft.
2. **Access Control Attacks**: Access control, a concept of information security, ensures that only those users who have been given permission to access a particular resource are granted that permission. Loss of the correct handling of access control can provocate the attack which affects all IoT applications and is unable to protect the specific one.
3. **Service Interruption Attacks**: The attackers besides DDoS (Distributed Denial of Service attacks) that are also called the service interruption are incidents where the servers are intentionally overloaded by sending requests to them from different points on the internet or from computers after exploiting their vulnerabilities.
4. **Malicious Code Injection**: Violation of built-in security features and taking advantage of vulnerabilities, also known as exploits, hackers use malware to insert their own malicious code, for example, most popular cross-site scripting. Due to the injected code, the attacker might take over the account and disable the system completely.
5. **Sniffing Attacks**: When sensitive information is not encrypted over the network, attackers may intercept it by sniffing it, ultimately reaching unauthorized access to protected information.
6. **Reprogramming Attacks**: For an attacker to circumvent security measures and gain control of the network, the process of programming the gadgets without proper security Apps/devices can be a flaw that exploited programming processes occurs.

11.5.6 Consequences of Lagging in IoT Security

Despite the accelerated utilization of Internet of Things (IoT) devices in various sectors, notwithstanding the progress there are also significant problems of security. Deficient IoT security brings various problems to the people, companies, and society's infrastructure. Such aspects are well-investigated in this segment.

1. **Privacy**: There is a massive data breach possibility when this data is stored on devices that are not properly secured, thus opening an opportunity to unauthorized entities to use and abuse it, thereby causing a very severe exposition of privacy. These infringements can make people the targets of identity theft, unauthorized surveillance, and other privacy violation habits. When these devices are not secured properly, unauthorized entities can access and misuse this data, leading to severe privacy breaches. Such invasions can expose individuals to identity theft, unauthorized surveillance, and other privacy violations.
2. **Physical Harm**: Innovative systems like industrial controls, autonomous cars, and the medical sector are rapidly adapting to IoT devices. The weaknesses of the security system in these devices may lead to physical harm and deaths. For example, when a scammer attacks a medical device and modifies its operation, patients might be injured and life will be lost. In a way likewise, tampering with industrial equipment or car control systems can result in unsafe states and disasters.
3. **Operational Disruptions**: IoT tech is the backbone of many industrial processes and is one of the critical services required to function. Suppliers, governmental utilities, and manufacturers can be one of the victims if a cyber attack hits those systems. This kind of disturbance can bring a big loss of money, decreased production, and also sometimes putting the people in danger. Disasters that occur as a result of the attack on a smart grid might result in millions of people being affected by massive power outages.
4. **Botnet Attacks**: IOT devices that do not have security can be accessed and employed to make botnets, a network that is used to attack the cyber. Distributed denial-of-service attack is one such assault when the server is flooded with traffic making the services unavailable. The famous Mirai botnet incident, which employed infected Internet of Things devices, showed the possible scope and severity of such attacks by having a huge impact on major internet services around the world.
5. **National Security Risks**: Cyberespionage by states most of the time would target the Internet of Things (IoT) devices embedded in their vital infrastructure, which includes transportation networks, energy grids, and water supply systems. The damage of these systems could lead to the country's security being at risk. These types of vulnerabilities are capable of posing serious risks to public safety, temporary disruption of critical services, and economic crisis.
6. **Loss of Consumer Trust**: Badly set security and regular unexpected events involving these smart gadgets may focus the lack of trust that the users have with IoT technology. The outcome of the declining trust that users of IoT have is that

the development, the market's growth, and the realization of its potential positive effects are all hindered. Given that they are cautious about the possible sequels of inadequate security, consumers may be discouraged from including IoT devices in their daily routines or commercial activities.

7. **Legal and Regulatory Consequences**: Businesses that don't carefully deal with security issues in their IoT devices should be ready to face legal and regulatory consequences. Cybersecurity and data protection are now the key issues of regulations, and the failure to comply with them can bring about massive penalties, lawsuits, and reputational injury to a company. Firms have to comply with the new terms and conditions and security protocols, so regulatory authorities will not punish them and pay more attention to the security of IoT devices.

One of the main facets of attacks against IoT is the high level of complexity and continuously changing the most of the time. Now we speak about IoT devices of different types like smart home gadgets or industrial sensors; these devices commonly have built-in vulnerabilities such as weak default passwords, outdated firmware, or deficient encryption protocols. The attackers exploit them to gain unauthorized access, manipulate device functionalities or intercept sensitive data. The process includes IoT devices as central access points into larger networks which in turn cause distributed denial-of-service (DDoS) attacks to occur or allow cybercriminal undertakings (such as data theft or ransom demands) to be launched. Connectedness of IoT ecosystems makes the effect of these attacks exceed even faster and therefore become able to take over vast numbers of connected devices and networks. Moreover, nasty IoT applications, from consumer electronics to critical infrastructure, are the targets of variety of attacks which requires customized security measures. So long as IoT becomes more and more common across industries, acknowledgment of these dynamic attack vectors and arsenal building of strict security are key to reducing threat and guaranteeing conference in this ever connected digital space.

11.6 DDoS Attacks in IoT

11.6.1 DoS/DDoS Attacks

A DoS assault involves a hacker trying to consume computer network resources such as bandwidth or CPU time by spoofing the well-intentioned users of a certain network. When many means of attack are used which are already infected nodes, the entire situation is described as the Distributed Denial of Service (DDoS) attack. The most typical method of launching a denial-of-service (DoS) attack is by overwhelming the network with heavy traffic, the CPU, and the rest of the network's resources. A particular form of DoS attacks constitutes such types as SYN Floods, DNS Floods, Ping Floods, UDP Floods, and ICMP Broadcast attacks. A typical DDoS assault scenario is shown in Fig. 11.15 (Sonar et al. 2014). The attacker in this scenario uses a variety of high-performance processing units called handlers. These handlers send

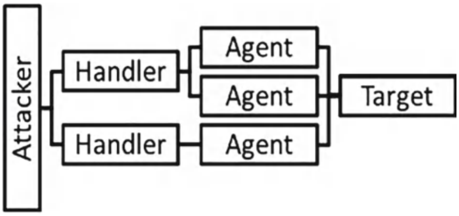


Fig. 11.15 Flow of DDoS Attacks

flood packets to the target host via various agents, using up network bandwidth and resource resources in the process (Sonar et al. 2014).

11.6.2 Types of DDoS Attacks

1. **UDP Flood**: A UDP flood occurs when a target host is flooded with different UDP packets on random ports (Sonar et al. 2014). It is sometimes referred to as a session-less networking protocol. This causes the host to keep checking for application ports that are open for listening, which results in ICMP Destination Unreachable packets being returned in response. This technique makes the target's resources inaccessible as a result.
2. **ICMP/PING Flood**: An ICMP flood is similar to a UDP flood in that it overloads the resources of a target host with ICMP Echo Request (ping) packets. These packets are transmitted quickly, without awaiting a response. When target servers try to reply with ICMP Echo Reply packets, this kind of attack can use up both incoming and outgoing bandwidth, severely slowing down the system (Sonar et al. 2014).
3. **SYN Flood**: An attack known as a SYN flood DDoS takes use of a flaw in the TCP connection process called the "three-way handshake." This technique involves the attacker sending out several SYN requests, but either they send the SYN requests from a spoof IP address, or they ignore the target host's SYN-ACK responses (Sonar et al. 2014). Consequently, the host system keeps waiting for acknowledgments, tying up resources until it is unable to establish any more connections, which finally results in a denial of service.
4. **Ping of Death**: Pings that are malicious or distorted are sent to a computer in a Ping of Death (POD) attack. The attacker makes the recipient reassemble an IP packet that is larger than the conventional maximum size of 65,535 bytes by changing the fragment content. A denial-of-service attack against valid packets may result from this overflow of memory buffers assigned to them (Sonar et al. 2014).
5. **Zero-Day DDoS**: A "zero-day" DDoS assault takes advantage of vulnerabilities that are unknown or undiscovered because no patches or remedies have been made available for them (Sonar et al. 2014). This kind of attack targets weaknesses that

the security community is not yet aware of, making defence difficult because there are no available fixes or solutions.

11.6.3 DDoS on Perception Layer

Significant security issues are DDoS assaults on the Perception Layer. Several sorts of attacks are as follows:

1. **Jamming**: Electromagnetic jamming disrupts communication between RFID tags and readers, preventing data transmission (Sonar et al. 2014).
2. **Kill Command Attack**: Attackers can disable RFID tags using a "kill command" that overrides the tag's protection settings. Although these settings are normally password-protected, brute-force attacks can be used to crack them because of the system's low memory and processing power. Attackers can now permanently disable tags thanks to this.
3. **De-synchronizing Attack**: A de-synchronization attack breaks synchronization, which prevents RFID tags and readers from authenticating each other. This attack disables the tag's authentication feature permanently, making it worthless.
4. **Pulse and Wide-Band Denial**: Blocking the entire RF spectrum makes it impossible for any user to access the impacted spectrum, making it the most straightforward method of jamming RFID traffic. While a generic RF generator can achieve this, it is more cost-effective to use 802.15.4 transceiver chips (Sonar et al. 2014).
5. **Denial That Is Message- and Node-Specific**: More sophisticated assaults aim to block specific signals rather than creating an overall disruption. To do this, the packet type and addressing information are retrieved from the first few bytes of the 802.15.4 MAC header. By analysing these bytes, an attacker can target and jam data transferred to specific addresses, nodes, or with specific characteristics.
6. **Attacks by Bootstrapping**: Nodes must safely connect to one another during a network's initial bootstrapping configuration. This process may involve simple methods like push-buttons on each node to initiate a special join mode. This method depends on the lack of attackers during the initial setting, which may not be safe enough for more critical applications, even though it might work for simple applications like remote controllers. Such a method is used for device bootstrapping in the ZigBee standard (Sonar et al. 2014).

11.6.4 DDoS on Network Layer

Sensor networks employ a wide range of communication technologies, such as Bluetooth, WiFi, Bluetooth, IrDA, ZigBee, RFID, NUWB, NFC, and Wireless Hart. These technologies satisfy many purposes and applications by enabling efficient data movement and communication within the network. The many types of network layer attacks are shown in Table 11.1 (Sonar et al. 2014).

Table 11.1 Types of attack on network layer

Form of the attack	An explanation
Attacks by flooding	This kind of attack entails flooding the victim's network with traffic in an attempt to prevent authenticated users from connecting. Such assaults can take the form of DNS floods, ICMP floods, UDP floods, and other techniques
Attacks using reflection based flooding	This kind of attack involves sending fictitious copies of requests to reflectors, which are parts of the routing system. Unaware of the attack, the reflectors reply to these requests, focusing their responses on the victim and wasting their resources in the process. The Smurf attack is one instance of this type of attack
Attacks using protocol exploitation flooding	In an attack of this kind, the attacker takes use of particular weaknesses in the victim's protocols' implementation or functionality to use an excessive quantity of the victim's resources. SYN floods, TCP SYNACK floods, and ACK PUSH floods are a few instances of these types of attacks
Attacks based on flooding amplification	This kind of attack aims to employ apps to produce messages or several replies, which are then used to increase traffic that is sent in the victim's direction. In these kinds of attacks, BOTNETs are frequently used for both amplification and reflection

A border gateway router in an IoT network exchanges data with sensors from the perception layer and transmits it to and from the upper application layer.

1. **Wi-Fi**:

Network layer denial of service attacks can happen on wired or wireless networks. A wireless network may be open to these types of assaults if it allows any client to connect. An attack that targets the victim's network infrastructure by sending massive amounts of data to a wireless network is known as a network layer denial of service (DoS) attack. The ICMP flood is one instance of this kind of attack. An ICMP flood attack overloads the target system with traffic by flooding it with a huge number of ICMP ECHO REQUEST packets. The attacker may direct all of its resources toward packet delivery while the target system must dedicate all of its resources to packet processing if it is able to spoof the source IP address (Sonar et al. 2014).

2. **ZigBee**:

Cheap, low-power wireless sensors are the aim of ZigBee, a standards-based wireless technology.

(a) **Hello Flooding**:

Attacker nodes use a high-power antenna to replay "hello" signals that they deliver to the one-hop network. By doing this, a one-hop network is produced without the need to crack encryption.

(b) **Homing Attack**:

In order to take down the entire network, the attacker examines traffic to find unique nodes (such cluster heads or key managers), then launches denial-of-service attacks on these nodes.

(c) **Black Hole Attack**:

The attacker becomes part of multiple routes and then drops all packets passing through (Sonar et al. 2014).

11.6.5 DDoS on Application Layer

The application layer is the topmost layer and contains the user interface as well as the core business logic of the entire program. There are two possible assault types against this layer:

1. **Reprogramming Attack**: This attack occurs when an attacker obtains access to the application's source code and changes it to the point that the program enters an infinite loop (Sonar et al. 2014), blocking access to network resources and making requests wait eternally for a response.
2. **Path-Based DoS**: An adversary can overwhelm distant sensor nodes in a path-based denial-of-service attack by flooding a multi-hop end-to-end communication path with replayed or inserted bogus packets (Sonar et al. 2014).

11.6.6 Anatomy and History of IoT Botnets

Attackers frequently target non-legacy IoT devices with DDoS assaults on servers and networks. These devices are particularly prone to abuse because they often have short battery lives and little processing power. An attacker must first put together a botnet, or a group of infected devices that act as malicious bots, in order carry out a successful DDoS assault (Kumari et al. 2023).

Particularly vulnerable to botnet infections are devices like routers, audio speakers linked to the internet, webcams, surveillance cameras, and a variety of home appliances, including TVs, refrigerators, smart heaters, and home security systems. These non-legacy IoT devices frequently have poor security features and flawed designs. Typically, attackers break the security of these devices via brute-force strategies to get over the cryptographic authentication measures protecting it. A common problem is that creators often set the same credentials among different gadgets. An attacker might be able to access many of the other vulnerable devices if they have the credentials for one device (Kumari et al. 2023). A majority of times, IoT device owners are not aware that their devices have been compromised. After taking control of these devices, attackers can launch a denial-of-service attack (DDoS) through sending an extensive number of packets from the compromised devices to a target. Attackers do not need to spoof the source addresses because these packets are supplied by reliable, innocent users, making it more difficult to recognize and prevent the attacks (Kumari et al. 2023). There are four essential phases in exploiting an IoT device as a botnet component and carrying out a DDoS attack.

11.6.6.1 How Botnets Are Created

IoT botnets are networks of breached smart Internet of Things devices used by hackers to carry out DDoS assaults. Usually, these botnets are deliberately constructed by people or groups collaborating with criminal organizations, who build malicious malware to infect Internet of Things devices (Kumari et al. 2023). Attackers can particularly target Internet of Things devices with this virus, which can dwell on any device that can execute code. After the virus is created, hackers utilize it to get into as many devices as they can, using the compromised devices to create a botnet depicted in Fig. 11.16 (Kumari et al. 2023).

There are four essential phases in exploiting an IoT device as a botnet component and carrying out a DDoS attack:

1. **Capture**: Locate and unlock Internet of Things devices (Kumari et al. 2023).
2. **Subvert**: Change the coding of the devices to carry out malicious activities.
3. **Activate**: Command the hacked device to start the attack.
4. **Initiate an attack**: Conduct a DDoS attack (Kumari et al. 2023).

IoT botnets are made up similarly to traditional botnets, with two main parts: the compromised systems that have been infected on their own, and the command-and-control (C&C) server used by cybercriminals to run the botnet shown in Fig. 11.17 (Kumari et al. 2023) Cybercriminals use these hacked computers' power to carry out dangerous processes, including DDoS attacks. While attackers utilize huge botnets built out of weak IoT devices, these attacks are more efficient now. Addressing the

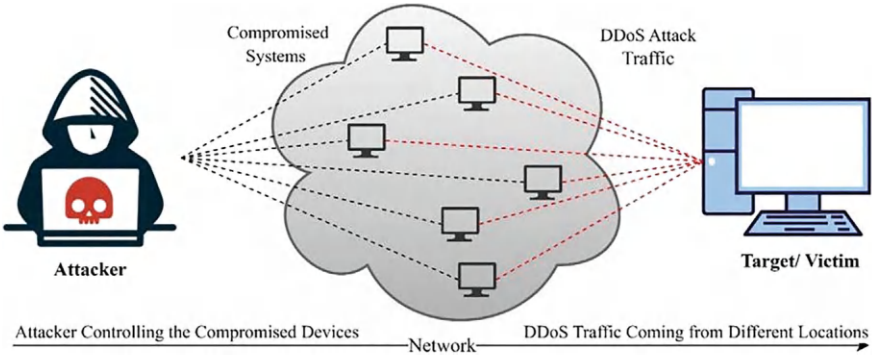

Compromised Systems
DDoS Attack Traffic
Attacker
Target/ Victim
Attacker Controlling the Compromised Devices
Network
DDoS Traffic Coming from Different Locations

Fig. 11.16 The way DDoS attacks are executed

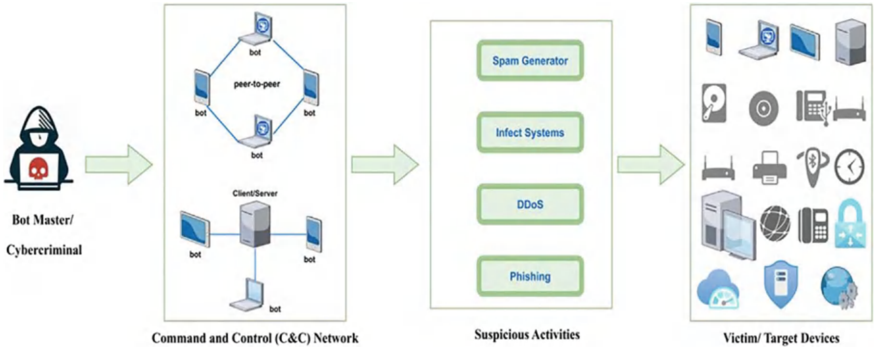


Fig. 11.17 Botnet command and control architecture

vulnerability of linked devices, experts have long given warning. Malware for IoT botnets has been created to take advantage of security holes and known vulnerabilities. In order to charge for the duration and intensity of the assault, some businesses even rent out botnets (Kumari et al. 2023). A variety of notable botnets have surfaced in the last years, including:

1. **Linux.Hydra**: Originating in 2008, Linux.Hydra was open-source software targeting routing devices with MIPS architecture. It laid the foundation for future IoT malware by using dictionary attacks or exploiting authentication weaknesses in D-Link switches to perform SYN flood attacks (Kumari et al. 2023).
2. **Psyb0t**: Similar to Linux.Hydra, Psyb0t appeared in 2009, capable of executing UDP and ICMP flood attacks. Though not directly comparable, Psyb0t is considered a successor to Linux.Hydra due to their similarities (Kumari et al. 2023).
3. **Chuck Norris**: Emerging in 2010 after the demise of Psyb0t (Kumari et al. 2023), Chuck Norris shared many similarities with Psyb0t, including the ability to carry out ACK flood attacks.
4. **Tsunami/Kaiten**: Combining features of Kaiten-Tsunami DDoS tools and Chuck Norris, this malware could perform SYN flood attacks and more complex attacks like HTTP Layer 7 Flood and TCP XMAS attacks (Kumari et al. 2023).
5. **BashLite**: Targeting Linux-based devices like cameras and DVRs, BashLite could launch UDP, TCP flooding, and HTTP attacks (Kumari et al. 2023). It exploited a Metasploit module for remote code execution, making it unique in its approach.
6. **Mirai**: Since its inception in 2016, Mirai has targeted security cameras, home routers, and other household IoT devices using a list of 64 common usernames and passwords (Kumari et al. 2023). It performs HTTP flood attacks and various network-level attacks, excluding specific IP ranges like those of General Electric and the U.S. Defense Department.

7. **Remaiten**: First observed in 2016, Remaiten combined Tsunami's DDoS features with BASHLITE's enhanced scanning capabilities. It could download executable files to spread across Linux-based devices, adding new bots to the botnet (Kumari et al. 2023).
8. **3ve**: This botnet, known for committing ad fraud, used a complex system to generate billions of fraudulent ad bid requests (Kumari et al. 2023). It was dismantled in 2018 by White Ops, Google, and law enforcement agencies.
9. **Wirex**: Discovered in 2017, Wirex included thousands of Android devices running malware disguised as legitimate apps (Kumari et al. 2023). Companies like Google removed the malware from the Play Store.
10. **Reaper**: Also known as IoT Troop, Reaper evolved from Mirai by exploiting vulnerabilities in D-Link, Netgear, and Linksys routers, among others, to gain control over devices (Kumari et al. 2023).
11. **Torii**: Discovered through Tor exit nodes, Torii targets devices with weak authentication, similar to Mirai, but is more sophisticated in its ability to deliver payloads to other devices (Kumari et al. 2023).
12. **Meris Botnet**: A recent DDoS botnet variant with approximately 30,000 compromised devices, Meris uses HTTP Pipelining and crashes servers to execute large-scale attacks (Kumari et al. 2023).

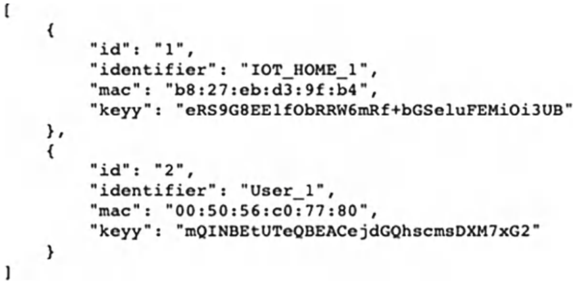
The number of Linux-based malware is increasing significantly each year, indicating a growing threat landscape for IoT devices.

11.7 Possible Solutions to Security Issues

11.7.1 Encryption Techniques for IoT Data

Public Key Server: A public key server is a server that distributes and keeps track of the public keys of different entities, giving everyone access to a shared database that can be used to encrypt communications to the appropriate entities. Using Apache Web Server, we have created our own Public Key Server in our model, which distributes the public key in accordance with the specifications supplied. The Public Key Server's answer is displayed in Fig. 11.18 (Bhandari and Kirubanand 2019). Upon first boot-up, an Internet of Things device establishes a secure connection with the Public Key Server via an HTTPS connection, supplying its Public Key and MAC Address as parameters to enable network communication. Similarly, a secure request is sent over HTTPS to the Public Key Server by the user when they launch their app for the first time in order to register the User/User's Device, along with the Public Key and MAC Address of the user's device (Bhandari and Kirubanand 2019).

Elliptic curve cryptography (ECC): Public keys produced by ECC (Elliptic Curve Cryptography) are managed by the Public Key Server. The asymmetric encryption method known as ECC is based on the algebraic structure of elliptic curves over



```
[
    {
        "id": "1",
        "identifier": "IOT_HOME_1",
        "mac": "b8:27:eb:d3:9f:b4",
        "keyy": "eRS9G8EE1fObRRW6mRf+bGSeluFEMiOi3UB"
    },
    {
        "id": "2",
        "identifier": "User_1",
        "mac": "00:50:56:c0:77:80",
        "keyy": "mQINBEtUTeQBEACejdGQhscmsDXM7xG2"
    }
]
```

Fig. 11.18 Querying to a public key server

finite fields. ECC uses the degree of points on the curve to produce keys rather than the product of two prime numbers, as is the case with traditional key generation methods. ECC is also used to generate private and public key pairs in blockchain implementations (Bhandari and Kirubanand 2019).

Advanced Encryption Standard (AES)

The Advanced Encryption Standard (AES) is a symmetric key block cipher algorithm that is well-known for its security and efficacy in cryptographic operations. AES provides key sizes of 128, 192, and 256 bits and operates with three fixed block sizes of 128 bits, as shown in Fig. 11.19 (Bhandari and Kirubanand 2019). The maximum block size is 256 bits, although the key size can vary. Unlike the Feistel network employed in the Data Encryption Standard (DES), AES uses a substitution-permutation network (SPN) (Bhandari and Kirubanand 2019).

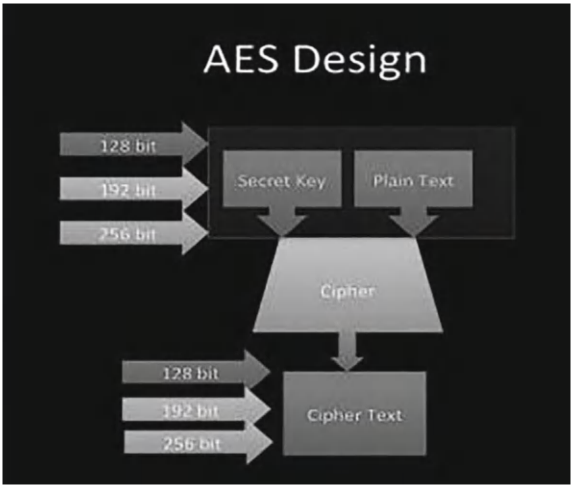


Fig. 11.19 AES design

11.7.2 *Intrusion Detection in Internet of Things*

It is not easy to designing a dedicated Protocol for IoT security, because of its fragmented nature, heterogeneous nature and lack of interoperability. To mitigate IoT security, various solutions have been proposed focusing on data confidentiality, authentication and access control and user and device privacy protection. Nevertheless, even with the application of these solutions IoT networks are not free from attacks. The deployment of Intrusion Detection Systems (IDS) in particular can improve IoT security against diverse threats. However the effectiveness of conventional IDS structures in IoT environment is limited since they cannot adapt operationally to the highly complex and heterogeneous network environment of Internet of Things. This highlights the needs for design IDS that are appropriate specifically for IoT environment due to its resource-constrained devices, unique network architecture as well as different protocol stacks and standard (Ahmed et al. 2019).

11.7.2.1 Detection Methods

The identification of intrusion detection methods is conducted in several ways; these methods can be applied to the respective groups of intrusion detection systems namely specification-based, anomaly-based, hybrid, and signature-based. This section aims at demonstrating how such techniques get implemented in creating Intrusion Detection Systems (IDS) used for Internet of Things contexts (Zarpelão et al. 2017).

Signature-based approaches: Signature-based intrusion detection systems (IDSs) derive their accuracy by comparing the system or network behavior with the predefined attack signatures that are stored in the IDS databases. This technology is becoming more deeply integrated into everyday life, with security cameras monitoring everything and the location of phones being constantly tracked. When a signature in the trace is matched by the IDS with an event, an alarm will be activated. Devices of this type are very basic but still operate with high accuracy, precisely identifying well-known threats. However, since there are no matched signatures to identify these bad guys, they can't find any new attacks that were not detected before or the ones that were changed (Zarpelão et al. 2017).

Anomaly-based approaches: Intrusion detection systems (IDSs), based on signatures, are able to raise an alarm that is termed as the anomaly in the system functioning. There are several reports about the type of attacks in the intrusion detection system which are not confirmed by the simulated IDS that makes them effective. This approach is especially effective in finding novel threats that cover resource management issues. Incorrect alerts could be high as almost any tampering activity is dealt with as a threat. There may be difficulties in creating a full characterization of the normal behavior profile, hence the study must typically use techniques such as machine learning and statistics. Among other things, these are known for their resource-hungry nature for low-capacity IoT devices. Therefore, it is important to examine this particular confinement in anomaly-based method sphere for IoT networks (Zarpelão et al. 2017).

Specification-based approaches: The foundation of specification-based intrusion detection systems (IDSs) consists of specified rules and thresholds, which clearly lays down the expected behavior of the network (its nodes, protocols, and routing tables). They are able to diagnose and alert the system administrator whenever there is a change from the predetermined parameters thus preventing intrusions. Although anomaly-based intrusion detection systems try to catch abnormal behavior, specification-based ones require a qualified person to make up a rule. It has been noted that the standardized manual setting has many findings in common with anomaly-based detection frequently resulting in lower false positive rates. Specification-based intrusion detection systems, in addition, do not require any training time and are ready to be up and running immediately after the installation. In addition, the procedure of heading through a laborious and error-prone process, whereas manual specifications might not be converted well to various situations (Zarpelão et al. 2017).

Hybrid approaches: By combining aspects of specification-based, anomaly-based, and signature-based detection, hybrid techniques minimize the shortcomings of each technique while maximizing the benefits of each (Zarpelão et al. 2017).

11.7.3 Blockchain for IoT Security

Blockchain technology can improve IoT frameworks in a number of ways and have a number of potential advantages. Here is a summary of these benefits:

- **Strengthened Security**: The use of Blockchain ensures the data produced by IoT is safe and can be done by saving it as encrypted and cryptographically signed transactions. Furthermore, the blockchain-based automatic software updates are dealing with security vulnerabilities, which raise the whole system's resiliency from attacks.
- **Enhanced Interoperability**: Blockchain technology offers new methods of processing IoT generated data mostly done on distributed ledgers by storing and handling the data through a centralized technology. IoT system interoperability can be improved by letting the blockchain technology assist in the process. Decentralized blockchain networks assure the safe recording, mining, resizing, processing, and transformation of various IoT data sets (Ekanayake 2022).
- **Independent Partnerships**: Blockchain technology now allows the IoT devices to connect automatically. This is not true of the way technology has evolved from large installations to small sensors and the necessity of traditional middlemen. Automated contingent liabilities provided by Decentralized Autonomous Corporations can be advanced without central oversight.

- **Reliability**: Data in blockchain-based systems is decentralized and time-immutable, making it easier for system staff to verify and guarantee the data integrity. Blockchain, additionally, is a means of achieving accountability and traceability of sensor data.
- **Safe Code Deployment**: The blockchain's motive storage feature ensures the secure use of transferring technology issue. These advancements assist in the improvement of the reliability of the data, the automation to the subsequent, the security, the interconnection, and the secure deployment of code by incorporating IoT frameworks with blockchain (Ekanayake 2022).

The vulnerabilities within the system show an enormous increase on the number of physical devices that are included in the internet, which leads to increased security problems. One example of a security threat in IoT networks is endpoints which are widely assumed, however, these devices are particularly under attack from a wide range of hacking methods such as Man-in-the-Middle (MITM) attacks or Blackhole, eavesdropping, Distributed Denial-of-Service (DDoS) attacks. Botnet shows a big danger from a number of small devices or nodes that have malicious intentions with the IoT infrastructure as a target. Moreover, the consolidation in proprietary solutions not only results in risks of accessibility but also results in privacy and validation deficits (Ekanayake 2022). IoT security is now largely reliant on the centralized security protocols that integrate the services of the third party. The security of the internet of things can be developed by blockchain technology. The blockchain-known Internet of things (IoT) system tells of "Jamming" encryption that is used for prevention of the hurdles thus the identity of each transaction is masked to the public. Blockchain-based IoT systems that can be tampered through a unique code provided to each transaction are another way. Moreover, the blockchain, in an open source protocol, the consensus method is achieved by the imposition of a fee on each transaction. Thus, the blockchain technology becomes a need for IoT security measures. Thus, employing blockchains in the security framework can augment the level of IoT security. The use of blockchains to run security rules and maintain a visible record of IoT transactions, while at the same time reducing dependence on third-party security providers, is the best thing IoT companies can do (Ekanayake 2022).

11.7.4 AI in IoT

AI for Device Authentication: Conventional techniques for device authentication, such as password and digital certificate authentication, are still in use today but have their own set of drawbacks. For user authentication, digital certificates depend on a reliable Certification Authority (CA), whereas password authentication requires users to save their credentials and compare them when logging in. However, devices can be counterfeited due to static identification information, and passwords can be intercepted during transmission, providing privacy problems. (Wu et al. 2020). Manual set information such as white lists or black lists is often employed for device authentication to screen and intercept connected devices. White lists contain authorized devices, offering better efficiency and security compared to black lists, which list potentially dangerous devices. Machine learning (ML) methods, such as combining ensemble learning with white lists, have proved high accuracy in device classification and authentication (Wu et al. 2020). Biometric identification based on human body characteristics such as fingerprints or voice patterns is advantageous, but at the same time, it raises serious questions on the issue of data privacy of the users. Cancellable Biometric Systems (CBS) are the systems that biometric data can be canceled by abstracting it to changed form with a secret key, whereas it is still usable to authenticate (Wu et al. 2020). Human handwritings and gaits are examples of human behavior traits, which are used to recognize personal identity, especially in indoor spaces equipped with smart devices. Deep learning algorithms are effective in recognizing actions of the users, abstracting information from Wi-Fi signals. (Wu et al. 2020). Information about the static nature of a device that includes RF fingerprints complements hardware differences and is used as a basis for authentication. ML algorithms, such as random forest, boost the accuracy of device identification by using RF fingerprints. Dynamic behavior data, like the IP addresses, and network access information combined with static device data, forms dynamic device information that is used in network security solutions. ML methods such as decision trees and gradient boosting improve authentication accuracy based on this dynamic fingerprint. In conclusion, these different signatures, some of which are biometric based, combined with ML techniques, could serve as security measures for IoT in a way that would address the privacy concerns and still make secure identifications (Wu et al. 2020).

AI For Dos/DDoS Attack Detection and Defence: In the IoT field, identifying and blocking anomalous traffic is especially important as one of the main DoS/DDoS attacks defense. The conventional defenses, such as firewalls, load balancing, and anti-DDoS devices, are widely used but still, have a great number of IoT devices. For this, PIR technology certainly has a key role to play in this because traffic filtering processes that are efficient and accurate are highly preferred. (Wu et al. 2020). All Ill-in-all machine learning opens up many ways to help IoT environments in the prevention and remedy of DDoS attacks, enabling them to stand up to the cyber threat and use their resources in the best way. It is important to note that these techniques are scenario-specific and can be addressed by machine learning techniques:

- **Software-Defined Network (SDN)**: SDN provides centralized control and flexing by eliminating the data planes and the network control. Yet because of its openness, DDoS attacks may be a possibility. SVM is used.
- **Wireless Sensor Network (WSN)**: IoT is incomplete without WSNs which are exact but can fall under the umbrella of DoS attacks.
- **Consumer IoT**: The idea of wearables and smart devices being turned into DDoS zombies should not be easily dismissed.
- **Smart City and Social IoT**: The demand for smart cities equipped with the Internet of Things is facing growing challenges with a DoS attack or disruptions caused by different vicious intruders (Wu et al. 2020).

AI for Intrusion Detection: Back in the day, spreading synthetic intelligence in the shadow of conventional methods of using misuse discovery and image detection to chase wrongdoings generally have influenced the way the whole personal security industry operates. The art of misuse detection is certainly useful. However, its smooth operation is mostly associated with the identification of incidents deeply rooted in established attack patterns compared to the situation when the attack is not an expected one, for instance, zero-day incursion attacks. In contrast, a threat-detection system or an anomaly detector is a framework which is supposed to detect new assaults hence, to make it more accurate continuously training samples is necessary. It is also essential to reduce processing time by performing mathematically advantageous calculations, thus ensuring that the detection system is effective. These developmental stages are setting the stage for the deployment of adaptive security solutions that are collaborative with new threats to be realized; hence, the use of AI-like mechanisms to enhance them is a necessity. Summing up, Artificial intelligence illumination in intrusion detection has been proven by the integrated detection, efficient feature selection, and the distributed frameworks which manage devices in IoT applications (Table 11.2) (Wu et al. 2020).

11.8 Limitations, Hurdles, and Future Scope of IoT Security

11.8.1 Limitations

1. **Resource Constraints**: The implementation of powerful security tools, for instance, the usage of very complex encryption techniques, can hardly be realized on IoT devices that usually come with limitations on memory, computation, and battery life.
2. **Heterogeneity**: There is a great diversity in the technologies and communication protocols involved in IoT, which is the reason why an effective global security solution is difficult to establish. The problem arises from the fact that differentiated devices have different needs and capabilities.
3. **Scalability**: The fact that an increased number of devices are being connected to the network results in escalating challenges for keeping the whole network secure in a short time. The only way to have a million devices connected without the network performing poorly is to use security frameworks that can scale.
4. **Standardization**: It is a variety of approaches to security without a generally acknowledged security framework for IoT, from which organizations can pick and choose security policies and procedures.

Table 11.2 Comparison of AI and Blockchain in IOT

Aspect	AI in IoT	Blockchain in IoT
Pros	Strengthens analytical and decision-making skills	Offers a transaction ledger that is safe from tampering
	Makes anomaly detection and predictive maintenance possible	Decentralizes data administration to minimize failure spots
	Makes automation and intelligent operations easier	Improves IoT network traceability and transparency
	Enhances user satisfaction with tailored services	Through immutable records, fosters greater confidence across IoT devices
Cons	Needs a lot of processing power and resources	The consensus procedure may cause latency and scalability problems
	Because of the substantial data collection, raises privacy issues	High energy and computing costs related to blockchain operations
	Requires complex models and algorithms for precise analysis	The difficulty of integrating blockchain technology with current IoT systems
	The necessity of high-quality data for efficient operation	Transaction speeds and throughput limitations may impede real-time applications
Future scope	Improved artificial intelligence algorithms for low-resource IoT devices	Utilizing low weight blockchain technologies to enhance scalability and energy efficiency
	Deeper edge computing and AI integration for real time analytics	The growth of blockchain based IoT applications enabling decentralized IoT marketplaces and safe data exchange
	Improved IoT cybersecurity solutions powered by AI	Creation of interoperability standards to enable the blockchain's smooth integration with the Internet of Things

11.8.2 Hurdles

1. **Complexity of Deployment**: The modern systems of security implementation in the Internet of Things are not easy to do and are also costly and of course very important to the programming of the devices, and they can make the producers very discouraged to do that because of that.
2. **User Awareness**: IoT devices are primarily sold to users with devices of limited capabilities that are more dangerous than others. Increasing public awareness is essential for increased security. Insufficient knowledge about IoT security best practices among makers and users leads to inadequately secured devices. Increasing awareness is essential to enhancing security.
3. **Regulatory Compliance**: For IoT developers and suppliers, the situation is such that they cannot avoid the change rules and be assured that they can comply with the laws in all countries.

4. **Quick Technological Advancements**: Due to the reason that IoT technology advances at a better pace than the security issues which are related to it, it happens that new devices are often neglected.

11.8.3 Future Scope

1. **Developments in Cryptographic Techniques**: In this way, the establishment of light cryptographic methods is very important in the use of hardware of low-resource capacity, but it is well known that it should be used since it would be less harmful, in the low power or battery of the machine.
2. **Edge and Fog Computing Security**: The application of security mechanisms in the edge and fog layers will make the information closer to their source be encrypted and thus will result in less sensitivity and reaction time that is faster and better real-time protection.
3. **AI-Driven Security Solutions**: Adaptive and intelligent systems with the help of AI and machine learning can detect and solve threats in real time.
4. **Standardization and Interoperability**: For the establishment of global security standards to be seamless and the widespread use of IoT technology the world over to be easy for people and businesses, the setting up of international IoT safety rules is of utmost importance.

11.9 Conclusion

The ability to collect, analyze, and optimize data from a wide array of digital devices which is generating the Internet of Things (IoT) to change in the majority of fields. Distributed Denial of Service (DDoS) attacks are one big security hole root by the spreading of IoT gadgets that can be dire to networks. The programs, infrastructure, and composition of IoT have all been reviewed at length in this study. The matter of the concern was IT security as a specific DDoS attack called botnet was outlined, and how attacks are covered, as well as how IoT devices can be used to build botnets were discussed in the chapter, which is the theme of this case study. In addition, the study encompassed new ways of reacting to such issues, among which, blockchain technologies and artificial intelligence (AI) would be included. In this respect, IoT system security and resilience will be increased by blockchain which has a decentralized structure and is maintained by AI for knowing the predictions. The comparison of the most recent DDoS preventive techniques is a helpful tool for gaining insight into the problem of DDoS, and it gives IoT-related security problems a chance to be well tackled. It is imperative that proactive actions are taken to prevent these security challenges as IoT is still evolving with predictions that 30 billion internet connected devices could be there by 2020. The greatest areas of the study are enhanced security, fast discovery of strange things, and the safety of private data. The problems reach the healthy world and sustainability can be realized with the help of IOT technologies are some of our best prospects if we can solve these problems.

References

Ahmed HI, Nasr AA, Abdel-Mageid S, Aslan HK (2019) A survey of IoT security threats and defenses. Int J Adv Comput Res 9(45):325–350. https://doi.org/10.19101/IJACR.2019.940088

Aqeel M (2020) Internet of Things: systematic literature review of security and future research. Dissertation. https://urn.kb.se/resolve?urn=urn%3Anbn%3Ase%3Auu%3Adiva-420118

Balaji S, Nathani K, Santhakumar R (2019) IoT technology, applications and challenges: a contemporary survey. Wirel Pers Commun 108:363–388. https://doi.org/10.1007/s11277-019-06407-w

Bhandari R, Kirubanand VB (2019) Enhanced encryption technique for secure iot data transmission. Int J Electr Comput Eng (IJECE) 9(5):3732–3738, ISSN: 2088-8708. https://doi.org/10.11591/ijece.v9i5.pp3732-3738

Chanal PM, Kakkasageri MS (2020) Security and privacy in IoT: a survey. Wirel Pers Commun 115:1667–1693. https://doi.org/10.1007/s11277-020-07649-9

Ekanayake E (2022) Securing IOT devices using blockchain technology. https://www.researchgate.net/publication/357954240_SECURING_IOT_DEVICES_USING_BLOCKCHAIN_TECHNOLOGY

Hassija V, Chamola V, Saxena V, Jain D, Goyal P, Sikdar B (2019) A survey on IoT security: application areas, security threats, and solution architectures. IEEE Access 7:82721–82743. https://doi.org/10.1109/ACCESS.2019.2924045

Khan H, Singh P (2022) Issues and challenges of Internet of Things: a survey. J Inform Electr Electron Eng 2(3, S No. 002):1–8. https://doi.org/10.54060/JIEEE/002.03.002

Kour K et al (2021) IoT: systematic review, architecture, applications and dual impact on industries. In: IOP conference series: materials, science and engineering, vol 1022, p 01205. https://doi.org/10.1088/1757-899X/1022/1/012053

Kumari P, Jain AK (2023) A comprehensive study of DDoS attacks over IoT network and their countermeasures. Comput Secur 127:103096, ISSN 0167-4048. https://doi.org/10.1016/j.cose.2023.103096

Roman R, Lopez J, Gritzalis S (2018) Evolution and trends in the security of the Internet of Things. IEEE Comput 51:16–25. https://doi.org/10.1109/MC.2018.3011051

Said O, Masud M (2013) Towards Internet of Things: survey and future vision. Int J Comput Netw 5:1–17. https://www.researchgate.net/publication/297141894_Towards_Internet_of_Things_Survey_and_Future_Vision

Sonar K, Upadhyay H (2014) A survey: DDOS attack on Internet of Things. Int J Eng Res Dev 10(11):58–63, e-ISSN: 2278-067X, p-ISSN: 2278-800X. https://www.ijerd.com/paper/vol10-issue11/Version_3/I10115863.pdf

Tiwary A, Mahato M, Chidar A, Chandrol MK, Shrivastava M, Tripathi M (2018) Internet of Things (IoT): research, architectures and applications. Int J Future Revol Comput Sci Commun Eng 4(3):23–27. http://www.ijfrcsce.org/index.php/ijfrcsce/article/view/1257

Wang F, Hu L, Zhou J, Zhao K (2015) A survey from the perspective of evolutionary process in the Internet of Things. Int J Distrib Sens Netw 11(3):462752. https://doi.org/10.1155/2015/462752

Wu H, Han H, Wang X, Sun S (2020) Research on artificial intelligence enhancing Internet of Things security: a survey. IEEE Access 8:153826–153848. https://doi.org/10.1109/ACCESS.2020.3018170

Zarpelão BB, Miani RS, Kawakani CT, de Alvarenga SC (2017) A survey of intrusion detection in Internet of Things. J Netw Comput Appl 84:25–37, ISSN 1084-8045. https://doi.org/10.1016/j.jnca.2017.02.009

Chapter 12
Synergy of Big Data and Blockchain for Secure Analytics Solution

Vijaya Kumbhar, Vinaya Keskar, and Pallavi Yarde

Abstract The rapid surge in data in the digital era has taken sophisticated analytics to transform raw data into profound insights. This chapter examines the cooperation of big data and blockchain technology to run safe and effective analytics results. With its intrinsic devolution, immutability, and transparency, blockchain offers a capable basis for firmly keeping and analyzing large-scale information. Big data analytics, on the other hand, provides clever approaches to obtaining big patterns, making projections, and gaining visions that may help industries like banking, healthcare, and supply chain organizations. The chapter opens with a review of blockchain, big data, and analytics, emphasizing their importance in advancing decision-making processes. The dealings of these technologies are further explored, proving the need to chain analytics with blockchain for safe data management and important analytics. Blockchain analytics, involving network, transaction, brilliant contract investigation, machine learning, and data mining, are tackled to accentuate its potential security. The tools, platforms, and technology that facilitate blockchain analytics are examined, with practical perceptions of their deployment. The chapter also examines real-world applications and uses improvements in which blockchain analytics betters big data operations, increasing security, efficiency, and faith. Open research questions and future proposals include increased data scalability, privacy defense, and efficient computer science methodologies. Initial topics in Blockchain analytics, such as analytical modeling, natural language processing, and optimization approaches, are also emphasized to give a road map for future development. The conclusion provides significant insights, emphasizing the rising consequences of blockchain analytics in

V. Kumbhar · P. Yarde (✉)
School of Computer Studies, Sri Balaji University, Pune, India
e-mail: pallavi.yarde@bitmpune.edu.in

V. Kumbhar
e-mail: veejeya.kumbhar@gmail.com

V. Keskar
Department of Computer Science and Application, ATSS College of Business Studies and Computer Application, Pune, India
e-mail: vasanti.keskar@gmail.com

K. Ahmad et al. (eds.), *Fostering Machine Learning and IoT for Blockchain Technology*,
Transactions on Computer Systems and Networks,
https://doi.org/10.1007/978-981-96-4078-2_12

securely continuing and reading big data. This chapter provides complete intelligence of how blockchain and big data may be operated to create innovative and safe analytics solutions.

12.1 Introduction

The inception of blockchain technology has overhauled the framework for loading, regulating, and assessing records. It was initially designed for a decentralized ledger for cryptocurrency transactions. It has evolved to sustain different uses, from supply chain management to healthcare records (Swan 2015). As data grows with its 4Vs (Volume, Velocity, Variety, and Veracity), the requirement for operative analytics solutions has become very important. Big data analytics serves as an essential facilitator for administrations to extract insights and assessments from large datasets (Manyika et al. 2011).

Inversely, the decentralized and distributed activities of blockchain data generate exceptional challenges for outmoded analytics methods. Blockchain analytics has evolved as a separate field, emphasizing emerging tools and performing abstract insights from blockchain data (Kim et al. 2020). This chapter proposes to determine the venture of blockchain analytics and big data, deliberating their opportunities, difficulties, and future directions in this moving field.

This outline sets up the scene for the chapter, underlining the juncture of blockchain and big data analytics, with a brief overview of the importance and challenges of this field. The references cited are a combination of academic and industry sources, giving a solid foundation for further exploration.

12.1.1 Compendium of Blockchain Technology

Blockchain technology is a dispersed, distributed ledger that is considered for transactions sideways a network of computers steadily and clearly (Nakamoto 2008). It empowers the conception of a perpetual, resilient record of transactions, disadvantaged by the need for an essential intercessor. The core attributes of blockchain technology include:

- *Decentralization*: It defines that blockchain functions on a network of computers, and not on a main server.
- *Immutable*: The blockchain transactions are permanent and cannot be altered.
- *Transparent*: All transactions are recorded visibly, allowing for thorough transparency.
- *Consensus*: Network nodes decide on the cogency of transactions over compound processes.

12.1.2 Compendium of Big Data

Big data discusses huge structured, semi-structured, and unstructured data that officialdoms generate, accumulate, and analyze (Kambatla et al. 2014). Big data is pigeon-holed by its:

- *Volume*: Substantial quantities of data, often above petabytes.
- *Velocity*: Expeditious data generation and dispensation.
- *Variety*: Miscellaneous data formats, including typescript, pictures, and sensor data.
- *Veracity*: Uncertainty and noise in the data.

The intersection of blockchain and big data presents prospects for secure, decentralized, and transparent information governance and analytics.

This segment provides a brief overview of blockchain technology and big data, emphasizing distinct critical components and qualities.

12.1.3 Compendium of Analytics

To gain insights and meaning from data, the mathematical algorithmic approach is implemented and called data analytics. It involves numerous tools, and methods to extract perceptions from data, including:

- *Descriptive Analytics*: Inspects previously available data to recognize patterns and trends (https: analytics.google.com/analytics/academy).
- *Predictive Analytics*: Implements quantitative models and machine learning algorithms to predict forthcoming actions (https:www.marketingprofs.com/topic/all/marketing-analytics).
- *Prescriptive Analytics*: Suggests analytical models and optimization techniques (Tapscott and Tapscott 2016).
- *Diagnostic Analytics*: Classifies the core reasons for difficulties and incongruities (Gartner 2020).

Big data analytics comes up as a shining area with the cumulative quantity of data being produced. It assesses huge datasets to uncover patterns, associations, and insights (https:researchmethod.net, descriptive-analytics).

The collaboration of blockchain technology and blockchain analytics is a growing field that combines blockchain technology with data analytics to provide secure, transparent, and tamper-proof insights. It has applications in supply chain management, finance, and healthcare (https:www.ibm.com/topics/predictive-analytics). Analytics has various applications across industries, including:

- *Business Analytics*: Supports business decision-making with data-driven insights (https: www.ibm.com/topics/prescriptive-analytics).

- *Data Analytics*: Business decisions depend on the insights taken from the data (https:www.thoughtspot.com/data-trends/analytics/diagnostic-analytics).
- *Web Analytics*: User behavior is observed through the analysis of website data (https:www.ibm.com/topics/big-data-analytics).
- *Marketing Analytics:* Here, the effectiveness of marketing campaigns is measured (https:coredevsltd.com, articles, Blockchain-analytics, #:~:text=Blockchain%20analytics%20is%20a%20field%20that%20combines%20the, i dentifying%2C%20clustering%2C%20and%20interpreting%20data%20from%20Blockchain% 20transactions).

12.1.4 Importance of Analytics in Blockchain

This portion stressed the significance of analytics in blockchain as well as big data (https:online.hbs.edu/blog/post/importance-of-business-analytics), It enables:

- *Network monitoring*: To ensure safety and veracity in blockchain networks, the glitches must be detected.
- *Transaction tracking*: Frauds or washing funds can be prevented by monitoring the transaction in blockchain networks.
- *Smart Contract Optimization*: The performance and efficiency of the blockchain network are improved by analyzing contract execution.
- *Decentralized application (dApp) development*: Gearing analytics to shape data-backed dApps.

12.1.5 Importance of Analytics in Big Data

Insights taken out from big data are necessary for analytics (https:hbr.org/topic/subject/analytics-and-data-science), which enables:

- *Pattern discovery*: Recognizing tendencies and patterns in hefty datasets.
- *Predictive modeling*: To forecast future events and behaviors, predictive models are suggested here.
- *Decision-making*: Decisions related to business are taken and Informed based on data-driven insights.
- *Optimization*: Data analysis helps to improve processes and operations.

12.1.6 Chemistry Between Blockchain and Big Data Analytics

The amalgamation of blockchain technology and big data analytics generates a powerful synergy (https:online.hbs.edu/blog/post/importance-of-business-analytics; https:hbr.org/topic/subject/analytics-and-data-science), that enables:

- *Secure data management*: The insights extracted by data analytics ensure the integrity and security of blockchain.
- *Transparent decision-making*: Blockchain's transparency balances evidence-based decision-making.
- *Decentralized insights*: The decentralized nature of blockchain enables decentralized analytics and its insights.

The organizations can open new projections for growth, innovation, and competitiveness by integrating blockchain and big data analytics.

12.2 Types of Blockchain Analytics

Numerous types of blockchain analytics give visions for blockchain memory-resident data and its usages.

12.2.1 Network Analysis

Network analysis is an important method in blockchain analytics which allows the examination of the blockchain network architecture, including nodes, transactions, and links (https:analytics.google.com/analytics/academy). Network analysis is a procedure used to study the associations and communications among nodes in a blockchain network.

12.2.1.1 Types of Network Analysis (https:www.marketingprofs.com/topic/all/marketing-analytics)

- *Node Analysis*: It analyzes individual nodes, their degree, significance, and bunching factor.
- *Edge Analysis*: Analyze the links between nodes, and include the edge's weight, direction, and strength.
- *Community Detection*: Checks for clusters or groups inside the network.
- *Network Centrality*: Observes the distinction of nodes inside the network.

12.2.2 Transaction Analysis

The examination of individual transactions within the blockchain network is called transaction analysis. It comprises analysis of the transaction data, counting sender, receiver, value, timestamp, and other relevant information (https:analytics.google.com/analytics/academy).

12.2.2.1 Types of Transaction Analysis

- *Transaction Profiling*: The study of individual transactions to identify the patterns and anomalies is considered transaction profiling. This analyzes the pieces of information like sender, receiver, value, and timestamp. The analysts can recognize scarce activity by creating profiles of typical transactions. This specifies untrustworthy or apprehensive behavior of transactions (https:analytics.google.com/analytics/academy).
- *Transaction Linking*: Transaction linking shows relationships between transactions, such as multi-hop transactions. This can help analysts understand the flow of resources and identify potential money laundering or terrorist financing activities (https:analytics.google.com/analytics/academy).
- *Transaction Prediction*: Transaction prediction includes envisaging imminent transactions predicated on historical data applying machine learning algorithms. This comforts analysts antedate and avoid fraudulent activity (https:analytics.google.com/analytics/academy).

12.2.3 Smart Contract Analysis

Observing the code and implementation of smart contracts to recognize latent vulnerabilities, errors, and inefficiencies is called Smart contract analysis. Smart contracts are self-governing contracts under the agreement engraved into lines of code (Nicholas et al. 2018).

12.2.3.1 Types of Smart Contract Analysis

- *Static Analysis*: The fixed or traditional analysis of a Smart Contract is its static analysis. (Nicholas et al. 2018).
- *Dynamic Analysis*: The Analysis of the Smart Contract's performance and its communications with the blockchain comes under dynamic analysis (Nicholas et al. 2018).
- *Symbolic Analysis*: Analysis of the Smart Contract behavior using symbolic execution (Nicholas et al. 2018).

- *Formal Verification*: It uses mathematical proofs to verify the accuracy of the Smart Contract (Nicholas et al. 2018).

12.2.4 Data Mining

Establishing designs and insights from bulky data arrays, like blockchain data is called data mining. It covers numerous methods to uncover valuable information from data (Han et al. 2011).

12.2.4.1 Types of Data Mining (Witten et al. 2016)

- *Descriptive Data Mining*: Labels the basic structures of the data.
- *Predictive Data Mining*: Envisages future scopes and patterns.
- *Prescriptive Data Mining*: Endorses plan of action centered on insights.

12.2.5 Machine Learning

Machine Learning enables the analysis of large datasets to expose perceptions, practices, and inclinations. Blockchain analytics encompasses the use of data analytics examine and interpret data warehoused on a blockchain.

12.2.5.1 Types of Machine Learning in Blockchain Analytics

- *Supervised Learning*: Used for Anomaly detection, Market trend forecasting, and identifying mistrustful transactions (Singh and Singh 2020).
- *Unsupervised Learning*: Employed for grouping, fraud detection, and finding patterns of transactions (Liu et al. 2020).
- *Reinforcement Learning*: It optimizes Smart Contracts and predicts the most lucrative transactions (Breiman 2001).

12.2.5.2 Machine Learning Algorithms for Blockchain Analytics

- *Decision Trees*: Used for classification and predictive analysis tasks, such as predicting transaction fees (MacQueen 1967).
- *Clustering Algorithms*: Designed for recognizing clusters of alike transactions or users (Rumelhart et al. 1986).
- *Neural Networks*: Employed for predicting asset prices, detecting fraud, and analyzing sentiment in blockchain data (Angles and Gutierrez 2008).

- *Graph Analytics*: Used for analyzing relationships between transactions, addresses, and entities (Pham and Lee 2020).

12.2.5.3 Big Data Challenges in Blockchain Analytics

- *Scalability*: It shows managing huge blockchain data (Singh and Singh 2020).
- *Data Quality:* Data quality is ensured by the precision and fullness of blockchain data (Zhang et al. 2019).
- *Data Privacy*: The sensitive information in blockchain data is protected (Pham and Lee 2020).
- *Computational Complexity*: Supervising the computational resources (Apache spark (n.d.)).

12.2.6 Visualization

Visualization in blockchain analytics for big data plays a critical part. It enables the interpretation and communication of complex data insights. Its effective techniques help to recognize the patterns, trends, and correlations in hefty datasets. Following are the types of visualization.

12.2.6.1 Network Visualization

Exemplifies relationships between transactions, addresses, and entities in a blockchain network. It helps to identify clusters, communities, and patterns in the network (Swan 2015).

12.2.6.2 Geospatial Visualization

It exhibits the physical scattering of blockchain activity, such as trading activity, asset prices, or data extraction. Also, helps to ascertain provincial drifts and patterns (Kim et al. 2020).

12.2.6.3 Temporal Visualization

It shows time-series data, such as price swings, transaction volumes, or block creation rates (Kim et al. 2020). In short, it helps identify trends, patterns, and correlations over time (Kambatla et al. 2014).

12.2.6.4 Hierarchical Visualization

It displays hierarchical relationships, like Smart Contract execution, transaction linkages, or blockchain network topology (https:www.marketingprofs.com/topic/all/marketing-analytics).

- *Transaction Pattern Analysis*: It analyzes suspicious activity such as illicit finance or

  Malfeasant financial operations in the blockchain network (Kambatla et al. 2014;https:www.marketingprofs.com/topic/all/marketing-analytics).

- *Asset Price Movement Analysis*: Involves visualizing asset price movements to find trends, patterns, and associations (Gartner 2020; https:online.hbs.edu/blog/post/importance-of-business-analytics).
- *Blockchain Network Analysis*: Involves visualizing blockchain networks to recognize connections in entities, such as transactions, addresses, and Smart Contracts (https:www.ibm.com/topics/predictive-analytics; https:hbr.org/topic/subject/analytics-and-data-science).
- *Smart Contract Visualization*: It visualizes Smart Contract execution to augment presentation, recognize logjams, and progress scalability (https:www.thoughtspot.com/data-trends/analytics/diagnostic-analytics; https:online.hbs.edu/blog/post/importance-of-business-analytics).

12.2.7 Time-Series Analysis

It is a crucial procedure in blockchain analytics that enables the examination of data points collected over time. It is used to identify patterns, trends, and correlations. There are the following types of Time-Series analysis.

12.2.7.1 Descriptive Analysis

It is the first step in time-series analysis which observes historical data to know patterns and trends (Swan 2015). This visualizes time-series data by calculating synthesis statistics (mean, median, standard deviation) to distinguish data distribution. It highlights the regular, repeating patterns and unexpected occurrences in data.

12.2.7.2 Inferential Analysis

It implements statistical functions to estimate future values based on chronological data (Manyika et al. 2011). This involves autoregressive (AR) models (using previous values to future values), Moving Average (MA) models (using imprecisions from past forecasts to estimate imminent standards), Self-Regressive Integrated Moving Average (ARIMA) models (combining AR and MA models with Incremental variation assessment to forecast future values).

12.2.7.3 Predictive Analysis

This involves working out machine learning models (e.g., neural networks, decision trees) on chronological data, evaluating model performance using metrics (e.g., mean absolute error, mean squared error), and predicting future values (Kim et al. 2020).

12.2.8 Clustering

12.2.8.1 K-means Clustering

It is a valuable tool in blockchain analytics that collects comparable data points into clusters founded on their structures. This technique can be implemented to design clusters of related transactions by looking at attributes such as value, gas price, and sender/receiver addresses. It can also group similar Smart Contracts by investigating their working and performance. A real-world deployment of clustering is Ethereum Smart Contracts. This methodology helps in identifying patterns and perceptions within blockchain data, improving empathy, and controlling transactions and Smart Contracts (MacQueen 1967).

12.2.8.2 Hierarchical Clustering

It creates a tree-like structure of clusters by amalgamation of dominant clusters. This approach can be used to identify tiered relations between transactions or smart contracts and provide deeper insights into their interconnections. The clustering structure can be visualized using Hierarchical clustering plots. It strongly represents the relationships and hierarchy among the data points. A real-world deployment of hierarchical clustering is analyzing the hierarchical structure of Bitcoin transactions. This study supports discovering patterns and relationships within the blockchain, to facilitate better empathy and management of the data (Johnson 1967).

12.2.8.3 DBSCAN Clustering

It binds statistics points into clusters founded on their density and proximity. This approach can recognize solid clusters of transactions or smart contracts. It also detects exceptions and unwanted values in the data. A practical application of DBSCAN clustering is identifying dense clusters of Ethereum transactions to detect potential cross-trading schemes. It focuses on density and proximity, to uncover patterns and indiscretions within the blockchain that enhance the detection and analysis of suspicious activities (Ester et al. 1996).

12.2.8.4 K-Medoids Clustering

This method converts similar information points into clusters to a central medoid point, which makes it operative in blockchain analytics. It identifies transactions or Smart Contracts for each cluster. Moreover, it is robust besides deafening or outlier data points that ensure more precise results. Its real-world application is clustering Bitcoin transactions which identifies transactions for each cluster. This method helps to understand the main features of various transaction groups that improve the analysis and supervision of blockchain data (Kaufman and Rousseeuw 1987).

12.2.8.5 Spectral Clustering

It is a useful technique in blockchain analytics that collects data themes into clusters created based on spectral properties, making. This method can identify clusters of transactions or Smart Contracts which share alike spectral properties. This allows the detection of relationships and patterns. It is operative in treating high-dimensional data, which is common in blockchain datasets. Ethereum smart contracts are the real-world application of spectral clustering that identifies similar spectral properties clusters (Ng et al. 2001).

12.2.9 Decision Trees

Decision trees are supervised learning procedures and use tree-like organization to categorize data or predicted outcomes. It also classifies transactions as genuine or fake and forecasts the possibility of a Smart Contract being oppressed. Decision trees provide strong and interpretable models for classification and prediction within the blockchain ecosystem, by analyzing various attributes and decision points. Building a decision tree to classify Ethereum transactions as legitimate or fraudulent is a real-world application of this technique (Quinlan 1986).

Popular decision tree procedures comprise CART (Classification and Regression Trees), ID3 (Iterative Dichotomizer 3), C4.5 (successor to ID3), and Random Forest (an ensemble learning method). In blockchain analytics, the C4.5 algorithm predicts the likelihood of a Bitcoin transaction. This application helps in considering transaction validation patterns and cultivating the efficiency of blockchain transaction processing (Quinlan 1986). Evaluation of Decision tree performance uses several metrics, like accuracy, exactitude, and remembrance. These performance indicators deliver complete visions of the efficacy and consistency of the decision tree model. This thorough evaluation ensures the decision tree model is robust and reliable for practical applications (Provost and Fawcett 2013).

Visualizing a decision tree predicts the prospect of a Bitcoin transaction. A tree diagram provides a clear representation of the conclusion paths and criteria used for guesses. Pruning can be used to remove less important branches, making the tree simpler and more interpretable. Feature importance plots highlight the most significant factors influencing the prediction, offering insights into which attributes most affect the confirmation likelihood. This comprehensive visualization aids in understanding and validating the decision tree model's predictive capabilities (Pedregosa et al. 2011).

12.3 Regression Analysis

12.3.1 Simple linear regression

It imitates the connotation of the dependent variables and a single independent variable of investigational data by suitable a linear equation. In blockchain analytics, it can be used to understand numerous relationships, such as demonstrating the linking between transaction volume and asset price, or estimating how a single variable, like transaction volume, influences transaction fees. The association between Bitcoin transaction volume and its price is a real-world application of simple linear regression. This analysis helps in understanding the fluctuations in transaction volume which influence the price of Bitcoin. This provides valuable perceptions for market analysis and prediction (Draper and Smith 1998).

12.3.1.1 Multiple Linear Regression

Multiple Linear Regression signifies the connection between a dependent variable and multiple independent variables. It supplies an extra comprehensive interpretation of the data by considering for various factors simultaneously. In blockchain analytics, this method is used to define the bond between transaction volume, asset price, and other market indicators, or to analyze the consequence of multiple variables on transaction fees. Modeling the association between Ethereum transaction size, cost, and gas price is a real-world application of multiple linear regression. This approach helps to show the interaction of variables and inflection of each other. By examining these relationships, analysts can upsurge a deeper understanding of market changes and make further knowledgeable estimations and conclusions (Kutner et al. 2004).

12.3.1.2 Logistic Regression

Logistic regression shows the possibility of a double importance founded on at least one independent variable. It can be employed to predict the prospect of a transaction's existence or to analyze the influence of various factors on Smart Contract espousal in blockchain analytics. In the real world, logistic regression envisages the chance of a Bitcoin transaction based on factors such as transaction fees, volume, and network congestion that will be comprised in the blockchain. This helps in accepting and heightening transaction confirmation processes in the blockchain ecosystem (Hosmer and Lemeshow 2000).

12.3.1.3 Ridge Regression

Ridge regression is a model stabilization method designed to alleviate hyper-detailed articulation in regression models by including a forfeit term into the cost function. It is valuable in blockchain analytics to enhance the sturdiness and comprehensibility of prototypes. The analysts can address the challenge of excessive model complexity. It can be worked to improve predictive models, such as those forecasting Ethereum transaction fees in blockchain analytics (Hoerl and Kennard 1970).

12.3.1.4 Lasso Regression

Lasso regression is a constraint optimization method that adds a penalty time to the cost function. This encourages the model to choose a portion of the most significant features. It is used to improve model transparency and focus on the most relevant variables that benefit blockchain analytics. Here, analysts can well recognize and line up salient attributes that make an impact on predictive outcomes. Hence, refining the model's simplicity and performance. Practically, Lasso regression refines models predicting Bitcoin transaction volume. It helps in isolating and choosing critical factors swaying transaction volume, leading to more rationalized and interpretable models that provide workable perceptions of Bitcoin market dynamics (Tibshirani 1996).

12.3.1.5 Anomaly Detection

Anomaly detection employs statistical methods to identify data points that diverge from recognized standards. It is decisive for noticing misdeeds and possible issues within blockchain data. With this, analysts can efficiently identify rare transaction patterns and detect susceptibilities in Smart Contracts. It can be used to examine Bitcoin transaction data and uncover transactions. This strays from anticipated patterns and potentially signals fake activities or security breaks. This approach enhances the ability to maintain the truth of blockchain networks by indicating anomalies and protection against potential threats (Chandola et al. 2009).

12.3.1.6 Machine Learning Anomaly Detection

It utilizes advanced algorithms to know data points that evocatively swerve from the standard. It enables the discovery of composite and delicate anomalies. It is used to uncover complicated patterns within transaction data and identify erudite attacks on Smart Contracts. It can analyze massive and polygonal datasets. For example, in Ethereum blockchain analytics, machine learning anomaly detection can be engaged to recognize complex patterns in transaction data. This discloses unusual activities and potential risk exposures with superior exactness (Aggarwal 2017).

12.3.1.7 One-class SVM Anomaly Detection

It uses a unique approach to identify data points that stand out from the usual pattern. It focuses on identifying aberrations in a single class of data. In blockchain analytics, this is predominantly valuable for seeing uncommon action and possible susceptibilities. In Bitcoin transactions, one-class SVM can help detect transactions that do not have suitable distinctive patterns. This indicates fake behavior or system weaknesses. Likewise, it is used to expose weaknesses in Smart Contracts by classifying nonconformities from normal operations. With this, analysts can monitor and secure blockchain networks (Schölkopf et al. 2001).

12.3.1.8 Local Outlier Factor (LOF) Anomaly Detection

It influences an algorithm to classify data points that meaningfully fluctuate from their neighbors. It is effective for noticing irregularities in data w.r.t. neighboring points. In blockchain analytics, LOF can be used to detect unusual transaction designs and weaknesses in Smart Contracts. In Ethereum transactions, LOF anomaly detection can help uncover transactions that deviate from typical behaviors, signaling potential fraud or irregularities. LOF provides a polished outlook of anomalies that enhances monitoring and secures blockchain networks effectively (Breunig et al. 2000).

12.3.1.9 Isolation Forest Anomaly Detection

It employs an exclusive algorithm to isolate anomalies by producing a forest of random trees. It emphasizes individual data points that are different from the standard. It is done by separating them with fewer splits related to normal data points. Isolation forest can be applied to identify unfamiliar transaction patterns and potential vulnerabilities in Smart Contracts. In Bitcoin analytics, this can recognize transactions that depart from expected patterns which highlights probable fraud or irregular activity. Analysts can increase monitoring skills for blockchain networks and address anomalies effectively. This ensures the authenticity and protection of the transactions and Smart Contracts within the ecosystem (Liu et al. 2008).

12.3.1.10 Graph Analysis

Blockchain data is analyzed using the graph structure which involves investigating the connections between nodes (vertices) and edges (connections). This represents the connections between various data entities. This method is appreciated for indulging intricate connections and interfaces within the network in blockchain analytics. Graphs model the transaction relationships among addresses, illuminating patterns and connections that specify their behaviors or trends. Moreover, they analyze interactions between Smart Contracts, providing insights into how contracts interact with one another and the flow of assets. For example, in Bitcoin analytics, graph analysis can model the relationships between transaction addresses, helping to uncover patterns such as clusters of addresses involved in specific types of transactions or potential networks of fraudulent activities. This method improves the unusual patterns, improves security, and gains visions into blockchain networks (West 2001).

12.3.1.11 Community Detection Algorithms

These algorithms are designed to know clusters of nodes that are densely present within a graph structure. In blockchain analytics, these algorithms discover expressive patterns and relations in the data. It can help find groups of addresses that display similar transaction behaviors. Which is beneficial for spotting patterns of coordinated activity or latent fraud. Also, this method can analyze interactions between different contracts for Smart Contract communities and create clusters based on their functionality or usage (Fortunato 2010). The tangible application of Community detection is used to find groups of addresses with comparable transaction patterns based on their transaction history and interactions in Bitcoin. Here, analysts can uncover networks of addresses that frequently transact with each other. This provides visions into market behaviors, fake activities, or collaborative schemes. This method improves the skill to analyze and construe complex blockchain networks and offers a purer assessment of transaction dynamics and Smart Contract exchanges (Fortunato 2010).

12.3.1.12 Shortest Path Analysis

The Shortest path analysis includes deciding the extremely organized route among nodes in a chart for factors such as distance or cost. In blockchain analytics, this analysis can be implemented in various facets of transaction and contract interactions.

It identifies the most efficient transaction paths between addresses, which is useful for optimizing transaction processing and reducing transaction fees. Additionally, it can be used to analyze the execution flows of smart contracts, identifying the most direct paths through contract functions and interactions. Practically, it is implemented in Ethereum.

Analysts can improve transaction processing strategies, enhance the network's performance, and potentially reduce costs associated with transactions by mapping out transaction routes and optimizing for efficiency. This approach delivers meaningful acumens hooked on the flow of assets and interactions within the blockchain, prominent to extra proficient and cost-effective network management (Dijkstra 1959).

12.3.1.13 Graph Embedding

It transforms nodes into vectors in a graph. This allows Machine Learning practices to analyze relationships and patterns in blockchain analytics. For example, in Bitcoin transaction networks, graph embedding can be used to analyze node relations by adapting transaction addresses into vector representations. This alteration agrees with the identification of clusters of similar nodes, revealing patterns such as groups of addresses with similar transaction behaviors or interactions. This method enhances the ability to detect anomalies, identify influential nodes, and understand the overall structure of the transaction network, providing deeper insights into the dynamics of blockchain activities (Perozzi et al. 2014).

12.3.1.14 Natural Language Processing (NLP)

There are a lot of processing that come under the Natural Language Processing (NLP). Some of these are given below.

Text Pre-processing (TPP)

Text Pre-processing (TPP) is an important step in preparing text data for NLP analysis. It includes standardizing blockchain-related text data, such as transaction descriptions or contract code comments in the form of data in proper preparation for assisting analysis. It applied to numerous types of text data associated with blockchain transactions or Smart Contracts. This includes eliminating inappropriate information, normalizing terminology, and modifying discrepancies. This way, analysts can efficiently perform sentiment analysis and topic modeling.

Sentiment Analysis

It evaluates the expressive nature or attitude conveyed by text data. It offers an understanding of the feelings about a particular topic or entity by people. Practically, sentiment analysis is valuable for knowing market dynamics and user perceptions. By sentimental analysis, analysts can measure public opinion and identify trends and patterns that may influence market behavior. In reality, sentiment analysis is examining Bitcoin-related Twitter posts. This helps analysts provide insights into public attitudes towards Bitcoin, track sentiment trends over time, and assess how sentiment shifts might correlate with market movements or events. This also helps in considerate the wider effect of social and media sentiment on the cryptocurrency market (Pang and Lee 2008).

Topic Modeling

It is a method used to uncover fundamental themes within a collection of text data. Topic modeling can be realistic to various types of text data to obtain intuitions and trends related to blockchain technologies. For example, topic modeling can be discarded to classify key topics and themes in blockchain-related text data, such as forum posts, white papers, or technical documentation. By analyzing these texts, analysts can uncover the main areas of focus and concern within the blockchain community. Identifying the key topics in Ethereum-related forum posts is one of the practical example of topic modeling. This shows the emerging issues, and the overall sentiment of the Ethereum community (Blei 2012).

Named Entity Recognition (NER)

NER identifies and classifies objects in text data, for example, names, dates, or some other items in the form of key information. In the context of blockchain analytics and blockchain transactions and interactions, NER is used to find and sort entities like addresses, Smart Contracts, and other relevant identifiers within text data. Practically, it identifies and categorizes Bitcoin addresses within transaction descriptions to analyze transaction patterns, network activity, and the relationships between different addresses that help in pursuing, indulgent transaction flows and identifying potential anomalies or significant entities within the Bitcoin ecosystem (Nadeau and Sekine 2007).

Machine Translation

This technology converts text from one language to another enabling trans-lingual understanding and analysis. In blockchain analytics, Machine translation expands the room for data analysis by breaking down language barriers. It is used in blockchain-related text data to translate from various languages making it accessible for global analysis. According to blockchain trends and activities, it is used to examine news articles, research papers, or forum discussions in different languages, which can provide a broader perspective. In reality, it is deciphering Chinese Bitcoin-related newscasts and articles for global analysis. Translating these non-English articles into English or any other popular language leads to a more comprehensive understanding of global market sentiment, trends, and developments related to Bitcoin (Koehn 2010).

12.3.1.15 Predictive Modeling

Predictive modeling in big data analytics for blockchain uses a progressive set of rules to analyze blockchain data, forecast inclinations, and detect irregularities in transactions. It improves security, augments practices, and cares about decision-making within decentralized networks. There are various models implemented here.

12.3.1.16 Regression Model

These models remove the continuous outcomes founded on input variables. In blockchain analytics, analyzing historical and current data is treasured for predicting forthcoming measures. The regression models can be implemented to:

- *Predict asset prices* stuck on several market indicators, like transaction volume, market sentiment, and macroeconomic factors.
- *Forecast transaction volumes* by analyzing historical transaction data and knowing trends or patterns that impact future activity.

A real-life use case of Regression models is predicting Bitcoin prices. By manipulating old charge data, trading volumes, and other applicable bazaar indicators, it can estimate future Bitcoin prices, assisting stakeholders (Hastie et al. 2009).

12.3.1.17 Decision Trees

Decision trees construct a tree-like model of conclusions and their conceivable significances and predict the outcomes by learning verdict rules from data. For blockchain analytics, Decision trees can be effectively used to:

- *Classify transactions* as genuine or fake by analyzing attributes such as transaction value, frequency, and sender/receiver details.
- *Predict Smart Contract outcome*: It helps in determining the possible triumph or dissatisfaction of Contract executions based on input parameters.

Practically, decision trees classify Ethereum transactions as genuine or fake. Decision trees help identify patterns and irregularities which leads to accurate detection of deceitful activities within the Ethereum network. This enhances safety and belief in blockchain transactions (Breiman 2001).

12.3.1.18 Random Forests

It combines multiple decision trees, leveraging the collective wisdom of several trees to improve accuracy and robustness and predict outcomes. In blockchain analytics, Random Forests can be used to:

- *Predict asset prices*: Based on various market indicators, such as trading volume, historical prices, and market sentiment it predicts asset prices.
- *Identify key factors*: It influences transaction volumes by analyzing multiple Decision trees and aggregating their results.

Practically, Random forests predict Bitcoin prices. By combining multiple decision trees that analyze different aspects of the market, random forests provide a more accurate and stable prediction of Bitcoin prices, helping investors and analysts make informed decisions based on comprehensive insights (Liaw and Wiener 2002).

Neural Networks (NN)

It predicts results by learning complex patterns and connotations in data through connected layers of neurons. In blockchain analytics, NNs can be utilized to:

- *Predict asset prices* based on various market indicators, such as trading volume, former values, and other relevant features.
- *Identify anomalies* in transaction data by learning patterns that diverge from the norm. This helps in noticing uncommon or fraudulent activities.

In reality, NNs are predicting Ethereum prices. Training on extensive historical price data and market indicators can uncover complex patterns and trends, generating extra accurate and nuanced predictions of Ethereum prices (Goodfellow et al. 2016).

12.3.1.19 Optimization Techniques

Optimization techniques are mathematical approaches to finding the most efficient key to a problem. It does this by maximizing or minimizing a specific impartiality. They are mostly applied in fields like engineering, economics, and Machine Learning to improve performance and outcomes. The optimization technique is a collection lot of techniques, some of these are given below.

12.3.1.20 Linear Programming

Linear programming is used to augment linear objective w.r.t. linear constraints. In blockchain analytics, it can be used to:

- *Optimize portfolio allocation* for cryptocurrency funds. It helps investors to allocate resources efficiently across various cryptocurrencies. It maximizes a specific investment goal.
- *Minimize transaction fees* for blockchain networks by heightening the allocation of resources or picking transaction parameters to cut whole charges.

In reality, it is improving portfolio allocation for Bitcoin and Ethereum investments. With this, investors can regulate the best allocation policy that steadies risk and reward which leads to more efficient and profitable investment decisions in the cryptocurrency market (Charnes and Cooper 1961).

12.3.1.21 Integer Programming

Integer programming optimizes objective functions where decision variables are constrained to integer values, handling scenarios where solutions must be whole numbers. In blockchain analytics, integer programming can be utilized to:

- *Optimize smart contract execution* for blockchain networks by determining the most efficient execution paths and resource allocation, ensuring that smart contracts are executed in a manner that maximizes efficiency and reduces costs.
- *Minimize energy consumption* for cryptocurrency mining by optimizing the allocation of mining resources and scheduling, which helps in reducing operational costs and environmental impact.

A real-world application of integer programming is optimizing smart contract execution for Ethereum networks. By applying integer programming techniques, it is possible to determine the most efficient strategies for executing smart contracts, improving performance, reducing transaction costs, and optimizing resource usage within the Ethereum network (Wolsey 1998).

12.3.1.22 Dynamic Programming

Dynamic programming is a technique for resolving compound glitches by infringing them down into simpler, overlying sub-problems and resolving separately sub-problem just as soon as, packing the results for reuse. In blockchain analytics, dynamic programming can be realistic to:

- *Augment cryptocurrency trading strategies* by finding the most effective trading decisions over time, considering various constraints and market conditions.
- *Minimize transaction confirmation times* for blockchain networks by optimizing the sequence and timing of transactions to reduce delays and enhance processing efficiency.

A real-world application of dynamic programming is optimizing cryptocurrency trading strategies. By using dynamic programming techniques, traders can develop strategies that account for historical price data, trading costs, and market conditions, leading to more informed and profitable trading decisions (Bellman 1957).

12.3.1.23 Stochastic Optimization

Stochastic optimization is used to address problems involving uncertainty by optimizing objective functions in the presence of random variables or uncertain parameters. In blockchain analytics, stochastic optimization can be applied to:

- *Optimize portfolio allocation* for cryptocurrency investments where returns are uncertain, allowing investors to develop strategies that account for variability in market performance.
- *Minimize transaction fees* for blockchain networks where transaction volumes are uncertain, helping to optimize resource allocation and cost management under unpredictable conditions.

A real-world application of stochastic optimization is optimizing portfolio allocation for Bitcoin and Ethereum investments with uncertain returns. By incorporating uncertainty into the optimization model, investors can develop more robust strategies that account for potential fluctuations in returns, improving their ability to manage risk and maximize investment outcomes (Powell 2011).

12.3.1.24 Metaheuristics Algorithms

Metaheuristics are high-level algorithms designed to decipher composite optimization problems by reconnoitering a great key space and finding near-optimal solutions through various strategies. In blockchain analytics, metaheuristics can be applied to:

- *Optimize smart contract execution* for blockchain networks by finding efficient ways to execute contracts while balancing factors such as cost, speed, and resource utilization.
- *Minimize energy consumption* for cryptocurrency mining by optimizing the allocation of computational resources and scheduling, thereby reducing operational costs and environmental impact.

A real-world application of metaheuristics is optimizing smart contract execution for Ethereum networks. By using metaheuristic approaches, such as Genetic Algorithms, Simulated Annealing, or Particle Swarm Optimization, one can explore different execution strategies and configurations to improve efficacy, diminish outlays, and recover whole enactment within the Ethereum network (Glover and Kochenberger 2003).

12.4 Tools and Platforms for Blockchain Analytics

The field of blockchain analytics has seen significant growth, driven by the increasing complexity and volume of blockchain data. Several tools and platforms have emerged to help users analyze and interpret this data, providing valuable insights for various applications, from compliance and fraud detection to investment and market analysis. Some of the prominent tools and platforms are:

12.4.1 Chainalysis

Chainalysis does blockchain analysis for transaction monitoring, investigative analysis, and compliance reporting and provides consent solutions. It helps organizations adhere to regulatory requirements and detect suspicious activities (Chainalysis (n.d.)).

12.4.2 Elliptic

With the help of blockchain analytics tools, Elliptic offers transaction monitoring and forensic analysis that benefit financial institutions and regulators to detect and avoid cryptocurrency-related crimes like Anti-Money Laundering (AML) regulations (Elliptic (n.d.)).

12.4.3 CipherTrace

CipherTrace provides cryptocurrency AML and blockchain analytics solutions. The tools of CipherTrace permits organizations to trace blockchain transactions and detect fraud and helping them manage risks and comply with regulatory standards effectively (CipherTrace (n.d.)).

12.4.4 Nansen

Nansen provides a blockchain analytics platform with a particular emphasis on Ethereum. It gives visions into Ethereum transactions, DeFi doings, and wallet behaviors, users informed investment decisions and stalking inclinations within the Ethereum ecosystem (Nansen (n.d.)).

12.4.5 Dune Analytics

Dune analytics helps consumers to produce, share, and explore blockchain data queries and conjuring up, focusing on Ethereum and other blockchain networks.

Focused on Ethereum and other blockchain networks. It empowers users to generate custom analytics, share insights, and interact with blockchain data through an intuitive interface (Dune Analytics (n.d.)).

12.4.6 Glassnode

Glassnode provides on-chain data and analytics for various cryptocurrencies, including Bitcoin and Ethereum. The platform offers detailed insights into network health, market sentiment, and user behavior, aiding in market analysis and investment strategies (Glassnode (n.d.)).

12.4.7 Bitquery

Bitquery offers blockchain data infrastructure and analytics tools, including Application Program Interfaces (APIs) and data visualizations. The platform supports Research & Development (R&D) by providing access to complete blockchain data. This enables users to analyze and shape applications on top of blockchain networks (Bitquery (n.d.)).

These tools and platforms signify a broad range of capabilities in blockchain analytics, catering to various needs such as regulatory compliance, investment analysis, and transaction monitoring. As the blockchain ecosystem endures to progress, these solutions play a critical role in revealing intuitions and driving current decision-making.

12.5 Technologies for Blockchain Analytics

Blockchain analytics believes on a variety of technologies to process, analyze, and interpret the vast measures of data produced by blockchain networks. These skills helps to gain visions, leanings, and make well-versed conclusions built on blockchain data. Here’s an overview of some key technologies used in blockchain analytics:

12.5.1 Big Data Technologies

Big data technologies handles bulk data produced by blockchain transactions. These includes distributed computing frameworks and data storage solutions to facilitate the processing and analysis of blockchain data.

- *Apache Hadoop*: It is an open-source framework which permits distributed execution of massive datasets across clusters of computers implemented with simple programming models (Apache Hadoop (n.d.)).
- *Apache Spark*: It is a combined analytics engine for big data processing, recognized for its haste and capacity to manage real-time data analytics (Apache Spark (n.d.)).

12.5.2 Machine Learning and Artificial Intelligence

These technologies are used to obtain patterns, predict trends, and detect anomalies in blockchain data with the help of algorithms and models for classification, regression, and clustering.

- *Neural Networks (NNs)*: These are used for foretelling blockchain trends and analyzing complex patterns in transaction data (Goodfellow et al. 2016).
- *Natural Language Processing (NLP)*: These applied to analyze and understand word-based data from blockchain-related documents and communications (Manning et al. 2008).

12.5.3 Graph Databases

Graph databases manages and analyzes associations in data points like transactions, addresses, and smart contracts efficiently, and producing them perfectly for blockchain analytics.

- *Neo4j*: A popular graph database that supports complex queries and analytics on interconnected data, such as blockchain transactions and smart contract interactions (Wang and Wang 2020).
- *ArangoDB*: A multi-model record that supports graph, document, and key-value data models, suitable for various blockchain data analysis tasks (ArangoBD (n.d.)).

12.5.4 Blockchain Explorers

Blockchain explorers allow users to view and search blockchain transactions, blocks, and addresses. These tools provide comprehension of blockchain activity and support various forms of data analysis.

- *Etherscan*: An Ethereum blockchain explorer that provides detailed information on transactions, smart contracts, and addresses on the Ethereum network (Etherscan (n.d.)).

- *Blockchair*: A multi-currency blockchain explorer that supports Bitcoin, Ethereum, and other blockchain networks (Blockchair (n.d.)).

12.5.5 Smart Contract Analysis Tools

Smart Contract analysis tools focus on reviewing, checking, and optimizing Smart Contracts. This ensures Smart Contracts are secure, efficient, and free from vulnerabilities.

- *Mythril*: A safety analysis tool for Ethereum smart contracts that identifies potential vulnerabilities and security issues (Mythril (n.d.)).
- *Slither*: A fixed analysis framework for smart contracts that provides in-depth analysis and vulnerability detection (Slither (n.d.)).

12.5.6 Data Visualization Tools

The assistance of these tools visually represents the interpretation and analysis of complex datasets.

- *Tableau*: A powerful data visualization tool that supports interactive and shareable dashboards, useful for visualizing blockchain analytics (Tableau (n.d.)).
- *Power BI*: A business analytics tool by Microsoft that provides interactive visualizations and business intelligence capabilities with a user-friendly interface (MicrosoftPowerBI (n.d.)).
- *Graphite*: It is an open-source, highly scalable graphing and monitoring tool that allows users to track and visualize metrics from various sources like blockchain data. (Graphite (n.d.)).

These technologies play a fundamental role in blockchain analytics vis-à-vis managing and analyzing blockchain data efficiently. By leveraging these tools, organizations can gain valued perceptions, optimize blockchain operations, and augment decision-making processes.

12.6 Applications and Use Cases of Blockchain Analytics in Big Data

Blockchain analytics against the backdrop of big data offers powerful perceptions and applications across various sectors. The organizations can address complex challenges for misuse of enormous amounts of data generated by blockchain networks. Some prominent key applications and use cases are:

12.6.1 Financial Fraud Detection

Blockchain analytics notices and balks financial scams by analyzing operation patterns and identifying indiscretions. Big data techniques help in processing huge blockchain transactions to spot infrequent actions that may designate fraudulent behavior.

Use Case: Financial organizations monitor cryptocurrency transactions for symbols of money laundering or fraudulent actions. For example, Chainalysis and Elliptic provide tools to track and analyze transactions across blockchain networks to recognize doubtful behavior (Chainalysis (n.d.)).

12.6.2 Regulatory Compliance

Description: Blockchain Ensures agreement with regulatory requirements. Analytics platforms help organizations adhere to laws and regulations by providing detailed transaction records and insights into blockchain activities.

Use Case: Regulatory bodies and financial institutions use blockchain analytics tools to observe Anti-money Laundering (AML) and Know-your-customer (KYC) protocols. Tools like CipherTrace and Elliptic help in maintaining compliance by analyzing transaction data and verifying user identities (CipherTrace (n.d.)).

12.6.3 Market Intelligence and Investment Strategies

Description: Blockchain analytics aids investors and market analysts in making informed decisions by analyzing market trends, asset prices, and trading volumes. Big data techniques provide actionable insights for developing investment strategies.

Use Case: Investors use analytics platforms like Glassnode and Nansen to track cryptocurrency market trends, monitor price movements, and assess the health of blockchain networks. This information helps in making strategic investment decisions and managing portfolios (Glassnode (n.d.)).

12.6.4 Smart Contract Optimization

Description: Analyzing smart contracts helps optimize their performance and security. Blockchain analytics can identify inefficiencies, vulnerabilities, and patterns that affect smart contract execution.

Use Case: Tools such as Mythril and Slither analyze Ethereum smart contracts to detect vulnerabilities and optimize code performance. By analyzing execution patterns and potential security issues, developers can enhance the reliability and efficiency of smart contracts (Mythril (n.d.)).

12.6.5 Supply Chain Management

Description: Blockchain analytics can improve supply chain management by furnishing transparency and ascribability. Analyzing blockchain data helps observe the crusade of properties, confirm legitimacy, and guarantee acquiescence with standards.

Use Case: Companies like IBM and Maersk use blockchain analytics to monitor and optimize supply chains. By analyzing transaction data recorded on Blockchains, they enhance transparency, reduce fraud, and improve efficiency in logistics and supply chain operations (IBM Blockchain (n.d.)).

12.6.6 Customer Insights and Personalization

Description: Blockchain analytics provides perceptions of client behavior and partialities by analyzing transaction data. This evidence can be used to personalize offerings and improve customer engagement.

Use Case: Businesses use blockchain analytics to analyze customer transaction patterns and preferences, enabling them to tailor marketing strategies and enhance customer experiences. Platforms that integrate blockchain data with big data analytics facilitate these personalized approaches (Bitquery (n.d.)).

12.6.7 Cybersecurity

Description: Blockchain analytics can bolster cybersecurity by identifying and mitigating threats. Analyzing blockchain data helps detect malicious activities and strengthen security protocols.

Use Case: Cybersecurity firms use blockchain analytics to monitor for unusual activity and potential attacks on blockchain networks. By analyzing transaction data and network behaviors, they can identify vulnerabilities and respond to threats effectively (ArangoDB (n.d.))

These applications of blockchain analytics in big data illustrate its transformative potential across various sectors. By leveraging these insights, organizations can enhance security, optimize operations, and create data-driven verdicts that initiate innovation and efficiency.

12.7 Open Issues and Future Research Directions

12.7.1 Emerging trends and advancements in Blockchain analytics

12.7.1.1 Interoperability and Cross-chain Solutions

Compatible keys collapse obstacles by edifice connections among numerous blockchain ecosystems, making the system more scalable and opening up possibilities for novel cross-chain applications and DeFi. Interoperability enables the seamless exchange of data and assets between different blockchain networks, fostering a more connected and robust ecosystem (Belchior and Freitas 2020).

12.7.1.2 Decentralized Finance (DeFi) Evolution

DeFi is changing from basic lending and borrowing protocols to more advanced financial instruments, such as decentralized derivatives, options trading, and algorithmic stablecoins. DeFi's growth has led to the development of more complex financial instruments, increasing the scope of decentralized finance (Zheng and Xie 2020).

12.7.1.3 NFTs Beyond Digital Art

NFTs' application beyond digital art and collectibles has increased, with tokenizing physical assets such as real estate and intellectual property, and games utilizing NFTs to represent in-game items or generate unique gaming experiences. NFTs have expanded beyond digital art, enabling the tokenization of physical assets and creating new use cases in gaming and beyond (Wang and Wang 2020).

12.7.1.4 Sustainability and Green Blockchain

All-green blockchain projects are looking at energy-efficient consensus algorithms such as PoS and DPoS to reduce environmental impact. The shift towards sustainable blockchain solutions addresses environmental concerns, promoting eco-friendly practices in the industry (Krause and Tolaymat 2020).

12.7.1.5 Integration of Blockchain with the Internet of Things (IoT)

Blockchain technology combined with IoT is transforming how devices interoperate securely and share data quickly, with decentralization and an unalterable ledger guaranteeing the safety and authenticity of IoT data. The integration of blockchain and IoT enables secure, decentralized, and efficient data sharing between devices, revolutionizing industries such as supply chain management and smart cities (Khan and Salah 2020).

12.7.1.6 Blockchain Advancement Increases With Enterprise Adoption

BlackRock's recent introduction of a Bitcoin spot ETF has ignited renewed institutional interest in cryptocurrencies and sparked discussions about the implications of investments in the broader financial ecosystem. Enterprise adoption of blockchain technology is driving innovation and growth, with institutions exploring its potential for various use cases (BlackRock 2022).

12.7.1.7 Wall Street Embraces Asset Tokenization and Digital Transactions

Prominent financial institutions such as JPMorgan Chase, Goldman Sachs, BlackRock, and Fidelity are recognizing the transformative potential of blockchain technology by exploring the reaches of asset tokenization and digital transactions. Wall Street's embrace of blockchain Technology marks a significant shift towards digitalization and tokenization of assets, transforming traditional financial systems (JPMorgan Chase 2022).

12.7.1.8 The Rising Power of Decentralized Finance (DeFi)

DeFi expands its influence in the FinTech scene, enabling the birth of new facets, like GameFi, SocialFi, and InsureTech, with the majority of these platforms owing their existence to Ethereum's powerful smart contract abilities. DeFi's growth has led to the emergence of new financial paradigms, such as GameFi and SocialFi, built on blockchain technology (Zheng and Xie 2020).

12.7.2 Open research challenges and future directions

12.7.2.1 Scalability and Performance

Improving the scalability and performance of blockchain analytics to handle large-scale data and complex queries. Scalability is a major challenge in blockchain analytics, as the volume of data generated by blockchain networks is increasing rapidly. Distributed Databases and Parallel Computing are the technologies that explore new data storage to address the challenges (Wang and Wang 2020; Alharby and Moorsel 2020).

12.7.2.2 Advanced Data Analytics

The progressive new data analytics techniques, such as ML and AI extract insights from blockchain data. These help to uncover hidden patterns and relations in blockchain data, enabling more accurate predictions and better decision-making (Singh and Singh 2020).

12.7.2.3 Interoperability

The data exchange and analysis can be easily implemented with interoperability between different blockchain platforms and analytics tools (Belchior and Freitas 2020; Zhang and Wen 2020).

12.8 Conclusion

12.8.1 Summary of key takeaways

Blockchain analytics is a rapidly evolving field with immense potential to transform industries by providing insights and intelligence from blockchain data. The following are the important key points:

- Ability to progress transparency and trust, boost security and compliance, upsurge efficiency and productivity, and permit better decision-making and visions makes it a game-changer for businesses.
- Prioritizing data quality and integrity ensures accurate, reliable, and actionable blockchain data.
- Collaboration and knowledge-sharing between industry stakeholders, researchers, and practitioners are essential for advancing the field of blockchain analytics. This enables organizations to explore new applications, improve existing solutions, and address emerging challenges.
- Safeguarding regulatory and authorized compliance mitigates the risks and maximizes the benefits.

The applications of blockchain analytics are far-reaching, straddling businesses like supply chain management, finance, healthcare, and cybersecurity. Implementing innovation and experimentation in organizations unlocks the full potential of blockchain analytics which helps them to achieve their goals in a progressively complex and competitive landscape.

12.8.2 Importance of Blockchain analytics in big data

Blockchain analytics offers a safe, devolved, clear direction to manage and analyze big datasets. Big data analytics is important for businesses to gain perceptions and make informed decisions. However, big data analytics also poses significant challenges, such as data security, quality, and integrity. This ensures the accuracy, completeness, and consistency of their data.

Blockchain analytics enables real-time data analysis so that businesses respond quickly to changing market conditions and make data-driven decisions. It also integrates the data from multiple sources, creating a single, unified view of the data. This is particularly important in industries such as finance, healthcare, and supply chain management, where data accuracy and integrity are critical.

In addition, blockchain analytics provides a tamper-proof audit trail, which ensures the un-alteration or deletion of data. This is essential for regulatory compliance and data governance. Overall, blockchain analytics is a powerful tool for big data, enabling organizations to unlock the full potential of their data while ensuring its security, quality, and integrity.

12.8.3 Final thoughts and recommendations

In inference, blockchain analytics is an evolving field and has the potential to transform various industries by informing insights and intelligence from blockchain data. While there are challenges and limitations to consider, such as expandability and performance concerns, data eminence and reliability concerns, regulatory and legal reservations, and talent and skills gaps, the benefits of blockchain analytics far offset the disadvantages.

In anticipation of the future, blockchain analytics will critically drive business value and innovation. To get the best out of its potential, organizations should line up data quality, integrity, and security when implemented with blockchain analytics solutions.

To remain at the forefront, organizations should invest in blockchain analytics research and development. Develop and implement solutions that resolve business needs and pain points. The benefits of blockchain analytics will maximize with fostering the culture of innovation and experimentation, and encourage cross-functional collaboration. Finally, state-of-the-art regulatory and legal developments in the blockchain space are essential to ensure compliance and reduce risks.

References

Aggarwal CC (2017) Outlier analysis. Springer

Alharby M, van Moorsel A (2020) Blockchain scalability: a survey of solutions and research directions. IEEE Access 8:164055–164075

Angles R, Gutierrez C (2008) Survey of graph database models. ACM Comput Surv 40(1):1–39

Apache Hadoop (n.d.) Apache Hadoop. https://hadoop.apache.org/. Newman ME (2010) Networks: an introduction. Oxford University Press

Apache Spark (n.d.) Apache spark. https://spark.apache.org/

ArangoDB (n.d.) ArangoDB: the multi-model database. https://www.arangodb.com/

Belchior R, Freitas A (2020) Interoperability in Blockchain systems: a systematic review. IEEE Access 8:174055–174075

Bellman RE (1957) Dynamic programming. Princeton University Press

Bitquery (n.d.) Bitquery: blockchain data infrastructure. https://bitquery.io/

BlackRock (2022) Bitcoin spot ETF

Blei DM (2012) Probabilistic topic models. Commun ACM 55(4):77–84

Blockchair (n.d.) Blockchair: blockchain search engine. https://blockchair.com/

Breiman L (2001) Random forests. Mach Learn 45(1):5–32

Breunig MM, Kriegel HP, Ng RT, Sander J (2000) LOF: identifying density-based local outliers. In: Proceedings of the 2000 ACM SIGMOD international conference on management of data, pp 93–104

Chainalysis (n.d.) Chainalysis: the blockchain analysis company. https://www.chainalysis.com/

Chandola V, Banerjee A, Kumar V (2009) Anomaly detection: a survey. ACM Comput Surv 41(3):1–58

Charnes A, Cooper WW (1961) Management models and industrial applications of linear programming. Wiley

CipherTrace (n.d.) CipherTrace: cryptocurrency intelligence and blockchain analytics. https://ciphertrace.com/

Dijkstra EW (1959) A note on two problems in connexion with graphs. Numer Math 1(1):269–271

Draper NR, Smith H (1998) Applied regression analysis. Wiley

Dune Analytics (n.d.) Dune analytics: the leading platform for Blockchain analytics. https://dune.com/

Elliptic (n.d.) Elliptic: blockchain analytics and compliance. https://www.elliptic.co/

Ester M, Kriegel HP, Sander J, Xu X (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the second international conference on knowledge discovery and data mining, pp 226–231

Etherscan (n.d.) Etherscan: ethereum block explorer. https://etherscan.io/

Fortunato S (2010) Community detection in graphs. Phys Rep 486(3–5):75–174

Gartner (2020) Gartner's analytics and business intelligence summi.

Glassnode. (n.d.). *Glassnode: On-chain analytics*. Retrieved from https://glassnode.com/

Glover F, Kochenberger GA (2003) Handbook of metaheuristics

Goodfellow I, Bengio Y, Courville A (2016) Deep learning. MIT Press

Graphite (n.d.) graphite: open-source monitoring and graphing. https://graphiteapp.org/

Han et al (2011) Data mining: concepts and techniques. Kumar et al (2018). Blockchain technology: a review of supply chain management

Hastie T, Tibshirani R, Friedman J (2009) The elements of statistical learning: data mining, inference, and prediction. Springer

Hoerl AE, Kennard RW (1970) Ridge regression: biased estimation for nonorthogonal problems. Technometrics 12(1):55–67

Hosmer DW, Lemeshow S (2000) Applied logistic regression. Wiley

https://analytics.google.com/analytics/academy/

https://coredevsltd.com/articles/Blockchain-analytics/#:~:text=Blockchain%20analytics%20is%20a%20field%20that%20combines%20the,identifying%2C%20clustering%2C%20and%20interpreting%20data%20from%20Blockchain%20transactions

https://hbr.org/topic/subject/analytics-and-data-science

https://online.hbs.edu/blog/post/importance-of-business-analytics

https://researchmethod.net/descriptive-analytics/

https://www.ibm.com/topics/big-data-analytics
https://www.ibm.com/topics/predictive-analytics
https://www.ibm.com/topics/prescriptive-analytics
https://www.marketingprofs.com/topic/all/marketing-analytics
https://www.thoughtspot.com/data-trends/analytics/diagnostic-analytics

IBM Blockchain (n.d.) IBM Blockchain: transforming supply chain management. https://www.ibm.com/Blockchain/supply-chain

Johnson SC (1967) Hierarchical clustering schemes. Psychometrika 32(3):241–254

JPMorgan Chase (2022) Blockchain and tokenization

Kambatla K, Kollias G, Kumar V, Grama A (2014) Trends in big data analytics. J Parallel Distrib Comput 74(7):2561–2573

Kaufman L, Rousseeuw PJ (1987) Clustering by means of medoids. In: Statistical data analysis based on the L1 norm, pp 405–416

Khan MA, Salah K (2020) Integration of blockchain and IoT: a systematic review. IEEE Access 8:175055–175075

Kim S, Lee J, Kim B (2020) Blockchain analytics: a review of the current state and future directions. Int J Inf Manage 55:102234

Koehn P (2010) Statistical machine translation. Cambridge University Press

Krause M, Tolaymat A (2020) Sustainable blockchain: a survey of energy-efficient consensus algorithms. IEEE Access 8:136055–136075

Kutner MH, Nachtsheim CJ, Neter J (2004) Applied linear regression models. McGraw-Hill

Liaw A, Wiener M (2002) Classification and regression by random forest. R News 2(3):18–22

Liu X, Wang W, Li Z (2020) Blockchain-based data analytics: a survey. IEEE Trans Industr Inf 16(4):2314–2323

Liu FT, Ting KM, Zhou ZH (2008) Isolation forest. In: Proceedings of the 2008 eighth IEEE international conference on data mining, pp 413–422

MacQueen J (1967) Some methods for classification and analysis of multivariate observations. In: Proceedings of the 5th berkeley symposium on mathematical statistics and probability, vol 1, pp 281–297

Manning CD, Raghavan P, Schütze H (2008) Introduction to information retrieval. MIT Press

Manyika J, Chui M, Brown B, Bughin J, Dobbs R, Roxburgh C, Byers AH (2011) Big data: the next frontier for innovation, competition, and productivity. McKinsey Global Institute

Microsoft PowerBI (n.d.) Power BI: business analytics. https://powerbi.microsoft.com/

Mythril (n.d.) Mythril: ethereum smart contract security analysis. https://mythril.io/

Nadeau D, Sekine S (2007) A survey of named entity recognition and classification. J Artif Intell Res 30:1–55

Nakamoto S (2008) Bitcoin: a peer-to-peer electronic cash system

Nansen (n.d.) Nansen: analytics for ethereum. https://www.nansen.ai/

Ng AY, Jordan MI, Weiss Y (2001) On spectral clustering: analysis and an algorithm. In: Advances in neural information processing systems, pp 849–856

Nicolas et al (2018) Smart contract vulnerabilities: a survey

Pang B, Lee L (2008) Opinion mining and sentiment analysis. Found Trends Inf Retr 2(1–2):1–135

Pedregosa F et al (2011) Scikit-learn: machine learning in Python. J Mach Learn Res 12:2825–2830

Perozzi B, Al-Rfou R, Skiena S (2014) DeepWalk: online learning of social representations. In: Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining, pp 701–710

Pham T, Lee S (2020) Fraud detection in blockchain-based systems using machine learning techniques. IEEE Access 8:103066–103075

Pham T, Lee S (2020) Computational complexity of machine learning algorithms in blockchain-based systems. IEEE Access 8:103096–103105

Powell WB (2011) Approximate dynamic programming: solving the curses of dimensionality. Wiley

Provost F, Fawcett T (2013) Data science for business: what you need to know about data mining and data-analytic thinking. O’Reilly Media

Quinlan JR (1986) Induction of decision trees. Mach Learn 1(1):81–106

Rumelhart DE, Hinton GE, Williams RJ (1986) Learning internal representations by error propagation. In: Parallel distributed processing: explorations in the microstructure of cognition, vol 1, pp 318–362

Schölkopf B, Platt JC, Shawe-Taylor J, Smola AJ, Williamson RC (2001) Estimating the support of a high-dimensional distribution. Neural Comput 13(7):1443–1471

Singh S, Singh N (2020) Data quality issues in blockchain-based systems. IEEE Access 8:103086–103095

Singh S, Singh N (2020) Blockchain and machine learning: a systematic review. IEEE Access 8:103046–103065

Slither (n.d.) Slither: static analysis for smart contracts. https://github.com/TrailOfBits/slither

Swan M (2015) Blockchain: blueprint for a new economy. O'Reilly Media

Tableau (n.d.) Tableau: data visualization. https://www.tableau.com/

Tapscott D, Tapscott A (2016) Blockchain revolution: how the technology behind bitcoin is changing money, business, and the world

Tibshirani R (1996) Regression shrinkage and selection via the lasso. J Roy Stat Soc B 58(1):267–288

Wang W, Wang D (2020) NFTs: a survey of non-fungible tokens. IEEE Access 8:155055–155075

Wang W, Wang D (2020) Scalable Blockchain analytics: a survey. IEEE Access 8:143055–143075

West DB (2001) Introduction to graph theory. Prentice Hall. Wasserman, Faust (1994) Social network analysis: methods and applications

Witten et al (2016) Data mining: practical machine learning tools and techniques. Li et al (2019) Blockchain-based data management for big data

Wolsey LA (1998) Integer programming. Wiley

Zhang et al (2019) Blockchain-based data analytics for supply chain management

Zhang Y, Wen F (2020) Blockchain interoperability: a survey. IEEE Access 8:185055–185075

Zheng Z, Xie S (2020) Decentralized finance (DeFi): a systematic review. IEEE Access 8:143055–143075

Chapter 13
Blockchain Application in Healthcare

Elham Ghanbari, Sara Najafzadeh, and Fatemeh Nasiri

Abstract In healthcare, blockchains have emerged as a probable solution for various challenges especially those associated with data management. This state-of-the-art technology supports the safe storage and dissemination of sensitive medical information through an unbroken, secure platform. Ensuring data transparency and traceability allows blockchains to improve processes such as drug Supply Chain Management (SCM) and make health devices safer than before. Moreover, this technology can be integrated into mobile-assistive devices used in monitoring patients' conditions and transferring their health data via smart gadgets thereby improving the quality of healthcare delivered to them. Further still, there is more privacy and security in data across the use of blockchain thus leading to major advancements in digital medicine. While this chapter introduces blockchain in healthcare it outlines several applications that utilize blockchain to enhance the security and privacy of health data, including Electronic Health Records (EHR), Remote Patient Monitoring (RPM), and clinical trials. The subsequent sections will discuss the pros and cons of using blockchain. Some benefits of adopting this type of decentralized network include increasing the reliability of records through the elimination of human errors.

Keywords Healthcare · Blockchain · Electronic health records (EHR) · Remote patient monitoring (RPM)

E. Ghanbari (✉) · S. Najafzadeh · F. Nasiri
Department of Computer Engineering, YI, Islamic Azad University, Tehran, Iran
e-mail: el.ghanbari@iau.ac.ir

S. Najafzadeh
e-mail: sa.najafzadeh@iau.ac.ir

F. Nasiri
e-mail: fa.nasiri@iau.ac.ir

K. Ahmad et al. (eds.), *Fostering Machine Learning and IoT for Blockchain Technology*,
Transactions on Computer Systems and Networks,
https://doi.org/10.1007/978-981-96-4078-2_13

13.1 Introduction to Blockchain in Healthcare

A brief definition of blockchain and its operational principles will be discussed in this section, followed by a demonstration of the importance of using blockchain in the healthcare industry.

13.1.1 Blockchain Technology and Fundamental Principles

Blockchain is a decentralized, digital distributed ledger for storing all varieties of information without a central authority. Unlike traditional databases or spreadsheets, a blockchain does not store data in central servers. It distributes not just one but multiple copies of data to avoid any misuse, failure, or other malpractice. In a blockchain network, data is ordered into a sequence of "blocks" in some order. A block is constructed over a preceding block; thus, it is in chain form. Blocks are linked by putting a specific piece of information within a block that, in turn, is linked to the preceding one. In this way, a "chain" is formed, i.e., the blocks are chained together. Now, take a digital ledger where entries—referred to as blocks—are secured and interconnected cryptographically. Each of these blocks includes a cryptographic hash of the previous block, some timestamp forms, and transactions that have occurred within the piece of information. This makes an unbreakable chain of blocks, considering that all the information cannot be changed or deleted without knocking out all the rest. This inherent immutability immediately makes transactions on a blockchain irrevocable.

Blockchain works on the principles of decentralization, transparency, and immutability. The data stored is distributed across several nodes in the blockchain network without any point of control or failure. All transactions are kept transparent and accessible to other participants in the network. The cryptographic linking of blocks secures an extremely high level of tamper resistance and immutability of historical records. Finally, blockchain allows one to store information in a secure and decentralized manner, without the existence of a trusted party. Its novelty in application means it can perform a wide array of applications well beyond financial transactions—all through the features of decentralization and transparency.

Figure 13.1 demonstrates the processes of a blockchain network. At its core, blockchain's decentralized, peer-to-peer architecture and its distributed ledger facilitate secure distributed learning. This is achieved by recording local gradients on the blockchain, enabling federated learning across heterogeneous datasets while maintaining data privacy (Shinde et al. 2024).

1. A transaction request is initiated by a user.
2. For every transaction, a block is created which will hold the transactional data.
3. A request for the transaction is broadcast to each node in the peer-to-peer network, constituting the blockchain network.

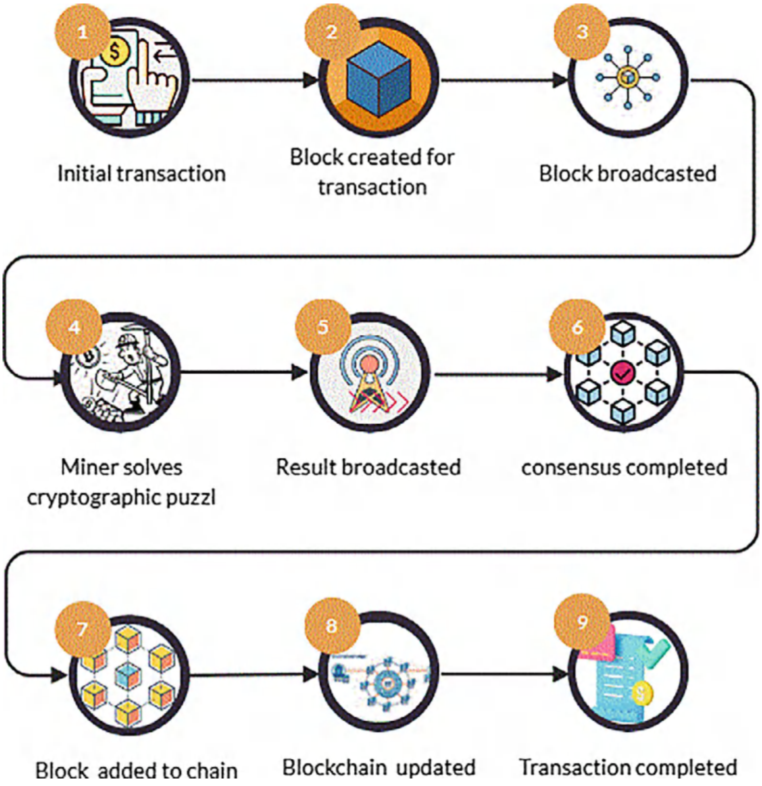


Fig. 13.1 Processes of blockchain

4. Next, using computing power, they try to solve a cryptographic puzzle in order to add the new block to the blockchain ledger. Committing this block to the blockchain—essentially putting it to the very end of the ledger—uses running consensus algorithms, namely, involving miners in solving complex cryptographic puzzles for a given block and sharing the results with miners. Miners subsequently race to unlock a new block addition to the blockchain ledger by utilizing their computational resources to solve the cryptographic puzzle.
5. The first miner to resolve the puzzle broadcasts the new validated block to all nodes in the network.
6. These consensus algorithms are some of the essential enabling technologies in AI-empowered healthcare to achieve collective decision-making and ensure secure yet auditable medical records. Since all blocks are cryptographically linked, once a block has been added to the chain, it becomes immutable and thus can be easily traced.
7. The miner who first solves the puzzle receives the legal right to mine transaction blocks into the existing chain of blocks and replicates the new chain across all nodes.

8. Other nodes receive the new block, verify that all transactions are valid, and then add it to their copy of the blockchain ledger.
9. Once a transaction is added to the blockchain, it is considered complete and irreversibly confirmed by the entire network. The highest degree of availability and transparency in the blockchain ledger is achieved with the help of many replicas within each node of a network. The copies of the ledger are identical, thus ensuring data integrity and accessibility.

13.1.2 The Impact of Blockchain in the Healthcare

Blockchains can be viewed as a decentralized, distributed ledger that protects electronic forms of information distributed among nodes on a computer network. This technology involves gathering information and forming blocks that create a blockchain, or a chain of data. One of the benefits of blockchain is a high level of security, which ensures that different types of data, depending on the network, technology used, or sector, cannot be altered, hacked, or falsified. The main application of blockchains is to store digital information securely without the risk of erasure, leakage, or hacking; therefore, this technology has been used in plenty of industries and businesses like the healthcare area creating huge changes in their key sectors. Based on the research conducted, many health researchers believe that some forms of blockchains will happen in the medical system in the future (Shinde et al. 2024; Bazel et al. 2023).

Health protection has been an important priority for humanity, yet the current healthcare and treatment systems are often slow, complicated, and costly. These systems are prone to human errors and negligence, leading to inefficiencies and waste. They are also easily exposed to human errors and negligence. However, one of the ways to optimize the performance of this field is the use of blockchain. In other words, one of the most important roles of blockchain in healthcare is its ability to create a secure and integrated database of individuals. As a result, we will no longer worry about the problem of disruption and manipulation of information, and we owe this to blockchain technology. This technology can be used in a practical way to record and track medical information, medications, referred medical centers, and even the reasons for many patients' referrals (Isravel et al. 2023). However, blockchains offer a promising approach to optimize healthcare performance. They can revolutionize the management of health data, therefore, the current systems can be transformed into new blockchain platforms to manage patient information and records. Blockchain in the healthcare industry can act like a secure database that sorts and saves patient information and records immutably, including disease medical history, medications, medical centers visited, and the reason for them. Also, blockchain in healthcare can enhance the patient treatment process and the interaction between patients and treatment staff. In brief, using blockchain reduces treatment costs and improves patient outcomes.

Table 13.1 Comparison between blockchain-based systems and traditional one

Factors	Traditional	Blockchain based
Storage	Centralized	Decentralized
Availability	Delayed	Up to date
Integrity	Less	More
Security	Less	More
Immutability	No	Yes
Access	Internal	Anywhere

The growth of blockchain in digital healthcare has happened due to its user-friendly features, including security enhancement and seamless data sharing, as maintaining the privacy and security of health data is very challenging. Therefore, blockchain has the potential to contribute to digital healthcare systems by improving security and speeding up data processing between providers.

The integration of patient records, drug development, and supply chain; prevention of fraud in pharmaceuticals, medical research, insurance fraud; management of complaints and medical bills; access; and transparency are among the main applications of blockchain in healthcare. The important parts of blockchain in healthcare are data storage and management, creating electronic files, recording information, and then tracking each patient's health, just to name a few. The high sensitivity of the fields of health, healthcare, and treatment requires that recorded information be properly protected and that access to it is controlled meticulously. The role of blockchain in healthcare is vast and promising, improving everything from secure record-keeping to SCM.

In the following, the comparison between blockchain-based healthcare systems with traditional ones is conducted in Table 13.1. In (Singh et al. 2023a), key factors are identified for comparison of blockchain-based healthcare management systems against traditional ones. This comparison verifies that blockchain-based systems have the potential to bring significant improvements and benefits to society in contrast to traditional ones.

13.1.3 Data Security, Interoperability, and Transparency in Healthcare

The central application of blockchain in healthcare is to store and manage health data. For instance, blockchain in EHRs records patients' information and tracks each patient's health. The high sensitivity of the healthcare and medical field requires that the registered information be properly protected, and the level of access to it be controlled meticulously.

Blockchain technology in healthcare, with its two key components of cryptography and immutability, provides the necessary security and adds transparency to the activities performed in the system. This, in turn, brings an efficient system to both healthcare providers and patients. With this technology, healthcare systems can record digital transactions in blocks and create an immutable distributed ledger without any alteration once recorded and published.

Blockchain technology is also decentralized, which means the dependence on a trusted third party to simplify transactions is omitted. It actives customers and other blockchain users to take the data ownership they enter into the network. As a result, each user has their own version of the information, and the validation of updates to a patient's file, even though all copies are stored on different computers, is verified by the blockchain's peer-to-peer architecture. So, blockchain can be used as a powerful tool for preserving health data, which is why it has capabilities such as secure, decentralized, and transparent record-keeping capabilities. The encryption feature of blockchain safeguards patient privacy, while its decentralized one enables secure and efficient information sharing among different users and providers.

In addition to simplifying the process of recording and sharing information, healthcare blockchains provide each patient with easier and more secure access to medical records. One of the excellent methods for inspecting the pharmaceutical industry at all stages of the SCM is blockchain due to its transparency and reliability. This technology can provide an effective solution for addressing counterfeiting and illegal activities related to the pharmaceutical mafia as well as counterfeit drugs. If blockchain technology in healthcare is combined with the Internet of Things (IoT), it is able to accurately track and monitor various conditions of drugs, such as temperature and humidity during transportation or storage (Singh et al. 2023a). Another blockchain application in healthcare is tackling common crimes and illegal activities in patient insurance.

The non-hackable, non-editable, and non-deletable structure of blockchain in healthcare allows medical centers to link with insurance organizations and share medical records and patient histories with the insurance company, enabling them to prevent some common illegal activities and counterfeiting, such as illegal services like charging for surgeries that were not performed. Blockchain technology in healthcare, through powerful and efficient protocols that enhance traditional security and speed, will soon become the foundation for many industries that still struggle with outdated methods, and will be widely accepted by various administrative and organizational systems.

13.2 Technical Overview of Blockchain in Healthcare

The healthcare industry has been transformed by blockchain technology through its ability to simplify data management and operational processes A scientific overview of blockchain in healthcare will be presented in this section to explore core functionalities and types of blockchain networks. Blockchains, shown in Fig. 13.2, are generally divided into two categories including permissionless blockchains and permissioned blockchains.

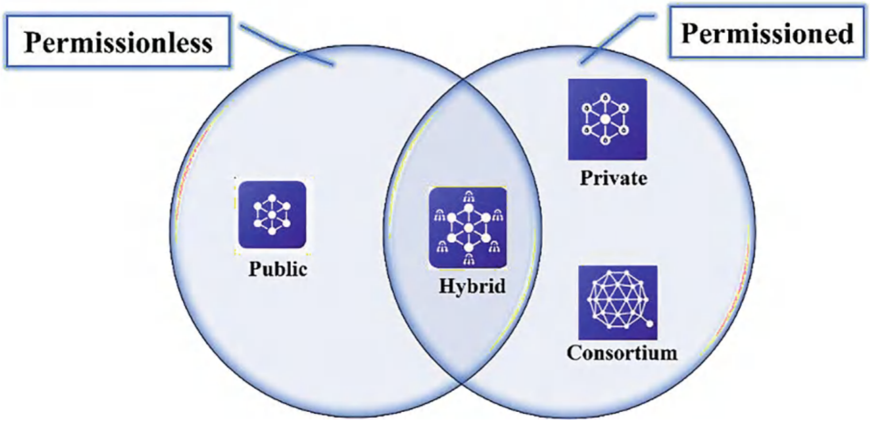


Fig. 13.2 Blockchain categories

One of the important permissionless blockchains is called the public blockchain which does not require any permission. In contrast, private and consortium blockchains are in the category of licensed permissioned ones. Finally, a hybrid blockchain is somewhere in between, which combines some features of both permissionless and permissioned blockchains.

Public Blockchain: Access to this type of blockchain doesn't need permission, this means that allows anyone to join. All nodes in these blockchains have the same privileges to create new blocks. Public blockchains have the highest degree of transparency, trust, and security, nevertheless, the transaction processing speed is low. Also, the busy networks suffer from the "scalability" problem. Some public blockchains including Bitcoin blockchain, Ethereum blockchain, and other digital currencies are mostly used in cryptocurrency projects like mining and exchanging.

Private Blockchain: The upstream power or group decides who is the node or the network operator. Private blockchains are somewhat decentralized because of the restriction of access to these blockchains. The private blockchain is not accessible to everyone. This blockchain is mostly used by companies and organizations and only specific nodes are allowed to join the network. Determining the level of security, access, and activities of each node is controlled by a department. Therefore, private blockchains are likely to be implemented on a smaller scale and are mostly used for applications, namely, SCM and digital identity. Corda and Hyperledger are some instances of private blockchains.

Consortium Blockchain: Unlike private blockchains, this form of blockchain belongs to the permissioned category where different users can work together. Although consortium blockchains are so complex and more decentralized than private blockchains, they can improve security. These blockchains have a suitable balance between transparency and efficiency, but they also have disadvantages such as the lack of efficient rules and standards. Quorum and Multichain are some famous instances of this type of blockchain.

Table 13.2 Comparison of blockchain types

Factor	Public	Private	Consortium	Hybrid
Efficiency	Low	High	High	Low to high
Authorization	Public	Public or restricted	Public or restricted	Public and restricted
Decentralization	High	Low	Medium	Medium to high
Data privacy	Low	High	Medium	Medium to high
Scalability	Low	High	High	Medium to high
Cost	Low	High	Medium	Medium
Governance	Decentralized	Centralized	Shared	Combination
Examples	Bitcoin, Ethereum	Hyperledger, Corda	Quorum, Multichain	Ripple, Dragonchain

Hybrid Blockchain: Both public and private blockchains integrated to create a balance between accessibility and control are named hybrid blockchains. One of the essential factors in this type is that organizers can choose which part is accessible by the user and which part is not accessible. So, accessibility to the public part doesn't need permission, however, the permission to access certain information for each specific user is vital. Some examples of this category of blockchain are Ripple and Dragonchain.

In Table 13.2, these types of blockchains are compared in different aspects. One of the important problems in healthcare is "Which blockchain can be used in this domain?" The choice of different blockchains for the healthcare domain is dependent on goals and applications. That is because creating a resistant system for protecting and managing data in different areas, namely, patient, clinical, and doctors is one of the aims of this domain, and blockchain can provide it. Also, by using blockchain, a safe environment for multiple transactions is created for this domain. Take, for instance, smart contracts in healthcare are used to show the accountability and transparency of each user and reduce errors (Ghosh et al. 2023).

13.2.1 Different Blockchains for Healthcare Applications

According to the previous section, there are multiple instances of implemented blockchains in distinct types. As we mentioned, which blockchain is useful for the healthcare domain is an important issue. To address this issue, a comparison of well-known implemented blockchains (Singh et al. 2023a) is shown in Table 13.3. This comparison would suggest multiple important factors for choosing blockchain in a particular field, such as healthcare. Some abbreviations in this table are used, which are introduced in the below table. Nevertheless, a lot of papers have been published in this field (Agbo et al. 2019), but it is in the early stages. Generally, blockchains in healthcare are known as a recent field. Some blockchain applications in health

Table 13.3 The comparison among different popular blockchain developments

Factor	Ethereum	Hyperledger	Ripple	Quorum
Blockchain type	Public	Private	Hybrid	Consortium
Applications	Crypto-currency, smart contract	Smart contract	Crypto-currency	Smart contract
Programming language	Solidity, LLL	Golang, Java	C + +	Solidity
Consensus	Proof of work, proof of stake	Practical byzantine fault tolerance	Probability Voting	Raft, byzantine fault tolerance
Throughput	~ 200 tps	> 2000 tps	> 1500 tps	~ few hundred
Operation mode	Trustless	Validator node	Trustless	Trustless
Authentication	Digital signature	Enrollment certificate	Biometric	Password

include Healthchain built on Hyperledger (Ahram et al. 2017), and Acile based on the Ethereum blockchain (Dagher et al. 2018).

13.3 Applications of Blockchain in Healthcare

Blockchains have a huge effect on the improvement of healthcare systems by providing a safe way to manage medical data (Isravel et al. 2023; Singh et al. 2023a, 2023b; Prokofieva and Miah 2019; McGhin et al. 2019). Some blockchain applications in healthcare (Fig. 13.3) are explained in this section.

13.3.1 Electronic Health Record

Converting paper records containing multiple medical data to electronic health records is one of the important areas improved by blockchain. Unlike paper records, EHRs are digital copies and always updated. They allow authorized users to access data securely and immediately (Agbo et al. 2019). Also, they can let users share data quickly. Before using blockchain in EHRs, patients' data was scattered across various healthcare providers even within digital systems, so data was unreadable and inaccessible (Isravel et al. 2023; Singh et al. 2023a; Agbo et al. 2019). Nowadays, EHRs are developed with blockchain technology and solve problems by guaranteeing that data is updated accessible, and securely preserved forever, FHIRChain (Zhang et al. 2018), MedBlock (Fan et al. 2018), just to name a few.

The purpose of EHR systems is to share health records on several platforms safely, but despite the many advantages that these systems have, there are still many

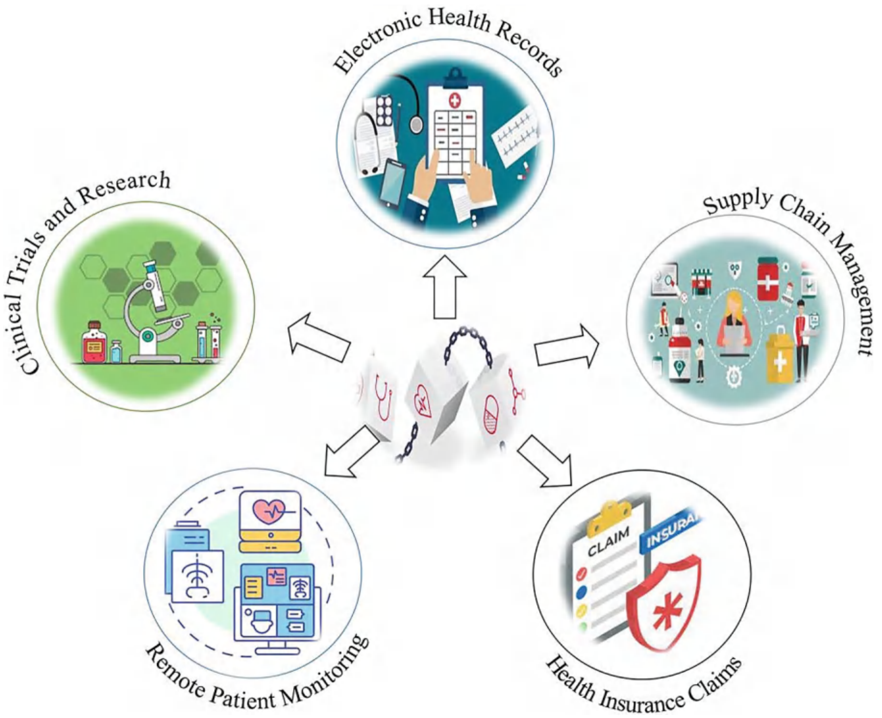

Electronic Health Records
Clinical Trials and Research
Supply Chain Management
Remote Patient Monitoring
Health Insurance Claims
CLAIM
INSURA

Fig. 13.3 Real-world applications of blockchain in healthcare

disadvantages for them, including data asymmetry and interoperability issues (Isravel et al. 2023). Interoperability in EHRs for sharing data is very important and causes error and cost reduction by simplifying processes and securing the exchange of data. Researchers provide this interoperability in EHRs by using blockchain technology (Villarreal et al. 2023). Moreover, Personal Health Records (PHRs) are blockchain systems for managing patients' data and patients can use PHRs to monitor how their data is shared and used (Isravel et al. 2023). Thus, blockchain can be applied in EHRs in order to improve data management. The process of this technique is described in four simple steps:

1. During the examination of a patient, the doctor records the patient's report. To create a transaction, firstly, the patient's data is transferred to the blockchain using APIs.
2. Transactions on the blockchain are authenticated by receiving a distinct public key.
3. Healthcare users can utilize APIs with the patient's decryption key to retrieve queries.
4. Patients can authorize healthcare users to decrypt their data by using their private key which acts as a password. For anyone lacking this private key, the information stays encrypted and inaccessible.

Nowadays, a reliable approach for sharing health data is so vital because of the growth of EHRs. Blockchain which provides a decentralized approach can be used in EHRs to preserve data from unauthorized access and manipulation.

Some benefits of blockchain in healthcare are mentioned in the following: 1. Decentralization: Because blockchain systems do not use a primary authority to keep and share data. 2. Immutability: This means that it is not possible to change or delete data in a block system. 3. Transparency: In blockchain systems, a transparent environment can be provided for data sharing with the possibility of tracking and monitoring transactions.

As mentioned, secure storage and sharing of EHRs is important to protect patient privacy. Although blockchain systems provide a secure way to store EHRs, they do not protect against hacking. Primary security concerns include poor detection during phishing attacks which can occur for a variety of reasons, such as using weak passwords. So, the consideration of both the technology and the human factors for improving the security level of blockchain is important, namely, having strong data management policies and technical rules (Urwongse and Culver 2020).

13.3.2 Drug Supply Chain Management

One of the essential healthcare applications based on blockchain is related to SCM, including medication (Singh et al. 2023a) due to its rising complexity (Siyal et al. 2019).

Blockchain technology can provide a reliable solution aimed at improving data transparency and product traceability to monitor supply chain activities and ensure the distribution of medicines to authorized users (Siyal et al. 2019; Angraal et al. 2017). Verification of a medical product's authenticity and quality is very crucial because the delivery of such drugs can have severe consequences for the patients. Today, several companies have been working on developing blockchain-based solutions for detecting drug fraud such as Hyperledger Fabric (Androulaki et al., 2018) which records each transaction of prescription drugs through blockchain. Thereby, it connects manufacturers, distributors, doctors, patients, and pharmacists. Hence, they can verify the authenticity of prescriptions (Engelhardt 2017; Rejeb et al. 2021). One of the other systems is Modum.io AG, which applies blockchain technology to have pharmaceutical product record immutability. In this system, data, while accurate, is also kept unchanged (Hellani et al. 2021).

The process of an SCM system in blockchain starts with the identification of new drugs, followed by the creation of a block to store the clinical trial data regarding those drugs. For drug production, the patent must be moved to the manufacturing places. Then drugs, once produced, are kept in warehouses. After this, the drugs are distributed from the warehouse to various pharmacies, although all the information is maintained in the blockchain to avoid counterfeiting. Finally, the distributors' information is maintained, and patients can trace back the whole process to validate its authenticity by checking the data through the SCM in the blockchain (Ghosh et al. 2023; Agbo et al. 2019).

13.3.3 Remote Patient Monitoring

Another part of modern healthcare is PRM, which allows medical staff to monitor patients' problems, such as diabetes or blood pressure, while they are away from the hospital. The technology allows patients to measure their health status from their homes with many instruments. IoBHealth can also be depicted as a data-flowing approach bound by the IoT, and blockchain for healthcare to make EHR access more available and less volatile This allows healthcare users to monitor the patient's condition from any places, enabling necessary and timely interventions. When a patient is using one of these physiological monitoring devices, it takes the appropriate health readings and forwards them to healthcare professionals who are enabled to monitor trends in their patient's health conditions, identify condition changes (when applicable), get real-time alerts for when patients may be about to deteriorate so they can initiate swift action and hence improve outcomes. Thus, this technology collects biomedical data from patients and transmits it to healthcare centers using devices with the support of IoT (Singh et al. 2023b; Boikanyo et al. 2023).

In the remote RPM, blockchain has changed how healthcare providers handle patient data. Healthcare providers can collect, analyze, and store biomedical data rapidly without security issues as secure logging would license to integrate IOT devices with blockchain expediently. Figure 13.4 shows the RPM system based on blockchain. Blockchain is decentralized, where all the patient information obtained from different hospitals is placed on the blockchain; thus, this eliminates fears of centralization like single-point failure, data manipulation, and privacy. This approach additionally allows healthcare providers to securely exchange data among attendees, which in turn enhances teamwork and patient care.

Several researches have depicted the effectiveness of blockchain in RPM (Agbo et al. 2019; Ben Fekih and Lahami 2020). For example, the smart contracts on Ethereum provide automated interventions in real-time patient monitoring applications, ensuring that healthcare providers respond immediately to changes in patient conditions (Ben Fekih and Lahami 2020; Griggs et al. 2018). Blockchain technology is utilized in mobile-assisted devices to monitor diabetic patients, enabling the transmission of data through smartphones via the Hyperledger Fabric platform that is used for blockchain-based applications, including data collection and sharing among healthcare organizers (Singh et al. 2023a). Mobile devices are utilized for the transmission of data to blockchain applications, namely, the Smart Medical Device for Diabetes (SMEDA) (Makroum et al. 2022). A patient-centric agent (PCA) is another platform that offers end-to-end data security for patient monitoring (Agbo et al. 2019; Uddin et al. 2018).

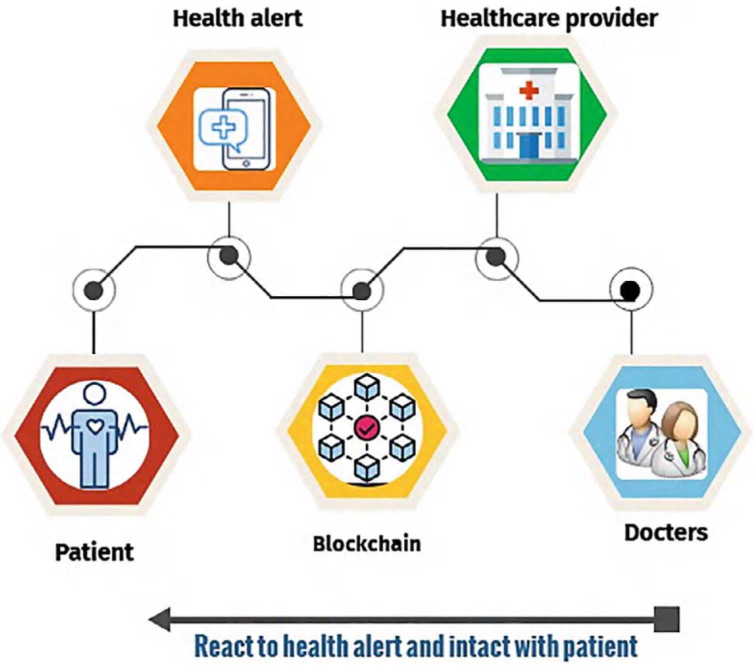


Fig. 13.4 RPM system

13.3.4 Clinical Trials

One essential component in the world of healthcare is clinical trials, thus, call for intense and comprehensive monitoring at each stage, especially data collection, trial mechanisms, monitoring, and data management. It is an examination or period of trial in one's health, with several interventions, experimentation or investigation, measures, diagnostic tests, prevention, devices, or treatment strategies. The multiple parties involved, coupled with the trust that must be maintained among them, make this process an expensive one (Urwongse and Culver 2020). Pharmaceutical and device trials aim to commercialize new drugs and conduct controlled designs, as well as protocols that define prescribed measures. All aspects in which stakeholders comply with the provisions of the competent regulatory agent as regards such a repetitive approach as both the enhancement of the scientific base and ethical safeguards of human subjects (Urwongse and Culver 2020).

Yet, there are a lot of issues that clinical trials face such as data privacy, sharing the data, record-keeping, and patient enrollment. Challenges like these can affect the accuracy and robustness of clinical research. It is one ingenious tool that can be used to overcome the challenges faced in clinical trials—Blockchain Technology. The integration of blockchain with Artificial Intelligence (AI) is believed to revolutionize the healthcare industry soon (Siyal et al. 2019). Blockchain allows researchers track and coordinate their data transparently, reducing the risk of exploitation. It also has the potential to address reproducibility concerns by allowing other parties immediate access to peer-reviewed protocols as soon as they are reported on any electronic medium using decentralized ledger technology. Unlike human beings, AI can help increase the effectiveness and accuracy of health research by analyzing data, identifying patterns, and predicting outcomes at scale. According to Siyal et al. (2019), combining blockchain-based solutions with AI would result in improved clinical trials and thereby help deliver personalized therapy strategies. Several applications of blockchain in this field have been presented by various researchers, some of which will be reviewed below.

A permissioned Ethereum blockchain with a clinic-based data management system is presented in Nugent et al. (2016) to address the problem of patient enrollment in clinical trials. Because Ethereum provides quicker transactions compared to some other blockchains, thus the possibility of using Ethereum smart contracts for increasing transparency in clinical trials could be increased. The problem of the publication of incorrect results can be improved by cryptographic assurances enabled by blockchain protocols (Bennacer et al. 2023). Another research in this field has presented a blockchain framework that can provide informed patient consent and participation in clinical trials (Benchoufi et al. 2017).

Blockchains can be used in genetics. This issue is vital because studying genes can take an effective step in identifying and preventing diseases caused by genetic disorders. Blockchain provides a decentralized framework for keeping and sharing DNA, without worrying about privacy and data ownership. Also, due to being secure and immutable, the data can be protected against hacking damage (Isravel et al. 2023).

As shown, blockchain has a great potential for change in the field of clinical trials. Blockchains can be utilized as an assist to eliminate fake data or exclude undesirable results during clinical trials. By enabling encryption, blockchain allows researchers to use patients' data in medical research without worrying about privacy. In addition, the integrity of clinical study data is guaranteed due to the immutability of the blockchain, and also its transparency facilitates the replication of clinical studies. These factors lead to the widespread use of blockchain in health research (Ghosh et al. 2023).

13.3.5 Health Insurance Claims

Health insurance tries to help people against high medical costs. Claims are received by insurance companies and the amount paid for those claims is determined by a process called claims handling. The insurer may accept the claim for payment or may deny it. Insurers can make their own claims, these claims can be accepted or rejected. Despite administrative automation in processing claims, complex claims are still prone to error and fraud (Agbo et al. 2019).

The presence of important aspects of blockchain technology in terms of transparency, decentralization, and immutability can be useful in healthcare claim processing, as many researchers verify that blockchain will revolutionize the way of addressing insurance claims in healthcare (Singh et al. 2023b). MIStore is one of these researches that was developed based on the Ethereum blockchain for health insurance systems (Zhou et al. 2018). In addition, Pokitdok has also announced its interest in working with Intel to develop a blockchain application to speed up the process of health insurance claims in healthcare (Ghosh et al. 2023; Agbo et al. 2019).

Since blockchains are suitable to use in the management of patient records due to their transparency, safety, and traceability, it will be possible for different organizations to access the patient's medical information. This process will be very useful in insurance claims (Ben Fekih and Lahami 2020). Also, blockchain smart contracts are very effective at automating processes associated with the above areas by reducing errors and fraud. That is because they can make claims visible to suppliers and insurers. In this technique, all parties can be informed about any changes in laws or regulations (Isravel et al. 2023).

13.4 Benefits and Challenges of Blockchain in Healthcare

There are many blockchain applications in healthcare. Just as with all the other domains, there exist pros and cons of using blockchain in healthcare, and a few of the major benefits and challenges of blockchain in healthcare will be explained in this section.

13.4.1 Benefits of Blockchain in Healthcare

Blockchains can change healthcare forever by enabling greater collaboration between industry partners. As medicine progresses with increasingly rapid advances, so too does reliance on technological methods for both treatment and research. This section indicates the advantages of blockchain in healthcare.

13.4.1.1 Enhanced Data Security and Privacy Protection

Other notable applications related to the blockchain provide both privacy and security for patients' medical records by restricting access through blockchain solutions where privacy could be achieved through encryption and decentralization of medical data integrity using solutions of blockchain technologies that have encryption (Shinde et al. 2024; Atadoga et al. 2024; Rathore et al. 2020).

The most significant application of blockchain in healthcare is its capacity to develop a secure, peer-to-peer database. More critically, its immutability takes care of data loss risks; thousands of patients' medical recordings and tracking data, in general terms, are all securely recorded on the blockchain. It also includes a higher level of security of information transfer, as ensured by the decentralized blockchain, which eliminates office expenses related to managing a centralized system as in the traditional centralized databases (Isravel et al. 2023). Blockchains are immune to technical mistakes or cyber-attacks, because of their decentralized nature, so they can be used as a valuable tool for hospitals often targeted by such threats.

13.4.1.2 Improved Interoperability and Collaboration

Another benefit of blockchain systems is their ability to create interoperability and collaboration between clinics, hospitals, and other medical institutions. Technical and technological differences between different medical centers often make it difficult to share documents and information, but blockchain technology can solve this problem by designing an integrated system where patients' medical information is stored. As a result, instead of different people and centers interacting with each other's internal systems, they can use a single, shared database (Bazel et al. 2023; Singh et al. 2023a; Molli 2023).

13.4.1.3 Enhanced Efficiency and Reduced Cost

Blockchains in healthcare simplify processes and significantly reduce administrative costs. By using blockchain's decentralized and automated identity, healthcare organizations can eliminate the additional costs associated with processing various transactions, including international ones, and achieve accurate financial management. Smart contracts, a key feature of blockchain, automatize the administration work conducted within insurance claim coverage, invoicing, or settlement otherwise in paying detail. This is done without banners, meaning that these administrative processes will require minimal human intervention in the three functions thus reducing administrative load and cost.

Additionally, blockchain eliminates the need for intermediaries in healthcare. Providing a basis for direct transactions among involved parties would lead to speedier procedures and lower associated costs, adding up to healthcare that's efficient and cost-effective. This technology's digital office system dramatically reduces the number of physical documents and administrative costs. The more significant SCM transparency provided by blockchain reduces inefficiencies, minimizes delays, and reduces tracking and authentication costs of medical equipment.

13.4.1.4 Enhanced Access and Transparency

Blockchain systems provide patients with a higher level of access to their own data, simplifying the process of information sharing and increasing the accuracy of information. This is achieved through a mechanism that allows patients to approve changes to their own records, providing a secure layer against human error and intentional misinformation.

The blockchain system verifies the availability of data by replicating it across several nodes in distinctive systems such as healthcare, thus increasing durability against data loss, corruption, and other security breaches. Blockchain technology enables trust in distributed healthcare applications. Thus, such transparency and trust can foster closer relationships between healthcare providers and patients to improve health systems (Benchoufi et al. 2017; Atadoga et al. 2024; Isravel et al. 2023).

13.4.1.5 Centralized Management

The incorporation of blockchain solutions results in the abolition of third parties' dependency and direct parties' interaction. Blockchain technology saves all the data in a centralized form, and it can be viewed only by the concerned parties (Singh et al. 2023b; Rathore et al. 2020; David et al. 2023). It is approached in decentralization conditions, decreasing the possibilities of single points of failure, increasing data integrity, and making it even more secure.

Additionally, health information based on blockchain technology is patient centered because it allows the patient to be in control of the information, thus embedding transparency in the patient–doctor relationship. The use of blockchain, supporting immutability and cryptography, empowers the health industry to become a much more secure, efficient, and dependable management of data (Singh et al. 2023b; David et al. 2023). Its decentralization means that no central authority does not exist; hence, the greatly reduced administrative cost and less cumbersome processes. It guarantees transparency and traceability of all transactions. It, therefore, becomes useful in auditing and related purposes in conformance. The parties involved remove fraud through transparency. It also assures parties in healthcare that the data being used is trusted because of the transparency it accords.

13.4.1.6 Improved Data Quality and Error Reduction

Blockchain technology considerably increases data quality in healthcare while, at the same time, reducing the count of errors. The decentralized nature of its ledger system helps in phrasing medical crimes like duplication of records, errors, and data manipulations through a very simple, user-friendly interface. Its decentralized ledger system pinpoints and eliminates duplicate records, leaving a single source of truth that is clear and precise visibility for any healthcare organization.

The level of immutability in the blockchain is such that once information is recorded, it is not able to be edited or deleted without any traces. This maintains the information integrity contained within medical records and greatly reduces the related risks of error and fraud. Real-time synchronization of information across all interacting network nodes regarding data means that any updating or changing of the information, at the very instance it occurs, automatically reflects in the information available to all relevant healthcare providers.

Moreover, its clear ledger tracks the entire audit trail for all transactions and data inputs, assuring the audit process and legality. This also allows for greater auditability, which helps promptly spot and correct discrepancies, thereby lowering the chances of mistakes.

Besides, blockchain enables easy data sharing among diverse healthcare providers, ensuring that access to information by all these entities is not only consistent and accurate but also free from errors emanating from data silos and inconsistent systems. Ensuring the administrative burden on healthcare workers is reduced through the automation of both validation processes and data management processes, blockchain has presented a great opportunity for workers to worry less about manual entry and error-repairing to provide more care to their patients. In brief, blockchain-based systems in healthcare greatly enhance the quality of data and reduce errors, hence providing a transparent, secure, and efficient system for managing medical records and transactions.

13.4.2 Challenges and Considerations

The healthcare industry is supposed to be transformed by blockchains, as they offer good solutions to healthcare operational obstacles, such as the security, privacy, authenticity, and compatibility of health data, while also meeting the demand for simultaneous updates and availability. However, blockchain technology is not without its limitations. Despite its potential benefits, the growth of blockchain in healthcare is supposed to pose severe research challenges that require additional study. In the following, these obstacles are mentioned.

13.4.2.1 Security and Privacy

Encryption offers a level of security and protection; however, healthcare systems acknowledge that the availability of an encrypted database still poses a major concern. In the blockchain domain, it is vital to establish strong access measures to safeguard sensitive healthcare data from unauthorized entry (Wenhua et al. 2023).

Acquiring health information from varied sources leads to many unintended breaches in issues concerning protection, privacy, and security. Taming such breaches, therefore, calls for the implication of offering the specified sensitive information with a deep sense of access control analysis and comprehensive thought. It calls for a proactive access control system where only the people or institutions with an obligation before the information carry the day (Isravel et al. 2023; Singh et al. 2023a).

A major weakness in implementing blockchain in healthcare is the insecurity of cyber-attacks on blockchain applications. Because blockchain applications are web based and are accessed over the web, they are undefended against multiple attacks, namely, tampering. These attacks compromise the blockchains' integrity, leading to significant consequences. For instance, the 51% attack, referred to as the majority attack (Agbo et al. 2019), can compromise a cryptocurrency system by rejecting transactions and allowing for own coins to be spent multiple times, known as double spend (Vaigandla et al. 2023).

One of the important concerns in blockchain technology, specifically in EHRs, is privacy. This is why some security protocols in EHRs cause unauthorized access to sensitive data, namely, using a replacement method that discards information without the owner's consent (Siyal et al. 2019). In this regard, users must have access to patient data in blockchain-based EHR systems to deliver personalized services. This would indicate a significant challenge in preserving patient information. A suitable solution is to implement a system based on blockchain that utilizes cryptographic mechanisms to verify the confidentiality and integrity of data (Isravel et al. 2023).

13.4.2.2 Scalability and Data Storage

The increase in the amount of data in blockchain networks can cause problems in the scalability and performance of the network. This is why the network needs to manage more data when the volume of blockchain transactions is increased, which causes the transaction time to take longer. Therefore, scalability in blockchain networks can be one essential challenge. In healthcare applications, due to the need for quick and timely access to data, the use of blockchain to manage large-scale healthcare data is an important challenge (Atadoga et al. 2024; Vaigandla et al. 2023).

The other challenge to scalability is the increase in the computing resources of IoT tools due to the growth of applicants. Moreover, when various smart tools or sensors are integrated into the blockchain network, this becomes a complex problem (Isravel et al. 2023; Singh et al. 2023a). Second, the computing power required by IoT devices can be too low to take advantage of blockchain capabilities, resulting in suboptimal or extreme speed. This can make it difficult to run the main device software and the blockchain software at the same time, which can lead to inefficiencies and possible errors. This problem verifies the need for a more efficient and scalable blockchain solution.

As a result, the scalability of blockchain systems in healthcare is a significant obstacle that firstly depends on the amount of data generated in healthcare. Because storing this amount of data in the blockchain can be inefficient and even impractical. This challenge causes the efficiency of the blockchain systems to decrease. To deal with this challenge, researchers have proposed various scaling solutions with the aim of reducing data that should be processed and verified.

13.4.2.3 Reliability and Data Access

Another huge challenge is how to ensure data availability and data reliability. This is not an easy task because of the distributed aspect of blockchains and creates several problems related to the consistency and integrity of the existing data, as well as challenges in keeping them updated and accurate. Furthermore, its decentralized characteristic puts the blockchain architecture at risk of cyber-attack threats by default. This, therefore, underlines the need to take adequate security measures that could prevent such attacks and assure the credibility of the blockchain data (Agbo et al. 2019; Atadoga et al. 2024).

In a healthcare setup, data integrity is crucial to confidentiality in the requirements of patient treatments and medical documentation. The loss of centralization in blockchain technology has made it challenging to guarantee an available set of data feeds to all the appropriate stakeholders. Thus, it has brought about issues relating to data silos and collaboration.

13.4.2.4 Interoperability and Standardization

Interoperability would involve the smooth sharing and transfer of data among systems. In other words, it is a requirement for healthcare, mainly to have accessibility and exchange of data among different providers, namely, hospitals, pharmacies. However, although a traditional approach of storing all healthcare data in one centralized database has been followed, it becomes a great limitation to truly realizing interoperability. This is because each system or organization might be running with different data formats, terminologies or languages, and protocols—thereby making sharing and integration of information very difficult.

Interoperability is a state that has, for a long time, been echoed in the corridors of healthcare environments. Blockchain technology is a feasible way forward that has been proposed to help achieve interoperability in healthcare. The ability to share and transfer data across distinct parties with a lack of the presence of a central authority is decentralized and, by itself, is well aligned with the definition of interoperability—the exchange of information between dissimilar systems and/or applications (McGhin et al. 2019; Wenhua et al. 2023).

Although blockchain holds promise to improve interoperability, the technology has faced several challenges, including the standardization of protocols and frameworks for developing blockchain-based applications in healthcare. They cannot be interoperable, as the vendor and platform for each application could be developed differently. This, in turn, will hamper the seamless exchange of data and limit the blockchain in the healthcare domain.

To tackle this challenge, the development of industry-wide standards and guidelines for blockchain-based healthcare applications is crucial. This will guarantee that various systems and platforms are able to effectively communicate and exchange data, facilitating genuine interoperability and unleashing the complete advantages of blockchain technology within the healthcare environment.

13.4.2.5 Transparency

The transparency associated with blockchain technology could be an advantage or disadvantage in the health field. While transparency is significant for maintaining the integrity of information, it also forms one of the greatest threats to patient confidentiality. This is because data replication on the many nodes that the data travel through may find its way into access that is not authorized and hence into the hands of unauthorized entities (Atadoga et al. 2024).

This faces the transparent tendency that goes against an organization's policies and the interests of individuals, especially in healthcare because of the sensitive data on patients it holds. The chances of unauthorized access to the data are a significant concern, as it can trade off patient confidentiality, which is vital for maintaining their health and well-being. Besides, hacker implementation can also be done through the transparency of blockchain data, and with the secret encryption, key of an end user worked out, all data that shall be connected with them in the application shall be under the hacker's control. Therefore, stringent securities are to be in place to protect the data of patients and keep them confidential (Benchoufi et al. 2017; Molli 2023).

Further, transparency is increased and confidentiality is lowered by the fact that all nodes within a blockchain can see data on its chain; this is a predicament to the healthcare organization in the aspect to what extent they are transparent because they have to protect patient confidentiality.

13.4.2.6 Technological Challenges

The implementation of blockchain in healthcare faces numerous obstacles. One of the primary obstacles is the lack of technical knowledge in blockchains, which requires a specific level of proficiency in blockchain concepts (Urwongse and Culver 2020). To solve this challenge, huge efforts are needed to improve the user experience and develop blockchain protocols (Urwongse and Culver 2020). The technical challenges that blockchain technology faces are numerous, including the following:

- Latency and Throughput Restrictions: Processing a blockchain transaction takes time, and this can become a big limitation in healthcare, where access to medical data is time-critical.
- Blockchain Size: In case all devices perform transactions, blockchain size can become a significant problem, which may cause the need for stronger miners and more storage space.
- Storage Requirements: Storing entire transactions in a blockchain network can cause a significant challenge.
- Implementation Complexity: This would be the blending of blockchain into the existing healthcare system, which at times gets very complex, costing huge amounts of time and expertise.

- Cost and Resource Constraints: Blockchain technology implementation is expensive, especially for healthcare organizations with limited resources. Setting up a digitalized platform and its maintenance, and the change from conventional health information systems, differ. The blockchain-driven care infrastructure requires a constant supply of resources regarding problem fixing, enhancement of capabilities, backup, and reporting.

13.5 Case Studies and Examples

In this section, some of the blockchain applications in healthcare as well as obstacles in creating healthcare blockchain systems based on many research papers are shown (Isravel et al. 2023; Ghosh et al. 2023; Agbo et al. 2019; Singh et al. 2023b; Urwongse and Culver 2020; Bennacer et al. 2023; Atadoga et al. 2024). The introduction of blockchain use cases in healthcare is conducted based on the category presented in Sect. 13.3. For this purpose, these use cases are explained in the fields of EHRs, SCM, RPM, clinical trials, and health insurance claims.

EHRs domain: There are numerous instances of blockchain-based applications in this domain such as Healthchain (Ahram et al. 2017), which is constructed on the Hyperledger blockchain. Moreover, some platforms are based on the Ethereum blockchain, namely, Acile (Dagher et al. 2018) and MedRec (Azaria et al. 2016). These applications are focused on enhancing the security, privacy, and accessibility of patient data through the utilization of blockchains. MeDShare (Xia et al. 2017), MediBlock (Babu et al. 2023) are other applications that use blockchain to share secure and auditable data between healthcare providers, patients, and others.

SCM domain: The AG platform (Bocek et al. 2017) is one of the most important use cases in this field as far as blockchain can be involved to avoid the distribution of fake drugs. BBTCD (Blockchain-Based Traceability of Counterfeited Drugs) (Rai 2023) is a blockchain application in order to trace drug origins and thereby curb fake drugs. Another blockchain platform designed for tracing pharmaceutical products throughout the supply chain is PharmaChain (Gomasta et al. 2023) designed to make it easier to record the origin of drugs safely and transparently.

RPM domain: There are many use cases in the RPM domain (Boikanyo et al. 2023; Uddin et al. 2018; Choudhury et al. 2019; Barka et al. 2021; Prabu Sankar and Usha 2024). Just to name a few, HapiChain (Kordestani et al. 2020) is a recent application that addresses the challenges of privacy and security concerns related to patient-centered telemedicine, supported by blockchain. In this application, patients are the center of the system where they own their data and have the right to securely communicate with their providers. Also, Health-ID (Javed et al. 2021) is another use case in this domain, which uses a unique identifier for each patient, which it stores and checks on the blockchain. Patients can control their details, including health records and decide which doctor will have access to it.

Clinical trials domain: Several case studies and research papers have highlighted the integration of blockchain into clinical research as beneficial (Nugent et al. 2016; Benchoufi et al. 2017; Maslove et al. 2018; Hirano et al. 2020; Anwar et al. 2023). BlockTrial (Maslove et al. 2018) is a research that explored the importance of using blockchain in clinical trials. This research would therefore confirm the importance of blockchain in data integrity and improvement of trust in clinical research. Another case study involves a Japanese breast cancer clinical trial conducted by researchers from SUSMED incorporated with the National Cancer Center Japan (Hirano et al. 2020). The study indicates how a blockchain system in data management ensures the security of clinical trial data. This application is robust against unauthorized changes in the data and sustains operations even in the event of servers going offline.

Health insurance claims domain: Among several applications of blockchain, EthInsurance (Sawalka et al. 2022) is one of the blockchain applications that handles health insurance claims to make things clearer between insurance companies by cutting out intermediaries. EthInsurance allows insurance companies, hospitals, and patients to communicate together directly. B-SAHIC (Khatun et al. 2023) is another application of blockchain, handling health insurance claim processing securely and automatically. By using blockchain technology, claims data are maintained in an immutable way. Also, smart contracts are used for checking and deciding on claims.

13.6 Future Trends

Blockchains are expected to reshape healthcare systems by promoting the security and efficiency of data. However, important technological obstacles such as complexity, interoperability, the difficulty of integration with existing health systems, and scalability must be addressed. Future research should address these challenges. So, some of the future research points on this topic can be mentioned in the following. The first topic is big data. Plenty of data exist in healthcare and since all of these data should be used for the advancement of healthcare systems, big data could be given as an important topic in the future. One of the problems related to big data is its security. For instance, a key challenge to healthcare systems is privacy while the management and processing of large volumes of healthcare data generated by individuals, especially from wearable devices and mobile phones. Blockchain technology could overcome all security problems related to big data by making traceability and immutability possible. Secondly, accompanying blockchains with artificial intelligence in various healthcare systems provides sustainable and efficient systems. Plenty of machine learning methods are proposed to identify fake EHR data and thus only valid EHRs are allowed to be stored in the blockchain. So, machine learning and deep learning methods are able to predict future trends in diseases by using data kept on the blockchain (Isravel 2023; Urwongse and Culver 2020).

One of the future tasks in this field is empowering patients with ownership of their data. Blockchains will have been attempting to allow users more control over their data. Therefore, in the field of healthcare, by providing the possibility of storing patient data records in a decentralized blockchain, patients will be able to take control of their data management.

Another challenge that should be addressed is the lack of interoperability and standardization among various applications. If blockchains want to be effectively used in healthcare, appropriate protocols must be provided in this area. Therefore, researchers focus on developing open protocols and standards in this area (Singh et al. 2023b).

Lastly, cost savings and efficiency boosting in healthcare systems are other future research. Blockchains have the capability to significantly reduce expenses and improve efficiency through simplifying processes and omitting intermediaries. So by focusing on these factors, the presentation of new systems will be continued.

The current developments verify that blockchain will increasingly become important in healthcare. Despite existing obstacles such as scalability, privacy, and integration, a possible option for addressing some important healthcare challenges, namely, security, transparency, and interoperability, is blockchain.

13.7 Conclusion

The role of blockchains in improving healthcare services is obvious. Blockchains provide ground-breaking solutions for addressing healthcare issues, namely, data management, security, and interoperability. Some healthcare domains that take advantage of basic blockchain aspects of decentralization and encryption include EHRs, SCM, RPM, clinical trials, and health insurance claims processing. For example, by using blockchain technology, patient data records become immutable and unalterable, therefore the levels of confidentiality and integrity of sensitive data are improved. Furthermore, blockchain can be used in the pharmaceuticals area. It is being implemented to provide supply chain traceability, reduce risks of fake drugs hitting the market, and ensure the authenticity of medical products. It can automate processes, increase transparency, and reduce fraud in blockchain platforms using smart contracts, particularly in insurance claims and clinical trials.

Blockchain technology in healthcare has advantages and disadvantages that are important in improving modern systems. Take, for example, obstacles such as scalability, interoperability, and regulatory compliance should be overcome in modern healthcare. The only way to achieve this is through collaboration among health organizations, developers, and regulators. The use of emerging technologies such as AI and IoT along with blockchains will cause massive changes in healthcare. As an example, consider that in a society where everything is safe and transparent due to blockchain technology while AI is used for data processing purposes as well and IoTs allow devices to communicate together, then these healthcare systems will be able to stand out stronger than before. So, people in this society will undoubtedly expect a better access rate to more personalized treatment accesses in the shortest time.

As a result, blockchain provides a highly secure, decentralized, and transparent framework to manage healthcare data whose consequent achievement was considerable improvements in the quality of healthcare along with savings on expenses. Even though there are obstacles facing this kind of technology, researchers are still trying hard to keep up with its increasing utilization and overcome these challenges later on in future designs.

References

Agbo CC, Mahmoud QH, Eklund JM (2019) Blockchain technology in healthcare: a systematic review. Healthcare 7:56. https://doi.org/10.3390/healthcare7020056

Ahram T, Sargolzaei A, Sargolzaei S et al (2017) Blockchain technology innovations. In: 2017 IEEE technology and engineering management conference (TEMSCON), pp 137–141

Angraal S, Krumholz HM, Schulz WL (2017) Blockchain technology: Applications in health care.CircCardiovascQualOutcomes10:1–3. https://doi.org/10.1161/CIRCOUTCOMES.117.003800

Anwar A, Goyal SB, Jan T (2023) Blockchain-based Clinical trials: a meta-model framework for enhancing security and transparency with a novel algorithm. Int J Technol 14:1380. https://doi.org/10.14716/ijtech.v14i6.6703

Atadoga A, Elufioye OA, Omaghomix TT et al (2024) Blockchain in healthcare: a comprehensive review of applications and security concerns. Int J Sci Res Arch 11:1605–1613. https://doi.org/10.30574/ijsra.2024.11.1.0244

Azaria A, Ekblaw A, Vieira T, Lippman A (2016) MedRec: using blockchain for medical data access and permission management. In: 2016 2nd international conference on open and big data (OBD), pp 25–30

Babu ES, Yadav BVRN, Nikhath AK et al (2023) MediBlocks: secure exchanging of electronic health records (EHRs) using trust-based blockchain network with privacy concerns. Cluster Comput 26:2217–2244. https://doi.org/10.1007/s10586-022-03652-w

Barka E, Dahmane S, Kerrache CA et al (2021) STHM: a secured and trusted healthcare monitoring architecture using SDN and blockchain. Electron 10:1787. https://doi.org/10.3390/electronics10151787

Bazel MA, Mohammed F, Ahmad M (2023) A systematic review on the adoption of blockchain technology in the healthcare industry. EAI Endorsed Trans Pervasive Heal Technol 9:e4. https://doi.org/10.4108/eetpht.v9i.2844

Ben Fekih R, Lahami M (2020) Application of blockchain technology in healthcare: a comprehensive study. The impact of digital technologies on public health in developed and developing countries: 18th international conference, ICOST 2020, Hammamet, Tunis. June 24–26, Proceedings, vol 12157, pp 268–276

Benchoufi M, Porcher R, Ravaud P (2017) Blockchain protocols in clinical trials: transparency and traceability of consent. F1000Research 6:66. https://doi.org/10.12688/f1000research.10531.1

Bennacer SA, Sabiri K, Aaroud A et al (2023) A comprehensive survey on blockchain-based healthcare industry: applications and challenges. Indones J Electr Eng Comput Sci 30:1558–1571. https://doi.org/10.11591/ijeecs.v30.i3.pp1558-1571

Bocek T, Rodrigues BB, Strasser T, Stiller B (2017) Blockchains everywhere—a use-case of blockchains in the pharma supply-chain. In: 2017 IFIP/IEEE symposium on integrated network and service management (IM), pp 772–777

Boikanyo K, Zungeru AM, Sigweni B et al (2023) Remote patient monitoring systems: applications, architecture, and challenges. Sci African 20:e01638. https://doi.org/10.1016/j.sciaf.2023.e01638

Choudhury O, Fairoza N, Sylla I, Das A (2019) A blockchain framework for managing and monitoring data in multi-site clinical trials

Dagher GG, Mohler J, Milojkovic M, Marella PB (2018) Ancile: privacy-preserving framework for access control and interoperability of electronic health records using blockchain technology. Sustain Cities Soc 39:283–297. https://doi.org/10.1016/j.scs.2018.02.014

David S, Duraipandian K, Chandrasekaran D et al (2023) Impact of blockchain in healthcare system. In: Unleashing the potentials of blockchain technology for healthcare industries, pp 37–57

Engelhardt MA (2017) Hitching healthcare to the chain: an introduction to blockchain technology in the healthcare sector. Technol Innov Manag Rev 7:22–34. https://doi.org/10.22215/timreview/1111

Fan K, Wang S, Ren Y et al (2018) MedBlock: efficient and secure medical data sharing via blockchain. J Med Syst 42:136. https://doi.org/10.1007/s10916-018-0993-7

Ghosh PK, Chakraborty A, Hasan M et al (2023) Blockchain application in healthcare systems: a review. Systems 11:1–44. https://doi.org/10.3390/systems11010038

Gomasta SS, Dhali A, Tahlil T et al (2023) PharmaChain: blockchain-based drug supply chain provenance verification system. Heliyon 9:e17957. https://doi.org/10.1016/j.heliyon.2023.e17957

Griggs KN, Ossipova O, Kohlios CP et al (2018) Healthcare blockchain system using smart contracts for secure automated remote patient monitoring. J Med Syst 42:130. https://doi.org/10.1007/s10916-018-0982-x

Hellani H, Sliman L, Samhat AE, Exposito E (2021) On blockchain integration with supply chain: overview on data transparency. Logistics 5:46. https://doi.org/10.3390/logistics5030046

Hirano T, Motohashi T, Okumura K et al (2020) Data validation and verification using blockchain in a clinical trial for breast cancer: regulatory sandbox. J Med Internet Res 22:e18938. https://doi.org/10.2196/18938

Andrew J, Isravel DP, Sagayam KM et al (2023) Blockchain for healthcare systems: architecture, security challenges, trends and future directions. J Netw Comput Appl 215:103633. https://doi.org/10.1016/j.jnca.2023.103633

Javed IT, Alharbi F, Bellaj B et al (2021) Health-ID: a blockchain-based decentralized identity management for remote healthcare. Healthcare 9:712. https://doi.org/10.3390/healthcare9060712

Khatun M, Islam RA, Islam S (2023) B-SAHIC: a blockchain based secured and automated health insurance claim processing system. J Intell Fuzzy Syst 44:4869–4890. https://doi.org/10.3233/JIFS-220690

Kordestani H, Barkaoui K, Zahran W (2020) HapiChain: a blockchain-based framework for patient-centric telemedicine. In: 2020 IEEE 8th international conference on serious games and applications for health (SeGAH). IEEE, pp 1–6

Makroum MA, Adda M, Bouzouane A, Ibrahim H (2022) Machine learning and smart devices for diabetes management: systematic review. Sensors (Basel) 22. https://doi.org/10.3390/s22051843

Maslove DM, Klein J, Brohman K, Martin P (2018) Using blockchain technology to manage clinical trials data: a proof-of-concept study. JMIR Med Inform 6:e11949. https://doi.org/10.2196/11949

McGhin T, Choo KKR, Liu CZ, He D (2019) Blockchain in healthcare applications: research challenges and opportunities. J Netw Comput Appl 135:62–75. https://doi.org/10.1016/j.jnca.2019.02.027

Molli VLJ (2023) Blockchain technology for secure and transparent health data management: opportunities and challenges. J Healthc AI ML 10:1–15

Nugent T, Upton D, Cimpoesu M (2016) Improving data transparency in clinical trials using blockchain smart contracts. F1000Research 5:2541. https://doi.org/10.12688/f1000research.9756.1

Prabu Sankar N, Usha D (2024) Advancing rural healthcare: a novel approach to designing blockchain-enabled IoB devices and integrating fuzzy intelligence systems for patient monitoring and record management. J Intell Fuzzy Syst Preprint 1–9. https://doi.org/10.3233/JIFS-233752

Prokofieva M, Miah SJ (2019) Blockchain in healthcare. Australas J Inf Syst 23:1–22. https://doi.org/10.3127/ajis.v23i0.2203

Rai BK (2023) BBTCD: blockchain based traceability of counterfeited drugs. Heal Serv Outcomes Res Methodol 23:337–353. https://doi.org/10.1007/s10742-022-00292-w

Rathore H, Mohamed A, Guizani M (2020) Blockchain applications for healthcare. Elsevier Inc.

Rejeb A, Treiblmaier H, Rejeb K, Zailani S (2021) Blockchain research in healthcare: a bibliometric review and current research trends. J Data Inf Manag 3:109–124. https://doi.org/10.1007/s42488-021-00046-2

Sawalka S, Lahiri A, Saveetha D (2022) EthInsurance: a blockchain based alternative approach for health insurance claim. In: 2022 international conference on computer communication and informatics (ICCCI). IEEE, pp 1–9

Shinde R, Patil S, Kotecha K et al (2024) Securing AI-based healthcare systems using blockchain technology: a state-of-the-art systematic literature review and future research directions. Trans Emerg Telecommun Technol 35:1–48. https://doi.org/10.1002/ett.4884

Singh D, Monga S, Tanwar S, et al (2023) Adoption of Blockchain Technology in Healthcare: Challenges, Solutions, and Comparisons. Appl Sci 13:. https://doi.org/10.3390/app13042380

Singh Y, Jabbar MA, Kumar Shandilya S et al (2023) Exploring applications of blockchain in healthcare: road map and future directions. Front Public Heal 11. https://doi.org/10.3389/fpubh.2023.1229386

Siyal AA, Junejo AZ, Zawish M et al (2019) Applications of blockchain technology in medicine and healthcare: challenges and future perspectives. Cryptography 3:1–16. https://doi.org/10.3390/cryptography3010003

Thenmozhi M, Dhanalakshmi R, Geetha S, Valli R (2021) WITHDRAWN: implementing blockchain technologies for health insurance claim processing in hospitals. Mater Today Proc. https://doi.org/10.1016/j.matpr.2021.02.776

Uddin MA, Stranieri A, Gondal I, Balasubramanian V (2018) Continuous patient monitoring with a patient centric agent: a block architecture. IEEE Access 6:32700–32726. https://doi.org/10.1109/ACCESS.2018.2846779

Urwongse R, Culver K (2020) Applications of blockchain in healthcare. In: Patient-centered digital healthcare technology: novel applications for next generation healthcare systems. Institution of engineering and technology, pp 205–242

Vaigandla KK, Siluveru M, Kesoju M, Karne R (2023) Review on blockchain technology: architecture, characteristics, benefits, algorithms, challenges and applications. MesopN J Cybersecur 2023:73–84

Villarreal ERD, Garcia-Alonso J, Moguel E, Alegria JAH (2023) Blockchain for healthcare management systems: a survey on interoperability and security. IEEE Access 11:5629–5652. https://doi.org/10.1109/ACCESS.2023.3236505

Wenhua Z, Qamar F, Abdali T-AN et al (2023) Blockchain technology: security issues, healthcare applications challenges and future trends. Electronics 12:546. https://doi.org/10.3390/electronics12030546

Xia Q, Sifah EB, Asamoah KO et al (2017) MeDShare: trust-less medical data sharing among cloud service providers via blockchain. IEEE Access 5:14757–14767. https://doi.org/10.1109/ACCESS.2017.2730843

Zhang P, White J, Schmidt DC et al (2018) FHIRChain: applying blockchain to securely and scalably share clinical data. Comput Struct Biotechnol J 16:267–278. https://doi.org/10.1016/j.csbj.2018.07.004

Zhou L, Wang L, Sun Y (2018) MIStore: a blockchain-based medical insurance storage system. J Med Syst 42:149. https://doi.org/10.1007/s10916-018-0996-4