ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue IV, April 2026

# Architecting Trust: An Analytical Overview of Blockchain Fundamentals and Real-World Applications

**Jeewesh Kumar Jha[1], Gurjeet Singh[2], Sudhir Pathak[3]**

**[1]Research Scholar: Lords University, Alwar (Raj.)**

**[2,3]Prof.(Dr.) Lords University, Alwar (Raj.)**

**DOI: https://doi.org/10.51583/IJLTEMAS.2026.150400076**

**Received: 13 April 2026; Accepted: 18 April 2026; Published: 09 May 2026**

## ABSTRACT

Over the past decade, the mainstream adoption of digital currencies has also brought renewed attention to their foundational technology, blockchain. Blockchain is a distributed data structure that maintains a shared ledger across multiple participants without requiring a central authority. Its design includes features like being decentralized, keeping records in a specific order with timestamps, ensuring data security through cryptography, and requiring agreement from participants to validate. Together, these properties make ledger entries highly resistant to unauthorized modification and support transparent auditing across participating nodes.

These features make blockchain a strong solution to many problems in traditional financial systems, such as having one point of failure, limited transparency, extra work for matching records, and needing trusted middlemen. Consequently, blockchain has attracted sustained interest from financial intermediaries, technology firms, and public sector institutions seeking more resilient and verifiable transaction and data-sharing mechanisms.

This article provides a structured overview of blockchain fundamentals and its application landscape. It begins by defining blockchain and outlining its core operating principles, with particular emphasis on decentralization, immutability, and transparency. Then, it looks at how blockchain has changed from early cryptocurrency-based systems to larger, multi-domain platforms that support decentralized, programmable applications. The main parts look at important parts of blockchain systems, the real-world challenges and limits they face, and examples of how they are used in different fields, as well as guidelines for deciding when to use blockchain as a solution. The study closes by outlining the current maturity of blockchain technologies, the principal technical and governance challenges that remain, and the research and engineering directions required to translate blockchain's potential into scalable, trustworthy deployments.

## INTRODUCTION

The last ten years have seen a lot of people use digital currencies, which has also sparked new interest in blockchain, the technology that powers them. Blockchain is a way to store data that doesn't need a central authority to keep track of a ledger that many people can use. It is decentralized, keeps records in a certain order with timestamps, uses cryptography to protect data, and needs everyone involved to agree to be valid. These features work together to make it very hard to change ledger entries without permission and easy for all nodes to check the entries.

Blockchain is a good solution to a lot of the problems with traditional financial systems. For instance, they have one point of failure, aren't very clear, need more work to match records, and need middlemen who can be trusted. Because of this, blockchain is still of interest to tech companies, financial intermediaries, and public sector organizations that want transaction and data-sharing systems that are more reliable and can be checked.

This article gives a clear overview of the basics of blockchain and the different ways it can be used. It begins by explaining what blockchain is and how it works, with a focus on decentralization, immutability, and transparency.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue IV, April 2026

Next, it talks about how blockchain has grown from small, single-domain systems that only worked with cryptocurrencies to larger, multi-domain platforms that can run decentralized, programmable apps. The main parts talk about the important parts of blockchain systems, the problems and limitations they face in the real world, and how they are used in different fields. They also tell you when to use blockchain as a solution. At the end of the study, it talks about how far along blockchain technologies are now, the main technical and governance issues that still need to be worked out, and the research and engineering paths that need to be taken to turn blockchain's potential into large-scale, reliable deployments.

A growing body of research characterizes blockchain as a potentially disruptive innovation capable of reshaping coordination, verification, and value exchange across sectors (l-Samarai, B., & Morato, J., 2025). A systematic literature review regarding blockchain technology and educational systems within the Gulf Cooperation Council (GCC). Applied Sciences, 15(5), 2404. Chinnasamy, P., Subashini, B., Ayyaswamy, R. K., and others (2025). Management of electronic educational documents on a blockchain with role-based access control and a machine learning model. Scientific Reports, 15, 18828. June, M. (2025). Platform framework for AIoT healthcare systems that use blockchain. Frontiers in Communications and Networks, 6, Article 1538965, by Mohammed, M. A., De-Pablos-Heredero, C., and Botella, J. L. M. (2025). A bibliometric analysis of the transformative realm of blockchain-enabled central bank digital currencies. Eurasian Economic Review, 15, 53–88. D. L. Silaghi and D. E. Popescu (2025). A systematic review of blockchain-based initiatives juxtapose.

At the foundation of these use cases is the concept of a shared ledger that is persistent, traceable, and verifiable across organizational boundaries. This capability supports transparent recording of transactions involving tangible and intangible assets while reducing dependence on centralized reconciliation and manual verification. In practice, organizations adopt blockchain to improve visibility across processes, strengthen provenance and accountability, and enhance transactional confidence where multi-party coordination is required. By enabling near real-time access to consistent records and limiting the feasibility of undetected changes, blockchain can support higher levels of transparency and trust, which in turn motivates firms to explore new operational models and efficiency gains in complex ecosystems.

**Background and motivation:**

Blockchain entered the public mainstream through the emergence of cryptocurrency systems, most notably Bitcoin, which demonstrated that digital value could be transferred over a peer-to-peer network without a central clearing authority. In such systems, digital assets can be exchanged as electronic cash using a distributed ledger that is replicated across network participants. The development and maturation of Bitcoin drew on advances in mathematics, cryptography, and computer science, integrating these foundations into an operational framework for decentralized record keeping.

A defining principle of early blockchain systems is decentralization, which enables participants to validate and record transactions collectively. This coordination is supported by mechanisms such as time stamping, distributed consensus, cryptographic protection of data, and incentive structures that encourage honest participation in the network (Zhu, Guo & Zhang, 2021). Together, these elements address persistent challenges found in centralized architectures, including single points of failure, limited transparency, and heightened exposure to tampering or unauthorized modification. As a result, blockchain has been widely examined as a technical and governance alternative for environments where multiple parties must share data or transact securely in the absence of a fully trusted intermediary.

The promise of improved integrity, resilience, and auditability has made blockchain particularly relevant for domains with stringent security and compliance requirements, including financial services and public sector applications (Al-Jaroodi & Mohamed, 2019; Bodkhe et al., 2020; Javaid et al., 2021). Moreover, the demonstrated viability of Bitcoin accelerated the development of subsequent blockchain platforms and currencies, such as Ethereum, which expanded the design space beyond payment transfer toward programmable transactions and decentralized applications. This progression has positioned blockchain as a general-purpose infrastructure, motivating research into its applicability across a broad range of nonfinancial settings that require trustworthy, multi-party data management, which will be examined in later sections.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue IV, April 2026

**Problem statement**

Blockchain adoption is accelerating across sectors, driven by the expectation that decentralized ledgers can strengthen transparency, reduce reliance on intermediaries, and improve the integrity of multi-party transactions. Despite this momentum, practical deployment remains constrained by unresolved technical and governance challenges. Key concerns include limited scalability under high transaction throughput, substantial energy requirements in certain consensus designs, weak interoperability across heterogeneous blockchain platforms, and regulatory uncertainty surrounding compliance, accountability, and cross-border operation. Although these issues are widely recognized, the evidence base is still fragmented, and comparatively fewer studies provide an integrated view that connects blockchain foundations to application-level security, privacy, and operational constraints (Pieters, Kokkinou & van Kollenburg, 2022).

Against this background, the present work undertakes a comprehensive review of blockchain systems, spanning core architectural constructs, enabling technologies, representative application domains, and the security and privacy risks that arise in blockchain-based deployments. The review emphasizes the fundamental building blocks of blockchain while also synthesizing current knowledge on threat surfaces, attack dynamics, and mitigation strategies relevant to real-world implementations. The primary aim of this study is to synthesize current advancements in blockchain technologies and their applications, examine key security and privacy issues, and highlight the unresolved research challenges that must be overcome to enable dependable and scalable implementation.

**The Article's Structure**

The rest of this paper is organized in a certain way. The part about survey methodology talks about how the review was done and what criteria were used to choose the studies that were relevant. The literature review section brings together the results of earlier research and shows the most common themes. The talk about important blockchain technologies covers the basic parts that are needed to make blockchain systems work. The part about blockchain types sorts different types of architecture and governance. The part about security in blockchain systems looks at common attack methods, how to stop them, and the real-world trade-offs that come with putting them into action. The part about future research directions looks at new methods and possible uses for them. The discussion about limitations looks at ongoing technical problems and problems with getting people to use them. The paper ends by going over the main points and talking about what they mean for future academic and practical developments.

**How the survey was done**

This study uses a systematic literature review method and follows established review protocols to ensure clarity, thoroughness, and consistency in the way the research is done. There were three main parts to the review: planning the research, finding the right literature and collecting data, and then doing a detailed analysis and synthesis of the chosen studies.

**Planning phase**

In the planning stage, we defined the scope of the review and aligned it with the objectives of the study, namely to synthesize blockchain fundamentals, applications, and security and privacy considerations. We also specified the thematic boundaries used to guide retrieval and screening, including core blockchain concepts, enabling technologies, application domains, and threat and defense perspectives.

**Data collection phase**

For the initial retrieval of scholarly literature, we conducted structured searches in Google Scholar and Web of Science using topic-relevant keywords related to blockchain technology and its applications. This broad retrieval step was intentionally inclusive to capture diverse disciplinary perspectives. Following the initial search, we applied a focused selection strategy by prioritizing papers that were strongly aligned with the study themes and demonstrated scholarly impact, including highly cited works where relevant to the objectives.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue IV, April 2026

Given the fast pace of change in blockchain research and practice, we complemented peer-reviewed sources with carefully selected practitioner-oriented materials. These included technical reports, industry publications, and other authoritative online sources that provide implementation insights, emerging trends, and operational considerations. This extension was used to enrich academic evidence with practice-informed perspectives, particularly in areas where industrial adoption or technical standards evolve more rapidly than journal publication cycles.

**Data review phase**

In the review stage, retrieved items were screened for relevance to the research questions and then analyzed using thematic synthesis. Sources were grouped by conceptual focus, such as blockchain architecture and consensus, platform types, application categories, and security and privacy issues. Evidence was compared across studies to identify convergence, inconsistencies, and research gaps, with attention to both technical contributions and deployment-oriented constraints.

Overall, this methodology combines scholarly depth with practice-grounded inputs to provide a comprehensive and current view of blockchain technology while maintaining the structure and discipline expected in systematic review reporting for Springer-style publications.

## LITERATURE REVIEW

Policy and industry developments around 2016 contributed to a marked expansion of scholarly and practitioner interest in blockchain beyond its original association with cryptocurrencies. In early 2016, the People's Bank of China publicly signaled an intention to advance work toward an official digital currency, which in turn prompted broader engagement from financial researchers and institutions examining the enabling role of blockchain in digital currency infrastructures (Pilkington, 2016). During the same period, the UK Government released the report Distributed Ledger Technology: Beyond Blockchain, reflecting growing policy attention to distributed ledger applications within public administration and government services (Hancock & Vaizey, 2016).

Parallel to these public sector signals, business and consulting analyses framed blockchain as a potentially disruptive general-purpose technology. For example, McKinsey & Company argued that blockchain could reshape processes and industries by enabling new models of coordination, verification, and value transfer, and positioned it alongside prior waves of industrial and digital transformation discussed in the innovation literature (McKinsey Company, 2016; Hancock & Vaizey, 2016). This narrative helped broaden the research agenda from cryptocurrency protocols toward enterprise architectures, governance models, and sector specific implementations.

Industry adoption also accelerated, particularly across large technology firms in Asia and North America. In China, organizations such as Baidu and Huaneng Trust were reported to have participated in early domestically backed blockchain initiatives, indicating experimentation with blockchain for financial and operational use cases. Additional corporate implementations focused on provenance and anti-counterfeiting, including blockchain-enabled traceability platforms developed by JD.com. Major platform providers also introduced blockchain-oriented enterprise offerings, including supply chain-oriented initiatives attributed to Tencent and managed blockchain infrastructure services introduced by Amazon, aimed at lowering adoption barriers for smaller firms and ecosystem partners.

Finally, Alibaba Group leveraged blockchain’s decentralized storage and tamper resistance to support commercial offerings and productized services, including solutions branded as AntChain (Kong, 2021). Beyond finance and supply chains, reported experimentation extended into digital content and gaming ecosystems, including implementations that referenced blockchain services within platform environments associated with Microsoft. Collectively, these developments motivated a shift in the literature from conceptual discussions toward deployment-focused questions, including scalability, governance, interoperability, and assurance in real-world blockchain-based systems.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue IV, April 2026

**Evolution of Blockchain Research and Deployment (2016–2025)**

The rise of blockchain research and use between 2016 and 2025

From 2016 to 2025, it was clear that blockchain technology went from being something people were excited about to something that institutions used in a structured way. Most of the talk about blockchain in 2016 was about cryptocurrencies, especially Bitcoin. The main topics of discussion in academic and policy circles were decentralization, distributed trust, and how peer-to-peer financial systems could change the way we do business. At this point, central banks and governments were mostly just looking into things. For example, the People's Bank of China's efforts and the UK Government's policy reflections showed that people were becoming more aware, but they had not yet led to full regulatory or deployment frameworks. At that time, most of the research was theoretical and focused on basic architecture, consensus mechanisms, and how they might affect financial intermediation.

By 2025, blockchain will have come a long way toward becoming a technology used by businesses. The European Union's Markets in Crypto-Assets (MiCA) regulation and coordinated research efforts led by international financial institutions have made things a lot clearer when it comes to regulations. Many places have started testing Central Bank Digital Currency (CBDC) pilots. They have gone from being just an idea to being put into practice in a controlled way. This change shows how blockchain has gone from being a small part of new financial technologies to a strategically managed tech infrastructure.

From a technological point of view, the change is also important. In 2016, the most common way for blockchain networks to agree on things was through proof-of-work consensus. This made people worry about how well it would work and how much energy it would use. Many of the early performance issues have been fixed by 2025, thanks to the widespread use of Proof-of-Stake, Layer-2 scaling solutions, and zero-knowledge cryptographic methods. Researchers are now more interested in how to get different networks to work together, how to formally check smart contracts, and how to send messages safely between chains. These changes show that blockchain environments are maturing from being separate to being part of bigger digital ecosystems.

This change is even more clear when you look at how businesses are adopting it. Pilot projects and proof-of-concept implementations that started out small have grown into systems that are ready for production and can work with cloud-native infrastructures, AI frameworks, and IoT environments. Apps now do more than just send and receive cryptocurrency. They also include asset tokenization, keeping track of compliance with ESG standards, managing digital identities, decentralized finance (DeFi), and experimenting with new ways of governing through decentralized autonomous organizations (DAOs).

The comparison between 2016 and 2025 shows that both the story and the way things are done have changed a lot. People no longer just see blockchain as a way to break up traditional systems. Now, though, it is seen as a complementary infrastructure that is subject to institutional governance, performance optimization, and regulatory oversight. Modern research reflects this practical approach by focusing on scalability, compliance alignment, cybersecurity resilience, and long-term deployment. The evolution means that blockchain is not only getting better, but it is also becoming a normal part of plans for digital transformation.

Important technologies for blockchain What cryptography does for blockchain technology Without a central authority, blockchain systems can provide security, integrity, and verifiable trust thanks to cryptography. It helps with basic security goals throughout the transaction lifecycle, such as keeping information private when necessary, verifying the identity of participants, preventing people from denying their actions, and making it difficult to change things. When users make a blockchain transaction, cryptographic methods keep the data safe and check that it is correct. Many designs keep information private by encrypting sensitive payloads. Access to the protected information is restricted exclusively to those who hold the appropriate decryption keys. . In a broader sense, public key cryptography lets you bind your identity and give permission without giving away private information. Digital signatures, which are made with a private key and checked by any node with the matching public key, are usually used to approve transactions. This signature method gives strong proof that the claimed sender approved the transaction and makes it possible to audit and trace transactions across the ledger.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue IV, April 2026

Cryptographic mechanisms are also embedded within consensus protocols to protect the network to prevent manipulation and unauthorized modifications . In Proof of Work (PoW) systems, people compete to solve cryptographic puzzles that are hard to solve and cost a lot of money. This makes it hard to change history and helps keep bad actors from taking control. In Proof of Stake (PoS) systems, cryptographic mechanisms are used to select validators and verify the legitimacy of the assets they have staked. The goal is to lower energy costs while keeping security guarantees under threat models that are clear and well-defined. Both types of protocols use cryptographic primitives and economic incentives to make the system more stable. The use of cryptographic hash functions, which make small, fixed-length summaries of data, is another important part. Hashes are used to (i) (i) make a unique summary of the contents of a block, (ii) connect blocks through hash pointers, and (iii) make it easy to find unauthorized changes. Hash-based linking provides practical tamper evidence because a small change in input produces a very different digest. If you change historical data, it breaks subsequent hash references, which honest nodes can easily see. Finally, it's important to manage your keys well. Private keys control who can access digital assets and who can give permission. If they are compromised, it usually means losing control. Cryptography makes it possible to safely store and handle private keys using things like hardware-backed key stores, multi-signature policies, threshold cryptography, and recovery schemes. You can also encrypt sensitive data that is written to or referenced by the ledger.

In blockchain architectures, transactions are aggregated into discrete data structures referred to as blocks, which serve as the fundamental units for updating the distributed ledger. Participating nodes assemble validated transactions into a standardized block format and generate a header containing essential metadata, including a timestamp and a cryptographic reference to the preceding block. This reference, typically implemented as a hash pointer, ensures chronological ordering and preserves ledger consistency.

To facilitate efficient verification, many systems employ hash-based authenticated data structures such as Merkle trees. Through hierarchical hashing, individual transaction digests are recursively combined to produce a single Merkle root representing the complete transaction set. Inclusion of this root within the block header allows nodes to verify transaction membership without retrieving the entire block, thereby enhancing scalability and supporting lightweight client participation.

Following construction, blocks are propagated across the network and appended to the ledger upon consensus agreement. Temporary forks may arise when competing blocks are proposed; these are resolved by protocol-defined selection rules, such as adopting the chain with the greatest cumulative work or the one endorsed by the active validator set. Overall, block structure, cryptographic linking, authenticated data structures, validation mechanisms, and consensus protocols operate cohesively to ensure that transaction data in distributed ledgers remains temporally ordered, tamper-evident, and securely maintained.

**Blockchain Block Architecture: Header Components and Transaction Data Specifications**

| Field | Size (bytes) | Purpose in validation and storage |
|---|---|---|
| Block size | 4 | Declares the total serialized block length, excluding this field, enabling efficient parsing and storage allocation. |
| Version number | 4 | Signals the block format and validation rules applicable to this block, supporting protocol evolution and compatibility. |
| Parent block hash | 32 | Hash pointer to the previous block header, linking blocks into an ordered chain and providing tamper evidence across history. |
| Merkle root | 32 | Compact commitment to the complete transaction set using a Merkle tree, enabling efficient inclusion proofs and integrity verification. |
| Timestamp | 4 | Records block creation time as defined by the protocol, supporting ordering, validation bounds, and difficulty adjustment logic. |

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue IV, April 2026

| | | |
|---|---|---|
| Difficulty target | 4 | Encodes the proof-of-work threshold that the block header hash must satisfy, determining the computational effort required to create the block. |
| Random number (nonce) | 4 | Mutable value varied during mining to search for a header hash that meets the difficulty target. |
| Number of transactions | 1–9 | Variable length integer that states how many transactions are included, reducing overhead when the count is small. |
| Transactions | Variable | Serialized transaction list whose total size depends on count and individual transaction formats, with integrity committed via the Merkle root. |

Block the workflow for acceptance and validation, a node checks both the structural and cryptographic correctness of a block during block verification. First, the node makes sure that the serialized block can be parsed in a consistent way. This includes checking that the block size matches the byte length that was received and that the version number matches known rule sets. Then, the node checks the chain link by recalculating the hash of the previous block header and making sure it matches the hash of the parent block. This makes sure that the ledger stays the same. The node then checks the integrity of the transaction by recomputing hashes for the transactions that are included and rebuilding the Merkle tree to get a Merkle root. This computed value must correspond exactly to the Merkle root recorded in the block header. This step makes sure that the list of transactions hasn't changed since the block header was created. The protocol's difficulty adjustment mechanism uses the timestamp, which is checked against rules set by the protocol, like allowable clock skew and monotonicity rules. In systems that use Proof of Work, the node recomputes the hash of the block header and checks to see if it meets the difficulty target. Miners changed the nonce as part of the header data to find a hash below the target. This means that validating nodes only need to check the final condition instead of doing the search again. Finally, the node checks the number of transactions against the parsed transaction list and makes sure that the rules for transactions, like correct signatures, no double spending, and following consensus rules for fees and scripts, are followed. Only blocks that pass all of the header checks, Merkle integrity checks, consensus requirements, and transaction validity checks are added to the local canonical chain. Basic building blocks for organizing data in blockchain systems .A blockchain system needs a tightly linked set of data organization and integrity tools, such as standardized block structures, cryptographic hashing, authenticated data structures like Merkle trees, and timestamping (Zhu, Guo & Zhang, 2021). Together, these parts make up an ordered record of ledger updates that makes it easy to check and provides strong evidence of tampering across the Blockchain. A block is the basic unit of data used to keep track of transactions in systems like Bitcoin. In theory, each block has two logical parts. The first part is the block header, which holds metadata that is needed for linking, validation, and consensus checks. This includes the hash pointer to the previous block, the Merkle root, a timestamp, and parameters related to consensus. The second part is the block body (or block content), which is a serialized list of all the transactions that are in that block. This separation is very important because the header gives a short promise to the full block contents and gives nodes the fields they need to check that the block is in the right place in the ledger history. The Merkle tree is a fundamental authenticated data structure that is used to summarize and check the set of transactions that are stored in a block (Mohan, Mohamed Asfak & Gladston, 2020; de Ocáriz Borde, 2022). The main benefit is that it is more efficient. Instead of storing and checking each transaction separately all the time, the system computes hashes of transactions and combines them in a hierarchical way to make a single digest, the Merkle root, which is stored in the block header. A binary Merkle tree is used in most blockchain implementations. At the leaf level, transactions are hashed. Parent nodes are generated by computing the hash of the combined child hash values. This iterative procedure proceeds upward through the structure until a single root hash is obtained This structure makes two important things possible. First, any change to an included transaction changes the leaf hash, which then spreads up the tree and creates a new Merkle root. This makes it possible to see if someone has changed something. Second, a node can check that a transaction is in a block by using a compact Merkle proof that only includes the hashes along the path from the transaction leaf to the root, instead of needing to see the whole list of transactions. Merkle trees allow for scalable verification, especially for lightweight clients, while still keeping strong integrity guarantees for block contents.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue IV, April 2026

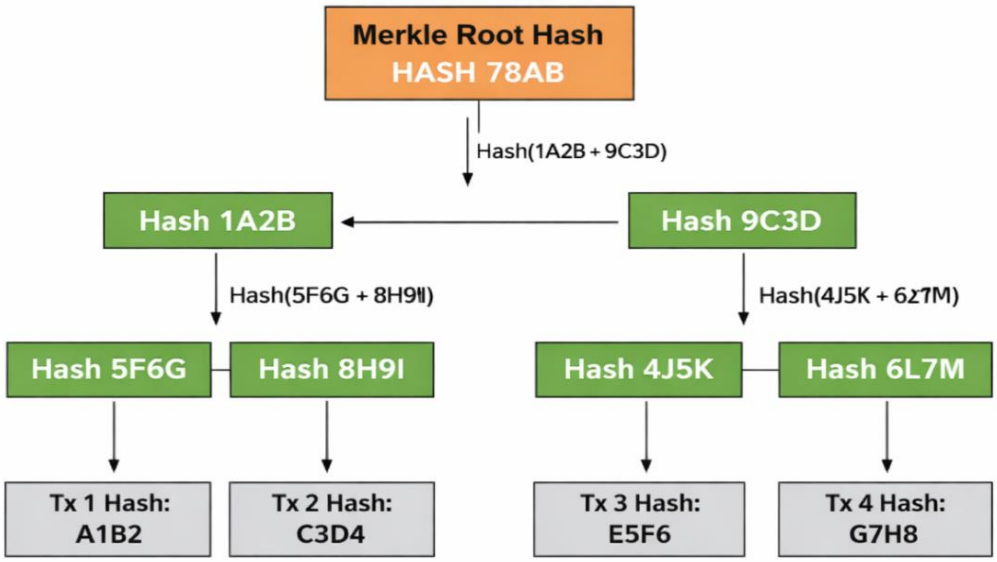



**Figure 1:** Transaction hash concept in Merkle tree.

**Merkle aggregation and commitment to transactions:**

Within a block, transactions are commonly organized using a Merkle tree, which aggregates individual transaction hashes into higher-level hashes until a single **Merkle root** is produced. This root acts as a compact cryptographic commitment to the entire transaction set and is stored in the block header. As a result, any change to a single transaction modifies its leaf hash and propagates upward, producing a different Merkle root and making unauthorized modification detectable. Figure 2 demonstrates the hierarchical aggregation of transaction hashes to ensure and represent the integrity of every transaction contained within a block

**Timestamp**

In blockchain protocols, the node responsible for proposing or validating a block typically inserts a **timestamp** into the block header. The timestamp records the time at which the block was created or broadcast according to protocol rules and supports the ordering of blocks along the canonical chain. By coupling timestamps with hash-linked blocks, the ledger maintains a time-ordered history that helps establish the sequence of recorded events.

Timestamping also contributes to auditability and integrity checking. Because each block header commits to its metadata and to the preceding block, attempts to rewrite historical data would require recomputing subsequent blocks and producing a consistent alternative history that satisfies consensus constraints. Although a timestamp alone does not prevent manipulation, in combination with hash chaining and consensus validation, it strengthens the system's ability to expose inconsistencies and supports time-related checks used by many protocols, including constraints on acceptable clock drift and difficulty adjustment logic.

**Peer-to-peer network technology**

The adoption of peer-to-peer (P2P) networking closely ties the operational resilience of early blockchain systems, including the Bitcoin network launched in 2009. In contrast to centralized client-server architectures, a P2P model distributes communication and data propagation responsibilities across participating nodes. This design reduces dependence on a single service endpoint and improves robustness under node failures or targeted disruption.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue IV, April 2026

In blockchain environments, P2P networking allows transactions and blocks to be shared in a decentralized way, supports redundancy through replication, and makes it easier to spread the load across the network. Prior studies note that P2P architectures can provide advantages such as decentralization, resilience, and more balanced resource utilization, while also influencing latency, throughput, and privacy characteristics depending on the overlay design and propagation strategy (Rajasekaran, Azees, & Al-Turjeman, 2022). Conceptually, the P2P layer acts as the transport substrate that allows independent nodes to coordinate ledger updates, exchange validation messages, and converge on a shared chain state without relying on a central coordinator. Cryptocurrencies facilitate the transfer of value worldwide without depending on traditional middlemen or centrally managed server systems. . Using a distributed network architecture, participants who wish to verify and propagate new blocks can run a full Bitcoin node, which maintains and updates a local copy of the ledger (Sharma, 2022). In this setting, the blockchain functions as a decentralized, append-only record of digital asset transactions maintained through a decentralized peer-to-peer (P2P) network.

Each participating machine stores a complete replica of the ledger, maintaining consistency through continuous cross-validation among nodes to detect discrepancies and preserve integrity. This operating framework differs from conventional banking structures, in which transaction data is usually stored within privately managed databases and governed by a centralized authority.

P2P networks themselves have evolved across multiple architectural paradigms. Drawing on differences in design assumptions, historical development, and overlay organization, P2P systems are commonly grouped into three broad generations: (i) hybrid P2P networks, which combine decentralized participation with some centralized coordination; unstructured P2P networks, which support flexible connectivity but rely on comparatively inefficient discovery mechanisms; and structured P2P networks, which impose defined topologies and indexing schemes to enable more efficient routing and lookup.

**Distributed Ledger Technology and Blockchain Architecture**

The primary distinction between blockchain platforms and conventional database systems resides in the types of operations they are architected to facilitate (Chowdhury et al., 2018; Koens & Poll, 2018; Mattila, 2016). . Conventional database management systems are designed around the full set of CRUD operations—create, read, update, and delete—allowing records to be inserted, retrieved, modified, or removed as application requirements evolve. In contrast, most blockchain architectures are deliberately restrictive. They typically permit data to be appended and later retrieved, but previously recorded entries cannot be altered or erased. This append-only characteristic is central to blockchain security, as it preserves historical states and ensures that modifications require consensus at the protocol level rather than administrative intervention.

Conventional database systems are generally classified into two principal categories: centralized and distributed architectures. . In centralized systems, a single administrative authority governs data storage and access control. Distributed databases, by comparison, replicate or partition data across multiple interconnected nodes while presenting a unified logical view to applications. Such designs enhance scalability, availability, and concurrency by distributing workloads across several locations.

It is also known as distributed ledger technology (DLT). DLTs and distributed databases may seem similar at first glance, but they are very different in how they store, copy, and check data. In blockchain networks, data replication is governed by consensus protocols that ensure collective agreement on both the legitimacy and the sequential ordering of transactions The ledger is a chain of records that can't be changed and is time-stamped and in order. Along with decentralization, these features make it possible for state transitions to be clear and lower the need for a single trusted authority to keep data safe.

There are two keys that are mathematically linked and made at the same time: a public key that anyone can see and a private key that only you can see. You can give the public key to anyone, but the private key is secret and can't be easily found out from the public key using current computer methods. In this model, only the person who has the private key can decrypt data that has been encrypted with a public key. Also, anyone who has the public key that goes with a user's private key can check if that user's digital signature is real.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue IV, April 2026

**Asymmetric encryption and digital signatures**

Digital signatures represent a foundational cryptographic technique for establishing message authenticity and preserving data integrity within secure communication frameworks (Shi et al., 2020).

In a typical signing process, the sender first applies a cryptographic hash function to the original message to generate a fixed-length message digest that uniquely represents the content. The computed hash value is then transformed with the sender's private key to generate the associated digital signature. . The signed digest is appended to the original message, and both components are transmitted concurrently to the designated recipient for verification.

After reception, the recipient independently derives a new hash from the obtained message and applies the sender's public key to verify the validity of the accompanying digital signature.

The verification procedure involves comparing the newly computed digest with the digest recovered from the signature. A successful match confirms that the message has not been modified during transmission and that the signature was created by the legitimate holder of the corresponding private key. Consequently, digital signatures provide cryptographic assurance of both data integrity and source authentication, as illustrated in Fig. 3. not been tampered with, as evidenced by the consistency of the digital signature (Li, Hu & Lan, 2020).

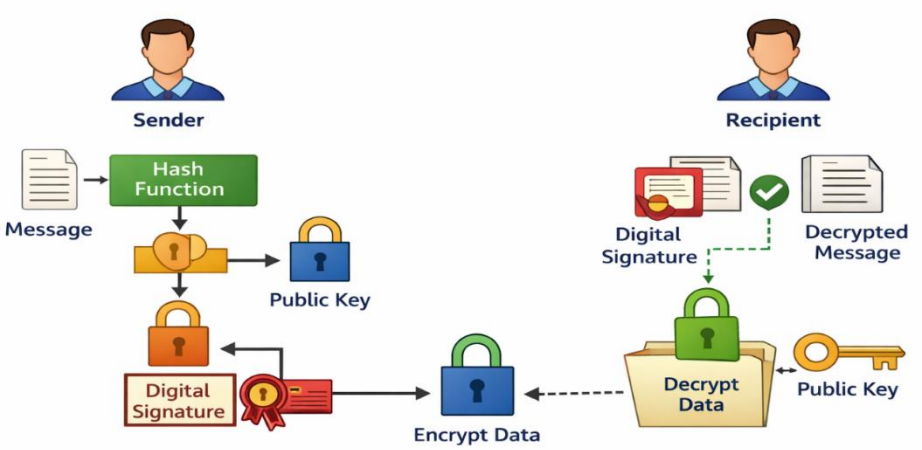
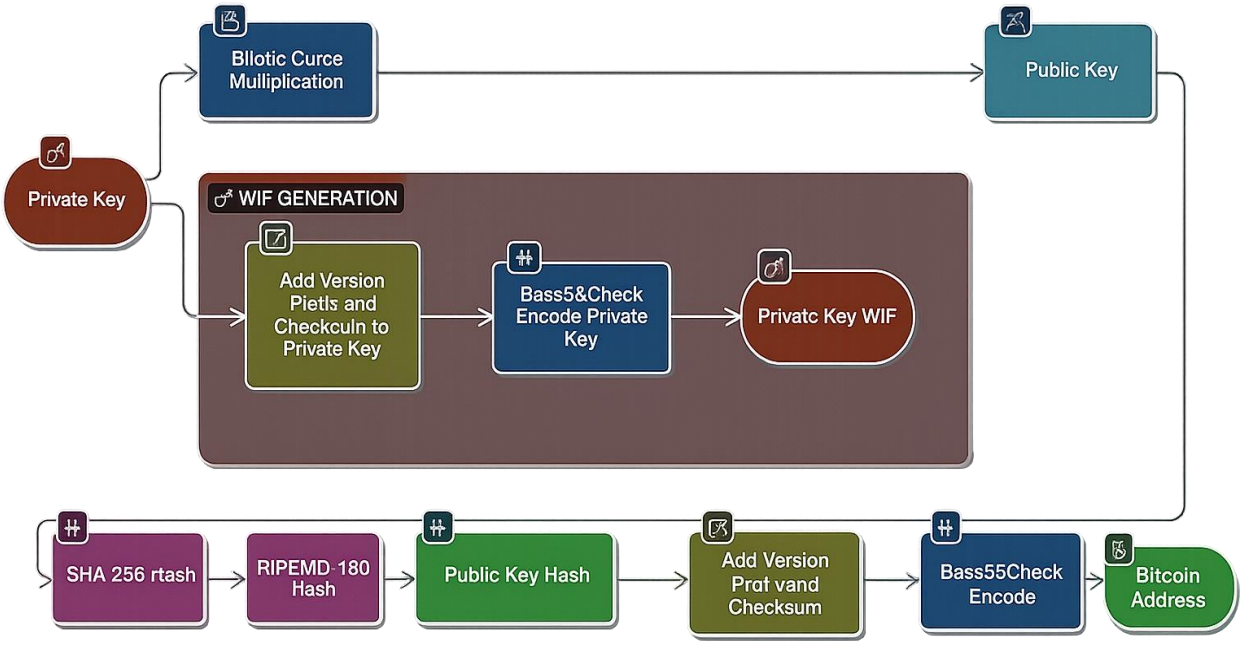



**Figure 3.** Bitcoin asymmetric encrypton mechanism



ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue IV, April 2026

**Elliptic curve cryptography**

Modern blockchain platforms primarily employ elliptic curve cryptography (ECC) due to its ability to provide robust security with relatively small key lengths, thereby enhancing efficiency in terms of bandwidth utilization, storage requirements, and computational overhead. ECC is founded on algebraic structures associated with elliptic curves, which are commonly represented in Weierstrass form over finite fields.

The security of ECC derives from the computational asymmetry between two related tasks: it is efficient to compute a scalar multiplication of a base point on the curve, yet computationally infeasible to reverse that operation at scale.

In simplified terms, ECC key generation selects a private scalar (k) and a publicly known base point (P) on the curve and computes the public key as [Q = k P], where (k) is a positive integer and (P) is the fixed generator point. Given (k) and (P), the point (Q) can be computed efficiently. However, given (Q) and (P), recovering (k) is considered intractable for appropriately chosen curve parameters.

The elliptic curve discrete logarithm problem (ECDLP) is the main computational hardness assumption that makes elliptic curve cryptographic systems safe.

. Consequently, (Q) can be distributed openly as the public key, while (k) is retained as the private key, ensuring that only the legitimate key holder can produce valid signatures or derive associated cryptographic secrets used in secure communication workflows (Umucu, 2022). In blockchain contexts, this mechanism supports identity binding, transaction authorization, and integrity protection without exposing the private signing material.

**Paxos algorithm**

The Paxos family of protocols is a foundational approach to achieving consensus in distributed systems and has been examined in blockchain and ledger settings where agreement must be maintained despite failures (Mingxiao et al., 2017; Charapko et al., 2018; Burchert & Wattenhofer, 2018; Deng et al., 2022).Paxos is formulated to enable a group of distributed nodes to reach consensus on a single proposed value, despite the occurrence of node failures or communication anomalies such as message delays and reordering. Its core contribution is a fault-tolerant coordination mechanism that preserves safety under asynchronous network conditions, typically assuming non-Byzantine failures.

Paxos is formulated to enable a group of distributed nodes to reach consensus on a single proposed value, despite the occurrence of node failures or communication anomalies such as message delays and reordering. Proposers suggest a value, acceptors vote under rules that prevent conflicting decisions, and learners observe the chosen outcome. By requiring quorums for progress, Paxos ensures that once a value is chosen, subsequent proposals cannot invalidate that decision. These properties make Paxos a common reference point for permissioned or consortium ledger designs in which participants are known and operational conditions favour crash fault tolerance over adversarial threat models.

**Paxos in crash fault-tolerant settings (non-Byzantine failures)**

Paxos was developed for environments in which nodes may fail by crashing or becoming temporarily unreachable, rather than behaving adversarial (Byzantine faults). In this crash fault-tolerant model, the protocol guarantees agreement on a single value despite message delay, reordering, and partial node failure. Conceptually, Paxos resembles a structured decision process in which proposals are introduced, evaluated, and adopted only when sufficient support is obtained from a quorum of participants.

Operationally, Paxos proceeds through coordinated phases that establish a safe ordering of proposals and prevent conflicting decisions. In a typical formulation, a proposer first enters a **prepare** stage, issuing a proposal number to solicit promises from a majority of acceptors not to accept earlier numbered proposals. If a quorum responds, the proposer moves on to the accept stage, where they ask acceptors to accept a value that follows the rules of the protocol. This value can be any value that was accepted before, as long as it has the highest proposal number

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue IV, April 2026

reported during preparation. Once a majority has accepted the proposal, the value is considered chosen and can be learned and applied by the system. If a proposer fails to collect quorum responses, or if competing proposals disrupt progress, the protocol is reattempted with a higher proposal number, continuing until a value is successfully chosen. Through quorum intersection, Paxos preserves safety under crash failures while enabling liveness under appropriate timing assumptions and retry strategies.

**SHA 256 and double hashing in Bitcoin**

For Bitcoin, the SHA 256 hash function makes summaries of transaction data that are always the same length. Then it uses it twice (double SHA 256) to turn any size input into a 256-bit output. Cryptographic hashing is useful because it (i) always gives outputs of the same length, no matter how big the input is; (ii) has a predictable computational cost for each evaluation; (iii) has strong one-way behavior that makes inversion impossible; and (iv) has avalanche properties that make small changes in input lead to very different outputs. These features make it easy to check that data is correct, quickly find transaction records, and link blocks in a way that shows if they have been changed. Bitcoin's proof of work system also uses SHA 256 evaluations over and over again to find a block header hash that meets the network's difficulty target (Ye et al., 2018).

**Consensus mechanisms in blockchain networks**

Consensus protocols are the rules that a blockchain network uses to make sure that everyone sees the ledger the same way. In the real world, nodes agree on which transactions are valid, the order in which they should be recorded, and who can add the next block. This lets people trust each other even if they don't know or trust each other directly. In many public networks, people agree to do something in exchange for money. For example, block producers get paid for following the rules of the protocol and suggesting valid blocks. In proof-of-work systems, this is often called "mining" (Wang et al., 2019b).

In practice, consensus usually means (i) choosing or finding the node(s) that suggest the next record and (ii) checking that record and adding it to the shared ledger. This is usually called leader or primary selection, followed by replication and agreement on the suggested update, in places where permissions are needed. The design must clearly allow for Byzantine behaviour when people are fighting, on the other hand. PBFT and other traditional Byzantine fault tolerant methods make the process of replication and agreement more formal by breaking it down into three steps: proposing, verifying, and committing (Castro & Liskov, 1999). This makes sure that everyone is on the same page, even if some people are bad. Since these ideas were first put into action, many different ways to come to an agreement have been suggested and tried. Some examples are proof of work (PoW), proof of stake (PoS), delegated PoS (DPoS), practical Byzantine fault tolerance (BFT) variants, and BFT protocols that are only for consortia. Some, like Ripple-style federated agreement and IOTA's tangle-based data structure, go into more detail. These systems have different ideas about how they will work, how much power they will use, how decentralized they will be, and what kinds of threats they will face. Because of this, people pick them based on how safe they need to be and where they will be used.

There are now many more consensus protocols than just classical proof of work and Byzantine fault tolerant replication. Modern surveys and system proposals talk about things like Raft for crash fault tolerant replication in permissioned deployments, the Stellar consensus protocol for federated agreement, proof of believability and other reputation weighted variants, directed acyclic graph based ledgers, Hashgraph style gossip and voting approaches, proof of authority for identity anchored validator sets, and more designs like Holochain, SPECTRE, Byteball, and LibraBFT (Chepurnoy et al., 2017; Pilkington, 2016; Underwood, 2016; Mukhopadhyay et al., 2016; Wang et al., 2019b; Amsden et al., 2020; Lashkari & Musilek, 2021; Zhang et al., 2020b; Sanka et al., 2021; Kaur et al., 2021; Yao et al., 2021; Guru et al., 2023). This variety shows both innovation and specialization: different application contexts need different trade-offs between latency, throughput, energy use, fault model, governance, and decentralization.

Because the threat landscape is changing so quickly, we need to pay attention to both the quality and the assumptions of the protocols at the same time. People who want to hurt you can now use a lot of computers and networks. This can make systems less secure if they aren't set up or managed correctly. When you choose a

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue IV, April 2026

protocol, you need to do a full security check, set it up correctly, and keep an eye on it all the time. This is especially true when blockchains are used in places where safety is important or there aren't many resources.

**PBFT consensus**

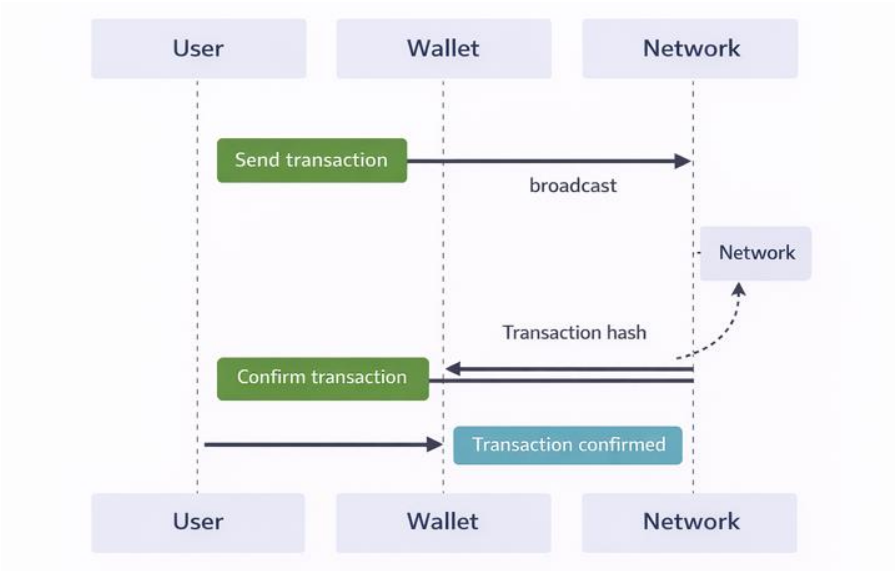
Practical Byzantine Fault Tolerance (PBFT) is a Byzantine fault tolerant state machine replication protocol that models a service as a replicated deterministic state machine operating over an ordered stream of client requests (Castro & Liskov, 1999). Compared with earlier Byzantine fault tolerant approaches, PBFT replaces combinatorial message patterns with a structured multi-phase exchange, achieving substantially improved efficiency and reducing computational and communication complexity from exponential growth to polynomial bounds under standard assumptions. In this study, PBFT is included among the benchmark consensus protocols, and its key characteristics are summarized alongside other widely used mechanisms in Table 1.

**Blockchain wallet**

A blockchain wallet is a software-based credential and transaction management interface that enables users to store cryptographic keys and to initiate, receive, and monitor transfers of digital assets on a blockchain network (Dai et al., 2018; Eyal, 2022). Rather than "holding" coins in the traditional sense, the wallet maintains the key material required to authorize spending and to derive addresses used for receiving funds. Typical implementations support multiple assets, provide mechanisms for transaction creation and signing, and expose user interfaces through web, mobile, or desktop environments. Many wallets also incorporate privacy and identity protection features, such as address rotation, hierarchical deterministic key derivation, and controlled disclosure options, depending on the underlying protocol and the wallet design.

Wallet security is primarily determined by the protection of the private key. Each wallet is associated with a public key or address, which can be shared to receive funds, and a corresponding private key, which must remain confidential because it is used to generate valid signatures for spending (Suratkar, Shirole & Bhirud, 2020). If an adversary obtains the private key, they can authorize unauthorized transfers, and recovery is typically infeasible due to the irreversible nature of blockchain transactions. For this reason, secure key storage, strong access controls, backup and recovery procedures (for example, seed phrases), and the use of hardware backed secure elements are commonly recommended in practice.

Figure-4 shows the transaction sequence at a conceptual level. The user creates a transaction through the wallet, the wallet signs the transaction using the private key and broadcasts it to the peer-to-peer network, and network nodes validate the transaction according to protocol rules. Once the transaction is accepted and included in a block, confirmation information is propagated through the network and reflected back in the wallet interface, completing the end-to-end communication flow among the user, wallet software, and blockchain network participants.



ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue IV, April 2026

When a user initiates a transfer through a digital wallet application, the software constructs the transaction and propagates it to the underlying blockchain network for validation. In accordance with the network's consensus mechanism, participating nodes independently verify the transaction by confirming its structural correctness and ensuring that the sender possesses sufficient balance. After successful validation, the transaction is incorporated into a confirmed block and appended to the ledger. The network subsequently relays the transaction confirmation to the wallet application, prompting it to refresh the user interface to reflect the updated status.

This coordinated interaction among the user, the wallet application, and the distributed ledger infrastructure enhances the dependability of cryptocurrency transactions by reducing reliance on a centralized intermediary while preserving transactional integrity and transparency.

**Blockchain algorithms for IOT security**

Recent growth in Internet of Things infrastructure, including smart buildings and smart cities, continues to expose two persistent concerns. First, stakeholders often lack a uniform basis for trust across heterogeneous devices, operators, and service domains. Second, centralized security and coordination components can introduce a single point of failure that may disrupt system availability and compromise data integrity. In this context, blockchain can provide a decentralized coordination layer that improves resilience by distributing record keeping and validation across multiple nodes.

Blockchain's append-only ledger and distributed replication enable participants to verify interactions without depending on a single authority. Each interaction can be captured as a timestamped transaction, creating an auditable sequence of events that supports both accountability and anomaly detection. In IoT environments, transactions may represent user movement across controlled zones, secure data exchange among devices and users, device telemetry submissions, user-to-device access logs in urban or building deployments, service coordination across organizations, and device-to-device communication events. By preserving these records across multiple nodes, the system becomes more robust to tampering and operational failures, while authorized participants can trace the provenance of data and actions within the network. For large-scale IoT deployments with diverse nodes and trust boundaries, these properties can strengthen secure communication and coordinated governance (Roman, Zhou & Lopez, 2013; Agrawal et al., 2018; Alam, 2019).

**Cryptographic methods used to secure IoT communication in blockchain systems**

Blockchain-supported IoT security commonly relies on cryptographic primitives that protect integrity, authenticity, and confidentiality.

**Hashing algorithms**

- Secure Hash Algorithm 256-bit (SHA 256): Widely used in prominent blockchain systems, SHA 256 produces a fixed-length digest that enables efficient integrity checks. Any modification to the underlying data yields a different digest, which supports tamper detection and reduces the feasibility of collision-based manipulation.

- Secure Hash Algorithm 3 (SHA 3): Built on the Keccak family, SHA 3 provides a modern alternative with a distinct internal design and strong resistance properties, offering an additional option for integrity protection in blockchain-enabled IoT settings.

**Asymmetric cryptography algorithms**

- Rivest-Shamir-Adleman (RSA): RSA is used for public key encryption and digital signatures, supporting authenticated communication and secure key exchange among IoT participants interacting through blockchain-based infrastructures.

Elliptic Curve Cryptography (ECC): ECC achieves comparable security to traditional public key schemes with shorter keys, which reduces computational and storage overhead. This benefit is especially significant for

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue IV, April 2026

resource-constrained IoT devices, which need to maintain robust security while operating under limited computational capacity and energy resources.

**Zero-knowledge proofs**

his benefit is especially relevant for resource-limited IoT devices, which must maintain strong security despite restrictions in computing power and energy availability.

This capability is increasingly used to support privacy-oriented authentication and transaction validation, including in Internet of Things settings where device identities and operational data should not be exposed beyond what is strictly necessary.

- **ZK-SNARKs (Zero Knowledge Succinct Non-Interactive Argument of Knowledge):** zk-SNARKs provide succinct proofs that can be verified efficiently, allowing a party to prove knowledge of a secret witness while keeping the witness confidential. In blockchain-enabled IoT deployments, this property can support privacy-oriented access control, device attestation, and transaction validation with limited information disclosure.

- **ZK STARKs (Zero Knowledge Scalable Transparent Argument of Knowledge):** ZK STARKs extend the zero-knowledge proof paradigm with improved scalability characteristics and transparent setup assumptions. They are often positioned as an alternative for use cases where proof generation and verification must scale to larger computations while maintaining verifiability and limited disclosure.

**Smart contract languages**

Smart contracts operationalize business logic as deterministic programs executed by blockchain nodes, enabling automated enforcement of rules, conditional transfers, and event-driven workflows.

- **Solidity:** It is a widely recognized programming language created for developing smart contracts on the Ethereum blockchain. It supports the development of decentralized applications by allowing programmers to define governance mechanisms, handle digital assets, and execute automated workflows directly on-chain.

- **Vyper:** Vyper provides an Ethereum-compatible alternative that emphasizes readability and a reduced feature set intended to support safer contract design and auditing practices.

**Privacy-oriented algorithms**

In addition to zero-knowledge proofs, blockchain systems employ dedicated privacy mechanisms to limit the revelation of sensitive transaction data

- **Confidential transactions (for example, Bulletproof-based constructions):** These approaches conceal transaction amounts while preserving correctness through cryptographic proofs. They are used to enhance financial privacy by preventing observers from inferring value flows while still enabling network-wide validation of transaction rules.

**Interoperability protocols**

Interoperability mechanisms aim to reduce fragmentation across blockchain networks by enabling asset and data exchange without requiring a centralized intermediary.

- **Atomic swaps:** Atomic swap protocols enable direct cross-chain exchange of assets using cryptographic conditions that ensure either both transfers succeed or both fail. This supports peer-to-peer exchange across distinct ledgers without relying on trusted escrow services.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue IV, April 2026

- **Polka dot:** Polka dot is a multi-chain framework designed to support cross-chain messaging and interoperability among heterogeneous blockchains, enabling coordination across specialized networks through a shared security and communication model.

**Security Measures in Blockchain Systems: Understanding Attack Dynamics**

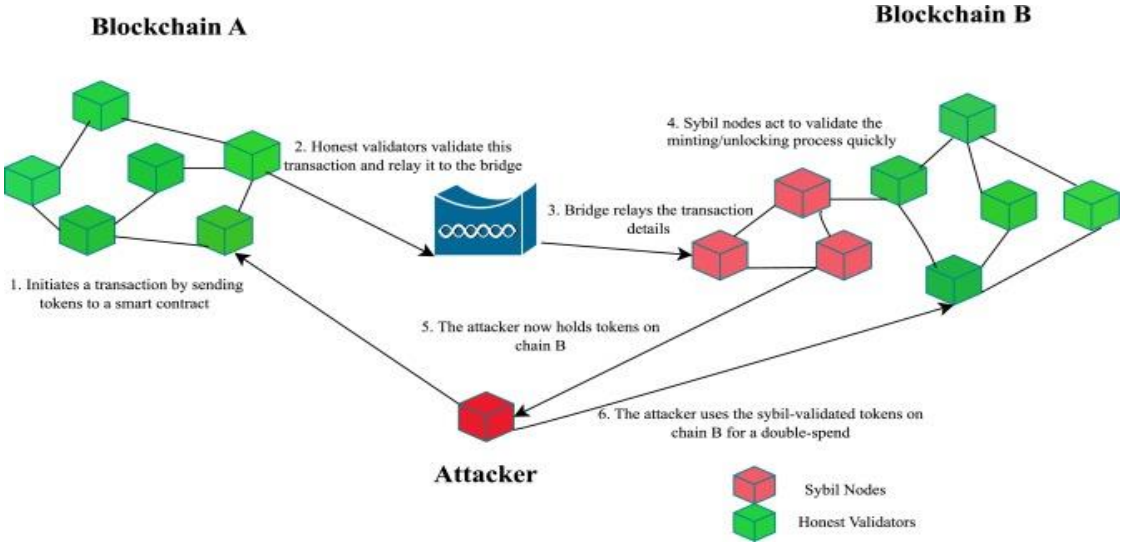
Blockchain security refers to the set of strategies, protocols, and technical controls designed to prevent unauthorized access, data leakage, and adversarial manipulation. In most security assessments, the central objectives are integrity, confidentiality, and availability of the network and its data. Accordingly, security measures are evaluated by how effectively they mitigate known attack vectors and preserve the reliability and verifiability of on chain records. The principal security measures and their corresponding threats are summarized in Table 3.

**Security**

The security of blockchain systems is primarily derived from decentralized operation, distributed consensus, and cryptographic data structures that constrain unilateral changes. Because the ledger is maintained across a network of participating nodes, control over data is not concentrated in a single authority, which reduces the risk of single point compromise. Consensus mechanisms further strengthen this model by requiring collective validation of proposed transactions and blocks, thereby limiting the ability of a malicious participant to introduce invalid state transitions without sufficient network influence.

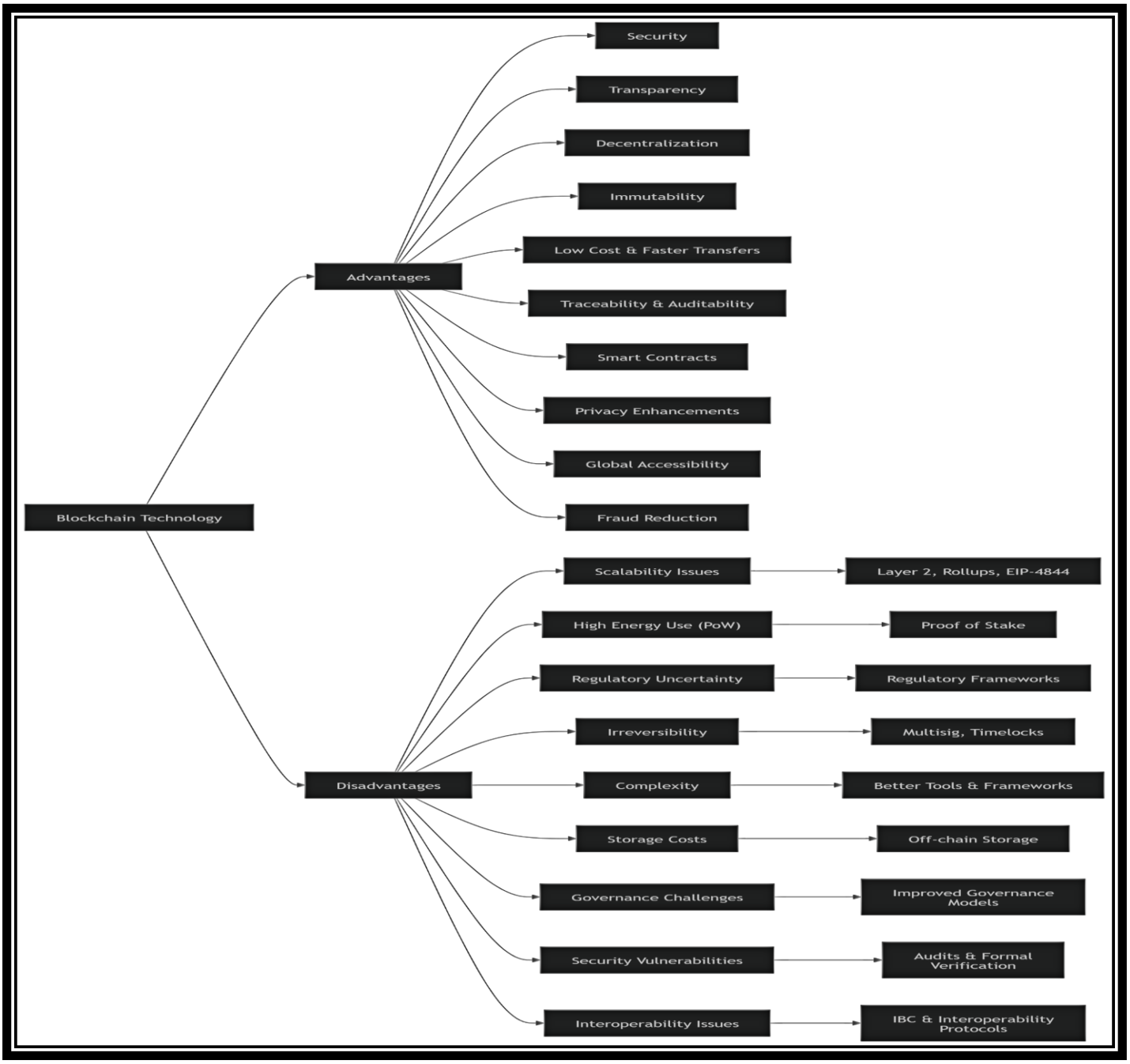
Once transactions are confirmed and appended to the ledger, cryptographic linking between blocks provides strong resistance to retrospective modification. Any attempt to alter historical records would require recomputing dependent cryptographic values and overcoming the network's agreement process, which is computationally and operationally prohibitive under standard assumptions. Access control within blockchain interactions is typically enforced through public key cryptography, where participants authorize actions by signing transactions with private keys. Consequently, the confidentiality and safety of private keys become a foundational requirement for end user security and operational trust (Alangot et al., 2020). Transparency also contributes to security by enabling auditing and continuous monitoring. Since transaction histories can be inspected and verified, suspicious behaviours can be detected earlier, and accountability can be strengthened through traceable evidence, depending on the permission model and privacy design of the system (Sun, Zhang, and Han, 2023). Building on these principles, the following section outlines major attack categories and summarizes the tactics and decision dynamics commonly modelled for honest miners and adversaries in security assessment frameworks (Zeng et al., 2019). Understanding these mechanisms is essential for interpreting security guarantees and for selecting appropriate controls for specific blockchain deployments.

Below is an enhanced and updated version of your points, with current mitigation patterns used in modern blockchain deployments.



ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue IV, April 2026

**Advantages and Disadvantages**




**Security measures in blockchain systems, benefits, limitations, and current enhancements**

| Dimension | Security value or advantage | Key limitation or risk | Recent enhancement or mitigation pattern |
|---|---|---|---|
| Security (cryptography + distributed validation) | Protects data integrity and transaction authenticity through signatures, hashing, and consensus | Private key loss or theft, wallet compromise, flawed implementations | Smart wallet controls such as account abstraction patterns, policy checks, and stronger wallet recovery design. (docs.erc4337.io) |
| Transparency | Shared ledger supports consistent verification and easier auditing | Sensitive metadata exposure, link ability, compliance concerns | Per missioning, selective disclosure designs, and privacy preserving proof systems where needed |

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue IV, April 2026

| | | | |
|---|---|---|---|
| Decentralization | Reduces single point of failure and limits unilateral manipulation | Slower coordination, governance complexity, validator centralization risk | Increased client and validator diversity plus clearer governance and upgrade processes |
| Immutability | Tamper resistance for confirmed records strengthens evidence quality | Irreversible errors, difficult remediation after fraud or mistakes | Time lock patterns, multi signature approvals, dispute workflows, and layered controls before final settlement |
| Fast and low friction settlement (peer to peer) | Fewer intermediaries can reduce delays and coordination overhead | Congestion can raise fees and delay confirmations | Layer 2 rollups plus cheaper rollup data posting via Proto Danks Harding (EIP 4844) activated in Duncan. (ethereum.org) |
| Scalability and throughput | Enables multi party data sharing on a common ledger | Limited throughput on many base layers, latency for finality | Rollup centric scaling and data availability upgrades such as EIP 4844 blobs to lower rollup operating cost. (ethereum.org) |
| Energy consumption | Security can be maintained without heavy energy usage in modern designs | Proof of work systems can be energy intensive | Proof of stake adoption, for example Ethereum after the Merge with energy reduction reported around 99.95%. (ethereum.org) |
| Traceability and auditability | Time stamped history improves provenance and investigations | Privacy trade-offs, surveillance risk, regulatory constraints | Role based access in permissioned networks, selective disclosure proofs, and robust analytics controls |
| Data storage | On chain records improve non repudiation for critical events | Large data is expensive and impractical to store on chain | Store hashes on chain and keep bulk data off chain, with temporary blob style data for rollup posting under EIP 4844. (ethereum.org) |
| Smart contracts | Automated enforcement of rules and workflows reduces manual intervention | Contract bugs, upgrade risk, oracle risk | Secure development lifecycle, independent audits, formal verification for critical logic, safer upgrade patterns |
| Regulation and misuse controls | Clearer rules can increase institutional trust and consumer protection | Regulatory uncertainty, jurisdictional fragmentation, potential misuse | Formal regimes such as EU MiCA with staged application dates (30 June 2024 for Titles III and IV, 30 December 2024 for general application). (EUR-Lex) |
| Interoperability | Cross network workflows expand utility beyond a single chain | Bridge attacks and inconsistent trust assumptions | Standardized protocols such as IBC for authenticated data transport between chains that implement the required interfaces. (docs.cosmos.network) |

**Various Types of Attacks in Blockchain Systems**

In blockchain environments, an attack refers to any intentional action aimed at undermining the integrity of the distributed ledger, disrupting system availability, or violating the security guarantees provided by consensus protocols and cryptographic mechanisms. The threat landscape spans multiple layers, including protocol-level exploits (e.g., majority control and network partitioning) and user- or application-level attacks such as phishing, private key compromise, and ransomware-based extortion. Adversaries may attempt to steal digital assets, censor transactions, gain unauthorized access, or degrade overall network performance.

To address these risks, contemporary blockchain architectures integrate robust consensus algorithms, advanced cryptographic techniques, secure communication protocols, and continuous monitoring frameworks to minimize

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue IV, April 2026

vulnerabilities and enhance resilience. The following subsections summarize widely recognized attack vectors discussed in the literature.

**Majority (51%) Attack**

A majority attack happens when one person or a group of people working together takes control of more than half of the network's ability to generate blocks. In Proof-of-Work (PoW) systems, this capability is directly associated with computational hash power. Such dominance enables the attacker to disproportionately influence chain growth, increasing the likelihood of extending a preferred chain over the honest one.

This control allows adversaries to censor transactions and potentially execute double-spending by reorganizing previously confirmed blocks. The feasibility of this attack depends on several factors, including the cost of acquiring computational resources, network propagation delays, and the responsiveness of honest participants.

The probability of a successful chain reorganization can be conceptually expressed as:

- If the attacker's resource share q≥pq \geq pq≥p, the probability of catching up approaches 1
- If q < pq < p, the probability decreases exponentially with confirmation depth zzz, approximately proportional to (q/p)z(q/p)^z(q/p)z

where ppp and qqq represent the probabilities of block generation by honest nodes and the attacker, respectively. This relationship highlights that deeper confirmation thresholds significantly reduce attack success probability.

**Double-Spending Attack**

Double spending involves attempting to use the same digital asset in more than one transaction. Typically, an attacker sends a payment to a recipient while simultaneously preparing a conflicting transaction that redirects the same funds elsewhere. If the adversary succeeds in having the conflicting transaction confirmed—either through majority influence or favorable network conditions—the original transaction may be invalidated.

Blockchain systems mitigate this risk through strict validation rules and consensus mechanisms that enforce transaction uniqueness. Additionally, the probability of reversing a transaction decreases as more blocks are appended after it, reinforcing the importance of confirmation depth.

**Cryptographic Attacks**

Cryptographic attacks target the foundational security mechanisms of blockchain systems, including digital signatures, hashing, and encryption. In practice, direct cryptographic breaks are highly improbable due to the computational strength of modern algorithms.

However, vulnerabilities often arise from poor key management practices, insecure storage, weak randomness in key generation, or flawed implementations. Side-channel attacks and software bugs can also expose sensitive cryptographic material, leading to potential data breaches and unauthorized access to encrypted information. Consequently, operational security and implementation correctness remain critical factors in mitigating these threats.

**Denial-of-Service (DoS) Attacks**

Denial-of-service attacks aim to reduce system availability by overwhelming network resources or disrupting communication among nodes. These attacks can significantly impact throughput and increase transaction confirmation delays. Common variants include

- **Transaction flooding:** Excessive submission of transactions to congest mempools and delay processing
- **Distributed DoS (DDoS):** Coordinated traffic from multiple sources targeting node infrastructure

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue IV, April 2026

- **Resource exhaustion:** Exploiting limitations in bandwidth, CPU, memory, or storage

Mitigation strategies include rate limiting, adaptive fee mechanisms, peer filtering, load balancing, and protocol-level safeguards that increase the cost of malicious activity.

**Sybil Attack**

In a Sybil attack, an adversary generates numerous fake identities to gain disproportionate influence within the network. An adversary may use these identities to manipulate peer discovery, disrupt routing, or bias consensus-related processes.

Such attacks can facilitate network partitioning, transaction censorship, or even double-spending under certain conditions. Preventive measures include resource-based entry barriers (e.g., PoW or Proof-of-Stake), identity verification in permissioned systems, and enforcing diversity in peer connections.

**Ransomware Exploiting Cryptocurrencies**

Campaigns like WannaCry and Petya demonstrate the use of cryptocurrencies in illicit financial transactions. These attacks encrypt victims' data and demand payment—typically in cryptocurrencies—due to their global accessibility and pseudonymous nature.

It is important to note that these incidents do not compromise blockchain protocols directly. Instead, they exploit the properties of cryptocurrencies as payment systems, highlighting the broader security implications of their use in cybercrime, such as the challenges in tracing illicit transactions and the potential for increased criminal activity.

**Transaction Malleability**

Transaction malleability refers to the ability to alter a transaction's representation without changing its underlying intent. This modification can result in a different transaction identifier (hash), potentially causing confusion in systems that rely on transaction IDs before confirmation.

Such discrepancies may lead users or services to mistakenly assume that a transaction has failed, prompting duplicate submissions. Robust wallet design and careful handling of transaction references are essential to mitigate this issue.

**Time jacking Attack**

Time jacking exploits discrepancies in node time synchronization. By manipulating timestamps or peer-reported times, an attacker can influence how a node validates blocks or selects peers.

Such changes can lead to incorrect acceptance of blocks or network desynchronization. Countermeasures include restricting acceptable time deviations, using secure time sources, and implementing consistency checks across peers.

**Routing and Network Partition Attacks**

Routing attacks aim to disrupt communication by isolating subsets of nodes or delaying message propagation. By controlling network paths or exploiting routing protocols, attackers can create inconsistencies in the distributed ledger view.

Such isolation can facilitate double-spending attempts or reduce overall network reliability. Mitigation approaches include maintaining diverse peer connections, employing multi-path data propagation, and

monitoring network anomalies.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue IV, April 2026

**Delay and jellyfish attacks**

Delay-based attacks, including jellyfish-style behavior, aim to selectively delay packet forwarding or block propagation while maintaining the appearance of normal participation. The attacker can degrade consensus performance and increase confirmation times without immediately triggering simple failure detection. Mitigation commonly relies on peer performance scoring, propagation redundancy, prioritization policies for critical messages, and improved time synchronization to reduce exploitability in time-sensitive workflows (Zhong and Guo, 2021).

**Eclipse attack**

An eclipse attack isolates a target node by monopolizing its inbound and outbound peer connections, forcing it to receive a curated view of the network. Once isolated, the victim can be fed delayed blocks, censored transactions, or a manipulated chain view that supports downstream attacks such as double spending against services that rely on the victim's confirmation signals. Prevention includes peer diversity rules, anti-monopolization controls in peer selection, limiting connections from a single network range, and secure peer rotation practices (Singh and Singh, 2016).

**Phishing and social engineering**

Phishing attacks exploit human trust to obtain private keys, seed phrases, or authentication credentials for exchanges and wallets. Attackers often impersonate legitimate services through deceptive messages and counterfeit websites. Once credentials are captured, funds can be transferred irreversibly. Recent reports also describe browser-based and transaction manipulation variants that target user workflows and approval habits (Katte, 2022; Katte, 2023). Common forms include spear phishing (targeted phishing attacks), whaling (phishing attacks aimed at high-profile individuals), clone phishing (replicating a legitimate email with malicious links), pharming (redirecting users from legitimate websites to fraudulent ones), malicious Wi-Fi traps (rogue Wi-Fi networks that capture data), voice phishing (using phone calls to trick individuals), and SMS phishing (sending fraudulent text messages).

**Vulnerable signatures**

Signature-related vulnerabilities arise when attackers can forge, replay, or reuse signatures due to weak algorithms, improper nonce handling, flawed validation, or insecure implementation. If signatures can be replicated or manipulated, attackers may authorize unauthorized transfers or impersonate users. Mitigation requires robust signature schemes, secure implementation practices, constant-time operations where relevant, strict verification rules, and regular audits to identify cryptographic misuse and boundary condition errors.

**Dictionary-style credential attacks in blockchain ecosystems**

Traditional dictionary attacks target weak passwords through systematic guessing. In blockchain systems, direct guessing of private keys is computationally infeasible under standard cryptographic assumptions. However, dictionary-style behavior remains relevant where users rely on weak passwords for wallets, keystores, exchange accounts, or encrypted backups, and where attackers attempt common phrases or known patterns. The operational implication is that secure password hygiene, strong encryption for stored keys, multi-factor authentication, and hardened recovery processes remain essential, even when the underlying private key cryptography is strong (Tosh et al., 2017; Houy, Schmid, and Bartel, 2024).

**Flawed key generation and weak key management**

Weak randomness, outdated key generation libraries, or negligent key rotation can produce predictable keys that attackers can recover. In addition, poor storage practices, such as saving keys in plaintext or exposing them to malware, can lead to compromise. Mitigation requires high-quality entropy sources, vetted cryptographic libraries, secure key lifecycle governance, hardware-backed storage where possible, and incident response procedures for suspected compromise.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue IV, April 2026

**Attacks on hot wallet key storage**

Attacks on hot wallet storage target private keys maintained in online environments, including web wallets, exchange hot wallets, and application-connected keystores. Because these keys remain reachable through network-exposed systems, they are vulnerable to remote exploitation, malware, credential theft, and insider threats. A standard mitigation approach includes minimizing hot wallet exposure, using multi-signature authorization for high-value transfers, adopting hardware security modules, enforcing role-based access controls, monitoring abnormal withdrawal behavior, and maintaining cold storage reserves for the majority of assets.

Attackers may exploit weaknesses in wallet or application components that handle key material to obtain unauthorized access to private keys. Once key control is achieved, the adversary can authorize fraudulent transfers, censor or reorder transactions within the application workflow, or manipulate contract interactions in ways that result in direct financial loss or integrity violations (Moubarak, Filiol, and Chamoun, 2018).

**(i)Definition of the attack strategy.** Although many blockchain systems provide pseudonymity by design, this property does not guarantee that end users can reliably identify subtle integrity violations at the application layer, particularly when validation and signing are delegated to software components. In adversarial settings, the attacker may possess an enhanced view of the evolving block and transaction state, including privileged observations of candidate blocks, pending transaction queues, or local confirmation status. Using this information advantage, the attacker can selectively participate in block propagation or chain extension to maximize the probability that a fraudulent transaction set is confirmed under favorable conditions.

Formally, let ({B}) denote the set of candidate blocks visible at time (t), and let ({B}_t) represent the attacker's observation of block state, for example, composition of transactions, timing, and local acceptance. The attacker's selection strategy can be modeled as a policy function.

$\pi A:\sigma(Bt)\rightarrow at$,where the action ($a_t$) may include joining a specific candidate block, extending a preferred branch, withholding a competing block, or releasing a block to influence network convergence. Under this model, the attacker preferentially selects the newly created block whose state best supports the fraudulent objective, for example, by positioning a malicious transaction behind a block that increases acceptance likelihood or reduces the chance of immediate detection.

Attack model and node selection rules

(i)Attacker strategy. Let Rt denote the currently linked reference block selected by the adversary at time t, where each block is modeled as a node in a tree-structured view of the blockchain. The function child(·) is used to determine whether a node has descendants and, in implementation, to identify candidate attachment points for a newly created block. If an attacker-generated block already exists on the chain, the adversary attaches the new block to the longest continuation that extends beyond the attacker block in order to maximize the probability of acceptance. If no attacker block is present, the adversary attaches to the comparatively longest available branch. This rule reflects a rational objective: increase inclusion likelihood while positioning fraudulent content where it benefits from the strongest apparent chain support (Zeng et al., 2019).

(ii) Honest miner strategy. For an honest miner, the internal type or origin of a candidate block is not observable. The protocol treats the longest valid chain as the authoritative state, and only the transaction set contained in that chain is ultimately recognized. From a probabilistic perspective, an honest miner may temporarily connect a newly generated node to any eligible block, but convergence occurs toward the longest leaf node as information propagates and consensus stabilizes. When multiple leaf nodes share the same chain length, selection is typically modeled as a uniform random choice among those leaves. Under the tree model used in Eq. (4) and Eq. (5), the probability of selecting a node decreases geometrically as the node is closer to the root. The total probability mass across all candidate nodes sums to one:

where L denotes the total chain length, and p is the probability of selecting the longest leaf node, determined by the current network state and the distribution of visible branches. In practical deployments, the probability that an honest miner selects an internal node before reaching a leaf is typically small because miners aim to extend

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue IV, April 2026

tips that maximize confirmation utility and reduce orphan risk (Zeng et al., 2019). Consequently, as node depth decreases by one level, the selection likelihood decreases by approximately a factor of two, consistent with the geometric weighting in Eq. (5).

Transparency and operational constraints

Distributed bookkeeping, full replication across participating nodes, and traceable transaction histories collectively enable end-to-end inspection of blockchain interactions, which strengthens transparency and auditability. At the same time, blockchain systems remain technically complex, drawing on cryptography, distributed systems, and computational mathematics (Garay, Kiayias, and Leonardos, 2015). In many settings, adoption is constrained by the limited availability of skilled personnel and practical limitations, including energy-intensive proof-of-work designs, restricted block capacities, and longer confirmation times. Further challenges include privacy exposure caused by transparent data flows, integration friction with existing enterprise systems, and unresolved legal and regulatory questions that require sustained research attention (Cui, 2022).

Blockchain volume and storage burden

In many blockchain architectures, each full node stores the complete ledger, which improves verifiability but increases storage and computation burdens as the ledger grows. As chain size expands, new participants face higher initial synchronization costs because they must download and validate a large volume of historical data before fully participating in validation and monitoring. Prior reports noted that the Bitcoin blockchain exceeded tens of gigabytes by 2019 and that initial synchronization could require multiple days using a full client under typical network conditions (Liu and Zou, 2019). More recent public estimates have reported sizes at the scale of several hundred gigabytes by 2023, illustrating the persistent growth trend and the corresponding pressure on storage and bandwidth resources (de Best, 2023).

Introduction to future research methods and application areas

The acceleration of new information technologies is occurring alongside rapid progress in blockchain policy initiatives, protocol engineering, and real-world deployment environments. Many countries are investing heavily to secure strategic advantages in frontier technologies, which intensifies competition and motivates experimentation in scalable and compliant blockchain infrastructures. Because blockchain supports programmable, distributed, chronological, encrypted, and tamper-resistant record keeping, financial and industrial research communities continue to explore it as a foundation for new trust architectures and cooperative platforms (Huckle et al., 2016).

Characteristics of systems that benefit from blockchain

Systems most likely to benefit from blockchain typically share several properties. They involve multiple stakeholders who do not fully trust one another yet must coordinate around a shared record. They require frequent transactions or data exchanges, enforceable digital ownership, and reliable audit trails. They often benefit from unique digital identifiers, decentralized naming, and mechanisms that reduce manual dispute resolution. In regulated contexts, continuous or near real-time oversight may be required, along with complete provenance for assets and actions.

Suitable application domains

Blockchain is well suited for scenarios that require secure, transparent, and decentralized coordination of transactions or data. Reported domains include supply chain management, finance, healthcare, digital identity, voting, and real estate, among others (Levy, 2022; Zheng et al., 2018b; Dai, Zheng, and Zhang, 2019). Representative applications include:

a. Agricultural quality and safety traceability: Recording production and distribution events enables provenance tracking from origin to consumer, supporting integrity and accountability (Hua et al.,

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue IV, April 2026

2018; Srivastava, Zhang, and Eachempati, 2023).

b. Education: Credential verification, student credit management, and qualification certification can benefit from immutable records and verifiable proofs, subject to governance and privacy constraints (Budiharso and Tarman, 2020; Terzi et al., 2021; Yin et al., 2022).

c. IoT and blockchain convergence: Blockchain can reduce reliance on centralized servers by enabling distributed verification and can strengthen integrity for device interactions, although device constraints and lifecycle management remain open challenges (Khan and Salah, 2018; Christidis and Devetsikiotis, 2016; Samaniego and Deters, 2016).

d. Energy and transactive systems: Blockchain can support peer coordination in distributed energy trading and improve auditability of energy transactions, while performance and regulatory alignment remain critical (Münsing, Mather, and Moura, 2017; Bergquist et al., 2017; Lundqvist, de Blanche, and Andersson, 2017).

e. Digital identity: Blockchain-based identity models can support verification while reducing exposure of raw personal data through hash-based validation and asymmetric cryptography (Dunphy and Petitcolas, 2018; Takemiya and Vanieiev, 2018).

f. Finance: Research emphasizes transaction performance, privacy protection, and automation of financial agreements through smart contracts (Peters and Panayi, 2016; Egelund Müller et al., 2017; Momtaz, Rennertseder, and Schröder, 2019).

g. Healthcare: Secure sharing of electronic medical records, research data sharing, drug supply chain integrity, and insurance claims processing are frequently cited applications, with strong requirements for privacy, access control, and compliance (Gordon and Catalini, 2018; Benchoufi, Porcher, and Ravaud, 2017; Tseng et al., 2018; Zhou, Wang, and Sun, 2018).

h. Government: Blockchain can support secure digital services, identity verification, and tamper-resistant registries, and it is studied for voting systems where integrity and transparency are critical (Batubara, Ubacht, and Janssen, 2018; Pawlak, Guziur, and Poniszewska Marańda, 2019; Ramya et al., 2019).

i. AI and blockchain: Blockchain can improve provenance tracking for models and data, while AI can enhance anomaly detection and operational optimization in distributed networks (Sarpatwar et al., 2019; Ogundokun et al., 2022). Decentralized AI extends this concept by distributing control and verification across multiple nodes to improve transparency and accountability (Cao, 2022; Adel, Elhakeem, and Marzouk, 2022).

j. Big data: Blockchain can support secure data sharing, integrity verification, and immutable event logging for analytics pipelines, while scalability and storage costs require careful design (Chen and Xue, 2017; Abdullah, Hakansson, and Moradian, 2017).

k. Payments, lending, and insurance: Cross-border transfers and automated lending workflows can be supported using smart contracts, and insurance can benefit from tamper-resistant claims records and improved fraud detection, subject to governance and accountability constraints (Hashemi Joo, Nishikawa, and Dandapani, 2020; Chen et al., 2018; Raikwar et al., 2018; Saldamli et al., 2020).

l. Proof of ownership and intellectual property: Timestamped records and cryptographic proofs can support ownership verification and rights management, including tracking usage and automated royalty distribution (Gürkaynak et al., 2018; Tsai et al., 2017; Wang et al., 2019a).

m. Maritime supply chain management: Global logistics can benefit from provenance and data-sharing mechanisms integrated with IoT and analytics (Jiaguo Liu and Zhen, 2023).

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue IV, April 2026

Limitations of blockchain technology

Despite its potential, blockchain adoption is constrained by several technical and operational limitations (Yaga et al., 2018; Malik et al., 2019; Shen et al., 2022; Taherdoost, 2022):

1. Interoperability: Differences in protocols, data structures, and smart contract environments hinder cross-chain data exchange and multi-network applications. Active research seeks to improve interoperability, but it remains a key barrier (Belchior et al., 2021).

2. Distributed operation complexity: Maintaining consensus and synchronization across widely distributed participants can be difficult, especially under network latency and adversarial conditions (Patel et al., 2020).

3. Private key risk: Private keys are central to authorization and integrity. Loss results in permanent loss of access, while exposure enables theft. Keys are not practically reversible or changeable in the same way as traditional passwords, which elevates operational risk (Malik et al., 2019).

4. Accountability without intermediaries: The absence of a trusted third party reduces dependence on institutions, but it can complicate dispute resolution and responsibility assignment when failures occur (Halaburda, 2018; Gamage, Weerasinghe, and Dias, 2020).

5. High cost: Skilled personnel, application development, audits, and supporting infrastructure can make blockchain deployment capital intensive (Zhang et al., 2020a; Alammary et al., 2019).

6. Workflow mismatch: Blockchain is effective for environments that require frequent state changes and shared verification. It may be unnecessary or inefficient for workflows that do not benefit from replicated consensus (Evermann and Kim, 2019; Lokshina, 2022).

7. Anonymity and misuse: Pseudonymity can protect privacy but also increases misuse risk, including laundering concerns, which creates compliance challenges (Andola et al., 2021).

8. Immutability tradeoffs: Immutability strengthens integrity, but it can conflict with use cases that require updates to metadata or legally mandated changes, which motivates research into controlled mutability mechanisms (Politou et al., 2019; Hughes et al., 2019; Stančić and Bralić, 2021).

9. Storage growth: Full replication leads to rapid ledger expansion, increasing long-term storage, bandwidth, and synchronization requirements (Xu et al., 2020; Jia et al., 2021; Zhang et al., 2021).

10. Real-time monitoring and provenance needs: Many deployments require continuous oversight and complete asset histories. While blockchains can provide provenance, operational monitoring and regulatory reporting still require robust off-chain tooling and governance (Helo and Shamsuzzoha, 2020; Zheng et al., 2018a).

## CONCLUSIONS

Blockchain technology offers a strong foundation for secure, transparent, and decentralized record keeping and transaction coordination. Smart contracts extend this foundation by enabling programmable execution of business rules and automated settlement. Evidence across multiple domains suggests meaningful potential in finance, supply chains, healthcare, identity, and governance-oriented services, particularly where multiple parties require a shared, verifiable source of truth. However, large-scale deployment remains constrained by interoperability gaps, key management risk, storage growth, performance limits, and legal and regulatory uncertainty. Continued research is required to improve scalability, privacy-preserving verification, secure integration with legacy infrastructures, and governance models that provide accountability without undermining decentralization.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue IV, April 2026

# REFERENCES


1. Karim MM, Van DH, Khan S, Qu Q, Kholodov Y. 2025. AI Agents Meet Blockchain: A Survey on Secure and Scalable Collaboration for Multi-Agents. Future Internet 17(2):57. DOI: 10.3390/fi17020057.
2. Ning W, Zhu Y, Song C, Li H, Zhu L, Xie J, Chen T, Xu T, Xu X, Gao J. 2024. Blockchain-Based Federated Learning: A Survey and New Perspectives. Applied Sciences 14(20):9459. DOI: 10.3390/app14209459.
3. Zheng C, Tao X, Dong L, Zukaib U, Tang J, Zhou H, Cheng JCP, Cui X, Shen Z. 2024. Decentralized artificial intelligence in construction using blockchain. Automation in Construction 166:105669. DOI: 10.1016/j.autcon.2024.105669.
4. Li W, Chen Z, Yu Y, Luo B, et al. 2025. Blockchain interoperability: A systematic review. Information and Software Technology 180:107767. DOI: 10.1016/j.infsof.2024.107767.
5. Jain S, Prakash O, Kumar A, Rashid M. 2025. Scalable consensus algorithms in blockchain: A comprehensive survey. Blockchain: Research and Applications 3:100065. DOI: 10.1016/j.bcra.2024.100065.
6. Liu J, Ning Z, Zhang H, et al. 2025. A comprehensive survey of blockchain consensus mechanisms. Computer Networks 246:110608. DOI: 10.1016/j.comnet.2024.110608.
7. Wu H, Yao Q, Liu Z, Huang B, Zhuang Y, Tang H, Liu E. 2025. Blockchain security threats and attacks: Classification and solutions. Computer Networks 265:111284. DOI: 10.1016/j.comnet.2025.111284.
8. Sahraoui N, Bachir S. 2025. Lightweight consensus mechanisms in the Internet of Blockchained Things: Thorough analysis and research directions. Digital Communications and Networks (in press). DOI: 10.1016/j.dcan.2024.12.007.
9. Shujaa S, Salah K, Ahmad RW, et al. 2025. Integration of blockchain and IoT in security: A survey. Frontiers in Computer Science 7:1670473. DOI: 10.3389/fcomp.2025.1670473.
10. Cheikhrouhou O, et al. 2025. Blockchain and emerging technologies for next-generation secure healthcare: A systematic review. Blockchain: Research and Applications 6(4):100314. DOI: 10.1016/j.bcra.2025.100314.
11. Ziegler S, Nowostawski M, Katt B. 2025. Information privacy in blockchain systems: A systematic literature review. Journal of Cybersecurity and Privacy 5(3):65. DOI: 10.3390/jcp5030065.
12. Wu H, Yao Q, Liu Z, Huang B, Zhuang Y, Tang H, Liu E. 2024. Blockchain for finance: A survey. IET Blockchain 4(2):101–123. DOI: 10.1049/blc2.12067.
13. Raghav A, Tripathi AM. 2024. A Brief Overview of Various Blockchain Interoperability Models in Healthcare. In: 2024 International Conference on Computing, Power, and Communication Technologies (IC2PCT). DOI: 10.1109/IC2PCT60090.2024.10486597.
14. Abdullah N, Hakansson A, Moradian E. 2017. Blockchain-based approach to enhance big data authentication in a distributed environment. In: 2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN). 887–892.
15. Adel K, Elhakeem A, Marzouk M. 2022. Decentralizing construction AI applications using blockchain technology. Expert Systems with Applications 194(1):116548. DOI: 10.1016/j.eswa.2022.116548.
16. Agrawal R, Verma P, Sonanis R, Goel U, De A, Kondaveeti SA, Shekhar S. 2018. Continuous security in IoT using blockchain. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Piscataway: IEEE, 6423–6427.
17. Ahmed WAH. 2022. Blockchain technology applications in the supply chain: a critical analysis. Ph.D. thesis, University of Nottingham, UK.
18. Al-Jaroodi J, Mohamed N. 2019. Blockchain in industries: a survey. IEEE Access 7:36500–36515. DOI: 10.1109/ACCESS.2019.2903554.
19. Alam T. 2019. Blockchain and its role in the internet of things (IoT). International Journal of Scientific Research in Computer Science, Engineering and Information Technology 5(1):151–157. DOI: 10.32628/CSEIT195137.
20. Alammary A, Alhazmi S, Almasri M, Gillani S. 2019. Blockchain-based applications in education: a systematic review. Applied Sciences 9(12):2400. DOI: 10.3390/app9122400.
21. Alangot B, Reijsbergen D, Venugopalan S, Szalachowski P. 2020. Decentralized lightweight detection of eclipse attacks on Bitcoin clients. In: 2020 IEEE International Conference on Blockchain

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue IV, April 2026

(Blockchain). 337–342.
22. Amsden Z, Arora R, Bano S, Baudet M, Blackshear S, Bothra A, Cabrera G, Catalini C, Chalkias K, Cheng E, Ching A, Chursin A, Danezis G, Di Giacomo G, Dill DL, Ding H, Doudchenko N, Gao V, Gao Z, Garillot F, Gorven M, Hayes P, Hou JM, Hu Y, Hurley K, Lewi K, Li C, Li Z, Malkhi D, Margulis S, Maurer B, Mohassel P, de Naurois L, Nikolaenko V, Nowacki T, Orlov O, Perelman D, Pott A, Proctor B, Qadeer S, Rain, Russi D, Schwab B, Sezer S, Sonnino A, Venter H, Wei L, Wernerfelt N, Williams B, Wu Q, Yan X, Zakian T, Zhou R. 2020. The Libra Blockchain. Available at: https://diem-developers-components.netlify.app/papers/the-diem- (accessed 11 November 2023).
23. Andola N, Raghav, Yadav VK, Venkatesan S, Verma S. 2021. Anonymity on blockchain-based e-cash protocols—A survey. Computer Science Review 40(2):100394. DOI: 10.1016/j.cosrev.2021.100394.
24. Balon B, Kalinowski K, Paprocka I. 2022. Application of blockchain technology in production scheduling and management of human resources competencies. Sensors 22(8):2844. DOI: 10.3390/s22082844.
25. Batubara FR, Ubacht J, Janssen M. 2018. Challenges of blockchain technology adoption for e-government: a systematic literature review. In: Proceedings of the 19th Annual International Conference on Digital Government Research: Governance in the Data Age, dg.o '18. New York, NY, USA: Association for Computing Machinery.
26. Belchior R, Vasconcelos A, Guerreiro S, Correia M. 2021. A survey on blockchain interoperability: past, present, and future trends. ACM Computing Surveys (CSUR) 54(8):168. DOI: 10.1145/3471140.
27. Benchoufi M, Porcher R, Ravaud P. 2017. Blockchain protocols in clinical trials: transparency and traceability of consent. F1000Research 6:66. DOI: 10.12688/F1000RESEARCH.10531.1.
28. Bergquist J, Laszka A, Sturm M, Dubey A. 2017. On the design of communication and transaction anonymity in blockchain-based transactive microgrids.