pp. 141:168

# A Survey on Blockchain: Challenges, Attacks, Security, and Privacy

Hourieh Alsadat Hosseini , Alireza Hedayati*

Department of Computer Engineering, Central Tehran Branch, Islamic Azad University, Tehran, Iran, Hedayati@iauctb.ac.ir

## Abstract

Currently, industry and academia have shown much interest in security and privacy protection on the blockchain used in various applications. Attacks like privacy leakage and data loss make conventional methods vulnerable before emerging blockchain technology. Blockchain is a decentralized and tamper-resistant public ledger technology that guarantees security and data reliability in a peer-to-peer network. Many fields have employed blockchain, from the beginning cryptocurrency to the smart contract, social services, industry, and artificial intelligence. There are blockchain reports on vulnerabilities and security, but they lack a comprehensive survey in attacks, privacy, and security views. In this survey, we first briefly overviewed blockchain. Second, we discussed challenges and issues on the blockchain. Third, we focused on the blockchain attacks, including their cause and targeted area. We also displayed possible preventive measures in the blockchain attack. Finally, we conducted a systematic study on solutions to the blockchain security increase. In addition, this survey included blockchain privacy techniques.

*Keywords:* blockchain, privacy, network security, cryptocurrency, consensus algorithms
Article history: Received 02-Aug-2021; Revised 15-Aug-2021; Accepted 18-Aug-2021. Article Type: Research Paper

http://dx.doi.org/10.30495/ijsee.2021.684027

## 1. Introduction

The initial concept of the blockchain emerged in 1991 when a data chain was employed as a ledger. It guaranteed that any malicious user couldn't tamper with signed documents in the chain [1]. Since 2009, blockchain has caught much attention in industry and academia. It has been used in many domains, containing economics, Internet of things, smart cities, medicine, software engineering, or intelligent transport systems. Blockchain supports digital asset transfer in decentralized techniques utilizing the ledger with no need for trusted third-party authority and intermediaries [2].

Using blockchain technology in the financial technology industry made users concern about blockchain security due to reporting some security vulnerabilities (e.g., financial losses in smart contracts), attacks, and privacy issues (e.g., the leakage in the original identity of users and the amount of transaction) [3]. This paper systematically reviews blockchain challenges, blockchain attacks with their possible defensive measures, and security/privacy requirements and techniques on the blockchain. Some surveys on security and privacy in blockchain have already existed. Some review articles [4], [5] survey security issues and blockchain challenges. [6] surveys some blockchain applications, challenges, and problems. It also abstracts their problem-solving approaches. Some review the privacy challenges of decentralized cryptocurrencies like [7]-[9]. [3] reviews only security in addition to risks and attacks and not directly concentrated on privacy. [10] surveys distributed ledgers' security and privacy and [11] mainly concentrated on cryptocurrencies like bitcoin. In addition, [12] and [13] survey security and privacy techniques on the blockchain as well [2]. [14] reviews attacks on cryptocurrency, but none is as complete as this survey paper. It contains a review of challenges, attacks, privacy-preserving, and security methods for different blockchain systems.

A comparative study of blockchain attacks, privacy-preserving, and security techniques is the objective of this survey. At first, we survey challenges and issues on blockchain. Second, we study blockchain attacks and analyze their vulnerabilities and overview the possible defensive measures. Third, we review the security requirements of transactions, study the privacy-preserving solutions for blockchain and discuss techniques leveraged to improve the security and privacy on the blockchain. Finally, we can say this survey is a helpful reference for users and researchers. The remainder of this study is surveyed as follows (Figure 1):

– Section 2 overviews the blockchain, consists of how it works, its classification, its architecture, and its evolution.
– Section 3 includes challenges and issues on blockchain.
– Section 4 summarizes the typical consensus algorithms employed in the blockchain.
– Section 5 surveys blockchain attacks, including their target, the negative impact, and the possible defensive measures.
– Section 6 reviews the security and privacy properties of blockchain, including requirements and techniques.
– Section 7 concludes the survey.

## 2. Overview of Blockchain

The first blockchain design was documented in 2008 and implemented in 2009 by Satoshi Nakamoto to record all Bitcoin transactions without misbehaving or cheating. Blockchain is a safe, private, and reliable public storage for all bitcoin transactions and stores these transactions in secured chained blocks. Bitcoin blockchain implements three significant abilities:

– The digital signature, verifying data through a cryptographic algorithm.
– Hash chained storage, including hash pointer and Merkle tree.
– The commitment consensus, which uses the network majority about whose valid block should be joined into the blockchain.

The Bitcoin blockchain can stop both the double-spending problem and the transaction data change in a block after successfully committing to the blockchain using security methods and consensus schemes [12]. For understanding the blockchain concept and its technology, we will overview blockchain in this part.

### 2 1. How the Blockchain Works

A blockchain is a secure database of transaction logs. Client A will generate a bitcoin transaction when he/she sends some bitcoins to another client B. Miners should confirm the transaction and spread it to every node in the network. They validate it during a mining process by solving a complex computational mathematical problem [15] and then scatter the block with its confirmation to the network utilizing a consensus protocol. New blocks can only be joined into the blockchain when the users achieve a consensus. So original miner is rewarded, and this transaction from A to B will be legitimate. Figure 2 shows this process [12]. Afterward, a malicious user should gain the whole blockchain network's control for modifying the transaction because every node has the transaction copy [14].

### 2 2. Classification of Blockchain

Because of the growing demand for blockchain technology, there are various ways to classify blockchain. Okada et al. There is a classification based on whether or not a trusted authority with specific control exists on the blockchain. Also, another categorizes blockchain based on permission rights, chain ownership, the decentralization degree, and the computing method to deliver a service, including peer-to-peer or cloud-based. One can classify three main kinds of blockchains based on access rights: public, consortium, and private [15].

Private and consortium blockchains are both kinds of permissioned chains, but a public blockchain is permissionless. In a permissionless environment, all the peers of the network are equally responsible. Nobody is responsible if failures have occurred, like code failure, and there is no way to recover these failures. In permissioned blockchain, parties permit trusted nodes, who are the charge of the verifying process. A risk may emerge if granted permissions are incorrect [15]. Table 1**Error! Reference source not found.** and Figure 3 compare the blockchain classification.

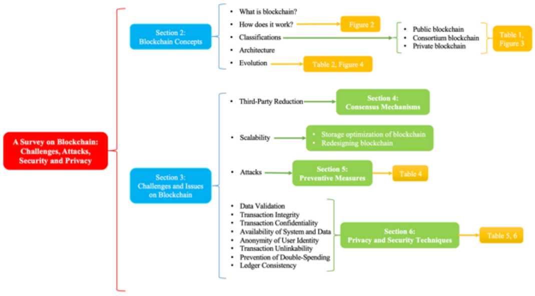
A Survey on Blockchain: Challenges, Attacks, Security and Privacy
Section 2: Blockchain Concepts
• What is blockchain?
• How does it work?
Figure 2
• Classifications
• Public blockchain
• Consortium blockchain
• Private blockchain
Table 1, Figure 3
• Architecture
• Evolution
Table 2, Figure 4
Section 3: Challenges and Issues on Blockchain
• Third-Party Reduction
Section 4: Consensus Mechanisms
• Scalability
• Storage optimization of blockchain
• Redesigning blockchain
• Attacks
Section 5: Preventive Measures
Table 4
• Data Validation
• Transaction Integrity
• Transaction Confidentiality
• Availability of System and Data
• Anonymity of User Identity
• Transaction Unlinkability
• Prevention of Double-Spending
• Ledger Consistency
Section 6: Privacy and Security Techniques
Table 5, 6

Fig. 1. Survey taxonomy

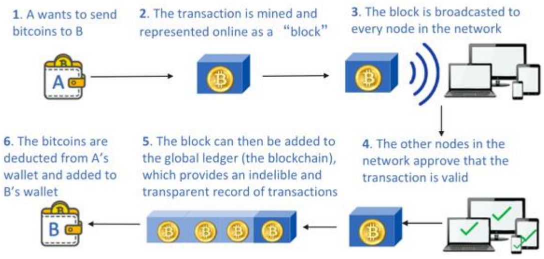
1. A wants to send bitcoins to B
2. The transaction is mined and represented online as a "block"
3. The block is broadcasted to every node in the network
4. The other nodes in the network approve that the transaction is valid
5. The block can then be added to the global ledger (the blockchain), which provides an indelible and transparent record of transactions
6. The bitcoins are deducted from A's wallet and added to B's wallet
A
B

Fig. 2. How the blockchain works [12]

Table.1.
Comparisons of the blockchain classification [14], [6]

| Parameters | Public Blockchain | Consortium Blockchain | Private Blockchain |
|---|---|---|---|
| Throughput | Low | High | High |
| Participation in Consensus Process | Authentication not required /Permission less | Authentication required /Permissioned | Authentication required /Permissioned |
| Immutability | Nearly impossible to tamper | Can be tampered | Can be tampered |
| Security | Proof of Stake | Proof of Work | Pre-approved participants |
| Identity | Anonymous | Pseudonymous | Known Identities |
| Speed | Slower | Slower | Faster |
| Read Access | public | Decided by organization | Decided by organization |
| Central Authority | Decentralized | Partially centralized | Fully centralized |
| Efficiency | Low | High | High |
| Block Authentication | All miners | Selected nodes | Specific organization |

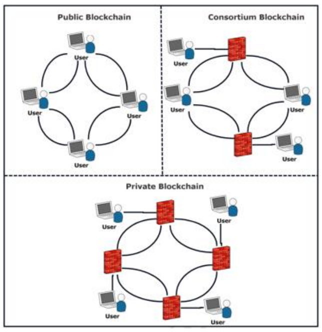

Fig. 3. Overview of blockchain classification [14]

### 2 2.1. Public Blockchain

A public blockchain is considered as "completely distributed." Participants in the network can read, send, or receive the transactions and decide which are legitimate and can get added to the blockchain through the consensus process. In addition, there is no need for trusted third parties and intermediaries [14].

### 2 2.2. Consortium Blockchain

A consortium blockchain is considered as "moderately distributed." Although any participant in the network has read permissions, there are some constraints on write permissions like influencing and controlling the process validation through several pre-selected nodes, a set of participants in the network [16].

### 2 2.3. Private Blockchain

In a private blockchain, one organization keeps centralized the write permissions in a fully private blockchain. Read permissions of the blockchain can be available to the public or restrict to an arbitrary extent through the owner organization permission if necessary. Management of database and review are some applications of private blockchain for a company [14].

### 2 3. Architecture and Evolution of Blockchain

Nowadays, there are five blockchain technology phases or generations (Figure 4), including the blockchain 1.0 phase as digital currency displayed by Bitcoin, the blockchain 2.0 phase as digital economy presented by Ethereum, Blockchain 3.0 phase as digital society represented by Hyperledger, Blockchain 4.0 phase as the industry displayed by Industrial IoT (IIoT), and Blockchain 5.0 phase as artificial intelligence [17], [19]. Blockchain 1.0 begins with a genesis block and joins chronologically winning blocks. Blockchain 2.0 utilizes smart contracts that contain economic activity [20]. Blockchain 3.0 is an application collection that does not include economic activity but includes health, identity, governance, science, education, art, and different culture and communication perspectives [21]. Blockchain 4.0 is proposed for industrial challenges and improving secure real-world applications in a decentralized method. Blockchain 5.0 focuses on AI and DLT integration for developing data privacy and security in the next generation of decentralized Web 3.0. [19]. Blockchain 5.0, also termed Relictum Pro Blockchain, covered the previous generations' technology improvements. The main specific features of Blockchain 5.0 are HyperNet that includes virtual communication channels, the new architecture of blocks and chains, Proof of Tsar consensus mechanism. Other specifications of Relictum Pro Blockchain are a decrease in block size and node filling speed, increase in the transactions' number per second. Table 2 compares features of different blockchain generations [22].

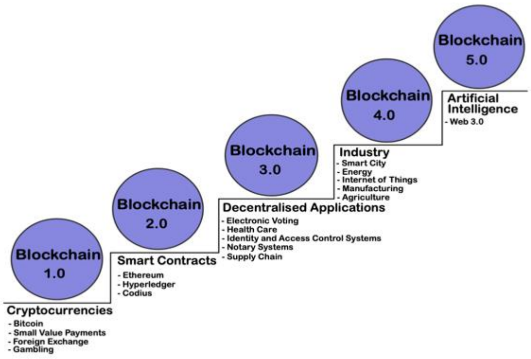

Fig. 4. Blockchain Evolution [19]

Table.2.
Blockchain generations comparison [22]

| Blockchain Generation | Block Size (byte) | Number of transactions per second | Node filling speed |
|---|---|---|---|
| Gen. 1 (Bitcoin) | 1,024k | up to 10 transactions | 10 minutes |
| Gen. 2 (Ethereum) | 512k | up to 20 transactions | 5 minutes |
| Gen. 3 (EOS) | 128k | from 1,000 to 900,000 | 5 seconds |
| Gen. 4 (Seele.pro) | 268 | from 1,000 to 900,000 | 3 seconds |
| Gen. 5 (Relictum Pro) | 120 | > 1,000,000 transactions | from 0.5 to 1 second |

There are six levels of the blockchain platform, including data layer, network layer, incentive layer, consensus layer, contract layer, and application layer [23], [19].

### 2.3.1. Data Layer

Pointers, variables point to another variable location, and a linked list, a chained blocks list, are two main parts of the blockchain data structure. The data layer guarantees data storage integrity and includes block storage techniques and chain structure techniques.

Each block contains an owner signature, a timestamp that determines the precise moment of mining the block and validating by the blockchain network, a hash pointer for linkage to the parent block (previous block), and a nonce that is a counter incremented for each computation of hash value [14]. It also includes a Merkle Hash Tree, the block number, and a hash algorithm, as you can see in Figure 5 [23], [24].

Hash pointers join nodes together in a Merkle tree data structure. The Merkle tree algorithm generates a new data node for each couple of nodes in the lower level repeatedly till arriving at the tree root. It can prevent data tampering using hash pointers because a malicious user can't tamper with a leaf node without modifying its upper parent node hash value. If the node hash pointer changes and does not equal the saved one on the root, data tampering is disclosed [12]. So it can verify the data node membership and give transaction integrity [2].

Security, integrity, and irrefutability in the blockchain are the result of using a Merkle tree. For example, storing information in the Ethereum blockchain uses a database of the Patricia tree (Trie), which has a root hash similar to the Merkle tree used to point to the whole tree. So, changing the tree content without modifying the root hash is impossible. Data storage in blocks relies on the blockchain type. Bitcoin blockchain has the sender, recipient, and amount of data, but Hyperledger Fabric's blocks have information of the channel [24].

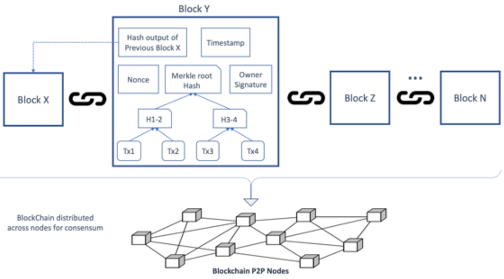

Fig. 5. blockchain structure [2]

### 2.3.2. Incentive Layer

The incentive layer distributes incentives to nodes that insert valid blocks into the blockchain. The issuance and allocation of incentives are two incentive methods used in the incentive layer. This layer also incentives nodes to join in Blockchain verification. For example, miners are encouraged with bitcoins as a reward [19].

### 2.3.3. Network Layer

The network performs a decentralized consensus scheme to check the new block acceptance in the blockchain, the data content consistency on each node, and the read protocol for safe blockchain confirmation [12].

The network layer termed the peer-to-peer (P2P) layer or propagation layer [24], contains data transmission protocols between nodes and mechanisms of transaction verification. In the blockchain P2P network, all nodes are interconnected with no need for a central authority. There is internode communication between every two nodes in the P2P network [23]. This layer protects transactions and blocks spread [24].

Every node both provides and consumes information. It processes the network routing, finds and keeps the adjacent peers' communication, distributes and validates the transactions, and synchronizes the data blocks [13]. The bitcoin blockchain can prevent double-spending transactions in a completely decentralized P2P network without dependence on any trusted central authority [12].

#### 2.3.4. Consensus Layer

The consensus layer contains a consensus mechanism for reaching an agreement of block verification in the decentralized network [12]. We explain the consensus mechanism in the next section. For example, bitcoin uses PoW, Ethereum utilizes more PoW and PoS, Hyperledger applies more PBFT and SBFT, and blockchain 4.0 uses consensus protocols based on gossip [12], [20].

#### 2.3.5. Contract Layer

The contract layer includes the smart contract that is the programmable trait of the blockchain. Nodes in blockchain perform smart contracts in a distributed manner [23]. It is in charge of processing requests and validation of the transactions.

#### 2.3.6. Application Layer

The application layer contains Bitcoin for digital currency transactions, Ethereum for digital economy transactions, Hyperledger for digital society, IIoT for industry, and web 3.0 applications for AI [23]. A Hyperledger, also named chaincode, can include multiple smart contracts and manages their packaging and deployment when they include transaction management [21].

The application layer can be subdivided into the application layer applied by end-users in the network and the execution layer comprised of smart contracts and chain code. The application layer contains application program interfaces (APIs), scripts, frameworks, and user interfaces. A transaction scatters instructions from the application sub-layer to the execution sub-layer [24].

## 3. Issues and Challenges on Blockchain

In this section, we discuss challenges and issues on the blockchain. We start by explaining some challenges and then study the countermeasures employed in these challenges in the following parts.

### *3.1. Third-Party Reduction*

Centralized Information systems need a trusted network among involved parties. A decentralized system causes an improvement in interoperability, a need reduction in third-party, and the prevention of tampering with transactions [25]. Instead of trusted third-party authority, decentralized blockchain systems use consensus mechanisms to assure data/transaction consistency and reliability [3]. There is a wide variety of trusted third-party mechanisms. But we survey some consensus mechanisms employed in blockchain in section 4.

### 3.2. Blockchain Scalability

The blockchain becomes heavy every day because of the increase in the number of transactions stored for verifying. Additionally, there is a constraint in block size and the period employed for generating a new block. So millions of transactions, by processing nearly seven transactions per second for bitcoin, cannot be processed. By the way, the small blocks' capacity causes a delay in many small transactions, which miners choose with a high transaction fee. On the other hand, a large block size would reduce the propagation speed and cause blockchain forks. So, the scalability issue of the blockchain is very hard. There are two attempts for the problem of blockchain scalability [6]:

#### 3.2.1. Blockchain Storage Optimization

In 2014, a new cryptocurrency design was introduced for solving the problem of blockchain scalability. In this scheme, the network eliminates old records of the transaction, and an account tree, as a database, keeps all nonempty addresses balance. So, there is no need to save all transactions for validation of their legitimacy. In addition, VerSum is another solution using lightweight nodes which outsource costly computations over vast inputs. It guarantees the accuracy of the computation result by comparing from various servers.

#### 3.2.2. Redesigning Blockchain

There is a trade-off between network security and block size in the blockchain. For addressing this trade-off, Eyal et al. introduced Bitcoin-NG (Next Generation) in 2016. Bitcoin-NG separates original blocks into two sections: 1) Key Block to elect the leader; 2) Microblock for saving transactions. There is competition between miners to become a leader who is responsible for generating microblock. Bitcoin-NG also lengthened the chain and redesigned the blockchain by counting only key blocks and carrying no-weight microblocks.

### 3.3. Blockchain Attacks

One of the main issues and challenges of the blockchain is blockchain attacks. According to five common attack vectors on the blockchain, we classify and analyze the attacks' vulnerabilities and their possible preventive measures in section 4. There are five common attack vectors on blockchain:

- Transaction verification mechanism attacks;
- Mining pool Attacks;
- Network Attacks;
- Private key/ User wallet attacks;
- Smart Contract-based attacks [26].

### 3.4. Security and Privacy Requirements

Two requirements are needed for privacy preservation in the blockchain: (i) the transaction links should be invisible and undiscoverable; (ii) The transaction content is only identified to their participants. An access control policy could be set to satisfy the privacy requirements of the private blockchain. It indicates that complete data transparency is not a problem. The privacy requirements of blockchain should be regarded in two parts: Identity Privacy and Transaction Privacy.

- Like the relationships of the user transactions, Identity Privacy is intractability between scripts of the transaction and the partaker's real identities. Users can provide restricted identity privacy by applying random addresses (or pseudonyms) in the blockchain. There are some strategies of behavioral analysis, like the Know Your Customer (KYC) policy and Anti-Money Laundering (AML) regulation, that may expose some information by the use of monitoring the network without encryption and traversing within the public blockchain. This information is about who is utilizing blockchain or why.
- In Transaction Privacy, certain users only have access to the transaction contents, including the amount of the transaction patterns. So, the transaction contents are kept private to the public network on blockchain. Many applications, including blockchain-based electrical health record management and anonymous big data authentication, desire privacy in transactions [13].

In this part, we discuss the security requirements of transactions that each of them is targeted at one kind of vulnerabilities of the blockchain. In online transactions, the security and privacy requirements are classified into the following types [12]:

#### 3.4.1. Data Validation and Transaction Integrity

For integrity and confidentiality, personal data security includes security on unauthorized or illegal processing and unexpected loss, destruction, or damage [2]. In a decentralized blockchain network, the technology is a good candidate for data validation and transaction integrity. Current methods employed in the industry are not decentralized that they depend on a trusted third party instead. Depending on a trusted third party can cause vulnerability. We are aiming to achieve the integrity of the third party with no concern. Using the concept of "smart contracts" is an approach to assuring transaction integrity. The goal of smart contracts is to apply blockchain to verify that a contract has been signed by two parties [25].

Applying online transactions raises the cost of transactions and causes the risk of intentional forging or falsifying the certificates. So, the system needs to ensure transaction integrity and prevent tampering with transactions [12].

#### 3.4.2. Transaction Confidentiality

In confidentiality, privacy is the personal data protection against unauthorized access and anonymization [2].

Revealing of transactions and information of accounts in online financial transactions should be minimal. This minimal includes the following three disclosures:

- An unauthorized user cannot access the transaction information of any users.
- The participant or administrator of the network cannot reveal information of any users to others without their permission.
- All user data access should be securely and consistently, even when malicious cyber-attacks or sudden failures happen. Such a kind of confidentiality is acceptable in various non-financial scenarios [12].

#### 3.4.3. Availability of System and Data

In online systems, the users should be able to access the transaction data anywhere and anytime. Both system and transaction should be available. System availability means the system should be reliably run even when a network attack has occurred. Transaction availability means authorized users can access transaction data without inconsistency, unattainability, or corruption [12].

#### 3.4.4. Anonymity of User Identity

Every entity in blockchain communicates with others within a produced address. Anonymity means that these addresses do not expose the real identity of the involved users. Blockchain does not guarantee perfect privacy-preserving due to some inherent limitations [14].

Repeating user authentication may cost high because secure and efficient user data sharing has difficulty among different financial institutions. It also causes the revealing risk of user identity via trusted third parties [12].

#### 3.4.5. Transaction Unlinkability

Unlinkability means that participants will not know the joint transaction details, including transaction destinations with which the addresses of senders are joined [13].

Users need both identity anonymity, not exposing original identity, and unlinkable transactions. Because understanding other information about the user, including the account balance, transaction type, and transaction frequency, is easy when the user related-transactions are linkable. Using such transaction data, account information, and some background knowledge about a user, the original user identity may be inferred by adversaries with high confidence [12].

Users always produce pseudonyms in connection to the bitcoin system. So bitcoin provides a restricted form of unlinkability [13].

### 3.4.6. Prevention of Double-Spending

A centralized trusted third party of a centralized network is in charge of verifying if a digital currency has been double-spent. For the decentralized network, robust security methods and countermeasures are needed to prevent double-spending. Prevention of double-spending on a fully decentralized peer-to-peer network is a considerable novation of the blockchain that doesn't depend on any trusted central authority [12].

### 3.4.7. Ledger Consistency

The liquidation and clearing process between various financial institutions causes inconsistencies, errors of Ledgers, and high transaction expenses because of manual processes, the architecture, and different business processes [12]. According to these security and privacy requirements, we survey some techniques for improving the security and privacy of blockchain in section 6. We also summarize the advantages and disadvantages of each security method on the blockchain.

## 4. Consensus Mechanisms

Dynamically reaching an agreement in a group is consensus [12]. Miners validate the transaction and create a block. After solving the puzzle and building a new valid block, it is distributed to the blockchain network using a consensus algorithm that guarantees its validation for adding to the blockchain. Miners get the reward of block creation using consensus algorithms like PoW or PoS [24]. We review some consensus mechanisms used in blockchain as follows:

### 4.1. Proof of Work (PoW)

Satoshi Nakamoto created the PoW consensus algorithm for bitcoin as a pioneer [11]. PoW validates each bitcoin transaction by reaching an agreement in the network [12].

A miner, participated nodes in the mining process, solves a puzzle, a complicated business problem, before adding a block to the blockchain. So, it is rewarded by cryptocurrency and competes with others for gaining a correct hash. Then it propagates the block to the P2P network's nodes (Figure 6). Before joining the block to the blockchain, other nodes should validate it. If multiple miners solve the puzzle simultaneously, the longest chain wins. This mechanism solves the double-spending problem. But it is slow, needs a lot of computing power and energy, and not scalable [24].

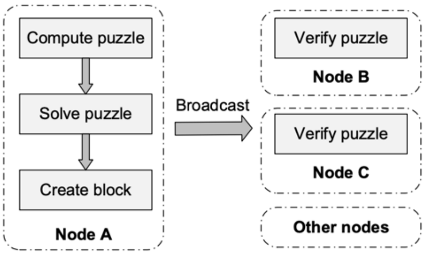

Fig. 6. PoW consensus mechanism [3]

### 4.2. *Proof of* State *(PoS)*

The PoS mechanism, an alternative to PoW suggested in 2011 by Peercoin [24], utilizes cryptocurrency possession proof for data verification. Users should spend a specific cryptocurrency amount in the flow of block creation or transactions in the PoS mechanism. The original node can be rewarded or fined. When the created block or transaction is verified, the spent cryptocurrency is returned as a reward. In the PoS mechanism, the possibility of mining relies on the stake, coins of miners.

Compare to the PoW mechanism, the PoS mechanism can considerably decrease the computation amount, and the blockchain throughput is improved [3]. The blockchain network attack is costly based on PoS consensus, and this mechanism is also energy efficient [24].

### *4.3. Proof-of-Activity (PoA)*

The PoW needs an immense computing power amount and more electricity consumption. It also requires expensive new-age hardware devices for mining and transaction verification for adding a block to the blockchain network. In PoS, mining relies on the cryptocurrency amount that exists in a node. in PoS, and it applies low-cost hardware. Proof-of-Activity (PoA) makes the best of both and prevents a 51% attack possibility like both Pow and PoS.

How PoA Works:

At first, the PoA uses the PoW mechanism until mining the new block. Then it uses the PoS mechanism. A group of random validators verifies or signs the new block by considering the header details. A validator who owns more crypto coins has more signing chances. So, the newly created block joins the blockchain. When the chosen signers are not present, a group of random validators will select the next winning block. The reward is given to the primary miner and the several validators who have verified the new block [27].

### *4.4. Practical Byzantine Fault Tolerance (PBFT)*

Byzantine Fault Tolerance (BFT) algorithm is not from the Proof algorithms group. Its title is obtained from the famous Byzantine general's problem (BGP). An army surrounded a fort city. If all the Byzantine generals attack at the same time, they will succeed in the war. Reaching a consensus to attack requires interacting with each other [24]. Only when the honest general majority agree on a strategy, Byzantine Fault Tolerance can be performed [12].

The first BFT solutions were proposed by Shostak, Pease, and Lamport in 1982. The Practical Byzantine Fault Tolerance algorithm, introduced by Miguel Castro and Barbara Liskov, can help resolve the BFT. A novel Byzantine agreement protocol, AlgoRAND, is considerably more efficient than all formers with its new property, named player replaceability. It secures the adversarial environment. Hyperledger Fabric utilizes the Practical Byzantine Fault Tolerance (PBFT) consensus mechanism [24].

How PBFT Works:A user requests a leader (primary) node. Then the message is scattered to all the follower (secondary) nodes through the leader node. All leader and follower nodes will do the task demanded by the user. Then they reply to the user. If the user gets the same n + 1 responses, it will ensure having a successful request, including attack or scuttle. N is the maximum of the malicious nodes number [24].

Compared to the old PBFT protocol, the first asynchronous practical BFT protocol, HoneyBadgerBFT, works based on a novel spread protocol with better throughput [12].

### *4.5. Simplified Byzantine Fault Tolerance (SBFT)*

Simplified Byzantine Fault Tolerance (SBFT), a state-of-the-art BFT algorithm, is a decentralized trust infrastructure mechanism that is scalable and has a better performance for wide deployments. The performance of SBFT rises with the increase in the client number.

Compare to the PBFT mechanism, the SBFT mechanism adds four main design parts, including applying linear PBFT, utilizing cryptography, adding a quick path, and using additional servers to enhance performance and flexibility. Compare to a very optimized system using the PBFT mechanism, SBFT improves throughput and latency around 2x and 1.5x, respectively [28].

### *4.6. Delegated Proof of Stake (DPoS)*

Larimer D proposed a Delegated Proof of Stake (DPoS) consensus mechanism in 2014 [24]. A randomized delegated proof of stake algorithm, Roll-DPoS, is suggested by Fan for blockchain applications in IoT [29]. The DPoS, in comparison to the PoS consensus mechanism, chooses different block accounting nodes. In this mechanism, a candidacy node, every node with tokens, can vote and elect some agent nodes that can create and confirm the blocks. The mining process of DPoS decreases nodes' number in creating and verifying the block. So it can reach the second-level verification.

### *4.7. Proof of Elapsed Time (PoET)*

The Proof of Elapsed Time (PoET) consensus algorithm relies on wait time when validators create randomly picked time and sleep. Who wakens at first can create a new block and scatter that data to other peer-to-peer network nodes. Each node has the same chance for joining the block to the blockchain by this random delay. Fairness, low computing consumption [12], low cost of validators participation, and improving consensus algorithm robustness are advantageous of this mechanism [24].

### *4.8. Sleepy Consensus*

There are two sleepy statuses of "awake/active" (online) or "asleep" (offline) in this consensus mechanism. Members can modify their modes meanwhile the execution of the protocol. The majority of the honest members can prove the Sleepy consensus mechanism. So, it cannot work when the majority of online members are dishonest [12].

### *4.9. Proof of Authority*

Relatively quick transactions use the Proof of Authority (PoA) consensus algorithm, which validates transactions and novel blocks only by validators. The validator, authoritative nodes, is a participating node with a high score credit. Two reasons considered PoA is more robust than PoS: 1) Transactions and blocks should be confirmed honestly by validators. 2) One validator cannot validate two blocks sequentially. So it stops centralized trust [12].

### *4.10. Proof of Reputation (PoR)*

The extension of the Proof of Authority causes the Proof of Reputation (PoR) consensus mechanism introduced by several investigation groups and organizations. It has various types and parameters for performance adjustment. An authoritative node can be voted into the network when a node obtains a reputation. At that time, it acts as a Proof of Authority that only validators can confirm blocks [12].

### *4.11. Directed Acyclic Graph (DAG)*

Transactions are stored in blocks in the blockchain, but they are saved in nodes in a Directed Acyclic Graph (DAG) structure [30]. The concept of acyclic flow on a DAG-based network refers to the information flows in just one direction without returning to the sender. There is no connection between nodes and their prior ones that reduces the block times [31].

### *4.12. Proof of Tsar (PoT)*

PoT defends hash collisions and majority attacks. The network reconnects all nodes every 0.5 seconds as Tsar and General nodes. General nodes get transactions and send them to Tsar for the process. Then, Tsar transfers blocks further down the chain. These two types of nodes are chosen automatically and adjust frequently [22].

## **5. Blockchain Attacks**

In this part, we study attacks on blockchain systems and analyze the vulnerabilities in these attacks according to five common blockchain attack vectors in section 3.3. Table 4 provides a general overview of the potential blockchain attacks with their possible countermeasures.

### *5.1. Transaction Verification Mechanism attacks*

Transactions on the blockchain should be validated via all node agreements in the network. The confirmation of a block takes time. This delay can be utilized to deceive the system by attackers [26].

#### 5.1.1. Double-Spending Attack

If a miner mine blocks faster than others on the network, a successful double-spending attack is probable. The speed of block mining relies on solving the associated PoW and miner's computing power. Besides, there are other factors for a successful double-spending attack, including propagation delay of the network, connectivity of bitcoin exchange services, client, vendor, and honest miners number. While the transaction confirmation number rises, the probability of an invalid transaction at a later stage reduces. So the double-spend possibility reduces. Oppositely, when the miner's computing resources increase, the double-spend success possibility rises [10].

If a user spends the identical bitcoin set for two different transactions simultaneously, double-spending will be performed in the bitcoin network. Five stages express how double-spending perform [32]:

- Block adding process. At first, users request transactions in a pool where the transactions are picked. The miner solves the mathematical problem with complications using POW consensus to get a unique hash output. Then he/she spread them to add the block to the blockchain only when other miners confirm these hashes.
- The corrupted miner creates his/her chain with the block verified by the honest miners. At that time, the block is joined the original blockchain. The corrupted miner expends all his/her coins and transfers this information to the original blockchain, not to his/her private chain.
- After picking the transactions, the corrupted miner verifies and adds the block to his/her private chain quicker than the honest miners add the block to the original blockchain.
- when the private chain is longer than the original chain, the corrupted miner spread the transaction of the private blockchain to the original blockchain.
- According to the democratic governance rule, the blocks will join the larger chains by deleting the former records. The block on the original blockchain possessed the transaction information because the corrupted miner expends all his/her coins. But the private chain does not have the transaction information. So, when the blocks try to join the private chain, they will delete the former transaction information. Thus, in the novel private chain, the corrupted miner can spend all his/her coins had expended once in the original blockchain.

#### 5.1.2. Race Attack

When a vendor admits a payment before confirming the transaction in blockchains based on PoW, a race attack occurs. While an attacker spreads a conflicting transaction to the network, the payment is sent to a recipient user. The second transaction will probably be accepted as original in the network. Losing a product by a vendor, generating blockchain forks, and banning legitimate users are race attack effects [33].

### 5.1.3. Finney Attack

A Finney attack is a form of double-spending attack. In a Finney attack, a block(Bp) consisting of transaction TUm_Um is mined by malicious users (Um) privately. Using the same bitcoins set for the merchant (M), The malicious user creates a transaction TUm_M. If miners validate that TUm_M is legitimate and added it to the blockchain, the merchant (M) confirms TUm_M. When the merchant (M) confirms transaction TUm_M, Bp is published to the bitcoin network. Um obtains merchant product if the Bp is spread in the network by the malicious user. So a blockchain fork (FO) with an equal length as the general fork (F) is generated. Then the fork FO, instead of F, is grown by mined block, and all network miners should mine on FO, according to the bitcoin protocol. Whenever FO becomes the longest blockchain, all miners neglect F. So the first block in F, including the transaction TUm_M, becomes illegitimate, and the merchant product will be missed. In the end, the malicious user will earn its coins by executing the TUm_Um and double spends if the vendor validates the transaction only once. Figure 7 displays the details of the Finney attack [14].

### 5.1.4. Brute-force or Alternative History Attack

A Brute-force attack, an improvement on the Finney attack [33], is employed to gather secret information [32]. A clever adversary governs on some nodes (N) in the bitcoin network via a brute-force attack. These N nodes collectively try to mine blocks privately with the purpose of double-spending. An attacker incorporates a double-spending transaction in some blocks working on the private chain extension (FO) at the same time.

If a merchant anticipates X validations before confirming a transaction, the product will be transferred after receiving X validations. Then, the X blocks may be privately mined and broadcasted in the bitcoin network. Because this will cause a longer FO than F, all miners in the bitcoin network will expand the fork FO. So this results in successful double-spending [14].

### 5.1.5. Vector 76 attack or One-confirmation Attack

Vector 76 attack privately uses the mined block for accomplishing double-spending attacks in Bitcoin Exchange Networks. A Bitcoin Exchange (BE) is a digital market in which bitcoins can be bought, exchanged, or sold. A malicious user (Um) contains a pre-mined block, including a deposit transaction. The Um anticipates the next block spread and transfers both pre-mined block and newly mined block to the BE or its adjacent peers. Some of the miners, mining on the blockchain that includes a pre-mined block (FO), are expected as a prime chain. Um immediately sends a withdrawal transaction from the trade of the identical bitcoins collection submitted by the Um in its previous transaction. The other fork (F) does not contain the transaction used to credit bitcoins by the adversary. The credit will be canceled if the fork F lasts. Um has already performed the withdrawal until this time. So, the exchange causes to miss of the bitcoins [14].

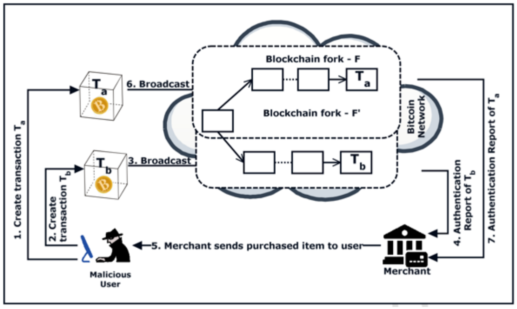
Blockchain fork - F
6. Broadcast
$T_a$
Blockchain fork - F'
Bitcoin Network
$T_b$
3. Broadcast
1. Create transaction $T_a$
2. Create transaction $T_b$
5. Merchant sends purchased item to user
4. Authentication Report of $T_b$
7. Authentication Report of $T_a$
Malicious User
Merchant

Fig. 7. Finney attack on a Bitcoin network [14]

### 5.1.6. Balance Attack

The Balance attack against PoW-based blockchain was proposed by Christopher et al.. Communications between similar-mining-power subgroups are momently disrupted by a low-mining-power attacker in the Balance attack. In the Balance attack, blockchain is abstracted into a DAG tree (DAG = < B, P >). The nodes indicating information of blocks are B. Directed edges P connected these nodes B. A delay between similar-mining-power subgroups is introduced. The transactions are issued in the “transaction subgroup,” and blocks are mined in the “block subgroup” to ensure that the block subgroup tree overweighs the transaction subgroup tree. The attacker can overweigh the tree, including this committed transaction, and rewrite blocks with strong possibility even if the transactions are committed.

The Balance attack permits double-spending. The attacker requires to produce transactions to buy goods from the merchants after recognizing the merchant-involved subgroup. Then he/she publishes transactions to this subgroup, scatters the mined blocks to the rest group nodes, and pauses delaying messages as long as the merchant ships goods. The tree of DAG, seen by the merchant, is overweighed by another tree with a strong possibility. Another transaction is successfully reissued by the attacker utilizing the same coins. Balance attack proves that PoW-based blockchain is blocked oblivious. The attacker can cancel or remove the block containing this transaction while writing a transaction into the main chain [3].

### 5.1.7. Nothing at Stake Attack

Validators motivate to operate on various forks, notwithstanding the protocol diversity of PoS. Conflicting blocks on the probable forks could be made by validators with nothing at stake. This problem is called the nothing at stake attack, which reduces the consensus time of the network, lessens the system efficiency, and decreases the blockchain capability to solve double-spending attacks and other threats [34].

## *5.2. Mining Pool Attacks*

Mining pools are made for the computing power raise that directly influences the block verification time. So the winning possibility of the mining reward increases and lots of mining pools have been produced by pool managers. The pool managers send units of unsolved work to members of the pool (i.e., miners). The miners create full proof-of-work (FPoWs) and partial proof-of-work (PPoWs), submitted to the manager as shares. While the miner finds a new block, he/she sends it with the FPoW to the manager. The manager spreads the block in the bitcoin network to obtain a mining reward and then scatters the reward to participant miners. The reward is spread relying on the shares parts in the pool, compared with the other miners [10].

### 5.2.1. Selfish Mining or Block Discarding Attack

Attackers conduct selfish mining attacks to gain undue rewards or waste honest miner computing power. The attacker keeps discovered blocks privately and tries to fork a private chain. Then, he/she would mine on the private chain and attempt to keep it longer than the public branch. In the meantime, legitimate miners mine on the public chain. When the public branch comes close to the length of the private branch, the attacker exposes newly mined blocks. So legitimate miners stop wasting computing power and do not obtain any reward because new blocks of selfish miners are published exactly before honest miners [3].

Selfish-Mining attack, also called block discarding attack [35], [36], was proposed as an attack strategy that makes the honest miners do wasted computations on the old public branch. In the initial Selfish-Mine, the public chain and private chain have the same length.

### 5.2.2. Long-Range Attack

In Long-Range attacks with PoS protocols, an attacker adds blocks, which are kept hidden by forking on the blockchain like selfish mining attacks with PoW protocols. Although both attacks use a chain fork, there is a difference that Long-Range attacks return to the origin block of PoS protocols. Posterior Corruption and Stake bleeding are different categories of Long-Range attacks. In simple attacks, blocks timestamp not be controlled. So every validator can confirm blocks in the PoS protocol. Posterior Corruption is an effort to create more blocks in parallel than the main chain to change the main chain history. In Stake bleeding, the adversary copies a transaction from the honest chain to a private chain [37].

### 5.2.3. Block withholding (BWH) Attack

Block withholding (BWH) is an attack very similar to the selfish mining performed on a mining pool in which a mined block never be sent to destroy the pool resources by a pool member. However, submit shares contains PPoWs, not FPoWs. In the BWH attack, a miner, who has gained a legitimate block, decides not to submit but discard it. This attack causes that all bitcoin rewards to be lost in the mining pool. There are two kinds of BWH scenarios called “Sabotage” and “Lie in wait” [10].

### 5.2.4. Fork After Withholding (FAW) Attack

The reward of BWH attackers is lower than FAW attackers, and FAW attack occurs up to four times more per pool than in a BWH attack. Besides, the additional reward of a FAW attack is about 56% more than the BWH attack when performing on various mining pools. Moreover, when two pools perform a FAW attack, the larger one always wins. FAW attack is more practical to perform using deliberate forks dissimilar selfish mining and FAW attack [10].

### 5.2.5. Bribery Attack

In the Bribery attack, an attacker can gain the computing resource majority for a short time through bribery. There are three ways to present bribery attack in the network:

(1) Out-of-Band Payment. In this way, the malicious user pays to the computing resource owner. Then the owners mine the adversary blocks.

(2) Negative-Fee Mining Pool. In this way, the attacker pays more return to form a pool.

(3) In-Band Payment via Forking. In this way, the attacker tries to bribe through bitcoin. He/she produces a fork comprising bribe money, which is easily reachable to any miner using the fork. The attacker can start several attacks like double-spending and DDoS if he/she has the hash power majority. So the briber miners will take short-term advantages that might be weakened by the losses in

the long-term via DDoS and >50% attacks or the rate crash of exchange [10].

5.2.6. Goldfinger, >50% Hash-rate, or Majority

The probability of the double-spending accomplishment leading to the Goldfinger attack is increased when the computation resources for the mining block increment. In this attack, one miner or mining pool affects more than half of the computation resources in the network. So it is also named the >50% hash rate attack. This attack can destroy the stability of the whole network by introducing any action, including transaction rejection or inclusion. This bitcoin network instability causes growing the adversary place while legal miners start leaving the network [10].

5.2.7. Feather and Punitive forking

A malicious user with much less than a 50% hash rate, through punitive forking, could do an optional blacklisting successfully. Punitive forking aims to control the bitcoin addresses of specific people, Alice, and stop them from bitcoins spending. Also, an adversary with a lower hash rate can produce lags and difficulties for Alice’s transaction. Feather forking is a malicious mining approach to obtain a blacklist. In feather forking, an attacker tries to fork when detecting a transaction block of Alice in the blockchain, but he/she will stop afterward [10].

*5.3. Network Attacks*

The blockchain nodes produce and run transactions and implement different services. For example, bitcoin network nodes transmit, receive, and approve transactions [26]. Bitcoin Network attacks exploit the existent vulnerabilities in the networking protocols of peer-to-peer relationships and the bitcoin protocols design and implementation [10].

5.3.1. Eclipse Attack or Netsplit Attack

A node selects eight peers accidentally in a network to expand and save another peer information. That node is attacked via the Eclipse attack to take advantage of the peer-to-peer network. In the eclipse attack, an attacker monopolizes all of the incoming and outgoing relationships of the victim. Then he/she separates the victim from the other peers in the network. Next, he/she can filter the blockchain view of the victim or permit the victim to cost redundant computing power on the old blockchain views. Besides, he/she can leverage the computing power of the victim to manage its malicious acts. Botnet attacks and infrastructure attacks are two sorts of eclipse attacks on bitcoin peer-to-peer networks. Bots with different ranges of IP addresses initiated the botnet attack. The threat from an ISP, company, or nation-state with contiguous IP addresses is modeled by the infrastructure attack. However, the eclipse attack is a helpful base for other attacks. Table 3 shows some attacks caused by the eclipse attack [3].

5.3.2. DDOS Attack

Distributed Denial-of-Service (DDoS) is the most common networking attack. Mining pools, eWallets, Bitcoin currency exchanges, and other economic bitcoin services are targeted by DDoS. Because the bitcoin network and its consensus protocol are distributed in nature, beginning a DoS attack has no or insignificant negative impact on the functionalities of the network. Therefore, a powerful DDoS should be launched to disturb the tasks of the network. Dissimilar to the DoS attack carried out by a single attacker, various attackers begin the attack concurrently in DDoS. Performing DDoS attacks is inexpensive, although it is completely disruptive. A DDoS can be performed on competing miners by malicious miners through accessibility to a distributed Botnet. The competing miners are expelled out of the network, and the effective hashrate of malicious miners is raised. For disrupting actual user access, the network resources are exhausted by the adversary [10].

Table.3.
Some attacks caused by the eclipse attack [3]

| Attack | Harm |
|---|---|
| Engineering block races | Orphan blocks waste mining power |
| Splitting mining power | Triggering 51% vulnerability |
| Selfish mining | Obtaining more than usual mining rewards by the attacker |
| 0-confirmation double spend | No gaining rewards for vendor service |
| N-confirmation double spend | |

5.3.3. Liveness Denial Attack

The liveness attack, a kind of DoS attack in Proof of Stake protocols proposed by Aggelos et al., can procrastinate as much as possible to confirm a target transaction time. It includes three following phases [3] (shown in Figure 8):

– Attack preparation phase. Before broadcasting the target transaction, TX, to the public chain, an attacker builds an advantage over honest miners, similar to the selfish mining attack. The private chain, longer than the public chain, is produced by the attacker.

- Transaction denial phase. The block, including TX, is held privately to slow down the public chain growth rate by not writing TX into the public chain.
- Blockchain retarder phase. The attacker should publish the privately held blocks, including TX, at the proper time when the public chain grows. TX is valid if the block depth is higher than a constant in some blockchain systems. When TX is valid in the public chain, the liveness attack ends [3].

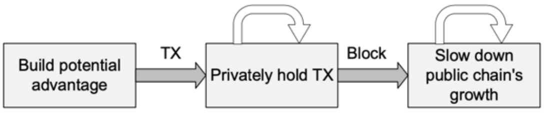

Fig. 8. Overview of the liveness attack process [3]

### 5.3.4. Transaction malleability

Malleability attacks simplifies the DDoS attacks in bitcoin. In a malleability attack, an adversary closes the transaction queue. The transaction queue contains all the unfinished transactions almost supposed to be serviced in the network. Meanwhile, a malicious user puts in false transactions with a high preference to represent itself as the highest incentive payer for the miners. The miners find that these transactions are false when they attempt to verify them. But much time has been spent to verify these false transactions. So the network and the time and resources of the miners are wasted [10].

### 5.3.5. Refund Attack

A refund attack can be performed because of the vulnerabilities in the refund policies of the bitcoin payment protocol. In this attack, the adversary exploits the refund payment protocols. BIP70 is a bitcoin payment protocol that rules whereby customers and vendors perform payments in bitcoin [10].

### 5.3.6. Routing Attack

Regarding both small and large size attacks, [38] presented the routing attack effect on the bitcoin network. Routing attacks are made practical by two main bitcoin network properties. These properties contain the routing manipulation easiness and the fast-rising bitcoin centralization regarding routing and mining power [10].

It contains two different attacks: a partition attack, which isolates the network nodes into discrete groups, and a delayed attack, which manipulates spread messages because of block propagation delay [38].

#### 5.3.6.1. Tampering or Delay Attack

In a bitcoin network, the miners scatter the newly mined block information after mining a block. The novel transactions will be spread now and then in the network. It is supposed that the messages will transfer to the other nodes in the network at a great rate. Speed reduction of propagation is the aim of delay attack [33]. But the malicious user could cause procrastination in spreading the packets. Imposing network congestion or busying a victim node by broadcasting requests to all its ports causes this kind of delay. Such a type of tampering can perform other sorts of attacks in the network [10].

#### 5.3.6.2. BGP Hijacking Attack or Partition Attack

BGP, which stands for Border Gateway Protocol, also called partition attack, is a routing protocol and manages how IP packets are sent to their destination. BGP routing is leveraged or manipulated by attackers to prevent blockchain network traffic. BGP hijacking usually needs network operator control to postpone network messages. The impact of routing attacks, which consists of node-level and network-level attacks on bitcoin, indicates that the number of Internet prefixes hijacked successfully relies on mining power distribution. The attackers can break the bitcoin network or slow down the block propagation speed. Dell SecureWorks in 2014 analyzed that BGP hijacking is used to prevent connections of bitcoin miners to a mining pool server. Stealing cryptocurrency from the victim is possible through rerouting traffic to a mining pool managed by the attacker [3].

### 5.3.7. Packet Sniffing

Receiving and sending transactions can be monitored by an attacker who can observe a node's Internet traffic. The transaction information is not sensitive for the user. Tor, enabling anonymous communication, would decrease the possibility of tracking blockchain personal information [33].

### 5.3.8. Sybil Attack

In a Sybil attack, the attacker creates many identities pseudonymously in the peer-to-peer network by hijacking an unsafe computer and operates these identities in separate nodes to obtain a disproportionately large impact [32]. The attacker controls various nodes in the network, so false nodes surround the victim for double-spending attacks. Some preventive measures can be effective in the

Sybil attack: raising new identity create-cost, needing trust for network joining, or determining reputation-based user power [26].

5.3.9. Spam Attack

In a spam attack [39], [40], slowing down the network and procrastinating the block creation affect a committed transaction.

5.3.10. Time Jacking Attack

There is a time counter within all the participant nodes to maintain the bitcoin network time. Its value relies on the average time of the peer nodes. When a peer connects first, a message is transmitted. However, the network time counter will replace the system time when the average time varies more than 70 minutes from the system time. A malicious user could use multiple fake peers that will state incorrect timestamps. It can slow down or speed up the network time counter of the node. So a malicious user modifies the node time. Then the transaction accept-time window should be decreased. It speeds up the recovery of the node from the attacks. Time jacking can divide the network into multiple sections and separate the victim node. Defining the block timestamp upper limit using the system time rather than network time, tightening the ranges of the admissable time, and using just honest peers are methods proposed to avoid time jacking [10].

5.3.11. Deanonymization Attack

For connecting an IP address to a client, the peer-to-peer network of bitcoin is deanonymized. It is because the node IP address leaked within a transaction spread. Utilizing the network information is a way to connect IP addresses to hosts. In a deanonymization attack, a malicious user utilizes a "supernode" joined with the active peers. He/she monitors the legitimate node traffic in the transaction. Using the symmetric distribution of the network transaction, the probability of connecting the public keys of bitcoin users with their IP addresses is about 30% [10].

*5.4. Private Key/User wallet Attacks*

To access the bitcoin account or wallet, every user has a collection of public or private keys. So, secure management methods are needed to secure the wallet [10].

Losing the user's key can cause missing the coins because there is no TTP. Moreover, account tampering and identity theft may be the result of a stolen key [33].

5.4.1. Wallet theft

Wallet theft utilizes methods that are consisting of wallet wrong usage, buggy software installation, and system hacking [10]. In this attack, the adversary steals or destroys the user's private key that causes loss of bitcoin in the wallet [33].

5.4.2. Man-in-the-middle (Address attack)

A man-in-the-middle modifies the transaction recipient address before signing the transaction instead of directly aiming for private keys. The transaction recipient address is replaced with the thief address by the malware. An address attack is a kind of this attack against hardware wallets users [33].

*5.5. Smart Contract-based attacks*

The blockchain security issues in the smart contracts include source code bugs, the blockchain itself, network VM, and smart contract runtime. The general Ethereum Virtual Machine (EVM) vulnerabilities are immutable defects, the cryptocurrency lost in the transfer, bugs in access control, and short address attacks [26].

5.5.1. DAO Attack

DAO Attack, which stands for Decentralized Autonomous Organization, was a smart contract attack. The DAO contract was attacked after 20 days using. It has raised the biggest crowdfund, about $150M, before 18 June 2016. An adversary stole about $60M until the transactions, which included the malicious activity, were invalid by the blockchain fork. The attacker employed the reentrancy vulnerability. A malicious smart contract contains a withdraw() function call to DAO. The callee will receive Ether sent by the withdraw() function. Hence, the callback function of the malicious smart contract will be invoked again. In this way, all the Ether from DAO can be stolen by the attacker [14]. In other words, an undefine-behavior function is called from contract A to B. In turn, a malicious-purposes function can be called from contract B to contract A [26].

## 6. Privacy and Security Techniques

We discuss the security and privacy requirements of the blockchain in section 3.4. In this section, we display a comprehensive overview of privacy-preserving solutions for blockchain and provide a complete discussion on techniques for improving the security and privacy of blockchain.

### *6.1. Mixing*

Mixing is a random exchange of a user's coins with others. The ownership of coins is obfuscated for the observer, but mixing services do not protect from coin theft. Senders and receivers of a transaction are linkable in the blockchain. Some privacy information can be inferred by analyzing the public content (like the analytical attack). Obfuscating the transaction connections with a mixer (also named tumbler or laundry) is one solution to decrease mixing attacks. In a mixing service, users conceal the communication content, the correspondences between each originator and message destination, and whom a participant communicates with [13].

The blockchain services obfuscate the transaction history and mitigate the risk of de-anonymization. This research focuses on two main methods: centralized mixing and decentralized mixing.

#### 6.1.1. Centralized Mixing Services

Different mixing websites are available like OnionBC, Bitcoin fog, Bitmixer, Helix Light by Grams, Bit laundry, and Bitblender. These websites mix transactions anonymously at the cost of some service fees. They act as online mixers and exchange the transactions among several users to conceal the incoming and outgoing transaction relationship [13].

These sites have two principal disadvantages: (i) The service provider could be a possible attacker and steal user assets by not transferring them to the receivers. (ii) The service providers are in the middle, so they continuously maintain logs to route the transactions for a specific time.

One solution to solve the first problem is conditional execution, which means that the mixer can get a reward only if it operates correctly, oppositely receives nothing, Like CoinSwap [71], a third-party-based mixing protocol.

Auditing the misbehaved mixer, which means using irrefutable evidence for controlling the mixer activities, is another solution to solve the first problem. For example, Mixcoin [72] uses a signature-based accountability mechanism to detect stealing if the mixer has misbehaved. Mixcoin increases the anonymity set to allow users to mix coins concurrently.

A blind signature scheme, a digital signature in which the message is blinded and then signed, is a beneficial solution for the second question. The methods involve three procedures: blinding, signing, and unblinding. In the blinding procedure, a random "blinding factor" covers the actual message. In the signing procedure, the blinded message is signed using the standard sign algorithm. The unblinding procedure removes the "blinding factor" to get a valid signature on the original message. For example, mixing the blind signature method with an append-only public log proposed Blindcoin [73] to keep the mixing process accountable and give evidence in the misbehaved mixer.

TumbleBit [74] achieves full unlinkability and avoids coin stealing simultaneously. It is based on a centralized mixing service but uses secure two-party computation and zero-knowledge proofs for privacy-preserving.

As a result, the centralized mixing services mainly have three limitations: (1) Waiting delay is high for enough online participants to be mixed. (2) The centralized mixing server may be vulnerable to denial of service (DOS) attacks and remains a single point of failure. (3) Users should pay high mixing fees.

#### 6.1.2. Decentralized Mixing Services

A decentralized mixing method is proposed to lessen the DOS attack caused by the centralized services. This method does not need a third party and enables untrusted peers to distribute their messages simultaneously in an anonymous way. Another advantage of this method is the removal of mixing fees.

For example, the core idea of CoinJoin [64] is to make a joint payment. When there is one transaction from user A to user C and another transaction from user B to user D, these transactions can be mixed into one CoinJoin transaction if their inputs and outputs are fixed. In this method, the exact data flow direction will be remained anonymous to the other peers by mixing the link between inputs and outputs.

CoinJoin has three main drawbacks: (i) Participants will know the details, including transaction destinations, about the joint transaction. So, this method lacks internal unlinkability. The possibility of a Sybil attack will increase with the growth in the number of available participants. (ii) The Denial-of-Service (DoS) attack may occur in this method. This attack can block the mixing process by denying to sign the transactions in CoinJoin. (iii) The maximum number of participants (N) is a practical concern in CoinJoin. Increasing the vulnerability of DoS attacks and exponential communication overhead are reasons for this maximum. While N is small, the impact of anonymity and unlinkability will be lower [2].

CoinShuffle [65] was proposed in 2014 to obtain internal unlinkability. CoinShuffle extends the CoinJoin concept and improves privacy by avoiding the necessity of a trusted third party for mixing transactions. To conceal the participant identities from each other, CoinShufflet uses an anonymous group communication protocol which is called Dissent.

### *6.2. Anonymous Signatures*

There are several variants of the digital signature scheme. Some can provide anonymity for the signer, called the anonymous signature. There are two principal anonymous signature schemes: group signature and ring signature. A group manager entity is defined in Group Signatures which anonymizes any signature by defining the set of users in a group. Nevertheless, any user can define a custom set of users in the Ring Signature method and sign a message without revealing the origin of the signer [2].

#### 6.2.1. Group Signature

Group signature, proposed initially in 1991, is a cryptography method. In this method, any group members can sign a message for the entire group anonymously using their secret key. Any of them can control and validate the created signature by the group public key. The real identity of the signer, except the group membership, does not reveal in the signature verification process.

The group manager handles adding group members and the happening of disputes, like exposing the original signer. An authorized entity also is needed to create/revoke the group, add new members to the group, and delete some participant membership from the group in a blockchain system [12]. An example of this method is PlatON [75] added a group signature in its platform for anonymizing.

#### 6.2.2. Ring Signature

The ring signature is a kind of digital signature performed by one of the group members that sign a message on behalf of the ring of members but does not reveal which member produced the signature. The name of the ring signature comes from the signature algorithm that applies the ring-like structure [12]. The principal concept of this method is to choose a set with no central manager for improving privacy in blockchain [13].

Signing by any group member causes anonymity for the signer in a ring signature because defining which group member uses his/her key to sign the message is difficult. Ring signatures are different from group signatures for two main reasons: (i) Because there is no group manager in a ring signature, the original identity of the signer cannot be exposed in the happening of a dispute. (ii) Any users can provide a ring by themselves with no extra setup [12].

Two main existing ring implementations achieve anonymity in the blockchain: CryptoNote and Monero. Ring-based privacy preservation protocols (e.g., [76]-[78]) are some existing ring implementations that achieve anonymity and linkability for blockchain. Users can sign only one valid transaction with one private key in CryptoNote [76]. It decreases the double-spending attack by replacing the tag with a key image computed from the one-time private key of the user. The signer identity is not distinguishable from the other users with public keys in the set until the owner creates a second signature by the same key pairs.

A CryptoNote employs a one-time-key pair for each transformation at the receiver end, also for the same sender and receiver. Each CryptoNote output destination is a unique public key that originated from the one-time address of the receiver and random data of the sender [13]. Ring Confidential Transaction (RingCT), proposed by Noether [77], is an improvement of CryptoNote. It hides the amount, which can simultaneously provide identity privacy and transaction privacy. Monero [78] did the most successful of this method implementation. However strong anonymity is provided by the ring signature, it has three limitations: (i) Transactions, especially RingCT, have a large size. It is about thousands of bytes per transaction and will grow the storage space in the entire blockchain. (ii) The size of the signature is directly proportional to the participant numbers that is a drawback of a ring signature. So the number of foreign outputs is confined in each transaction. By default, Monero uses four outputs in each transaction. (iii) Auditing, which confirms whether new cryptocurrencies have been created privately in the transaction, is complex due to the hidden amount [13].

### *6.3. Homomorphic Encryption (HE)*

Homomorphic encryption, which introduces privacy homomorphism, is another method for privacy-preserving.

Homomorphic encryption methods can operate over the ciphertext with the identical encrypted result on the cleartext [2] and store data without significant changes in the blockchain properties. So, it ensures the encryption of data on the blockchain. Ready access to encrypted data on the public blockchain for auditing, such as handling expenses of the employee, is provided by using the homomorphic encryption method [12]. The RSA encryption method is an example of homomorphic hiding, one of the fundamental means for creating zkSNARKs and private-distributed computations. Bitcoin ECDSA key pairs use homomorphic encryption with additive and multiplicative homomorphic properties [2].

There are two typical homomorphic cryptographic implementations: Pedersen commitment scheme [79] and Paillier cryptosystem [80]. Pedersen's commitment scheme supports homomorphic operations, including addition or multiplication, on the commitments.

The Paillier cryptosystem is an efficient additive homomorphic encryption system in privacy-preserving financial scenarios. A framework is designed by Wang et al. [13].

### 6.4. Secure Multi-Party Computation (SMPC)

Through secret sharing, data or program states are splits between N parties by SMPC. Some of the N parties generate the output and expose data. Each party received only part of the input, and an adversary does not learn about the original party input [12]. The participants' majority should be honest in this scheme. Working as a part of the MPC or managing the incentives to participants is difficult for them [2].

Enigma is a decentralized SMPC platform proposed in 2015 by Zyskind et al. [81] and uses an advanced version of SMPC. Enigma uses a valid secret-sharing method to guarantee computational privacy. It can control and protect personal data like the Bitcoin system, and also it does not need a trusted third party.

### 6.5. Non-Interactive Zero-Knowledge(NIZK) Proof

In a Zero-Knowledge Proof (ZKP), a cryptographic protocol, a party can prove the correctness of a given statement to another entity without exposing any information except the accuracy of the proof. The party is called the prover or the certifier, and the entity is called a verifier. For example, ZKP can prove the statement of knowing a secret value conserved private to the prover.

There are three properties for a ZKP protocol: Completeness, Soundness, and Zero-Knowledge. (i) Completeness means that if the proved statement is true, the prover can always perform a proof successfully. (ii) Soundness means that if the proved statement is false, the verifier cannot be convinced true by a deceiving prover, except for a short possibility. (iii) In Zero-Knowledge, a simulator, i.e., a polynomial-time bound algorithm, can generate on the protocol transcriptions that a successful proof between a prover and a verifier is not distinguishable. Both the verifier and an eavesdropper couldn't get any additional information from an original transcript if a third party, which does not distinguish whether the statement is true or false, can produce a valid protocol's transcript.

The applicability of ZKP is confined to synchronous scenarios of the certifier and verifier because a ZKP protocol is interactive. In practice, instead of the verifier, the prover produces a proof with a hash function [2].

In the Non-interactive variant of zero-knowledge proofs(NIZK), computational zero-knowledge can be achieved with no need for interaction between the certifier and the verifier. When money is transferred in a blockchain application, a user can easily prove that the balance of another user is enough for the transfer with ZKP without exposing the balance of the account [13].

Zerocoin [82], a ZKP based cryptocurrencies, uses NIZK to stop transaction graph analysis because of its three properties: completeness, soundness, and zero-knowledge. The chief concept behind this project is similar to decentralized mixing, where a mined coin is replaced with a new one without historical information. In contrast to Zerocoin, Zerocash [83] obtains the highest level of simultaneous anonymity and transaction privacy preservation for the blockchain with the high computational costs when generating the transaction proofs [13]. Zerocash performs better than Zerocoin because it lessens the transaction size and verification time, conceals transaction amounts, and supports transactions of any kind [2].

### 6.6. Commitment schemes

A commitment scheme is a cryptographic method to conceal a secret value and simultaneously binding a party, like Alice, to real value when she shows the real to Bob. If Alice is lying, Bob can verify. So Alice commits to a secret value without revealing it. In the blockchain, they are used to conceal the value of transactions and bind the owner to the attributes of the real secret, e.g., Zerocoin or Zcash. A Commitment scheme is based on unconditionally binding or hiding. Unconditional hiding protects the private value saved in the unchangeable chain, and preserving the ledger from likely attackers is unconditional binding [2].

The Confidential Transaction (CT) [84] was first proposed with the range proof method to the transaction privacy preservation of the blockchain. In CT, random blinding factors commit the amounts of the transaction before sending to the recipients. Then the recipients approve them [13].

### 6.7. Differential Privacy

Differential privacy is a privacy preservation method that uses to study if information about an individual is revealed or not by a data analysis methodology.

It includes a specific amount of random noise for data queries so that any statistical analysis over the entire set is remarkably near to the actual consequences. But it is impossible to deduce over any individual. Differential privacy applies to access private databases through queries. It collects the data and gets the individual data with statistical changes from the sources while the PII gathered from individuals is decreased [2].

In the blockchain, Differential privacy protects user's privacy in various scenarios. In the first database scenario, the third parties can use the anonymized data. The second instance of a database scenario is applicable for log or sensor data gathering blockchains, where the entire chain can be utilized for statistical analysis. But a single transaction has changed data statistically.

Nevertheless, there is a trade-off between privacy and utility. Differential privacy cannot completely anonymize data when being beneficial for analysis. Differential privacy methods can achieve certain degrees of privacy and the number of performed queries over time [2]. For example, [86] employs Differential Privacy to avoid an adversary.

### *6.8. Data Protection Methods*

Because of the structure of the linked hash pointers list, Blockchain is not changeable. A deleted or modified transaction or block can change the block hash pointer. Changing the block hash pointer is against privacy-preserving principles and regulations like the General Data Protection Regulation (GDPR). Different encryption methods can protect the data running on Blockchain by achieving confidentiality and privacy. So, it is appropriate in scenarios such as eHealth [2].

#### 6.8.1. Asymmetric Encryption

Different encryption and storage methods can be applied according to the consumers that encrypted data. In traditional symmetric or asymmetric encryption, the data creator would upload the transaction encrypted and then scatters the decryption key off-ledger or utilizing a typify of a decentralized PKI on Blockchain similar to Sovrin for managing public keys [2]. Asymmetric cryptography offers better security using two different keys for encryption and decryption than symmetric cryptography that uses a single key. A public key only gets used to encrypt data and makes it safe for anyone to have. A private key decrypts data that never needs to be shared [86]. A random number algorithm usually generates the private key, and the public key is calculated by executing an irreversible algorithm. The advantage of asymmetric encryption is to separate public and private keys by broadcasting over unsecured channels. Similarly, its disadvantages are low speed in processing and little strength in encryption [23].

#### 6.8.2. Attribute-Based Encryption (ABE)

Different ways to sharing ciphertext between multiple peers are concentrated on authorizing a set of nodes to decrypt the data based on attributes [2]. ABE, proposed in 2005 with a single authority, is a cryptographic scheme that the secret key of a user's attributes is defined for the ciphertext encrypted. If the user attributes match with the properties of the ciphertext, he/she can decrypt the encrypted data using the secret key. The collusion-resistance is a security ABE property. If a malicious user colludes with others, he/she only can decrypt data with his/her private key and cannot access other data [12]. After 2005, multiple authorities proposed several additions of ABE.

In Key-Policy Attribute-Based Encryption (KP-ABE) or Ciphertext-Policy Attribute-Based Encryption (CP-ABE), the users with the right attributes can decrypt the data by defining access control policies in the encryption [2].

#### 6.8.3. Secret Sharing

Another method is Secret Sharing, also named a (t, N )-threshold scheme, originated separately in [87] and [88]. It splits a document into N various pieces and shares them with N different nodes. The original document can be built when t out of N nodes cooperate, or one node gets t out of N shared pieces.

#### 6.8.4. Transaction removal

Subject to the blockchain application, certain transactions may only use to verify the hash integrity of the chain or may contain private information that the user wants to remove from the chain. To solve this problem, [89] is proposed to change the structures of blockchain data to provide removal of a transaction, with no effect on the validity of the hash integrity of the chain. It alters the unchangeableness property of a blockchain by integrity, as the validation of hash would still achieve, and allows the removal of a transaction by the rules of consensus on the network. Although, it doesn't guarantee that the privacy issue is solved. The data is copied in all blockchain nodes, and some of them may still save it after removal from the chain.

### *6.9. Smart Contracts*

The smart contract is another technology that has appeared where programs running in the blockchain are defined by users arbitrarily. Considering the programmability aspect with no revealing of transactions and data in cleartext to the public (i.e., no party involved in the contract) is significant. Providing transactional privacy and programmability simultaneously in the blockchain was the first attempt in this method. This scheme is formed on the concept of Zerocash and the smart contract system. Users send the information, is encrypted and committed, to the smart contract, and depend on the NIZK proofs, confirm the accuracy of contract execution and currency transfer. The whole sequence of transaction operations in the contract are kept private from the public when the smart contract's result can be publicly validated [13].

### 6.9.1. The Trusted Execution Environment (TEE) Based Smart Contracts

TEE is an execution environment that provides a fully isolated environment for the execution of the application, which efficiently stops tampering with other software applications and operating systems and learning the application's running state. The Intel Software Guard eXtensions (SGX) is an implementation of the TEE scheme. Ekiden [90], an example of the SGX-based solution for preserving the confidentiality of smart contracts, discrete computation from consensus. Computing of smart contracts in TEEs are performed off-chain on compute nodes. Then a remote protocol of certification is used to verify the accuracy of the execution on-chain on compute nodes. These nodes are utilized for maintaining the blockchain and do not need the trusted hardware use. In Enigma [81], which uses hardware privacy technology TEE, users preserve privacy on smart contracts by an algorithm of decentralized credit scoring. The number and types of accounts, history of payment, and use of credit are factors for credit scoring [12].

### 6.9.2. Game-based Smart Contracts

TrueBit [91] and Arbitrum [92] represented the game-based solutions for smart contract verification.

Applying an interactive verification game, TrueBit determines if a computation task is performed correctly. TrueBit gives rewards to players to control computation tasks and discover bugs to perform a computation task securely with correct properties in a smart contract. TrueBit considerably lessens the computational burden on its nodes via recursively controlling a smaller and smaller subset of the computation by the verifier in each round of verification game [12].

An incentive mechanism of off-chain verification of virtual machine behavior, designed by Arbitrum, only needs the verifiers to confirm the contracts' digital signatures. For recognizing and punishing dishonest parties, who lie about the virtual machines' behavior, Arbitrum uses its efficient challenge-based protocol [12]. Arbitrum is a scalable, private, game-based smart contract to confirm if the computation was done perfectly. The Arbitrum protocol has four phases: (i) the verifier; (ii) the key; (iii) the virtual machine; (iv) the manager, as shown in Figure 9 [93].

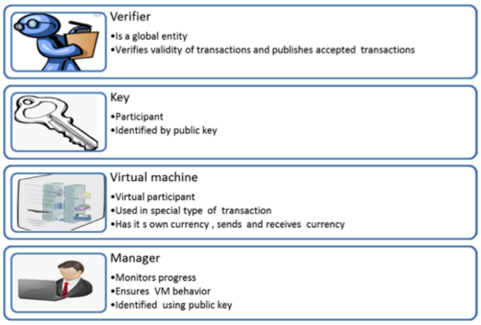
Verifier
•Is a global entity
•Verifies validity of transactions and publishes accepted transactions
Key
•Participant
•Identified by public key
Virtual machine
•Virtual participant
•Used in special type of transaction
•Has it s own currency , sends and receives currency
Manager
•Monitors progress
•Ensures VM behavior
•Identified using public key

Fig. 9. Roles involved in Arbitrum [93]

### *6.10. Another Classification of Privacy-Preserving Mechanisms*

According to the principal privacy-preservation purpose, the privacy-preserving methods can be classified into four categories that shows in Table 5:

- Identity Data Anonymization. This category hides the user identity in transactions. This privacy-preserving mechanism contains Mixing Services that hides the payee and payer, Ring Signature scheme that anonymizes signer, Homomorphic Hiding, ZKP, and Commitment Scheme.
- Transaction Data anonymization. This category preserves the contents' privacy of the blockchain transactions. Mechanisms with this privacy-preservation category contain Mixing that anonymizes traded coins, Homomorphic Hiding that conceals the original amount of each transaction, ZKP, and Differential privacy schemes.
- Smart Contracts and Key Management that employs SMPC methods.
- On-chain Data Protection Method. This group, which protects the blockchain data by encryption method, contains Asymmetric Encryption, ABE, and Secret Sharing.

Some privacy-preserving methods anonymize both identity and transaction data, e.g., Mixing Services, ZKPs, and Homomorphic Hiding [2].

### 6.11. *Discussion*

There are three remarks to perform security and privacy on the blockchain, needing desired properties:

- A single technology is not a solution for a secure and private blockchain. There is no single technology solution for the security and privacy of blockchain. According to the security and privacy requirements and the application context, the proper techniques should be selected. Generally, using the integrating of multiple methods is more efficient than a single method. For instance, Enigma [81] uses the integration of SMPC and TEE.
- A technology with perfection in all aspects or no defects doesn't exist. A new form(s) of attack(s) or problem(s) can be generated when a new technology is added to a complex system. So careful attention to the pitfalls and possible harms caused by combining some security and privacy methods into the blockchains is needed.
- A trade-off between privacy, security, and efficiency always exists. The techniques that enhance the security and privacy of blockchain and increase the practical deployment of applications in blockchain with a satisfactory performance simultaneously should be supported [12].

We summarize security and privacy techniques and cryptographic policies used in blockchain containing their significant advantages and disadvantages in Table 6. As explained before, mixing services (centralized and decentralized) intend to preserve the relationship between the sender and receiver address. What causes the difference is that the former requires a centralized mixer to do a task, provided the latter performs the mixing simultaneously among the participants. So, some disadvantages have existed: (i) Extra delay of waiting for mixing. (ii) No protection on transaction content because of the fixed denomination need in a mixing. (iii) Some unique constraints like needing high mixing service fees. (iv) Possible vulnerability of single point of failure. (v) Possibility of rejecting protocols' execution, that destroys the mixing process, by participants. (vi) limitation in the scalability of a mixing session. In decentralized mixing, the communications frequency between participants will create high overheads of transmission on the network channel.

The ring signature conceals the signer's identity, but it does not secrete the message to be signed. The restriction of this method is the signature size that is proportional to the participant numbers. The storage and costs of the communication are grown by the addition of participants. Moreover, there is a restriction on the participant numbers in the signing stage to maintain the transaction size within a reasonable range. It facilitates the sender address analysis [13].

The homomorphic cryptosystem (HC) preserves the transaction contents in computing protected data. The NIZK 's combination method with the commitment provides a complete privacy preservation structure for the blockchain like Zcash. The coin ownership is verified by NIZK proof in an anonymous and unlinkable way. The commitment scheme hides the transaction content. So, with the combination of these two schemes, Zcash can provide extensive user anonymity and the highest protection on the content of the transactions simultaneously with no size restriction of anonymity set in each transaction. However, the NIZK protocol causes high overheads of computation.

As explained before, there have been many attempts for privacy preservation in the blockchain that concentrated on the following notes:

- Relationships of the transaction are obfuscated to prevent analysis of linking or tracing;
- The primitives of complex cryptography conceal the sender and the receiver identities;
- The content of transactions is blinded during the same time that preserving the verifiability and computability [13].

## 7. Conclusion

Blockchain decentralized technology can solve distrust problems of the traditional centralized network and enhance the privacy and security of data. It provides a distinct way of storing and sharing data through blocks chained together.

In this survey, we concentrated systematically and comprehensively on the existing attacks, privacy, and security issues associated with the blockchain. At first, we studied challenges and problems with the blockchain. We also reviewed the consensus algorithms employed in the blockchain. For each attack, we analyzed its target, causes, negative impacts, and possible preventive measures.

Moreover, we surveyed privacy and security requirements and techniques on the blockchain. Finally, we comparatively summarized blockchain security pros and cons and their applications and projects.

Table.4.
Blockchain Attacks

| ***Attack*** | ***Description*** | ***Target*** | ***Negative Impact*** | ***Possible Preventive Measures*** |
|---|---|---|---|---|
| Double-Spending Attack | Use the same bitcoins for more than one transactions | Bitcoin transaction<br>Pow Consensus<br>Merchant/ seller | Generate blockchain fork<br>Deny legitimate clients' service<br>Lose merchants' Products | Monitor the network, and send the message of double spending alerts to peers [41]<br>Neighbor peers should inform the merchant about the double-spending attack [42]<br>Inactive incoming connections [43], [35]<br>Transaction of recipient oriented [44] |
| Race Arrack | Admit a payment before confirming the transaction in blockchains based on PoW | Vendors | Losing a product by a vendor<br>Generating blockchain forks<br>Banning legitimate users | Monitor the network, and send the message of double spending alerts to peers [10]<br>Neighbour peers should inform the merchant about the double-spending attack [10]<br>Inactive incoming connections and select specific outbound connections [45] |
| Finney Attack | Attackers mine a blockchain fork privately. When the purchased product is received, they spread the mined blockchain fork over the network. | Merchant/ seller<br>Pow Consensus | Generate blockchain fork<br>Deny legitimate clients' service<br>Lose merchants' Products | The merchant should wait for multi-confirmations before admitting the payment and transferring the product [33] |
| Brute Force Attack or Alternative History Attack | Attackers mine a blockchain fork privately | Computing Power<br>Pow Consensus<br>Merchant/ seller | Generate large blockchain forks<br>Deny legitimate clients' service<br>Lose merchants' Products | Monitor the network, and send the message of alerts to peers [41]<br>Neighbour peers should inform the merchant about the double-spending attack [43]<br>Inactive incoming connections [14] |
| Vector 76 or One-confirmation attack | Generate a deposit transaction, followed by a new fork(F), and then a withdrawal transaction. The attack occurs when the deposit transaction is denied. | Bitcoin exchange services | Generate blockchain fork<br>Deny legitimate clients' service<br>Lost large amount of bitcoin | Inactive incoming connections and select specific outbound connections [33]<br>The merchant should wait for multi-confirmations before transferring the asset [14] |
| Balance Attack | Momently disruption of communications between similar-mining-power subgroups by a low-mining-power attacker | Block | Allow double-spending | None |
| Nothing at Stake Attack | Validators motivate to operate on various forks and make conflicting blocks on all forks that are probable | Block | Lessen system efficiency<br>Reduce the consensus time in the network | Slasher Protocol [10] |
| Selfish mining or Block Discarding Attack | Produce forks in blockchain, consider the longest blockchain, and discard the rest. | Legitimate miners or mining pools | The attacker can obtain more than normal rewards of mining | Freshness preferred [10] that is a timestamp-based method<br>ZeroBlock technique [10], [46]<br>DECOR+ protocol [47] |
| Long-Range Attack | In Long-Range attacks with PoS protocols, an attacker adds blocks like selfish mining attacks with PoW protocols | Database | Change transaction history | Implement trusted hardware [48] |
| Block With Holding (BWH) Attack | Submit partial PoW | Legitimate miners (honest miners) or mining pools | Drop the network capital<br>Deplete peers resources<br>Reduce the pool revenue | Methods of cryptographic commitment [10]<br>The network includes honest miners [10]<br>Dissolve a mining pool while income decreases from expected [36] |
| Fork After Withholding (FAW) Attack | Improve on negative impacts of selfish mining and BWH attack | Legitimate miners or mining pools | Drop the network capital<br>Deplete peers resources<br>Reduce the pool revenue | None |
| Bribery Attack | Adversary bribe the nodes to mine for them | Merchant<br>Miner, Mining nodes | Increases the possibility of a double-spending or BWH attack | Raise the honest miners' rewards [10]<br>Inform the miners of the bribery's long-term losses [10] |

| ***Attack*** | ***Description*** | ***Target*** | ***Negative Impact*** | ***Possible Preventive Measures*** |
|---|---|---|---|---|
| Goldfinger or >50% Hash power or Majority Attack | Having more than half of the computing resources | Mining nodes, Miners<br>Users, Clients<br>Bitcoin network | Deny legitimate users' service<br>Weaken consensus protocol | Monitor the network, and send the message of double spending alerts to peers [41]<br>PieceWork [49], TwinsCoin [50]<br>Unmotivated huge mining pools [51], [52] |
| Punitive and Feather forking | Illegitimate miners blacklist the particular address' transactions | Users | Suspend the user's bitcoins forevermore | None |
| Eclipse Attack or Netsplit Attack | An attacker monopolizes all of the incoming and outgoing relationships of the victim | Mining nodes, Miners<br>Users, Clients<br>Bitcoin network | Inconsistent network and blockchain's view<br>Multiple authentications<br>enable the concept of double-spending | Inactive incoming connections [53]<br>Use whitelists [53] to select specific outbound connections with known or well-connected mining nodes |
| DDoS Attack | Exhaust network resources | Mining pools, Miners<br>Clients<br>Bitcoin network | Deny of services to honest miners<br>Discrete mining nodes or miners | Proof-of-Activity (PoA) protocol<br>Authentication based on signature [14]<br>Employ just trusted peers [54]<br>Make use of NTP (Network Time Protocol) [10]<br>Utilize the system time of the node rather than the network time [33] |
| Liveness Attack | procrastinate as much as possible to confirm a target transaction time | Block | Delay transaction time | |
| Transaction malleability Attack | Malicious user modify the TXID with no validating the transaction | Bitcoin exchange centers<br>Users | Exchange losses assets due to the increase in double credit or double debit | Multiple transaction verification metrics [55]<br>"refund" transaction [56]<br>Using a legal signature, independent id for signature [57] |
| Refund Attack | Adversary exploits the refund payment protocols | Merchants<br>Users | Loss of money by merchants<br>Loss of legitimate miner's reputation | Publicly authenticated evidence [58], [59] |
| Routing Attack | Separate a set of nodes from the Bitcoin network<br>Procrastination of block propagation | Miner, Mining nodes<br>Users, Clients | DoS attack<br>Waste the pools mining power<br>Mount fork rate<br>Mount 0-confirmation double-spends probability | Mount the diversity connections of the node [38]<br>Supervise round-trip time [38]<br>Utilize gateways in diverse Ases [38] |
| Tampering or Delay Attack | Delay the spread of blocks and transactions to nodes | Mining nodes, Miners<br>Users, Clients<br>Blockchain network | Increase DoS attacks<br>Incorrectly increase mining advantage<br>Possibility of a double-spend attack | Improve management system of block requests [60]<br>Tampering reductions like round-trip time (RTT) monitoring and UDP heartbeats [38] |
| BGP Hijacking Attack or Partition Atatack | Prevent connections of bitcoin miners to a mining pool serve | Mining nodes, Miners<br>Users, Clients<br>Blockchain network | False transaction<br>Split the bitcoin network<br>Slow down block propagation speed | A human-driven process including configuration's modifying or attacker disconnection [48]<br>Security extensions on BGP [33]<br>System monitoring [3] |
| Packet Sniffing Attack | monitor transactions by an attackr | Single User | Not provide anonymous transaction | Using Tor, enabling anonymous communication, to decrease the probability of tracking blockchain personal information [33] |
| Sybil Attack | The adversary is in charge of making multiple virtual Identities | Mining nodes, Miners<br>Users, Clients<br>Blockchain network | Pseudonymous identities<br>Menace privacy of user<br>Enable Double-spend and DDoS | Use a protocol known as Xim, i.e. a two-party mixing protocol [10]<br>Restrict the outbound connections to one IP address per /16 (x.y.0.0) [33] |
| Spam Attack | Slowing down the network and procrastinating of the block creation affect a committed transaction | Blockchain network | Slow down network, transaction, and computing Power | Constant fee of the nominal transaction [61] |
| Time Jacking Attack | The malicious user speeds up the majority of the mining nodes' clock | Miner, Mining nodes | Separate a miner<br>Waste all miner resources | Put limitations on tolerance ranges [54]<br>Use Network Time Protocol (NTP) or time sampling on the obtained values from honest peers [62] |

| ***Attack*** | ***Description*** | ***Target*** | ***Negative Impact*** | ***Possible Preventive Measures*** |
|---|---|---|---|---|
| Deanonymization | Connect IP addresses with a client in Bitcoin wallet | Users | Violate user privacy | Mixing services [63]<br>CoinJoin [64],<br>CoinShuffle [65] |
| Wallet theft | the adversary steals or destroys the user's private Key | Businesses<br>Users, Clients | Loss of bitcoin in the wallet | Hardware wallets [66]<br>Two-factor security of threshold signature-based [67]<br>Password-Protected Secret Sharing (PPSS) [68]<br>Bitcoin wallet supported with TrustZone [69] |
| Man-in-the-middle (Address attack) | Replace the transaction recipient address with the thief address | Users | Theft from the wallet | Stop man-in-the-middle strategies like Intrusion Detection Systems (IDS) [33] |
| Decentralized Autonomous Organization (DAO) Attack | Recalling the function of the malicious smart contract | Computing Power | Fake transaction | Use hard/soft fork [70] |

Table.5.
Privacy-preserving techniques for blockchain

| ***Techniques*** | | ***Identity Data Anonymization*** | ***Transaction Data anonymization*** | ***Smart Contracts and Key Management*** | ***On-Chain Data protection*** |
|---|---|---|---|---|---|
| Mixing Services | | Yes | Yes | No | No |
| Anonymous Signatures | Ring Signature | Yes | No | No | No |
| Homomorphic Encryption (HE) | | Yes | Yes | No | No |
| Secure Multi-Party Computation (SMPC) | | No | No | Yes | No |
| Zero-Knowledge Proofs (ZKPs) | Interactive ZKPs | Yes | Yes | No | No |
| | Non-Interactive Zero-Knowledge (NIZK) Proof | Yes | No | No | No |
| Commitment Schemes | | Yes | No | No | No |
| Differential Privacy | | No | Yes | No | No |
| Data Protection Method (Asymmetric Encryption, ABE, Secret Sharing / threshold) | | No | No | No | Yes |
| Transaction Removal | | No | No | No | Yes |
| Smart Contracts | | No | No | Yes | No |

| ***Techniques*** | | ***Description*** | ***Disadvantages*** | ***Advantages*** | ***Applications & Projects*** |
|---|---|---|---|---|---|
| Mixing Services | Centralized | Coordinate a set of users to perform transactions by concealing the originator | 1) Waiting delay<br>2) No protection on transaction content<br>3) Single point of failure<br>4) Need high service fees | 1) Prevent linking users' addresses<br>2) Operates on existing solutions | Mixing Websites [13], CoinSwap [71], Mixcoin [72], Blindcoin [73], TumbleBit [74] |
| | Decentralized | | 1) Waiting delay<br>2) No protection on transaction content<br>3) Sybil attack<br>4) Heavy overhead of communication | 3) Efficiency | CoinJoin [64], CoinShuffle [65] |
| Anonymous Signatures | Group Signature | Any of the members of the group can sign a message for the entire group anonymously using the secret key | Need a trusted third-party( group manager) | 1) Hide the identity of the signer among a group of users.<br>2) Expose the identity of the signer, in the event of a dispute. | PlatOn [75] |
| | Ring Signature | One of the group members signs a message for the | 1) Management and coordination of different signer entities are difficult. | 1) Hide the identity of the signer among a group of users. | CryptoNote [76], RingCT [77], |

| ***Techniques*** | ***Description*** | ***Disadvantages*** | ***Advantages*** | ***Applications & Projects*** |
|---|---|---|---|---|
| | entire group but does not reveal which member produced the signature | 2) Not to Expose the identity of the signer, in the event of a dispute.<br>3) Heavy overhead of storage<br>4) No protection on signed data and transaction target<br>5) Restricted size of anonymity set | 2) No need a trusted third-party | Monero [78], |
| Homomorphic Encryption (HE) | Translate arithmetic computations in ciphertext to cleartext | 1) Confined homomorphic operations<br>2) The implementation efficiency of only some kinds of operations, such as addition and multiplication<br>3) The poor computational efficiency of complex functions<br>4) No support for auditing | 1) Distributed computation with private values<br>2) Achieve privacy-preserving computation with direct computing on the ciphertext<br>3) No need a trusted third-party | Pedersen commitment scheme [79],<br>The Paillier cryptosystem [80] |
| Secure Multi-Party Computation (SMPC) | Split data between N parties through secret sharing and generate the output through jointly performing the input's distributed computation by some of N parties | Support only some simple functions and poor efficiency of complex functions | 1) Without breaking the input privacy, multi-party can jointly perform some computation over the private data inputs<br>2) No need a trusted third-party | Enigma [81] |
| Non-Interactive Zero-Knowledge (NIZK) Proof | Proof a statement without exposing the private information and needing for interaction between the certifier and the verifier. Conceal transaction relationship and content. | 1) Less efficient<br>2) Heavy overhead of computation<br>3) High computational costs in ZeroCash | 1) Prove that the user has enough balance for the transfer with NIZK without exposing the account balance<br>3) No need a trusted third-party | ZeroCoin [82]<br>ZeroCash [83] |
| Commitment Schemes | Conceal a value when binding the user to real value | Either unconditional hiding or unconditional binding not both | 1) Efficient<br>2) Computational binding and hiding<br>3) Need extra ways for transaction anonymity<br>4) No need a trusted third-party | Confidential Transaction (CT) [84] |
| Differential Privacy | Statistical data change with irrecoverable individual data | Intrinsically applicable to blockchain | Data Statistical usefulness in individual privacy | [85] |
| Data Protection Methods<br>Asymmetric Encryption | using two different keys: public key and private key | low processing speed , low encryption strength | Separate public and private keys, which can be transmitted over unsecured channels | None |
| Attribute-Based Encryption (ABE) | When the user has valid attributes, ciphered data can be deciphered | Need the credential system to issue and revoke attribute certificates in a distributed environment | 1) Policies determine authorized users<br>2) Achieve confidentiality of data and fine-grained access control simultaneously | [94], [95] |

| ***Techniques*** | | ***Description*** | ***Disadvantages*** | ***Advantages*** | ***Applications & Projects*** |
|---|---|---|---|---|---|
| | Secret Sharing /threshold | A document is divided and shared between N parties. The document can be reconstructed if t out of those N parties cooperate. | Collusion attacks | Recovery of the document doesn't need all nodes | [87], [88] |
| Transaction Removal | | Substitute a transaction in the chain for the validity of integrity calculation | 1) Not guarantee deletion<br>2) Should approve the operation of deletion in consensus | Right to remove private data | [89] |
| Smart Contracts | The Trusted Execution Environment (TEE) Based Smart Contracts | Provide a fully isolated environment for the execution of the application | 1) Need to equip the compute nodes with a CPU including TEE, e.g. Intel SGX<br>2) Need to resolve the attacks on SGX | Privacy- preservation of smart contracts through running them in TEE | Ekiden [90]<br>Enigma [81] |
| | Game-based Smart Contracts | Verify the correctness of smart contracts | Risk of being deceived by a malicious user | Encourage parties to verify whether smart contracts are true by incentive mechanisms | TrueBit [91], Arbitrum [92] |

## References

[1] A. Narayanan, J. Bonneau, E. Felten, A. Miller, and S. Goldfeder, "Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction," ed: Princeton University Press, 2017.

[2] J. B. Bernabe, J. L. Canovas, J. L. Hernandez-Ramos, R. T. Moreno, and A. Skarmeta, "Privacy-preserving solutions for blockchain: Review and challenges," IEEE Access, vol. 7, pp. 164908-164940, 2019.

[3] X. Li, P. Jiang, T. Chen, X. Luo, and Q. Wen, "A survey on the security of blockchain systems," *Future Generation Computer Systems*, vol. 107, pp. 841-853, 2020.

[4] N. S. bt Abd Halim, M. A. Rahman, S. Azad, and M. N. Kabir, "Blockchain security hole: issues and solutions," in *International Conference of Reliable Information and Communication Technology*, 2017: Springer, pp. 739-746.

[5] J. Bonneau, A. Miller, J. Clark, A. Narayanan, J. A. Kroll, and E. W. Felten, "Sok: Research perspectives and challenges for bitcoin and cryptocurrencies," in 2015 IEEE symposium on security and privacy, 2015: IEEE, pp. 104-121.

[6] Z. Zheng, S. Xie, H.-N. Dai, X. Chen, and H. Wang, "Blockchain challenges and opportunities: A survey," International Journal of Web and Grid Services, vol. 14, no. 4, pp. 352-375, 2018.

[7] D. Genkin, D. Papadopoulos, and C. Papamanthou, "Privacy in decentralized cryptocurrencies," Communications of the ACM, vol. 61, no. 6, pp. 78-88, 2018.

[8] S. Meiklejohn and C. Orlandi, "Privacy-enhancing overlays in bitcoin," in International Conference on Financial Cryptography and Data Security, 2015: Springer, pp. 127-141.

[9] J. Yli-Huumo, D. Ko, S. Choi, S. Park, and K. Smolander, "Where is current research on blockchain technology?—a systematic review," PloS one, vol. 11, no. 10, p. e0163477, 2016.

[10] M. Conti, E. S. Kumar, C. Lal, and S. Ruj, "A survey on security and privacy issues of bitcoin," IEEE Communications Surveys & Tutorials, vol. 20, no. 4, pp. 3416-3452, 2018.

[11] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," Decentralized Business Review, p. 21260, 2008.

[12] R. Zhang, R. Xue, and L. Liu, "Security and privacy on blockchain," ACM Computing Surveys (CSUR), vol. 52, no. 3, pp. 1-34, 2019.

[13] Q. Feng, D. He, S. Zeadally, M. K. Khan, and N. Kumar, "A survey on privacy protection in blockchain system," Journal of Network and Computer Applications, vol. 126, pp. 45-58, 2019.

[14] A. Ghosh, S. Gupta, A. Dua, and N. Kumar, "Security of Cryptocurrencies in blockchain technology: State-of-art, challenges and future prospects," Journal of Network and Computer Applications, vol. 163, p. 102635, 2020.

[15] E. Zamani, Y. He, and M. Phillips, "On the security risks of the blockchain," Journal of Computer Information Systems, vol. 60, no. 6, pp. 495-506, 2020.

[16] D. E. O'Leary, "Configuring blockchain architectures for transaction information in blockchain consortiums: The case of accounting and supply chain systems," Intelligent Systems in Accounting, Finance and Management, vol. 24, no. 4, pp. 138-147, 2017.

[17] M. Swan, Blockchain: Blueprint for a new economy. " O'Reilly Media, Inc.", 2015.

[18] J. L. Zhao, S. Fan, and J. Yan, "Overview of business innovations and research opportunities in blockchain and introduction to the special issue," ed: SpringerOpen, 2016.

[19] K. Hameed, M. Barika, S. Garg, M. B. Amin, and B. Kang, "A Taxonomy Study on Securing Blockchain-based Industrial Applications: An Overview, Application Perspectives, Requirements, Attacks, Countermeasures, and Open Issues," arXiv preprint arXiv:2105.11665, 2021.

[20] F. Jameel, U. Javaid, W. U. Khan, M. N. Aman, H. Pervaiz, and R. Jäntti, "Reinforcement learning in blockchain-enabled IIoT networks: A survey of recent advances and open challenges," Sustainability, vol. 12, no. 12, p. 5161, 2020.

[21] D. Efanov and P. Roschin, "The all-pervasiveness of the blockchain technology," Procedia computer science, vol. 123, pp. 116-121, 2018.

[22] 2019. [Online]. Available: https://consensus.relictum.pro.

[23] S. Zhai, Y. Yang, J. Li, C. Qiu, and J. Zhao, "Research on the Application of Cryptography on the Blockchain," in Journal of Physics: Conference Series, 2019, vol. 1168, no. 3: IOP Publishing, p. 032077.

[24] V. Acharya, A. E. Yerrapati, and N. Prakash, Oracle Blockchain Quick Start Guide: A practical approach to implementing blockchain in your enterprise. Packt Publishing Ltd, 2019.

[25] D. Berdik, S. Otoum, N. Schmidt, D. Porter, and Y. Jararweh, "A survey on blockchain for information systems management and security," Information Processing & Management, vol. 58, no. 1, p. 102397, 2021.

[26] A. Bryk, "Blockchain attack vectors: vulnerabilities of the most secure technology," Accessed: Sep, vol. 14, p. 2019, 2018.

[27] S. Dhar, "What Is Proof-of-Activity (PoA)?," 2019. [Online]. Available: https://theblockchaincafe.com/what-is-proof-of-activity-poa/.

[28] G. G. Gueta et al., "Sbft: a scalable and decentralized trust infrastructure," in 2019 49th Annual IEEE/IFIP international conference on dependable systems and networks (DSN), 2019: IEEE, pp. 568-580.

[29] X. Fan and Q. Chai, "Roll-DPoS: a randomized delegated proof of stake scheme for scalable blockchain-based internet of things systems," in Proceedings of the 15th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services, 2018, pp. 482-484.

[30] F. M. Benčić and I. P. Žarko, "Distributed ledger technology: Blockchain compared to directed acyclic graph," in 2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS), 2018: IEEE, pp. 1569-1570.

[31] Naveen Joshi., "Everything you need to know about blockchain 3.0," 2021. [Online]. Available: https://www.bbntimes.com/technology/everything-you-need-to-know-about-blockchain-3-0.

[32] A. Begum, A. Tareq, M. Sultana, M. Sohel, T. Rahman, and A. Sarwar, "Blockchain attacks analysis and a model to solve double spending attack," International Journal of Machine Learning and Computing, vol. 10, no. 2, pp. 352-357, 2020.

[33] G. Morganti, E. Schiavone, and A. Bondavalli, "Risk Assessment of Blockchain Technology," in 2018 Eighth Latin-American Symposium on Dependable Computing (LADC), 2018: IEEE, pp. 87-96.

[34] W. Li, S. Andreina, J.-M. Bohli, and G. Karame, "Securing proof-of-stake blockchain protocols," in Data Privacy Management, Cryptocurrencies and Blockchain Technology: Springer, 2017, pp. 297-315.

[35] L. Bahack, "Theoretical bitcoin attacks with less than half of the computational power (draft)," arXiv preprint arXiv:1312.7013, 2013.

[36] N. T. Courtois and L. Bahack, "On subversive miner strategies and block withholding attack in bitcoin digital currency," arXiv preprint arXiv:1402.1718, 2014.

[37] E. Deirmentzoglou, G. Papakyriakopoulos, and C. Patsakis, "A survey on long-range attacks for proof of stake protocols," IEEE Access, vol. 7, pp. 28712-28725, 2019.

[38] M. Apostolaki, A. Zohar, and L. Vanbever, "Hijacking bitcoin: Routing attacks on cryptocurrencies," in 2017 IEEE Symposium on Security and Privacy (SP), 2017: IEEE, pp. 375-392.

[39] L. Parker, "Bitcoin 'spam attack'stressed network for at least 18 months, claims software developer," Brave Newcoin, 2017.

[40] K. Nakayama, Y. Moriyama, and C. Oshima, "An Algorithm that Prevents SPAM Attacks using Blockchain," International Journal of Advanced Computer Science and Applications, vol. 9, no. 7, pp. 204-208, 2018.

[41] G. O. Karame, E. Androulaki, and S. Capkun, "Double-spending fast payments in bitcoin," in Proceedings of the 2012 ACM conference on Computer and communications security, 2012, pp. 906-917.

[42] G. O. Karame, E. Androulaki, M. Roeschlin, A. Gervais, and S. Čapkun, "Misbehavior in bitcoin: A study of double-spending and accountability," ACM Transactions on Information and System Security (TISSEC), vol. 18, no. 1, pp. 1-32, 2015.

[43] T. Bamert, C. Decker, L. Elsen, R. Wattenhofer, and S. Welten, "Have a snack, pay with Bitcoins," in IEEE P2P 2013 Proceedings, 2013: IEEE, pp. 1-5.

[44] K. Krombholz, A. Judmayer, M. Gusenbauer, and E. Weippl, "The other side of the coin: User experiences with bitcoin security and privacy," in International conference on financial cryptography and data security, 2016: Springer, pp. 555-580.

[45] G. Karame, E. Androulaki, and S. Capkun, "Two Bitcoins at the Price of One? Double-Spending Attacks on Fast Payments in Bitcoin," IACR Cryptol. ePrint Arch., vol. 2012, no. 248, pp. 1-17, 2012.

[46] S. Solat and M. Potop-Butucaru, "Zeroblock: Preventing selfish mining in bitcoin," arXiv preprint arXiv:1605.02435, 2016.

[47] 2014. [Online]. Available: https://bitslog.com/2014/05/07/decor-2/.

[48] A. Gkaniatsou, M. Arapinis, and A. Kiayias, "Low-level attacks in bitcoin wallets," in International Conference on Information Security, 2017: Springer, pp. 233-253.

[49] P. Daian, I. Eyal, A. Juels, and E. G. Sirer, "(Short Paper) piecework: Generalized outsourcing control for proofs of work," in International Conference on Financial Cryptography and Data Security, 2017: Springer, pp. 182-190.

[50] T. Duong, A. Chepurnoy, L. Fan, and H.-S. Zhou, "Twinscoin: A cryptocurrency via proof-of-work and proof-of-stake," in Proceedings of the 2nd ACM Workshop on Blockchains, Cryptocurrencies, and Contracts, 2018, pp. 1-13.

[51] Ittay Eyal and Emin Gün Sirer, 2014. [Online]. Available: https://hackingdistributed.com/2014/06/18/how-to-disincentivize-large-bitcoin-mining-pools/.

[52] Bastiaan, Martijn., 2015. [Online]. Available: https://www.semanticscholar.org/paper/Preventing-the-51-Attack%3A-a-Stochastic-Analysis-of-Bastiaan/03366d1fda3b24651c71ec6ce21bb88f34872e40.

[53] E. Heilman, A. Kendler, A. Zohar, and S. Goldberg, "Eclipse attacks on bitcoin’s peer-to-peer network," in 24th {USENIX} Security Symposium ({USENIX} Security 15), 2015, pp. 129-144.

[54] "Timejacking and bitcoin," 2011. [Online]. Available: http://culubas.blogspot.com/2011/05/timejacking-bitcoin_802.html.

[55] "Bip 62: Dealing with malleability," 2014. [Online]. Available: https://github.com/bitcoin/bips/blob/master/bip-0062.mediawiki.

[56] M. Andrychowicz, S. Dziembowski, D. Malinowski, and Ł. Mazurek, "On the malleability of bitcoin transactions," in International Conference on Financial Cryptography and Data Security, 2015: Springer, pp. 1-18.

[57] "Transaction malleability in cryptocurrencies," 2016. [Online]. Available: https://iohk.io/blog/research/transaction-malleability-in-cryptocurrencies/.

[58] P. McCorry, S. F. Shahandashti, and F. Hao, "Refund attacks on Bitcoin’s payment protocol," in International Conference on Financial Cryptography and Data Security, 2016: Springer, pp. 581-599.

[59] E.-R. Latifa and A. Omar, "Blockchain: Bitcoin wallet cryptography security, challenges and countermeasures," Journal of Internet Banking and Commerce, vol. 22, no. 3, pp. 1-29, 2017.

[60] A. Gervais, H. Ritzdorf, G. O. Karame, and S. Capkun, "Tampering with the delivery of blocks and transactions in bitcoin," in Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, 2015, pp. 692-705.

[61] "Japanese Cryptocurrency Monacoin Hit by Selfish Mining Attack," 2018. [Online]. Available: https://www.ccn.com/japanese-cryptocurrency-monacoin-hit-by-selfish-mining-attack/.

[62] "IETF RFC 5905-2010 - Network Time Protocol Version 4: Protocol and Algorithms Specification," [Online]. Available: https://joinup.ec.europa.eu/collection/ict-standards-procurement/solution/ietf-rfc-5905-2010-network-time-protocol-version-4-protocol-and-algorithms-specification/distribution/ietf-rfc-5905-2010-network-time-protocol-version-4-protocol-and-algorithms-specification.

[63] G. Danezis, R. Dingledine, and N. Mathewson, "Mixminion: Design of a type III anonymous remailer protocol," in 2003 Symposium on Security and Privacy, 2003., 2003: IEEE, pp. 2-15.

[64] G. Maxwell,, "Coinjoin: Bitcoin privacy for the real world," 2013. [Online]. Available: https://bitcointalk.org/index.php?topic=279249.0.

[65] T. Ruffing, P. Moreno-Sanchez, and A. Kate, "Coinshuffle: Practical decentralized coin mixing for bitcoin," in European Symposium on Research in Computer Security, 2014: Springer, pp. 345-364.

[66] T. Bamert, C. Decker, R. Wattenhofer, and S. Welten, "Bluewallet: The secure bitcoin wallet," in International Workshop on Security and Trust Management, 2014: Springer, pp. 65-80.

[67] R. Gennaro, S. Goldfeder, and A. Narayanan, "Threshold-optimal DSA/ECDSA signatures and an application to Bitcoin wallet security," in International Conference on Applied Cryptography and Network Security, 2016: Springer, pp. 156-174.

[68] S. Jarecki, A. Kiayias, H. Krawczyk, and J. Xu, "Highly-efficient and composable password-protected secret sharing (or: How to protect your bitcoin wallet online)," in 2016 IEEE European Symposium on Security and Privacy (EuroS&P), 2016: IEEE, pp. 276-291.

[69] M. Gentilal, P. Martins, and L. Sousa, "TrustZone-backed bitcoin wallet," in Proceedings of the Fourth Workshop on Cryptography and Security in Computing Systems, 2017, pp. 25-28.

[70] "Coincentral," 2018. [Online]. Available: https://coincentral.com/sybil-attack-blockchain/.

[71] G. Maxwell, "CoinSwap: Transaction graph disjoint trustless trading," CoinSwap: Transactiongraphdisjointtrustlesstrading (October 2013), 2013.

[72] J. Bonneau, A. Narayanan, A. Miller, J. Clark, J. A. Kroll, and E. W. Felten, "Mixcoin: Anonymity for bitcoin with accountable mixes," in International Conference on Financial Cryptography and Data Security, 2014: Springer, pp. 486-504.

[73] L. Valenta and B. Rowan, "Blindcoin: Blinded, accountable mixes for bitcoin," in International Conference on Financial Cryptography and Data Security, 2015: Springer, pp. 112-126.

[74] E. Heilman, L. Alshenibr, F. Baldimtsi, A. Scafuro, and S. Goldberg, "Tumblebit: An untrusted bitcoin-compatible anonymous payment hub," in Network and Distributed System Security Symposium, 2017.

[75] "PlatOn," 2018. [Online]. Available: https://www.platon.network/en.

[76] N. Van Saberhagen, "CryptoNote v 2.0," ed, 2013.

[77] S. Noether and A. Mackenzie, "Ring confidential transactions," Ledger, vol. 1, pp. 1-18, 2016.

[78] "Monero project," [Online]. Available: https://getmonero.org/.

[79] T. P. Pedersen, "Non-interactive and information-theoretic secure verifiable secret sharing," in Annual international cryptology conference, 1991: Springer, pp. 129-140.

[80] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in International conference on the theory and applications of cryptographic techniques, 1999: Springer, pp. 223-238.

[81] G. Zyskind, O. Nathan, and A. Pentland, "Enigma: Decentralized computation platform with guaranteed privacy," arXiv preprint arXiv:1506.03471, 2015.

[82] I. Miers, C. Garman, M. Green, and A. D. Rubin, "Zerocoin: Anonymous distributed e-cash from bitcoin," in 2013 IEEE Symposium on Security and Privacy, 2013: IEEE, pp. 397-411.

[83] 2014. [Online]. Available: http://zerocash-project.org.

[84] G. Maxwell, "Confidential transactions," 2015. [Online]. Available: https://elementsproject.org/features/confidential-transactions.

[85] Y. Zhao, J. Zhao, L. Jiang, R. Tan, and D. Niyato, "Mobile edge computing, blockchain and reputation-based crowdsourcing iot federated learning: A secure, decentralized and privacy-preserving system," 2020.

[86] "Cryptography in Blockchain: Types & Applications," 2021. [Online]. Available: https://www.upgrad.com/blog/cryptography-in-blockchain/.

[87] A. Shamir, "How to share a secret," Communications of the ACM, vol. 22, no. 11, pp. 612-613, 1979.

[88] G. R. Blakley, "Safeguarding cryptographic keys," in Managing Requirements Knowledge, International Workshop on, 1979: IEEE Computer Society, pp. 313-313.

[89] D. R. Kuhn, "A data structure for integrity protection with erasure capability," NIST Cybersecurity Whitepaper, 2018.

[90] R. Cheng et al., "Ekiden: A platform for confidentiality-preserving, trustworthy, and performant smart contracts," in 2019 IEEE European Symposium on Security and Privacy (EuroS&P), 2019: IEEE, pp. 185-200.

[91] J. Teutsch and C. Reitwießner, "Truebit: a scalable verification solution for blockchains," White Papers, 2018.

[92] H. Kalodner, S. Goldfeder, X. Chen, S. M. Weinberg, and E. W. Felten, "Arbitrum: Scalable, private smart contracts," in 27th {USENIX} Security Symposium ({USENIX} Security 18), 2018, pp. 1353-1370.

[93] S. Shakya, "Efficient security and privacy mechanism for block chain application," Journal of Information Technology, vol. 1, no. 02, pp. 58-67, 2019.

[94] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in Proceedings of the 13th ACM conference on Computer and communications security, 2006, pp. 89-98.

[95] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in 2007 IEEE symposium on security and privacy (SP'07), 2007: IEEE, pp. 321-334.