High-Confidence Computing 2 (2022) 100048

Contents lists available at ScienceDirect

High-Confidence Computing

homepage: www.elsevier.com/locate/hcc

A survey on blockchain systems: Attacks, defenses, and privacy preservation

Yourong Chen[a,d], Hao Chen[b], Yang Zhang[b], Meng Han[c,d,*], Madhuri Siddula[e], Zhipeng Cai[f,*]

[a] Zhejiang Shuren University, Hangzhou, Zhejiang, 310015, China
[b] Changzhou University, Changzhou, Jiangshu, 213164, China
[c] Binjiang Insititute of Zhejiang University, Hangzhou, Zhejiang, 310053, China
[d] Zhejiang University, Hangzhou, Zhejiang, 310058, China
[e] North Carolina A&T State University, Greensboro, NC 27411, USA
[f] Georgia State University, Atlanta, GA 30303, USA

ARTICLE INFO

Keywords:
Blockchain
System
Defense
Attack
Privacy

ABSTRACT

Owing to the incremental and diverse applications of cryptocurrencies and the continuous development of distributed system technology, blockchain has been broadly used in fintech, smart homes, public health, and intelligent transportation due to its properties of decentralization, collective maintenance, and immutability. Although the dynamism of blockchain abounds in various fields, concerns in terms of network communication interference and privacy leakage are gradually increasing. Because of the lack of reliable attack analysis systems, fully understanding some attacks on the blockchain, such as mining, network communication, smart contract, and privacy theft attacks, has remained challenging. Therefore, in this study, we examine the security and privacy of the blockchain and analyze possible solutions. We systematical classify the blockchain attack techniques into three categories, then discuss the corresponding attack and defense methods based on these categories. We focus on (1) the attack and defense methods of mining pool attacks for blockchain security issues, such as block withholding, 51%, pool hopping, selfish mining, and fork after withholding attacks, in the attack type of consensus excitation; (2) the attack and defense methods of network communication and smart contracts for blockchain security issues, such as distributed denial-of-service, Sybil, eclipse, and reentrancy attacks, in the attack type of middle protocol; and (3) the attack and defense methods of privacy thefts for blockchain privacy issues, such as identity privacy and transaction information attacks, in the attack type of application service. Finally, we discuss future research directions for blockchain security.

1. Introduction

Recently, some essential characteristics of blockchain[1], such as decentralization, collective maintenance, and immutability, were identified and have led to its explosive growth. Blockchain has been defined as the fifth disruptive innovation of the computing paradigm after the mainframe, personal computer, internet, and mobile and social network [1]. Because the nodes in the blockchain follow the same accounting transaction rules and consensus under the consensus algorithm, adopt the one-way hash algorithm, and strictly generate blocks in chronological order, blockchain has the advantages of immutability and encryption security. Therefore, blockchain is applied to digital currency [2], smart finance [3,143], smart homes [4], smart medical care [5], smart human resource [142], smart transportation [6,7], and so on. International data corporation (IDC) reported that although the blockchain market worldwide was affected by COVID-19 in 2020, the global spending on blockchain solutions was nearly $4 billion US dollars. As the economy recovers, the rate of global blockchain market spending in 2023 will usher in a strong rebound and the global spending on blockchain solutions will reach $16 billion US dollars [8].

Currently, the security and privacy issue related to blockchain remain at large and primal solutions have been proposed [9,10]. Presently, there are four major security and privacy issues in the blockchain. First, to effectively solve the problem of stable revenue for miners, blockchain helps miners cooperatively mine by creating mining pools. However, at-

* Corresponding authors.
E-mail addresses: mhan@zju.edu.cn (M. Han), zcai@gsu.edu (Z. Cai).

[1] We alternatively use “chain” and “blockchain” to represent blockchain in the remaining of this work.

https://doi.org/10.1016/j.hcc.2021.100048
Received 13 September 2021; Accepted 15 November 2021

tackers launch attacks on the mining pool to improve their revenues. For example, on May 22, 2018, hackers launched a 51% attack on the blockchain Verge, successfully stealing nearly 35 million anonymous coins [11]. On May 16, 2018, the EquiHash mining algorithm was adopted by Bitcoin gold (BTG)-supported graphic card mining in the network and hackers launched a 51% attack on the blockchain BTG by renting the computing power of the graphic card, resulting in 12,239 illegally traded gold bits [12]. On January 5, 2019, hackers launched a 51% attack on blockchain Ethereum classic (ETC) by renting the computing power of the graphic card, inducing a loss of $1.1 million US dollars [13]. Second, because the blockchain's peer-to-peer network must maintain timely communication between nodes, attackers launch network communication attacks on the blockchain, seriously affecting the network performance and considerably reduced the communication efficiency among miners. For example, on September 22, 2016, hackers launched a distributed denial-of-service (DDoS) attack on the Ethereum (ETH) blockchain, which greatly decelerated its network operation speed, resulting in two hard forks of ETH [14]. Third, although smart contracts improve the convenience and extensibility of blockchain applications, some loopholes still exist in the process. Targeting this vulnerability, attackers launch an smart contract attack and steal huge revenues. For example, on June 17, 2016, hackers maneuvered the smart contract of the decentralized autonomous organization (DAO) function to obtain illegal gains of >3 million ETH, eventually forcing ETH to hard fork [15]. On April 23, 2018, hackers attacked an integer overflow loophole in a Beauty Chain (BEC) smart contract, illegally generating 7 billion nonexistent tokens [16]. Fourth, operations on the blockchain are decentralized, traceable, tamper proof, and autonomous, among other features. However, to ensure the accuracy of block consensus, blockchain grants other users access to the trading information of the blockchain. Therefore, attackers steal the identity and transaction information of other users by analyzing their key transaction information or launch privacy theft attacks to realize other malicious purposes. For example, on March 7, 2018, hackers performed numerous malicious purchases of digital currencies by stealing users' information pertaining of Binance exchange, ultimately affecting the price of digital currencies on the network [17].

Such incessant attacks necessitate a comprehensive study on blockchain security and privacy issues. Many recent studies on blockchain attacks using game theory, artificial intelligence and other technologies have emerged and multiple attack mechanisms and defense mechanisms have been proposed. A survey [18] analyzed the risks associated with Blockchains 1.0 and 2.0. Another survey [19] analyzed blockchain safety issues in game theory. Moreover, a survey [20] discussed the security of the blockchain architecture. However, these surveys [18–20,139] did not discuss the mining pool, network communication, smart contract, and privacy theft attacks in detail. Therefore, we propose a blockchain attack classification system to analyze each attack in detail and proposed possible defense mechanisms based on the attack method. The main contributions of the study are listed below.

- We first review the workflow, classification, characteristics, and block structures of the blockchain and introduce the basic knowledge of attack and defense methods.
- Based on the characteristics of the blockchain's attack method, we propose a comprehensive blockchain attack classification system to categorize the blockchain security and privacy issues.
- On the issue of blockchain security, we detailed analyze various attacks, including block withholding, 51%, pool hopping, selfish mining, and fork after withholding (FAW) attacks, in the attack type of consensus excitation and their defense methods. Further, we focus on DDoS, Sybil, eclipse, reentrancy, and other smart contract attacks in the attack type of middle protocol and their defense methods. On the issue of blockchain privacy, we also analyze identity privacy and transaction information privacy in the attack type of application service and their defense methods.
- Furthermore, we comprehensively investigate studies on blockchain security and privacy issues and summarize the characteristics of various attack methods and defense methods as significant guiding tools for future research.
- Last but not least, we point out five promising future research directions for blockchain security and privacy, and analyze based on all the evaluations and analysis of the state-of-arts.

The remainder of the paper is organized as follows. Section 2 introduces the basic principle of blockchain and the basic knowledge of attack and defense methods. Section 3 introduces blockchain attack classification system to categorize theblockchain security and privacy issues. Section 4 focuses on the mining attack and defense methods of the mining pool in the attack type of consensus excitation. Section 5 focuses on the attack and defense methods of network communication and smart contracts in the attack type of middle protocol. Section 6 focuses on the attack and defense methods of privacy thefts in the attack type of application service. Section 7 discusses future research directions for blockchain security. Finally, Section 8 presents the conclusion.

2. Overview and preliminary preparation of blockchain

2.1. Overview of blockchain

The development of blockchain occurred in three stages: Blockchains 1.0, 2.0, and 3.0 [21,22].

Blockchain 1.0: The blockchain is mainly applied to digital currency. Bitcoin was proposed in 2008 by S. Nakamoto in a secret cryptography discussion group [23]. He described Bitcoin as the most typical representative of Blockchain 1.0. Bitcoin uses the proof-of-work consensus algorithm to solve double-spending and byzantine problems in digital currency, thereby realizing a decentralized transaction payment function for digital currencies. Moreover, the development of Bitcoin has led to the production of "fake" digital money, including Dogecoin and Litecoin.

Blockchain 2.0: The digital currency, such as Bitcoin, satisfies only the demand of the virtual currency, and the shortage of consumed resources is vital. Therefore, the users prefer to obtain financial transaction functions by performing smart contracts on the blockchain, thereby considerably expanding the application scope of the blockchain. As a blockchain foundation development platform, ETH provides a reliable smart contract programming environment, allowing users to write suitable and intelligent contracts based on their own needs and application scenarios, such as equity-crowdfunding voting and security trading and issuance.

Blockchain 3.0: With the rapid development of the blockchain, the blockchain confirms the property right of the information representing the value on the Internet. Moreover, the blockchain measures and stores the information representing the value on the Internet; hence, the blockchain can track and control assets while trading them. Generally, blockchain is not limited to money, economy, and markets; it has expanded to other areas in demand, such as health, identity certification, logistics, and voting. Currently, the scope of blockchain application is on the whole social level.

2.1.1. Blockchain characteristics and block structure

A blockchain is an integrated innovation that combines computer technologies, such as distributed data storage, networking, consensus algorithms, and encryption algorithms [24,25,141]. It is a type of chain data structure combined in chronological order, ensuring the stability of distributed ledger using cryptography. The properties of a blockchain are below.

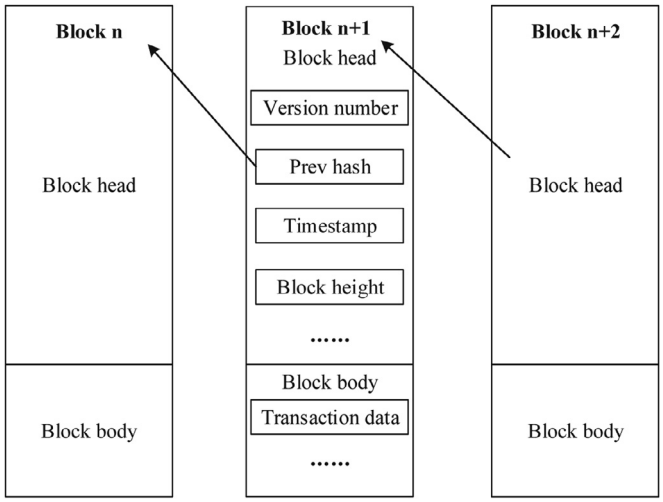

Fig. 1. Block structure.

Decentralization: Compared with the centralized data management of traditional applications, the blockchain allows multiple nodes to charge accounting and uses the consensus algorithm to ensure consistency among the nodes. It prevents the intrusion of a data center and the involvement of third-party credit agencies. Furthermore, it considerably reduces resource wastage caused by the credibility of a transaction.

Tamper proof: When the transaction information is validated and added to the blockchain, attacking a single node is worthless unless the attacker controls >51% of the node number. Thus, the blockchain effectively ensures the reliability of data.

Traceability: Blockchain inserts a transaction consensus result at the current moment into the block when constructing a new block and associates it with the hash value of the previous block to form a blockchain data structure. Therefore, when analyzing a certain state, we quickly trace all relevant historical transaction information from the block time.

Autonomy: Blockchain enables all nodes in the entire network to exchange, record, and update data in a trustworthy environment using consensus-based specifications and protocols. Therefore, it ensures the accuracy and authenticity of every transaction recorded on the blockchain without human intervention.

Because a block structure is used for data representation, it is instrumental in the blockchain architecture. The relevant concepts are listed below.

- Block structure [26]: As shown in Fig. 1, each block has two parts: a blockhead and a block body. The blockhead contains the version number, the hash value of the previous block, timestamp, and block height. The block body records the transaction data for the block.
- Hash algorithm [27]: It is a cryptographic algorithm that converts the original information of any length into a fixed-length hash value. It produces the same hash value when the original information is the same. Because the attacker cannot determine the correspondence between the original information and the hash value, obtaining the original information based on the hash value becomes difficult.
- Merkle tree [28]: The Merkle tree generates a digital fingerprint of the transaction set to detect and position transactions rapidly. It considers the hash value of each transaction as a leaf node and recursively rehashes two adjacent hash values until only one hash value remains, i.e., the root of the Merkle tree.

2.1.2. Workflow of blockchain

The development of blockchain technology and its use in Bitcoin has increased its applications in various fields. For an in-depth understanding, we introduce the general workflow of a blockchain. Note that this process may not cover all working contents of a blockchain; however, it reflects the core idea of most of the working processes of a blockchain.

As shown in Fig. 2, a user initiates a transaction using their private key and other information as a client. Then, they broadcast the transaction information to other nodes in the network. The nodes in the network construct a candidate block by combining all received transactions using the hash value of the previous block, version number, block height, and other information. To solve the attributions of the accounting rights of the candidate blocks, each node must implement a consensus algorithm. For example, the proof-of-work consensus algorithm uses block mining to compete for the accounting rights of the current block, i.e., to determine a random number to ensure that the hash results of the candidate blocks meet specific conditions. Second, the node that wins the accounting rights sends its verified candidate block to other nodes. The other nodes conduct a series of verification operations on the block, including confirming the validity of transaction status and duplicated spending. After successful block verification, the node adds the block to the blockchain, which it maintains. When most nodes add the block to the blockchain, which they maintain, the block and its internal transactions are officially introduced into the chain.

2.1.3. Blockchain classification

According to the participation mode of a node, the blockchain is divided into three types, namely, public, alliance, and private blockchains [29,30,140]. Data in a public blockchain network are entirely open and transparent. Any node in this chain can freely join or quit the network to participate in maintaining and reading of blockchain data. The public blockchain guarantees the security and privacy of the blockchain data using an encryption algorithm that employs cryptography and realizes the consensus of the nodes of the entire network using a consensus algorithm such as the proof-of-work consensus algorithm. Therefore, it is completely decentralized. The current typical examples of the public chain are Bitcoin by S. Nakamoto and ETH of the Ethereum Foundation. The user can use functions such as wallet address creation, transfer transaction, and mining only by downloading the client of the public blockchain. The alliance blockchain is a multicenter or partially decentralized sub-blockchain. The alliance blockchain only allows alliance nodes to participate in block access. The block consensus process must be completed by the alliance nodes within the limited scope. Therefore, compared with the public blockchain, the alliance blockchain can achieve better security and privacy protection by adopting a practical byzantine fault tolerance consensus algorithm to improve the consensus efficiency. Examples of the alliance blockchain mainly include Hyperledger Fabric of the Linux Foundation and Corrda of the R3 blockchain alliance. In a private blockchain, the writing rights of the blockchain are controlled by a single institution, while the reading rights are adjusted externally based on the actual situation. Therefore, the private blockchain is mainly used in supply chain management, accounting audits, bill replacement, and other application scenarios. Compared with the public and alliance blockchains, the private blockchain fully utilizes the information protection mechanism within the organization, provides a traceable and immutable platform, and finally prevents internal and external attacks. A typical example of a private blockchain is the MultiChain platform launched by Coin Sciences Ltd, which mainly deploys the private chain environment for enterprise privacy protection and permission control.

Based on the authority of nodes in the consensus process, a blockchain is divided into permissioned and permissionless blockchains. In a permissioned blockchain, every node participating in the blockchain must be licensed. Because the characteristics of the alliance blockchain require new nodes to meet certain conditions, it corresponds to the permissioned blockchain and is often applied in the payment settlement of banks and the data sharing of a government. Similar to the alliance blockchain, the private blockchain sets certain conditions for new nodes to join, making the private blockchain a permissioned blockchain. A permissionless blockchain allows nodes to participate in the blockchain

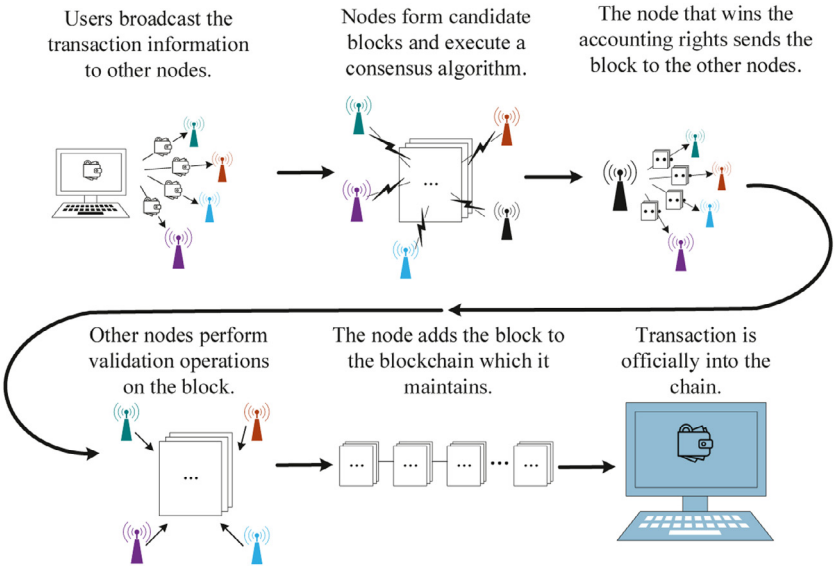

Fig. 2. Workflow of a blockchain.

without a license. The public blockchain allows the nodes in a network to freely participate in blockchain data reading/writing processes, transaction execution, and block consensus. Therefore, it is a permissionless blockchain.

2.2. Preliminary preparation

2.2.1. Basics of game theory

In game theory, the rational strategies of players in the game and the equilibrium results of those strategies are studied. Each player must change their confrontation strategy according to those of other players to win [31,32]. Thus, game theory analyzes the attack strategy of an attacker against the blockchain. The basic concepts are described below.

- Player: A player selects a strategy in the game and obtains revenue. In the blockchain, a player can be the mining pool, miner, or attacker.
- Strategy: A player uses a complete action plan to achieve their expectations. In the blockchain, a strategy can be the allocation of the computing power of the mining pool and the choice of the attack method.
- Revenue: This refers to the gains and losses of each player when a round is over. In the blockchain, revenue can be the mining revenue.
- Information: A player knowes the strategy of all players before selecting a strategy.

Suppose there are N mining pools in a blockchain, and each mining pool acts as a player in the game. Further, each mining pool selects an attack strategy s_i from its strategy space S_i to improve its revenue. After all the mining pools have determined their respective strategies, we obtain the strategy combination of the game $s = (s_1, s_2, s_3, \dots s_N)$ and use the revenue of each mining pool as the result of the game. Each mining pool must select the optimal strategy that maximizes its revenue $e_i(s), i \in N$. If there is a set of strategy combinations $s' = (s'_1, s'_2, s'_3, \dots s'_N)$, the strategy adopted by one player is the optimal strategy adopted by all other players under the strategy combinations. In other words, when other mining pools do not change their strategies, no mining pool changes their strategies, resulting in a Nash equilibrium combination strategy.

2.2.2. Basics of cryptography

Cryptography involves encoding data to ensure communication secrecy and decoding data to decipher and obtain the information. Blockchain can use symmetric and asymmetric encryption algorithms to ensure the security and privacy of data sharing [33,34]. The basic concepts are described below.

- Key: Each symmetric encryption algorithm includes encryption and decryption keys. An asymmetric encryption algorithm includes public and private keys.
- Plaintext: These data directly represent the true meaning.
- Ciphertext: These data are encrypted to hide the true meaning.
- Encryption: Encryption involves the use of an encryption key and an encryption algorithm to convert the plaintext into the ciphertext.
- Decryption: Decryption involves the use of a decryption key and an encryption algorithm to convert the ciphertext into the plaintext.

Currently, the major encryption algorithms in blockchain technology include symmetric and asymmetric encryption algorithms. In a symmetric encryption algorithm, communication from both parties (sender/receiver) uses the same key to encrypt and decrypt large amounts of data and is transmitted over the network. Some algorithms in this category include advanced encryption standards and data encryption standards. However, because encryption algorithms use the same private key, for algorithm security, the encryption algorithms themselves and the security of key management must be considered. In an asymmetric encryption algorithm, the sender first uses a public key to encrypt the data. Then, the receiver decrypts the ciphertext using the corresponding private key. Some algorithms in this category include elliptic curve cryptography and the digital signature algorithm. Although asymmetric encryption algorithms effectively prevent users from exchanging keys using the public-private key encryption method, the algorithm implementation process is more complex, making their encryption and decryption speeds incomparable with symmetric encryption algorithms.

2.2.3. Basics of reinforcement learning

Reinforcement learning involves maximizing revenue or solving specific goals by constantly learning optimal strategies to interact with an environment. Therefore, an attacker uses reinforcement learning to optimize the attack strategy against the blockchain [35,36]. The basic concepts are described below.

- Agent: An agent refers to a decision-maker who selects an action.
- Environment: This involves everything that interacts with the agent.
- Action: An action represents the behavior of the agent.
- State: This refers to the information acquired by the agent from the environment.

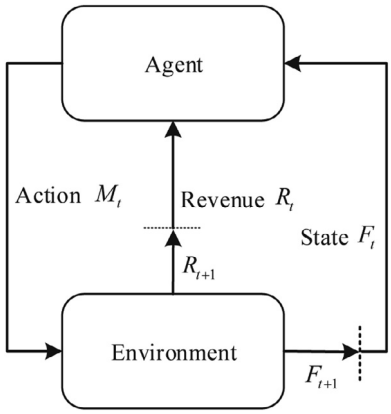

Fig. 3. Basic structure of reinforcement learning.

- Revenue: Revenue involves feedback from the environment corresponding to the actions of the agent.
- Strategy: This refers to a function of the agent for calculating the next action based on the state.
- Transition probability: The probability that the agent will proceed to the next state after performing the action refers to a transition probability.

As shown in Fig. 3, the agent obtains its current state F_t and revenue information R_t from the environment at time t and determines the action M_t to be performed at time $t+1$ using a strategy. The environment calculates the revenue information R_{t+1} of the agent according to the action M_t and drives the agent to obtain a new state F_{t+1} using the transition probability. Therefore, in the subsequent process, the agent adjusts its strategy based on the revenue information to determine the optimal strategy that achieves the largest long-term revenue.

3. Blockchain attack classification system

As shown in Fig. 4, we category the blockchain security and privacy issues into three types: the attack of consensus excitation, the middle protocol, and the application service. On the issue of blockchain security, the consensus excitation attack refers to the attacker obtain additional revenue by interfering with the block consensus result of the blockchain. And the middle protocol attack refers to the behavior of attackers launching attacks on node communication and smart contracts. On the issue of blockchain privacy, the application service attack refers to the behavior of an attacker targeting user privacy in an application scenario. In this structure, we divide blockchain attack and blockchain defense methods into the attack and defense methods of the mining pool, those of network communication and smart contracts, and those of privacy thefts. In the consensus excitation attack, the attack methods of the mining pool include block withholding, 51%, pool hopping, selfish mining, and FAW attacks. In the attack of middle protocol, the attack methods of network communication include DDoS, eclipse, and Sybil attacks. The attack methods of smart contracts include reentrancy attacks. In the application service attack, privacy theft attacks include identity privacy and transaction information attacks. The defense methods include the Mixcoin protocol, zero-knowledge proof, and ring signature.

4. Mining attack and defense methods of mining pool

4.1. Block withholding attack and defense methods

4.1.1. Block withholding attack methods

As shown in Fig. 5, M. Rosenfeld first proposed the block withholding attack [37] and stated that an attacker always sends the partial proof of work to the pool manager and discards the complete proof of work. In terms of attack conditions, a block withholding attack only requires the attacker to reasonably allocate the attack computing power. For attack protection, because the miner who launches block withholding attack always submits the partial proof of work, the pool manager only discovers that it has encountered a block withholding attack but cannot identify the malicious miners in the mining pool. Because the block withholding attack cannot provide any effective help to the attacked mining pool, it even helps the attacker get part of the revenue in the attacked mining pool. Therefore, this attack can cause serious harm to the mining process. Many studies have discussed some block withholding attack methods, including a block withholding attack in dual mining pools, block withholding attack among multiple mining pools, and hybrid block withholding attacks, as shown in Table 1.

Summary of block withholding attack methods

As shown in Fig. 6, a block withholding attack in dual mining pools refers to two mining pools launching a block withholding attack on each other. Because the mining pool distributes revenue based on the proof of work of each miner, the mining pool increases its revenue by launching block withholding attacks on the other mining pool. However, the block withholding attack in dual mining pools can easily lead to the mining dilemma. In other words, both pools cannot achieve better revenues. Many studies have focused on simultaneously improving the revenue in two pools [38–43]. We divide the relevant research into two aspects: model building and zero determinant (ZD) strategy.

In terms of model building, a study [38] considered that the network could perform a reward and punishment mechanism for the attack behavior of mining pools. In other words, the mining pool that did not launch block withholding attacks was rewarded a. However, the mining pool that launched block withholding attacks was punished via revenue deduction ka. Therefore, the study [38] established the revenue matrix of two mining pools (formula 1) and analyzed the Nash equilibrium of pure (the mining pool only selects a specific strategy) and mixed (the mining pool randomly selects a strategy with a certain probability) strategies.

$$P_1 = \begin{array}{c} \\ C \\ A \end{array} \begin{array}{c} \begin{array}{cc} P_2{=}C & \qquad A \end{array} \\ \begin{pmatrix} p_1, p_2 & p_1 + a - d, p_2 - ka + d \\ p_1 - ka + d, p_2 + a - d & p_1 - ka - d', p_2 - ka + d' \end{pmatrix} \end{array}, \tag{1}$$

where P_1 and P_2 represent two mining pools during block mining, A represents a mining pool use the total computing power to launch the block withholding attack, C represents a mining pool use the total computing power used to launch honest mining, p_1 and p_2 represent the revenue gained by mining pools P_1 and P_2, respectively, without implementing the punishment and reward mechanism, a represents the reward gained by a mining pool that does not launch an attack on the other mining pool, k represents the penalty factor of the reward and punishment mechanism, d represents the gain achieved by the attacking mining pool and lost by the attacked mining pool when the former launches an attack and the latter is honestly mining, and d' represents the revenue gained by mining pool P_2 and lost by mining pool P_1 when the mining pools opt to attack each other and the revenue obtained by mining pool P_2 is greater than that obtained by mining pool P_1. Based on these conditions, considering the permeability of the mining pool (the proportion of malicious miners assigned by the attacking mining pool against the target mining pool) and the betrayal rate (the proportion of malicious miners helping the attacked mining pool to identify the complete proof of work), the study [38] established a revenue model for each mining pool to analyze the Nash equilibrium and conditions under the permeability assumption. A study [39] assumed that two mining pools freely opted to cooperate or launch block withholding attacks. Moreover, considering that the mining pool consumes resources, such as water and electricity when performing the proof of work, the cost of both honest mining and attack was considered between the mining pools based on the computing power to obtain the revenue matrix of two mining pools (formula 2). Given the above revenue matrix, the study [39] analyzed

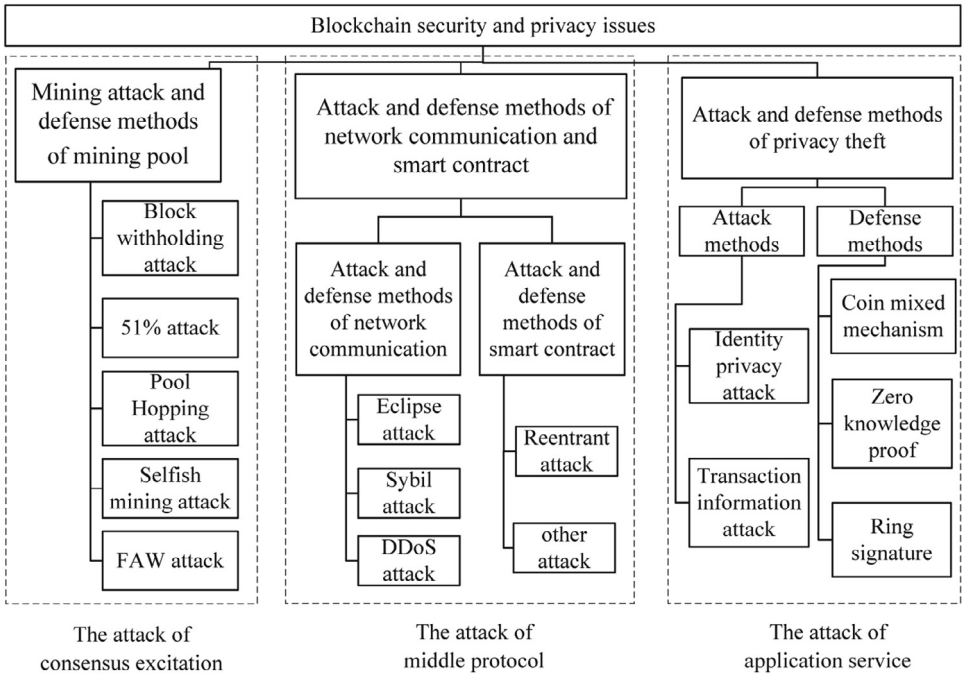

Fig. 4. Blockchain attack classification system.

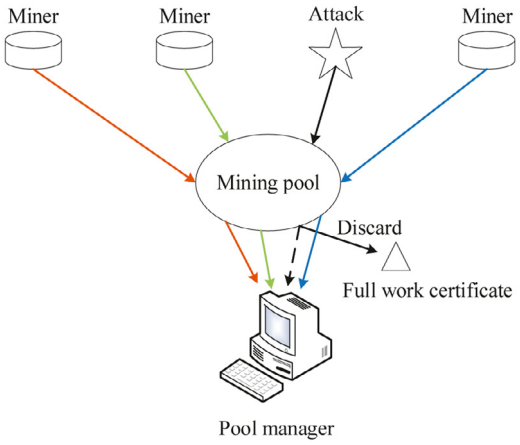

Fig. 5. Schematic of a block withholding attack.

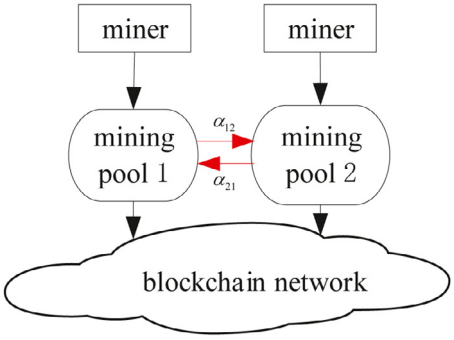

Fig. 6. Diagram of a block withholding attack in dual mining pools.

the Nash equilibrium under the pure and mixed strategies.

$$P_1 = \begin{matrix} \\ C \\ A \end{matrix} \begin{matrix} P_2 = \\ C \\ \begin{pmatrix} f(a_1+a_2)\frac{a_1}{a_1+a_2} - C_h(a_1), \\ f(a_1+a_2)\frac{a_2}{a_1+a_2} - C_h(a_2) \end{pmatrix} \\ A \\ \begin{pmatrix} f(a_2)\frac{a_2}{a_1+a_2} - C_p(a_1), \\ f(a_2)\frac{a_2}{a_1+a_2} - C_h(a_2) \end{pmatrix} \end{matrix} \begin{matrix} \\ A \\ \begin{pmatrix} f(a_1)\frac{a_1}{a_1+a_2} - C_h(a_1), \\ f(a_1)\frac{a_1}{a_1+a_2} - C_p(a_2) \end{pmatrix} \\ \begin{pmatrix} -C_p(a_1), \\ -C_p(a_2) \end{pmatrix} \end{matrix}, \tag{2}$$

where C_p represents the cost per unit of computing power for a block withholding attack, C_h represents the cost per unit of computing power for honest mining, a_1 and a_2 represent the computing power of mining pools P_1 and P_2, respectively, and $f()$ represents the total revenue of a mining pool under the current computing power. In another study [40], only P_1 and P_2 were assumed to exist in the network; the effective total computing power of the two mining pools were c_1 and c_2, respectively. P_1 assigned computing power $y_{1,2}$ for joining P_2 to launch a block withholding attack and used the rest of the computing power $c_1 - y_{1,2}$ for honest mining. However, P_2 could not launch a block withholding attack on P_1; it only performed honest mining. Considering these assumptions, a study [40] proposed an optimization model (formula (3)) to obtain the optimal computing power and execution conditions for the block withholding attack assigned by P_1 against P_2.

$$\begin{cases} \max\left(1 - \frac{c_2^2 q_2}{(c_2+y_{1,2}(1-q_1))((c-y_{1,2})q_1+c_2q_2)}\right) \\ \text{s.t.} \quad 0 \le y_{1,2} \le c_1 \end{cases}, \tag{3}$$

where q_1 and q_2 represent the probability that P_1 and P_2, respectively, determine the complete proof of work. Considering that two mining pools can have an unlimited number of game processes, a study [41] introduced a repeated prisoner's dilemma model in the modeling process. In this model, each player could punish the noncooperative behavior of the other player in the previous round to determine a strategy that can achieve the Nash equilibrium.

As an emerging method in game theory, the ZD strategy optimizes the prisoner's dilemma model by solving the low-system revenue problem and forces a linear relationship between the opponent and its own, irrespective of the strategy adopted by the opponent. Therefore, some studies have adopted the ZD strategy for studying block withholding

Table 1
Summary of block withholding attack methods.

Reference	Mechanism	Type	Solution	Purpose	Application
[38,39]	Complete information game	Game theory	Pure strategy and mixed strategy	Achieve Nash equilibrium	Block withholding attack in dual mining pools
[40]	Revenue optimization model of attack mining pool methods	Optimization theory	Formula derivation	Maximize revenue of attacking the mining pool	
[41]	Iterative prisoner's dilemma game	Game theory	Valuation strategy	Achieve Nash equilibrium	
[42]	Complete game information		ZD strategy		
[43]				Maximize the revenue of attacking the mining pool	
[44]	Stochastic game	Reinforcement learning	Tile coding method	Analyze the impact of miner migration on computing power	Block withholding attack among multiple mining pools
[45]	Evolutionary game	Game theory	Replicator dynamics		
[46]	Iterative prisoner's dilemma game	Reinforcement learning	Policy gradient algorithm	Achieve Nash equilibrium	
[47]	Sponsored block withholding attack model	Optimization theory	Formula derivation	Maximize revenue of attacking the mining pool	
[48]	Revenue optimization model of attack mining pool		Improved artificial Pareto-based bee colony algorithm		
[49]			Formula derivation	Achieve Nash equilibrium	
[50]	Self-sustaining attack model		Markov chain	Maximize revenue of attacking the mining pool	Hybrid block withholding attacks
[51]	New block withholding attack model				
[52]	Intermittent block withholding attack model		Formula derivation		
[53]	Uncle-block attack model				
[54]	Revenue optimization model of attack mining pool methods	Reinforcement learning	Markov chain		

attacks [42,43]. For example, a study [42] adopted the ZD strategy to optimize the selection of the strategy of the mining pools. Moreover, a condition was considered that the honest mining pool must satisfy when launching the ZD strategy (as shown in formula (4)) based on the revenue matrix of the attacking and honest mining pools. Therefore, the honest mining pool controls the revenue of the attacking mining pool through the ZD strategy and the revenue of two mining pools that reach the Nash equilibrium is effectively improved.

$$lm = \alpha U^L + \beta U^S - \gamma, \tag{4}$$

where lm represents the probability that the honest mining pool adopts the mixed strategy, U^L and U^S represent the steady-state expected revenues of the honest and attacking mining pool, respectively, and α, β, and γ represent relevant real number parameters. A study [43] considered a situation that two mining pools simultaneously used the ZD strategy and established a model for maximizing the overall revenue of the network (formula 5).

$$\max \quad E_{all} = E_1(p,q) + E_2(p,q), \forall q, \tag{5}$$

where E_{all} represents the overall revenue of the network, $E_1(p,q)$ and $E_2(p,q)$ represent the revenues of mining pools P_1 and P_2, respectively, and p and q represent the selection probability of attack and cooperation when mining pools P_1 and P_2, respectively, adopt the ZD strategy.

Block withholding attack among multiple mining pools

As shown in Fig. 7, in a block withholding attack among multiple mining pools, multiple mining pools launch block withholding attacks on other mining pools. As the numbers of mining pools and miners increase, directly employing the dual mining pool methods for optimizing block withholding attacks becomes difficult. Some scholars have studied this phenomenon from the perspective of miners and mining pools [44–49].

From the miners' perspective, a study [44] considered that miners dynamically migrate among multiple mining pools with random migration goals. In other words, a miner cannot assess whether a mining pool has encountered an attack when randomly migrating to another mining pool. From the perspective of the mining pool revenue, a study [44] formulated equations for attracting the mining pools to miners and the probability of miner migration to establish an optimization model (6). Based on the real computing power of Bitcoin, the paper selected the tile coding method in reinforcement learning to analyze the influence of miner migration on the computing power of the mining pools.

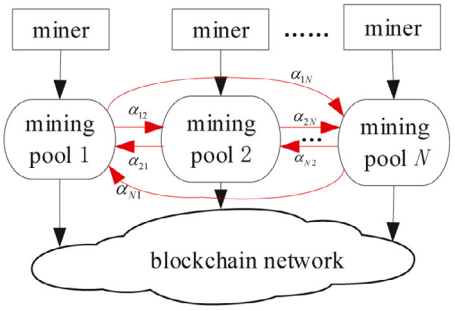

Fig. 7. Diagram of a block withholding attack among multiple mining pools.

$$\chi_i^t = \frac{\delta_i^t - \sum_{j=1}^{n} \delta_{a_{i,j}}^t}{\delta_T - \sum_{i=1}^{n} \delta_{a_i}^t} + \sum_{j=1}^{n} \left(\chi_j^t \times \frac{\delta_{a_{i,j}}^t}{\delta_j^t + \sum_{k=1}^{n} \delta_{a_{k,j}}^t} \right), \tag{6}$$

where χ_i^t represents the revenue of mining pool i in game round t, δ_i^t represents the computing power of i for honest mining in game round t, δ_T represents the total computing power of the network, $\delta_{a_{i,j}}^t$ represents the computing power assigned to mining pool j by i in game round t, $\delta_{a_i}^t$ represents the sum of the computing power of i for honest mining

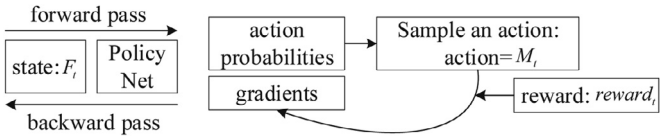

Fig. 8. Network structure diagram of the policy gradient algorithm.

in game round t and the computing power assigned by other mining pools, χ_j^t represents the revenue of j in game round t, $\delta_{a_{k,j}}^t$ represents the computing power assigned to j by mining pool k in game round t, and n represents the number of mining pools. To study the impact of malicious attackers on dynamic changes in the mining pools and the feasibility of autonomous migration among miners, a study [45] established a revenue model (7) that allowed multiple mining pools to launch block withholding attacks based on information such as the network propagation delay and attacker's penetration rate. For the model (7), the corresponding study adopted the replicator dynamics of evolutionary game theory from the mining pool perspective to obtain an evolutionary stability strategy and conditions for analyzing the change in the number of miners in the mining pool.

$$E_i = v \times \frac{h_i(\theta_i - \sum_{k \neq i} \theta_{i,k})}{\sum_{j=1}^{N} h_j(\theta_j - \sum_{k \neq i} \theta_{j,k})} e^{-\tau(s)/T} - \lambda h_i, \tag{7}$$

where E_i represents the revenue of i, v represents the reward for miners who submit the complete proof of work through honest mining, h_i represents the hash rate of each miner in i, θ_i represents the number of miners in i, $\theta_{i,k}$ represents the permeability of i with respect to k, N represents the number of mining pools in the network, $\tau(s)$ represents the latency of network propagation, T represents the average mining time, and λ represents the cost consumed by the hash rate of each miner.

From the mining pool perspective, another study [46] considered the behavior of multiple mining pools attacking each other as an iterative prisoner's dilemma model. In that study, the policy gradient algorithm in reinforcement learning was adopted for the timely adjustment of the mining strategy in the mining pool. In Fig. 8, the mining pool optimizes the penetration rate of the block withholding attack. First, it uses the policy network to obtain the probability distribution of the behavior using forward propagation. Then, it selects a behavior M_t, namely, the next step of its mining pool penetration rate. Finally, it obtains the mining pool revenue $reward_t$ and the new state F_{t+1} based on F_t and M_t. Furthermore, the policy gradient algorithm adjusts the relevant parameters in a network using backpropagation. This will ensure that the behavior of a mining pool with more revenue has a greater selection probability in subsequent processes than a mining pool with less revenue. This is geared toward efficiently and accurately selecting the optimal strategy. Therefore, this algorithm maximizes the revenue for each unit computing power in the mining pool and achieves mutual revenue, gaining a win-win status. Consider a case where the attacking mining pool conspires with other mining pools to attack a target mining pool. In an earlier study [47], a sponsored block withholding attack strategy was proposed for reducing the chance of the target mining pool successfully mining the block. The attacking mining pool hires a certain proportion of computing power from other mining pools to launch the block withholding attack on the target mining pool. Therefore, the attacking mining pool achieves the maximum revenue in the multiple mining pool environment. Considering a scenario where miners can opt to attack mining pools to join and launch honest mining or attack other pools, an improved artificial Pareto-based bee colony algorithm was proposed [48]. This algorithm obtains the composition scheme of each attacking mining pool and the working scheme of each miner under a block withholding attack to guarantee the group revenue of the mining pool. First, the algorithm randomly generates two arrays for each food source as the working scheme of each miner. Then, in terms of bee employment, the algorithm launches crossover, mutation, and Pareto operations based on the generated group revenue to preserve higher-yielding food sources. In terms of the onlooker operation, the algorithm performs crowding, crossover, mutation, and Pareto operations for the food source to ensure the diversity of food sources. Finally, to avoid falling into the optimal local solution in the scout bee operation, the algorithm updates the food source that cannot produce offspring on time. Considering a situation where multiple mining pools attack each other based on computing power allocation, a revenue optimization objective of mining pools was established (formula (8)) and the fixed computing power allocation algorithm and optimal computing power allocation algorithm were adopted to analyze the revenue change in the mining pools [49].

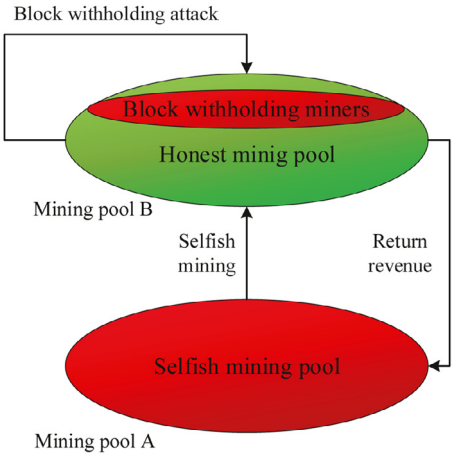

Fig. 9. Self-maintaining attack strategy.

$$\max \left(\frac{\alpha_{ii}}{\sum_{i=1}^{N} x_i \alpha_{ii}} \frac{\alpha_{ii}}{\alpha_{ii} + \sum_{j \neq i} \beta_{ji} \alpha_{ji}} - C_w \alpha_{ii} + \sum_{j \neq i} \left(\frac{\alpha_{jj}}{\sum_{i=1}^{N} x_i \alpha_{ii}} \frac{\alpha_{ij}}{\alpha_{jj} + \sum_{i \neq j} \beta_{ij} \alpha_{ij}} - C_p \alpha_{ij} \right) \right)$$
$$\text{s.t.} \quad 0 \leq \alpha_{ii} \leq 1$$
$$0 \leq \alpha_{ij} \leq 1, \forall j \tag{8}$$
$$\alpha_{ii} + \sum_{j \neq i} \alpha_{ij} = 1,$$

where α_{ii} represents the proportion of the computing power of i required for honest mining, N represents the number of mining pools, α_{ji} represents the computing power of the block withholding attack launched by j on i, C_w represents the cost required for honest mining by the mining pool, C_p represents the cost required for launching the block withholding attack by the mining pool, x_i represents the computing power of i, and β_{ij} represents a parameter such that $\beta_{ij} = x_i / x_j$.

Hybrid block withholding attack

In the mining process of a mining pool, the attacking mode is not limited to the block withholding attack alone; it also includes selfish mining, pool hopping, FAW, and 51% attacks. Some studies have focused on a combination of block withholding, selfish mining, 51%, and other attacks to improve the attack methods and the revenue of the mining pool.

A self-sustaining attack strategy based on selfish mining (refer to Section 4.4) and block withholding attacks was proposed [50]. Fig. 9 shows honest mining B and attacking mining pool A, respectively, in the network. By attacking mining pool A, a selfish mining attack on the honest mining pool B is launched and a portion of its computing power is assigned to the honest mining pool B to launch a block withholding attack to gain additional revenue. Based on classic selfish mining, when honest mining pool B successfully delivers a block, the attacking mining pool A gains a portion of its revenue by launching the block withholding attack. Considering that the attack computing power involves simply discarding the block without selfish mining, a study [51] proposed a block withholding attack combined with selfish mining. Because the pool manager cannot effectively distinguish honest miners from the min-

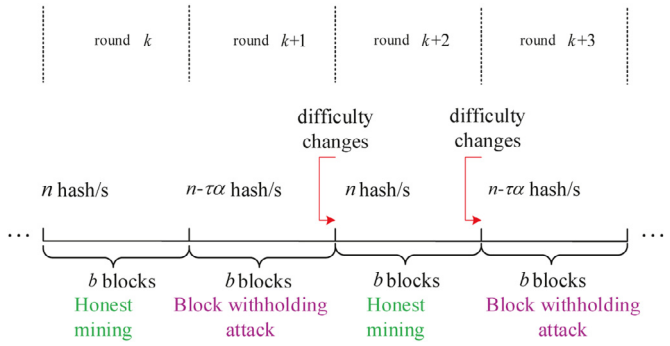

Fig. 10. Intermittent block withholding attack strategy.

ers who launch the block withholding attack, the latter selfishly mines a block reserved for obtaining revenue.

The launch of the block withholding attack affects the effective computing power of the entire network; hence, the network must adjust the difficulty value of block mining, that is, the requirement for miners to find the complete proof of work, to maintain the consensus efficiency of block mining according to the change in the effective computing power of the entire network. Considering the above situation, an intermittent block withholding attack strategy was proposed [52]. In this strategy, when the difficulty value of the entire network is high, the attacker uses the computing power to launch the block withholding attack. Conversely, when the difficulty value is low, the attacker transforms the block withholding attack into honest mining. As shown in Fig. 10, when the effective computing power of the entire network in round t is n hash/s, the attacker opts to launch a block withholding attack in round $k+1$ to gain more revenue, causing the effective computing power of the entire network to be $n-\tau\alpha$ hash/s (where $\tau\alpha$ represents the effective computing power used by the attacker to launch a block withholding attack). Therefore, the network must reduce the difficulty value of block mining to ensure block verification efficiency. Considering the above situation, the attacker opts to perform honest mining in round $k+2$ to gain revenue. Hence, the effective computing power of the entire network changes to n hash/s, thereby increasing the difficulty value of block mining in the network. Therefore, the attacker must relaunch a block withholding attack in round $k+3$.

In ETH, an uncle block refers to a block that is submitted but not selected as a primary block. An uncle block is provided a portion of the reward to increase fairness among miners. Considering this situation, an uncle-block attack strategy was proposed [53]. Compared with a block withholding attack strategy, the uncle-block attack strategy requires the attacker to submit all reserved blocks when other miners submit blocks, thus helping the attacker obtain multiple block rewards. Because the attack launched by the attacker heavily depends on the network deployment, once the network deployment changes dynamically, the revenue of the attacker decreases considerably. A study [54] proposed a mixed block withholding attack strategy. This strategy dynamically adjusts the attack behavior based on the environment. In other words, an attacker uses the Markov decision process based on reinforcement learning to realize a strategic switch between block withholding, FAW (refer to Section 4.5), and power adjusting withholding (PAW) attacks (refer to Section 4.5) to determine the optimal attack strategy according to the current network.

4.1.2. Defense methods

In a block withholding attack, the mining protocol of the mining pool is mainly attacked, which is destructive to any open mining pool. The revenue of honest miners in the mining pool is damaged, and the enthusiasm of miners to participate in block mining is decreased. Then, such an attack induces severe security threats to the blockchain. Therefore, many studies on the defense methods of block withholding attacks have been reported. Table 2 shows a division of the defense methods of block withholding attacks into three aspects: the revenue adjustment distribution, mining protocol adjustment, and credit mechanism establishment methods.

Revenue adjustment distribution method

The block withholding attack mainly attack the mining protocol of the mining pool so that the attacking computing power obtains a portion of the revenue of the attacked mining pool when the attacked mining pool distributes revenue. This process reduces the revenue of the attacked mining pool. Some studies have been devoted to the revenue distribution scheme of the mining pool [55–58]. Designing a new revenue distribution scheme reduces the revenue of malicious miners in the mining pool and hampers further block withholding attacks from attackers. For example, in a study [55], a new revenue distribution scheme was proposed by analyzing both the current mainstream proportional revenue distribution and shared revenue distribution schemes. A proportional revenue distribution scheme distributes revenue by calculating the proportion between the amount of the proof of work of each miner and the total amount of the proof of work. Similarly, a shared revenue distribution scheme distributes revenue using the principle of the shareholding system. In the proposed scheme [55], if the amount of the proof of work submitted by all miners exceeds the threshold value, the pool manager calculates its revenue using the proportional revenue distribution scheme; otherwise, the pool manager performs revenue distribution using the shared revenue distribution scheme. After completing the revenue distribution of all miners, the pool manager distributes the remaining revenue to the miner who submits the complete proof of work. Assuming an attacker can damage the revenue of honest miners in a mining pool via a block withholding attack, a special revenue to reward the miner who submits the complete proof of work was introduced [56]. Here, the pool manager first allocates a fixed proportion of revenue to miners who submits the complete proof of work and then distributes the remaining revenue to all miners based on the amount of the proof of work to ensure the revenue of honest miners in the mining pool. Combining the block revenue distribution scheme with the supervision mechanism, a study [57] proposed a miner's revenue model. Based on formula (9), the pool manager receives both the proof of work submitted by the miners and supervises the miners with a certain frequency ω. If the pool manager identifies a miner who does not mine honestly, the pool manager deducts an amount K from its revenue. Further, the corresponding study analyzed the influence of different supervision methods and supervision intensity on the strategies of miners to propose corresponding suggestions and countermeasures for supervision measures.

$$M_1 = \begin{array}{c} \\ MC \\ MA \end{array} \begin{array}{c} \begin{array}{cc} M_2 = MC & MA \end{array} \\ \begin{pmatrix} \frac{\nu}{2} - C_1, \frac{\nu}{2} - C_1 & \frac{\nu}{4} - C_1 - C_3, \\ & \frac{\nu}{4} - \omega K - C_2 \\ \frac{\nu}{4} - \omega K - C_2, \frac{\nu}{4} - C_1 - C_3 & -K - C_2, -K - C_2 \end{pmatrix} \end{array}, \tag{9}$$

where MC represents honest mining, MA represents a block withholding attack, M_1 and M_2 represent miners 1 and 2, respectively, ν represents the total revenue of the mining pool, C_1 represents the cost required for honest mining, C_2 represents the cost required for a block withholding attack, C_3 represents the additional cost required by an honest miner when a miner in launches a block withholding attack in the mining pool. Considering the block withholding attack strategy of delaying block submission to the pool manager, a study [58] proposed a payment method of the Takagi-Sugeno-Kang fuzzy system, which dynamically allocates revenues based on the fuzzy delay time of miners in the mining pool. A longer time required by an attacker for block submission signifies a lesser revenue for the attacker, thus ensuring effective resistance to a block withholding attack.

Mining protocol adjustment method

Because malicious miners only assign the partial proof of work to the pool manager and deliberately discard the complete proof of work

Table 2
Summary of defense methods of a block withholding attack.

Reference	Mechanism	Type	Solution	Purpose
[55]	Income distribution model	Optimization theory	Formula Derivation	Minimize the revenue of malicious miners
[56]	Revenue distribution model			
[57]	Evolutionary game	Game theory	Evolutionarily stable strategy	
[58]	Revenue distribution model	Fuzzy control theory	Fuzzy logic control	
[59]	Mining agreement adjustment	Cryptography theory	Hash function encryption and commitment protocol encryption	Avoid block withholding attack
[37]			Hash function encryption	
[60]			Public and private key encryption	
[61]	Zero-block addition model	Optimization theory	Formula derivation	Minimize the revenue of malicious miners
[62]	Miner credit mechanism	Statistical theory	Interval estimation	
[63] [64] [65]	Credit value model	Optimization theory	Formula derivation	

for block withholding attacks, some studies have focused on the mining protocol for defending the mining pool [37,59–61]. Such a mining protocol hinders miners from recognizing both the partial proof of work and complete proof of work, thus reducing block withholding attacks. To reduce the discarding of the complete proof of work by miners, a study [59] improved the existing mining protocol and proposed two schemes based on the hash function and encryption promise protocol. Here, miners were prevented from distinguishing the partial proof of work from the complete proof of work. The scheme based on the hash function requires the pool manager to combine random string information s and the difference value r between the difficulty values z' and z of proofs of partial work and complete work, respectively, and calculate p using the hash function. Then, the pool manager broadcasts p and z' to the miners in the mining pool to ensure that the miners cannot directly obtain the difficulty requirements of the complete proof of work. However, based on the proof of work reported by the miners, the pool manager calculates the hash value to determine the complete proof of work. Moreover, the encryption promise protocol replaces the hash function with the promise protocol. Another study [37] proposed a defense method of the oblivious share, where the pool manager obtains the field Extrahash from the randomly selected string Secertseed using the hash function and sends it to the miner along with the block mining task. The miner only submits the proof of work by assessing whether the hash value of the block is less than $2^{256}/2^{32}$. Therefore, malicious miners cannot directly launch block withholding attacks. The pool manager determines whether the block meets the difficulty requirements of the network setting by combining the proof of work and Secertseed information.

By combining the public-private key with the mining protocol, a study [60] proposed an improved mining protocol. In this protocol, the mining pool manager generates a public-private key and sends the public key information along with the block mining task to the miners. The miners with the public key information only determine whether the hash value of the block meets the difficulty requirement D_m of the mining pool manager. Therefore, the miners cannot discard the complete proof of work. The pool manager recalculates the hash value of the block by combining the proof of work and private key information. If the hash value equals the difficulty value D_{op}, the pool manager broadcasts the block to the network. Another study [61] proposed a zero-block mining protocol. In this protocol, the miner estimates the block mining time and delay time for block propagation using the effective computing power and difficulty requirement information of the entire network to calculate the time interval mat (formula (10)). If a miner cannot receive and mine a block within the time interval mat, a zero block containing the index of mat and the hash value of the previous block is generated, which is added to the blockchain. Because a malicious miner who launches the block withholding attack cannot add zero blocks within the time interval mat, the reserved block does not meet the requirements of the complete proof of work; thus, the attacker fails to launch the block withholding attack.

$$mat = \frac{D \times 2^{32}}{nethp} + ipt, \tag{10}$$

where D represents the difficulty requirements of the network, $nethp$ represents the effective computing power of the network, and ipt represents the delay time for block propagation in the network.

Credit mechanism establishment method

In this method, a miner who launches a block withholding attack sends only the partial proof of work. Moreover, the credit mechanism comprehensively evaluates the behavior of the target object and provides an accurate credit value. Therefore, studies have combined credit mechanisms and the existing mining agreement to evaluate the proof of work submitted by miners for the timely elimination of malicious miners [62–65]. To eliminate malicious miners on time, a study [62] proposed a credit mechanism based on the interval estimation and credit fluctuation. The pool manager calculates the performance satisfaction value and credit fluctuation value of the miners based on their proof of work and obtains their credit intervals using the interval estimation method. Assume that the performance satisfaction variance of the miner is in the rejection range of the credit interval or the credit fluctuation value of the miner exceeds the threshold value; then, the pool manager eliminates the miner. Furthermore, another study [63] proposed a credit value model (formula (11)) based on the proof of work to comprehensively evaluate the behavior of miners. This model combined the computing power and credit information of the miners to set the mining cost (formula (12)). The miners participated in the revenue distribution of the mining pool only when their credit values were used to pay the corresponding mining cost.

$$\varepsilon_l = \frac{npPoWs_l^{\frac{1}{\vartheta}}}{Y_l} + \varphi \times nfPoWs_l, \tag{11}$$

where ε_l represents the credit value of miner l, Y_l represents the computing power of miner l, $npPoWs_l$ represents the amount of the partial proof of work of miner l, $nfPoWs_l$ represents the amount of the complete proof of work of miner l, φ represents the revenue factor for the complete proof of work, and ϑ represents the revenue factor for the partial proof of work.

$$C_l = \frac{Y_l}{\varepsilon_l^{\frac{1}{Y_l+\alpha}}}, \tag{12}$$

where C_l represents the mining cost of miner l and α represents the mining cost parameter. By considering the history of malicious behaviors of miners, a study [64] proposed a verification process, in which the pool manager considers the number of historical attacks and the amount of currency owned by miners as evaluation indices for determining the credit value. The miner joins the mining pool only if it meets the credit requirements set by the pool manager, thereby improving the reliability of the miner. Another study [65] proposed a calculation scheme of the proof of work based on the miners' credit values to avoid the problem of miners with low credit values rejoining the mining pool using

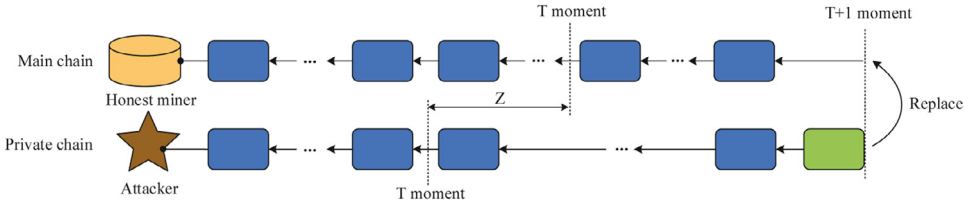

Fig. 11. Schematic of a 51% attack.

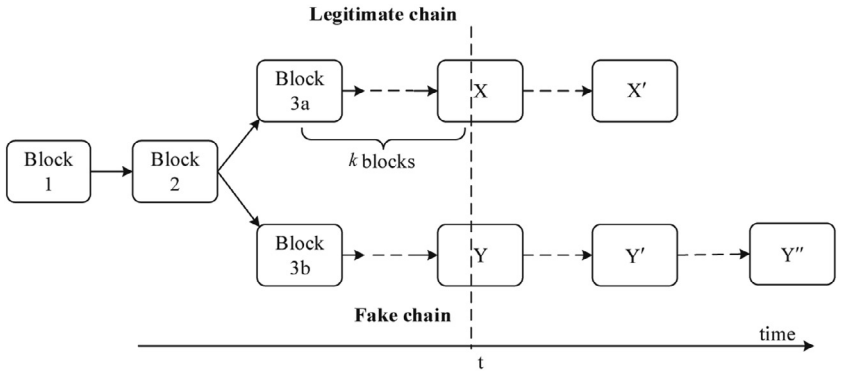

Fig. 12. Schematic of a 51% attack in an IoV environment.

a new identity. The scheme considers the credit value and life cycle of miners (the time required to join the mining pool), allowing the miner with a long life cycle to obtain high credit values during the mining process. Additionally, the pool manager reselects the miners in the mining pool at regular intervals; in other words, the pool manager increases the number of miners with high credit values and reduces the number of miners with low credit values in the mining pool. Therefore, the scheme prevents miners with low credit values from rejoining the mining pool.

4.2. 51% attack and defense methods

K. Sunny et al. [66] initially proposed the 51% attack and stated that an attacker shows malicious behavior, such as tampering with transactions and forging blocks, by controlling 51% of the computing power of the entire network to achieve double spending. As shown in Fig. 11, after transacting with other nodes, the attacker first regenerates a transaction based on the current main chain, making the original transaction invalid. Then, the attacker mobilizes all its miners to mine blocks that create a private chain. When the length of the attacker's private chain exceeds that of the main chain, the attacker broadcasts his private chain. Because other miners must obey the principle of mining on the longest chain, the attacker's private chain successfully replaces the main chain to achieve double spending. Because the 51% attack causes serious harm to the blockchain system, some studies have analyzed its success conditions [67,68]. A model was constructed by considering the influence of this attack on the blockchain of the internet of vehicles [67] (Fig. 12). Assuming the delivery time of a malicious miner is less than or equal to those of normal miners, the attacker realizes the rapid growth of the fake chain using malicious miners. Moreover, the corresponding study considers a block interval k from the beginning of the attack to the success of the attack as the key parameter to measure the success rate of the 51% attack. Moreover, the influence of factors, such as the message transmission time and computing power of the miner, on the 51% attack was examined. To analyze the attack strength and safety of 51% attacks, another study [68] proposed a method for simulating a blockchain. In this method, the actual operation process of the blockchain was simulated using the Java language. This method obtained a relationship between the number of process attacks and the number of states of the blockchain by adjusting the attack intensity of the 51% attack to evaluate the security of the blockchain in different scenarios.

Because a 51% attack requires the attacker to employ considerable amounts of computing power to construct a private chain, some studies have investigated its defense methods [69–72]. Assuming both legitimate and malicious blockchain branches exist in a network, a study [69] proposed a protocol of the proof of work based on historical weighted difficulty. In this protocol, when two branches conflict, honest miners calculate the historical weighted difficulty *HWD* of different branches based on formula (13) and continue mining on the branch with a larger *HWD* than the other branch. Because the attacker who launches the 51% attack only temporarily transfers its computing power to the malicious blockchain branch, the miner's block generation frequency is relatively small compared to normal conditions. The historical weighted difficulty of the malicious blockchain branch is less than that of the legitimate blockchain branch; hence, the malicious blockchain branch cannot become the main chain.

$$HWD_b = \sum_{g=1}^{sum} r_g \times \sum_{g=1}^{sum} d_g, \tag{13}$$

where HWD_b represents the historically weighted difficulty of branch b, d_g represents the difficulty value of block g, r_g represents the block generation frequency of miner g, and sum represents the number of miners mining on branch b. Analyzing the selfish mining behavior of the 51% attack, another study [70] proposed a punishment mechanism, in which the network increases the number of blocks that the attacker must privately mine by comparing the block number of the current main chain with the received block number, thus increasing the cost of attacking considerably. In the literature [71], a method for randomly selecting mining groups was proposed, where the network uses hash functions and wallet addresses to group miners. When a block is successfully mined, the network determines the next group to perform block mining based on its hash value and hash function. In addition, only miners in the group can participate in the competition for block mining, thus reducing the probability of a 51% attack. Another study [72] proposed a hybrid blockchain construction scheme that combined the revenues obtained from miners with the traditional proof of work, increasing the difficulty of a 51% attack. In this scheme, when a miner successfully completes mining, other miners calculate the probability of each miner

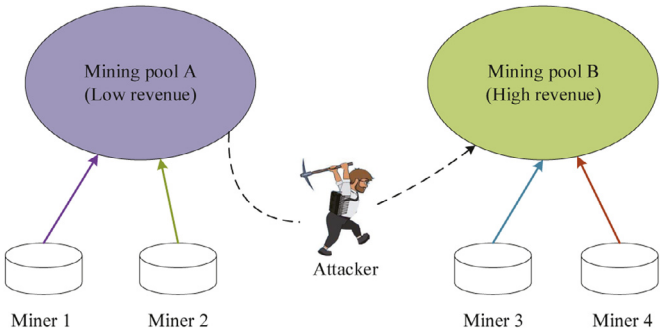

Fig. 13. Diagram of a pool hopping attack.

participating in the block construction using formula (14) based on public information (predefined system parameters and information on the blockchain). Then, using the probability information of each miner, the network uses a roulette-type wheel to determine that miner will eventually perform the block construction. When the miner successfully completes the block construction, the network provides it a certain amount of revenue.

$$o = \begin{cases} \xi \cdot stake + (1-\xi), & if\ it\ completes\ the\ mining \\ \xi \cdot stake, & if\ it\ fails\ to\ complete\ the\ mining \end{cases}, \quad (14)$$

where ξ represents the weight factor $\xi \in [0, 1]$ and *stake* represents the proportion of the revenue owned by the miner of the overall revenue of the miner.

4.3. Pool hopping attack and defense methods

In a pool hopping attack, an attacker selectively switches between mining pools to increase its revenue [37]. As shown in Fig. 13, the attacker joins the mining pool with the highest revenue after analyzing the revenues of multiple mining pools. This attack improves the stable revenue of the attacker and affects the effective computing power in the mining pool. Therefore, some studies have analyzed factors affecting the pool hopping attack [73,74]. When Internet of things (IoT) devices join a mining pool, they are challenged by limited computing power (IoT devices are unable to supervise the behavior of the mining pool manager) and rights (IoT devices are unable to check the contribution of their computing powers). Considering this situation, a study [73] proposed a pool hopping attack method initiated by the mining pool manager. In this method, when the pool manager determines that the ETH-based blockchain network can provide higher revenues than other blockchain networks, it transfers its computing power to the ETH-based blockchain network. Because the revenue distribution scheme of the miners still employs the original scheme, the pool manager can obtain additional revenues. Based on the differences in the difficulty values of various blockchain networks, another study [74] proposed a pool hopping attack method for different blockchain networks. In this method, if the difficulty value of the blockchain network is low, the attacker assigns additional amounts of computing power to block mining. However, when the difficulty value of the network exceeds the threshold value, the attacker withdraws the computing power from the blockchain network on time to identify other blockchain networks with lower difficulty values than the blockchain network to maximize the attacker's revenue.

Attackers must perform frequent mining pool switching when launching pool hopping attacks, enabling a path for the scholarly study of their defense methods. Considering that the pool manager obtains its working time based on the change in miners' revenues, a study [75] proposed a pool jump detection method based on revenue transaction sorting. This method sorts the revenue time of the miners in the mining pool based on the rounds of block mining. If the same miner is simultaneously rewarded in multiple mining pools, this method considers that the miner can launch a pool hopping attack. Another study [76] proposed a defensive model for pool hopping attacks based on a smart contract. The model includes three main parts: evaluation, contract signing, and updating. In the evaluation part, the pool manager provides certificates to the miners joining the pool, then each certificate can only be bounded to one miner address. At the same time, the mining pool manager calculates the miner's risk value based on the number of pool hopping and the number of violations on the smart contract in the certificate. In the contract signing step, the pool manager requires the miners with high-risk values to deposit a digital currency and prevents miners from leaving the mining pool without completing block mining. In the updating step, the pool manager updates the miner's certificate based on whether the miner has completed the block mining task, thus tracking the historical behavior of the miner and protecting the revenue of the existing miners. When a network encounters a pool hopping attack, the computing power of the entire network increases considerably. However, the network cannot adjust the difficulty value of block mining on time. After the completion of the pool hopping attack, the difficulty value of block mining is excessively high, inducing the problem of a long wait time for a block to be successfully mined [74,77]. Considering the above situation, a study [74] proposed a difficulty adjustment algorithm based on the block weight. Using the generation time of the last five blocks, the algorithm determines changes in the effective computing power of the network and adjusts the difficulty value of the network on time to effectively handle the sudden increase in the computing power caused by a pool hopping attack.

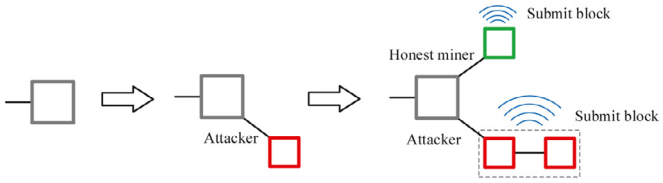

Fig. 14. Diagram of a selfish mining attack.

4.4. Selfish mining attack and defense methods

In a selfish mining attack, the attacker obtains additional revenues by continuously broadcasts blocks to the blockchain network [78]. As shown in Fig. 14, the attacker does not broadcast the block to the blockchain network even when mining the block first. When an honest miner successfully mines the block, the attacker broadcasts the block to the blockchain network, causing the current network to fork. If the attacker broadcasts multiple blocks to the blockchain network, additional revenues are gained when the network forks. Therefore, some studies have combined factors [79,80], such as the delay time for block propagation, in the mining process to achieve the optimal attack method. Considering that multiple miners launch selfish mining attacks in the mining process, a study [79] proposed a blockchain simulator. Because the block propagation delay is large, the simulator uses the Python language to implement actions performed by each miner under the mining event (the miner successfully mines the block) and receiving event (the miner receives the block mined by other miners) to analyze the revenue when multiple miners launch selfish mining attacks. Assuming that the miners in a network can accept the bribe from an attacker to lend their computing power, another work [80] proposed a smart bribery selfish mining attack strategy. When the attackers' private chain and the honest miner's public chain undergo forking, the attacker bribes the miners to transfer additional amounts of computing power to the private chain for mining. Furthermore, the attacker decides whether to launch an attack at the current moment using the Markov decision process based on reinforcement learning to maximize the attacker's revenue.

Some scholars have investigated the defense mechanism of selfish mining attacks because it enables an attacker to cause a bifurcation problem in the current network [81–83]. A study [81] proposed a detec-

tion mechanism of selfish mining attacks by considering that an attacker can obtain additional amounts of computing power via bribery when launching a selfish mining attack. This mechanism sets the expected recognition height for each transaction by analyzing the transaction size, sequence number, and block mining costs in the network. Because the attacker has a large amount of computing power to continuously mine blocks, a considerable difference exists between the block height of the subsequent block and the average expected confirmation height of all transactions in the previous block. However, honest miners cannot complete continuous block mining; hence, the difference between the current block height and the average expected confirmed height is small. Therefore, the proposed mechanism detects selfish mining attacks on time by analyzing this difference. Another study [82] proposed a defense method against selfish mining attacks using unforgeable time stamps. In this method, when a miner receives the block submitted by an attacker and an honest miner, the miner only accepts the block with the latest timestamp; otherwise, the miner accepts the latest received block. Therefore, to ensure the realization of the attack, the attacker must increase the attack computing power. To improve the ability of miners to handle network bifurcation, a study [83] proposed a new bifurcation solution strategy to alleviate the selfish mining behavior of attackers. The strategy requires miners to compare the weights of different branches based on the number of published blocks and uncle-blocks and transfer their computing powers to the branch with the highest weight when encountering network forking. If the weights of different branches are the same, the miner selects them randomly. Therefore, irrespective of whether the attacker publishes their block, the attacker cannot influence the choice of miners.

4.5. FAW attack and defense methods

A FAW attack refers to a new attack method that combines block withholding and selfish mining attacks [84]. It consists of malicious mining, target mining, and other mining pools in the network. Among them, the malicious mining pool retains some miners for attacking the target mining pool and some miners for honest mining. If the malicious mining pool successfully mines a block using an honest miner, it will immediately broadcast the block to the blockchain network and obtain revenue. If a miner assigned by the malicious mining pool successfully mines a block in the target pool, whether other mining pools have found the block must be determined. If other mining pools do not mine the block, the miner retains the block and does not broadcast; otherwise, the miner immediately broadcasts the block to the blockchain network, resulting in network forking. Conclusively, a FAW attack can help the attacker obtain the revenue of a block withholding attack and achieve additional revenues after network forking. Therefore, some studies have been devoted to optimizing the attack strategy of the FAW attack [85,86]. Because the traditional FAW attack only fixes the computing power distribution of the attacker, the attacker waste excessive amounts of computing power on the target pool with small revenues. A study [85] proposed a PAW attack based on the FAW attack. If malicious miners assigned by the malicious mining pool identify the complete proof of work in multiple target mining pools and when the complete proof of work is found in another pool, they submit the complete proof of work to the pool manager. Therefore, this strategy induces multiple forks in the network to maximize the revenue of the attacker. Another study [86] proposed an improved FAW attack strategy to increase the revenue of the forked network attributed to the FAW attack. In this strategy, when a miner assigned by a malicious mining pool induces network forking, the miner immediately shifts from the FAW attack to honest mining. Moreover, when the next block is successfully mined, the miner adjusts the strategy to the FAW attack.

Because FAW attacks exhibit the hazards of both block withholding and selfish mining attacks, some studies have investigated their defense methods [87,88]. In the literature [87], a silent timestamp method was proposed, where miners randomly send the generated silent timestamp and the proof of work to the pool manager. Based on the received silent timestamp, the pool manager sorts the proof of work of miners to ensure a time-sensitive submission of the proof of work. Therefore, when launching the FAW attack, the attacker only opts to discard the block, i.e., the block withholding attack. Another study [88] proposed an antiblock withholding reward mechanism for block withholding behavior in the FAW attack process, wherein greater revenues are provided to miners who show the complete proof of work to eliminate the motivation of attackers to launch FAW attacks.

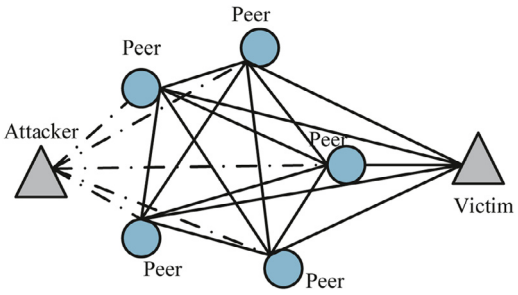

Fig. 15. Diagram of a DDoS attack.

5. Attack and defense methods of network communication and smart contracts

5.1. Attack and defense methods of network communication

5.1.1. DDoS attack and defense methods

In a DDoS attack, an attacker launches an attack on the target node by controlling multiple devices [89]. As shown in Fig. 15, the attacker first understands the network communication of the target node, then controls the devices to communicate with the target node, and finally sends a large amount of false information to the target node using the devices, making the target node unable to complete the block mining task. Some studies have focused on the DDoS attack methods for Bitcoin transactions [90,91]. For example, a study [90] proposed a DDoS attack strategy for a Bitcoin mining pool (a set of transactions generated in the network and is unlinked). This attack strategy involves two parts: distribution and attack stages. In the distribution stage, the attacker estimates the lowest transaction cost in the network and sends the Bitcoin to the malicious nodes in multiple transactions. In the attack stage, the malicious nodes launch numerous dust transactions (the cost of executing the transaction is considerably greater than the value of the transaction itself) between each other, yielding a considerably higher transaction generation speed in the network than that on the chain. Owing to the limited computing power of mining in the network, other legitimate users must pay additional transaction fees to ensure timely uploading of their transactions. Because cryptocurrency transactions allow many users to buy, store and sell cryptocurrencies online, many users can easily become targets for attackers to launch DDoS attacks. A study [91] proposed an event study method that combines the relationship between the Bitcoin transaction volume and the change in exchange prices to predict the expected Bitcoin transaction volume when subjected to a DDoS attack and analyze the impact of DDoS attacks on cryptocurrency transactions.

Because a DDoS attack requires the attacker to control many devices, some studies have investigated its defense methods [92–94]. For example, a study [92] proposed a distributed intrusion detection scheme based on fog computing for the DDoS attack of blockchain in IoT devices. This scheme mainly includes a sensor, fog, and interplanetary file system (IPFS) nodes. In this scheme, the sensor nodes efficiently collect the surrounding environment data by grouping and send the collected data to the IPFS nodes. The IPFS nodes distribute the received data based on the IPFS protocol to eliminate duplicate data as much as possible to ensure data security. Finally, the fog nodes use the feature selection method

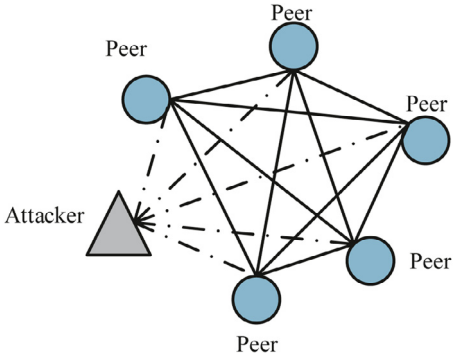

Fig. 16. Diagram of a Sybil attack.

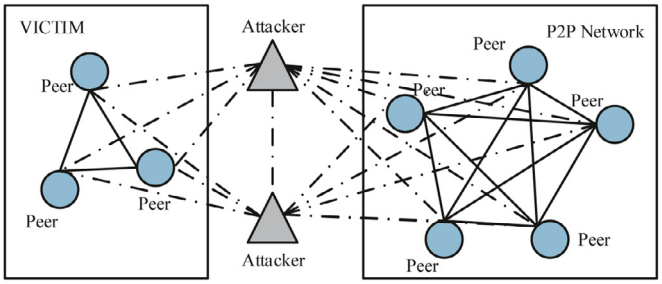

Fig. 17. Diagram of an eclipse attack.

based on mutual information to select the appropriate feature as the detection approach and realize attack detection using a smart contract. A study [93] proposed a detection method based on hybrid ensemble learning to improve the generalization performance of detecting DDoS attacks by considering that a combination of multiple classifiers exhibits better generalization ability than a single classifier. This method applies different ensemble learning algorithms to different blockchains. Moreover, for the classifier of the ensemble learning algorithm, this method integrates different lightweight classifiers into the same ensemble learning algorithm to improve the DDoS attack detection efficiency. Another study [94] proposed a deep learning-based attack detection method for DDoS attacks in the Bitcoin network. In this method, the data preprocessing stage involves the use of principal component analysis to extract features. In the DDoS attack detection stage, this method divides the real case data of the DDoS attack into training and test datasets. Further, it obtains the detection model using the neural network multilayer perceptron algorithm on the training set.

5.1.2. Sybil attack and defense methods

In a Sybil attack, the attacker disguises the identities of multiple nodes to deceive other nodes [95]. As shown in Fig. 16, when nodes engage the block consensus protocol, the attacker sends messages to other nodes by disguising the identities of multiple nodes to obtain the connection status of the blockchain network and mislead the routing of other nodes. Finally, when the number of nodes disguised by the attacker reaches a certain level, the consensus result of the block may be directly affected. Some studies have focused on an improved Sybil attack strategy [96]. A study [96] proposed an improved Sybil attack strategy by combining Sybil and 51% attacks to improve the probability of attackers achieving double spending. This strategy assumes that the Bitcoin network has a certain delay when synchronizing blocks. Therefore, the attacker uses numerous false identities to communicate with other nodes. Consequently, other nodes use considerable time and computation resources on communication and fail to obtain the current block information from other honest nodes on time. Therefore, the attacker uses this strategy to decelerate the growth rate of blocks on the main chain.

Because attackers must forge many false identities to achieve the Sybil attack, some studies have propagated its defense methods [97–99]. For example, a study [97] proposed a credit-based block consensus protocol that reduces the impact of a Sybil attack on the block consensus. This protocol ranks each node in descending order based on its credit score and ensures that the node with the highest number is selected to join the committee responsible for block consensus. If the committee successfully completes the current block consensus, the network slightly increases the credit values of all nodes in the committee; otherwise, it considerably reduces the credit values of all nodes. Therefore, this protocol can retain the credit values of malicious nodes at a low level. Another study [98] proposed a NetFlow algorithm to resist Sybil attacks. When the algorithm selects agent nodes for block consensus, it must calculate the credibility of the network nodes based on the transaction information and ensure that the authorized agent nodes lost part of the revenue for the block consensus. Therefore, it is difficult for attackers to launch a Sybil attack using the agent node. A study [99] proposed a solution to limit the Sybil attack of an attacker because an attacker uses a false identity forged by the Sybil attack to improve the propagation speed of their block and reduce the propagation speed of other users' blocks. In this solution, each participating node monitors the behavior of other nodes. Assume that a node only forwards the block of a specific user within a period. Then, other nodes believe that the node may launch a Sybil attack and blacklist the node to defend the Sybil attack from affecting the block consensus.

5.1.3. Eclipse attack and defense methods

In an eclipse attack, the attacker affecting the synchronization of the block by controlling the communication of the target node [100]. As shown in Fig. 17, the attacker first affects the communication process between the target node and the surrounding honest nodes and changes its communicable list to malicious nodes using the Sybil attack. When the target node only connects with a malicious node, the attacker can prevent the target node from achieving the block in the consensus process and further control the effective computing power of the target node. Therefore, some studies have focused on examining the improved eclipse attack strategy [101,102]. For example, a study [101] proposed an improved eclipse attack strategy under the condition of few internet protocol addresses. In this strategy, the attacker first sends an address message containing a controlled list of IP addresses to the target node. Then, the target node adds the controlled IP address to its communicable list. Next, the attacker captures valid messages in the network via network sniffing and uses different transmission control protocol ports to connect with the target node. Moreover, the attacker guarantees the validity of the connection by sending valid messages. In Bitcoin, a study [102] proposed a TendrilStaller attack strategy to making the target node cannot synchronize with the normal block on time. To increase the probability of a malicious node joining the target node's communicable list, the attacker increases the rate at which the malicious node sends new block information to the target node. Assume that a malicious node successfully joins the communicable list. Then, the attacker slightly reduces the communication efficiency between the malicious and target nodes so that other malicious nodes can join the communicable list. In the case of three malicious nodes on the communicable list of the target node, the attacker controls the malicious node to delay sending the block, thereby affecting the block synchronization of the node.

Because the attacker must control the communication between the target node and the network to realize the Sybil attack, its defense methods have been proposed [103].

1. Deterministic random eviction: The target node deletes some nodes based on the total number of nodes in the communicable list, thereby maintaining the dynamic change in the communicable list.
2. Random selection: The target node establishes a communicable list by randomly selecting surrounding nodes; hence, the attacker requires additional malicious nodes to achieve the eclipse attack.

3. Test before eviction: When updating the communicable list, the target node performs a test on the original node to ensure that the number of malicious nodes in the communicable list does not continue to increase.

A study [104] proposed a new dynamic network configuration protocol for depending against solar eclipse attacks. The protocol includes preprocessing, connection maintenance, and replacement node selection modules. In the preprocessing module, the network selects some nodes to form a high-level dynamic network and the remaining nodes form a low-level dynamic network. Furthermore, the nodes of the low-level dynamic network connect with those of the high-level dynamic network. In the replacement node selection module, the nodes of the high-level dynamic network must select a replacement node for updating the network before rebuilding the high-level dynamic network. The connection maintenance module retains the connection between the nodes of the low-level dynamic network and those of the high-level dynamic network when updating the high-level dynamic network. Another study [105] proposed a method for detecting eclipse attacks using a random forest classification algorithm. This method collects data under normal conditions and eclipse attacks and divides them into training and test datasets. Then, it uses the calculating information entropy method for extracting the features of the dataset and training the classifier using the random forest classification algorithm and the features of the dataset.

5.2. Attack and defense methods of smart contracts

5.2.1. Reentrancy attack and defense methods

In a reentrancy attack, an attacker obtains a large amount of revenue by recursively employing the vulnerability function of a smart contract [106]. In other words, when the target contract has gained a large amount of revenue on the blockchain, the attacker obtains all the revenue of the target contract by recursively adopting the vulnerability function of the target contract using a malicious contract. Some factors affecting reentrancy attacks have been studied. For example, a study [107] comprehensively analyzed the reentrancy attack and subsequent solutions of the DAO smart contract and proposed two paradoxes of hackers using the propositional logic that employs the vulnerabilities of smart contracts to steal the funds of DAO and the project developers should freeze the hackers' accounts. The two paradoxes are reminding us that we must pay attention to the security of smart contracts.

A reentrancy attack requires an attacker to identify vulnerabilities in recursive functions in smart contracts. Hence, studies on its defense mechanisms are essential. In the paper [108], a framework that combines static and dynamic analyses was proposed for the application binary interface coding generated using smart contracts to improve the detection efficiency of reentrancy attack vulnerabilities. First, the framework uses static analysis to identify vulnerable functions in smart contracts. Then, the network generates a related attacker contract using the static analysis and simulates the attack scenario based on the interaction between the attacker and related contracts to dynamically detect the reentrancy attack vulnerability. A study [109] proposed the Reguard, a fuzzing-based analyzer to detect reentrancy attack vulnerabilities in the blockchain. This method uses fuzzy-based technology to randomly generate different transactions as a test set. For a smart contract handling the test set, this method launches a reentrancy attack on some transactions to detect potential reentrancy attack vulnerabilities. Moreover, another study [110] proposed a method for establishing a Nusmv model by considering the reentrancy attack of ETH to ensure that the implementation of smart contracts meets standard requirements. This method includes the kernel layer for analyzing the blockchain behavior, application layer for implementing smart contracts, and framework environment layer for implementing the execution environment of smart contracts. Therefore, the method verifies whether the smart contract conforms to the behavioral norms of the stakeholders to detect reentrancy attack vulnerabilities.

5.2.2. Other attack and defense methods

An overflow attack occurs when an attacker sends wrong transaction information to the smart contract, inducing an error in the calculation of the revenue of the attacker in a smart contract [111]. Smart contracts are written using a bounded integer type. Therefore, the attacker sends the wrong transaction information, with the transaction value close to the bounded integer type. The smart contract overflows (the calculation result exceeds the upper bound) or underflows (the calculation result is lower than the lower bound) when calculating the attacker's revenue. In a short address attack, an attacker omits the last parameter of the transaction address, inducing errors in the smart contract during the transfer process [112]. Assume that a smart contract identifies that the length of a transaction address cannot meet the specified requirements during the transfer process. Then, it considers the corresponding length from the transfer amount and adds the corresponding length "0" to the right side of the transfer amount as a supplement. Therefore, the attacker can deliberately increase the transfer amount by omitting the last parameter of the transaction address.

A study [113] proposed an improved smart contract vulnerability detector using machine learning and dynamic fuzzy to handle various attack strategies by considering that an attacker realizes an attack using the vulnerability of smart contracts. This detector involves a vulnerability analyzer and a dynamic fuzzier. First, the network compiles the smart contract into an opcode and verifies it using the classifier of the vulnerability analyzer. Then the network performs outlier analysis on it using the dynamic fuzzier to improve the detection ability of unknown vulnerabilities. Another study [114] used a model checking method to explore all possible implementations that lead to security vulnerabilities to prevent malicious attacks. This method establishes all possible execution finite-state transition systems based on the actions performed by the smart contract and the labels used to indicate the execution results. Then, it analyzes the potential vulnerabilities according to security requirements.

6. Attack and defense methods of privacy thefts

Because blockchain is open and transparent, attackers track the transaction process using data mining to obtain privacy information of users. Therefore, on the issue of blockchain privacy, traders' user and transaction information are at a theft risk owing to privacy theft attacks. These attacks include identity privacy and transaction information attacks.

6.1. Attack methods

6.1.1. Identity privacy attack

In an identity privacy attack, an attacker obtains user privacy information using the connection between the trader's address on the chain and the user's real identity. The attacker infers a user's identity by monitoring public data in the global ledger and analyzes related transactions between addresses. Presently, common identity privacy attack mainly includes the key, replay, and impersonation attacks.

- Key attack: It occurs when an attacker illegally obtains a private key. Key attacks are performed using software and physical methods. In software methods, an attacker uses specific malicious software to obtain private key data from a cryptographic software system to steal the user's privacy information. In physical methods, an attacker directly steals the connection between the user's real identity and the trader's address on the chain and obtains the privacy information of the transaction participants using the transaction associated graph.
- Replay attack: It occurs when an attacker intercepts the user's transaction data and sends a packet received by the destination host,

thereby damaging the authentication of the user identity. Because the blockchain generates a private key during the signing process, an attacker combines the private key information to launch a replay attack [115], thus affecting the signing process of the blockchain.
- Impersonation attack: It occurs when the attacker pretends to be a legitimate user to perform unauthorized operations. In the transaction process, an attacker impersonates both parties of the transaction and simulates the exchange to steal privacy information. Once an attacker successfully fakes the identity of legitimate users, the user's privacy information is greatly threatened.

Another significant attack that has been extensively studied is the identity privacy attack. This attack focuses on disclosing the user's identity. A study [116] proposed a modular framework to obtain users' identity by considering a large amount of open and transparent information on the blockchain. This framework analyzes the transaction information on the blockchain and classifies the addresses that may belong to the same user in a cluster. Then, it attaches an identity to the user using tags and tracks their transaction liquidity. Finally, it obtains the connection between the trader's address on the chain and the user's identity. Another study [117] proposed a method for determining the relationship between the Bitcoin address and the IP address of its owner by considering the anonymous nature of Bitcoin exchanges. According to the transaction behavior of a single trader, this method deletes the transaction behavior jointly initiated by multiple traders in the trading information to find potential owners. Finally, by combining the frequency of each Bitcoin address in each IP address, this method calculates the relationship value between the Bitcoin address and the potential owner and determines the owner of the Bitcoin address based on this relationship value.

6.1.2. Transaction information attack

Transaction information attack means that the attacker uses the transaction diagram to analyze its potential information and obtain the user's transaction privacy. For example, although transaction information is disclosed anonymously, using this attack, the attacker can still download all the transaction history of the Bitcoin and analyze the user's associated transaction graph to obtain information on the type and operation behaviors of the user [118]. This attack includes privacy tracking, false data, and information leakage.

- Privacy tracking: After obtaining the user' information, an attacker traces the associated transactions and users based on the transaction relationship graph. A study [119] proposed an improved privacy-tracking method for the Bitcoin system. In this method, the attacker extracts a group of addresses from the public log on the chain and obtains the user's privacy information using a clustering algorithm.
- False data: Another study [120] pointed out that the attacker links to different data fragments of the same anonymous user, that is, the user's public key. If users are deanonymized, their information can be disclosed and falsified, in addition to displacing and impersonating users to perform false transactions.
- Information leakage: In the blockchain, users can use their public and private transactions; however, all transaction values and the remaining amount of each public key are publicly visible. Therefore, the disclosure of the user's information cannot be avoided. An earlier study [121] proposed a de-anonymization attack method that associates the pseudonym of the Bitcoin user behind the network address translation with the public IP address of the host generating the transaction, so as to obtain the information of the user who performed the transaction. This attack method involves four steps. 1. The attacker obtains a list of servers using a message query. 2. Based on the address information in the network and the list of servers, the attacker creates an anonymous list comprising attack targets. 3. The attacker uses other nodes to uniquely identify the client (i.e., network address translation or peer behind a firewall) in the anonymized list. 4. The attacker monitors the transaction information sent by the client using other nodes and obtains the relevant information of the transaction sender based on the data transmitted by the client.

6.2. Defense methods

6.2.1. Mixcoin protocol

D. L. Chaum initially proposed the Mixcoin protocol [122]. In this protocol, because the network integrates a large number of users' transactions into a single transaction, attackers are unable to determine the mapping pair of each user in the input and output. Thus, determining the connection between the input and output to address the privacy-tracking problem is impossible. Based on the Bitcoin mixing protocol Mixcoin, the paper [123] hides the input address and output address of any user in the mixing server by using a blind signature scheme and optimizing the public log. Some scholars apply the currency mixing mechanism to the field of blockchain security. Some studies have applied the currency Mixcoin protocol to blockchain security. A study [124] proposed a safe and efficient Bitcoin mixer Obscuro. By ensuring the consistency and integrity of the code and data, the mixer realizes the correct mixing operation and the protection of sensitive data in a trusted execution environment to exclude coin thefts and address link attacks of malicious service providers. In another study [125], a digital currency DashCoin based on the Bitcoin code was proposed. This digital currency uses the coin mixing mechanism. Moreover, it builds a two-layer network comprising a master node and miners to realize the Mixcoin protocol. In the first layer of the network, miners use mining methods for network security protection. In the second layer of the network, networks bundle transactions in a mix and send them when multiple funds are combined to achieve privacy protection.

Although the Mixcoin protocol reduces the attack risk of privacy tracking and defends users' privacy, it allows illegal personnel to conduct money laundering. A study [126] proposed a detection method based on the Mixcoin protocol to detect transactions that involve money-laundering schemes in a timely manner. This method decomposes the transaction into multiple subtransactions and analyzes the input and output of the transaction. Then, it determines the probability of multiple inputs or outputs belonging to the same transaction. Therefore, this method can track money-laundering transactions using the Mixcoin protocol to determine whether relevant users participate in money-laundering transactions.

6.2.2. Zero-knowledge proof

The zero-knowledge proof was proposed by S. Goldwasser et al. [127]. The zero-knowledge proof implies that a verifier completes the confirmation of transaction information without providing any valid information. Therefore, it improves the user's privacy protection and solves the problem of information leakage. The zero-knowledge proof involves interactive and noninteractive categories. Because a non-interactive zero-knowledge proof does not require an interactive process and avoids the possibility of attacker collusion, it is widely used in blockchain [128]. Some studies have applied the zero-knowledge proof to blockchain security. For example, a study [129] proposed a program for protecting the user's data privacy using the zero-knowledge proof to process the user's original data. After the network completes the zero-knowledge proof, a smart contract monitors the user's data. Based on monitored data deviations, the program verifies the user's behavior and determines whether the relevant data are being tampered with. Another study [130] used the zero-knowledge proof to propose a digital currency Zerocoin by considering that the realization of the Mixcoin protocol requires the user's complete trust. Zerocoin uses the Zerocoin protocol to break the connection between individual Bitcoin transactions, thereby preventing the disclosure of the relevant address information of both parties in the transaction. Based on this study [130], another study [131] proposed a digital currency Zcash to improve the privacy

and anonymity of transactions. The combination of digital currency and zero-knowledge proof technology zk-SNARK enables the sender to prove their corresponding assets using mathematics, ensuring that only the user with the view key assesses the transaction information. Then, it encrypts the end result to hide the parties and amounts on the transaction records and allows miners to verify transactions, even when the exact details of the transaction are unknown. In an earlier study [132], a digital currency super zero coin was proposed. This digital currency uses a noninteractive zero-knowledge proof mechanism to increase the encryption speed of individual transactions and ensure the processing speed of anonymous transactions.

6.2.3. Ring signature

R. L. Rivest et al. proposed the ring signature [133], which indicates a network that allows a ring member to sign other members using its own private key and the public key of other members. The verifier cannot determine the actual signer; however, they can confirm that the signer is in the ring. Such a method satisfies the complete anonymity of the signer and solves the problem of identity attacks in the signing process. In the ring signature, any user randomly selects a set of signers. The signer uses their private key and the public key of other users to sign any message without the approval or help of other users. The ring signature has been applied to blockchain security. For example, a study [134] proposed a certificateless key protocol based on pairing using the ring signature. In the first phase of communication, this protocol generates a session key that enables a user to save and use sensitive data. In the second phase, the protocol uses the certificateless ring signature to verify the user's identity, thus reducing computational costs while maintaining the user's anonymity. In an earlier study [135], a ring signature scheme based on an elliptic curve algorithm was proposed. This scheme uses the complete anonymity of ring signatures to construct a privacy data storage protocol and ensure the privacy of data and user's identity in blockchain applications. Because the public key infrastructure (PKI) model uses registration keys for ring signatures, the security of the existing signature and the privacy of user's identity is not guaranteed once an attacker obtains the registration keys. Considering the above situation, as study [136] proposed a strong forward secure ring signature scheme based on the Rivest-Shamir-Adleman algorithm. This method ensures anonymity and provides forward and backward securities to the signer. Forward security ensures that although the attackers obtain the user's current key, they cannot calculate the key of the previous stage. Similarly, backward security ensures that even if the attacker obtains the user's current key, they cannot calculate the user's key at a later stage. This method divides the signature cycle into several periods and updates the user's private key in stages. Therefore, even if the attacker obtains the user's private key at the current moment, it cannot achieve the private key of the previous stage or calculate the later moment, thus realizing the strong forward secure ring signature scheme. In another study [137], a digital currency Monero was proposed based on the ring signature technology. This digital currency allows the transaction sender to join a transaction group and perform ring signatures as a transaction group instead of a single user. Finally, the verifier proves the legitimacy of the transaction group. Because the ring signature of the transaction group contains multiple signatures, the attacker cannot determine which signature is sent by the real trader, thus concealing the sender's identity in a real transaction. Based on this study [137], another study [138] proposed an improved protocol that requires other users to determine whether the ring signature of the transaction satisfies the relevant conditions when forwarding the transaction information to improve the reliability of the transaction.

7. Future research direction

In Section 4, Section 5, and Section 6, we combine the proposed attack classification system to conduct in-depth investigations on the mining attack and defense methods of the mining pool, network communication and smart contracts, and privacy thefts, respectively. However, the rapid development of the blockchain technology and its applications has been associated with several emerging issues that need to be further studied. Therefore, we discuss future research directions for blockchain security and privacy.

- **Mining attack and defense methods of mining pool:**
  By analyzing the mining protocol vulnerability, the attack of the mining pool is performed using different attack methods to improve the attacker's revenues. However, the following problems exist in the attack of the mining pool. (1) A block withholding attack requires the attacker to reasonably allocate the computing power. (2) A 51% attack requires the attacker to control 51% of the computing power of the network. (3) A pool hopping attack requires improved network conditions. (4) A selfish mining attack requires the attacker to have some amounts of computing power and improved network conditions. Therefore, based on the advantages of different attack strategies, we integrate multiple attack methods, such as the block withholding, 51%, pool hopping, and selfish mining attacks, to study the fusion of multiple attack methods of the mining pool. This attack method switches among attack methods or forms a new type of attack method based on the current attack conditions to improve the attack efficiency of the attackers.
  The defense method for the attacks of the mining pools is mainly performed by analyzing the attack modes and attack purposes to reduce the frequency of attacks. However, the following problems are still evident in the defense methods of the mining pool. (1) The revenue adjustment distribution method only reduces the revenue of malicious miners. (2) The mining protocol adjustment method only increases the attack difficulty of malicious miners. (3) The credit mechanism establishment method fails to comprehensively analyze the behavior of malicious miners. Therefore, when studying the methods for depending attacks on mining pools, we calculate the miner's credit value using the miner's reporting status of the proof of work, offline rate, and other characteristics in the mining process and study defense methods based on the miners' mining behavior. This method realizes the distribution of malicious miners in the mining pool and ensures the efficiency of block mining and the revenues of honest miners.
- **Attack and defense methods of network communication:**
  Interfering with the normal communication of nodes affects the block consensus process and obtains large revenues. However, the present study on network communication attacks is challenged by the insufficiently comprehensive established attack model. In the actual attack process, malicious nodes can obtain information such as the node communication delay, revenue distribution scheme of the mining pool, and attack costs. Therefore, when studying the attack strategy of network communication, we integrate the above factors to improve the practicability of the attack strategy.
  The defense method of a network communication attack is mainly based on the extraction of a large number of communication data characteristics between nodes to assess whether the network is attacked. Because the data characteristic calculation method analyzes the entire network communication situation and cannot comprehensively evaluate the communication situation of a single node, detecting malicious nodes launching the network communication attack is difficult. Therefore, we can introduce machine learning to further improve the data feature calculation method and study new defense methods of a network communication attack to detect malicious nodes launching attacks.
- **Attack and defense methods of smart contract:**
  Smart contract attacks use different attack strategies by studying the code vulnerability of smart contracts to improve the attacker's revenue. However, owing to the limited storage space of the blockchain, the smart contract stores a part of the data outside the chain, reducing the storage capacity of the on-chain data. However, it does not

guarantee the malicious tampering of the off-chain data. Therefore, we can study a new attack strategy of the smart contract using the security vulnerabilities of the off-chain data to increase the attack mode of attackers.

The defense method of a smart contract attack mainly detects whether there are similar vulnerabilities in the smart contracts based on the characteristics of the vulnerability, thereby reducing the probability of malicious attacks. However, this method must establish multiple models to learn sample characteristics or conduct dynamic tests by simulating attack scenarios during the implementation process to reduce the detection efficiency of vulnerabilities to a certain extent. Therefore, we can study a new defense method of smart contract attacks that reduces the time complexity and improves the detection efficiency of vulnerabilities.

- **Attack and defense methods of privacy thefts:**

To obtain the privacy information of the transaction user on the chain, a privacy theft attack studies the link between the trader's address on the chain and the user's real identity using machine learning methods, such as IP clustering and the address clustering of transaction data. With an increase in the forking phenomenon and the number of exchanges in the blockchain network, attacking the forked chains or transactions to steal privacy is predominant. For example, in SushiSwap, the attacker obtains revenue by operating the exchange price of the trading pair. Therefore, we can study new privacy theft attack strategies using vulnerabilities of exchanges and forked chains to strengthen the new attack methods.

The defense methods of a privacy theft attack mainly combine cryptographic algorithms to enhance the anonymity of blockchain to block the attack or increase the attack cost. However, the current methods face the following problems. (1) For the Mixcoin protocol, the attacker analyzes the anonymous transaction set of the mixed-currency protocol and can determine that it is associated with the transaction address. (2) The zero-knowledge proof technology requires long computational times to generate proof and consumes considerable computational resources, affecting the throughput of transactions. (3) Because the ring signature randomly selects a certain number of users to sign the transaction, the attack resistance of the anonymous set is insufficient. Therefore, we can study the new defense method of privacy theft attacks using big data analysis and cryptography technology to realize the encryption scheme with improved efficiency and performance.

- **Application of attack and defense methods in different scenarios:**

Blockchain applications in smart finance, smart home, smart medical care, smart transportation, and other fields have become a future development trend. However, different application scenarios have different characteristics. For example, in smart homes, a central gateway with good performance usually functions as the mining pool manager of the blockchain. Moreover, several IoT devices with limited computing power are the miners performing block mining. In intelligent transportation, the vehicle nodes responsible for block mining often move quickly in an area. However, the roadside base station, which is the mining pool manager, is always static. In smart medicine, the network provides the medical institution with different consensus rights based on the level of the medical institution; hence, a medical institution with a higher level exhibits more mining power. In the process of realizing the multidomain applications of blockchain, problems, such as the inconsistent performance of node devices, inconsistent mobility of nodes, and different right scope of nodes, exist. Therefore, we must further improve or study new attack and defense methods of the blockchain under different application scenarios to expand the practicability and applicability of the methods and maximize the value of blockchain.

8. Conclusion

This work comprehensively reviews the attack and defense methods of the blockchain in the state-of-arts to provide a bird-view of the security and privacy issues on blockchain. First, we introduce the characteristics, structure, workflow, and classification of the blockchain. Afterward, we introduce the basic knowledge of game theory, cryptography, and reinforcement learning. Then, we propose a blockchain attack classification system to categorize the blockchain security and privacy issues. Furthermore, we discuss the attack and defense methods for blockchain security and privacy issues from three categories, namely, (1) the attack and defense methods of the mining pool for blockchain security issues, such as the block withholding, 51%, pool hopping, selfish mining, and FAW attacks, in the attack type of consensus excitation ; (2) the attack and defense methods of network communication and smart contracts for blockchain security issues, such as the DDoS, Sybil, eclipse, and re-entry attacks, in the attack type of middle protocol; and (3) the attack and defense methods of privacy thefts for blockchain privacy issues, such as identity privacy and transaction information attacks, in the attack type of application service. Finally, we point out the most promising future research directions for blockchain security and privacy.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was supported by the "Ling Yan" research and development project of Zhejiang Province of China under Grant No.2022C03122, Project Intelligentizationand Digitization for Airline Revolution #2018R02008, and Public Welfare Technology Application and Research Projects of Zhejiang Province of China under Grants No. LGF22F020006 and No. LGF21F010004.

References

[1] M.B. Hoy, An introduction to the blockchain and its implications for libraries and medicine, Med Ref Serv Q 36 (3) (2017) 273–279.
[2] J. Zhang, R. Tian, Y. Cao, X. Yuan, Z. Yu, X. Yan, X. Zhang, A hybrid model for central bank digital currency based on blockchain, IEEE Access 9 (2021) 53589–53601.
[3] M. Du, Q. Chen, J. Xiao, H. Yang, X. Ma, Supply chain finance innovation using blockchain, IEEE Trans. Eng. Manage. 67 (4) (2020) 1045–1058.
[4] S. Zhang, J. Rong, B. Wang, A privacy protection scheme of smart meter for decentralized smart home environment based on alliance blockchain, International Journal of Electrical Power & Energy Systems 121 (10) (2020) 1–10.
[5] S. Kim, J. Huh, Artificial neural network blockchain techniques for healthcare system: focusing on the personal health records, Electronics (Basel) 9 (5) (2020) 1–30.
[6] J. Kang, Z. Xiong, D. Niyato, D. Ye, D.I. Kim, J. Zhao, Toward secure blockchain-enabled internet of vehicles: optimizing consensus management using reputation and contract theory, IEEE Trans. Veh. Technol. 68 (3) (2019) 2906–2920.
[7] M. Firdaus, K. Rhee, On blockchain-enhanced secure data storage and sharing in vehicular edge computing networks, Applied Sciences 11 (1) (2021) 1–21.
[8] Global blockchain spending to hit US$16B by 2023, Aug. 2020. https://www.asiablockchainreview.com/global-blockchain-spending-to-hit-us16b-by-2023/.
[9] J. Han, J. Zou, H. Jiang, Q. Xu, Research on mining attacks in bitcoin, Journal of Cryptologic Research 5 (5) (2018) 470–483.
[10] X. Han, Y. Yuan, F. Wang, Security problems on blockchain: the state of the art and future trends, Acta Automatica Sincia 45 (1) (2019) 206–225.
[11] Z.H. Li Fang Li Zhuoran, Research on the progress in cross-chain technology of blockchains, Journal of Software 30 (6) (2019) 1649–1660.
[12] K.C. Chaudhary, V. Chand, A. Fehnker, Double-spending analysis of bitcoin, in: Pacific Asia Conference on Information Systems Proceedings, Association for Information Systems, 2020.
[13] M. Iqbal, R. Matulevičius, Exploring sybil and double-spending risks in blockchain systems, IEEE Access 9 (2021) 76153–76177.
[14] C. Natoli, V. Gramoli, The balance attack or why forkable blockchains are ill-suited for consortium, in: 2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), IEEE, 2017, pp. 579–590.

[15] L. Chen, J. Peng, Y. Liu, J. Li, F. Xie, Z. Zheng, Phishing scams detection in ethereum transaction network, ACM Transactions on Internet Technology (TOIT) 21 (1) (2020) 1–16.

[16] K. Hu, J. Zhu, Y. Ding, X. Bai, J. Huang, Smart contract engineering, Electronics (Basel) 9 (12) (2020) 2042.

[17] A.F. Aysan, A.U.I. KHAN, H. TOPUZ, A.S. TUNALI, Survival of the fittest: a natural experiment from crypto exchanges, The Singapore Economic Review (2021) 1–20.

[18] X. Li, P. Jiang, T. Chen, X. Luo, Q. Wen, A survey on the security of blockchain systems, Future Generation Computer Systems 107 (1) (2020) 1–13.

[19] Z. Liu, N.C. Luong, W. Wang, D. Niyato, P. Wang, Y.-C. Liang, D.I. Kim, A survey on blockchain: a game theoretical perspective, IEEE Acces 7 (7) (2019) 1–29.

[20] J. Cheng, L. Xie, X. Tang, N. Xiong, B. Liu, A survey of security threats and defense on blockchain, Multimed Tools Appl 8 (8) (2020) 1–30.

[21] J. Angelis, E.R. da Silva, Blockchain adoption: a value driver perspective, Bus Horiz 62 (3) (2019) 307–314.

[22] V. Gatteschi, F. Lamberti, C. Demartini, C. Pranteda, V. Santamaria, To blockchain or not to blockchain: that is the question, IT Prof 20 (2) (2018) 62–74.

[23] S. Nakamoto, Bitcoin: a peer-to-peer electronic cash system, Decentralized Business Review (2008) 1–9.

[24] Z. Zheng, S. Xie, H. Dai, X. Chen, H. Wang, An overview of blockchain technology: architecture, consensus, and future trends, in: 2017 IEEE international congress on big data (BigData congress), 2017, pp. 557–564.

[25] D. Efanov, P. Roschin, The all-pervasiveness of the blockchain technology, Procedia Comput Sci 123 (1) (2018) 116–121.

[26] M.D. Pierro, What is the blockchain? Computing in Science & Engineering 19 (5) (2017) 1–4.

[27] S. Wang, L. Ouyang, Y. Yuan, X. Ni, X. Han, F.-Y. Wang, Blockchain-enabled smart contracts: architecture, applications, and future trends, IEEE Transactions on Systems, Man, and Cybernetics: Systems 49 (11) (2019) 2266–2277.

[28] P. Ruan, T.T. Anh Dinh, Q. Lin, M. Zhang, G. Chen, B. Chin Ooi, Revealing every story of data in blockchain systems, ACM SIGMOD Record 49 (1) (2020) 70–77,.

[29] J. Zhang, Y. Liu, Z. Zhang, Research on cross-chain technology architecture system based on blockchain, in: International Conference in Communications, Signal Processing, and Systems, 2019, pp. 2609–2617.

[30] S. Gao, D. Zheng, R. Guo, C. Jing, C. Hu, An anti-quantum e-voting protocol in blockchain with audit function, IEEE Access 7 (7) (2019) 115304–115316.

[31] Q. Zhu, S. Rass, Game theory meets network security: a tutorial, in: the 2018 ACM SIGSAC Conference on Computer and Communications Security, 2018, pp. 2163–2165.

[32] Y. Jiang, K. Zhou, X. Lu, S. Yang, Electricity trading pricing among prosumers with game theory-based model in energy blockchain environment, Appl Energy 271 (8) (2020) 1–16.

[33] W. Zhou, P. Li, Q. Wang, N. Nabipour, Research on data transmission of wireless sensor networks based on symmetric key algorithm, Measurement 153 (3) (2020) 1–18.

[34] Y. Luo, X. Ouyang, J. Liu, L. Cao, An image encryption method based on elliptic curve elgamal encryption and chaotic systems, IEEE Access 7 (3) (2019) 38507–38522.

[35] K. Cobbe, O. Klimov, C. Hesse, T. Kim, J. Schulman, Quantifying generalization in reinforcement learning, in: the 36th International Conference on Machine Learning, 2019, pp. 1282–1289.

[36] N. Cong Luong, D.T. Hoang, S. Gong, D. Niyato, P. Wang, Y.-C. Liang, D. In Kim, Applications of deep reinforcement learning in communications and networking: a survey, IEEE Communications Surveys & Tutorials 21 (4) (2019) 3133–3174.

[37] M. Rosenfeld, Analysis of Bitcoin Pooled Mining Systems, 2011. https://arxiv.org/pdf/1112.4980.pdf.

[38] W. Li, M. Cao, Y. Wang, C. Tang, F. Lin, Mining pool game model and nash equilibrium analysis for pow-based blockchain networks, IEEE Access 8 (8) (2020) 101049–101060.

[39] D. Wu, X. Liu, X. Yan, R. Peng, G. Li, Equilibrium analysis of bitcoin block withholding attack: a generalized model, Reliability Engineering and System Safety 185 (5) (2019) 318–328.

[40] R. Qin, Y. Yuan, F. Wang, Optimal block withholding strategies for blockchain mining pools, IEEE Trans. Comput. Social Syst. 7 (3) (2020) 709–717.

[41] T. Yang, Z. Xue, Game theory among mining pools in blockchain system, Communications Technology 52 (5) (2019) 1189–1195.

[42] C. Tang, Z. Yang, Z. Zheng, Z. Chen, L. Xiang, Game dilemma analysis and optimization of pow consensus algorithm, Acta Automatica Sincia 43 (9) (2017) 1520–1531.

[43] Q. Hu, S. Wang, X. Cheng, A game theoretic analysis on block withholding attacks using the zero-determinant strategy, in: Proceedings of the International Symposium on Quality of Service., 2019, pp. 1–10.

[44] A.T. Haghighat, M. Shajari, Block withholding game among bitcoin mining pools, Future Generation Computer Systems 97 (8) (2019) 482–491.

[45] S. Kim, S. Hahn, Mining pool manipulation in blockchain network over evolutionary block withholding attack, IEEE Access 7 (7) (2019) 144230–144244.

[46] T. Wang, S. Yu, B. Xu, Research on proof of work mining dilemma based on policy gradient algorithm, Journal of Computer Applications 39 (5) (2019) 1336–1342.

[47] S. Bag, S. Ruj, K. Sakurai, Bitcoin block withholding attack: analysis and mitigation, IEEE Trans. Inf. Forensics Secur. 12 (8) (2017) 1967–1978.

[48] H. Chen, Y. Chen, M. Han, B. Liu, Q. Chen, Z. Ma, A novel anti-attack revenue optimization algorithm in the proof-of-work based blockchain, in: International Conference on Wireless Algorithms, Systems, and Applications, 2020, pp. 40–50.

[49] Y. Chen, H. Chen, M. Han, B. Liu, Q. Chen, T. Ren, A novel computing power allocation algorithm for blockchain system in multiple mining pools under withholding attack, IEEE Access 8 (8) (2020) 155630–155644.

[50] X. Dong, F. Wu, A. Faree, D. Guo, Y. Shen, J. Ma, Selfholding: a combined attack model using selfish mining with block withholding attack, Computers & Security 87 (11) (2019) 1–11.

[51] J. So, Ieice transactions on fundamentals of electronics, communications and computer sciences, Computers & Security 102 (1) (2019) 300–302.

[52] J. Ke, P. Szalachowski, J. Zhou, Q. Xu, Z. Yang, Ibwh: an intermittent block withholding attack with optimal mining reward rate, in: International Conference on Information Security, 2019, pp. 3–24.

[53] S. Chang, Y. Park, S. Wuthier, C. Chen, Uncle-block attack: blockchain mining threat beyond block withholding for rational and uncooperative miners, in: International Conference on Applied Cryptography and Network Security, 2019, pp. 241–258.

[54] Y. Wang, G. Yang, T. Li, L. Zhang, Y. Wang, L. Ke, Y. Dou, S. Li, X. Yu, Optimal mixed block withholding attacks based on reinforcement learning, Int. J. Intell. Syst. 9 (9) (2020) 1–17.

[55] O. Schrijvers, J. Bonneau, D. Boneh, T. Roughgarden, Incentive compatibility of bitcoin mining pool reward functions, in: International Conference on Financial Cryptography and Data Security, 2016, pp. 477–498.

[56] S. Bag, K. Sakurai, Yet another note on block withholding attack on bitcoin mining pools, in: International Conference on Information Security, 2016, pp. 167–180.

[57] Y. Cheng, Z. Xu, Study on the block withholding attack based on the evolutionary game, Journal of XiDian University 47 (5) (2020) 1–12.

[58] L. Liu, W. Chen, L. Zhang, J. Liu, J. Qin, A type of block withholding delay attack and the countermeasure based on type-2 fuzzy inference, Mathematical biosciences and engineering: MBE 1 (1) (2019) 237–309.

[59] S. Bag, S. Ruj, K. Sakurai, Bitcoin block withholding attack: analysis and mitigation, IEEE Trans. Inf. Forensics Secur. 12 (8) (2017) 1967–1978.

[60] Z. Fan, Y. Zhen, W. Xie, Blockexchain: a method of resisting block withholding attack of blockchain, in: Basic & Clinical Pharmacology & Toxicology, 2018, pp. 54–55.

[61] S. Solat, M. Potop-Butucaru, Zeroblock: timestamp-free prevention of block-withholding attack in bitcoin, in: SSS 2017: Stabilization, Safety, and Security of Distributed Systems, 2017, pp. 356–360.

[62] T. Changbing, W. Luya, W. Guanghui, Z. Zhonglong, Incentivizing honest mining in blockchain networks: a reputation approach, IEEE Trans. Circuits Syst. II Express Briefs 67 (1) (2019) 117–121.

[63] K. Abdellah, R. Abderrezak, Poolcoin: Toward a distributed trust model for miners' reputation management in blockchain, in: 2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC), 2020, pp. 1–6.

[64] R. Bala, R. Manoharan, Security enhancement in bitcoin protocol, in: 2018 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), 2018, pp. 1–4.

[65] M. NojoumianEmail, A. Golchubian, L. Njilla, K. Kwiat, C. Kamhoua, Incentivizing blockchain miners to avoid dishonest mining strategies by a reputation-based paradigm, in: SAI 2018: Intelligent Computing, 2018, pp. 1118–1134.

[66] K. Sunny, N. Scott, Ppcoin: Peer-to-peer Crypto-currency with Proof-of-stake, 2012. https://www.researchgate.net/publication/265116876_PPCoin_Peer-to-Peer_Crypto-Currency_with_Proof-of-Stake.

[67] S. Rakesh, Y. Seung, Regional blockchain for vehicular networks to prevent 51% attacks, IEEE Access 7 (7) (2019) 95021–95033.

[68] Y. Congcong, L. Guoqiang, C. Hongming, G. Yonggen, F. Akira, Analysis of security in blockchain: case study in 51%-attack detecting, in: 2018 5th International Conference on Dependable Systems and Their Applications (DSA), 2018, pp. 15–24.

[69] Y. Xinle, C. Yang, C. Xiaohu, Effective scheme against 51% attack on proof-of–work blockchain with history weighted information, in: 2019 IEEE International Conference on Blockchain (Blockchain), 2019, pp. 261–265.

[70] G. Alberto, S. Pier, V. Robert, S. Uri, A Penalty System for Delayed Block Submission, 2018. https://www.horizen.io/assets/files/A-Penalty-System-for-Delayed-Block-Submission-by-Horizen.pdf.

[71] B. Jaewon, L. Hyuk, Random mining group selection to prevent 51% attacks on bitcoin, in: 2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W), 2018, pp. 81–82.

[72] B. Jaewon, L. Hyuk, Protecting early stage proof-of-work based public blockchain, in: 2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W), 2018, pp. 122–127.

[73] S. Zhu, W. Li, H. Li, L. Tian, G. Luo, Z. Cai, Coin hopping attack in blockchain-based iot, IEEE Internet Things J. 6 (3) (2018) 4614–4626.

[74] M. Hu, J. Chen, W. Gan, C.-M. Chen, A jumping mining attack and solution, Applied Intelligence 9 (9) (2020) 1–12.

[75] M. Belotti, S. Kirati, S. Secci, Bitcoin pool-hopping detection, in: 2018 IEEE 4th International Forum on Research and Technology for Society and Industry (RTSI), 2018, pp. 1–6.

[76] S.K. Singh, M.M. Salim, M. Cho, J. Cha, Y. Pan, J.H. Park, Smart contract-based pool hopping attack prevention for blockchain networks, Symmetry (Basel) 11 (7) (2019) 1–19.

[77] D. Meshkov, A. Chepurnoy, M. Jansen, Short paper: revisiting difficulty control for blockchain systems, in: Data Privacy Management, Cryptocurrencies and Blockchain Technology, Springer, 2017, pp. 429–436.

[78] I. Eyal, E.G. Sirer, Majority is not enough: Bitcoin mining is vulnerable, in: International Conference on Financial Cryptography and Data Security, 2014, pp. 436–454.

[79] H. Azimy, A. Ghorbani, Competitive selfish mining, in: 2019 17th International Conference on Privacy, Security and Trust (PST), 2019, pp. 1–8.

[80] G. Yang, Y. Wang, Z. Wang, Y. Tian, X. Yu, S. Li, Ipbsm: an optimal bribery selfish mining in the presence of intelligent and pure attackers, Int. J. Intell. Syst. 35 (11) (2020) 1735–1748.

[81] M. Saad, L. Njilla, C. Kamhoua, A. Mohaisen, Countering selfish mining in blockchains, in: 2019 International Conference on Computing, Networking and Communications (ICNC), 2019, pp. 360–364.
[82] E. Heilman, One weird trick to stop selfish miners: Fresh bitcoins, a solution for the honest miner, in: International Conference on Financial Cryptography and Data Security, 2014, pp. 161–162.
[83] R. Zhang, B. Preneel, Publish or perish: A backward-compatible defense against selfish mining in bitcoin, in: Cryptographers' Track at the RSA Conference, 2017, pp. 277–292.
[84] Y. Kwon, D. Kim, Y. Son, E. Vasserman, Y. Kim, Be selfish and avoid dilemmas: Fork after withholding (FAW) attacks on bitcoin, in: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, 2017, pp. 195–209.
[85] S. Gao, Z. Li, Z. Peng, B. Xiao, Power adjusting and bribery racing: Novel mining attacks in the bitcoin system, in: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, 2019, pp. 833–850.
[86] J. Ke, H. Jiang, X. Song, S. Zhao, H. Wang, Q. Xu, Analysis on the block reward of fork after withholding (faw), in: International Conference on Network and System Security, 2018, pp. 16–31.
[87] S. Chang, Y. Park, Silent timestamping for blockchain mining pool security, in: 2019 International Conference on Computing, Networking and Communications (ICNC), 2019, pp. 1–5.
[88] A. Sarker, S. Wuthier, S. Chang, Anti-withholding reward system to secure blockchain mining pools, in: 2019 Crypto Valley Conference on Blockchain Technology (CVCBT), 2019, pp. 43–46.
[89] M. Vasek, M. Thornton, T. Moore, Empirical analysis of denial-of-service attacks in the bitcoin ecosystem, in: International Conference on Financial Cryptography and Data Security, 2014, pp. 57–71.
[90] M. Saad, L. Njilla, C. Kamhoua, J. Kim, D. Nyang, A. Mohaisen, Mempool optimization for defending against ddos attacks in pow-based blockchain systems, in: 2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), 2019, pp. 285–292.
[91] A. Abhishta, R. Joosten, S. Dragomiretskiy, L.J. Nieuwenhuis, Impact of successful ddos attacks on a major crypto-currency exchange, in: 2019 27th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP), 2019, pp. 379–384.
[92] P. Kumar, R. Kumar, G.P. Gupta, R. Tripathi, A distributed framework for detecting ddos attacks in smart contract-based blockchain-iot systems by leveraging fog computing, Transactions on Emerging Telecommunications Technologies 7 (7) (2020) 1–31.
[93] B. Jia, Y. Liang, Anti-d chain: a lightweight ddos attack detection scheme based on heterogeneous ensemble learning in blockchain, China Commun. 17 (9) (2020) 11–24.
[94] U. Baek, S. Ji, J. Park, M. Lee, J. Park, M. Kim, Ddos attack detection on bitcoin ecosystem using deep-learning, in: 2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS), 2019, pp. 1–4.
[95] G. Bissias, A.P. Ozisik, B.N. Levine, M. Liberatore, Sybil-resistant mixing for bitcoin, in: Proceedings of the 13th Workshop on Privacy in the Electronic Society, 2014, pp. 149–158.
[96] S. Zhang, J. Lee, Double-spending with a sybil attack in the bitcoin decentralized network, IEEE Trans. Ind. Inf. 15 (10) (2019) 5715–5722.
[97] A. Biryukov, D. Feher, Recon: sybil-resistant consensus from reputation, Pervasive Mob Comput 61 (1) (2020) 1–34.
[98] P. Otte, M. de Vos, J. Pouwelse, Trustchain: a sybil-resistant scalable blockchain, Future Generation Computer Systems 107 (6) (2020) 770–780.
[99] P. Swathi, C. Modi, D. Patel, Preventing sybil attack in blockchain using distributed behavior monitoring of miners, in: 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 2019, pp. 1–6.
[100] E. Heilman, A. Kendler, A. Zohar, S. Goldberg, Eclipse attacks on bitcoin's peer–to-peer network, in: 24th USENIX Security Symposium, 2015, pp. 129–144.
[101] A.E. Yveschristian, B. Hammi, A. Serhrouchni, H. Labiod, Total eclipse: How to completely isolate a bitcoin peer, in: 2018 Third International Conference on Security of Smart Cities, Industrial Control System and Communications (SSIC), 2018, pp. 1–7.
[102] H.S.K. M. Walck K. Wang, Endrilstaller: block delay attack in bitcoin, in: 2019 IEEE International Conference on Blockchain (Blockchain), 2019, pp. 1–9.
[103] E. Heilman, A. Kendler, A. Zohar, S. Goldberg, Eclipse attacks on bitcoin's peer-to-peer network, in: 24th USENIX Security Symposium (USENIX Security 15), 2015, pp. 129–144.
[104] Z. Jiang, C. Lv, B. Zhang, C. Zhang, W. Lu, S. Ji, Dynamic network configuration: An effective defensive protocol for public blockchain, in: International Conference on Security with Intelligent Computing and Big-data Services, 2018, pp. 398–413.
[105] G. Xu, B. Guo, C. Su, X. Zheng, K. Liang, D.S. Wong, H. Wang, Am i eclipsed? a smart detector of eclipse attacks for ethereum, Computers & Security 88 (1) (2020) 1–10.
[106] N. Atzei, M. Bartoletti, T. Cimoli, A survey of attacks on ethereum smart contracts (sok), in: International Conference on Principles of Security and Trust, 2017, pp. 164–186.
[107] X. Zhao, Z. Chen, X. Chen, Y. Wang, C. Tang, The dao attack paradoxes in propositional logic, in: 2017 4th International Conference on Systems and Informatics (ICSAI), 2017, pp. 1743–1746.
[108] N.F. Samreen, M.H. Alalfi, Reentrancy vulnerability identification in ethereum smart contracts, in: 2020 IEEE International Workshop on Blockchain Oriented Software Engineering (IWBOSE), 2020, pp. 22–29.
[109] C. Liu, H. Liu, Z. Cao, Z. Chen, B. Chen, B. Roscoe, Reguard: finding reentrancy bugs in smart contracts, in: 2018 IEEE/ACM 40th International Conference on Software Engineering: Companion (ICSE-Companion), 2018, pp. 65–68.
[110] Z. Nehai, P.-Y. Piriou, F. Daumas, Model-checking of smart contracts, in: 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), 2018, pp. 980–987.
[111] J. Gao, H. Liu, C. Liu, Q. Li, Z. Guan, Z. Chen, Easyflow: Keep ethereum away from overflow, in: 2019 IEEE/ACM 41st International Conference on Software Engineering: Companion Proceedings (ICSE-Companion), 2019, pp. 23–26.
[112] S. Sayeed, H. Marco-Gisbert, T. Caira, Smart contract: attacks and protections, IEEE Access 8 (8) (2020) 24416–24427.
[113] J.-W. Liao, T.-T. Tsai, C.-K. He, C.-W. Tien, Soliaudit: Smart contract vulnerability assessment based on machine learning and fuzz testing, in: 2019 Sixth International Conference on Internet of Things: Systems, Management and Security (IOTSMS), 2019, pp. 458–465.
[114] J. Kongmanee, P. Kijsanayothin, R. Hewett, Securing smart contracts in blockchain, in: 2019 34th IEEE/ACM International Conference on Automated Software Engineering Workshop (ASEW), 2019, pp. 69–76.
[115] M.A. Ferrag, M. Derdour, M. Mukherjee, A. Derhab, L. Maglaras, H. Janicke, Blockchain technologies for the internet of things: research issues and challenges, IEEE Internet Things J. 6 (2) (2018) 2188–2204.
[116] M. Spagnuolo, F. Maggi, S. Zanero, Bitiodine: Extracting intelligence from the bitcoin network, in: International conference on financial cryptography and data security, 2014, pp. 457–468.
[117] P. Koshy, D. Koshy, P. McDaniel, An analysis of anonymity in bitcoin using p2p network traffic, in: International Conference on Financial Cryptography and Data Security, 2014, pp. 469–485.
[118] D. Ron, A. Shamir, Quantitative analysis of the full bitcoin transaction graph, in: International Conference on Financial Cryptography and Data Security, 2013, pp. 6–24.
[119] E. Androulaki, G.O. Karame, M. Roeschlin, T. Scherer, S. Capkun, Evaluating user privacy in bitcoin, in: International Conference on Financial Cryptography and Data Security, 2013, pp. 34–51.
[120] A. Dorri, M. Steger, S.S. Kanhere, R. Jurdak, Blockchain: a distributed solution to automotive security and privacy, IEEE Commun. Mag. 55 (12) (2017) 119–125.
[121] A. Biryukov, D. Khovratovich, I. Pustogarov, Deanonymisation of clients in bitcoin p2p network, in: Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, 2014, pp. 15–29.
[122] D.L. Chaum, Untraceable electronic mail, return addresses, and digital pseudonyms, Commun ACM 24 (2) (1981) 84–90.
[123] L. Valenta, B. Rowan, Blindcoin: Blinded, accountable mixes for bitcoin, in: International Conference on Financial Cryptography and Data Security, 2015, pp. 112–126.
[124] M. Tran, L. Luu, M.S. Kang, I. Bentov, P. Saxena, Obscuro: A bitcoin mixer using trusted execution environments, in: Proceedings of the 34th Annual Computer Security Applications Conference, 2018, pp. 692–701.
[125] E. Duffield, D. Diaz, Dash: A Privacycentric Cryptocurrency, 2015. https://whitepaperdatabase.com/wp-content/uploads/2017/09/Dash-Whitepaper.pdf.
[126] A.A. Maksutov, M.S. Alexeev, N.O. Fedorova, D.A. Andreev, Detection of blockchain transactions used in blockchain mixer of coin join type, in: 2019 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus), 2019, pp. 274–277.
[127] S. Goldwasser, S. Micali, C. Rackoff, The knowledge complexity of interactive proof systems, SIAM J. Comput. 18 (1) (1989) 186–208.
[128] S. Hu, C. Cai, Q. Wang, C. Wang, X. Luo, K. Ren, Searching an encrypted cloud meets blockchain: A decentralized, reliable and fair realization, in: IEEE INFOCOM 2018-IEEE Conference on Computer Communications, 2018, pp. 792–800.
[129] C.D. Pop, M. Antal, T. Cioara, I. Anghel, I. Salomie, Blockchain and demand response: zero-knowledge proofs for energy transactions privacy, Sensors 20 (19) (2020) 1–21.
[130] I. Miers, C. Garman, M. Green, A.D. Rubin, Zerocoin: anonymous distributed e–cash from bitcoin, in: 2013 IEEE Symposium on Security and Privacy, 2013, pp. 397–411.
[131] E.B. Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, M. Virza, Zerocash: Decentralized anonymous payments from bitcoin, in: 2014 IEEE Symposium on Security and Privacy, 2014, pp. 459–474.
[132] S.Z. Team, Super Zero(sero) Technical White Paper, 2019. https://www.chainwhy.com/whitepaper/serowhitepaper2.html.
[133] R.L. Rivest, A. Shamir, Y. Tauman, How to leak a secret, in: International Conference on the Theory and Application of Cryptology and Information Security, 2001, pp. 552–565.
[134] G. Mwitende, Y. Ye, I. Ali, F. Li, Certificateless authenticated key agreement for blockchain-based wbans, J. Syst. Archit. 110 (11) (2020) 1–31.
[135] X. Li, Y. Mei, J. Gong, F. Xiang, Z. Sun, A blockchain privacy protection scheme based on ring signature, IEEE Access 8 (8) (2020) 76765–76772.
[136] F. Li, Z. Liu, T. Li, H. Ju, H. Wang, H. Zhou, Privacy-aware PKI model with strong forward security, Int. J. Intell. Syst. 8 (8) (2020) 1–17.
[137] S. Noether, S. Noether, Mmonero Is Not That Mysterious, 2014. https://web.getmonero.org/ru/resources/research-lab/pubs/MRL-0003.pdf.
[138] N.S. Mackenzie A, Monero Core Team: Improving Obfuscation in the Cryptonote Protocol, 2015. https://cryptochainuni.com/wp-content/uploads/Monero-Improving-Obfuscation-in-the-CryptoNote-Protocol.pdf.

[139] A. Joshi, M. Han, Y. Wang, A survey on security and privacy issues of blockchain technology, Mathematical Foundations of Computing 1 (2) (2018) 121–147.
[140] M. Han, Z. Li, J. He, D. Wu, Y. Xie, A. Baba, A novel blockchain-based education records verification solution, Proceedings of the 19th Annual SIG Conference on Information Technology Education (2018) 178–183.
[141] Y. Zhou, M. Han, L. Liu, Y. Wang, Y. Liang, L. Tian, Improving iot services in smart-home using blockchain smart contract, 2018 IEEE International Conference on Internet of Things (iThings) (2018) 81–87.
[142] L. Liu, M. Han, Y. Zhou, R. Parizi, E2C-chain: a two-stage incentive education employment and skill certification blockchain, The 2nd IEEE International Conference on Blockchain (Blockchain-2019) (2019) 140–147.
[143] S. Cao, W. Lin, M. Han, Q. Hou, B. Yang, Blockchain architecture for auditing automation and trust building in public markets, Computer 53 (7) (2020) 20–28.