# On Identity, Transaction, and Smart Contract Privacy on Permissioned and Permissionless Blockchain: A Comprehensive Survey

WEI LIANG, School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan, China, College of Computer Science and Electronic Engineering, Hunan University, Changsha, China, and Hunan Key Laboratory for Service Computing and Novel Software Technology, Xiangtan, China
YAQIN LIU, College of Computer Science and Electronic Engineering, Hunan University, Changsha, China
CE YANG, School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan, China and Hunan Key Laboratory for Service Computing and Novel Software Technology, Xiangtan, China
SONGYOU XIE, College of Computer Science and Electronic Engineering, Hunan University, Changsha, China
KUANCHING LI, School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan, China and Hunan Key Laboratory for Service Computing and Novel Software Technology, Xiangtan, China
WILLY SUSILO, Institute of Cybersecurity and Cryptology (iC2), University of Wollongong, Wollongong, Australia

Blockchain is a decentralized distributed ledger that combines multiple technologies, including chain data structures, P2P networks, consensus algorithms, cryptography, and smart contracts. This gives the blockchain the characteristics of decentralization, immutability, and traceability. However, blockchain stores smart contracts and transactions in blocks publicly, which poses the risk of data leakage and misuse. For example, by mining and analyzing blockchain transaction information, attackers can correlate transactions with user information, resulting in the disclosure of user privacy. Many current reviews focus on the privacy of permissionless blockchains or cryptocurrencies, requiring more in-depth investigations and detailed categorical analysis. To fill this gap, this work comprehensively reviews the latest and traditional methods related to iden-

This work was partially supported by the National Key Research and Development Program of China under grant 2021YFA1000600 and the National Natural Science Foundation of China under grant 62072170.
Authors' Contact Information: Wei Liang, School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan, China and Hunan Key Laboratory for Service Computing and Novel Software Technology, Xiangtan, China; e-mail: wliang@hnust.edu.cn; Yaqin Liu, College of Computer Science and Electronic Engineering, Hunan University, Changsha, China; e-mail: liuyaqin963@hnu.edu.cn; Ce Yang, School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan, China and Hunan Key Laboratory for Service Computing and Novel Software Technology, Xiangtan, China; e-mail: yangce@hnust.edu.cn; Songyou Xie, College of Computer Science and Electronic Engineering, Hunan University, Changsha, China; e-mail: syxie@hnu.edu.cn; Kuanching Li, School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan, China and Hunan Key Laboratory for Service Computing and Novel Software Technology, Xiangtan, China; e-mail: aliric@hnust.edu.cn; Willy Susilo, Institute of Cybersecurity and Cryptology (iC2), University of Wollongong, Wollongong, New South Wales, Australia; e-mail: susilo@uow.edu.au.

ACM 0360-0300/2024/07-ART298
https://doi.org/10.1145/3676164

tity, transaction, and smart contract privacy within permissioned and permissionless blockchains. Additionally, we summarize the existing problems, threats, and challenges of data management in different blockchain architectures. Last, we discuss future research directions for blockchain privacy protection technology, which can offer feasible ideas for researchers to explore further.

CCS Concepts: • **Security and privacy** → **Network security**; *Database and storage security*; *Information accountability and usage control;*

Additional Key Words and Phrases: Blockchain privacy protection, transaction privacy, identity privacy, smart contract privacy

**ACM Reference Format:**
Wei Liang, Yaqin Liu, Ce Yang, Songyou Xie, Kuanching Li, and Willy Susilo. 2024. On Identity, Transaction, and Smart Contract Privacy on Permissioned and Permissionless Blockchain: A Comprehensive Survey. *ACM Comput. Surv.* 56, 12, Article 298 (July 2024), 35 pages. https://doi.org/10.1145/3676164

## 1 Introduction

Nakamoto [74] proposed a decentralized digital currency system that keeps transaction records in a distributed database (i.e., blockchain) and employs a **proof-of-work (PoW)** mechanism to consensus transaction data. Blockchain has the characteristics of decentralization and traceability, which quickly attracted the attention of academia and industry. The progress of the blockchain can be summarized as four stages: (1) Blockchain 1.0 refers to the era of cryptocurrencies represented by Bitcoin, which has a decentralized transaction payment function that does not rely on third parties; (2) Blockchain 2.0 supports the development and deployment of smart contracts, which can realize a variety of complex transaction logic, making blockchain more widely used in the financial field; (3) Blockchain 3.0 achieves stronger logical capabilities through a decentralized application, which can solve the problem of mutual trust and share of data in various scenarios and can be applied to various fields; and (4) Blockchain 4.0 is a term for blockchain solutions that establish data trust mechanisms for the digital processes of multiparty enterprises, ensuring data security under different needs. Furthermore, it introduces a permissioned blockchain to build trust between enterprises. Compared to centralized applications, a trusted channel can be established between the blockchain nodes without the participation of a third-party organization, which can be widely used in many decentralized scenarios, such as insurance claims, supply chain traceability, and mobile crowdsourcing.

With the widespread application of blockchain, the privacy security threats faced by blockchain are becoming increasingly prominent [62]. Due to the ledger transparency of blockchain, the transaction addresses and amounts are stored on the chain, and any user can obtain this information through public channels [41]. Attackers can generate transaction correlation graphs by studying user behavior and breaking transaction anonymity. For example, remote side-channel attacks [107] can be used to detect transaction amounts and user IP addresses in encrypted currency systems, which can cause the leakage of private information. In Ethereum, attackers can analyze ledger information through address clustering technology, then obtain the identity relationship between anonymous accounts and real users [25, 59]. From the perspective of privacy protection, Cheng et al. [27] described the security threats of blockchain, and put forward future research directions on consensus, incentive mechanisms, privacy preservation, and encryption algorithms to further enhance the security and privacy of blockchain-based multimedia. Especially in scenarios such as mobile crowdsourcing and the Internet of Things, transaction records uploaded by users may contain sensitive information such as the user's location. The transparency of blockchain may also lead to competitive companies or individuals benefiting from analyzing transaction data to

Table 1. Comparison with Existing Surveys

| Literature | Year | Identity Privacy | Transaction Privacy | Smart Contract Privacy | Permissionless Blockchain | Permissioned Blockchain |
|---|---|---|---|---|---|---|
| Ours | 2024 | ✔ | ✔ | ✔ | ✔ | ✔ |
| [26] | 2022 | ✔ | ✔ | ✖ | ✖ | ✖ |
| [83] | 2021 | ✖ | ✔ | ✔ | ✔ | ✖ |
| [113] | 2020 | ✔ | ✔ | ✖ | ✔ | ✔ |
| [36] | 2019 | ✔ | ✔ | ✖ | ✔ | ✖ |

obtain users' sensitive information (e.g., user habits). These privacy risks greatly limit the practical application of blockchain technology.

However, preservation of privacy on the blockchain is not trivial. Compared to traditional centralized systems, blockchain is an open and decentralized system that does not have integrated system maintenance and privacy protection mechanisms. Currently, there are many encryption-based schemes that aim to achieve privacy protection. However, Bitcoin only supports simple script operations. Although Ethereum's built-in Turing-complete smart contracts can support complex cryptographic operations, they often require large amounts of computing resources, resulting in a significant increase in gas costs. Additionally, computationally expensive cryptographic operations can impact a blockchain's throughput, reducing its performance and availability. In recent years, some research works have aimed to transfer some transactions off-chain to solve privacy protection issues. However, how to ensure safe interactions on and off the chain and correct updates of off-chain transactions are also issues that need to be solved. Therefore, it is a challenging task to achieve practical privacy protection in blockchain systems.

There are currently many works that have reviewed the privacy and security issues of blockchain. Zhang et al. [128] reviewed the security and privacy of techniques, including representative consensus algorithms, hash-chained storage, mixing protocols, anonymous signatures, and **non-interactive zero-knowledge (NIZK)** proof. Despite the lack of smart contract analysis, Feng et al. [36] analyzed blockchain privacy threats and existing cryptographic defense mechanisms. Wang et al. [113] surveyed some existing solutions to current user identity and transaction privacy protection problems and divided technology applications into a coin mixing protocol, an encryption protocol, and a secure channel protocol, but there is an insufficient description of smart contract privacy protection. Peng et al. [83] conducted a comprehensive unified standard classification of privacy protection for permissionless blockchains, which lacks research on privacy protection for permissioned blockchains. Chen et al. [26] investigated the types, principles, and impacts of attacks and analyzed various defensive measures adopted by blockchain. However, Chen et al. [26] only mentioned the protection methods of identity and transaction privacy but did not discuss the privacy security of smart contracts. Under such a condition, most research reviews focus only on the privacy security of the permissionless blockchain and lack a detailed description of the privacy security of the permissioned blockchain. The comparison of the preceding results is depicted in Table 1.

Much literature has been published on blockchain privacy protection. This survey aims to provide a comprehensive introduction to blockchain privacy protection, with a particular focus on state-of-the-art developments. We selected ACM Digital Library, Google Scholar, IEEE Xplore, DBLP Bibliography, SpringerLink, Science Direct, and Web of Science and limited the search year range from 2016 to 2024 (some classic schemes were added during the research process). This work sorts out the data privacy problems encountered on the blockchain, indicates the definition and classification of privacy protection, and introduces privacy and security issues from the aspects of

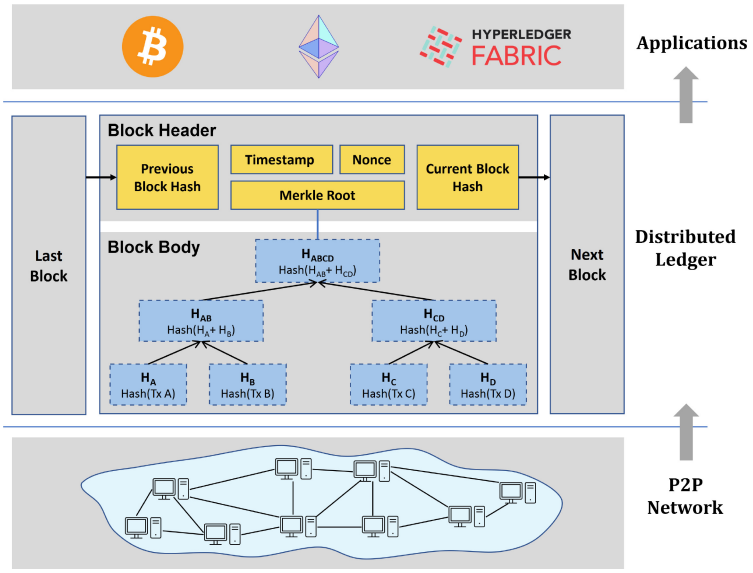


Fig. 1. Blockchain technology framework.

identity, transaction, and smart contract privacy. Specifically, the contributions of this work are summarized as follows:

— To divide the privacy security of permissioned and permissionless blockchains into three categories: identity, transaction, and smart contract privacy, based on privacy concerns and blockchain architecture, then perform a systematic review based on classification.
— To highlight the key technologies for blockchain privacy protection, including advanced execution mechanisms, cryptography techniques, and **Trusted Execution Environments (TEEs)**.
— To introduce different solutions under the three categories and analyze the advantages and disadvantages of these solutions. Implementing secure privacy protection on the blockchain is challenging due to the public availability of on-chain data, so we will sum up and analyze state-of-the-art solutions.
— To summarize the main approaches taken in different situations based on privacy threats and architecture in blockchain, providing next constructive recommendations and insightful observations for future work.

The rest of this work is organized as follows. Section 2 introduces an overview of blockchain. In Section 3, the privacy requirement and threats in blockchain are presented. In Section 4, the methodology to protect identity, transactions, and smart contract privacy is described. Future directions are discussed in Section 5, and concluding remarks are presented in Section 6.

## 2 Overview of Blockchain

Blockchain is a distributed ledger technology in which transaction data is replicated and stored in multiple nodes. Data blocks are combined into a chained data structure in chronological order, and each data block is linked to the next block through a hash algorithm to ensure that the data cannot be tampered with. At the same time, the node uses consensus algorithms to confirm the generation of new blocks, applies cryptography to ensure the security of information transmission and access, and uses script codes or smart contracts to process complex transaction logic. As shown in Figure 1,

the nodes communicate through a P2P network to ensure the consistency of the data records in each node. The data records in each node constitute the distributed ledger, and many decentralized applications can be built on this distributed ledger. There are two main accounting models on the blockchain, the **Unspent Transaction Output (UTXO)** and the account model. Bitcoin uses the UTXO model, in which the input of each transaction needs to indicate the unspent output from the previous transaction. Each output contains a specific output script, and only the address with the correct key can spend this output. The account model adopted by Ethereum creates a global state for each account that holds funds, and each account has its balance. Each transaction is interpreted and executed by the virtual machine in the network, and corresponding state changes are made to all accounts in the global state. This section will introduce basic technologies, including consensus algorithms, smart contracts, and cryptography, and then illustrate the classification of the blockchain.

To ensure the consistency and reliability of the information in a distributed architecture, the blockchain adopts a decentralized consensus mechanism. In the blockchain system, there are mainly four typical consensus mechanisms: PoW [74], **proof of stake (PoS)** [99], **delegated proof of stake (DPoS)** [125], and **practical Byzantine fault tolerance (PBFT)** [22]. In PoW, each node needs to solve a difficult but easy-to-verify mathematical problem, and the node that takes the least time to solve the problem gets the right to make the next block and can get a certain amount of virtual currency as a reward. PoS determines the next block's generation right and reward distribution through the amount and time of the currency the node holds. Compared to PoW, PoS can reduce the waste of computing power resources and prevent miners from continuously computing difficult problems. DPoS is an improved version of PoS. Unlike PoS, DPoS allows token holders to vote for a certain number of representatives to generate new blocks rather than having all token holders participate in this process. DPoS can solve the problem of centralizing rights in PoS to some extent. PBFT reaches a consensus by using a master node and multiple backup nodes to ensure that the system's security and consistency can be guaranteed even if some nodes fail or are attacked.

A smart contract is a program stored on the blockchain that can automatically execute and verify transactions, making the transaction process more transparent, safe, and reliable. When a transaction is initiated, the smart contract will automatically verify the legality and validity of the transaction according to the pre-set rules and conditions. If the transaction complies with the rules, the smart contract will automatically execute the transaction, making the transaction process faster, more efficient, and safer. The implementation of smart contracts can use a variety of programming languages (e.g., Solidity, Go) and needs to be deployed to different blockchain platforms after writing [63].

Compared to traditional centralized databases, the blockchain significantly enhances security against malicious tampering, forgery, and **denial-of-service (DoS)** attacks by leveraging cryptographic techniques such as hash functions with collision resistance and digital signatures for anti-counterfeiting authentication. Blockchain can greatly improve data security by increasing the difficulty and cost of tampering with data. Consequently, the blockchain incorporates cryptographic techniques, including hash algorithms, encryption, and decryption algorithms, to ensure data integrity and confidentiality [61].

According to the permission access method of the nodes, the blockchain can be divided into a permissionless blockchain and permissioned blockchain. Permissionless blockchains (i.e., public blockchains) are open networks wherein anybody can participate in the consensus process for verifying transactions and data. A permissioned blockchain is a system in which each node has been permitted to participate and nodes without authorization are unable to access the system. Permissioned blockchains are made up of both private and consortium blockchains in general. The main differences are shown next:

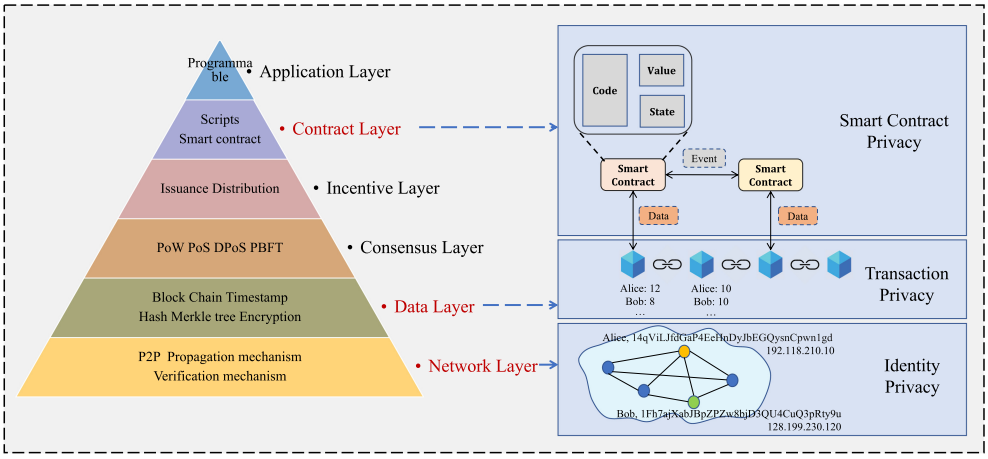


Fig. 2. Blockchain architecture and privacy classification basis.

(1) *Permissionless blockchain:* Permissionless blockchain has no centralized management organization, and any node can join the blockchain and participate in transactions. All transactions are open and transparent; anyone can view and verify transactions. Since access to nodes is not restricted, maintaining ledgers requires high calculations and time costs. It is usually used in public domain applications such as Bitcoin and Ethereum.
(2) *Permissioned blockchain:* Permissioned blockchain requires users to be authenticated and authorized to become a blockchain node and participate in transactions. The permissioned blockchain is classified into a consortium blockchain and a private blockchain. The former is built and maintained by enterprises or organizations, such as supply chain management and transactions between financial institutions, whereas the latter is used by one participant to record personal private data.

## 3 Privacy Classification and Challenges

As seen in the previous section, blockchain and traditional databases (including centralized and distributed databases) differ in several ways, such as data storage and management methods, and many privacy protection solutions designed for traditional databases may not be suitable for blockchain. In traditional centralized or distributed databases, problems such as single points of failure and data tampering are prone to occur, and there is also a lack of effective means to trace and review data. In blockchain, data are stored on distributed nodes and the authenticity and integrity of the data are guaranteed through consensus algorithms. However, there are gaps in computing performance among blockchain nodes, making some nodes vulnerable. In this section, we will detail the classification of privacy and the challenges of blockchain.

### 3.1 Privacy Classification

To reach a consensus on data, personal information, transaction content, and smart contracts are public to all nodes on the blockchain, so there will be problems with data leakage and privacy abuse. As shown in Figure 2, we divide blockchain privacy into three categories, namely identity, transaction, and smart contract privacy, which correspond to the network, data, and contract layers of the blockchain architecture:

(1) *Identity privacy:* The relationship between user identity and blockchain addresses is referred to as identity privacy. The blockchain address, the pseudonym of users in the blockchain, is

commonly used as the transaction's input or output account. The blockchain address does not contain any user's identity information. However, when users use blockchain addresses to participate in blockchain commerce, sensitive information such as the IP address or the transaction transmission path at the network layer may be leaked, which might be used to speculate on the real identity of blockchain addresses.

(2) *Transaction privacy:* Transaction content is often exposed in cryptocurrencies, including information such as the transaction amount and the address of the sender and receiver of the transaction, which can reveal sensitive information about users.

(3) *Smart contract privacy:* A smart contract is a public blockchain program code that can be enforced when pre-set conditions are met. Because the information for function calls in the contract is public, attackers can obtain sensitive information by analyzing the contents of each transaction. Smart contract privacy is divided into contract code privacy and contract data privacy [47]. Contract code privacy mainly refers to code and state variable privacy. Contract data privacy includes mostly participants and transaction information privacy.

Identity, transaction, and smart contracts contain sensitive user information and must be protected. Because data stored on the blockchain are difficult to delete or tamper with, even if users discover that address or transaction data have been compromised, there is no remedial action. Therefore, blockchain systems should pay more attention to privacy issues and improve privacy protection capabilities.

### 3.2 Research Challenges

In a permissioned blockchain, the access rules of private data can be set to realize the data access mode based on user node permissions. However, in a permissionless blockchain, a large amount of information will be stored on the chain in a public form. Implementing privacy-preserving data sharing on the blockchain poses many challenges. In the following, we describe these challenges in detail.

#### *3.2.1 Challenges to Identity Privacy.*

(1) *Transaction address analysis:* Each user in the blockchain can generate multiple different addresses, and each transaction requires the sender's signature, so multiple inputs for a transaction may come from the same entity. In the UTXO model, the user will set a change address in the transaction to obtain the remaining coins. This means that the inputs of a transaction are associated with at least one of the outputs in some way. These approaches are analyzed based on the transaction generation designed into the blockchain itself, and solving these challenges requires modifications to the underlying logic of the blockchain.

(2) *Routing analysis:* Blockchain nodes typically must interact in the network through a web browser, whereas web application processes transmit data over the HTTP protocol. For convenience, the user may choose the browser to remember the password when making a transaction, and the associated cookie will be saved on the client side. Hackers can intercept data as it is transmitted to an internet service provider. The threat is usually invisible to blockchain participants in a routing attack, so everything looks normal. However, behind the scenes, fraudsters steal confidential data or cryptocurrency.

(3) *Network analysis:* The blockchain nodes broadcast transactions among themselves via a P2P network, meaning that each submitted transaction corresponds to an IP address. Looking at the pseudonymous address of the user's blockchain and the IP address corresponding to the initiated transaction, although the user can generate multiple random anonymous addresses, an attacker may link the address with a wallet and a real user [12]. To address this threat, users can hide the IP address of the transaction sender in the network layer so that it is difficult for an attacker to link a pseudonym with an IP address.

(4) *DoS attacks:* DoS attacks are an attempt by malicious attackers to make a machine or resource unavailable to a client by interrupting a normal user's connection to the network. In Bitcoin's P2P network, networks such as onion routing (Tor) are usually used to ensure anonymity. However, an attacker could use a DoS attack to disable many nodes in the Tor network, thereby disconnecting the Tor exit node from the Bitcoin network. The attacker controls the flow of information between Bitcoin users in the Tor network and tracks the node's IP address when it broadcasts the transaction [13].
(5) *Sybil attacks:* Sybil attacks are malicious attackers who generate many pseudonymous identities to take control of a blockchain network and undermine the anonymity of the blockchain. Bissias et al. [14] proposed that Sybil attacks can lead to identity privacy leaks. For example, in the coin mixing services, as the size of the Sybil node increases, the chances of parties, including attackers, to select the coin mixing service node will also increase. In a protocol that mixes pairwise, a Sybil attacker will know the destination of funds in any address that is paired with it. At the same time, some cryptocurrencies (e.g., CoinShuffle) do not charge nodes to participate in the mixing process, so the cost of launching a Sybil attack in the blockchain system is meager. To protect anonymity, users must design transaction schemes that are resistant to Sybil attacks.

#### *3.2.2 Challenges to Transaction Privacy.*

(1) *Cross-privilege access:* For a permissioned blockchain, different users have different access rights, and if the transaction ledger information is accessed by unauthorized nodes, privacy leakage is also prone to occur. For such problems, implementing a secure and efficient user fine-grained access control scheme is one of the challenges of transaction privacy.
(2) *Leakage of information:* No matter whether it is a permissionless or permissioned blockchain, verification nodes are needed to verify the transactions, but some sensitive information about the users' transactions can be obtained in the process of verifying the transactions. Therefore, ensuring the accuracy of the user's normal verification transactions without revealing the transaction information itself is also a major challenge for transaction privacy.
(3) *Transaction graph analysis:* Any user can perform data analysis on historical transactions to obtain some private information, which can reveal some information about the user's transaction behavior [60] and so on. Chen et al. [25] constructed the graph of token creators, holders, and token transfers by downloading the entire blockchain and parsing transaction records and event logs. Through further analysis of these graphs, potential relationships between tokens and other accounts can be discovered, such as two users conducting frequent transactions within a period or a user trading with himself multiple times, revealing the user's transaction privacy information. Wu et al. [117] modeled blockchain transaction records as transaction graphs and processed blockchain transaction tracking as a graph search task, achieving efficient tracking of fund transfers. Therefore, designing effective cryptographic algorithms to ensure that correlations between transactions are not analyzed is also a significant challenge.
(4) *Post-quantum computing resistance:* With the upcoming quantum computing, some PoW algorithms and signatures are at some level of risk. Quantum algorithms can break several basic blockchain encryption algorithms used for signatures in the future, such as ECDSA (Elliptic Curve Public Key encryption) or large integer decomposition problems. To mitigate this problem, a randomization process must be performed before the data can be chained, and, in addition, some cryptographic algorithms that are resistant to quantum attacks must also be investigated.

(5) *Cryptocurrency theft:* Cryptocurrency theft is one of the major threats facing cryptocurrency systems, stealing users' funds primarily through password theft, system intrusion, fraud, and so forth. Due to the decentralized and immutable characteristics of cryptocurrencies, it is difficult to deal with all means of theft and fraud successfully. If cybercriminals find and exploit vulnerabilities in the underlying protocols, it will lead to a serious loss of funds in the system. Therefore, strengthening the system's security and adopting a multilayered approach to protecting personal assets can help reduce the risk of theft in currency transactions.

*3.2.3 Challenges to Smart Contract Privacy.*

(1) *The leakage of smart contract code:* Smart contract code privacy includes contract code, state variables, functional privacy, and so on. To protect the privacy of smart contract code from being acquired and analyzed by attackers, how to ensure that the code privacy is not leaked without affecting the realization of the decentralized transaction function is one of the main challenges facing the privacy of smart contracts.
(2) *The leakage of smart contract data:* Smart contract data privacy includes input and output privacy and transaction content privacy. How to ensure the correct execution of smart contracts while protecting users' private information is another major challenge facing the privacy of smart contracts.
(3) *Oracle security:* Oracles are used to provide smart contracts with real-time data sources outside the blockchain. In essence, they submit external data to smart contracts by submitting transactions. However, data confidentiality caused by internal third-party programs, such as oracles, will also become a considerable obstacle to transferring owned data to the blockchain [81]. Research on Oracle security is certainly important, but this work mainly discusses the privacy protection issues of blockchain. Security issues such as data reliability caused by introducing oracles are not within our main scope of discussion.

Next, for identity, transaction, and smart contract privacy protection challenges, we will summarize the relevant privacy protection approaches for permissionless and permissioned blockchain, respectively.

## 4 Solutions on Blockchain Privacy Protection

In this section, we detail each privacy protection method in three categories: identity, transaction, and smart contract privacy. It should be noted that these categories are not mutually exclusive, and some schemes may use multiple categories of privacy protection technologies simultaneously. For example, Monero [77] not only applies ring signature technology to hide transaction information (i.e., transaction privacy) but also adopts Tor to hide IP addresses (i.e., identity privacy). However, based on its main novel idea, we placed a method in a single category.

### 4.1 Identity Privacy Protection

Permissionless and permissioned blockchains differ in identity management; therefore, their implementations differ. We summarize the identity privacy protection methods and compare them in seven aspects: year, scenario, anonymity, blockchain type, technology, validated, and approach, as shown in Table 2.

In the permissionless blockchain, an anonymous identity authentication mechanism is generally used, the user's public key is used as the identification, and the user can generate and manage the secret key by himself. In the permissioned blockchain, more emphasis is placed on the real name or controlled anonymous identity authentication mechanism. The user is identified by a digital certificate, which must be issued by a certificate authority. According to issues and challenges of identity privacy mentioned in Section 3.2.1, several researchers have focused on designing secure

Table 2. Comparison of Identity Privacy Protection in Blockchain: Anonymity (A), Blockchain Type (BT), and Validated (V)

| Literature | Year | Scenario | A | BT | Technology | V | Approach |
|---|---|---|---|---|---|---|---|
| BlindHub [86] | 2023 | Bitcoin | Yes | Permissionless | BAS and FBCS | Yes | Bi-directional payment channel protocol BlindChannel and three-party payment protocol BlindHub |
| Accio [38] | 2023 | Ethereum | Yes | Permissionless | RSUC | Yes | Utilizing RSUC allows the hub to update the hidden state correctly |
| LedgerView [93] | 2022 | Supply chain | No | Permissioned | Encryption and hash | Yes | Offer different access-control views for different roles |
| Ni et al. [76] | 2022 | Generic | Yes | Permissionless | Mixing | Yes | Find the minimum number of decomposed outputs for a given original output set |
| Tan et al. [105] | 2021 | Medical | No | Permissioned | CP-ABE | Yes | Propose a traceable and direct revocation scheme for COVID-19 medical records |
| PrivySharing [69] | 2020 | Smart city | No | Permissioned | ZKP | Yes | Divide various channels for different data sharing |
| Tornado.Cash [1] | 2019 | Ethereum | Yes | Permissionless | ZKP | Yes | The user deposits money into the smart contract and uses ZKP to withdraw the deposit to a new address |
| MatRiCT [35] | 2019 | Generic | Yes | Permissionless | Post-quantum RingCT | Yes | Improve ring signature, propose a novel post-quantum RingCT and extractable commitment |
| Aurora [9] | 2019 | Cryptocurrency | Yes | Permissionless | zk-STARKs | Yes | Reduce the messages proof overhead |
| MixEth [97] | 2019 | Ethereum | Yes | Permissionless | Verifiable shuffles | Yes | Participants shuffle the public key on-chain |
| Sarfraz et al. [96] | 2019 | IOTA | Yes | Permissionless | Decryption mixnets and multi-signatures | Yes | Each transaction needs to validate two previous transactions |
| FISCO BCOS [119] | 2018 | Generic | No | Permissioned | Group/ring signature | Yes | Use ring group signatures to form transactions of different entities |
| Liu et al. [67] | 2018 | Bitcoin | Yes | Permissionless | Ring signature | Yes | The mixing server only checks whether the output addresses belong to its customers |
| Corda [92] | 2017 | Generic | No | Permissioned | Signature | Yes | Use multiple ephemeral anonymous keys to initiate a transaction |
| Hyperledger Fabric [2] | 2017 | Generic | No | Permissioned | ZKP | Yes | User registration and transactions use different certificates |
| Monero [77] | 2015 | Cryptocurrency | Yes | Permissionless | RingCT, Tor | Yes | Implement a one-time public-private key pair to hide the transaction addresses; use Tor to hide the IP address of the node |
| Blindcoin [109] | 2015 | Cryptocurrency | Yes | Permissionless | Blind signature | Yes | Use a blind signature to protect the mixing process |
| CoinParty [130] | 2015 | Bitcoin | Yes | Permissionless | SMPC | Yes | Deploy a combination of decryption mixnets with threshold signatures |
| Dash [32] | 2015 | Cryptocurrency | Yes | Permissionless | Multi-center mixing | Yes | A user randomly select multiple master nodes; the master node must pay 1,000 Dash as a deposit to provide coin mixing services |
| Zerocash [8] | 2014 | Cryptocurrency | Yes | Permissionless | NIZK | Yes | Adopt NIZK to hide the sender, receiver addresses, and transaction amount |
| Mixcoin [17] | 2014 | Cryptocurrency | No | Permissionless | Hybrid center mixing | Yes | Each coin mixing transaction requires the signature of the central server |
| CoinShuffle [94] | 2014 | Bitcoin | Yes | Permissionless | Decrypt the hybrid network | Yes | Uses multi-level encryption to hide the address association of input and output |
| CoinJoin [70] | 2013 | Bitcoin | No | Permissionless | Multi-round signature | No | Users must negotiate the coin mixing process themselves |
| Zerocoin [72] | 2013 | Bitcoin | Yes | Permissionless | ZKP | Yes | Users add assets to the cryptographic accumulator, proving they have unspent assets to spend it |

mixing services, cryptographic methods, network information hiding, and certificate management. Next, we introduce identity privacy protection in two categories: permissionless and permissioned blockchain.

#### 4.1.1 *Permissionless Blockchain Based Approaches.*

*Mixing Service Approaches.* Transactions on the blockchain are public to all users, so attackers can easily deduce the implicit information of each transaction based on the transaction content. The main idea of the coin mixing strategy is to obfuscate the input and output of the transaction, to hide the true flow of the currency, and prevent the attacker from analyzing the correlation between the transaction addresses. Depending on whether mixed transaction operations need to be performed through a third party, mixing strategies are generally divided into centralized coin mixing and decentralized coin mixing.

**(1) Centralized Mixing Service**

Centralized coin mixing requires a third-party service provider to implement the coin mixing process. The coin mixing server mixes the input addresses of multiple transactions and then distributes the mixed funds to multiple output addresses. In this process, a certain fee is required.

Centralized coin mixing destroys the decentralization characteristics of the blockchain, and there is the problem of whether the centralized server is credible. To prevent third parties from maliciously leaking the mixing process, Bonneau et al. [17] described the Mixcoin mixing protocol in 2014, which introduces an audit mechanism to supervise the third party, each coin mixing transaction requires the signature of the centralized server, and users can complain to the malicious server to reduce its reputation. However, the server is completely aware of the transaction's input and output, so the transaction's anonymity is still not guaranteed. Valenta and Rowan [109] used blind signature technology to improve the Mixcoin protocol and then designed Blindcoin to protect the mixing process from leakage to other parties.

Unlike the preceding scheme based on a single fixed mixer, Dash [32] used a set of master nodes to provide users with coin mixing services, and each master node must pay 1,000 Dash as a deposit in advance to provide coin mixing services. Users who participate in mixing do not need to negotiate in advance and only need to send requests to the master node. When enough users send mixing requests to the same master node, the master node will create a mixing transaction that contains all input and output pairs. Additionally, users can randomly select multiple master nodes to mix coins to make the association between addresses invisible. But Dash only supports fixed denomination payments, which makes it very limited in real-world applications. At the same time, malicious master nodes also conduct internal attacks, leading to leakage of the privacy of the blockchain system.

Liu et al. [67] presented an unlinkable coin mixing scheme that allows users to mix their Bitcoins without trusting a third party for Bitcoin. In this way, the mixing server can only check whether the output addresses belong to its customers, but it cannot tell which address is owned by which customer.

The strength of privacy protection provided by centralized coin mixing is related to the number of participating addresses, and there are generally problems such as high mixing cost and low efficiency. At the same time, due to the lack of effective supervision mechanisms by third-party service providers, users participating in coin mixing cannot determine the true flow of their assets. Finally, centralized services are prone to a single point of failure and DoS attacks.

In recent years, there have been some off-chain transaction solutions based on a payment hub that can provide new solutions for centralized currency mixing services. They forward payments through the center, but the hub cannot link the payer and payee. Qin et al. [86] proposed a **payment channel (PC)** hub called *BlindHub* that is compatible with Bitcoin. It combines two encryption primitives, BAS (Blind Adapter Signature) and FBCS (Flexible Blind Conditional Signature), to achieve bi-directional payment of variable amounts with relational anonymity. Specifically, BlindHub built a BlindChannel in which the transaction amounts are invisible to one party, and the correctness of its updates is guaranteed by the NIZK proof. Ge et al. [38] proposed the optimized-unlinkable, NIZK-free, variable amount PC hub Accio. It only hides the state of the payee channel from the hub and uses RSUC (randomizable signature of updatable commitments) to update hidden states, with low communication overhead in off-chain transactions. However, this solution requires that the transaction flow in the channel be one-way and does not support bi-directional payment.

**(2) Decentralized Mixing Service**

To address the issue that centralized coin mixing is vulnerable to DoS attacks, Maxwell [70] proposed the decentralized currency mixing strategy CoinJoin designed for Bitcoin. The idea of CoinJoin can be viewed simply as when you want to transfer, first find other people who also want to transfer, and jointly create a transaction based on the transaction information of multiple users. This transaction has multiple inputs and multiple outputs, and users sign their own inputs

to construct a valid transaction. Specifically, any node acts as a mixing node and initiates a mixing service request by sending unsigned transactions to other nodes. Other nodes sign the input corresponding to the transaction and send it to the mixing node. The mixing node then signs its input to generate a valid transaction and then uploads it to the blockchain to complete the mixing process. Since CoinJoin does not have a central node, users must negotiate and execute the currency mixing process through other communication media, and the currency mixing node can understand the currency mixing situation of other nodes, so we classify it as a decentralized mixing service.

Ruffing et al. [94] improved the CoinJoin protocol, showing CoinShuffle, a decentralized Bitcoin mixing protocol. CoinShuffle uses centralized service nodes to randomly group users participating in the coin mixing service, and users can exchange their transaction information within the group. The server combines the single transactions received and sends them to the user, who then submits the transaction to the Bitcoin network for confirmation.

In 2015, Ziegeldorf et al. [130] built CoinParty using secure multi-party computation, tolerating some nodes' failure and malicious behavior. The redemption script set by CoinParty requires participants to execute ECDSA-based threshold signatures, which can ensure the effectiveness of the currency mixing service and effectively resist Sybil or DoS attacks. Sarfraz et al. [96] designed a novel decentralized mixing protocol for the IOTA ledger that incorporates a combination of decryption mixnets and multi-signatures. This technique does not require any (trusted or accountable) third party and is completely compatible with the IOTA protocol. Seres et al. [97] presented MixEth, which allows currency mixing services on Ethereum. Users deposit tokens into the smart contract's mixing pool, and the mixing pool randomly redistributes tokens to other users. In addition, MixEth uses **zero-knowledge proof (ZKP)** technology to prove the number of tokens in the mixing process, ensuring the security and anonymity of the mixing process.

For a set of original outputs with different amounts, mixing services need to decompose them into a set of decomposed outputs, where any decomposed output has some other decomposed outputs with the same amount. Since the transaction fee is related to the number of outputs, it is important to decompose the original output into a minimal set of decomposed outputs, which is challenging to guarantee the privacy-preserving effect simultaneously. Ni et al. [76] defined an AA-OD (anonymity-aware output decomposition) problem to simulate mixing services on the blockchain, and proposed an approximation algorithm, namely Boggart, to solve the AA-OD problem. This algorithm can find the decomposition with the least number of decompositions while satisfying the anonymity requirement and can efficiently mix transactions with arbitrary values on the blockchain. It is worth mentioning that this scheme only considers how to decompose a given original output, and it is a general solution that can be applied to the decomposing output step in both centralized and decentralized mixing services.

There are also some studies on the detection of currency mixing services. Wu et al. [115] described a method to identify mixing transactions that leverage the obfuscating mechanism, which identifies more than 92% of the mixing transactions. Sun et al. [103] implemented an LSTM transaction tree classifier (LSTM-TC) scheme that includes feature extraction and classification of Bitcoin transactions based on deep learning to detect coin mixing transactions.

A comparison of the characteristics of various mixing mechanisms is summarized in Table 3, including mixing characteristics, mixing fees, DoS attack risk, Sybil attack risk, theft risk, mixing scale, latency, anonymity (unlinkability), and number of mixed coins.

*Cryptographic Methods.* ZKP is a method by which the prover proves a proposition to the verifier. In such a proof process, no other information is disclosed except that "the proposition is true." In NIZK, the interaction between the prover and the verifier can be simulated by the prover without direct communication with the verifier, and the proof can be generated offline. Sun et al. [104] summarized the technology of applying ZKP to blockchain privacy protection issues.

Table 3. Comparison of Various Mixing Mechanisms: Mixing Fees (MF), DoS Attack Risks (DAR), Sybil Attack Risks (SAR), Risk of Theft (ROT), Mixing Scale (MS), and Amount of Mixed Coins (AOMC)

| Name | Features | MF | DAR | SAR | ROT | MS | Delay | Anonymity | AOMC |
|---|---|---|---|---|---|---|---|---|---|
| Mixcoin [17] | Hybrid center | Yes | High | Low | Low | No limitation | Large | Linkable at mixer | No limitation |
| Blindcoin [109] | Blind signature | Yes | High | Low | Middle | No limitation | Large | Unlinkable | No limitation |
| Dash [32] | Multi-center | Yes | Low | Low | Low | Small | Middle | Unlinkable | Small |
| CoinJoin [70] | Multi-round signature | No | High | High | Low | Small | Large | Internal unlinkable | Small |
| CoinShuffle [94] | Decrypt the hybrid network | No | Middle | Low | Low | Small | Large | Unlinkable | Small |
| CoinParty [130] | Multi-party secure computation | No | Low | Low | Middle | Large | Large | Unlinkable | Large |
| MixEth [97] | Smart contract | Yes | Low | Low | Low | Large | Middle | Unlinkable | No limitation |
| Ni et al. [76] | Decomposition transaction output | Yes | — | — | — | Large | Large | Unlinkable | No limitation |

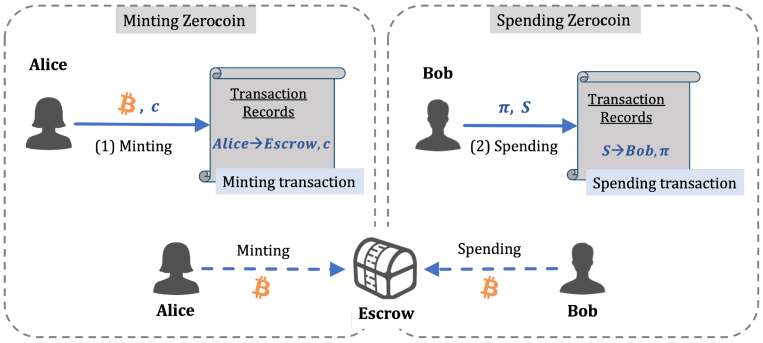


Fig. 3. The mechanism of Zerocoin. Here, $c$ represents user Alice's commitment to mint a Zerocoin, which is generated by $c \leftarrow g^S h^r \ mod \ p$. $(S, r)$ is the secret kept by Alice and is used to spend this Zerocoin, where $S$ is a serial number to identify a Zerocoin and $r$ is a random secret. Alice needs to generate a transaction with the output of $c$ and publish it to the blockchain. In this way, Bob converts a Bitcoin into a Zerocoin. $\pi$ represents a ZKP used to prove that Bob knows $(S, r)$. Bob publishes $\pi$ and $S$ to the blockchain in the form of transactions—that is, Bob spends this Zerocoin.

Using ZKP technology, Miers et al. [72] designed the Zerocoin protocol based on the Bitcoin system to prevent the analysis of transaction graphs on the blockchain. As shown in Figure 3, in this protocol, users can perform coin mixing operations alone and add assets to the cryptographic accumulator through coin minting transactions to generate private assets. When a user needs to use an asset, he only needs to prove that he owns an unspent asset in the cryptographic accumulator, and then spend the corresponding equivalent asset. However, Zerocoin has some shortcomings. For example, the denomination of Zerocoin is fixed, which is not convenient to use in practice. Second, it only hides the transaction originating address and cannot protect the transaction privacy based on the transaction amount or other metadata. Finally, the global initialization parameters of this algorithm must be generated by a trusted third party.

Sasson et al. [8] presented a Zerocash encryption scheme based on NIZK to improve these shortcomings. Compared with Zerocoin, Zerocash utilizes zk-SNARKs (zero-knowledge succinct non-interactive argument of knowledge, as it can hide the sender, receiver, and transaction amount of blockchain transactions, which has higher security. At the same time, the transaction amount is not fixed, which is more convenient for users. The proof process of Zerocoin is slow, and takes 450 ms (128-bit security level) to verify. Under the same configuration, Zerocash only needs 6 ms [39]. Zerocash, like Zerocoin, requires a trusted third party to initialize the encryption process's public and private key parameters. There are some improved studies for the preceding problems. Sasson et al. [9] introduced the Aurora scheme based on zk-STARKs (zero-knowledge scalable transparent argument of knowledge. There is no need to depend on trusted third-party initialization parameters. At the same time, it also increases the proof time and consumes more storage space.

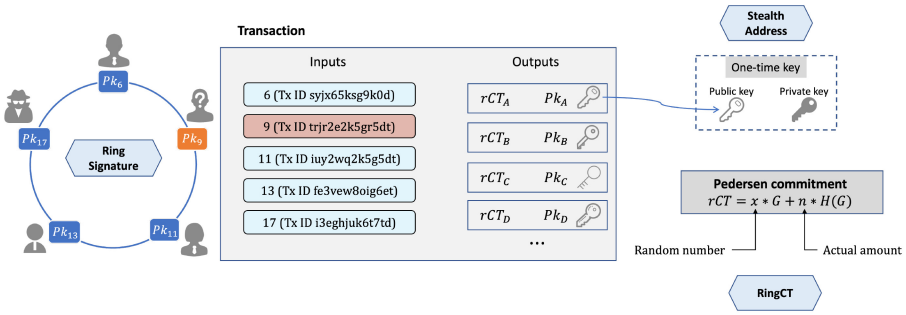


Fig. 4. The mechanism of Monero.

In addition to ZKP, the ring signature is a significant concept in blockchain cryptography. It provides an anonymous signature solution and allows users to sign without revealing their identity. Ring signature can ensure that the attacker cannot know the real signer's identity even if he can obtain all public and private keys. In the signing process, the signer can sign independently using his private key and some public keys in the public key pool. The applications of ring signature in blockchain mainly include Monero [77], which is based on CryptoNote [111], and ring confidential transaction (RingCT) [78]. As shown in Figure 4, this method uses a ring signature and one-time addresses to disconnect the input address and the output address in each transaction and hides the amount through RingCT, which can ensure both identity and transaction privacy. Although Monero allows for anonymous transactions and the concealment of transaction amounts, there are still certain issues. For example, the degree of protection for the privacy of the sender's identity depends on the number of members in the ring signature process. When the number of members in the ring signature is small (such as 2), there will be a risk of privacy leakage. Actually, most Monero transactions do not use a mixing service—that is, the transaction inputs do not contain any mixins at all but rather clearly identify one of the outputs of the transaction they spend. Not only do such 0-mixin transactions not provide privacy for the users who create them, but they also pose privacy risks to other transactions that use the output of this transaction as a mixin. In addition, Monero's mixing sampling distribution is inconsistent with the distribution of real transaction input. Usually, the real input is the latest, so the real input can be guessed based on the time distribution, with an accuracy of up to 80% [53, 73].

In Monero, when the user chooses the small anonymous set, the attacker can utilize transaction details to link the transaction to the user's identity [56]. Esgin et al. [35] introduced an efficient RingCT protocol named *MatRiCT* for blockchain confidential transactions. The proof length of the protocol is around two orders of magnitude shorter than the existing post-quantum proposal and scales efficiently to large anonymity sets. Tornado.Cash [1] is a decentralized mixer in Ethereum that improves privacy by breaking the on-chain link between source and destination addresses. To protect privacy, Tornado.Cash uses a smart contract to mix together a large number of deposits and withdrawals. Users can deposit different types of tokens into the Tornado contract. When withdrawing coins, they only need to show ZKP to prove that they have a deposit record in the Tornado contract and that the deposit has not been withdrawn. Then, they can use a new address to withdraw funds, thereby cutting off the correlation between deposit and withdrawal addresses.

*Network Information Hiding.* From the network layer perspective, information is transmitted between nodes through the P2P network. Attackers can obtain useful information by analyzing network node information and traffic, such as IP addresses, topological relationships between nodes,

and network transmission information. Network information hiding technology must hide the connection between the node's address in the blockchain and the IP address on the network layer.

The famous onion network Tor [90] is Bitcoin's most widely used anonymous communication system, integrating onion proxy, network topology, encryption, and other technologies to prevent attackers from tracking the user identity of transactions through monitoring, traffic analysis, and other means. To advance the privacy of the network layer, some nodes use the Tor network communication system to conceal the source IP address of the data packets. However, Tor network technology is imperfect; the theoretical threat model is weak and can only resist non-global active and passive attacks. Biryukov and Pustogarov [13] discussed that a DoS attack could disrupt the attack between Tor and blockchain networks. Even if both parties use the Tor network for anonymous transmission, the attacker can still use time and communication flow attacks to detect the network's communication delay and the correlation of data packets between onion proxies, judge the anonymous trail of the data packet, and then trace to the user's IP address.

Currently, Bitcoin's official way of safeguarding anonymity is the onion network. Monero adopts I2P instead of Tor as its anonymous communication technology. In Tor, data are sent and received via a single network link. However, I2P is an IPv6-based tunneling service that connects users to a host server in some hidden location, which helps to disguise an IP and avoid transaction tracking via network layer information.

*4.1.2 Permissioned Blockchain Based Approaches.* To meet the regulatory requirements of transactions, the permissioned blockchain adopts an identity authentication scheme based on digital certificates to achieve fine-grained access control to data [41]. At the same time, to solve the problem of identity privacy protection, the consortium blockchain represented by Hyperledger Fabric [2], Corda [92], and WeBank's FISCO BCOS [119] provides an anonymous identity authentication scheme.

*Certificate Management.* To ensure the unlinkability of transactions and identities in the ledger, the Fabric 0.6 version provides a two-level security certificate system based on ECert (the Enrollment Certificate) and TCert (the Transaction Certificate). When a user joins the Fabric network, a real-name certificate ECert needs to be issued by an enrollment certificate authority; when a registered user participates in a transaction, he can conduct a real-name or anonymous transaction. Real-name transactions are directly signed by the real-name certificate ECert, whereas anonymous transactions must be signed by a batch of anonymous transaction certificate TCerts derived from ECert generated by a transaction certificate authority.

The transactions cannot be linked in the user's anonymous transaction since a different TCert signs each transaction. To reduce the storage overhead of keys, the transaction key TCerts of Fabric 0.6 also adopt a key derivation method similar to Bitcoin's hierarchical deterministic wallet. At the same time, to ensure the need for supervision, TCerts contains the ciphertext of the user's real-name registration number EnrollmentID, and the supervisor can decrypt it with the corresponding decryption key to track the user's real-name identity. Corda also uses multiple ephemeral anonymous keys to protect user privacy through a key derivation scheme similar to Bitcoin hierarchical deterministic wallets. Like Fabric's TCerts solution, its transactions are divided into anonymous and public transactions.

The TCerts scheme provides a permissioned blockchain identity privacy protection scheme, but the maintenance cost of the TCerts certificate is very high, affecting the system's communication performance. For this problem, Fabric 2.0 achieved the idemix [21] scheme for improvement. The idemix scheme is built using blind signatures and ZKP that support multi-messages, issuing a credential containing a set of user attributes to the user, and when the user performs identity authentication, he needs to prove to the verifier that he has the credentials and present the corre-

sponding attributes. Since this proof process is implemented through ZKP, the user only needs to prove to the verifier that he has a signature and related attributes without providing a signature, and the user's real identity information will not be revealed.

Attribute-based encryption allows multiple peer nodes on the blockchain to share encrypted data based on attributes. **Ciphertext-policy attribute-based encryption (CP-ABE)** means that the ciphertext corresponds to an access structure, the key corresponds to an attribute set, and the message can be decrypted only if the attributes in the attribute set satisfy the ciphertext access structure [11]. To address security and privacy issues on the medical cloud platform, Tan et al. [105] illustrated a blockchain-empowered security and privacy protection scheme with traceable and direct revocation of COVID-19 medical records. Makhdoom et al. [69] constructed an innovative blockchain-based framework named *PrivySharing* to share privacy-preserving and secure Internet of Things data in a smart city environment. Dividing the blockchain network into various channels, every channel comprises a finite number of authorized organizations and processes a specific type of data such as health, smart energy, or financial details.

*Cryptographic Methods.* FISCO BCOS integrates the group/ring signature scheme to protect the user's identity privacy. FISCO BCOS version 2.3 began integrating the signature verification algorithms of the group signature scheme BBSO4 [16] and LSAG scheme [65] in the form of precompiled contracts. Among them, because no third party can open the signature in the ring signature, the LSAG scheme can determine whether the two ring signatures generated based on the same public key list come from the same signer. Users can use a group signature or a ring signature to sign transactions according to specific privacy requirements.

Ruan et al. [93] presented a system that adds access control views to Hyperledger Fabric with the idea of database authority management. They use cryptographic hash functions and encryption keys to implement two types of view (revocable and irrevocable) and to support role-based access control. At the same time, each user can verify that each view contains exactly the set of transactions it should consist of.

### 4.2 Transaction Privacy Protection

Many researchers have focused on developing a secure off-chain PC and a secure encryption method in response to the transaction privacy issues and challenges described previously. As explained in the following, we discuss these methods in a permissionless blockchain and a permissioned blockchain. We summarize the methods of transaction privacy protection and compare them from seven aspects: year, scenario, anonymity, blockchain type, technology, validated, and approach, as shown in Table 4.

#### *4.2.1 Permissionless Blockchain Based Approaches.*

*Off-Chain PC.* All users can access data on-chain for permissionless blockchain such as Bitcoin and Ethereum. To achieve access control of ledger information, we can establish an isolation channel (i.e., state channel technology) off-chain. Classic state channel application processes include the Lightning Network [84] and Raiden Network [42], used in Bitcoin and Ethereum, respectively.

Traditional Bitcoin transactions are not suitable for high-frequency micropayments due to expensive handling fees and long transaction times. In 2016, some research [44, 84] depicted the Lightning Network technology, which constructs a two-way PC by a **revocable sequence maturity contract (RSMC)**. If parties A and B do not build a two-way PC, but there is a two-way PC between A and C, B and C, then A and B can build a **payment channel network (PCN)** by using a **hash time lock contract (HTLC)** with intermediate party C. After the off-chain transaction

Table 4. Comparison of the Transaction Privacy Protection in Blockchain: Anonymity (A), Blockchain Type (BT), and Validated (V)

| Literature | Year | Scenario | A | BT | Technology | V | Approach |
|---|---|---|---|---|---|---|---|
| LightPay [66] | 2023 | Bitcoin | Yes | Permissionless | Adaptor signature | Yes | Design off-chain multi-path contracts based on the discrete logarithm problem |
| HCVC [118] | 2023 | Bitcoin | Yes | Permissionless | Time lock and digital signature | Yes | Bidirectional locking and punishment mechanism |
| Twilight [71] | 2022 | Bitcoin | Yes | Permissionless | Differential privacy and TEE | Yes | Add noise to transaction amounts and process transactions in TEE |
| Dharani et al. [49] | 2022 | Fabric | Yes | Permissioned | SMPC and zk-SNARKs | Yes | The endorsement process is divided into multiple stages and assigned to different nodes for calculation |
| CryptoMaze [71] | 2022 | Bitcoin | Yes | Permissionless | HTLC and ECC | Yes | Atomic multi-path payment protocol |
| Blitz [5] | 2021 | Bitcoin | Yes | Permissionless | Time lock and digital signature | Yes | Achieve fast single-round transaction confirmation; not vulnerable to wormhole attacks and coin stealing attacks |
| DIV [123] | 2021 | Generic | Yes | Both | ZKSM | Yes | Shard elements on blockchain into independent subsets with the same cardinality to reduce the effect of dynamic issues |
| PoFL [88] | 2021 | Generic | Yes | Permissioned | Federal learning | Yes | Propose a new consensus mechanism for federal learning |
| SlimChain [121] | 2021 | Generic | No | Permissionless | TEE | Yes | The complete state tree is stored off-chain; only verify on-chain |
| Qi et al. [85] | 2021 | Traffic | Yes | Permissioned | Differential privacy | Yes | With a noise addition mechanism to protect the location of the vehicles |
| Jia et al. [50] | 2021 | IIoT | Yes | Permissioned | Differential privacy and homomorphic encryption | Yes | The random forest with differential privacy and AdaBoost with homomorphic encryption methods |
| BFLC [58] | 2020 | Generic | Yes | Permissioned | Federal learning | Yes | Store the global model on-chain, and update the local model off-chain |
| Zether [18] | 2020 | Cryptocurrency | Yes | Permissionless | ZKP | Yes | Balance and transaction information are always encrypted to ensure data privacy |
| Duan et al. [31] | 2019 | Crowdsensing | Yes | Permissioned | Differential privacy, ZKP, and TEE | Yes | Data providers can safely contribute data to the transparent blockchain by employing a hardware-assisted transparent enclave |
| AMCU [34] | 2019 | Cryptocurrency | Yes | Permissionless | HTLC | Yes | Atomic multi-channel payment |
| Perun [33] | 2019 | Ethereum | Yes | Permissionless | Smart contract | Yes | Virtual channel |
| Mimblewimble [37] | 2019 | Cryptocurrency | Yes | Permissionless | Pedersen commitments and Schnorr or BLS signatures | Yes | An aggregate cash system |
| Bulletproofs [19] | 2018 | Generic | Yes | Permissionless | zk-SNARKs | Yes | Provide short proofs and do not need a trusted setup |
| zkLedger [75] | 2018 | Generic | Yes | Permissioned | Schnorr-type NIZK | Yes | A real-time audit of ciphertext on the blockchain |
| BPDS [64] | 2018 | Medical | Yes | Permissioned | CP-ABE | Yes | Full medical records are stored in the cloud, and the index is stored in the blockchain |
| Raiden Network [42] | 2018 | Ethereum | Yes | Permissionless | State channel | No | Off-chain transactions through smart contracts |
| Blot [40] | 2017 | ZeroCash | Yes | Permissionless | Blind signatures and ZKP | Yes | Anonymous PC off-chain |
| Lighting Network [84] | 2016 | Bitcoin | Yes | Permissionless | HTLC | Yes | Locking a fund on the chain through multi-signature constitutes an off-chain channel |
| BSC [44] | 2016 | Bitcoin | Yes | Permissionless | Blind signature | Yes | Off-chain PC for Bitcoin |
| TumbleBit [43] | 2016 | Bitcoin | Yes | Permissionless | RSA puzzle solver | Yes | Indirect unlinkable payment channel |

is completed, multiple parties must redistribute assets through multi-signature wallets. Lightning Network technology guarantees the security of off-chain transactions through RSMC and reduces the number of RSMC required in the network through HTLC. Aumayr et al. [5] presented a novel multi-hop payment protocol named *Blitz*, which can increase the number of concurrent payments per channel compared to Lighting Network. To solve privacy leakage problems in the existing PCN protocol, Xie et al. [118] proposed a lightweight virtual channel protocol HCVC based on the UTXO model. HCVC locks collateral to the PC through a two-way locking mechanism to increase the capacity of the virtual channel. In addition, a two-way penalty mechanism is also used to punish malicious nodes and achieve security across PCN peer-to-peer transactions. Mazumdar and Ruj [71] depicted a secure and privacy-preserving atomic multi-path payment protocol named *CryptoMaze*. It avoids the formation of multiple off-chain contracts on a channel shared by partial payments. Liu et al. [66] proposed LightPay, an atomic multi-path payment protocol based on adapter signatures and discrete logarithm problems. LightPay supports multi-path transactions and has a higher success rate than single-path payment solutions. Compared with CryptoMaze, LightPay uses adapter signatures to transmit the hidden values of the off-chain contract, which

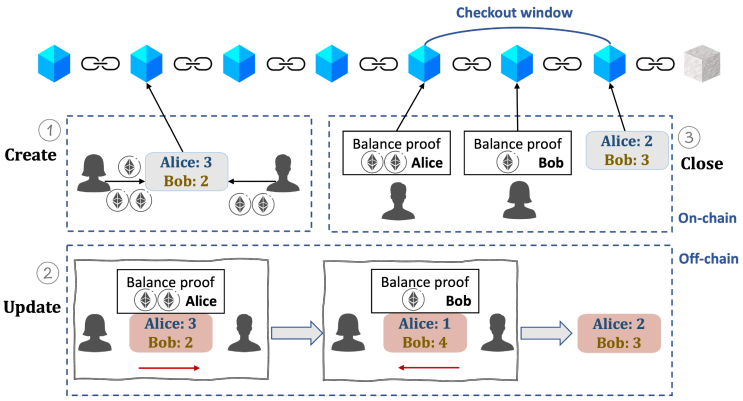


Fig. 5. The life cycle of the channel in the Raiden network.

has less communication overhead. In conclusion, an off-chain PC protects user privacy and reduces the pressure of blockchain transaction processing.

The Raiden Network is mainly used in Ethereum, and its technology and transaction processes are similar to the Lightning Network. The difference is that the transaction rules are specified for both parties through smart contracts that support more complex trading logic. The life cycle of an off-chain channel includes three stages: channel creation, update, and closure. Figure 5 presents a schematic diagram of the life cycle of the off-chain channel of Raiden Network. When the off-chain transaction channel is closed, the smart contract clears the transaction and sends the balance of both parties to the blockchain. Since the off-chain transaction is in the form of escrow hold tokens, they do not need multi-signature wallets and use hash time lock to ensure that users' funds are not lost.

The off-chain PC was first proposed by Heilman et al. [44]. Users willing to pay first need to create a PC on-chain. The PC contains the identity and coin information of the payer and payee. It ensures the anonymity and fairness through blind signatures and **Bitcoin smart contracts (BSCs)**. Green and Miers [40] designed an anonymous PC Blot based on Zerocash that provides three off-chain payment modes: unidirectional PC, bidirectional PC, and indirect PC. Users can perform transactions with untrusted third parties or off-chain channels. Blot's indirect PC is implemented by combining blind signatures and ZKP, which can effectively prevent third parties from gaining access to user data. TumbleBit [43] is an indirect non-linkable PC compatible with Bitcoin, which can combine multiple transactions of one party into two on-chain transactions. These transactions do not need to be stored or verified on the blockchain, using an off-chain solution Enigma to replace on-chain payments. TumbleBit ensures that off-chain transactions are completed in seconds, increasing the transaction speed of the blockchain. Egger et al. [34] presented a protocol for atomic multi-channel updates, lowering the quantity of collateral (i.e., tokens locked on payment pathways) and ensuring the atomicity of the protocol and transaction amount privacy. At the same time, the experiment also shows that the AMCU has the same performance as the Lightning Network and can be deployed in practice. Perun [33] offers a virtual channel to connect two endpoints that is more efficient than routing transaction schemes over multiple hop channels in Ethereum. Unlike PC, Perun uses virtual channels to connect multiple nodes. When the virtual channel is built, the two endpoints can skip the intermediate node for transactions, reducing the fee and improving the transaction confirmation speed. Table 5 shows a comparison of various off-chain channel methods.

Table 5. Comparison of Various Off-Chain Channel Methods: Unidirectional Payment (Unidi), Bidirectional Payment (Bidi), and Indirectional Payment (Indi)

| Technology | Protect Information | Compatibility | Unidi | Bidi | Indi | Functionality |
|---|---|---|---|---|---|---|
| Lighting Network [84] | Off-chain | Bitcoin | ✖ | ✔ | ✖ | Payment channel |
| Raiden Network [42] | Off-chain | Ethereum | ✖ | ✔ | ✖ | State channel |
| BSC [44] | Off-chain | Bitcoin | ✔ | ✖ | ✖ | Micropayment channel |
| Blot [40] | On-chain and off-chain | Zerocash | ✔ | ✔ | ✔ | Payment hub |
| TumbleBit [43] | On-chain and off-chain | Bitcoin | ✖ | ✔ | ✖ | Payment hub |
| Perun [33] | Off-chain | Ethereum | ✔ | ✔ | ✖ | Virtual channel |
| Blitz [5] | Off-chain | Bitcoin | ✔ | ✖ | ✔ | Payment channel |
| HCVC [118] | Off-chain | Bitcoin | ✖ | ✔ | ✖ | Virtual channel |
| CryptoMaze [71] | Off-chain | Bitcoin | ✔ | ✖ | ✔ | Payment channel |
| LightPay [66] | Off-chain | Bitcoin | ✔ | ✖ | ✔ | Payment channel |

*Cryptographic Methods.* Back [6] advocated the use of homomorphic encryption technology to conceal the transaction value, which is highly secure and concealable. The well-known schemes that use homomorphic encryption are the Pedersen commitment scheme and the Paillier encryption method. The Pedersen commitment scheme was first proposed by Pedersen [82], which encrypts the transaction amount through an elliptical curve, and other nodes can check the validity of the total amount of transaction input and output without knowing the transaction amount. Fuchsbauer et al. [37] presented Mimblewimble, which is an electronic cash system. It combines several privacy-enhancing techniques, including Pedersen's commitment. Paillier [80] was the first to propose an efficient additive homomorphic encryption method based on the composite residual class problem. Wang et al. [114] designed a Paillier-based encryption system that hides the transaction amount in the transaction and uses a commitment scheme to verify the validity of the encrypted amount. However, in some application scenarios that require supervision, the audit function cannot be supported. Singh et al. [100] applied homomorphic encryption to data aggregation in smart grids, reducing computational overhead and improving the performance of the blockchain. The combination of federated learning and homomorphic encryption is a potential research direction. Jia et al. [50] designed a data protection aggregation scheme and an application model of blockchain-enabled federated learning in the IIoT (Industrial Internet of Things). This scheme uses distributed K-means clustering with homomorphic encryption methods, which enable multiple data protection in data and model sharing.

In 2020, Zether, an Ethereum smart contract privacy protocol based on ZKP, was presented by Bünz et al. [18]. This protocol is deployed on Ethereum in the form of Zether smart contracts. Balance and transaction information is always encrypted to ensure data privacy and secure anonymous payments. However, Zether also has problems, such as high cost and network conditions that have a greater impact on transaction results. **Zero-knowledge set membership (ZKSM)** proof is widely used in blockchain to enable private membership attestation. Xu and Chen [123] exploited DIV to shard elements on blockchain into independent subsets with the same cardinality to reduce the effect of dynamic issues. It implements DIV on both Merkle tree and RSA-based ZKSM mechanisms to evaluate its efficiency and effectiveness and applies DIV on a ZKSM-based application named *zkSync* to demonstrate its applicability. Bulletproofs [19] provide short proofs and support the aggregation of range proofs, aggregated by generating a single proof through secure multi-party computing techniques without exposing its input to another party. In 2020, Raja et al. [89] designed an AI-powered blockchain decentralized, secure multi-party computation protocol for IoV, which provides an auto-coding feature for smart contracts, making it an intelligent contract.

IOTA [96] is a cryptocurrency designed for the Internet of Things industry. Unlike traditional public blockchain technology, IOTA does not use a chain structure to build the ledger. The "block" used by IOTA to store transactions is called *Tangle*. The essence of Tangle is a directed acyclic graph. IOTA uses the post-quantum signature scheme of W-OTS (Winternitz one-time signature) to resist quantum attacks. The W-OTS algorithm is used once since it cannot be reused after signing the name once; otherwise, there is a risk of losing money. IOTA allows multiple calls to an address, but after withdrawing funds once, it needs to be changed to a new address.

#### 4.2.2 *Permissioned Blockchain Based Approaches.*

*On-Chain Encryption Approaches.* For the transaction privacy protection of the permissioned chain, similar to the permissionless blockchain, the method of encrypting the data on the chain can also be used. For example, the transaction details on the public ledger can be hidden through homomorphic encryption, whereas ZKP can be combined to enable verification without revealing transaction information. Narula et al. [75] illustrated the blockchain system zkLedger for real-time auditing of ciphertext on the blockchain. zkLedger uses Schnorr-type NIZK to provide a fast-proof scheme that regulators can use to audit transactions on the blockchain between various banking entities. The scheme stores encrypted transaction information on the chain, allowing the auditor to conduct real-time audits on the chain while protecting transaction privacy between banking entities. In the work of Shao et al. [98], the proposed scheme allows users to create anonymous transactions using their transaction keys and attributes based on on-chain admission control and distributed identity management. AttriChain adopts an attribute-based signature scheme and ZKP to ensure the privacy of user transactions. In Fabric, since all nodes in the consensus process need to see the full content of the transaction, private information will be leaked. Dharani et al. [49] proposed a scheme to provide privacy protection for the consensus process in Fabric. The framework uses transaction verification based on multi-party computing protocols, and each node intelligently sees a part of the transaction. At the same time, ZKPs verify transactions without revealing the transaction content.

*Off-Chain Storage Approaches.* To ensure the privacy and scalability of the blockchain simultaneously, some solutions store transaction information off-chain and only save the hash of transaction information on the chain for users to verify whether the transaction has been tampered with. Liu et al. [64] constructed a privacy-preserving framework to protect patient electronic medical records, named *BPDS*, which stores the index of the medical record in the cloud. Patients can access electronic medical records through smart contracts based on pre-defined access rights.

*TEE Approaches.* TEE can build an independent execution environment that runs in parallel with the general operating system (Rich OS) to provide security services. TEE can isolate internal software and hardware resources from the applications running on it and provides a secure area for the device to protect important data and execute trusted code. Xu et al. [121] achieved a new trading framework, SlimChain, which transfers transaction processing and complete information storage to off-chain nodes and only performs verification on the chain. Specifically, SlimChain stores the complete state tree information of the blockchain by the storage node under the chain and runs the smart contract in the TEE environment to obtain the read-write set of changes in the state of the blockchain. The consensus nodes in the chain only store part of the state tree information, verify read and write sets from the storage nodes, and write legal read and write operations into the blockchain.

*Federated Learning Approaches.* Federated learning is distributed machine learning with the protection of local data privacy, which allows training data to build a global model through parameter

exchange without leaving individual distributed nodes [129]. Traditional federated learning uses a trusted third party for parameter exchange, which has problems such as single point of failure and privacy leakage [122]. The distributed ledger and decentralization technology of blockchain can well replace the central server of federated learning and protect the privacy of data and models trained in federated learning [126]. Qu et al. [88] proposed the consensus mechanism PoFL (Proof of Federated Learning) based on federated learning. They designed a reverse game-based data transaction mechanism and a privacy protection model to verify the mechanism, which prevents the leakage of training data and, at the same time, can effectively verify the accuracy of the data under the premise of protecting data privacy. Li et al. [58] illustrated a Blockchain-based distributed Federated Learning framework with Committee Consensus (BFLC). The framework uses blockchain as a platform for global model storage and local model update while designing a committee consensus mechanism to reduce consensus computation and resist malicious attacks.

*Differential Privacy Approaches.* Differential privacy protects the privacy of the original data by adding randomly generated "perturbations or noise"to the original data for obfuscation through rigorous mathematical models. In 2022, Dotan et al. [30] applied differential privacy to the off-chain payment network and proposed the Twilight scheme. In this scheme, the relay node needs to perform Gaussian noise processing on the transaction amount so that malicious users cannot detect the real balance in the off-chain channel. However, Twilight requires the relay nodes to lock enough funds on the chain to support differential privacy operations, and malicious relay nodes can try to bypass the differential privacy process and directly forward transactions. To solve this problem, Twilight puts the process of differential privacy in TEE, and the real output of TEE is invisible to the relay nodes. Qi et al. [85] propose a federated learning framework based on the consortium blockchain to predict traffic flow. Miners validate models from distributed vehicle nodes and then store the models on the blockchain. In addition, the framework adopts a local differential privacy method and a noise addition mechanism to protect the vehicle's location privacy. Jia et al. [50] demonstrated a federated learning framework for the IIoT. The framework designs a data aggregation scheme based on differential privacy and homomorphic encryption to achieve multiple data protection when sharing data and models, which implements a tradeoff between data availability and privacy. Duan et al. [31] adopted a crowd wisdom framework for privacy protection based on differential privacy and ZKP techniques. Among them, the user's personal data is integrated and stored through centralized differential privacy, and the confidentiality of user information is guaranteed through ZKP.

### 4.3 Smart Contract Privacy Protection

In response to the privacy issues mentioned previously and the challenges of smart contract privacy, a large and growing body of schemes has focused on designing an advanced smart contract execution framework, programming language, and hardware computing. In the following, we introduce these methods in the context of a permissioned blockchain (e.g., Bitcoin, Ethereum) and a permissionless blockchain (e.g., Hyperledger). We summarize the methods of smart contract privacy protection and compare them in seven aspects: year, scenario, anonymity, blockchain type, technology, validated, and approach, as depicted in Table 6.

#### *4.3.1 Permissionless Blockchain Based Approaches.*

*Advanced Smart Contract Execution Mechanism.* Blockchain platforms based on scripting languages like Bitcoin are limited in smart contract expression ability, execution efficiency, and privacy security. Currently, many research schemes overcome these problems by introducing a layer-2 protocol to transfer part of the computation, privacy code, and verification from on-chain to

Table 6. Comparison of the Smart Contract Privacy Protection in Blockchain: Anonymity (A), Blockchain Type (BT), and Validated (V)

| Literature | Year | Scenario | A | BT | Technology | V | Approach |
|---|---|---|---|---|---|---|---|
| Chen et al. [24] | 2024 | Generic | Yes | Both | Searchable Symmetric Encryption | Yes | Two miner nodes respectively are proposed to serve as malicious rule processors and detectors |
| VeriZexe [120] | 2023 | Generic | Yes | Both | Lightweight validator circuit | Yes | Off-chain computation and on-chain lightweight verification |
| DeCloak [91] | 2023 | Generic | Yes | Both | TEE and MPT | Yes | Off-chain contract execution framework |
| ZeeStar [101] | 2022 | Ethereum | Yes | Permissionless | Homomorphic encryption and NIZK | Yes | Privacy annotating and homomorphic operations on private data, to ensure input and output data privacy |
| AIASCG [106] | 2022 | Generic | Yes | Permissioned | Machine natural language | Yes | Convert legal contract documents into executable smart contracts |
| Aumayr et al. [4] | 2021 | Cryptocurrency | Yes | Permissionless | Time lock and digital signature | Yes | Need to lock a part of coins of off-chain nodes when the virtual channel is open or closed |
| KACHINA [54] | 2021 | Generic | Yes | Permissionless | NIZK and state oracles | Yes | Divide the smart contract state into a public on-chain and private off-chain state |
| Liu et al. [68] | 2021 | Generic | No | Permissioned | GNN and expert knowledge | Yes | A smart contract vulnerability detection tool |
| Ashizawa et al. [3] | 2021 | Generic | No | Permissioned | Machine learning | Yes | A machine learning based static analysis tool to detect smart contract vulnerabilities |
| SodsMPC [29] | 2020 | Generic | Yes | Permissionless | SMPC | Yes | A quantum-safe smart contract system |
| ChainIDE 2.0 [116] | 2020 | Generic | No | Both | Cloud based | Yes | Allow user to build a smart contract on different blockchain platforms efficiently |
| CONFIDE [124] | 2020 | Generic | Yes | Permissioned | TEE | Yes | Put the complete logic process off-chain, and only store the initial and final state on-chain |
| Zkay [102] | 2019 | Ethereum | Yes | Permissionless | NIZK | Yes | Allow programmers to specify data ownership by annotating variables as private data for a specific account |
| Origo [79] | 2019 | Ethereum | Yes | Permissionless | TEE and ZKP | Yes | Encrypt confidential information, transaction content, and contract execution details |
| Ekiden [28] | 2019 | Generic | Yes | Permissionless | TEE | Yes | Put the smart contract in TEE to complete the calculation of private data |
| FabZK [52] | 2019 | Generic | Yes | Permissioned | NIZK | Yes | An extension of Fabric that enables auditable privacy-safe smart contracts |
| Move [15] | 2019 | Libra | Yes | Permissioned | Specific language | Yes | Support issuing digital assets and validator management |
| CCF [95] | 2019 | Generic | Yes | Permissioned | TEE | Yes | Program the permissions of members, and each member's actions are recorded in the decentralized ledger |
| Hu and Zhang [46] | 2018 | Cryptocurrency | Yes | Permissionless | TEE | Yes | An efficient probabilistic payment system |
| BitML [7] | 2018 | Bitcoin | No | Permissionless | Specific language | Yes | Regulate transfers of Bitcoin among participants without relying on trusted intermediaries |
| Arbitrum [51] | 2018 | Generic | No | Permissionless | Digital signature | Yes | A committee-based approach to perform off-chain verification operations |
| Matrix [108] | 2017 | Generic | Yes | Permissioned | Deep learning | Yes | Automatic generation of smart contract code |
| Ivy [48] | 2017 | Bitcoin | No | Permissionless | Hash time lock | Yes | Support the Bitcoin for writing a complex smart contract |
| Hawk [55] | 2016 | Ethereum | Yes | Permissionless | Time lock | Yes | Divide smart contracts into on-chain public and off-chain private contracts |
| TC [127] | 2016 | Ethereum | Yes | Permissionless | TEE | Yes | Provide trusted input data for smart contract |

off-chain. In layer-2 protocol, only some necessary operations will be performed on the chain, such as storage, settlement, and dispute resolution, which protects transaction privacy. Additionally, it can effectively solve the impact of the blockchain's high latency and low throughput on the transaction process and improve the transaction confirmation speed. PC and probabilistic payment systems are the significant components of the layer-2 protocol. Aumayr et al. [5] proposed Blitz, an off-chain payment protocol based on the UTXO model. This protocol uses only time locks and digital signatures to achieve fast single-round transaction confirmation, which is not vulnerable to wormhole and coin stealing attacks. However, Aumayr et al. [5] cannot deal with the problems of off-chain node disconnection and high off-chain transaction fees. Therefore, Aumayr et al. [4] also presented a virtual channel protocol based on the UTXO model, which does not need to transmit the content of each transaction to the intermediate nodes off-chain.

Ethereum is another attractive permissionless blockchain architecture. In recent years, many scholars have achieved many advanced smart contract execution mechanisms to improve the efficiency and privacy of Ethereum [20]. Kosba et al. [55] proposed a Hawk smart contract system with

privacy protection capability, which divides smart contracts into on-chain and off-chain contracts: on-chain contracts contain public code, state information, and so on; off-chain contracts store some secret data and code functions, and private capital flows and transaction amounts. Hawk can help programmers without a professional cryptography background build smart contracts more intuitively. The Hawk compiler will automatically generate an effective encryption protocol to realize the interaction between the contracting parties and the blockchain. This framework sends encrypted information to the blockchain and relies on zk-SNARKs to ensure the correct execution of contracts. The system will punish malicious contract behaviors, thus realizing user privacy security and transaction security. However, Hawk does not consider data authenticity, which limits his application scenarios. Wan et al. [112] design an authenticated ZKP scheme called *zk-DASNARK* by extending zk-SNARK with an additional hash circuit that guarantees the authenticity of the data. Kalodner et al. [51] studied a highly scalable virtual machine architecture (i.e., Arbitrum), which uses a digital signature mechanism to reach a consensus on the behavior of virtual machines off-chain quickly. Arbitrum uses a committee-based approach to perform off-chain verification operations for smart contracts, which can significantly improve the scalability and privacy security of the system.

Given that the state of the smart contract is fully disclosed on-chain and executed by miners, the scalability and privacy of smart contracts are low. Li et al. [57] proposed a hybrid execution model to solve the problem of full on-chain sharing of smart contract content. This model splits smart contracts into shared on-chain contracts and off-chain privacy contracts according to users' computing and privacy needs and protects sensitive information in smart contracts. Dolev and Wang [29] provided SodsMPC, a quantum-safe smart contract system, which constructs the execution logic of the contract through secure multi-party computation and finite state machine, ensuring the correct execution of the contract while protecting data privacy. Wu et al. [116] proposed a cloud-based smart contract development system, ChainIDE 2.0, to allow users to build smart contracts on various blockchain platforms efficiently. Compared to ChainIDE 1.0 [87], ChainIDE 2.0 reduces the development cost of developers on the blockchain system and supports more blockchain platforms. Kerber et al. [54] constructed a unified security model for private contracts based on a general composable model, which supports the concurrent interaction of smart contracts. Kerber et al. [54] also presented the efficient contract composition protocol KACHINA that divides the smart contract state into a public on-chain state and a private off-chain state, in addition to utilizing NIZK and state oracles to implement private computing and complex smart contract functions.

*Advanced Smart Contract Programming Language.* Considering scripting language's functional and expressive limitations, many researchers have proposed new programming languages to increase Bitcoin's expressiveness and privacy security. Hu et al. [48] presented Ivy, one of the first high-level languages designed for Bitcoin. Ivy adds some specialized keywords for operations in Bitcoin, such as public key, value, and signature, to support Bitcoin for writing complex smart contracts. Notably, Ivy has been used to construct the HTLC [45]. However, Ivy lacks a rigorous security attestation process and may have security vulnerabilities. Bartoletti and Zunino [7] designed a high-level language, BitML, which improves the verifiability and expressiveness of Bitcoin scripts and enables Bitcoin transfers between legitimate participants without relying on trusted third parties. BitML integrates rich operation instructions to verify the privacy and security of programs through symbolic and computational models in terms of verifiability. However, BitML cannot provide a comprehensive expression for script-based blockchains.

Ethereum was one of the pioneering systems to deploy smart contracts. The input/output of smart contracts, contract functions, and participating users are all stored on the permissionless blockchain. The privacy leakage problem on Ethereum can be solved by hiding the contract code,

and NIZK can be used to ensure the correctness of the contract operation and state updates. However, the expressiveness of NIZK is limited and cannot be applied to all smart contract functions. Therefore, Steffen et al. [102] proposed the typed language Zkay that intuitively specifies contract functions and operation logic and uses static type checking to ensure private information will not be leaked. Zkay also defines smart contracts' read-and-write permission control, privacy variables logic, and private value types. Therefore, Zkay allows programmers to specify data ownership by annotating variables as private data for a specific account. When a Zkay contract is executed on a public blockchain, it is automatically converted into an executable contract equivalent in terms of privacy and functionality, and NIZK is used to ensure the correctness of contract execution and the privacy of the contract's private information. However, Zkay still has many functional limitations, such as insecure encryption and a lack of important language features. However, although Zkay can track the ownership of values in a privacy-type system through privacy annotations, it does not allow foreign expressions to participate in the calculation. To solve this problem, Steffen et al. [101] proposed ZeeStar, which uses the same privacy annotation as Zkay, but it combines homomorphic encryption and ZKP to encrypt input data and verify the correctness of smart contract execution results. ZeeStar tracks the ciphertext in the proof circuit and supports homomorphic addition and multiplication for most combinations of owners for foreign data. Xiong et al. [120] proposed a scheme to perform arbitrary computations off-chain without revealing program logic to the network. They proposed DPC, a new decentralized private computation that can support any number of applications with only a common setup. Ren et al. [91] propose a novel MPT (Multi-Party Transactions)-enabled off-chain contract execution framework Decloak, which solves identified properties with lower gas costs and a weaker assumption.

*Trusted Execution Environments.* Origo [79] proposed a privacy protection platform named *Origo* in response to the privacy issues of input data and execution results of Ethereum. Origo combines TEE and ZKP frameworks to encrypt confidential information, transaction content, and contract execution details and verify the validity of transactions. Origo supports a decentralized off-chain execution mechanism, enabling the creation and execution of smart contracts without revealing private inputs and outputs. To provide trusted input data for smart contracts, Zhang et al. [127] designed TC, a trusted data input system town crier, which combined Ethereum's smart contract front-end and SGX-based TEE back-end to provide source-authenticated data to smart contracts without a trusted third party. TC has high privacy protection capabilities, can process encrypted private data requests, and prevents unauthorized users from viewing private data. Cheng et al. [28] proposed Ekiden, a smart contract execution system with high performance and privacy. Ekiden places smart contracts in TEE to complete the calculation of private data, ensures the data integrity of transaction execution, and provides calculation proof to the blockchain after the calculation is complete. In Ekiden, consensus nodes jointly maintain the consistency of a decentralized ledger, which realizes the privacy protection of smart contracts based on TEE. Ekiden can process thousands of small transactions per second. Still, the lack of functionalities such as secure encryption technology and cross-node key management makes it unsuitable for concurrently executing smart contracts in complex business scenarios.

#### *4.3.2 Permissioned Blockchain Based Approaches.*

*Advanced Smart Contract Execution Mechanism.* Compared to permissionless blockchain, Hyperledger optimizes access control policies and refines the granularity of permissions for smart contract nodes. Users have different permissions to deploy, query, and execute smart contracts. Benhamouda et al. [10] implemented the privacy data protection of Hyperledger Fabric using secure multi-party computation. Users can store personal data encrypted on Hyperledger Fabric.

When the smart contract invokes these data, only the user with the corresponding decryption key can decrypt the ciphertext and perform secondary encryption as input to the SMPC protocol to ensure it is invisible to unauthorized users. To protect the privacy of transaction content and user identity, Kang et al. [52] implemented the FabZK framework, an extension of Fabric that enables auditable privacy-safe smart contracts through Pedersen commitments and NIZK. By providing a cluster of APIs for client code and chain code, FabZK only stores the encrypted data of each transaction and protects the identity information of the transaction parties. Furthermore, FabZK adopts zkLedger's table-structured ledger [75], which hides transaction details on the shared ledger and supports on-demand automatic verification based on encrypted data.

The advanced smart contract execution mechanism improves the system's privacy by splitting the smart contract and introducing cryptography. However, introducing too many cryptographic calculations increases the system's computational and time costs. For instance, ZKP needs to set a trusted proof for the smart contract in advance, which increases the system's complexity.

*Advanced Smart Contract Programming Language.* Another solution is to design an advanced smart contract programming language. As a secure and flexible programming language [15], Move supports functions such as issuing digital assets and validator management. With Move, developers can easily customize resource types, transaction logic, and access control strategies to build and manage digital asset members' permissions securely. Moreover, private resources will not be copied and discarded in Move, which can provide a secure and programmable basis for Libra, a virtual cryptocurrency launched by Facebook based on the consortium blockchain. However, these newly proposed languages deserve further improvement—for example, some languages currently suffer from problems such as complex expressions and syntax problems.

*Trusted Execution Environments.* To ensure the privacy and security of input data and computing processes, Hyperledger Fabric allows computing nodes to perform smart contract computing through TEE, which can be applied to more complex computing problems. Yan et al. [124] constructed CONFIDE, which supports on-chain confidentiality through TEE. It guarantees data confidentiality and integrity by a built-in data transmission protocol. Currently, CONFIDE supports millions of commercial transactions on the consortium chain. From the perspective of encrypting the contract code, in 2019, Microsoft built the framework CCF for the consortium blockchain [95]. By using TEE to simulate the operation and execution mode of the blockchain, smart contracts with high privacy, high availability, and low latency can be implemented. CCF can program members' permissions, whereas each member's actions are recorded in the decentralized ledger. However, TEEs are susceptible to different types of attacks. For example, on Qualcomm's TEEs, the trusted applications can be mapped directly into memory areas of the host operating system. So, by hijacking a vulnerable trusted application, an attacker can easily take control of the system using a buffer overflow [23]. In addition, side-channel attack methods can collect additional information generated when the TEE is running, thereby inferring confidential information within the system. The Foreshadow attack [110] exploits a feature in Intel processors called *transient out-of-order execution* to steal confidential data running in an SGX environment. When the TEE is attacked, CCF can perform fault recovery through verifiable replicated services to ensure the confidentiality and integrity of data and codes. At the same time, CCF detects and punishes malicious actions of members or replicas by replaying the service protocol.

*Contract Vulnerability Detection.* Smart contracts are essentially codes written by programmers, and loopholes will inevitably be difficult to detect. Once the smart contract is deployed on the blockchain, it cannot be changed, so it is imperative to perform certain security checks before deploying it. Liu et al. [68] proposed a smart contract vulnerability detection scheme based on a

GNN (graph neural network) and expert knowledge, which extracts expert patterns from smart contract source codes and then converts the codes into semantic graphs to extract depth graph features. Ashizawa et al. [3] introduced a machine learning based static analysis tool to detect smart contract vulnerability. Unlike existing machine learning static analysis tools, this scheme performs natural language processing through neural networks to extract features for each smart contract. At the same time, because the features are implicitly extracted by combining lexical semantics between smart contracts, the vulnerability can be accurately detected even after the code is rewritten.

Preliminary vulnerability detection of smart contracts can prevent privacy leakage. However, the preceding methods all use neural network methods to detect artificially written smart contracts. To make smart contracts more "smart,"we can consider using neural network characteristics to generate smart contract codes automatically. Tzu [108] designed the framework Matrix with automatic generation of smart contract code technology, where users themselves do not know the specific smart contract code. Matrix only requires users to construct the input, output, and transaction conditions of the contract in a scripting language, and then it can automatically convert the script into an equivalent smart contract program through a deep learning based code generator. The smart contracts generated by Matrix are invisible to users, guaranteeing smart contracts' code privacy. To solve the problem of incompatibility between blockchain smart contracts and legal contracts, Tong et al. [106] presented AIASCG, an AI-assisted Smart Contract Generation framework, utilizing AI-based automatic word segmentation technology to split legal contract sentences into words with specific semantics and generate machine-recognizable general representations. This method can reduce the risk of smart contract code privacy leakage. Chen et al. [24] proposed a practical and privacy-preserving malicious code detection method for encrypted smart contracts on a blockchain-based data trading platform. This scheme designs two types of miners, one kind of miner acts as a rule processor to generate an obfuscated map with the original open source malicious rule set, and another kind of miner acts as a detector to perform malicious inspection by inputting the obfuscated map and the randomized tokens of smart contract.

## 5 Discussion

This section compares existing privacy protection methods and discusses future research directions in blockchain privacy protection.

### 5.1 Summary

In this section, we summarize the privacy protection methods based on the classification of privacy depicted in this work.

*Methods for Identity Privacy.* Identity privacy aims to hide the mapping between blockchain addresses and the user identity. Common methods include mixing services, on-chain encryption, network information hiding, and certificate management. Mixing services are designed to safeguard individuals' identities' confidentiality by obfuscating their transactions' input and output addresses. It is achieved by blending multiple transactions, making it difficult to trace which input address corresponds to which output address. This scheme can improve the anonymity of the transaction, but the user needs to wait for the mixing services to process the transaction, which brings a certain transaction delay. On-chain encryption technology utilizes various cryptographic schemes to encrypt user addresses, which can effectively hide the user's actual address but also comes with the problem of high computational time overhead. Network information hiding refers to hiding the IP addresses of nodes forwarding transaction packets through a method like onion routing. This scheme can prevent attackers from obtaining users' IP addresses by detecting

network flow information. Certificate management is a universal solution for protecting and authenticating identity in the permissioned blockchain. By issuing digital certificates through a certificate authority, users can use their digital certificates for transactions. Although this method can effectively ensure the authenticity of the user identity, it requires complex certificate management mechanisms and is prone to a single point of failure.

*Methods for Transaction Privacy.* Methods to protect transaction privacy include off-chain PC, encryption schemes, off-chain storage, and differential privacy. Off-chain PC uses a portion of the funds staked on-chain to process off-chain transactions. The content of the off-chain transaction is invisible to the on-chain nodes, and the transaction will be executed on the chain only when the transaction is different to hide the transaction content. This type of solution requires the design of complex off-chain payment protocols, and as the number of nodes increases, the problem of off-chain node routing selection needs to be considered. Another common scheme is to perform homomorphic encryption on the transaction amount and verify it with ZKP. The advantage of this scheme is that it can directly operate on the ciphertext without disclosing the specific content of the transaction. In addition, it also has the issue of computational overhead. Off-chain storage dramatically reduces the storage overhead on the chain by storing transaction content offline and only storing the hash value of the transaction on the chain. However, off-chain data storage platforms are highly vulnerable to attacks, and the security of raw data cannot be guaranteed. The differential privacy scheme adds noise to the original transaction data through a mathematical model, preventing malicious users from detecting their true balance. In the process of designing differential privacy schemes, it is necessary to choose appropriate noise addition schemes to achieve a balance between data privacy and availability.

*Methods for Smart Contract Privacy.* The smart contract's code and function call information is publicly available, so the transaction information can be reconstructed by analyzing the results of the smart contract execution. The privacy protection schemes for smart contracts summarized in this article include new smart contract execution mechanisms, smart contract programming languages, TEE, and smart contract vulnerability mechanism analysis. We can design new smart contract execution mechanisms, such as distinguishing the processing of shared data and private data by separating the execution of on-chain and off-chain contracts, to prevent some privacy breaches involving private data processing contracts. Another solution is to design a new smart contract language to support read and write permission control, which has a high development cost and is highly likely to have security vulnerabilities and defects. Due to TEE's ability to ensure the security of the operating environment, we can incorporate the execution process of smart contracts into TEE to achieve privacy and verifiability of private data. However, protecting privacy in such schemes relies on the security of trusted hardware. In addition, we can adopt neural network based solutions to perform semantic analysis and feature extraction on smart contracts to detect potential vulnerabilities in smart contracts and take solutions. Moreover, exploring the technology that automates the creation of smart contracts can enhance the efficiency and quality of smart contract development.

### 5.2 Future Directions

Some existing privacy protection schemes have been applied to the blockchain, but many problems remain. By summarizing the problems of existing schemes and possible future research directions, it is our hope to inspire future researchers.

*Combination of Privacy Protection and Regulation.* Blockchain is a decentralized distributed database with data visible to other nodes. Current research on data privacy protection is mainly

conducted through cryptography. However, the proposed blockchain privacy protection scheme and the data access mechanism are difficult to support complex application scenarios, and there is a risk of data leakage. According to different application scenarios and requirements, there are differences in the privacy protection mechanisms adopted by the blockchain. For example, the consortium blockchain can combine technologies such as ZKP, PBFT, and a supervision mechanism to realize on-chain data storage and sharing. Therefore, in future work, researchers can design corresponding solutions according to different application scenarios and privacy protection requirements. For example, researchers can store data based on dynamic data desensitization technology and build a data transmission model based on cryptographic technologies such as ZKP and ciphertext calculation to achieve safe sharing and privacy of data. In some situations, lawbreakers may take advantage of the decentralized nature of the blockchain to commit illegal acts. In applying privacy protection technology, it is necessary to consider the combination with the supervision mechanism. When necessary, the identity and transaction content of the participants in the transaction can be restored to achieve a dynamic balance between privacy and supervision.

*Cross-Chain Data Sharing.* With the popularity of the blockchain, different organizations or government departments have their blockchain platforms, and the scenarios and demands for cross-chain interaction are increasing rapidly. Traditional cross-chain solutions protect the privacy of cross-chain data by combining sidechains and ZKPs. However, attackers can track sidechains' transaction privacy data by analyzing the main chain's transaction graph. In future research work, researchers can design data sharing schemes based on multi-threaded parallel cross-chain data synchronization mechanisms and a smart contract execution framework. Depending on different needs, the blockchain will adopt different privacy protection schemes. For example, medical platforms may use homomorphic encryption and attribute-based encryption to protect patient privacy, whereas transportation platforms may use stealth addresses to protect location privacy. Due to differences in architecture, data, and technology, it is difficult for different blockchains to interact. Therefore, it is necessary to define interface specifications for inter-chain authentication and access to achieve data sharing.

*Identity Authentication and Access Control.* Blockchain data are stored in blocks jointly maintained by multiple nodes. To prevent data from unauthorized access and tampering, it is necessary to design a secure data access control scheme. Traditional blockchain identity management and access control technologies have problems, such as strong dependence on centralization and information leakage. User identities can be verified through minimal information disclosure based on selective authorization methods, such as ZKP and attribute-based encryption. In complex application scenarios, user permissions must be dynamically adjusted. For example, when an authorized user behaves maliciously, the user's permission needs to be revoked in time to protect data security. Smart contracts are programs that can be automatically executed, so an automated authorization management solution can be built based on smart contracts to adjust data access rights dynamically. However, smart contracts are immutable codes stored on the blockchain, and the function and security of smart contracts need to be tested multiple times before being deployed on the blockchain. Typically, blockchains allow users to access their data, but not to change or delete the data. As a future direction, this functionality could be achieved by researching editable blockchain technologies such as sidechains or chameleon hash functions.

*High-Performance Blockchain Architecture.* The performance of the existing blockchain system is affected by the underlying architecture, and it is challenging to meet the requirements of low latency and high concurrent transactions. Improving performance through a scalable sharding architecture is a potential solution. For example, researchers can use network sharding technol-

ogy to divide the blockchain network into multiple independent shards according to the nodes' key characteristics to improve the system's reliability and security. When conducting cross-shard transactions, it is necessary to consider possible errors, failures, and malicious behaviors of nodes and establish an effective distributed verification mechanism.

*New Cryptography Primitives.* In this work, we identified that although many cryptography techniques have good privacy protection effects, they consume a lot of storage and verification time and are difficult to apply in scenarios with high real-time requirements. Researching more efficient encryption primitives is a possible research direction. In addition, from the application point of view, we can set the corresponding privacy protection levels for different data according to the specific needs of the data and select related technologies for a more efficient combination. For example, data computation privacy can be protected by combining secure multi-party computation with a TEE. TEE can protect data from the external environment during program execution. A secure storage and computing environment can be provided for private data by combining TEE and cryptography technologies. However, the problem of TEE schemes being vulnerable to attacks (e.g., hardware attacks and side-channel attacks) must be addressed in future work.

## 6 Concluding Remarks

Blockchain technology solves the problem of consistency in distributed networks and greatly impacts the architecture of applications that depend on trusted third parties. In recent years, blockchain applications have increased dramatically, involving many fields such as finance, the Internet of Things, supply chain, and electronic depository. This work provided a detailed analysis of the privacy issues of a permissioned and permissionless blockchain and discussed the latest relevant research results in recent years. In addition, we described the latest related methods in blockchain privacy protection for different aspects, such as identity, transaction, and smart contract privacy. We summarized the current privacy protection scheme by analyzing these methods, including the implementation technologies and existing problems. Finally, we discussed future research directions, among which is the design of privacy-preserving data storage and sharing schemes, which is one of the important directions. It is our hope that this article can give some researchers or engineers a guideline on how to set up corresponding privacy protection schemes according to actual demand.

## References

[1] Alexey Pertsev, Roman Semenov, and Roman Storm. 2019. Tornado Cash Privacy Solution Version 1.4. Retrieved July 4, 2024 from https://crebaco.com/planner/admin/uploads/whitepapers/2982941Tornado.cash_whitepaper_v1.4.pdf

[2] Alzubaidi (pseudonym) Ali. 2017. Hyperledger Fabric. Retrieved July 4, 2024 from https://github.com/hyperledger/fabric

[3] Nami Ashizawa, Naoto Yanai, Jason Paul Cruz, and Shingo Okamura. 2021. Eth2Vec: Learning contract-wide code representations for vulnerability detection on Ethereum smart contracts. In *Proceedings of the 3rd ACM International Symposium on Blockchain and Secure Critical Infrastructure (BSCI'21)*. ACM, New York, NY, USA, 47–59. https://doi.org/10.1145/3457337.3457841

[4] Lukas Aumayr, Matteo Maffei, Oğuzhan Ersoy, Andreas Erwig, Sebastian Faust, Siavash Riahi, Kristina Hostáková, and Pedro Moreno-Sanchez. 2021. Bitcoin-compatible virtual channels. In *Proceedings of the 2021 IEEE Symposium on Security and Privacy (SP'21)*. 901–918. https://doi.org/10.1109/SP40001.2021.00097

[5] Lukas Aumayr, Pedro Moreno-Sanchez, Aniket Kate, and Matteo Maffei. 2021. Blitz: Secure multi-hop payments without two-phase commits. In *Proceedings of the 30th USENIX Security Symposium (USENIX Security'21)*. 4043–4060. https://www.usenix.org/conference/usenixsecurity21/presentation/aumayr

[6] Adam Back. 2013. Bitcoins with homomorphic value (validatable but encrypted). *Bitcointalk*. Retrieved May 1, 2015 from https://bitcointalk.org/index.php

[7] Massimo Bartoletti and Roberto Zunino. 2018. BitML: A calculus for Bitcoin smart contracts. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (CCS'18)*. ACM, New York, NY, USA, 83–100. https://doi.org/10.1145/3243734.3243795

[8] Eli Ben Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza. 2014. Zerocash: Decentralized anonymous payments from Bitcoin. In *Proceedings of the 2014 IEEE Symposium on Security and Privacy*. IEEE, 459–474. https://doi.org/10.1109/SP.2014.36

[9] Eli Ben Sasson, Alessandro Chiesa, Michael Riabzev, Nicholas Spooner, Madars Virza, and Nicholas P. Ward. 2019. Aurora: Transparent succinct arguments for R1CS. In *Advances in Cryptology—EUROCRYPT 2019*, Yuval Ishai and Vincent Rijmen (Eds.). Springer International Publishing, Cham, 103–128.

[10] F. Benhamouda, S. Halevi, and T. Halevi. 2019. Supporting private data on hyperledger fabric with secure multiparty computation. *IBM Journal of Research and Development* 63, 2-3 (2019), Article 3, 8 pages. https://doi.org/10.1147/JRD.2019.2913621

[11] John Bethencourt, Amit Sahai, and Brent Waters. 2007. Ciphertext-policy attribute-based encryption. In *Proceedings of the 2007 IEEE Symposium on Security and Privacy (SP'07)*. IEEE, 321–334. https://doi.org/10.1109/SP.2007.11

[12] Alex Biryukov, Dmitry Khovratovich, and Ivan Pustogarov. 2014. Deanonymisation of clients in Bitcoin P2P network. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security (CCS'14)*. ACM, New York, NY, USA, 15–29. https://doi.org/10.1145/2660267.2660379

[13] Alex Biryukov and Ivan Pustogarov. 2015. Bitcoin over Tor isn't a good idea. In *Proceedings of the 2015 IEEE Symposium on Security and Privacy (SP'15)*. IEEE, 122–134. https://doi.org/10.1109/SP.2015.15

[14] George Bissias, A. Pinar Ozisik, Brian N. Levine, and Marc Liberatore. 2014. Sybil-resistant mixing for Bitcoin. In *Proceedings of the 13th Workshop on Privacy in the Electronic Society (WPES'14)*. ACM, New York, NY, USA, 149–158. https://doi.org/10.1145/2665943.2665955

[15] Sam Blackshear, Evan Cheng, David L. Dill, Victor Gao, Ben Maurer, Todd Nowacki, Alistair Pott, Shaz Qadeer, Dario Russi Rain, Stephane Sezer, Tim Zakian, and Runtian Zhou. 2019. *Move: A Language with Programmable Resources*. Libra Association

[16] Dan Boneh, Xavier Boyen, and Hovav Shacham. 2004. Short group signatures. In *Proceedings of the Annual International Cryptology Conference*. 41–55.

[17] Joseph Bonneau, Arvind Narayanan, Andrew Miller, Jeremy Clark, Joshua A. Kroll, and Edward W. Felten. 2014. Mixcoin: Anonymity for Bitcoin with accountable mixes. In *Financial Cryptography and Data Security*, Nicolas Christin and Reihaneh Safavi-Naini (Eds.). Springer, Berlin, Germany, 486–504.

[18] Benedikt Bünz, Shashank Agrawal, Mahdi Zamani, and Dan Boneh. 2020. Zether: Towards privacy in a smart contract world. In *Financial Cryptography and Data Security*, Joseph Bonneau and Nadia Heninger (Eds.). Springer International Publishing, Cham, 423–443.

[19] Benedikt Bünz, Jonathan Bootle, Dan Boneh, Andrew Poelstra, Pieter Wuille, and Greg Maxwell. 2018. Bulletproofs: Short proofs for confidential transactions and more. In *Proceedings of the 2018 IEEE Symposium on Security and Privacy (SP'18)*. IEEE, 315–334. https://doi.org/10.1109/SP.2018.00020

[20] Jiahong Cai, Wei Liang, Xiong Li, Kuanching Li, Zhenwen Gui, and Muhammad Khurram Khan. 2023. GTxChain: A secure IoT smart blockchain architecture based on graph neural network. *IEEE Internet of Things Journal* 10, 24 (2023), 21502–21514. https://doi.org/10.1109/JIOT.2023.3296469

[21] Jan Camenisch and Els Van Herreweghen. 2002. Design and implementation of the idemix anonymous credential system. In *Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS'02)*. ACM, New York, NY, USA, 21–30. https://doi.org/10.1145/586110.586114

[22] Miguel Castro and Barbara Liskov. 1999. Practical byzantine fault tolerance. In *Proceedings of the 3rd Symposium on Operating Systems Design and Implementation (OSDI'99)*. 173–186.

[23] David Cerdeira, Nuno Santos, Pedro Fonseca, and Sandro Pinto. 2020. SoK: Understanding the prevailing security vulnerabilities in TrustZone-assisted TEE systems. In *Proceedings of the 2020 IEEE Symposium on Security and Privacy (SP'20)*. 1416–1432. https://doi.org/10.1109/SP40000.2020.00061

[24] Dajiang Chen, Zeyu Liao, Ruidong Chen, Hao Wang, Chong Yu, Kuan Zhang, Ning Zhang, and Xuemin Shen. 2024. Privacy-preserving anomaly detection of encrypted smart contract for blockchain-based data trading. *IEEE Transactions on Dependable and Secure Computing*. Early Access, January 15, 2024. https://doi.org/10.1109/TDSC.2024.3353827

[25] Weili Chen, Tuo Zhang, Zhiguang Chen, Zibin Zheng, and Yutong Lu. 2020. Traveling the token world: A graph analysis of Ethereum ERC20 token ecosystem. In *Proceedings of the Web Conference 2020 (WWW'20)*. ACM, New York, NY, USA, 1411–1421. https://doi.org/10.1145/3366423.3380215

[26] Yourong Chen, Hao Chen, Yang Zhang, Meng Han, Madhuri Siddula, and Zhipeng Cai. 2022. A survey on blockchain systems: Attacks, defenses, and privacy preservation. *High-Confidence Computing* 2, 2 (2022), 100048. https://doi.org/10.1016/j.hcc.2021.100048

[27] Jieren Cheng, Luyi Xie, Xiangyan Tang, Naixue Xiong, and Boyi Liu. 2021. A survey of security threats and defense on blockchain. *Multimedia Tools and Applications* 80, 20 (2021), 30623–30652. https://doi.org/10.1007/s11042-020-09368-6

[28] Raymond Cheng, Fan Zhang, Jernej Kos, Warren He, Nicholas Hynes, Noah Johnson, Ari Juels, Andrew Miller, and Dawn Song. 2019. Ekiden: A platform for confidentiality-preserving, trustworthy, and performant smart contracts. In *Proceedings of the 2019 IEEE European Symposium on Security and Privacy (EuroS&P'19)*. IEEE, 185–200. https://doi.org/10.1109/EuroSP.2019.00023

[29] Shlomi Dolev and Ziyu Wang. 2020. SodsMPC: FSM based anonymous and private quantum-safe smart contracts. In *Proceedings of the 2020 IEEE 19th International Symposium on Network Computing and Applications (NCA'20)*. IEEE, 1–10. https://doi.org/10.1109/NCA51143.2020.9306699

[30] Maya Dotan, Saar Tochner, Aviv Zohar, and Yossi Gilad. 2022. Twilight: A differentially private payment channel network. In *Proceedings of the 31st USENIX Security Symposium (USENIX Security'22)*. 555–570. https://www.usenix.org/conference/usenixsecurity22/presentation/dotan

[31] Huayi Duan, Yifeng Zheng, Yuefeng Du, Anxin Zhou, Cong Wang, and Man Ho Au. 2019. Aggregating crowd wisdom via blockchain: A private, correct, and robust realization. In *Proceedings of the 2019 IEEE International Conference on Pervasive Computing and Communications (PerCom'19)*. IEEE, 1–10. https://doi.org/10.1109/PERCOM.2019.8767412

[32] Evan Duffield and Daniel Diaz. 2015. Dash: A privacy-centric crypto-currency. https://github.com/dashpay/dash/wiki/Whitepaper

[33] Stefan Dziembowski, Lisa Eckey, Sebastian Faust, and Daniel Malinowski. 2019. Perun: Virtual payment hubs over cryptocurrencies. In *Proceedings of the 2019 IEEE Symposium on Security and Privacy (SP'19)*. IEEE, 106–123. https://doi.org/10.1109/SP.2019.00020

[34] Christoph Egger, Pedro Moreno-Sanchez, and Matteo Maffei. 2019. Atomic multi-channel updates with constant collateral in Bitcoin-compatible payment-channel networks. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security (CCS'19)*. ACM, New York, NY, USA, 801–815. https://doi.org/10.1145/3319535.3345666

[35] Muhammed F. Esgin, Raymond K. Zhao, Ron Steinfeld, Joseph K. Liu, and Dongxi Liu. 2019. MatRiCT: Efficient, scalable and post-quantum blockchain confidential transactions protocol *(CCS'19)*. ACM, New York, NY, USA, 567–584. https://doi.org/10.1145/3319535.3354200

[36] Qi Feng, Debiao He, Sherali Zeadally, Muhammad Khurram Khan, and Neeraj Kumar. 2019. A survey on privacy protection in blockchain system. *Journal of Network and Computer Applications* 126 (2019), 45–58. https://doi.org/10.1016/j.jnca.2018.10.020

[37] Georg Fuchsbauer, Michele Orrù, and Yannick Seurin. 2019. Aggregate cash systems: A cryptographic investigation of Mimblewimble. In *Advances in Cryptology—EUROCRYPT 2019*, Yuval Ishai and Vincent Rijmen (Eds.). Springer International Publishing, Cham, 657–689.

[38] Zhonghui Ge, Jiayuan Gu, Chenke Wang, Yu Long, Xian Xu, and Dawu Gu. 2023. Accio: Variable-amount, optimized-unlinkable and NIZK-free off-chain payments via hubs. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security (CCS'23)*. ACM, New York, NY, USA, 1541–1555. https://doi.org/10.1145/3576915.3616577

[39] Daniel Genkin, Dimitrios Papadopoulos, and Charalampos Papamanthou. 2018. Privacy in decentralized cryptocurrencies. *Communications of the ACM* 61, 6 (2018), 78–88.

[40] Matthew Green and Ian Miers. 2017. Bolt: Anonymous payment channels for decentralized currencies *(CCS'17)*. ACM, New York, NY, USA, 473–489. https://doi.org/10.1145/3133956.3134093

[41] Dezhi Han, Yujie Zhu, Dun Li, Wei Liang, Alireza Souri, and Kuan-Ching Li. 2022. A blockchain-based auditable access control system for private data in service-centric IoT environments. *IEEE Transactions on Industrial Informatics* 18, 5 (2022), 3530–3540. https://doi.org/10.1109/TII.2021.3114621

[42] Heiko Hees. 2016. Raiden network: Off-chain state network for fast DApps. In *Devcon Two*. Ethereum Foundation.

[43] Ethan Heilman, Leen AlShenibr, Foteini Baldimtsi, Alessandra Scafuro, and Sharon Goldberg. 2017. TumbleBit: An Untrusted Bitcoin-Compatible Anonymous Payment Hub. Retrieved July 4, 2024 from https://open.bu.edu/handle/2144/29224

[44] Ethan Heilman, Foteini Baldimtsi, and Sharon Goldberg. 2016. Blindly signed contracts: Anonymous on-blockchain and off-blockchain Bitcoin transactions. In *Proceedings of the International Conference on Financial Cryptography and Data Security*. 43–60.

[45] Bin Hu, Zongyang Zhang, Jianwei Liu, Yizhong Liu, Jiayuan Yin, Rongxing Lu, and Xiaodong Lin. 2021. A comprehensive survey on smart contract construction and execution: Paradigms, tools, and systems. *Patterns* 2, 2 (2021), 100179. https://doi.org/10.1016/j.patter.2020.100179

[46] Kexin Hu and Zhenfeng Zhang. 2018. Fast lottery-based micropayments for decentralized currencies. In *Information Security and Privacy*, Willy Susilo and Guomin Yang (Eds.). Springer International Publishing, Cham, 669–686.

[47] Tianyuan Hu, Zecheng Li, and Bixin Li. 2021. Contractual security and privacy security of smart contract: A system mapping study. *Chinese Journal of Computers* 44, 12 (2021), 2485–2514.

[48] Ivy. 2017. Ivy Language Home Page. Retrieved July 4, 2024 from https://docs.ivylang.org/bitcoin/

[49] J. Dharani, K. Sundarakantham, Kunwar Singh, and S. Mercy Shalinie. 2022. A privacy-preserving framework for endorsement process in hyperledger fabric. *Computers & Security* 116 (2022), 102637. https://doi.org/10.1016/j.cose.2022.102637

[50] Bin Jia, Xiaosong Zhang, Jiewen Liu, Yang Zhang, Ke Huang, and Yongquan Liang. 2022. Blockchain-enabled federated learning data protection aggregation scheme with differential privacy and homomorphic encryption in IIoT. *IEEE Transactions on Industrial Informatics* 18, 6 (2022), 4049–4058. https://doi.org/10.1109/TII.2021.3085960

[51] Harry Kalodner, Steven Goldfeder, Xiaoqi Chen, S. Matthew Weinberg, and Edward W. Felten. 2018. Arbitrum: Scalable, private smart contracts. In *Proceedings of the 27th USENIX Security Symposium (USENIX Security'18)*. 1353–1370. https://www.usenix.org/conference/usenixsecurity18/presentation/kalodner

[52] Hui Kang, Ting Dai, Nerla Jean-Louis, Shu Tao, and Xiaohui Gu. 2019. FabZK: Supporting privacy-preserving, auditable smart contracts in Hyperledger Fabric. In *Proceedings of the 2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN'19)*. IEEE, 543–555. https://doi.org/10.1109/DSN.2019.00061

[53] George Kappos, Haaroon Yousaf, Mary Maller, and Sarah Meiklejohn. 2018. An empirical analysis of anonymity in Zcash. In *Proceedings of the 27th USENIX Security Symposium (USENIX Security'18)*. 463–477. https://www.usenix.org/conference/usenixsecurity18/presentation/kappos

[54] Thomas Kerber, Aggelos Kiayias, and Markulf Kohlweiss. 2021. KACHINA—Foundations of private smart contracts. In *Proceedings of the 2021 IEEE 34th Computer Security Foundations Symposium (CSF'21)*. IEEE, 1–16. https://doi.org/10.1109/CSF51468.2021.00002

[55] Ahmed Kosba, Andrew Miller, Elaine Shi, Zikai Wen, and Charalampos Papamanthou. 2016. Hawk: The blockchain model of cryptography and privacy-preserving smart contracts. In *Proceedings of the 2016 IEEE Symposium on Security and Privacy (SP'16)*. IEEE, 839–858. https://doi.org/10.1109/SP.2016.55

[56] Amrit Kumar, Clément Fischer, Shruti Tople, and Prateek Saxena. 2017. A traceability analysis of Monero's blockchain. In *Computer Security—ESORICS 2017*, Simon N. Foley, Dieter Gollmann, and Einar Snekkenes (Eds.). Springer International Publishing, Cham, 153–173.

[57] Chao Li, Balaji Palanisamy, and Runhua Xu. 2019. Scalable and privacy-preserving design of on/off-chain smart contracts. In *Proceedings of the 2019 IEEE 35th International Conference on Data Engineering Workshops (ICDEW'19)*. IEEE, 7–12. https://doi.org/10.1109/ICDEW.2019.00-43

[58] Yuzheng Li, Chuan Chen, Nan Liu, Huawei Huang, Zibin Zheng, and Qiang Yan. 2021. A blockchain-based decentralized federated learning framework with committee consensus. *IEEE Network* 35, 1 (2021), 234–241. https://doi.org/10.1109/MNET.011.2000263

[59] Wei Liang, Yuhui Li, Jianlong Xu, Zheng Qin, Dafang Zhang, and Kuan-Ching Li. 2024. QoS prediction and adversarial attack protection for distributed services under DLaaS. *IEEE Transactions on Computers* 73, 3 (2024), 669–682. https://doi.org/10.1109/TC.2021.3077738

[60] Wei Liang, Lijun Xiao, Ke Zhang, Mingdong Tang, Dacheng He, and Kuan-Ching Li. 2022. Data fusion approach for collaborative anomaly intrusion detection in blockchain-based systems. *IEEE Internet of Things Journal* 9, 16 (2022), 14741–14751. https://doi.org/10.1109/JIOT.2021.3053842

[61] Wei Liang, Siqi Xie, Kuan-Ching Li, Xiong Li, Xiaoyan Kui, and Albert Y. Zomaya. 2024. MC-DSC: A dynamic secure resource configuration scheme based on medical consortium blockchain. *IEEE Transactions on Information Forensics and Security* 19 (2024), 3525–3538. https://doi.org/10.1109/TIFS.2024.3364370

[62] Wei Liang, Yang Yang, Ce Yang, Yonghua Hu, Songyou Xie, Kuan-Ching Li, and Jiannong Cao. 2022. PDPChain: A consortium blockchain-based privacy protection scheme for personal data. *IEEE Transactions on Reliability*. Published Online, August 5, 2022. https://doi.org/10.1109/TR.2022.3190932

[63] Chao Liu, Han Liu, Zhao Cao, Zhong Chen, Bangdao Chen, and Bill Roscoe. 2018. ReGuard: Finding reentrancy bugs in smart contracts. In *Proceedings of the 40th International Conference on Software Engineering: Companion Proceedings (ICSE'18)*. ACM, New York, NY, USA, 65–68. https://doi.org/10.1145/3183440.3183495

[64] Jingwei Liu, Xiaolu Li, Lin Ye, Hongli Zhang, Xiaojiang Du, and Mohsen Guizani. 2018. BPDS: A blockchain based privacy-preserving data sharing for electronic medical records. In *Proceedings of the 2018 IEEE Global Communications Conference (GLOBECOM'18)*. IEEE, 1–6. https://doi.org/10.1109/GLOCOM.2018.8647713

[65] Joseph K. Liu, Victor K. Wei, and Duncan S. Wong. 2004. Linkable spontaneous anonymous group signature for ad hoc groups (extended abstract). In *Information Security and Privacy*. Lecture Notes in Computer Science, Vol. 3108. Springer, 325–335.

[66] Yaqin Liu, Wei Liang, Kun Xie, Songyou Xie, Kuanching Li, and Weizhi Meng. 2023. LightPay: A lightweight and secure off-chain multi-path payment scheme based on adapter signatures. *IEEE Transactions on Services Computing*. Early Access, November 17, 2023. https://doi.org/10.1109/TSC.2023.3333806

[67] Yi Liu, Xingtong Liu, Chaojing Tang, Jian Wang, and Lei Zhang. 2018. Unlinkable coin mixing scheme for transaction privacy enhancement of Bitcoin. *IEEE Access* 6 (2018), 23261–23270. https://doi.org/10.1109/ACCESS.2018.2827163

[68] Zhenguang Liu, Peng Qian, Xiaoyang Wang, Yuan Zhuang, Lin Qiu, and Xun Wang. 2023. Combining graph neural networks with expert knowledge for smart contract vulnerability detection. *IEEE Transactions on Knowledge and Data Engineering* 35, 2 (2023), 1296–1310. https://doi.org/10.1109/TKDE.2021.3095196

[69] Imran Makhdoom, Ian Zhou, Mehran Abolhasan, Justin Lipman, and Wei Ni. 2020. PrivySharing: A blockchain-based framework for privacy-preserving and secure data sharing in smart cities. *Computers & Security* 88 (2020), 101653. https://doi.org/10.1016/j.cose.2019.101653

[70] Gregory Maxwell. 2013. CoinJoin: Bitcoin privacy for the real world. *Bitcoin Forum*. Retrieved July 4, 2024 from https://bitcointalk.org/index.php?topic=279249.0

[71] Subhra Mazumdar and Sushmita Ruj. 2023. CryptoMaze: Privacy-preserving splitting of off-chain payments. *IEEE Transactions on Dependable and Secure Computing* 20, 2 (2023), 1060–1073. https://doi.org/10.1109/TDSC.2022.3148476

[72] Ian Miers, Christina Garman, Matthew Green, and Aviel D. Rubin. 2013. Zerocoin: Anonymous distributed E-cash from Bitcoin. In *Proceedings of the 2013 IEEE Symposium on Security and Privacy (SP'13)*. IEEE, 397–411. https://doi.org/10.1109/SP.2013.34

[73] Andrew Miller, Malte Moser, Kevin Lee, and Arvind Narayanan. 2017. An Empirical Analysis of Linkability in the Monero Blockchain. Retrieved July 4, 2024 from https://allquantor.at/blockchainbib/pdf/miller2017empirical.pdf

[74] Satoshi Nakamoto. 2008. Bitcoin: A peer-to-peer electronic cash system. *Decentralized Business Review* 2008 (2008), 21260.

[75] Neha Narula, Willy Vasquez, and Madars Virza. 2018. zkLedger: Privacy-preserving auditing for distributed ledgers. In *Proceedings of the 15th USENIX Symposium on Networked Systems Design and Implementation (NSDI'18)*. 65–80. https://www.usenix.org/conference/nsdi18/presentation/narula

[76] Wangze Ni, Peng Cheng, and Lei Chen. 2022. Mixing transactions with arbitrary values on blockchains. In *Proceedings of the 2022 IEEE 38th International Conference on Data Engineering (ICDE'22)*. IEEE, 2602–2614. https://doi.org/10.1109/ICDE53745.2022.00240

[77] Shen Noether. 2015. *Ring Signature Confidential Transactions for Monero*. Report 2015/1098. Cryptology ePrint Archive. https://eprint.iacr.org/2015/1098

[78] Shen Noether, Adam Mackenzie, and the Monero Research Lab. 2016. Ring confidential transactions. *Ledger* 1 (Dec. 2016), 1–18. https://doi.org/10.5195/ledger.2016.34

[79] F. Origo. 2019. Privacy Preserving Platform for Decentralized Applications. Retrieved July 4, 2024 from https://origo.network/whitepaper/

[80] Pascal Paillier. 1999. Public-key cryptosystems based on composite degree residuosity classes. In *Advances in Cryptology—EUROCRYPT'99*, Jacques Stern (Ed.). Springer, Berlin, Germany, 223–238.

[81] Amirmohammad Pasdar, Young Choon Lee, and Zhongli Dong. 2023. Connect API with blockchain: A survey on blockchain oracle implementation. *ACM Computing Surveys* 55, 10 (Feb. 2023), Article 208, 39 pages. https://doi.org/10.1145/3567582

[82] Torben Pryds Pedersen. 1992. Non-interactive and information-theoretic secure verifiable secret sharing. In *Advances in Cryptology—CRYPTO'91*, Joan Feigenbaum (Ed.). Springer, Berlin, Germany, 129–140.

[83] Li Peng, Wei Feng, Zheng Yan, Yafeng Li, Xiaokang Zhou, and Shohei Shimizu. 2021. Privacy preservation in permissionless blockchain: A survey. *Digital Communications and Networks* 7, 3 (2021), 295–307. https://doi.org/10.1016/j.dcan.2020.05.008

[84] Joseph Poon and Thaddeus Dryja. 2016. The Bitcoin Lightning Network: Scalable off-chain instant payments. Unpublished.

[85] Yuanhang Qi, M. Shamim Hossain, Jiangtian Nie, and Xuandi Li. 2021. Privacy-preserving blockchain-based federated learning for traffic flow prediction. *Future Generation Computer Systems* 117 (2021), 328–337. https://doi.org/10.1016/j.future.2020.12.003

[86] Xianrui Qin, Shimin Pan, Arash Mirzaei, Zhimei Sui, Oğuzhan Ersoy, Amin Sakzad, Muhammed F. Esgin, Joseph K. Liu, Jiangshan Yu, and Tsz Hon Yuen. 2023. BlindHub: Bitcoin-compatible privacy-preserving payment channel hubs supporting variable amounts. In *Proceedings of the 2023 IEEE Symposium on Security and Privacy (SP'23)*. 2462–2480. https://doi.org/10.1109/SP46215.2023.10179427

[87] Han Qiu, Xiao Wu, Shuyi Zhang, Victor C. M. Leung, and Wei Cai. 2019. ChainIDE: A cloud-based integrated development environment for cross-blockchain smart contracts. In *Proceedings of the 2019 IEEE International Conference on Cloud Computing Technology and Science (CloudCom'19)*. IEEE, 317–319. https://doi.org/10.1109/CloudCom.2019.00055

[88] Xidi Qu, Shengling Wang, Qin Hu, and Xiuzhen Cheng. 2021. Proof of federated learning: A novel energy-recycling consensus algorithm. *IEEE Transactions on Parallel and Distributed Systems* 32, 8 (2021), 2074–2085. https://doi.org/10.1109/TPDS.2021.3056773

[89] Gunasekaran Raja, Yelisetty Manaswini, Gaayathri Devi Vivekanandan, Harish Sampath, Kapal Dev, and Ali Kashif Bashir. 2020. AI-powered blockchain—A decentralized secure multiparty computation protocol for IoV. In *Proceedings of the IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS'20)*. IEEE, 865–870. https://doi.org/10.1109/INFOCOMWKSHPS50562.2020.9162866

[90] M. G. Reed, P. F. Syverson, and D. M. Goldschlag. 1998. Anonymous connections and onion routing. *IEEE Journal on Selected Areas in Communications* 16, 4 (1998), 482–494. https://doi.org/10.1109/49.668972

[91] Qian Ren, Yue Li, Yingjun Wu, Yuchen Wu, Hong Lei, Lei Wang, and Bangdao Chen. 2024. DeCloak: Enable secure and cheap multi-party transactions on legacy blockchains by a minimally trusted TEE network. *IEEE Transactions on Information Forensics and Security* 19 (2024), 88–103. https://doi.org/10.1109/TIFS.2023.3318935

[92] rnab-r3 (pseudonym). 2017. Corda. Retrieved July 4, 2024 from https://github.com/corda/corda

[93] Pingcheng Ruan, Yaron Kanza, Beng Chin Ooi, and Divesh Srivastava. 2022. LedgerView: Access-control views on hyperledger fabric. In *Proceedings of the 2022 International Conference on Management of Data (SIGMOD'22)*. ACM, New York, NY, USA, 2218–2231. https://doi.org/10.1145/3514221.3526046

[94] Tim Ruffing, Pedro Moreno-Sanchez, and Aniket Kate. 2014. CoinShuffle: Practical decentralized coin mixing for Bitcoin. In *Computer Security—ESORICS 2014*, Mirosław Kutyłowski and Jaideep Vaidya (Eds.). Springer International Publishing, Cham, 345–364.

[95] Mark Russinovich, Edward Ashton, Christine Avanessians, Miguel Castro, Amaury Chamayou, Sylvan Clebsch, Manuel Costa, Cédric Fournet, Matthew Kerner, Sid Krishna, Julien Maffre, Thomas Moscibroda, Kartik Nayak, Olga Ohrimenko, Felix Schuster, Roy Schuster, Alex Shamis, Olga Vrousgou, and Christoph M. Wintersteiger. 2019. *CCF: A Framework for Building Confidential Verifiable Replicated Services*. Technical Report. Microsoft.

[96] Umair Sarfraz, Masoom Alam, Sherali Zeadally, and Abid Khan. 2019. Privacy aware IOTA ledger: Decentralized mixing and unlinkable IOTA transactions. *Computer Networks* 148 (2019), 361–372. https://doi.org/10.1016/j.comnet.2018.11.019

[97] István András Seres, Dániel A. Nagy, Chris Buckland, and Péter Burcsi. 2019. MixEth: Efficient, trustless coin mixing service for Ethereum. *Cryptology ePrint Archive*. Retrieved July 4, 2024 from https://eprint.iacr.org/2019/341

[98] Wei Shao, Chunfu Jia, Yunkai Xu, Kefan Qiu, Yan Gao, and Yituo He. 2020. AttriChain: Decentralized traceable anonymous identities in privacy-preserving permissioned blockchain. *Computers & Security* 99 (2020), 102069. https://doi.org/10.1016/j.cose.2020.102069

[99] Janno Siim. 2017. Proof-of-stake. In Proceedings of the Research Seminar in Cryptography.

[100] Parminder Singh, Mehedi Masud, M. Shamim Hossain, and Avinash Kaur. 2021. Blockchain and homomorphic encryption-based privacy-preserving data aggregation model in smart grid. *Computers & Electrical Engineering* 93 (2021), 107209. https://doi.org/10.1016/j.compeleceng.2021.107209

[101] Samuel Steffen, Benjamin Bichsel, Roger Baumgartner, and Martin Vechev. 2022. ZeeStar: Private smart contracts by homomorphic encryption and zero-knowledge proofs. In *Proceedings of the 2022 IEEE Symposium on Security and Privacy (SP'22)*. IEEE, 179–197. https://doi.org/10.1109/SP46214.2022.9833732

[102] Samuel Steffen, Benjamin Bichsel, Mario Gersbach, Noa Melchior, Petar Tsankov, and Martin Vechev. 2019. zkay: Specifying and enforcing data privacy in smart contracts. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security (CCS'19)*. ACM, New York, NY, USA, 1759–1776. https://doi.org/10.1145/3319535.3363222

[103] Xiaowen Sun, Tan Yang, and Bo Hu. 2022. LSTM-TC: Bitcoin coin mixing detection method with a high recall. *Applied Intelligence* 52, 1 (2022), 780–793. https://doi.org/10.1007/s10489-021-02453-9

[104] Xiaoqiang Sun, F. Richard Yu, Peng Zhang, Zhiwei Sun, Weixin Xie, and Xiang Peng. 2021. A survey on zero-knowledge proof in blockchain. *IEEE Network* 35, 4 (2021), 198–205. https://doi.org/10.1109/MNET.011.2000473

[105] Liang Tan, Keping Yu, Na Shi, Caixia Yang, Wei Wei, and Huimin Lu. 2022. Towards secure and privacy-preserving data sharing for COVID-19 medical records: A blockchain-empowered approach. *IEEE Transactions on Network Science and Engineering* 9, 1 (2022), 271–281. https://doi.org/10.1109/TNSE.2021.3101842

[106] Yu Tong, Weiming Tan, Jingzhi Guo, Bingqing Shen, Peng Qin, and Shuaihe Zhuo. 2022. Smart contract generation assisted by AI-based word segmentation. *Applied Sciences* 12, 9 (2022), 4773. https://doi.org/10.3390/app12094773

[107] Florian Tramèr, Dan Boneh, and Kenneth G. Paterson. 2020. Remote side-channel attacks on anonymous transactions. In *Proceedings of the 29th USENIX Conference on Security Symposium (SEC'20)*. Article 154, 18 pages.

[108] Lao Tzu. 2017. MATRIX Technical Whitepaper. https://cryptoactu.com/wp-content/uploads/2017/12/MATRIXTechnicalWhitePaper.pdf

[109] Luke Valenta and Brendan Rowan. 2015. Blindcoin: Blinded, accountable mixes for Bitcoin. In *Financial Cryptography and Data Security*, Michael Brenner, Nicolas Christin, Benjamin Johnson, and Kurt Rohloff (Eds.). Springer, Berlin, Germany, 112–126.

[110] Jo Van Bulck, Marina Minkin, Ofir Weisse, Daniel Genkin, Baris Kasikci, Frank Piessens, Mark Silberstein, Thomas F. Wenisch, Yuval Yarom, and Raoul Strackx. 2018. Foreshadow: Extracting the keys to the Intel SGX kingdom with transient out-of-order execution. In *Proceedings of the 27th USENIX Conference on Security Symposium (SEC'18)*. 991–1008.

[111] Nicolas Van Saberhagen. 2013. CryptoNote v 2.0. https://c3.coinlore.com/pdf/bytecoin-bcn-white-paper.pdf
[112] Zhiguo Wan, Yan Zhou, and Kui Ren. 2023. zk-AuthFeed: Protecting data feed to smart contracts with authenticated zero knowledge proof. *IEEE Transactions on Dependable and Secure Computing* 20, 2 (2023), 1335–1347. https://doi.org/10.1109/TDSC.2022.3153084
[113] Dan Wang, Jindong Zhao, and Yingjie Wang. 2020. A survey on privacy protection of blockchain: The technology and application. *IEEE Access* 8 (2020), 108766–108781. https://doi.org/10.1109/ACCESS.2020.2994294
[114] Qin Wang, Bo Qin, Jiankun Hu, and Fu Xiao. 2020. Preserving transaction privacy in Bitcoin. *Future Generation Computer Systems* 107 (2020), 793–804. https://doi.org/10.1016/j.future.2017.08.026
[115] Lei Wu, Yufeng Hu, Yajin Zhou, Haoyu Wang, Xiapu Luo, Zhi Wang, Fan Zhang, and Kui Ren. 2021. Towards understanding and demystifying Bitcoin mixing services. In *Proceedings of the Web Conference 2021 (WWW'21)*. ACM, New York, NY, USA, 33–44. https://doi.org/10.1145/3442381.3449880
[116] Xiao Wu, Han Qiu, Shuyi Zhang, Gerard Memmi, Keke Gai, and Wei Cai. 2020. ChainIDE 2.0: Facilitating smart contract development for consortium blockchain. In *Proceedings of the IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS'20)*. IEEE, 388–393. https://doi.org/10.1109/INFOCOMWKSHPS50562.2020.9163051
[117] Zhiying Wu, Jieli Liu, Jiajing Wu, Zibin Zheng, and Ting Chen. 2023. TRacer: Scalable graph-based transaction tracing for account-based blockchain trading systems. *IEEE Transactions on Information Forensics and Security* 18 (2023), 2609–2621. https://doi.org/10.1109/TIFS.2023.3266162
[118] Songyou Xie, Lijun Xiao, Dezhi Han, Kun Xie, Xiong Li, and Wei Liang. 2023. HCVC: A high-capacity off-chain virtual channel scheme based on bidirectional locking mechanism. *IEEE Transactions on Network Science and Engineering*. Early Access, November 17, 2023. https://doi.org/10.1109/TNSE.2023.3332130
[119] Bai (pseudonym) XingQiang. 2018. FISCO BCOS. Retrieved July 4, 2024 from https://github.com/FISCO-BCOS/FISCO-BCOS
[120] Alex Luoyuan Xiong, Binyi Chen, Zhenfei Zhang, Benedikt Bünz, Ben Fisch, Fernando Krell, and Philippe Camacho. 2023. VeriZexe: Decentralized private computation with universal setup. In *Proceedings of the 32nd USENIX Security Symposium (USENIX Security'23)*. 4445–4462. https://www.usenix.org/conference/usenixsecurity23/presentation/xiong
[121] Cheng Xu, Ce Zhang, Jianliang Xu, and Jian Pei. 2021. SlimChain: Scaling blockchain transactions through off-chain storage and parallel processing. *Proceedings of the VLDB Endowment* 14, 11 (July 2021), 2314–2326. https://doi.org/10.14778/3476249.3476283
[122] Yang Xu, Md. Zakirul Alam Bhuiyan, Tian Wang, Xiaokang Zhou, and Amit Kumar Singh. 2023. C-FDRL: Context-aware privacy-preserving offloading through federated deep reinforcement learning in cloud-enabled IoT. *IEEE Transactions on Industrial Informatics* 19, 2 (2023), 1155–1164. https://doi.org/10.1109/TII.2022.3149335
[123] Zihuan Xu and Lei Chen. 2021. DIV: Resolving the dynamic issues of zero-knowledge set membership proof in the blockchain. In *Proceedings of the 2021 International Conference on Management of Data (SIGMOD'21)*. ACM, New York, NY, USA, 2036–2048. https://doi.org/10.1145/3448016.3457248
[124] Ying Yan, Changzheng Wei, Xuepeng Guo, Xuming Lu, Xiaofu Zheng, Qi Liu, Chenhui Zhou, Xuyang Song, Boran Zhao, Hui Zhang, and Guofei Jiang. 2020. Confidentiality support over financial grade consortium blockchain. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data (SIGMOD'20)*. ACM, New York, NY, USA, 2227–2240. https://doi.org/10.1145/3318464.3386127
[125] Fan Yang, Wei Zhou, QingQing Wu, Rui Long, Neal N. Xiong, and Meiqi Zhou. 2019. Delegated proof of stake with downgrade: A secure and efficient blockchain consensus algorithm with downgrade mechanism. *IEEE Access* 7 (2019), 118541–118555. https://doi.org/10.1109/ACCESS.2019.2935149
[126] Cheng Zhang, Yang Xu, Haroon Elahi, Deyu Zhang, Yunlin Tan, Junxian Chen, and Yaoxue Zhang. 2023. A blockchain-based model migration approach for secure and sustainable federated learning in IoT systems. *IEEE Internet of Things Journal* 10, 8 (2023), 6574–6585. https://doi.org/10.1109/JIOT.2022.3171926
[127] Fan Zhang, Ethan Cecchetti, Kyle Croman, Ari Juels, and Elaine Shi. 2016. Town Crier: An authenticated data feed for smart contracts. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS'16)*. ACM, New York, NY, USA, 270–282. https://doi.org/10.1145/2976749.2978326
[128] Rui Zhang, Rui Xue, and Ling Liu. 2019. Security and privacy on blockchain. *ACM Computing Surveys* 52, 3 (July 2019), Article 51, 34 pages. https://doi.org/10.1145/3316481
[129] Sisi Zhou, Kuanching Li, Yuxiang Chen, Ce Yang, Wei Liang, and Albert Y. Zomaya. 2024. TrustBCFL: Mitigating data bias in IoT through blockchain-enabled federated learning. *IEEE Internet of Things Journal*. Early Access, April 4, 2024. https://doi.org/10.1109/JIOT.2024.3379363
[130] Jan Henrik Ziegeldorf, Fred Grossmann, Martin Henze, Nicolas Inden, and Klaus Wehrle. 2015. CoinParty: Secure multi-party mixing of Bitcoins *(CODASPY'15)*. ACM, New York, NY, USA, 75–86. https://doi.org/10.1145/2699026.2699100

Received 18 August 2022; revised 19 April 2024; accepted 23 June 2024