Published online: 22 October 2024 DOI: 10.54254/2977-3903/12/2024130

Overview of privacy protection in blockchain transaction

Mingyuan Yu

School of Information and Artificial Intelligence, Zhejiang University of Finance & Economics Dongfang College, Haining, China

2120410138@zufedfc.edu.cn

Abstract. Blockchain technology has transformed digital transactions by enhancing security and trust through decentralization and cryptography. However, this technology also poses significant privacy challenges in transactions. Specifically, the transparency inherent in blockchain can lead to privacy concerns where sensitive transaction data and user identities are potentially exposed. This paper explores the dichotomy of data privacy and identity privacy within blockchain transactions, using Ethereum as a primary example. It delves into various privacy protection technologies currently in use, such as homomorphic encryption and zero-knowledge proofs, and evaluates their effectiveness and limitations. The paper further discusses the balance between privacy protection and regulatory compliance, which remains a critical challenge. Additionally, it considers future directions for enhancing privacy without compromising the security and functionality of blockchain systems. This exploration is vital for developing blockchain applications that maintain user confidentiality while benefiting from the transparency and security that blockchain provides.

Keywords: Blockchain, privacy protection, cryptography

1. Introduction

Since Satoshi Nakamoto released the Bitcoin white paper in 2008, the 16 years we have witnessed the metamorphosis from Web2.0 to Web3.0, and the birth of emerging concepts such as the meta-universe. In this process, blockchain technology has played an irreplaceable role, which also reflects the far-reaching impact of blockchain technology on the digital economy. As a decentralized network technology, blockchain provides the basis for the rise of cryptocurrencies such as Bitcoin, Ethereum and Monero by virtue of the P2P network architecture and the need to rely on any trusted authorization for tracking. At the same time, the application of blockchain technology in the financial fields such as insurance and banking, as well as in the Internet of Things fields such as medical care and transportation, has shown an exponential growth trend, which indicates that blockchain has experienced a wide range of market tests and market recognition, and also shows the huge potential of blockchain technology in improving efficiency and reducing costs.

However, with the continuous spread of commercial applications, the privacy issue of blockchain has gradually become a challenge that needs to be solved. Although the decentralized and open and transparent characteristics of blockchain improve the security and reliability of the system, it also brings the risk of privacy disclosure. For example, the transparency of sensitive information such as transaction amount and contract content in the peer-to-peer trusted transaction mechanism makes this information likely to be used by criminals, such as attackers may obtain sensitive information such as users' medical records and addresses through vulnerabilities in the medical blockchain system.

Therefore, how to effectively protect the information security and anonymity of users while ensuring the transparency and decentralization of the blockchain system has become a research hotspot in the current blockchain field.

2. Privacy issues in blockchain transactions

Blockchain ensures that data is not tampered with through cryptography technology. In blockchain transactions, the transaction information contained in a transaction is divided into two parts: account address and transaction data.

Among them, the account address part is linked to the real identity of the user, the exposure of the account address may lead to privacy disclosure, usually we use cryptographic-based obfuscation technology to protect.

The protection of the transaction data portion varies from case to case. In general, in the field of digital currencies, homomorphic encryption is commonly used to ensure data confidentiality and exploit its characteristics to verify the balance of

Copyright: © 2024 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

transaction amounts; In the field of medical chain, it is common to use attribute-based encryption technology for protection schemes.

Using Ethereum as an example, we analyze the privacy of blockchain transactions. Suppose there is an observer whose purpose is to analyze a particular Ethereum user or group of people using Ethereum, be it a security contractor, a researcher, or a rogue looking for vulnerabilities.

2.1. Transaction data privacy issues

Ethereum's transaction data is kept on the blockchain, and each transaction contains the originator, the recipient, and the amount of ether traded. Since both the input (where the money came from) and the output (where the money went) of the transaction are recorded on the blockchain, each transaction is traceable. The input and output of transactions are recorded through the interaction between smart contracts and accounts.

In Ethereum, due to the transparency of the blockchain, the observer can get two kinds of information from it: (1) The use of any funds.

In Ethereum, account funds (typically ether or ERC-20 tokens) are derived from "block rewards" (known as "mining") and can also be transferred through transactions and smart contracts. Since every transaction and smart contract execution is recorded on the blockchain, by analyzing this publicly available data, observers can trace the path of any money transfer. In particular, the execution of smart contracts leaves a detailed log in the transaction record, so that every change in state can be tracked.

2.1.1. Transactions related to any Ethereum address

Since all the sending and receiving addresses are recorded in detail in every Ethereum transaction, by retrieving a specific address, all relevant transaction records are found on the blockchain. In particular, by deploying smart contract addresses, observers can also track complex contract interactions, such as contract execution, state changes, and so on.

Through the above information, the observer can analyze the address with certain relevance and get a series of privacy-related information:

2.1.2. Money connections between different addresses

By analyzing the transaction record information obtained by the DeFi platform or by other means, the observer can find the law of the flow of funds inside the platform and the correlation between addresses to a certain extent. For example, criminals find the operation time of a fixed business on a platform, multiple addresses belong to the same user, etc., to be used for the detection of illegal acts, analysts analyze specific smart contract addresses and fund flows to identify potential abnormal activities, and lock the identity of illegal persons by analyzing the flow of illegal funds after the theft.

2.1.3. Track special deals

As mentioned in the previous article, for large transactions or transactions suspected of malicious behavior, you can track the flow of funds by monitoring their subsequent transactions. For example, analyzing smart contracts or pools of funds involved in hacking in Ethereum can trace subsequent transactions to those addresses to locate the bad guys or freeze stolen funds.

2.1.4. Discover trading patterns

Ethereum's smart contracts introduce a more diverse trading model. For example, by analyzing the characteristics of large ETH or token transactions, it is possible to summarize the smart contract operations involved (such as the transfer of funds from liquidity pools, repayment of loan agreements, etc.). These trading rules can not only help to find potential loopholes in the system, but also provide a breakthrough for illegal personnel to steal and other behaviors.

2.2. User identity privacy issues

Observers can infer a user's private information from the correlations that exist between different blockchain transactions. Since all transactions on the blockchain are stored in a public global ledger, observers can easily obtain transaction information for any transaction. By analyzing the correlation between transactions, the observer can effectively deduce the user's identity and other information, thus rendering the anonymization measure ineffective.

In the case of Ethereum, some design issues could cause observers to push out user identities:

2.2.1. Smart contracts and multi-signature wallets

In Ethereum, it is possible for multiple input or contract addresses to belong to the same set of users. For example, there may be multiple addresses belonging to the same multi-signature wallet or contract address, then they will most likely be controlled by the same entity; Multi-signature wallets require signatures from multiple addresses to execute transactions, and these incoming signed addresses are most likely initiated and controlled by the same user.

2.2.2. The internal state of the contract invocation

Observers can infer the relationship between some addresses by analyzing the time stamp and transaction frequency of contract calls. For example, the interaction between the creator and the user of a smart contract reflects the user's behavior pattern and the flow of money to some extent. If a contract has multiple addresses as inputs or outputs, those addresses are also likely to belong to the same user. Based on the above information, the observer can more accurately know the user's behavior or the use of funds.

2.2.3. Change address

Blockchain transactions often have a change address that is used to process change. During the transaction, if more than the required amount is paid, the change address accepts and stores the excess and sends it back to the sender. In the case of Ethereum, this process is automatically handled by the e-wallet, which includes transaction association and address association. When the observer performs feature extraction, it can analyze the entity represented by its clustering.

Features such as these, derived from the design of various virtual currencies, can be used by observers to discover correlations between different addresses and even deduce the true identity of users, rendering anonymization measures ineffective. For example, as early as 2013, Meiklejohn et al. pointed out that taking Bitcoin as an example, the use of heuristic cluster analysis technology to analyze user transaction data can effectively identify whether different addresses belong to the same user [1]. Alternatively, clustering algorithms such as K-means and DBSCAN can be used to identify addresses belonging to the same user to determine the identity of traders [2, 3]. Most of the relevant studies take Bitcoin as an example, but the research results are often universal to most virtual currencies.

In addition to the design characteristics of virtual currency, the behavior law of users when using virtual currency will also become the reason for the failure of anonymous means:

Since all transaction information on the blockchain is available in the global ledger, it is possible to infer the real-world activity of the corresponding trader from the transaction record. For example, the transaction time of mobile phone card monthly rent, rent, water and electricity bills are relatively fixed, and the transaction amount has a certain regularity. This regular consumption reflects the user's behavior characteristics, income level and other information, and the observer can make a user portrait of the anonymous user through the information reflected in the transaction data. Through user profiles, anonymous accounts can be matched with real identities.

For this problem, Monaco proposed a method to analyze user portraits and identify user identities through up to 12 parameter dimensions [4]. Through a six-month experiment, Monaco found that the accuracy rate of his method was as high as 62% and the error rate was less than 11%.

3. Privacy protection technology

In view of the privacy issues involved in the above blockchain, this article will describe the current mainstream technology for privacy protection through two aspects of data protection and identity protection. Among them, the protection of transaction data mainly uses homomorphic encryption and attribute-based encryption, and the protection of user identity mainly includes mixed coin technology, ring signature technology and zero-knowledge proof.

3.1. Coin mixing technology

Coin mixing technology increases the difficulty of tracing the whereabouts of digital currency by mixing several unrelated inputs before output, so as to achieve the purpose of protecting user identity information [5]. According to the implementation plan, it is mainly divided into two kinds: centralized mixing and decentralized mixing.

3.1.1. Centralized coin mixing mechanism

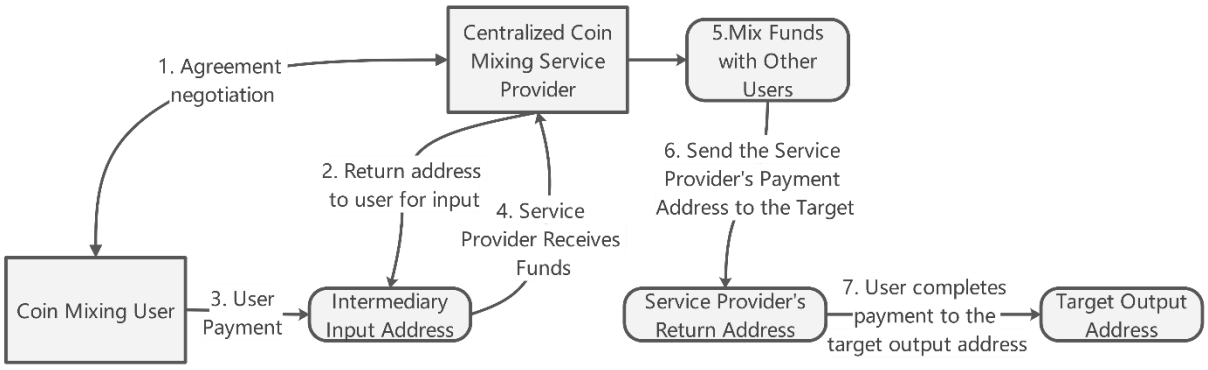

Figure 1. The basic process of centralized coin mixing mechanism (Photo credit: Original).

The service of centralized coin mixing is provided by a third-party provider, as shown in the figure 1. Each sender first sends the transaction content and funds that need to be mixed to the coin mixing node for coin mixing operation, and the coin mixing node scramps and mixes the funds received from each sender according to its coin mixing rules, and sends the amount required by each recipient to the designated receiver. Since the sender sends the funds to the node, this process protects the identities of both parties to the blockchain transaction to some extent.

However, this mechanism also has some shortcomings, the most direct is that the third-party node may not be reliable, there is a risk of information leakage and even stolen funds.

As for the credibility problem, the key is to ensure the transparency and accountability of the coin mixing operation. In this regard, Bonneau et al. pioneered the introduction of the ledger commitment mechanism in 2014 and proposed the Mixcoin protocol (which is also the source of the term "mixed coin") [6]. The biggest feature of the Mixcoin protocol is the use of accountability mechanism to punish the theft of the mixcoin node, and the use of random fees to avoid tracing the user's identity through fees.

However, this method still cannot avoid the mixed currency node itself. Since the coin mixing node has the ability to know all the input and output addresses that pass through it and record which users participate in the coin mixing process, it is still possible for the coin mixing node to track and crack the coin mixing process by analyzing the behavior pattern of these addresses and metadata such as transaction amount and time.

In response to this situation, Valenta et al. improved and proposed the Blindcoin scheme the following year [7]. This scheme adds blind signature on the basis of Mixcoin scheme, so as to achieve the effect that the mixing node can still carry out mixing operation without knowing the mapping relationship of input and output addresses, which avoids the problem mentioned above that the mixing node of Mixcoin protocol can crack the mixing process through the mapping relationship of addresses.

However, the use of blind signatures by Blindcoin requires an additional means of verifying that the node has actually completed the blending, so the scheme requires each user to send their own signature, generated by the blind signature mechanism, to a public log to prove that the user participated in a blending process. This makes it possible to analyze transaction patterns by signature, i.e., breaking the indistinguishable nature of mixing.

Therefore, in order to more thoroughly solve the problems caused by the centralized coin design itself, the solution of decentralized coin mixing came into being.

3.1.2. Decentralized coin mixing mechanism

Compared with the centralized coin mixing, the decentralized coin mixing operation process does not involve the participation of a central server, and the transaction participants reach an agreement among themselves to carry out the coin mixing transaction, thus confusing each other's identities in the process.

As shown in the figure 1-2, the process of decentralized coin mixing by three users is given as an example: First, the protocol negotiation is carried out by the users who intend to mix coins, and a smart contract is generated. The user deposits the funds to be mixed into the smart contract address, and formulates the address (output address) of the funds after receiving the mixed coins according to the protocol or algorithm. Then, the protocol issues a coin mixing scheme for participants to confirm, when the scheme is confirmed, the protocol will perform the coin mixing operation, and send back the equivalent amount of money after mixing to the user from the smart contract address.

Compared with centralized coin mixing, decentralized coin mixing does not rely on a large provider, there is no central server to act as an intermediary to protect the identity privacy of traders to a certain extent, but also traders have to bear the risk of third-party coin mixing nodes involved in the coin mixing process (which may be a user as a node or use smart contracts, etc.) is unreliable. If a third-party coin-mixing node is not reliable (the user of the node is itself a malicious person) or is attacked due to insufficient security, the security of funds cannot be guaranteed.

CoinJoin is the first decentralized coin mixing scheme, proposed by Gregory in 2013. Its process is similar to the above description, that is, users who intend to trade form a group, and initiate a mixed currency transaction within the group, and then the group will mix the input addresses of all users and randomly sorted output addresses, so as to achieve the effect of anonymity. As show in the figure 2.

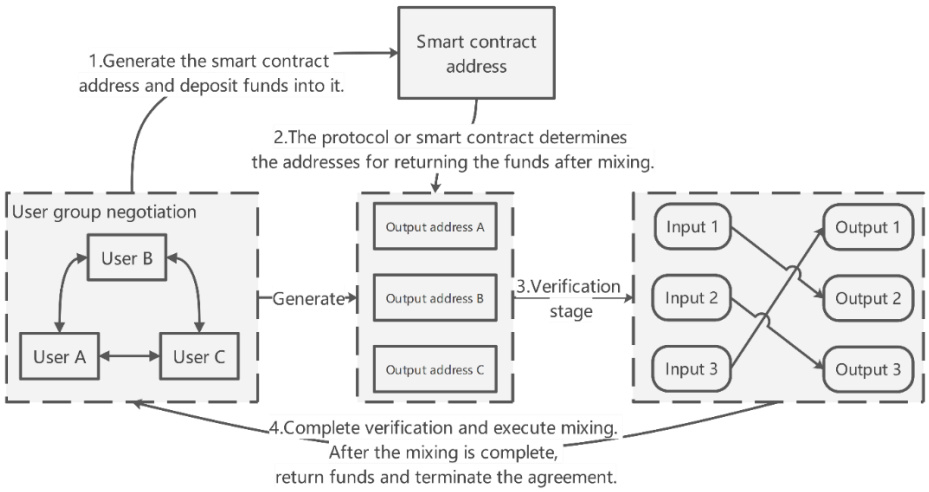

Figure 2. The basic process of decentralized coin mixing: Take three users as an example (Photo credit: Original).

However, CoinJoin has the problem that the transaction address is visible to all users in the group, for which there have been several solutions such as Coinshuffle Coinparty, Dash, etc., but they also have defects. For example, Coinshuffle uses the decryption hybrid network technology to solve this problem, but at the same time, it leads to the disadvantages of high computing cost, long time, and high requirements for the user's network environment; Coinparty uses a secure multi-party computing protocol to simulate a trusted third party to prevent malicious users from manipulating the sorting results. However, the computational load and network pressure of this protocol increase significantly with the increase in the number of users, and it is not suitable for large-scale mixing with a very high number of users. Dash protocol introduces Masternodes (master nodes) to construct mixed currency transactions. While Masternodes ensure anonymity among users, they are mixed among Masternodes to further ensure privacy. However, this process requires additional services provided by Masternodes. Therefore, the overall implementation has a high performance overhead and requires the user to submit a deposit.

In addition to CoinJoin and its enhancements, there have been other initiatives such as Tumblebit, CoinLayering, GroupShuffle, and others. However, they also have their own limitations, for example, Tumblebit has the problem of large transaction delays due to the use of multiple transaction channels to confuse transaction records and require too many trusted third parties. As show in the table 1.

Table 1. Comparative Analysis of Decentralized Coin Mixing Protocols: Enhancements and Challenges

Protocol Name	Improvement	Limitation
CoinJoin [8]	The earliest decentralized mixing protocol.	The internal anonymity of participants in transactions is insufficient.
CoinShuffle [9]	Based on CoinJoin, it introduces encryption network technology to strengthen internal anonymity.	CoinShuffle's limited concurrency causes mixing calculations to take longer and be more vulnerable to adversarial environments.
Coinparty [10]	Based on CoinJoin, it introduces multi-party computation to eliminate third-party trust assumptions.	The increase in the number of participants places additional burdens on network mixing computations, leading to inefficiency at scale.
Dash [11]	Based on CoinJoin, introduces Masternodes to improve internal anonymity, with fast transactions.	The introduction of Masternodes increases anonymity, but scaling the mix requires higher transaction costs, and running Masternodes introduces centralization risks.
Tumblebit [12]	Uses a multi-party payment channel to mix transactions, optimizing trust and the anonymity of payment steps.	The mixing process involves multiple steps and introduces trusted third parties, which are irreplaceable.
CoinLayering [13]	Uses Chinese cryptography theory and Schnorr signatures to create a mix of input and output addresses.	Lacks measures against Sybil attacks, and when multiple participants are involved, it is difficult to ensure the anonymity of the user.
GroupShuffle [14]	Uses a layered system to optimize the mixing process across multiple layers.	Requires a highly secure network and carries a higher risk of internal attacks.

From the above description and the summary of Table 1, different decentralized coin mixing technologies also have their own limitations, and they are often difficult to balance among the triadism of anonymity, performance cost, and process supervision. It can be seen that under the current technical level, it is difficult to take into account the needs of the three through a single coin mixing technology. Therefore, both centralized and decentralized coin mixing schemes have their own different limitations. However, in the current coin mixing protocol has not made a great breakthrough, we can increase the cost of malicious personnel attacks and theft through a series of cryptographic means, in order to achieve the purpose of protecting privacy and fund security.

3.2. Homomorphic encryption

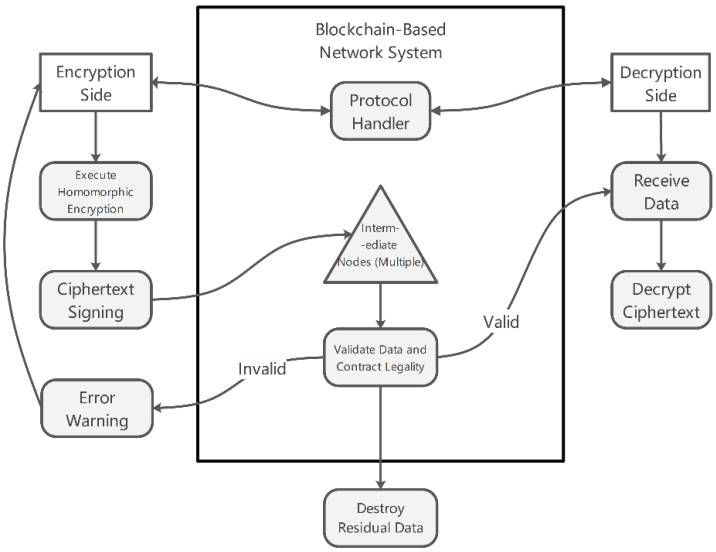

Figure 3. General flow of homomorphic encryption (Photo credit: Original).

As show in the figure 3. Homomorphic encryption is a secure computation based on a non-noisy method that allows calculations to be performed on encrypted data without access to the key, and the results of calculations using encrypted data are identical to the results of the same operations using the original data. Compared with traditional encryption methods, homomorphic encryption can not only protect the privacy in the calculation process better, but also protect the plaintext content even when the ciphertext or even the encryption algorithm is leaked due to its homomorphic nature, thus greatly improving the security and privacy of information.

There are many classical homomorphic encryption theories in blockchain, such as Pedersen commitment, ElGamal commitment, etc. These technologies often use one of homomorphic addition or homomorphic multiplication (here homomorphic addition AND multiplication refers to the AND operation and XOR operation, such as RSA algorithm is a typical homomorphic multiplication algorithm. Homomorphic addition algorithms include Paillier et al.) [15]. For example, Pedersen promised to use homomorphic addition, while ElGamal promised to use homomorphic multiplication. They used their own algorithms for privacy calculation, and used hash values to verify the validity of their results and ensure the reliability of exchange commitments in transactions.

The general process of homomorphic encryption (FIG. 3.2-1) is as follows: the algorithms of both sides shake hands and determine the initial variables and parameters required by the algorithm; Then, the encryption party performs homomorphic encryption and signature on the information. The encryption party uploards the data to the transaction node and waits for the transaction node to verify the validity of the signature; The transaction node completes the verification, the decryptor downloads the data, and each transaction node deletes the residual data; Finally, the decrypter can decrypt and view the plaintext.

At present, homomorphic encryption has been used in many blockchain applications, such as smart contract processing mechanisms and blockchain-based distributed systems, such as Hawk, Fabric, EPIoT and so on.

However, in blockchain transactions, although homomorphic encryption can effectively improve the security of transaction data and data calculation process, it cannot solve the problem of privacy disclosure in the interaction process of transaction parties and nodes. Therefore, techniques such as zero-knowledge proof are generally used in conjunction with homomorphic encryption to achieve as comprehensive privacy protection as possible.

3.3. Zero-knowledge proof

Zero-Knowledge Proof (ZKP), proposed by Goldwasser et al. In 1985, is a cryptographic technology that can prove the ownership of information without disclosing sensitive data, which can solve the problem of privacy disclosure of on-chain data in the field

of blockchain [16, 17]. Achieve a good balance between privacy protection and data transparency. From its characteristics, it is not difficult to find that using zero-knowledge proof can make up for the shortcomings of homomorphic encryption technology.

Through zero-knowledge proof, Miers et al. proposed zerocoin, a blockchain-based digital currency scheme, in 2013, which is the earliest application of zero-knowledge proof in digital currency, and it can achieve complete anonymity of users in digital currency transactions [18]. However, due to the problem that this scheme exposes the payment target and amount, Sasson, in cooperation with Miers and others, proposed an improved version of zerocoin based on general ledger the following year based on the idea proposed at the same time as zerocoin scheme [19, 20]. In 2016, zcash, a practical application of the scheme, was launched.

Zerocash introduced zk-SNARK (zeroknowledge Succinct Non-interactive arguments), the algorithm most widely used in blockchain for zero-knowledge proof [21]. The most significant feature of this algorithm is that the proof of transaction content can be completed without interaction. The zk-SNARK verifies the correctness of each value of the transaction content, and each network node can prove the correctness of the transaction without interaction to ensure that its specific content will not be leaked, that is, data such as the transaction amount can be hidden.

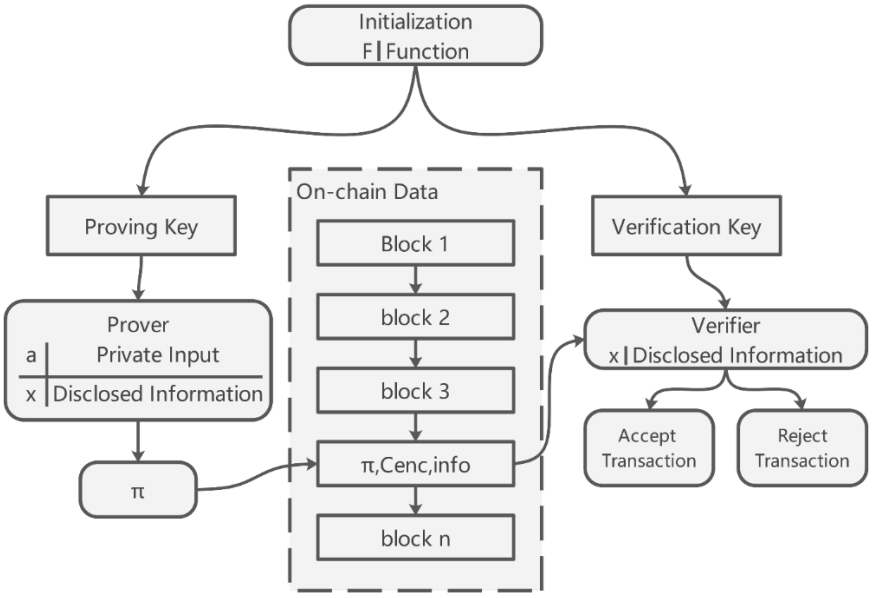

Figure 4. Implementation of ZK-SANRK in Zcash (Photo credit: Original).

What is shown Figure 4 is an implementation of ZK-SNARK in zcash: a Trusted Setup is performed to generate a proof key and an authentication key before execution. The prover (that is, the transaction sender) uses the proof key to privately enter information such as the transaction amount, the transaction party, and disclose the proof of information production, which is denoted here as π. The prover will chain π, the verifier (i.e., the transaction receiver) will pull π and verify it using the verification key, and depending on the success of the verification, the zcash wallet will accept or reject the transaction.

However, since zk-SNARK requires the Common Reference String (CRS) to be trusted before execution, if this process is carried out in an insecure environment or the participants are not trusted, the whole proof system will be exposed to serious security vulnerabilities. Therefore, Sasson et al. also proposed an improved zk-STARK algorithm based on trusted Settings in 2018, and introduced the hypothesis of anti-strong hash collision in the design idea, which greatly improved the arithmetic complexity of the verifier [22].

The PGC scheme proposed by Chen et al. is oriented to the design of blockchain electronic currency based on the account model [23]. This scheme uses the Decentralized Confidential Payment system (DCP), which retains the advantages of zero-knowledge proof. Combined with Twisted ElGamal homomorphic encryption algorithm to encrypt the transaction content, well shows the comprehensive application of zero-knowledge proof and homomorphic encryption.

More radical, such as zk-SNARK based Supersonic algorithm, chooses not to use trusted Settings in the case of instantiated with class groups, and achieves logarithmic complexity in both polynomial size and verification time [24]. And the data generated during the proof process has a much smaller space footprint than zk-SNARK.

Each of these schemes and their directions has its own room for improvement. For example, Supersonic algorithm brings extra performance cost, and Twisted ElGamal, a homomorphic algorithm combined with PGC scheme, has problems such as temporarily unable to realize transaction verification based on UTXO model. How to achieve the optimal performance cost while balancing transaction transparency and content privacy is the focus of current research.

3.4. Ring signature

Ring Signature was first proposed by Rivest et al in 2001, and its earliest application in blockchain was the electronic currency system CryptoNote v2.0 proposed by Saberhagen and in 2013 [25, 26]. zerocoin mentioned above (3.3) also has an application [27].

Ring signatures are developed from Group signatures. A group signature allows all members of a group to sign a message. External members can only be sure that the signature is from a certain group, but not the specific signer. However, group signature requires a trusted group manager to verify and manage member identities, which is a centralized design idea, while ring signature does not require a group manager to manage group members, and the public key generation process does not require coordination among members.

Ring signature Select any group of signers (you can also select yourself as one of them) from the group, and combine your private key with the public key of the selected member to sign the message. The authenticator can use the public key and ring signature to verify the signature without knowing the specific user who signed the message.

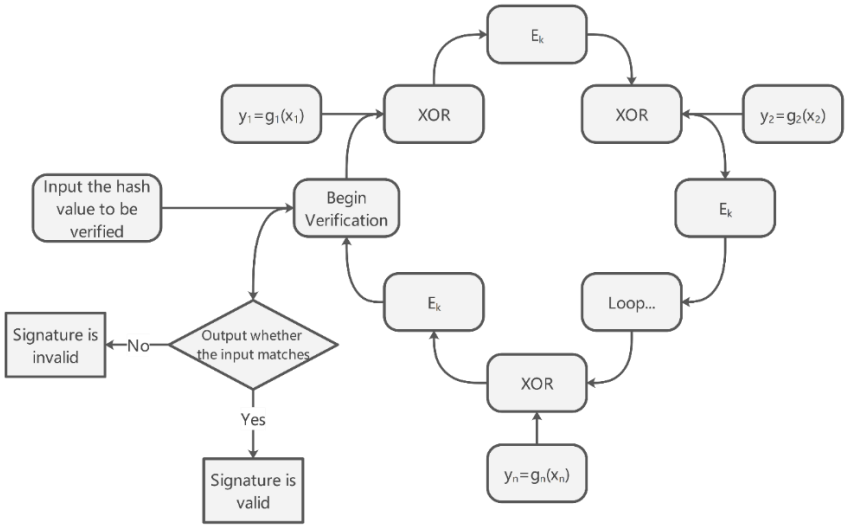

Figure 5. Basic flowchart of ring signature (Photo credit: Original).

The logical structure of a ring signature is shown in the Figure 5, which is similar to a ring, which is why it is named ring signature. We input the hash value V of the message to be verified, and Xor it with the public key of one participant, and then put the result into the link function Ek for calculation. The result of the calculation is followed by the same process for the next participant, and so on, when the public keys of all participants are calculated, if the signature is valid, the result should return to V.

In this way, the previously mentioned CryptoNote, combined with a one-time public key scheme, uses a one-time public key to generate a receiving address for the receiver to ensure that the receiver uses different outputs in different transactions, thus making it more difficult for an attacker to distinguish the same receiver from different transaction outputs.

However, a ring signature allows all members in a group to use their own private keys and the public keys of other members to create a ring signature. The specific members participating in the signature are not disclosed. Although this can confirm to the verifier that someone has completed the transaction without disclosing who has completed the transaction, in order to achieve the anonymity of the transaction, this mechanism that any user can create a group and sign the message anonymously is highly likely to be exploited by malicious users.

In response to this situation, Fujisaki et al proposed an improved scheme of ring signature with trace in 2007 [28]. The solution introduces a tagging mechanism that lists all group members and group issues. Each member can only sign a tag once, and if it is signed multiple times, the member's identity will be published immediately. This scheme is used in CryptoNote, and further turns the ring signature itself into a one-time, that is, a one-time ring signature, which further guarantees the anonymity of the transaction initiator.

In addition, Noether et al. also proposed an extended RingCT (Ring Confidential Transactions) applicable to CryptoNote in 2016 [29]. It combines ring signatures with Pedersen's commitment to both anonymity of identity and concealment of transaction content. This scheme was first implemented in Monero in January 2017 and has since become a mandatory feature for all Monero transactions [30].

Monero is currently the most popular electronic currency based on the CryptoNote protocol. The most significant feature of Monero is that any transaction can be verified by other members, which effectively prevents multiple payments or fake payments, making transactions conducted with Monero more reliable.

However, from the principle of ring signature, we can find that the anonymity intensity of ring signature largely depends on the number of members participating in the transaction and forming the "ring". Only when there are enough members, individual members can be effectively hidden in the whole. However, with the increase of members, the size of the signature generated by the ring signature will also increase in a linear way, which brings additional computing amount, but also greatly aggravate the storage pressure of each node of the blockchain.

3.5. Attribute base encryption

attribute based encryption (ABE) is an encryption technology based on the attribute identification of identity information, which was originally proposed by Amit Sahai and Brent Waters in 2005. Its core security is reflected in that only the users who meet the specific access policy can decrypt the ciphertext, while the attackers who do not meet the policy cannot decrypt even if they have the key. This allows ciphertext to be transmitted over insecure channels and uploaded to open network storage devices.

There are two main types of attribute-based encryption: Key-Policy Attribute-Based Encryption (KP-ABE, key-policy attribute-based Encryption) and ciphertext policy attribute-based encryption (CP-ABE, ciphertext) Ciphertext-Policy Attribute-Based Encryption).

Among them, for blockchain transactions, CP-ABE is more widely used, in CP-ABE, the data owner can set access policies, and only users who meet the policies can decrypt the data. As shown in the Figure 5, in general, the CP-ABe-based privacy protection scheme in the blockchain consists of four stages:

Initialization stage: According to random seeds, the main key MK and public parameter PP are generated, and the attribute set S is set, where the attribute set S contains the attributes of all users; Attribute set S and secret key MK generate multiple private keys SK; Assign the generated private key SK to each user.

Encryption stage: the data owner defines an access structure A, and inserts it into the plaintext M to be encrypted, and generates ciphertext CT combined with the public parameter PP.

Data link stage: The data owner uploards the ciphertext CT as part of the transaction data to the currently used blockchain network.

Data decryption stage: Users pull transaction information to the blockchain network, obtain ciphertext CT, and perform decryption. If the user's attributes in attribute set S conform to access structure A, private key SK can be used to complete decryption of ciphertext CT and obtain plaintext M. As show in the figure 6.

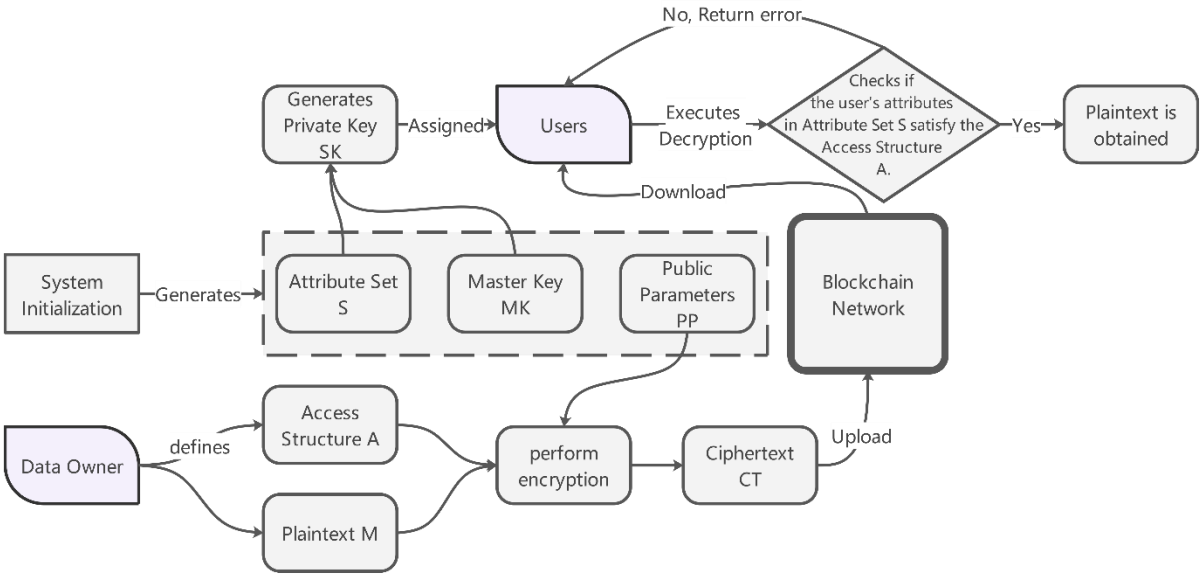

Figure 6. Basic CP-ABE execution process (Photo credit: Original).

Using CP-ABE technology, Yang et al. proposed a CP-ABE based multi-authorization access control system, and the access and privacy control mechanism of the scheme can well match the blockchain environment with fine-grained access control requirements. Rahulamathavan et al. use attribute based encryption to achieve end-to-end blockchain data privacy protection for the Internet of Things. Dai et al. proposed a data access control scheme for digital twin systems that combines blockchain and attribution-based encryption, which utilizes the decentralized characteristics of blockchain and the fine-grained access control of CP-ABE to ensure data security and reduce the single point of failure risk and storage pressure of traditional blockchain. In addition, Sasikumar et al. proposed a layered attribute based encryption scheme that combines edge computing and blockchain to further enhance data sharing and privacy protection.

However, because CP-ABE's access control relies on the access tree structure of the authorization set, it can effectively protect data privacy in a small group, but it will gradually face the performance cost of encryption when the application scale is expanded. For example, the computing process of attribute-based encryption technology requires a lot of bilinear mapping operations, which leads to a long time of encryption calculation. At the same time, the length of the attribute-based encrypted ciphertext will increase with the number of attributes, causing serious storage pressure in the blockchain. Therefore, optimizing the time and space of attribute-based encryption to improve the access and control performance of blockchain is the focus of current research.

4. Future outlook

Through the various technologies introduced in this paper and the various improvements in their development, it can be seen that in the development process of blockchain technology, the balance between the privacy protection of users and the performance cost and operation efficiency of the system has always been the general direction of research. Combined with coin mixing technology, cryptographic-based content hiding, network communication protection and other technologies, the security of the system can be improved from multiple angles, and different technologies can complement each other as much as possible. However, in this process, due to the protocol design and other reasons, the limitations of various schemes inevitably appear, and its improved model continues to lead to new problems. For example, on the development road of the mixed coin protocol, in order to further enhance the anonymity, various decentralized improvements have been produced, but these improvements also lead to the regulatory problems of the transaction link. Even the participants themselves may have malicious actors, and in order to prevent malicious behavior as much as possible while ensuring anonymity, the new improved model creates more performance overhead and inefficiency issues...

Therefore, while improving the algorithms, links and schemes of the technologies themselves, the application research focus is bound to strive to achieve a proper balance between the efficiency and performance costs of the system, the anonymity and privacy protection of participants, the supervision and security of the transaction links, so as to cope with the application of most scenarios. At the same time, on the basis of this comprehensive technology, various optional protection measures are provided to deal with a few special application scenarios, such as extreme anonymity and supervision can accept greater performance overhead, while ensuring that participants are reliable and strive for efficiency can sacrifice some anonymity.

Based on this point of view, this paper believes that future research may be promoted in the following two aspects:

4.1. Design solutions according to different data sensitivities

The application scenarios represented by digital currencies have high requirements for security and availability. Among the solutions for these scenarios, some of the privacy protection mechanisms use cryptography techniques such as zero-knowledge proof and ring signature to address privacy threats, but they also introduce new security issues of their own. For example, the ZK-SNARK adopted in the Zerocash scheme requires a third party to set up a trusted setting, so the credibility of the third party directly determines the security of the system, while its improved version increases the performance cost to varying degrees, or cannot achieve privacy protection in specific scenarios. Therefore, in the future, data sensitivity will be classified according to actual application scenarios, solutions will be studied for scenarios with different data sensitivity, and different indicators will be formulated for different data sensitivity, that is, different focuses on system cost, privacy protection and security supervision will be formulated and developed accordingly, so as to achieve a compromise between performance and privacy protection. And when the solution is turned into a product, they are compatible with each other as a set of systems (for example, a transaction can choose a component that enhances anonymity but is more expensive, or it can not be used), and can be deployed and run simultaneously. This may be a safer and more effective direction.

4.2. Multi-party supervision technology in scenarios with high data sensitivity

In application scenarios such as the financial chain, multi-party supervision is essential to maintain financial order and reduce market risks, and this scenario belongs to the data sensitivity classification mentioned above should be classified as highly data sensitive scenarios. Currently, most blockchain systems only support unilateral supervision (that is, the "trusted third party" mentioned in most schemes is often the only one in practice), and the security problems in many protocols and solutions come precisely from the unreliability and abuse of power of the regulator, and this model also has the risk of a single point of failure. The introduction of multi-party supervision can allocate regulatory power to multiple trusted institutions through access control, secret sharing, secure multi-party computing and other schemes to reduce security risks through mutual supervision and decentralized functions. Under this scheme, since regulation is not the main link that brings performance overhead, the overall operating efficiency of the blockchain system will usually not change much, but will sacrifice some flexibility on the process. But this direction of development has great potential in areas related to national security, as well as the digital currency that the government may launch in the future.

5. Conclusion

This paper has explored the intricate dynamics of privacy protection within blockchain transactions, delving into the dual aspects of data privacy and identity privacy. We have systematically dissected the privacy challenges posed by blockchain's inherent transparency, using Ethereum as a primary example to illustrate how transaction data and user identities can be exposed. Through our examination, various technologies such as homomorphic encryption, attribute-based encryption, and zero-knowledge proofs have been evaluated for their effectiveness in safeguarding privacy while maintaining the decentralized essence of blockchain. Our exploration revealed that while these technologies provide robust mechanisms to shield user data and identities, they each carry inherent limitations that could hinder their practicality or effectiveness. For instance, while homomorphic encryption offers comprehensive data privacy, it is computationally intensive and does not address privacy leaks during transaction interactions. Similarly, zero-knowledge proofs, despite their efficacy in ensuring transaction anonymity, require complex and often costly setups that may not be feasible in all operational contexts.

The balance between user privacy, system efficiency, and regulatory compliance presents a fertile ground for further research. Future studies could focus on developing hybrid systems that integrate the strengths of various privacy-preserving technologies while mitigating their weaknesses. Such integrations could potentially offer more scalable and efficient solutions without compromising on privacy or security.

References

[1] Meiklejohn, S., Pomarole, M., Jordan, G., Levchenko, K., McCoy, D., Voelker, G. M., & Savage, S. (2013). A fistful of bitcoins: Characterizing payments among men with no names. In Proceedings of the 2013 Conference on Internet Measurement Conference (pp. 127–140).

[2] Monamo, P., Marivate, V., & Twala, B. (2016). Unsupervised learning for robust Bitcoin fraud detection. In 2016 Information Security for South Africa (ISSA) (pp. 129–134). IEEE.

[3] Patil, S., Nemade, V., & Soni, P. K. (2018). Predictive modelling for credit card fraud detection using data analytics. *Procedia Computer Science*, 132, 385–395.

[4] Monaco, J. V. (2015). Identifying bitcoin users by transaction behavior. In Biometric and Surveillance Technology for Human and Activity Identification XII (Vol. 9457, pp. 25–39). SPIE.

[5] Ruffing, T., & Moreno-Sanchez, P. (2017). Mixing confidential transactions: Comprehensive transaction privacy for Bitcoin. IACR Cryptology ePrint Archive, 2017, 238–260.

[6] Bonneau, J., Narayanan, A., Miller, A., Clark, J., Kroll, J. A., & Felten, E. W. (2014). Mixcoin: Anonymity for bitcoin with accountable mixes. In Financial Cryptography and Data Security: 18th International Conference, Christ Church, Barbados, March 3–7, 2014, Revised Selected Papers (pp. 486–504). Springer Berlin Heidelberg.

[7] Valenta, L., & Rowan, B. (2015). Blindcoin: Blinded, accountable mixes for bitcoin. In Financial Cryptography and Data Security: FC 2015 International Workshops, BITCOIN, WAHC, and Wearable, San Juan, Puerto Rico, January 30, 2015, Revised Selected Papers (pp. 112–126). Springer Berlin Heidelberg.

[8] Maxwell, G. (2013). CoinJoin: Bitcoin privacy for the real world. Bitcoin Forum.

[9] Ruffing, T., Moreno-Sanchez, P., & Kate, A. (2014). Coinshuffle: Practical decentralized coin mixing for bitcoin. In Computer Security—ESORICS 2014: 19th European Symposium on Research in Computer Security, Wroclaw, Poland, September 7–11, 2014, Proceedings Part II (pp. 345–364). Springer International Publishing.

[10] Ziegeldorf, J. H., Grossmann, F., Henze, M., Inden, N., & Wehrle, K. (2015). Coinparty: Secure multi-party mixing of bitcoins. In Proceedings of the 5th ACM Conference on Data and Application Security and Privacy (pp. 75–86).

[11] Duffield, E., & Diaz, D. (2015). Dash: A privacy-centric cryptocurrency.

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

[13] Lu, N., Chang, Y., Shi, W., & Choo, K. K. R. (2020). CoinLayering: An efficient coin mixing scheme for large scale bitcoin transactions. *IEEE Transactions on Dependable and Secure Computing*, 19(3), 1974–1987.

[14] Yan, Y., Liu, Q., & Li, J. (2024). Blockchain collaborative coin mixing scheme based on hierarchical mechanism. *International Journal of Network Security*, 26(3), 442–453.

[15] Goldwasser, S., Micali, S., & Rackoff, C. (2019). The knowledge complexity of interactive proof-systems providing sound foundations for cryptography on the work of Shafi Goldwasser and Silvio Micali (pp. 203–225).

[16] Choc, B., Goldwasser, S., Micali, S., & Awerbuch, B. (1985). Verifiable secret sharing and achieving simultaneity in the presence of faults. In Annual Symposium on Foundations of Computer Science (Proceedings) (pp. 383–395).

[17] Wang, R., Zhu, J., Wang, S., Wang, T., Huang, J., & Zhu, X. (2024). Multi-modal emotion recognition using tensor decomposition fusion and self-supervised multi-tasking. *International Journal of Multimedia Information Retrieval*, 13(4), 39.

[18] Ben-Sasson, E. (2013). Universal and affordable computational integrity. Bitcoin.

[19] Sasson, E. B., Chiesa, A., Garman, C., Green, M., Miers, I., Tromer, E., & Virza, M. (2014). Zerocash: Decentralized anonymous payments from bitcoin. In 2014 *IEEE Symposium on Security and Privacy* (pp. 459–474). IEEE.

[20] Ben-Sasson, E., Bentov, I., Horesh, Y., & Riabzev, M. (2018). Scalable, transparent, and post-quantum secure computational integrity. Cryptology ePrint Archive.

[21] Bünz, B., Fisch, B., & Szepieniec, A. (2020). Transparent SNARKs from DARK compilers. In Advances in Cryptology—EUROCRYPT 2020: 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10–14, 2020, Proceedings Part I (pp. 677–706). Springer International Publishing.

[22] Rivest, R. L., Shamir, A., & Tauman, Y. (2001). How to leak a secret. In Advances in Cryptology—ASIACRYPT 2001: 7th International Conference on the Theory and Application of Cryptology and Information Security, Gold Coast, Australia, December 9–13, 2001, Proceedings (pp. 552–565). Springer Berlin Heidelberg.
[23] Van Saberhagen, N. (2013). CryptoNote v 2.0.
[24] Fujisaki, E., & Suzuki, K. (2007). Traceable ring signature. In International Workshop on Public Key Cryptography (pp. 181–200). Springer Berlin Heidelberg.
[25] Zhu, X., Guo, C., Feng, H., Huang, Y., Feng, Y., Wang, X., & Wang, R. (2024). A Review of Key Technologies for Emotion Analysis Using Multimodal Information. *Cogn Comput*, 1(1), 1–27.
[26] Akcora, C. G., Gel, Y. R., & Kantarcioglu, M. (2022). Blockchain networks: Data structures of bitcoin, monero, zcash, ethereum, ripple, and iota. *Wiley Interdisciplinary Reviews Data Mining and Knowledge Discovery*, 12(1), e1436.
[27] Pedersen, T. P. (1991). Non-interactive and information-theoretic secure verifiable secret sharing. In Annual International Cryptology Conference (pp. 129–140). Springer Berlin Heidelberg.
[28] ElGamal, T. (1985). A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31(4), 469–472.
[29] Ruffing, T., & Malavolta, G. (2017). Switch commitments: A safety switch for confidential transactions. In Financial Cryptography and Data Security: FC 2017 International Workshops, WAHC, BITCOIN, VOTING, WTSC and TA, Sliema, Malta, April 7, 2017, Revised Selected Papers (pp. 170–181). Springer International Publishing.
[30] Kosba, A., Miller, A., Shi, E., Wen, Z., & Papamanthou, C. (2016). Hawk: The blockchain model of cryptography and privacy-preserving smart contracts. In 2016 *IEEE Symposium on Security and Privacy* (SP) (pp. 839–858). IEEE.