A Survey on Zero-Knowledge Proof in Blockchain

Xiaoqiang Sun, F. Richard Yu, Peng Zhang, Zhiwei Sun, Weixin Xie, and Xiang Peng

Abstract

Blockchain, which is usually regarded as a public, decentralized and distributed ledger, has attracted significant attention recently. In the environment of blockchain, all historical transaction data are recorded and stored. However, because blockchain is open and transparent, a malicious user may illegally access private transaction data, including transaction amount, account address, and account balance. As a cryptographic technique, zero-knowledge proof (ZKP) can be used to verify whether the prover has enough transaction amount in the environment of blockchain without leaking any private transaction data. This article provides a comprehensive survey on ZKP in the environment of blockchain with the aim of highlighting security problems and challenges. It first discusses the framework, models and applications of ZKP. Next, it provides an introduction of blockchain, and proposes a framework of ZKP in the environment of blockchain. Then, it highlights the current state of ZKP in the environment of blockchain. Finally, it identifies some potential problems and future research directions.

Introduction

Based on peer-to-peer network, cryptographic algorithms and consensus mechanism, blockchain maintains a decentralized, irrevocable and verifiable ledger, which is used to archive all historical transaction data on the blockchain. Generally, transaction data include implementation details of the transaction, such as transaction amount, account address and account balance, which are individual privacy. Due to the openness and transparency of blockchain, anyone can access archived transaction data, so there exist various security and privacy challenges.

There exist several cryptographic approaches, including homomorphic encryption [1], ring signature [2], secure multiparty computation [3], and zero-knowledge proof (ZKP) [4]. Homomorphic encryption supports certain types of operations to be carried out on the ciphertext without decryption. It can be used to protect the security of the account balance and transaction amount. Unfortunately, it cannot ensure the security of the account address. Ring signature is a special digital signature, which does not reveal who signed it. Ring signature can be used to protect the security of the account address. However, it cannot guarantee the security of the account balance and transaction amount. Secure multiparty computation is a cryptographic protocol, which distributes a computation task across multiple participants, where no individual participant can know the other participants' data. It can protect the security of the account balance and transaction amount. But, it cannot protect the security of the account address. In addition, homomorphic encryption, ring signature, and secure multiparty computation cannot be used to verify whether the prover has enough transaction amount in the environment of blockchain without leaking the transaction amount, account address, and account balance.

ZKP is an interactive verification protocol. In this protocol, based on the execution of a sequence of predefined actions, the verifier can be convinced that the prover owns some secret data without leaking any private information, including the prover's data, the prover's identity, and the verifier's identity. The verifier only knows the fact that the prover owns these data. The implementation of this protocol does not need a complicated public key and its repeated implementation is not helpful for the malicious user to obtain additional useful information. ZKP is helpful for the implementation of anonymous verifiable voting, secure exchange of digital assets, secure remote biometric authentication, and secure auction.

Consequently, the verifier can use ZKP to verify whether the prover has enough transaction amount in the environment of blockchain without leaking any private transaction data. The verification process is described as follows. ZKP first generates a proof which always contains the prover's claim that he has enough transaction amount. Then, the generated proof is transmitted to the verifier. When the verifier receives this proof, it performs some predefined computations on this proof, and outputs a final computation result. Finally, the verifier can make a conclusion whether the prover's claim can be accepted. If the prover's claim is accepted, it means that they has enough transaction amount. The above verification process can be recorded on the blockchain without any falsification.

In this article, we focus on:

- Discussing the framework, models and applications of ZKP.
- Providing an introduction of blockchain, and proposing a framework of ZKP in the environment of blockchain.
- Highlighting the current state of ZKP in the environment of blockchain.
- Identifying some potential problems and future research directions.

Digital Object Identifier:
10.1109/MNET.011.2000473

Xiaoqiang Sun, Peng Zhang (corresponding author), Weixin Xie, and Xiang Peng are with Shenzhen University; Xiaoqiang Sun is also with the Shenzhen Institute of Information Technology; F. Richard Yu is with Carleton University; Zhiwei Sun is with Shenzhen Polytechnic.

0890-8044/21/$25.00 © 2021 IEEE

The article is organized as follows. The framework, models and applications of ZKP are presented in the following section. Then we discuss the introduction of blockchain and propose the framework of ZKP in the environment of blockchain. Following that we describe the current state of ZKP in the environment of blockchain. Open research problems are then analyzed. This article is concluded in the final section.

An Overview of Zero-Knowledge Proof

Here, we provide an overview of ZKP, including its framework, existing models and some applications of ZKP, which are described as follows.

Zero-Knowledge Proof Framework

As shown in Fig. 1, there are two entities in the framework of ZKP, namely the prover and the verifier. The implementation of this framework consists of following three phases.

- Witness Phase: The prover computes a proof that contains its statement. Then, the proof is transmitted to the verifier.
- Challenge Phase: The verifier asks the prover several questions.
- Response Phase: The prover answers these questions, which can be used for the verifier to accept or reject the generated proof.

In the above phases, no private data will be leaked. Furthermore, this framework has the following properties, which are described as follows.

Completeness: This property means that the verifier will always be convinced of the prover's true statement. Namely, if the prover can prove to the verifier that his statement is true, the verifier always accepts the generated proof.

Soundness: This property means that the prover cannot convince the verifier that his false statement is actually true, except for only a small probability. Namely, if the prover's statement is false, the verifier rejects the generated proof.

Zero-Knowledge: This property can be used to guarantee that the prover does not leak any useful information to the verifier. Namely, if the prover can prove to the verifier that his statement is true, the verifier learns nothing except for the fact that the prover's statement is true.

In order to better understand ZKP and its properties, we show an example in which ZKP is used to verify whether the prover owns some private information. This example can be applied for the secure authentication in the environment of blockchain. In this example, we suppose that the prover owns the secret number s. The implementation of this example includes three steps, which are described as follows:

- The prover first computes $v = s^2 \bmod n$, where $n = pq$, p and q are two large private primes,

$$\sqrt{n} \le s \le n-1,$$

  s is not equal to p or q. Based on the reasonable setup of p and q, the adversary cannot extract the private s from v. In addition, the prover selects a random integer r, where $1 \le r \le n-1$. The prover computes $x = r^2 \bmod n$. x is sent to the verifier.
- Next, the verifier selects a random bit $\alpha \in \{0, 1\}$. α is sent to the prover.

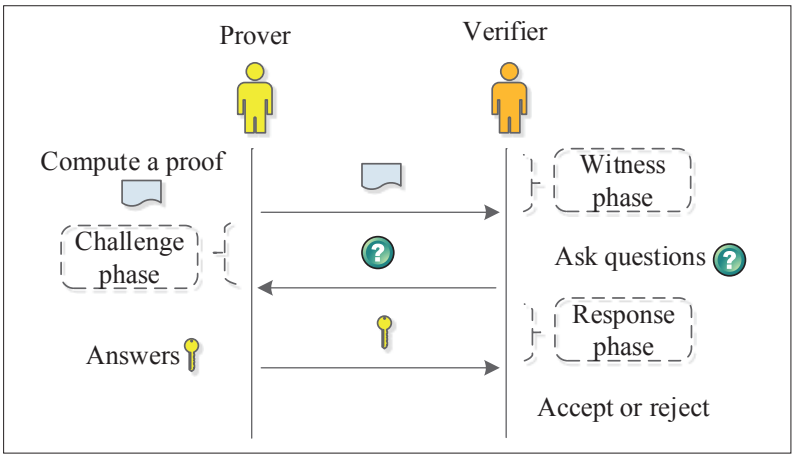

FIGURE 1. The framework of zero-knowledge proof.

- Then, if $\alpha = 0$, the prover sets the proof $\gamma = r$. If $\alpha = 1$, the prover calculates $\gamma = rs \bmod n$. The prover sends γ to the verifier. Finally, the verifier verifies γ. If $\gamma^2 = x \cdot v^{\alpha} \bmod n$, the verifier accepts γ.

This example satisfies properties of completeness, soundness, and zero-knowledge, which are described as follows. Completeness can insure that the prover can provide correct $\gamma = r$ or $\gamma = rs \bmod n$ to the verifier. Then, the verifier can verify the received γ and accept it. If the prover does not own the secret number s, soundness can guarantee that the verifier will reject the received γ with the probability $1/2$ in each verification process. If γ is verified t times, the probability that the verifier can be fooled is $(1/2)^t$. Zero-knowledge means that the verifier only knows v, x, and γ in each verification process. The verifier cannot obtain the prover's secret s.

Zero-Knowledge Proof Models

There are several ZKP models. This subsection shows existing ZKP models, which are described as follows.

zkSNARK: Zero-knowledge succinct non-interactive argument of knowledge (zkSNARK) is an improved ZKP mechanism. As shown in Fig. 2, zkSNARK mainly consists of setup, prover, and verifier [5], where setup is a procedure that generates the proving key PK and the verification key VK by using a predefined security parameter λ and an $\mathbb{F}$-arithmetic circuit C, which is a circuit that all inputs are elements in a field $\mathbb{F}$, and its gates output elements in $\mathbb{F}$. PK is used for generating the verifiable proof. VK is used for verifying the generated proof. Based on the generated PK, the input $x \in \mathbb{F}^n$ and the witness $W \in \mathbb{F}^h$, the prover generates a proof π, where $C(x, W) = 0^l$. $C(x, W) = 0^l$ denotes the output of C is 0^l. x and W are input parameters of C. n, h, and l are dimensions of x, W, and C's output, respectively. Finally, with the usage of VK, x and π, the verifier verifies π. According to the verification result, π is accepted or rejected.

Furthermore, zkSNARK has additional characteristics. First, the verification can be executed in a short running time. In addition, the proof size is just several bytes. Second, the prover and verifier are not required to communicate with each other synchronously. The only needed proof can be

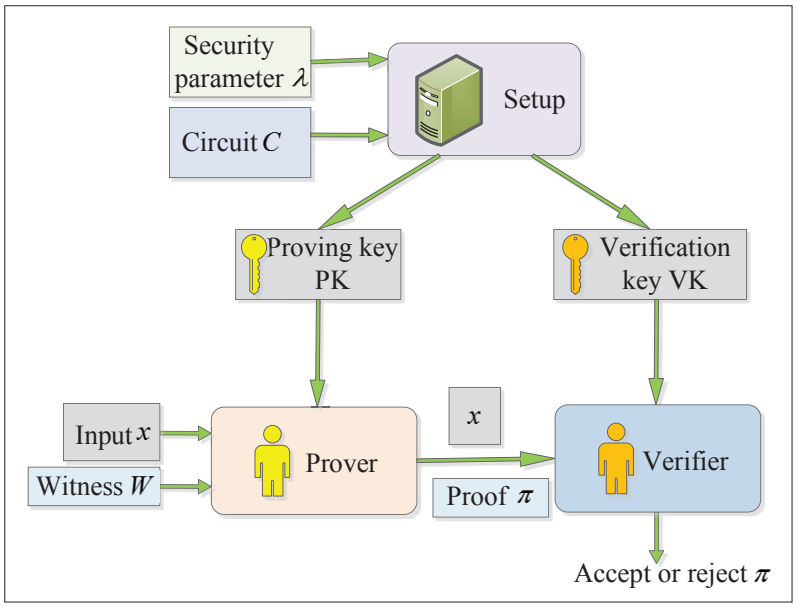

FIGURE 2. The framework of zero-knowledge succinct non-interactive argument of knowledge (zkSNARK).

verified by any verifier in a off-line way. Lastly, the prover algorithm can only be implemented in a polynomial time. Since then, there have emerged several improved zkSNARK models, which are described as follows.

Ben-Sasson's Model: Ben-Sasson *et al.* [4] first designed a new zkSNARK model for arithmetic circuits. Then, based on the proposed universal circuit generator, Ben-Sasson *et al.*, built a system, which is shown in Fig. 3. This system consists of offline phase and online phase. In the offline phase, the circuit generator takes program size bound, input size bound and time bound as inputs. The universal circuit is output. Then, the zkSNARK key generator outputs the proving key PK and the verification key VK. In the online phase, the witness map takes auxiliary input (nondeterminism), program and input as required parameters. Based on the circuit assignment and PK, the zkSNARK prover outputs the proof. The zkSNARK verifier decides to accept or reject the proof by using VK, program and input. This system supports the executions, which are implemented in a von Neumann reduced instruction set computing machine.

Ligero: Ames *et al.* [6] proposed a lightweight zero-knowledge argument model called Ligero. There is a positive proportional relationship between the communication complexity of Ligero and the square root of the size of the verification circuit. In addition, Ligero can be relied on any hash function, which can resist collision. Furthermore, Ligero can be a zkSNARK scheme in the random oracle model. This model does not need a trusted setup or public key cryptosystem. Ligero can be used for extremely large verification circuits. Meanwhile, it is suitable for medium large circuits in applications.

Bulletproofs: Based on the discrete logarithm problem, Bünz *et al.* [7] designed a novel non-interactive ZKP model called Bulletproofs. This model has a very short proof, which size is just logarithmic of the witness size. In addition, Bulletproofs does not need a trusted setup. In this model, the range proof, which size is linear, is improved significantly for secure transactions in Bitcoin or other cryptocurrencies. Furthermore, range proofs can be aggregated in Bulletproofs. Bulletproofs is particularly suitable for the distributed and trustless properties in the environment of blockchain.

Hyrax: Wahby *et al.* [8] first designed a zero-knowledge argument scheme, which has low communication, and low cost for the prover and verifier. In this scheme, there is no trusted setup in this argument. If it is applied to batched statements, the verification time has a sub-linear relationship with the arithmetic circuit size along with good constants. There is a linear relationship between the prover's running time and the arithmetic circuit size along with good constants. Based on the proposed scheme, Wahby *et al.* [8] designed an efficient zkSNARK model called Hyrax. Compared with some previous schemes,

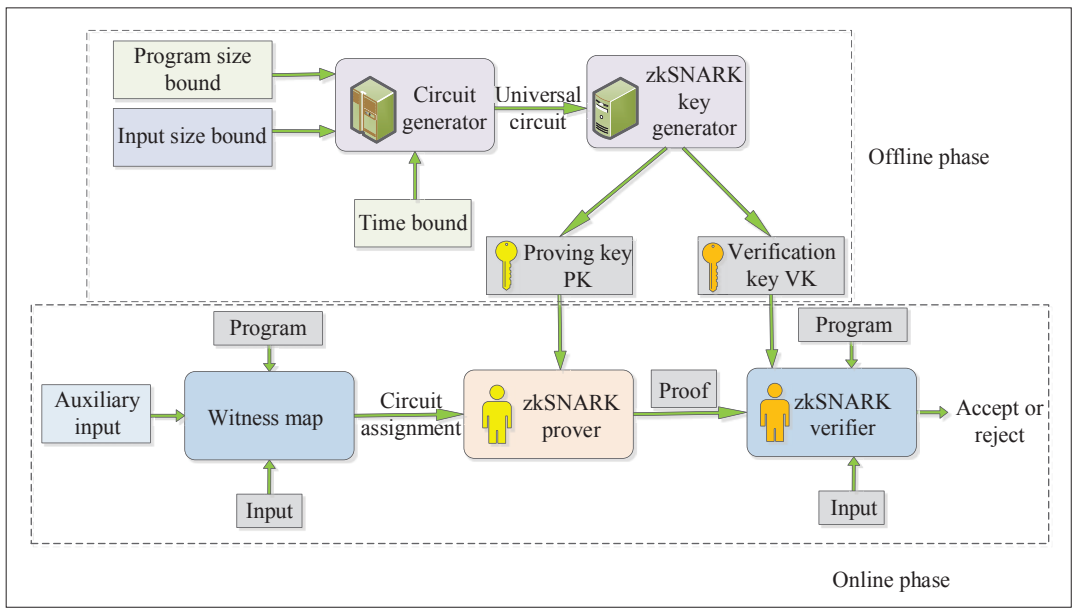

FIGURE 3. The framework of Ben-Sasson's model.

	Ben-Sasson's model	Ligero	Bulletproofs	Hyrax	Aurora	Libra
Trusted setup	Yes	N/A	N/A	N/A	N/A	Yes
Prover algorithm	$O(c \log c)$	$O(c \log c)$	$O(c)$	$O(c \log c)$	$O(c \log c)$	$O(c)$
Verification algorithm	$O(1)$	$O(c)$	$O(c)$	$O(\sqrt{s} + d \log c)$	$O(c)$	$O(d \log c)$
Proof size	$O(1)$	$O(\sqrt{c})$	$O(\log c)$	$O(\sqrt{s} + d \log c)$	$O(\log^2 c)$	$O(d \log c)$
Implementation technique	Quadratic arithmetic programs	Interactive oracle proofs	Discrete logarithm	Interactive proofs	Interactive oracle proofs	Interactive proofs

TABLE 1. Comparison of improved zkSNARK models.

Hyrax owns competitive performance of proof size and computational overhead.

Aurora: Ben-Sasson *et al.* [9] proposed a succinct and non-interactive zero-knowledge argument model called Aurora. It is suitable for the satisfiability of an arithmetic circuit. Aurora's argument size is polylogarithmic with the circuit size. In addition, Aurora has several attractive characteristics. In Aurora, there is a transparent setup. There does not exist an efficient quantum computing attack, which can crack Aurora. Furthermore, fast symmetric cryptography is used as a black box. Aurora has the optimized proof size. For example, if the security parameter is 128 bits, the proof size of Aurora is at most 250 kilobytes.

Libra: In 2019, Xie *et al.* [10] proposed a ZKP model called Libra. Libra is the first ZKP model that has linear prover time, succinct proof size and verification time. In Libra, in order to reduce the overhead on the verification, the zero-knowledge mechanism is implemented by a method, which can mask the responses of the prover with slight random polynomials. In addition, Libra needs one-time trusted setup, which only relies on the input size of the circuit. Libra has outstanding asymptotic performance and excellent efficiency of the prover. Its performance of proof size and verification time is also very efficient.

As shown in Table 1, we compare the above improved zkSNARK models' trusted setup, computation complexity of the prover algorithm, computation complexity of the verification algorithm, proof size and implementation technique, where c is the size of the log-space uniform circuit, which depth is d. s is the size of the circuit's inputs. c is usually a number of megabytes for the moderate-size circuit. Compared with c, d is negligible. Table 1 is described as follows.

In terms of trusted setup, Ben-Sasson's model needs a trusted setup for every statement. Libra only needs the one-time trusted setup that depends on s. Ligero, Bulletproofs, Hyrax and Aurora do not need the trusted setup. As for the computation complexity of the prover algorithm, Libra outperforms Ben-Sasson's model, Ligero, Hyrax and Aurora. In addition, Libra's computation complexity of the prover algorithm is irrespective of the circuit type. Bulletproofs is the only other mechanism that has the small computation complexity of the prover algorithm. As for the computation complexity of the verification algorithm, only Hyrax and Libra have computation complexities, which are poly-logarithmetic in c. Libra's computation complexity of the verification algorithm is smaller than that of Hyrax. In terms of proof size, Bulletproofs, Hyrax, Aurora and Libra have the proof sizes, which are poly-logarithmetic in c. Libra's proof size is smaller than those of Bulletproofs, Hyrax and Aurora. As for the implementation technique, Ben-Sasson's model is relied on the technique of quadratic arithmetic programs. The technique of interactive oracle proofs is used in Ligero and Aurora. Hyrax and Libra utilize the technique of interactive proofs. Bulletproofs is based on the discrete logarithm. In terms of all aspects, Libra is more efficient than other models.

Zero-Knowledge Proof Applications

Application scenarios of ZKP include anonymous verifiable voting, secure exchange of digital assets, secure remote biometric authentication, and secure auction, which are described as follows.

Anonymous Verifiable Voting: Voting is an essential component for guaranteeing democracy in a country or a stockholding company. However, the privacy of voters may be leaked during the process of voting. In addition, the voting result is difficult to be verified securely. ZKP is an available approach for the implementation of anonymous verifiable voting. Based on the use of ZKP, eligible voters can cast a ballot for showing their right without leaking their identities. Besides, ZKP allows eligible voters to request a verifiable proof that their ballots are contained in the final tally by the institution that is responsible for reporting the voting result.

Secure Exchange of Digital Assets: Digital assets are a collection of binary data, which are uniquely identifiable and valuable. If two users want to exchange their digital assets, the user's privacy, which includes identities and the content of exchanged digital assets, may be leaked in the exchange process. Based on the usage of ZKP, digital assets can be exchanged without leaking the user's privacy. In addition, ZKP generates a verifiable proof, which contains the process of the exchange of digital assets.

Secure Remote Biometric Authentication: Remote biometric authentication is a method that can be used to identify the user's access by using their biometric modalities, such as fingerprints, facial images, iris or vascular patterns. However, the user's biometric modalities may be leaked to an untrusted third party in the implementation of remote biometric authentication. This problem can be solved by using ZKP. In addition, ZKP generates a verifiable proof that includes the process of identifying the user's access.

Secure Auction: Governmental auction is an auction by which the government chooses the lowest bid from multiple suppliers, who sell their goods and services competitively. This auction includes two phases. In the first phase, multiple suppliers make bids, which are not known by the public. In the second phase, these bids are opened. The government selects the winning sup-

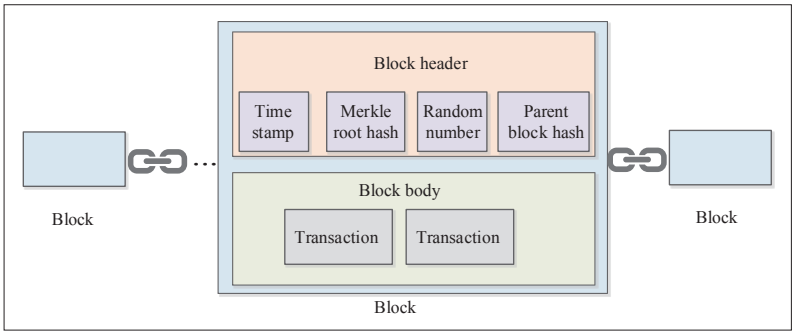

FIGURE 4. A model of blockchain.

plier, who makes the lowest bid. However, the selection of the winning supplier may leak other losing suppliers' bids and identities. ZKP can solve this problem. ZKP generates a verifiable proof for each losing supplier's bid. This proof verify that the difference between the losing supplier's bid and the winning supplier's bid is positive.

Zero-Knowledge Proof in Blockchain

In this section, we present a brief introduction to blockchain and propose a framework of ZKP in the environment of blockchain, which are described as follows.

An Introduction to Blockchain

As shown in Fig. 4, blockchain consists of a growing list of blocks, which are connected by using hash value. The block is a group of transactions, which establish a chronological sequence between them. It mainly consists of a block header and a block body. In the block header, there is a Merkle root hash, a timestamp, a random number, and a parent block hash, where the Merkle root hash represents a hash that can insure the integrity of all transactions in the block, timestamp is the current time in seconds, the random number begins from 0 and increases for each hash calculation, parent block hash is used for pointing the former block. The block body usually stores the information about transactions.

In order to implement a blockchain, necessary mechanisms include smart contract and consensus, which are described as follows. Smart contract is a computer program or transaction protocol, which can execute some predefined actions when the trigger conditions are met. If the smart contract meets the trigger conditions, it will be added into the contract queue. Then, it will be verified by periodical traversal of the state and trigger conditions. If the smart contract is verified successfully, it will be executed. In addition, smart contract can reduce the need for trusted intermediators, arbitrations cost, enforcement cost, and fraud losses.

Consensus mechanism, which is an agreement on the transaction among distributed nodes, is used for recording transactions in a blockchain. This mechanism determines the overall performance and scalability of blockchain. In addition, consensus mechanism usually supports fault tolerance, which can be used for resisting against a limited number of selfish, defective or malicious nodes. Furthermore, consensus mechanism satisfies either safety or liveness, where safety means that at least one honest node will produce a valid output and all other nodes generate or receive the same output. Liveness can guarantee that all benign nodes, which take part in a consensus mechanism, eventually generate a value and all correct requests will be processed. At present, typical consensus mechanisms include Proof of Work, Proof of Stake, and Practical Byzantine Fault Tolerance.

Blockchain has some promising characteristics, including decentralization, persistency, validity, anonymity, and auditability. Decentralization is an important feature of blockchain. The level of decentralization is related to the type of blockchain. Persistency is a feature of recorded transactions. If most nodes are not malicious, this feature can always be preserved. Furthermore, we can deduce some properties, which include immutability and transparency, from this feature. Validity is helpful for detecting transactions' malicious behaviors, double spending, and so on. In addition, this feature is usually implemented by the miner and mining instruments. Auditability means that former transactions are allowed to be verified and traced. The level of auditability relies on the category of blockchains and their executions.

The Framework of Zero-knowledge Proof in Blockchain

The architecture of ZKP in blockchain is shown in Fig. 5. There are two parts in this architecture: the onchain part and offchain part. Offchain, the prover claims that they owns enough transaction amount. The validation requester is responsible for announcing a verification task, collecting the verification result from the verifier, and paying the verification fee to the verifier. Onchain, the authenticity verification of the prover's claim is implemented by the verifier, which is usually a blockchain miner. In addition, blockchain has the incentive mechanism, which calculates the verification fee for the verifier. The implementation of this authenticity verification consists of eight steps, which are described as follows:

- Based on the the framework of zkSNARK shown in Fig. 2, trusted authority runs the setup procedure to generate the proving key and the verification key. Then, based on the usage of the proving key, the trusted authority generates a proof which always contains the prover's claim that they own enough transaction amount, rather than the transaction amount itself. The generated proof is transmitted to the prover.
- The prover transmits the generated proof to the blockchain by the Internet. Then, the proof is stored on the blockchain. The integrity and non-tamperability of the proof can be guaranteed.
- If the validation requester wants to know whether the prover owns enough transaction amount, the validation requester sends the verification task, which includes the task tag, the deadline for the verifier to make a response, and the total amount of reward for the verification task, to the blockchain.
- When the blockchain node receives the verification task, it checks the task tag in this task. If the task tag is valid, the verification request will be flooded to the verifier. Otherwise, the verification task will be canceled.

- If there exists the verifier that is interested in the verification task, it will send a response message, which includes the task tag and current time, to the validation requester before the deadline.
- When the validation requester receives the response message from the verifier, it will check this response message. If the task tag is legitimate, the reply time does not exceed the deadline, and there is no malicious behavior, the validation requester sends a confirmation message, which allows the verifier to implement the verification task, to the verifier. Otherwise, another verifier will be selected for verifying this proof.
- The selected verifier will implement the verification task by using the verification key. After the verification of this proof, an answer, which includes the verification result, task tag, current time, and confirmation message, is transmitted back to the validation requester before the deadline.
- When the validation requester receives the answer, it will check this answer. If the answer contains the confirmation message and the verification result is returned on time, the verification result can be accepted. Otherwise, the verification result will be discarded. Based on the accepted verification result, the validation requester can confirm that the prover owns enough transaction amount.

When the above authenticity verification is completed, the incentive mechanism calculates the verification fee for the verifier. Then, the verifier can obtain their verification fee from the validation requester's total amount of reward. In addition, blockchain generates one or several new blocks, which can be used to record the process of authenticity verification without any falsification. Hence, the third party can check authenticity verification from these blocks. Furthermore, authenticity verification can be tracked by using the Merkel root in the block.

Current State of Zero-Knowledge Proof in Blockchain

ZKP is well studied in the environment of blockchain. However, traditional ZKP schemes are usually not efficient. They may not be suitable for blockchain. Hence, there are schemes as described below that have been explored for the efficient and secure implementation of ZKP in the environment of blockchain.

Zerocoin: In order to realize completely anonymous currency transactions in Bitcoin, Miers *et al.* [11] proposed an extended cryptocurrency called Zerocoin. In Zerocoin, the double spending problem is solved by ZKP. Specifically, the behavior of double spending is prevented by a serial number, which corresponds to funding in the commitment. In this process, transaction privacy, which includes linkability and the source of the funding, will not be leaked. In addition, each participator tracks spent transactions according to displayed serial numbers. However, it still leaks the destinations and amounts of transactions.

Zerocash: Based on the improvement in zkSNARK, Sasson *et al.* [5] proposed a full-fledged ledger-based cryptocurrency called Zerocash. In Zerocash, based on the fixed address of the user, it can be paid directly and privately without interaction. Specifically, it can protect the source, destination and amount of the payment. Furthermore, Zerocash supports anonymous transactions with a variable amount. In terms of efficiency, the size of the transaction is at most one kilobyte. In addition, the verification time is less than 6 milliseconds. Compared with Zerocoin, the size of transactions, which are used for spending a coin, is reduced by 97.7 percent. And the verification time is reduced by 98.6 percent.

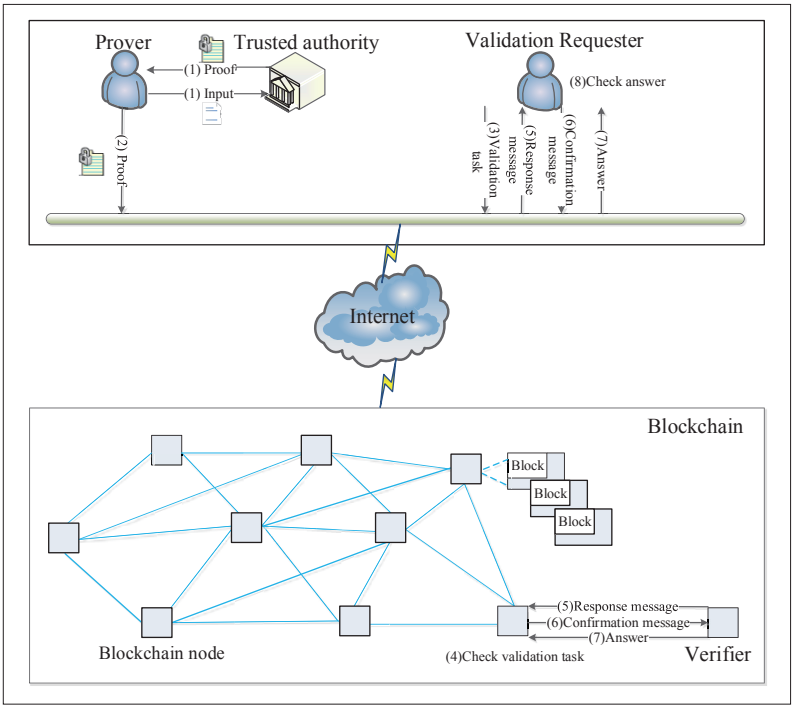

FIGURE 5. The architecture of zero-knowledge proof in blockchain.

Hawk: Based on Zerocash, Kosba *et al.* [12] designed a novel decentralized smart contract mechanism called Hawk. Hawk can offer both secure transactions and programmability in the environment of blockchain first. In this mechanism, non-interactive ZKP is used to guarantee the validity of contractual execution and funding transfer. Although the final outcome of smart contract can be verified securely, the entire sequence of adopted transaction actions is secret.

Bolt: Green *et al.* [13] proposed the concept of blind off-chain lightweight transactions (Bolt), which includes the construction of three secure payment channels in the decentralized cryptocurrency. Bidirectional channel is one of the secure payment channels. In this channel, blind signature and ZKP are combined to allow two users to implement the secure exchange of arbitrary valuable payments in either direction without leaking the linkage of their payments. In addition, a revocation token is used to prevent a dishonest user from preserving and using earlier versions of their refund token in the period of channel closure.

Baza's Mechanism: Based on the technique of smart contract, Baza *et al.* [14] designed a novel distributed firmware update mechanism, which can be used for the subsystems of autonomous vehicles. In addition, a consortium blockchain,

In the environment of blockchain, recorded and stored transaction data can be guaranteed that they are not to be tampered, forged or deleted by malicious users. However, because blockchain is open and transparent, there still exist various privacy issues in the implementation of the transaction.

which consists of different autonomous vehicle manufacturers, is utilized to guarantee authenticity and integrity for updating firmware. In this mechanism, distributors are able to join the distribution process, and the feature of mobility can ensure the availability and rapid delivery of the update. In order to get the proof of distribution, a ZKP scheme is utilized to exchange the update.

Li's Scheme: Based on the techniques of ring ZKP and blockchain, Li *et al.* [15] proposed an efficient and secure mechanism for impartial transactions in sharing economies. In this mechanism, ring ZKP is used to hide trade contents and transaction relationships without breaking verification and adding a new trusted participator. In order to ensure fairness, the verifier first confirms that the product from the provider is identical with the correct user's needed goods. Second, any participator cannot deceive the verifier in this distributed situation. Last but not the least, because all the transactions are recorded in blockchain, it can solve the problem of off-blockchain disputes effectively. Experiment results show that the proposed mechanism is more efficient than existing privacy-preserving blockchain-based mechanisms.

Challenges and Future Research Directions

In this section, we discuss some important and challenging problems. In addition, we outline several possible research directions.

Weaker Assumptions: One challenge of ZKP is whether it can be implemented efficiently by using some weaker assumptions. For example, zkSNARK is used in Zerocash. However, it needs a trusted third party, which is used for setup and system initialization. ZKP can be implemented without the trusted third party. However, it will affect the efficiency of ZKP. Hence, it is worth studying the efficient implementation of ZKP without the trusted third party.

Incorporation of Different Mechanisms: Different ZKP models have their own advantages. For instance, in Libra, the running time of the prover is linear. In addition, verification time and proof size are succinct in the uniform circuits, which space is logarithmic. Hyrax does not require any trusted setup. In order to make better use of these advantages, it is interesting to study whether these different mechanisms can be incorporated into a unified model.

Efficiency Optimization: In existing ZKP models, the efficiency optimization methods are usually suitable for the arithmetic circuits, which are over sufficient large fields. It is worth studying whether there exists a novel efficiency optimization method that can be used for arithmetic circuits over some small fields or the Boolean circuit. In addition, this possible method should not require any additional computation overhead. Furthermore, this method is associated with the decrease of the field size, and it will not affect the soundness of the proof.

Strongly Linear Version of Proof: It is interesting to research a new type of ZKP called strongly linear version of proof. It allows the verifier to implement the linear query to the input. In addition, the query is pointed to the proof, the verifier owns limited access to the input, and it realizes the standard concept of soundness. Although it is possible to design the strongly linear version of interactive oracle proof and probabilistically checkable proof, the best realizable parameters may not exist.

Other Mathematical Problems: Presently, in order to improve the efficiency of ZKP, most optimization approaches focus on researching the calculation of the bilinear group. Hence, it is worth researching the possibility of constructing highly efficient non-interactive ZKP models, which rely on other mathematical problems.

Cryptographic Tools: There are some cryptographic tools that can be incorporated with some existing non-interactive ZKP models. For example, it has been proven that the signature and commitment, which have the structure-preserving property, can be applied to non-interactive ZKP, then the mechanisms can be realized modularly and the efficiency of models can be ensured. However, in this research direction, there still exist several problems that are related to the efficiency and application of the model.

Lattice-Based Cryptography: The public key cryptographic algorithm is the key factor for the construction of ZKP models in the environment of blockchain. Unfortunately, common algorithms cannot resist quantum computing attacks. For example, the RSA algorithm can be solved by the Shor algorithm in a polynomial time. Because there is no effective quantum algorithm for solving lattice-based cryptography, it is worth researching efficient and secure ZKP models by using lattice-based cryptography.

Conclusion

In the environment of blockchain, recorded and stored transaction data can be guaranteed that they are not to be tampered, forged or deleted by malicious users. However, because blockchain is open and transparent, there still exist various privacy issues in the implementation of the transaction. Hence, we study the usage of ZKP in the environment of blockchain. In this article, we have carried out a survey of ZKP in the environment of blockchain. First, we introduced the framework, models and applications of ZKP. Next, we provided an introduction of blockchain, and proposed a framework of ZKP in the environment of blockchain. Then, we showed the current state of ZKP in the environment of blockchain. Lastly, we explored potential research problems.

Acknowledgment

This work is supported through the National Natural Science Foundation of China under Grant (61802118); the Science and Technology Innovation Projects of Shenzhen (JCYJ20190809152003992); and the China Postdoctoral Science Foundation (2019M653042).

References

[1] R. L. Rivest, L. Adleman, and M. L. Dertouzos, "On Data Banks and Privacy Homomorphisms," *Foundations of Secure Computation*, vol. 4, no. 11, 1978, pp. 169–80.

[2] R. L. Rivest, A. Shamir, and Y. Tauman, "How to Leak a Secret," *Proc. Advances in Cryptology — ASIACRYPT 2001*, 2001, pp. 552–65.
[3] A. C. Yao, "Protocols for Secure Computation," *Proc. 23rd Annual Symposium on Foundations of Computer Science*, 1982.
[4] E. Ben-Sasson *et al.*, "Succinct Noninteractive Zero Knowledge for a Von Neumann Architecture," *Proc. 23rd USENIX Security Symposium*, 2014, pp. 781–96.
[5] E. B. Sasson *et al.*, "Zerocash: Decentralized Anonymous Payments from Bitcoin," *Proc. 2014 IEEE Symposium on Security and Privacy*, 2014, pp. 459–74.
[6] S. Ames *et al.*, "Ligero: Lightweight Sublinear Arguments without a Trusted Setup," *Proc. 2017 ACM SIGSAC Conf. Computer and Commun. Security*, 2017, pp. 2087–2104.
[7] B. Bünz *et al.*, "Bulletproofs: Short Proofs for Confidential Transactions and More," *Proc. 2018 IEEE Symposium on Security and Privacy*, 2018, pp. 315–34.
[8] R. S. Wahby *et al.*, "Doubly-Efficient zkSNARKs without Trusted Setup," *Proc. 2018 IEEE Symposium on Security and Privacy*, 2018, pp. 926–43.
[9] E. Ben-Sasson *et al.*, "Aurora: Transparent Succinct Arguments for R1CS," *Advances in Cryptology — EUROCRYPT 2019*, 2019, pp. 103–28.
[10] T. Xie *et al.*, "Libra: Succinct Zero-Knowledge Proofs with Optimal Prover Computation," *Proc. Advances in Cryptology — CRYPTO 2019*, 2019, pp. 733–64.
[11] I. Miers *et al.*, "Zerocoin: Anonymous Distributed E-Cash from Bitcoin," *Proc. 2013 IEEE Symposium on Security and Privacy*, 2013, pp. 397–411.
[12] A. Kosba *et al.*, "Hawk: the Blockchain Model of Cryptography and Privacy-Preserving Smart Contracts," *Proc. 2016 IEEE Symposium on Security and Privacy*, 2016, pp. 839–58.
[13] M. Green and I. Miers, "Bolt: Anonymous Payment Channels for Decentralized Currencies," *Proc. 2017 ACM SIGSAC Conf. Computer and Commun. Security*, 2017, pp. 473–89.
[14] M. Baza *et al.*, "Blockchain-Based Firmware Update Scheme Tailored for Autonomous Vehicles," *Proc. 2019 IEEE Wireless Commun. and Networking Conf.*, 2019.
[15] B. Li and Y. Wang, "RZKPB: A Privacy-Preserving Blockchain-Based Fair Transaction Method for Sharing Economy," *Proc. 2018 17th IEEE Int'l. Conf. Trust, Security and Privacy in Computing and Commun./12th IEEE Int'l. Conf. Big Data Science And Engineering*, 2018, pp. 1164–69.

Biographies

Xiaoqiang Sun (xqsun@szu.edu.cn) received his Ph.D. degree from Shenzhen University in 2018. He was a post doctoral fellow with F. Richard Yu's group, Carleton University, Ottawa, from 2019 to 2020. He is now a post doctoral fellow at the Guangdong Key Laboratory of Intelligent Information Processing, College of Electronics and Information Engineering, Shenzhen University. His research interests include cryptography, information security, and fully homomorphic encryption. He has published more than 10 academic journal and conference papers.

F. Richard Yu (richard.yu@carleton.ca) is a professor at Carleton University, Canada. His research interests include connected/autonomous vehicles, artificial intelligence, cybersecurity, and wireless systems. He has received several professional awards, including the Ontario Early Researcher Award, Carleton Research Achievement Awards, and several Best Paper Awards from first-tier conferences. He is a Fellow of the IEEE, Canadian Academy of Engineering (CAE), Engineering Institute of Canada (EIC), and IET. He is a Distinguished Lecturer of IEEE in both VTS and ComSoc.

Peng Zhang (zhangp@szu.edu.cn) received her Ph.D. degree from Shenzhen University in 2011. She is now an associate professor at the College of Electronics and Information Engineering, Shenzhen University. Her current research interests include cryptography technology and security in Blockchain, cloud computing, and IoT. She has published more than 30 academic journal and conference papers.

Zhiwei Sun (smeker@szpt.edu.cn) received his Ph.D. degree from Sun Yat-sen University in 2014. He is now an associate professor at Shenzhen Polytechnic. His current research interests include communication security and quantum cryptography, among others.

Weixin Xie (wxxie@szu.edu.cn) is currently a professor at Shenzhen University. His research interests include signal processing and intelligent information processing, among others.

Xiang Peng (xpeng@szu.edu.cn) is currently a professor at Shenzhen University. His research interests include 3D imaging and modeling and optical imaging, among others.