ISBN 978-981-14-1684-2
Proceedings of 2019 the 9th International Workshop on Computer Science and Engineering
(WCSE 2019 SUMMER)
Hong Kong, 15-17 June, 2019, pp. 37-46
doi: 10.18178/wcse.2019.06.007

Research on the Privacy Security Puzzle Scheme of Blockchain

Tao Feng[+], Wentao Hao, Jinze Du

School of Computer and Communication, Lanzhou University of Technology,
Lanzhou 730050, China

Abstract. The blockchain represented by Bitcoin, the distributed ledger system can prevent double spending effectively and other issues. But it has caused the concentration of bitcoin mining power, which completely contradicts the essential feature of the complete decentralization of the blockchain. In the blockchain system, transaction data information among users is transparently stored on the chain, and each participant can obtain a complete data backup. These transaction data are stored in an open and transparent manner on each node of the entire network, although it can prevent data forgery and tampering, it brings data privacy problems. In this paper, we propose a multi-level reward structure for the problem of centralized power calculation in the mining pool. This structure can make the miners work alone to prove the workload and obtain the reward through their own calculation power, in order to weaken the role of the mining pool. At the same time, for the data privacy security problem in the bitcoin trading system, we have improved the bitcoin puzzle, and proposed a puzzle scheme based on non-interactive zero-knowledge proof to ensure data transaction security.

Keywords: Blockchain, decentralization, reward structure, transaction privacy, non-interactive zero-knowledge proof

1. Introduction

1.1. Research background

Blockchain [1] technology is a decentralized infrastructure that is widely used in emerging digital cryptocurrencies and has received attention and research as bitcoin is gradually accepted. Blockchain is highly sought after by financial institutions due to its features of decentralization, non-tampering of block data and de-trust [2]. Bitcoin system has many advantages in addition to its decentralization, open source, low transaction cost, fixed circulation, no external control, strong security, and prevention of double spending [3]. Also, there are many disadvantages. Firstly, due to the distributed nature of the system, the transaction is verified by all nodes, and it takes about one hour to confirm the success of the payment, and real-time transactions cannot be realized [4]. Secondly, the lack of scalability of the Bitcoin system [5], limited number of mining nodes, more and more transactions and a large amount of verification data will be detrimental to the normal operation of the P2P network, resulting in network congestion. Thirdly, the transparency of Bitcoin transaction block information has caused some user-related data information to be leaked [6-8], although the attacker only knows the user's data but does not know which user is specific, but with big data. With the development of technology, attackers using data analysis, data extraction and other technologies may crack user information soon, which poses a potential threat to Bitcoin users [9-10].

In the blockchain system, transactions between users are recorded on the block, and the blocks containing the transaction information are linked from the back to the front, all transaction data, transaction amount, capital flow, and payment account. Privacy such as clearly associated features are public, and each participant can get a complete data backup. With the development of various anti-anonymity identity screening technologies, these exposed contents allow attackers to track the entire historical transaction path and extract a large amount of information about the user's identity. Through the analysis of the above figure,

[+] Corresponding author. Tel.: 0931-2976016;
E-mail address: fengt@lut.cn.

we can find that the bitcoin system and Ethereum, all have pseudo anonymity [11], mainly using the HASH-256 encryption algorithm. All the transaction data of these currencies are publicly and transparently stored in each node of the whole network during the transaction process. Although it can prevent data forgery and tampering, it will bring data privacy problems.

1.2. Related work

Both the Bitcoin address and the Ethereum account address are hash values of the public key. There is no direct correspondence between the bitcoin address and the user's real identity information. A certain degree of anonymity is achieved by relying on the association between the block address and the user's identity. In addition, each transaction uses a brand-new address, and there is no association between the different addresses, which enables unrelatedness between multiple transactions. However, these solutions cannot guarantee absolute privacy, for example, real-name authentication of e-wallet or bitcoin exchange, association between multiple inputs in the same transaction, correspondence between IP addresses and bitcoin addresses in network messages, etc. Both of them may reveal the identity of the user.

Blockchain privacy protection needs to cover up the details of the transaction, but also the verification and the correctness of the transaction. The current privacy protection schemes for blockchains include CoinJoin, ring signature, Zero-Knowledge Proof, homomorphic encryption [12], and so on.

The Dash [13] proposed in the article applies the principle of CoinJoin, which combines multiple transactions into a single transaction to interrupt the connection between the sending address and the receiving address, so as to guarantee the privacy of block transaction data effectively.

The new currency monero [14] mentioned applies the ring signature algorithm, encrypts the transaction data with the sender's private key and the public key of the unrelated person, and then uses the public key of the owner to decrypt the data, thereby hiding the sender of the transaction, and thus ensuring the privacy of the transaction user identity.

In the literature [15], MIERS et al. proposed a Zerocoin scheme for extending the bitcoin scheme, by using zero-knowledge proof [16] to prevent transaction graph analysis. A global currency pool is implemented using a zero-knowledge-based encryption accumulator that provides built-in unlinkability and a distributed "cleaning" mechanism that is less efficient. The shortcomings of the scheme are: First, the amount of Zerocoin's currency is fixed, and the amount of the amount cannot be arbitrarily divided. Second, anonymity is provided by disconnecting the payment transaction from the source address, but it does not hide the amount or other metadata about the transaction that occurred on the network. Third, major changes to the blockchain system are required. Fundamentally speaking, it is a new currency. It does not support non-interactive transactions and the zero-knowledge proof process is too long. It seems unlikely because of computational overhead. The blockchain is backward compatible.

In the literature [17], the proposed Hawk model implements a smart contract with privacy protection to support the privacy protection of any transaction. The homomorphic encryption based on the homomorphic mapping ensures that the first operation is encrypted and the first encryption is the same, so that transaction verification can be implemented based on the encrypted data.

1.3. Contributions of this article

In order to improve the security and practicability of existing bitcoin puzzle solutions, we have formalized a bitcoin puzzle based on non-interactive zero-knowledge proof structure, which not only expands the function of bitcoin puzzles. Moreover, the privacy of the current blockchain system is solved by optimizing the improved algorithm structure.

By analysing the bitcoin puzzle scheme, this paper proposes a weak non-outsourced puzzle structure, introduces a multi-level reward structure, and encourages the nodes to separately perform the proof of the workload, in order to obtain the rewards, they deserve. Thereby weakening the role of the mining pool, and providing an idea for the complete decentralization of the blockchain technology.

The above-mentioned bitcoin puzzle scheme is improved to solve the privacy leakage problem of the open and transparent transaction information of the bitcoin system in the transaction chain. By introducing a

non-interactive zero-knowledge proof structure, the user transaction information privacy protection mechanism is improved, and the privacy leakage of transaction information in the blockchain is avoided.

2. A Scheme for Multi-Level Reward Structure Based on Bitcoin Scratching Puzzles

2.1. Formal definition of scraping puzzles

2.1.1. Formal definition of scraping puzzles

Assume that λ is a security parameter, and a scraping puzzle is instantiated by a set of parameter sets $M=(t_0,t_1,u,d)$. In general, t_0 represents the initialization overhead of the algorithm, and t_1 represents the attempt to solve a single scraping puzzle solution. Calculate the workload, where u refers to the maximum number of other miners' opponents looking for a solution, and d refers to the average number of times an adversary would affect an attempt to find a solution. We assume that $t_0 << 2^d t_1$, $2^d t_1$ is the time required to solve a scratching puzzle expectation.

2.1.2. Algorithm structure of the puzzle

A basic scraping puzzle consists of a set of algorithms $(\mathrm{G}(1^\lambda), \mathrm{Work}, \mathrm{Verify})$

(1) (G): $\mathrm{G}(1^\lambda)\rightarrow puz$, Generate an example of a scraping puzzle.

(2) (Work): $\mathrm{Work}(puz,m,t)\rightarrow ticket$,The workload calculation algorithm requires a puzzle instance puz, some transaction load m, and a time parameter t, t for the unit scratch attempt, where $t*t_1+t_0$ is used as a time step, where $t_1=poly(\lambda)$ is the unit scratch time, t_0 can be It is considered to be the workload cost of initialization and finalization.

(3) (Verify): $\mathrm{Verify}(puz,m,ticket)\rightarrow\{0,1\}$, Check if the ticket is valid for a puzzle instance a and transaction load. If we pass the inspection, we call (puz,m) the winning ticket.

In general, an honest miner performs t time scratch attempt, and the probability of each attempt to find a winning ticket is 2^{-d}, where d is called the difficulty parameter of the puzzle. We stipulate that $\zeta(t,d):=1-(1-2^{-d})^t$: is the probability of using the t scratch method to find the winning ticket, and we assume that the difficulty parameter d of the puzzle is fixed.

2.2. Security analysis of puzzles

(1) Correctness

For any puzzle (puz,m,t) to be solved, The winning ticket is found by the $\mathrm{Work}(puz,m,t)$ calculation, and if the winning ticket is not null, the verification $(puz,m,ticket)=1$ is performed.

(2) Feasibility

Solving a scratching puzzle is feasible. For any $\eta=poly(\lambda)$, for any t_1,t_2, ... $t_\eta=poly(\lambda)$, let:

$$\Pr\left[\begin{array}{l} puz\leftarrow G(1^\lambda),\\ m\leftarrow\{0,1\}^\lambda,\\ \forall i\in[\eta], ticket_i\leftarrow \mathrm{Work}(puz,m,t_i):\\ \exists i\in[\eta], \mathrm{Verify}(puz,m,ticket_i) \end{array}\right]\geq \zeta(t)-negl(\lambda) \quad (1)$$

Intuitively, the time taken for each scratching unit is t, and the probability of finding a winning ticket is 2^{-d}, So if potentially, go to parallel processing each time the scratch time t_1,t_2, $...t_\eta=poly(\lambda)$, The probability of finding a winning lottery ticket is $\zeta(t)\pm negl(\lambda)$, here $t:=\sum_{i\in[\eta]}t_i$.

(3) incompressibility

Simply put, the workload of solving a puzzle must be incompressible. Even the best adversary A can speed up the search for a puzzle solution, mainly to find an incompressible factor u. More specifically, if at most step t^*t_1 is required for any adversary A who spends a probabilistic polynomial time, then it can be stated that the scraping puzzle has u incompressible (where $u\geq 1$).

$$\Pr\left[\begin{array}{l} puz \leftarrow G(1^{\lambda}), \\ (m, ticket) \leftarrow \mathrm{A}(puz): \\ \mathrm{Verify}(puz, m, ticket) = 1 \end{array}\right] \leq \zeta^{+}(ut) \pm negl(\lambda) \tag{2}$$

note: $\zeta^{+}(t) = 1 - \left(1 - 2^{-d}\right)^{t+1}$ Probably the probability of outputting a winning ticket after scratching in t units, although we additionally allow the adversary to make the final guess at the end, the index is $t+1$ instead of t. Ideally, we want the compression factor u to be as close to 1 as possible.

When $u = 1$, honest work algorithms are the best way to solve a problem. This definition specifically states that previous puzzle solutions did not help solve newly generated puzzles.

2.3. The presentation of a multi-level reward mechanism

We want to design a new proof of work package that discourages centralized mining pools and managed mining services, but encourages individual miners to participate and provides overall functionality and security like today's Bitcoin.

In the Bitcoin system, 6 blocks are required for each transaction, and a 1M block is generated in 10 minutes. It takes 1 hour for the entire network to confirm a transaction. But every time the reward is obtained, it is always the miner node of the first verification block, while the other miner nodes of the subsequent calculated random number are not rewarded by the system. They waste time and waste the computing power, but ultimately, they do not get the reward. reward. This prompted them not to calculate the problem alone, but to join the mining pool for centralized calculation. The reason why the mining pool has such a great attraction lies in the reward mechanism of the mining pool. When the system issues a problem, the miners' nodes and the mining pools that work alone will compete to solve this problem. In terms of computing power, the miners' nodes are far less than the calculation power of the mining pool, and thus it is also a waste of computing power. The problem is solved by the mining pool, and the mining pool receives certain rewards. Then the mining pool will assign a certain reward to each node by analyzing the power of the miners in the pool to solve the problem. Under the economic interests, this allows more miners to join the pool without going to work alone.

Our multi-level design is inspired by the payment structure of the lottery game, and a lottery is usually set up in the lottery payment structure, and several other consolation prizes. The effectiveness of such lottery tickets to encourage broad participation has been confirmed in practice. Our implicit assumption is that if the miners often receive some rewards, they will not join the pool, and the problem of centralized concentration will be effectively solved. We offer such a multi-level rewards program, which we will discuss below. Every attempt to solve a puzzle is likely to result in three possible rewards, unlike today's bitcoin, where each bit gets the same reward.

2.4. Implement multi-level rewards (introduction of thresholds)

We provide such a multi-level reward program to adjust parameters based on different scenarios and discuss their basic principles. Every attempt to crack a puzzle has a certain probability of winning three possible prizes. In contrast, today's Bitcoin, each block can receive the same reward. Rewards are not only related to different reward values, but also to the weight of the blockchain “difficulty” scoring function.

Low Value Consolation Prize: Provides low variance incentives to independent miners. The consolation prize is awarded most often, with a small bonus, less than one-tenth of Bitcoin, but the total amount of the bonus is a large part of the expected value. The main role is to motivate the miners to work alone to prove the workload.

Median Main Award: Make sure the block confirmation arrives on a regular basis. A medium-value bonus is necessary to ensure that the speed and security of the transaction record is like Bitcoin. They are distributed at the same rate as regular Bitcoin blocks; they receive more rewards than low-value bonuses, but contribute much less to the overall expected return.

High-value bonuses: Prevent mining inspections from cheating hosted mining services. The jackpot is very rare, only a small part of the total expected value, but it disproportionately accounts for a large part. The role of this reward is to cause distrust of the managed mining provider.

On top of any known puzzle structure, for example, bitcoin puzzles, in these structures, the key step in the mining attempt is to compare the hash value to the threshold $\mathrm{H}(puz||ticket)<T$, where the threshold $T=2^{\lambda-d}$ is represented by the difficulty value d Chemical.

In order to achieve a three-tier reward, we introduce two thresholds, namely T_{medium} and T_{high}, and require $T_{\text{medium}}<T_{high}<T$, If they obtained hash value is between T_{high} and T, then this attempt to obtain a high value reward; If between T_{medium} and T_{high}, get a medium value reward; In addition, it can also get low value returns. These thresholds must be set according to the desired frequency of each reward type.

3. A puzzle structure scheme based on non-interactive zero-knowledge proof

3.1. Bitcoin puzzle structure security definition

3.1.1. Formal definition of bitcoin puzzles

We assume that each random oracle call requires time t_{RO}, and that all other work in each work iteration takes time t_{other}.

An abstract bitcoin puzzle structure is shown below:

Assume that: $\mathrm{H}:\{0,1\}^{*}\to\{0,1\}^{\lambda}$ is a model that models the hash function as a random predictive machine.

(1) $\mathrm{G}(1^{\lambda})$, Initialize the model and randomly build a puzzle model $puz\xleftarrow{\$}\{0,1\}^{\lambda}$

(2) $\mathrm{Work}(puz,m,t)$, Computational puzzles:

a. For $i\in[t]$, repeat the calculation of the scraping puzzle;

b. $r\xleftarrow{\$}\{0,1\}^{\lambda}$, Calculating a random value r;

c. If you calculate $\mathrm{H}(puz,||m,||r)<2^{\lambda-d}$, the equation holds, then the computer has won the ticket value, then return $ticket:=r$. Otherwise return $\perp$.

(3) $\mathrm{Verify}(puz,m,ticket)$, Verify the winning notes. Check if the winning ticket meets the inequality $\mathrm{H}(puz||m||ticket)<2^{\lambda-d}$.

Theorem 1: Incompressibility

The bitcoin puzzle structure shown above is a scraping puzzle, if and only if $t_1=\mathrm{O}(\lambda)$, $t_0=0$, while the factor $u=t_{RO}/(t_{RO}+t_{other})$ is not compressible.

Proof: Proof of correctness is easy, as is proof of feasibility and parallelism. For u incompressibility, we observe that for any adversary that calls a random oracle, its probability of successfully finding a winning ticket is at most $\zeta^{+}(t)$. Since the honest miner node passes the time $(t_{RO}+t_{other})*t$ required by the algorithm, the incompressible time of this scraping puzzle is $t_{RO}/(t_{RO}+t_{other})$, so the scraping puzzle is incompressible and the incompressible factor is $u=t_{RO}/(t_{RO}+t_{other})$.

3.1.2. Weak non-outsourcing of bitcoin puzzles

Bitcoin puzzles are easy to outsource relative to other jobs. We now describe an outsourcing protocol for bitcoin puzzles that is like the one currently used by Bitcoin mining pools to coordinate untrusted pool members. Let $d'<d$ be the parameter and d' be the sharing difficulty value.

Specifically, the pool operator selects a payload m (actually, it includes a reward payment for the pool operator's public key, and other new bitcoin transactions that the pool operator decides on its own), and then the staff performs a $2^{d'}$ scrap try.

On average, the miner node finds at least one value r called the “shared value”, so that $\mathrm{H}(puz||m||r)<2^{\lambda-d'}$ is established and the node sends this value to the pool operator, which the pool operator uses to distinguish between honest or dishonest workers.

Order: $u=t_{RO}/(t_{RO}+t_{other})$ is the maximum potential acceleration that an adversary can obtain through work. Let $t_g=2^{d'}-O(1)$be the lower bound of the number of random oracles that the adversary must take in order to produce a share with a statistically similar probability to the honest worker.

Theorem 2: weak non-outsourcing

The bitcoin puzzle structure we described above is not a weak non-outsourcing puzzle based on the outsourcing protocol (t_S, t_C, t_e, a, p_s) . When $t_C = O\{\lambda\}$, at the same time $t_S \le 2^{d'}$, $p_s > \zeta\left(u\left(t_e + a/t_1\right) - t_g\right)/\zeta\left(t_e\right)$.

Proof: Consider the (t_S, t_C, t_e) outsourcing protocol (S, C) as described above, and assume that $m \xleftarrow{\$} \{0,1\}^{\lambda}$ is randomly chosen and $m \neq m^*$. Assuming there is an adversary with a run time of $t_S{}^* t_1 + a$, the pool operator's view is indistinguishable between the execution protocols (A, C) and (S, C), and the miner node outputs a valid ticket m^* with probability $p_s \zeta\left(t_e\right)$.

3.2. Construction of weak non-outsourcing puzzles scheme

3.2.1. Algorithm structure of the puzzle

A bitcoin puzzle structure with a weak non-outsourced nature is as follows:

Assumption: $\mathrm{H}_1\mathrm{H}_2: \{0,1\}^* \to \{0,1\}^{\lambda}$ is a random prediction model.

1. $\mathrm{G}\left(1^{\lambda}\right)$: Initialize the model and randomly build a puzzle model, $puz \xleftarrow{\$} \{0,1\}^{\lambda}$
2. $\mathrm{Work}(puz, m, t)$ Computational puzzles:
   (1) Constructing a Merkle tree, Sample L , random string $leaf_1, \ldots\ldots leaf_L \xleftarrow{\$} \{0,1\}^{\lambda}$, Then construct a Merkle tree on the leaf, H_2 as a hash function. Let $digest$ represent the root summary of this tree.
   (2) Scratch attempt:
      a. $r \xleftarrow{\$} \{0,1\}^{\lambda}$, Calculate a random value r ;
      b. $h = \mathrm{H}(puz \| r \| digest)$ And use the value h to select q different leaf nodes from the tree.
      c. Let $B_1, B_2, \ldots\ldots B_{\mathrm{q}}$ denote the branch corresponding to the selected leaf. For a given leaf node, its branch is defined as the Merkle proof of the leaf node, including the leaf node itself, and the sibling of each node on the path from leaf to root.
      d. Sort by $\sigma_h := \{B_{\mathrm{i}}\}_{i \in [\mathrm{q}]}$
      e. If $\mathrm{H}(puz \| r \| \sigma_h) < 2^{\lambda - d}$, Record the effective resolution value (r, σ_h) and go to the valid signature payload.
   (3) Signature payload: If a resolution value is not found, $\perp$ is output, otherwise the following operations will continue:
      a. Calculate $h' := \mathrm{H}(puz \| m \| digest)$, use the value h' to select a set of $4q'$ leaves that are not the same from the tree, so that these leaves are not included in σ_h. From these, q' selects any subset of different leaves. Collect the corresponding branches of these q' leaves, denoted as $B_1, B_2, \ldots\ldots B_{\mathrm{q}}$
      b. Sort by $\sigma'_{\mathrm{h}} := \{B_{\mathrm{i}}\}_{i \in [\mathrm{q}']}$
      c. Return ticket value $ticket := (digest, r, \sigma_h, \sigma'_h)$
3. Verify the value of the ticket $\mathrm{Verify}(puz, m, ticket)$:
   (1) Parsing the value of the ticket $ticket := (digest, r, \sigma_h, \sigma'_h)$
   (2) Calculation $h := \mathrm{H}(puz \| r \| digest), h' := \mathrm{H}(puz \| m \| digest)$
   (3) Verify that σ_h and σ'_h contain leaf nodes selected by h and h', respectively.
   (4) Verify that σ_h and σ'_h contain valid Merkle paths for the $digest$
   (5) Verification inequality $\mathrm{H}(puz \| r \| \sigma_h) < 2^{\lambda - d}$

As described in the above structure, we construct a bitcoin puzzle scheme with weak non-outsourcing nature. The scheme is mainly composed of three algorithm sets. Next, we prove the security of the puzzle in the scheme.

3.2.2. Proof of security of the puzzle

(1) Correctness analysis

For any puzzle (puz, m, t) to be solved, the winning ticket is found by $\mathrm{Work}(puz, m, t)$ calculation, and if the winning ticket is not defective, then $(puz, m, ticket) = 1$ is verified.

(2) Feasibility analysis

Solving a scratching puzzle is feasible and can be parallelized. For any $\eta = poly(\lambda)$, for any $t_1, t_2, ..., t_\eta = poly(\lambda)$ let $t := \sum_{i\in[\eta]} t_i$:

$$\Pr\left[\begin{array}{l} puz \leftarrow \mathrm{G}(1^\lambda), \\ m \leftarrow \{0,1\}^\lambda, \\ \forall i \in [\eta]: ticket_i \leftarrow \mathrm{Work}(puz, m, t_i): \\ \exists i \in [\eta]: \mathrm{Verify}(puz, m, ticket_i) \end{array}\right] \geq \zeta(t)\text{-}negl(\lambda) \tag{3}$$

Intuitively, the time taken for each scratching unit is t, and the probability 2^{-d} of winning the ticket is found, so if, potentially, the parallel processing is performed in parallel for each scratching time $t_1, t_2, ..., t_\eta = poly(\lambda)$ The probability of finding a winning lottery is $\zeta(t) \pm negl(\lambda)$, here $t := \sum_{i\in[\eta]} t_i$.

3.3. Security scheme based on non-interactive zero-knowledge proof puzzle structure

3.3.1. Non-Interactive Zero Knowledge Proof Structure (NIZK)

A non-interactive zero-knowledge proof system [18] consists of three algorithm sets: $\mathrm{NIZK} = (\mathrm{Setup}, \mathrm{Prove}, \mathrm{Verify})$

(1) $crs \leftarrow \mathrm{Setup}(1^\lambda)$ Enter a security parameter λ to generate a public reference string crs .

(2) $\pi \leftarrow \mathrm{Prove}(crs, x, w)$ Enter strings crs , statements x , and evidence w , if $(x,w) \in L$,Then the proof π is output.

(3) $b \leftarrow \mathrm{Verify}(crs, stmt, \pi)$, Enter strings crs , and statements x , and a proof process π , Output 0 or 1, indicating rejection and receipt of statements and evidence.

3.3.2. Security Analysis of NIZK

(1) Integrity analysis

If a demonstrator with valid evidence can always convince an honest verifier, then the non-interactive zero-knowledge proof system NIZK is complete. Formally speaking, for any $(x,w) \in R$, we have the following properties:

$$\Pr\left[\begin{array}{l} crs \leftarrow \mathrm{Setup}(1^\lambda) \\ \pi \leftarrow \mathrm{Prove}(crs, x, w): \\ \mathrm{Verify}(crs, x, \pi) = 1 \end{array}\right] = 1 - negl(\lambda) \tag{4}$$

(2) Robustness analysis

The non-interactive zero-knowledge proof system NIZK system is considered reasonable and stable if it is not feasible to prove a false statement for any polynomial time opponent A . Then there are the following properties:

$$\Pr\left[\begin{array}{l} crs \leftarrow \mathrm{Setup}(1^\lambda) \\ (x,\pi) \leftarrow A(crs): \\ (x \notin L) \cap (\mathrm{Verify}(crs, x, \pi) = 1) \end{array}\right] = negl(\lambda) \tag{5}$$

(3) Zero knowledge analysis

If a non-interactive zero-knowledge proof system does not disclose any information about the evidence to any polynomial time adversary A during the proof process, the *NIZK* system is computationally zero-knowledge. Specifically, if simulator $\mathrm{S} = (\mathrm{SimSetup}, \mathrm{SimProve})$ is present, the NIZK system is considered to be computationally zero knowledge, for all polynomial time opponents A , for any statement x , the evidence w makes $(x,w) \in R$, we all consider the following properties:

$$\Pr\left[\begin{array}{l} crs \leftarrow \mathrm{Setup}(1^\lambda), \\ \pi \leftarrow \mathrm{Prove}(crs, x, w): \\ A(crs, x, \pi) = 1 \end{array}\right] - \Pr\left[\begin{array}{l} (crs', \tau) \leftarrow \mathrm{SimSetup}(1^\lambda, x), \\ \pi \leftarrow \mathrm{SimProve}(crs, x, \tau): \\ A(crs', x, \pi') \end{array}\right] = negl(\lambda) \tag{6}$$

3.3.3. Puzzle structure based on non-interactive zero-knowledge proof

Make NIZK a non-interactive, zero-knowledge proof system. We assume that $\varepsilon = (\mathrm{Key}, \mathrm{Enc}, \mathrm{Dec})$ is a CPA secure public key encryption scheme.

It is also assumed that $(\mathrm{G}', \mathrm{Work}', \mathrm{Verify}')$ is a puzzle scheme based on non-interactive zero-knowledge proofs. The following are the construction steps of our scheme:

(1) $\mathrm{G}'(1^{\lambda})$ Initialize, run the underlying scheme to generate puzzles, $puz' \leftarrow \mathrm{G}'(1^{\lambda})$, $crs \leftarrow \mathrm{NIZK.Setup}(1^{\lambda})$, make $(sk_{\varepsilon}, pk_{\varepsilon}) \leftarrow \varepsilon.Key(1^{\lambda})$ established, The output is $puz \leftarrow (puz', crs, pk_{\varepsilon})$.

(2) $\mathrm{Work}'(puz, m, t)$ Puzzle calculation.

a. Initialization puzzle, $puz := (puz', crs, pk_{\varepsilon})$;

b. Calculated ticket value, $ticket' \leftarrow \mathrm{Work}'(puz', m, t)$;

c. Encrypted ticket, $c \leftarrow \mathrm{Enc}(pk_{\varepsilon}; ticket'; r)$;

d. Forming the certification process;

$\pi \leftarrow \mathrm{NIZK.Prove}(crs, (c, m, pk_{\varepsilon}, puz'), (ticket', r))$, If the following inequality is true for the NP statement,

$\mathrm{Verify}'(puz', m, ticket') \cap \varepsilon = \mathrm{Enc}(pk_{\varepsilon}; ticket'; r)$, Then output solution value $ticket := (c, \pi)$

(3) $\mathrm{Verify}'(puz, m, ticket)$ Verification solution, Analytic puzzle $puz := (puz', crs, pk_{\varepsilon})$, Parsing the ticket as (c, π), Check if $\mathrm{Verify}(crs, (c, m, pk_{\varepsilon}, puz'), \pi) = 1$ is true.

Next, we state a theorem that this generic transformation basically preserves the non-transferability of the underlying puzzle.

Theorem 3:

If the basic puzzle $(\mathrm{G}', \mathrm{Work}', \mathrm{Verify}')$ is σ' non-transferable, then the puzzles derived from the non-interactive zero-knowledge proof are also non-transferable:

$$u + \sigma' \le \frac{(u+\sigma)t}{t^* t_1 + (t_{enc} + t_{nizk})\eta} \tag{7}$$

Where t_{enc} and t_{nizk} are the time to perform each encryption and the non-interactive zero-knowledge proof NIZK time.

3.4. Security Analysis of Puzzle Structure Based on Non-Interactive Zero Knowledge Proof

3.4.1. Analysis of correctness

Correctness: If the honest demonstrator gives a true proof that the verifier can accept, then the puzzle scheme based on the non-interactive zero-knowledge proof satisfies the correctness.

Proof: If a non-interactive zero-knowledge system $(\mathrm{Setup}, \mathrm{Prove}, \mathrm{Verify})$ is complete for all languages L, then for all x and w, satisfying $M_L = (x, w)$ the following holds true:

$$\Pr\begin{bmatrix} params \leftarrow \mathrm{Setup}(1^{\lambda}); \\ \pi \leftarrow \mathrm{Prove}(params, x, w, M_L): \\ \mathrm{Verify}(params, x, w, M_L) = 1 \end{bmatrix} = 1 \tag{8}$$

3.4.2. Robustness Analysis

Robustness: Refers to any statement $x \in L$, polynomial time adversary A forged evidence is not accepted by honest verifier.

Proof: A non-interactive zero-knowledge system $(\mathrm{Setup}, \mathrm{Prove}, \mathrm{Verify})$ is complete for all languages L, then the following conditions must be met:

$$\Pr\begin{bmatrix} params \leftarrow \mathrm{Setup}(1^{\lambda}); \\ (\pi, \mathrm{x}, L) \leftarrow A(params, x): \\ x \notin L \cap \mathrm{Verify}(params, x, \pi, M_L) = 1 \end{bmatrix} \le v(\lambda) \tag{9}$$

Where A is a polynomial time adversary and M_L is a Turing algorithm, If and only if $x \in L$, the Turing algorithm M_L receives (x, w).

For a non-interactive zero-knowledge proof system $(Setup, Prove, Verify)$, the non-interactive proof system is perfectly robust, if $\mathrm{v}(\lambda) = 0$ is made for all polynomial time adversaries.

3.4.3. Zero knowledge analysis

Zero knowledge: The verifier does not know anything other than knowing that the statement is correct. If an honest witness uses the simulator to forge a proof in the absence of evidence w, in the form of $x \in L$, it is indistinguishable from the true proof process. The simulator must have the ability to allow the adversary to produce similar proofs. Two systems exist in the proof, one is zero-knowledge proof that one of the bulletin reference strings is used only in a separate proof, and the other is that the same bulletin reference string is used in multiple proofs.

Proof: If the system satisfies the following equation, then the non-interactive zero-knowledge proof system $(\text{Setup}, \text{Prove}, \text{Verify})$ is a single zero-knowledge proof system and is true for all languages L.

For all $x \in L$, there is evidence w has $M_L(x,w)=1$, and all adversaries A, the following two algorithms are indistinguishable:

$$\text{Real}(\lambda)=\left\{\begin{array}{l} params \leftarrow \text{Setup}(1^{\lambda}); \\ \pi \leftarrow \text{Prove}(params, x, w, M_L): (params, \pi) \end{array}\right\} \tag{10}$$

$$\text{Sim}(\lambda)=\left\{(params, \pi) \leftarrow \text{SimProveOne}(1^{\lambda}, x, M_L): (params, \pi)\right\} \tag{11}$$

$$b \leftarrow A^{\text{Prove}(params)} \tag{12}$$

3.5. Scheme comparison

In this chapter, we discuss and analyze the puzzle structure of Bitcoin, and propose a puzzle structure based on non-interactive zero-knowledge proof. By comparison, we can know that the hash-256 encryption algorithm is used in Bitcoin and Ethereum to ensure the identity privacy of the transaction users. as well as the CoinJoin principle adopted by Dash and the Ring Signature adopted by Monero, and the zero-knowledge proof principle adopted by Zerocoin. In addition, there is a Hawk model that uses homomorphic encryption. Although they all adopt encryption algorithms, they can only play a role in the privacy of the transaction users, but they do not have the function of protecting the privacy of transaction data. Our solution is in data privacy. And privacy and privacy have good privacy and security features.

Scheme	Representative	Privacy principle	Identity privacy	Data privacy
literature [5]	Bitcoin	HASH-256	√	×
literature [11]	Ethereum	HASH-256	√	×
literature [13]	Dash	CoinJoin principle	√	×
literature [14]	Monero	Ring signature	√	×
literature [17]	Hawk	Homomorphic encryption	√	×
This Scheme	------	Non-interactive zero knowledge proof	√	√

4. Conclusion

With the rapid development of Internet technology, the emerging technology blockchain represented by Bitcoin has gradually emerged. Blockchain technology is a decentralized infrastructure widely used in emerging digital cryptocurrencies, with the gradual adoption of Bitcoin. Accepted and received attention and research. Blockchain technology has the advantages of decentralization, trust, and data tamper resistance. It is also anonymous and has the characteristics of data transparency. However, this new pattern of transparent and transparent data in the chain also brings new security risks. With the development of various anti-anonymous identity screening technologies, these exposed contents allow attackers to track the entire historical transaction path and extract a large amount of information about the user's identity characteristics. This poses a significant security risk to user information. Establishing a perfect privacy protection mechanism has become an important factor in the development and promotion of blockchain systems.

In order to realize the transparent and transparent privacy security of the blockchain system during the transaction process, this paper proposes a privacy security scheme based on non-interactive zero-knowledge proof, and solves the privacy problem existing in the current bitcoin puzzle scheme. The transaction data of

the blockchain establishes a relatively complete privacy protection mechanism. The scheme improves the algorithm and system structure of Bitcoin puzzles, and presents the structure and algorithm of puzzle based on non-interactive zero-knowledge proof system. The security proof and analysis of the scheme are given. Finally, the transaction data privacy protection mechanism adopted in this paper improves the security of the system, but the computational and communication overhead of the system is still large, the complexity of the algorithm is also great, the system efficiency needs to be improved, and a secure and efficient blockchain system is established. It is the focus of our further research in the future.

5. Acknowledgement

The authors would like to thank the National Natural Science Foundation of China (No.61462060, No.61562059) for its support.

6. References

[1] Nakamoto S. Bitcoin: A peer-to-peer electronic cash system. 2008.

[2] Barber S, Boyne X, Shi E, et al. Bitter to Better — How to Make Bitcoin a Better Currency[M]. Financial Cryptography and Data Security. Springer Berlin Heidelberg, 2012:399-414.

[3] Luu L, Narayanan V, Zheng C et al. A Secure Sharding Protocol For Open Blockchains[C]. ACM Sigsac Conference on Computer and Communications Security. ACM, 2016:17-30.

[4] Decker C, Guthrie J, Seidel J, et al. Making Bitcoin Exchanges Transparent[C]. European Symposium on Research in Computer Security. Springer, Cham, 2015:561-576.

[5] http://www.marketwatch.com/story/and-2016s-best-performing-commodity-is-bitcoin-2016-12-228

[6] A. Ekblaw, A. Azaria, J.D. Halamka, A. Lippman, A case study for blockchain in healthcare: medrec prototype for electronic health records and medical research data, 2016.

[7] Azaria A, Ekblaw A, Vieira T, et al. MedRec: Using Blockchain for Medical Data Access and Permission Management[C]. International Conference on Open and Big Data. IEEE, 2016:25-30.

[8] Xiao Y, Wang H, Jin D, et al. Healthcare Data Gateways: Found Healthcare Intelligence on Blockchain with Novel Privacy Risk Control[J]. Journal of Medical Systems, 2016, 40(10):218.

[9] Huckle S, Bhattacharya R, White M, et al. Internet of Things, Blockchain and Shared Economy Applications[J]. Procedia Computer Science, 2016, 98(C): 461-466.

[10] Dwyer, Gerald P . The economics of Bitcoin and similar private digital currencies[J]. Journal of Financial Stability, 2015, 17:81-91.

[11] John Douceur. 2002. The Sybil Attack. In IPTPS '02: Proceedings of the 1st International Workshop on Peer-to-Peer Systems.

[12] Tschorsch F, Scheuermann B. Bitcoin and Beyond: A Technical Survey on Decentralized Digital Currencies[J]. IEEE Communications Surveys & Tutorials, 2016, 18(3):2084-2123.

[13] Rivest R, Shamir A, Tauman Y. How to leak a secret. Advances inCryptology—ASIACRYPT 2001, 2001: 552-565.

[14] Miers I, Garman C, Green M, et al. Zerocoin: Anonymous Distributed E-Cash from Bitcoin[J]. IEEE Symposium on Security and Privacy, 2013:397-411.

[15] Sasson E B, Chiesa A, Garman C, et al. Zerocash: Decentralized Anonymous Payments from Bitcoin[C]. IEEE Symposium on Security and Privacy. IEEE Computer Society, 2014:459-474.

[16] S.Goldwasser, S.Micall—and C Raekoff—The knowledge complexity of interactive proof—systems, SIAM J. Comput. , 18(1 989), PP. 186-208(Proceedings of STOC 1985, PP. 291～304).

[17] Ben-Sasson E, Chiesa A, Genkin D, et al. SNARKs for C: Verifying Program Executions Succinctly and in Zero Knowledge[M]. Advances in Cryptology – CRYPTO 2013. Springer Berlin Heidelberg, 2013:90-108.

[18] M. Blum .P Feldman, and S M icali, Non—interactive zero knowledge proof systems and applications, in Proc.20th Annual ACM Symposium on Theory of Computing, Chicago 1988, PP. 103-112.