One-Stop Efficient PKI Authentication Service Model Based on Blockchain

Tao Feng(✉), Wuyang Chen(✉), Di Zhang, and Chunyan Liu

Lanzhou University of Technology, Lanzhou 730050, China
fengt@lut.cn, 820680184@qq.com

Abstract. Public Key Infrastructure (PKI) technology is a widely used identity authentication technology. This paper uses blockchain technology to improve it and implements decentralized PKI authentication, which resolves the issues in the traditional PKI such as single point of failure and certificate transparency. However, most of the current research uses the method of traversing the blockchain to query the certificate (identity, public key) to realize identity authentication, which is inefficient. And as the size of blockchain continues to grow, storage overhead is growing. In this paper, we combine the blockchain and the dynamic accumulator to construct a blockchain PKI model that can batch update certificates, which improves the efficiency of identity authentication. The model can effectively add, revoke and update user certificates. Meanwhile, this paper builds a one-stop PKI authentication service model based on blockchain, Through the certificate blockchain, we can provide one-stop user authentication service to third-party service providers. Finally, we verify the security and effectiveness of the scheme.

Keywords: Blockchain · Dynamic accumulator · PKI · One-stop identity authentication

1 Introduction

PKI is a universal security infrastructure that provides information security services based on public key cryptography, so that users can communicate and make e-commerce transactions through a series of trust relationships based on certificates when they do not know each other's identity. As the foundation and core of current network security, PKI is the basic guarantee for e-commerce security development. To ensure the secure transmission of information, an effective PKI system must be secure and transparent. However, faced with the biggest problem that the CAs are not trusted, traditional centralized PKI in a distributed environment results in an untrustworthy problem of the identity of the entity. A CA that is attacked or maliciously issues certificate will bring significant security risks to the information system. The hacker can achieve a man-in-the-middle attack by attacking a trusted CA to perform malicious operations, such as issuing a user's certificate containing false information. The user cannot verify the process of issuing a certificate by CAs, and there is a certificate transparency issue. In addition, the centralized CA management architecture will lead to

© Springer Nature Singapore Pte Ltd. 2020
X. Si et al. (Eds.): CBCC 2019, CCIS 1176, pp. 31–47, 2020.
https://doi.org/10.1007/978-981-15-3278-8_3

single point of failure [1]. As a new type of distributed technology that cannot be tampered with, blockchain brings new ideas to the implementation of decentralized PKI.

At present, the blockchain-based PKI uses the blockchain to store information such as identity and public key. In the process of implementing identity authentication, the method of traversing the blockchain is generally used to look up the certificate, and then check whether the public key belongs to its declared identity. Finally, verifying the digital signature to determine whether the other party holds the matching private key by sending a challenge information. However, the block-chain is a public chain that can only be added. Its characteristics ensure that the amount of data will continue to grow. In recent years, the blockchain has exceeded 100 Gb in volume and will continue to grow in the future. By then, the method of traversing the blockchain will be more inefficient, and the time required for identity authentication will be difficult to meet the actual needs. At the same time, such a large amount of data cannot be stored for carriers such as mobile phones. The dynamic accumulator maps a collection containing multiple elements to an accumulated value and provides a smaller witness to prove that a given element does belong to the set. Its introduction can resolve the issue that member verification is inefficient in the process of identity authentication.

In this paper, we improve the traditional PKI model by using dynamic accumulator and blockchain, and propose a PKI authentication service model based on blockchain. First, we build an interaction model between users, miners, and supervisory nodes. The miner is responsible for the distribution and management of the certificate, at the same time, provides authorization tickets to third-party service providers. The supervisory node reviews the transaction submitted by the user and ensures the consistency of the block transaction with miners through the consensus mechanism. It resolves the problem of single point of failure and certificate transparency in the traditional PKI. Secondly, in view of the shortage of certificate management methods, this paper proposes a certificate management method that can batch update and revoke certificates based on dynamic accumulators, which improves the efficiency of identity authentication. Thirdly, this paper builds a one-stop PKI authentication service model based on blockchain to ensure that users can access the third-party services by registering the certificate simply in the certificate blockchain. Finally, we analyze the security of the scheme in detail, and the results show that the scheme can resist the enemy forgery attack and Sybil attack. In terms of efficiency, the space complexity of storage overhead is $O(1)$.

The rest of the paper is organized as follows: In Sect. 2, we introduce the relevant research of blockchain-based PKI systems. In Sect. 3 we introduce the supporting techniques of this scheme. In Sect. 4, we describe the system model, security model and threat model. The specific construction of the program is discussed in detailed in Sect. 5. A security analysis and efficiency analysis for the scheme are described in Sect. 6. In Sect. 7 we compare the relevant scheme. Section 8 draws a conclusion of our scheme.

2 Related Work

An important application of blockchain in the direction of identity authentication is to build a distributed public PKI based on blockchain [2]. PKI can be established based on public general ledger, which can eliminate the trust center CA of PKI and realize real distributed PKI construction.

In 2014, MIT scholar Conner proposed the first distributed PKI solution based on blockchain which called Certcoin [3, 4]. The core idea is to record the user certificate through the public general ledger, and associate the user identity with the certificate public key in a public manner to realize the decentralized PKI construction. Any user can query the certificate issuance process and resolve the issue of certificate transparency and CA single point of failure. Certcoin implements the registration, update and revocation of certificates by publishing users and their public keys in the form of blockchain transactions. The normal operation of the PKI is guaranteed by the attributes of the blockchain that cannot be tampered with. The Merkle root only records the hash value of the transaction, and users do not need to download all blockchain transaction data to complete the verification of the certificate. However, on the one hand, Certcoin cannot prevent the illegal occupancy of legitimate users like other schemes. On the other hand, the scheme completes the user's certificate revocation by retaining the certificate blacklist and periodically recalculating the accumulator from zero, which will increase the computational overhead.

Authcoin is a decentralized PKI scheme proposed by Benjamin [5]. To reduce illegal occupancy and Sybil attack, Authcoin emphasizes the actual binding of the user when registering the public key by adding a complex challenge response step that makes it resilient to Sybil attack. However, as the number of interactive communication steps increases, so does the performance cost. This scheme does not take into account the credibility of the person performing the operations during the verification and authentication process.

BIX protocol is more flexible for cyber-attack and doesn't cause single point of failure [6]. The BIX protocol is designed to distribute the role of CAs and preserve security features. In fact, the BIX protocol is designed with a blockchain-like structure, with a decentralized structure replacing CAs, which implements distributed certificate distribution. The certificate is a block in the blockchain, an effective user can attach their certificates to the blockchain by proper interaction protocol. Then Longo et al. proposed improvements to the BIX protocol and security proof. The formalized analysis shows the PKI system based on BIX protocol is more suitable for large-scale network attacks than the standard PKI protocol based on CA [7]. However, the protocol is still incomplete and there are no steps to revoke and update certificates.

Matsumoto et al. proposed a timely and automatic response PKI framework IKP (instant karma PKI) [8]. Based on the Ethereum platform, IKP uses the smart contract and consensus mechanism to stimulate the CA center to issue certificates correctly. It introduces detector to give reward to report illegal certificates, and imposes financial penalties on CAs that issue illegal certificates. In addition, the detector also needs to pay for the report. If the reported certificate is indeed an illegal certificate, the detector will receive a corresponding reward which can effectively prevent the detector from

reporting all certificates to defraud the reward. However, the problem is that a malicious user may maliciously register a fake identity for execution fraud.

BKI is a blockchain-based PKI [9]. It uses a tunable number of CAs to issue certificates, but it is not extendable. In addition, BKI requires all clients to contact third parties (blockchain-based log maintainers) during certificate verification, which can cause latency and privacy issues. Syta et al. proposed an efficient method for joint signature of statements issued by CA using multiple signatures [10]. Each certificate requires a certain number of witnesses to sign together in order to be accepted by others. Therefore, even if an attacker compromises a certain privilege, all malicious statements need to be made public before being used for the attack. But CoSi needs to be coordinated in the cosign protocol and relies on direct communication between witnesses. In addition, the security of CoSi is still limited by its weakest link, because witnesses only approve statements issued by CA, without full domain verification, and the attacker can still exploit the vulnerability. Based on base on BKI and Cosi, Dykcik et al. propose an automated public key infrastructure relying on smart contracts called BlockPKI [11], in which CAs use multi-signature to sign and verify certificates. BlockPKI uses the smart contract to realize the automated certificate creation and the automated domain verification, and it encourages the CAs to participate in the authentication and obtain the reward.

Qin et al. proposed a distributed certificate scheme called Cecoin [12]. Cecoin treats certificates as currency processing and records them on the blockchain to eliminate single points of failure. Miners can verify the validity of a certificate against a set of rules to ensure consistency of ownership and allow identity to bind multiple public key certificates. At the same time, based on the Merkle Patricia tree, this paper describes the distributed management of certificates, including efficient retrieval and verification of certificates, and fast operations, also supports the transaction of certificates. However, this solution does not consider the correspondence between nodes and identities. One identity can correspond to several certificates, which will lead to the risk of being attacked by Sybil. At the same time, for the average user, it cannot withstand the huge storage overhead brought by the distributed certificate library.

3 Preliminary Knowledge

3.1 Cryptographic Accumulator

Benaloh et al. first proposed the use of a cryptographic accumulator as a decentralized digital signature alternative in 1993 [13]. It is a constant size representation of a set of elements. When an element is added to the cryptographic accumulator, a witness is generated that can be used to prove that the added element has been accumulated.

Definition 1. The cryptographic accumulator scheme consists of the following four polynomial time algorithms:

KeyGen(k, M): A probabilistic algorithm for instantiating a scheme. Enter the security parameter 1^k and the upper bound M on the number of accumulated elements, returning an accumulator key $\mathcal{P} = (PK, SK)$ where PK is the public key and SK is the private key.

AccVal$(L, \mathcal{P})$: A probabilistic algorithm for calculating the cumulative value. Enter a set of elements $L = \{c_1, \ldots, c_m\}(1 \leq m \leq M)$ based on set C and parameters $\mathcal{P}$, returning an accumulated value v and auxiliary information Aux that can be used by other algorithms.

WitGen$(a_c, Aux, \mathcal{P})$: A probabilistic algorithm that generates a witness for an element. Enter auxiliary information Aux, parameters $\mathcal{P}$, and elements $c_i\{i = 1, \ldots, m\}$, and if the element c_i is indeed in the collection L, return a corresponding witness W_i.

Verify(c_i, W_i, v, PK): A deterministic algorithm that checks if a given element is in the accumulated value v. Input c_i, W_i, v, and accumulator public key PK, verify whether c_i is accumulated in v according to W_i, then output Yes or No.

Applying a password accumulator to authentication not only enables efficient authentication, but also ensures security. However, when a general password accumulator adds or deletes an element, it needs to recalculate the current accumulated value and the respective witnesses. The accumulator cannot operate efficiently to cope with the actual application requirements when the element set dynamically changes. How to ensure that the accumulated value and the witness of each element can be updated and revoked efficiently when the set of elements changes. Thus, Camenisch and Lysyanskaya proposed the concept of a dynamic accumulator [14]. The dynamic accumulator accumulates a set of input values into a value such that the input values can prove themselves in the accumulated value, while allowing the operator to dynamically add or delete a value such that the cost of adding or deleting is independent of the number of members being added. In 2008, Peishun Wang et al. summarized the formal definition of the accumulator and proposed a new dynamic accumulator [15]. The dynamic accumulator adds adding, deleting and updating operations on the four algorithms of the original accumulator scheme.

Definition 2: A dynamic accumulator consists of the following seven polynomial time algorithms:

KeyGen, AccVal, WitGen and Verify are consistent with the algorithm in Definition 1.

Add$(L^+, Aux, v, \mathcal{P})$: A probability algorithm for adding new elements to the accumulated value. Enter a set of new elements $L^+ = \{c_1^+, \ldots, c_k^+\}(L^+ \subset C, 1 \leq k \leq M - m)$ that are to be added, the auxiliary information Aux, the accumulated values v and the parameters $\mathcal{P}$, return the new accumulated value v' corresponding to the set $L^+ \cup L$, the witness $W_1^+, \ldots, W_k^+$ of the newly added element $\{c_1^+, \ldots, c_k^+\}$ and the new auxiliary information Aux' for future updates.

Delete$(L^-, Aux, v, \mathcal{P})$: A probability algorithm for deleting certain elements. Enter a set of elements $L^-\{c_1^-, \ldots, c_k^-\}$ $(L^- \subset L, 1 \leq k < m)$ that are to be deleted, auxiliary information Aux, the accumulated values v and the parameters $\mathcal{P}$, and output a new accumulated value v' corresponding to the set $L \backslash L^-$, and the new auxiliary information Aux' being used in future update operations.

UpdWit(W_i, Aux, pk): Deterministic algorithm for updating the witness of element which has been added to v'. Enter the witness W_i, the auxiliary information Aux, and the accumulator public key pk, return an updated witness W_i'.

3.2 Complexity Assumption

Let $n = pq$, p, q are different odd prime numbers, so the elements in the multiplicative group Z_n^* which contains $\phi(n) = (p-1)(q-1)$ elements are all positive integers smaller than n and mutually prime with n. $\phi(n)$ is the Euler function and $\phi(n^2) = n\phi(n)$. Carmichael number $\lambda(n) = lcm(p-1, q-1)$, $\lambda(n^2) = lcm((p-1)p, (q-1)q)$. There are three difficult assumptions described as below.

Strong RSA Assumption: Given the security parameters n and random numbers $y \in Z_n^*$, there is no polynomial time algorithm to find s and x make $y \equiv x^s (\text{mod } n)$.

CSR Assumption: Given security parameters n, integers $s \in Z_{n^2}^* (s > 2)$ and random numbers $y \in Z_{n^2}^*$, there is no polynomial time algorithm to find out $x \in Z_n$ and make $y \equiv x^s (\text{mod } n^2)$.

es-RSA Assumption: Given the security parameters n and random numbers $y \in Z_{n^2}^*$, there is no polynomial time algorithm to find s and $x \in Z_n$ make $y \equiv x^s (\text{mod } n^2)$, where $n^2 > s > 2$.

Lemma 1: If the CSR hypothesis and the strong RSA assumption are true, the es-RSA assumption is true.

4 System Model

Conner Fromknecht proposed in Certcoin that there are two ways to deploy a password accumulator in a blockchain [3]. One is that each user node maintains its own password accumulator, and the other is that the entire blockchain maintains one Cryptographic accumulator. Since the general cryptographic accumulator accumulates the number of elements subject to the threshold, it is not sufficient to maintain only one accumulator in the blockchain, especially as the number of users in the blockchain increases. Therefore, this paper adopts the method of grouping users. Each user group jointly maintains a password accumulator. Since this paper uses the dynamic accumulator proposed by [15], its own function of batch dynamic update members can be a very good solution to the problem of not being able to effectively test new members (values) in [3]. Compared with the global accumulator and the solution for accumulator information attached to each block proposed in [3], our solution is relatively simple, and the required storage space is small, which effectively saves computational overhead and improves verification efficiency.

The system model of the proposed scheme is shown in the Fig. 1. The whole system includes five participating entities: user, miner node, supervisory node, certificate blockchain and third-party service provider.

- User: Submit the identity and its own public key to the supervisory node for investigation in the registration phase. After joining the system, apply to the miner node or query the blockchain to obtain its own witness for future identity authentication.
- Miner node: Initialize the system, generate the system parameter, accumulate the initial participating user information and output the initial accumulated value and the witness corresponding to each user. Select certificate transactions signed by the

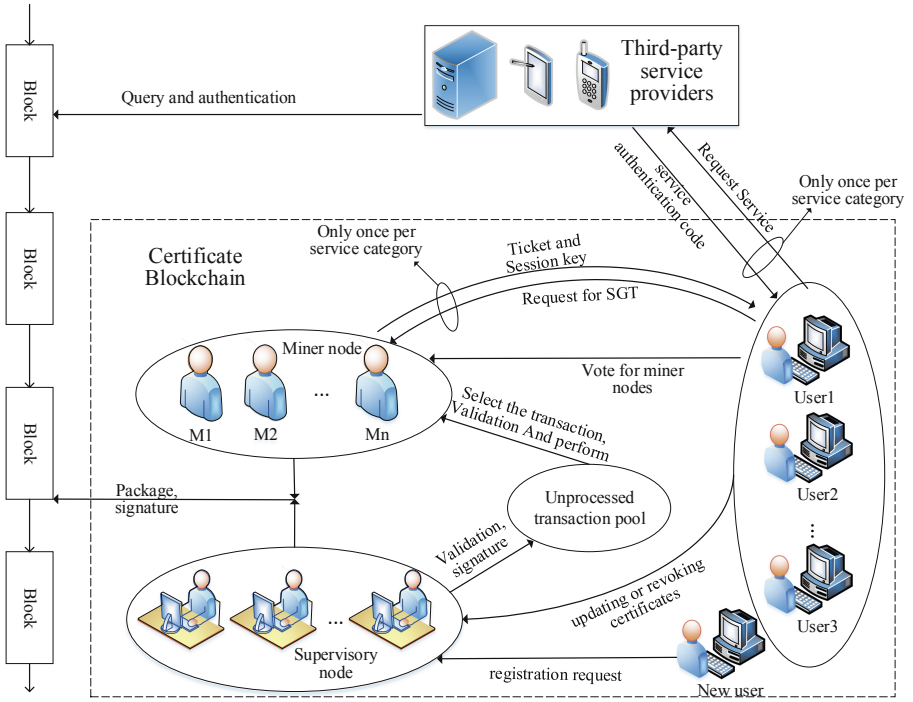

Fig. 1. One-stop PKI authentication service model based on blockchain

supervisory node and execute the corresponding algorithm, then package the corresponding information into blocks for broadcast to the network. Provides the user with a Server-Granting Ticket (SGT). Receives the authorization request sent by the user, verifies and returns session key $K_{c,V}$ and the SGT $Ticket_V$ for the user to use. The miner node was initially 21.

- Supervisory node: It is composed of 11 institutions (such as government agencies, core enterprise nodes, etc.), which are responsible for receiving user certificate registration, update or revocation requests, and questioning the transaction initiator. After verification, sign the transaction and sent it to the Pool for further processing.
- Certificate blockchain: After the miner node broadcasts new block information, the supervisory node and other miner nodes respectively verify the block, and after the consensus is reached, the block is mined.
- Third-party service providers: Provide third-party applications or services to users. Receive user service requests, verify and provide related services. Third-party service providers itself have completed the authentication in the certificate blockchain, that is, each service *ID* has a corresponding witness.

When A proves identity to B, since the nodes in the blockchain are divided into full nodes and light nodes, the efficiency of identity verification is different corresponding to the different node states of B. If B is a full node, you must query all the locally stored information on the chain, that is, traverse the entire blockchain. The authentication

efficiency decreases as the blockchain size increases. If B is a light node, the local area is not stored. Blockchain, unable to authenticate, can only request queries from all nodes, which will traverse the blockchain again. The introduction of a dynamic cryptographic accumulator can alleviate the problem of reducing authentication efficiency due to the increase in blockchain size. The authentication procedure of introducing of the dynamic accumulator is improved as follows:

1. A sends to B (c_A, W_A, pk_A, v_A), where $c_A = h(id_A, AD_A)$, h is a hash function, AD_A is a hash of the network address which uses the unidirectionality of the hash function to guarantee One-to-one correspondence between c_A, id_A and AD_A, at the same time, it can ensure that the private information is not stolen. It is called that witness W_A belongs to user c_A.
2. B compares the accumulated value v with v_A which query from the blockchain, if they are consistent, then B runs the algorithm $\text{Verify}(c_A, W_A, v, PK)$ to verify Whether the user's identity and witness are legal.
3. B sends a random challenge string ch to A, A signs $\sigma = sig(sk_A, ch)$ for the information containing the string.
4. B uses pk_A to verify the digital signature, if $\text{Verify}(pk_A, \sigma, ch) = 1$, it proves that A holds the private key sk_A, that means A has identity $c_A(id_A, AD_A)$.

4.1 Threat Model

This article assumes that communication is secure, it means the private keys of participating entities and systems are not compromised. This makes the supervisory nodes in the proposed scheme completely credible; most miners are honest but curious, will participate in block production and certificate registration, revocation and update according to the rules, but may steal users when participating. Identity privacy information. For the miner node M1 that is partially faulty or evil, when it does not produce the block or even falsify the false certificate according to the regulations, the right of the production block is handed over to the next miner node M2; some users are malicious and may initiate false transactions and malicious preemption registration, even forgery of identity and witness.

4.2 Security Model

In this paper, we define the security model of this scheme by the Chosen Element Attack security game which is described as follows:

Setup: The challenger $\mathcal{B}$ executes the initialization algorithm, and the adversary $\mathcal{A}$ adaptively selects a set of elements $L^* \in C$ to send to the challenger who calculates their accumulated value and witness return to the adversary.

Query: The adversary $\mathcal{A}$ chooses the element to be added or the element to be deleted and sends it to the challenger $\mathcal{B}$. The challenger returns the witness of the added element after adding or deleting, the new accumulated value, and the auxiliary information of the updated witness. The adversary calculates the witness after each element is updated.

Challenge: After performing several inquiries, the adversary $\mathcal{A}$ selects a set of elements $L \in C$ to send to the challenger $\mathcal{B}$, and the challenger returns the corresponding accumulated value and witness. The adversary gives an element c_i and its corresponding witness W_i, then sent them to the challenger who verifies whether the element and its corresponding witness are legal, that is mean, whether the element has been accumulated in the accumulator.

If the polynomial time adversary $\mathcal{A}$ forges a legitimate element c_i and witness W_i with a non-negligible advantage, the witness W_i can prove that the element is included in the set corresponding to the accumulated value, which means that the adversary can forge a legal certificate, and the adversary wins this game.

5 Specific Construction

In this part, we present the specific algorithm structure and concrete implementation of the blockchain-based PKI authentication service model.

1. *Initialization:* First, a node group elects miner nodes according to the consensus mechanism such as DPOS, and the miner node M1 with the highest weight creates a security parameter n of length k-bit and an empty set A_u. Let $C = Z_{n^2}^* \backslash \{1\}$, $T' = \{3, \cdots, n^2\}$, set the initial participating member list $L = \{c_1, \ldots, c_m\}$, the number of members m $1 \leq m \leq M$, then proceed with the following steps:

   - Adaptability choose $\sigma \in Z_{n^2}$, calculate $\beta = \sigma\lambda \bmod \phi(n^2)$, $\beta \in T'$. Uniform random choose $\gamma \xleftarrow{R} Z_{\phi(n^2)}$, $\gamma \notin (\beta, \sigma)$, remember the dynamic accumulator key $\mathcal{P} = (PK, SK)$, where $PK = (n, \beta)$ $SK = (\sigma, \lambda, \gamma)$.
   - Choose $c_{m+1} \xleftarrow{R} C$, calculates

$$
\begin{aligned}
x_i &= F(c_i^{\gamma\sigma^{-1}} \bmod n^2) \bmod n \quad (i = 1, \cdots, m+1), \\
v &= \sigma \sum_{i=1}^{m+1} x_i \bmod n, \\
y_i &= c_i^{\gamma\beta^{-1}} \bmod n^2 \quad (i = 1, \cdots, m+1), \\
a_c &= \prod_{i=1}^{m+1} y_i \bmod n^2
\end{aligned}
\tag{1}
$$

     Output initial v_0, auxiliary information a_c and $A_l = (y_1, \cdots, y_m)$. P.S. $F(x) = (x-1)/n$.
   - Package v_0, a_c, A_l and other related parameters into block and broadcast to network. If block is verified by other miners and supervisory nodes, mining blocks will be successful. Otherwise mining right takes turns to the next miner node M2. The initialization is completed.

2. *Certificate generation:* After the system is initialized, the accumulated users can calculate their own witnesses according to the information disclosed on the blockchain, or they can initiate an application to the miner node, and the miner node signs the corresponding witness. The specific steps are as follows:

   Query to get existing auxiliary information a_c, A_l parameters $\mathcal{P}$, and randomly select a collection $T = (t_1, \cdots, t_m) \subset T' \backslash \{\beta, \gamma\}$ and calculated:

$$w_i = a_c y_i^{\frac{-t_i}{\gamma}} \bmod n^2 \ \ (i = 1, \cdots, m) \tag{2}$$

   $W_i = (w_i, t_i)$ is the witness for the user c_i. Think of the $(c_i = h(id_i, AD_i), W_i, pk_i, v_i)$ quad as the user's public key certificate.

3. *Verify:* Give c_i, W_i, v and PK, check if $\{c_i, w_i\} \subset C$, $t_i \in T'$ and $F(w_i^{\beta} c_i^{t_i} \bmod n^2) \equiv v \,(\text{mod } n)$, if true, output Yes which proved the user c_i has indeed been accumulated in v, otherwise output No.
4. *New user certificate registration:* The new user c_i^+ submits encrypted identity information c_i, id_i, AD_i, and public key pk_i to the supervisory node, initiates a registration transaction request, the supervisory node checks $c_A = h(id_A, AD_A)$ and initiates an acknowledgment to the network address. If supervisory node receives the acknowledgment, that is, the verification transaction can be legal. Then supervisory node signs and puts the transaction into the unprocessed transaction pool. The miner node selects some new user certificate registration transactions from the pool, which is recorded as the set $L^+ = \{c_1^+, \ldots, c_k^+\}$ to be added. Then select $c_{k+1} \xleftarrow{R} C$ and $T^+ = \{t_1^+, \ldots, t_k^+\} \xleftarrow{R} T' \backslash \{T \cup \{\beta, \gamma\}\}$, calculate:

$$\begin{aligned}
x_i^+ &= F((c_i^+)^{\gamma\sigma^{-1}} \bmod n^2) \bmod n \ \ (i = 1, \cdots, k+1), \\
v' &= v + \sigma \sum_{i=1}^{k+1} x_i^+ \bmod n, \\
y_i^+ &= (c_i^+)^{\gamma\beta^{-1}} \bmod n^2 \ \ (i = 1, \cdots, k+1), \\
a_u &= \prod_{i=1}^{k+1} y_i^+ \bmod n^2, \\
w_i^+ &= a_u a_c (y_i^+)^{\frac{-t_i^+}{\gamma}} \bmod n^2 \ \ (i = 1, \cdots, k+1)
\end{aligned} \tag{3}$$

   Let $T = T \cup T^+$, $A_u = A_u \cup \{a_u\}$, $a_c = a_c a_u \bmod n^2$, then get new accumulated values v', new auxiliary information a_u, a_c and new witnesses $W_i^+ = (w_i^+, t_i^+)$ of user c_i^+. Being similar to the initialization, the miner node packages the corresponding information and broadcasts, and other miners add the new block to the blockchain. In addition, the recommended value is in the actual application.

5. *User certificate revocation:* The pre-revoked user c_i^- presents his own witness $W_i = (w_i, t_i)$ and signature σ to the supervisory node, initiates an identity revocation

request, and also counts the signature into the unprocessed transaction pool. The miner node selects some user identity revocation transactions from the unprocessed transaction pools, which is recorded as the set $L^{-}\{c_1^{-}, \ldots, c_k^{-}\}$ $(L^{-} \subset L, 1 \leq k < m)$ to be revoked. For a user identity revocation transaction, the supervisory node first verifies the signature σ to verify that the witness actually belongs to the user, and then proceeds to step 3 to verify that the user identity has been accumulated.

If yes, select $c_{k+1}^{-} \xleftarrow{R} C$ and calculate:

$$\begin{aligned} x_i^{-} &= F((c_i^{-})^{\gamma\sigma^{-1}} \bmod n^2) \bmod n \ \ (i = 1, \cdots, k+1), \\ v' &= v - \sigma \sum_{i=1}^{k} x_i^{-} + \sigma x_{k+1}^{-} \bmod n \\ y_i^{-} &= (c_i^{-})^{\gamma\beta^{-1}} \bmod n^2 \ \ (i = 1, \cdots, k+1), \\ a_u &= y_{k+1}^{-} \prod_{i=1}^{k} (y_i^{-})^{-1} \bmod n^2 \end{aligned} \tag{4}$$

Let $a_c = a_c a_u \bmod n^2$, $A_u = A_u \cup \{a_u\}$ then get the new accumulated value v', the new auxiliary information a_c and a_u. Then, being similar to the initialization, the miner node packages and broadcasts the corresponding information. Other miners and supervisory nodes verify and add the new block to the blockchain, and the certificate revocation transaction is recorded on the chain. The witness $W_i = (w_i, t_i)$ expires, that is, the user certificate has expired.

6. *Certificate update:* There are two ways to update a certificate. The first is to update the witness only. User presents his own witness and signature to the supervisory node, initiates a certificate update transaction request, which is verified into the unprocessed transaction pool. The miner node selects some user certificate update transactions from the pool, which is recorded as the set $L'\{c_1, \ldots, c_k\}(L' \subset L, 1 \leq k < m)$ to be updated. Then it calculates $w_i' = w_i a_u \bmod n^2$ and the user's update witness is $W_i' = (w_i', t_i)$. t_i is generated when the user is added to the accumulator, it remains the same, and only changes with witness update and other transactions. Therefore, t_i can also be used as an alternative identifier in the accumulator.

   It should be noted that each certificate has a corresponding time stamp and accumulator related information. Whenever a miner performs a certificate transaction, the parameters a_u are updated once and are credited to the collection A_u. When a user initiates a certificate update transaction, the miner needs to query the user's for finding the elements $a_{u_i}(i = 1 \cdots k)$ in the collection A_u from last certificate update or registration to this time, k is the number of times for a_u changes between the last certificate update and this transaction. Calculate $a_u = a_{u_1} \cdots a_{u_k} \bmod n^2$, $w_i' = w_i a_u \bmod n^2$, then the user's new witness is $W_i' = (w_i', t_i)$. In this way, the user certificate update is independent of the change of the accumulated value.

The second way is to update the witness and key. The user submits c_i, $W_i = (w_i, t_i)$, pk_i, pk_i', AD_i and v_i, where pk_i' is the new public key. When the user issues a request transaction to update the key, the supervisory node first performs a third step to verify that the user is registered and verifies the consistency of the network address with the user. Then find the current certificate of the user and verify that pk and pk_i are consistent. This is to prevent the adversary from maliciously updating the user certificate with the old public key that the user has previously leaked. After the verification, the miner updates the user's witness, then packages the user's new public key and other information into block and broadcastes. Other nodes verify the block and add it to blockchain.

7. *User-service authentication exchange:* The specific description is shown in Table 1. The user c sends c, id_V of the service and the witness W_c to the miners for Server-Granting Ticket, and the miner returns the session key $K_{c,V}$ and SGT $Ticket_V$. Then user c sends $Ticket_V$ and $Authenticator_c$ to the third-party service provider V, and the server provider gives corresponding respond after verification. If mutual authentication is required, a reply message should be sent to c according to message (4) by V. Obviously, the message is encrypted by $K_{c,V}$, which guarantees that the message is only generated by V, and confirms the source of the message by verifying W_V.

$Authenticator_c$ is a legal authentication ticket generated by the user which ensures that the owner of the ticket is the same as the owner when the SGT generated. $Authenticator_c$ can only be used once and has a very short lifetime. Miner queries blockchain for pk_c according to id_c and W_c to make decryption of the authentication ticket. The session key $K_{c,V}$ is issued by the miner to ensure secure exchange of information between the user and the third-party service provider. AD_c is network address that used to prevent the ticket from being used on wrong workstations. *Lifetime* is used to prevent the ticket from using after it expires; *TS* is a timestamp for the ticket.

Table 1. User-service authentication exchange

Object	The message format
(1) $c \rightarrow$ Miner	$c \| \| W_c \| \| id_V \| \| Authenticator_c$
(2) Miner $\rightarrow c$	$c \| \| Ticket_V \| \| E(pk_c, [K_{c,V} \| \| id_V \| \| TS_2])$
	$Ticket_V = E(pk_V, [K_{c,V} \| \| id_c \| \| AD_c \| \| TS_2 \| \| Lifetime])$
	$Authenticator_c = E(sk_c, [AD_c \| \| TS_1])$
(3) $c \rightarrow V$	$c \| \| W_c \| \| Ticket_V \| \| Authenticator_c$
(4) $V \rightarrow c$	$E(K_{c,V}, [W_V \| \| TS_3 + 1])$
	$Ticket_V = E(pk_V, [K_{c,V} \| \| id_c \| \| AD_c \| \| TS_2 \| \| Lifetime])$
	$Authenticator_c = E(sk_c, [AD_c \| \| TS_3])$

6 Security Analysis

According to the attack form summarized in the threat model, if the user issues a false transaction, the miner node can detect whether the transaction is legal when packing the block. When the enemy maliciously seizes the identity of others for registration, he must submit $c = h(id, AD)$ and pk_c, and the supervisory node initiates an acknowledgment to the network address. If the network address is false, no agreement can be reached between c, id and AD, no malicious preemption is formed. If true, the preempted user will receive a confirmation message, then he can refuse to register and the transaction is invalid. In order to achieve the preemption registration, the adversary must ensure that the preempted user cannot receive the confirmation message and reply to the supervisory node with the correct network address, so that the certificate ownership belongs to the preempted user, as long as he logs in, the certificate can be found, and the preempted user can revise the certificate at any time.

Theorem 1. Based on the es-RSA assumption, this scheme can resist Chosen Element Attack.

Proof. Assume that a polynomial time adversary $\mathcal{A}$ wins a CEA game with a non-negligible advantage in a defined security model, which means that for input (n, β), the adversary $\mathcal{A}$ gets l elements $L^* : \{c_1, \cdots, c_l\} \subset C$, $\{W_1, \cdots, W_l\}$ and corresponding accumulated values v, he can find element $c' \in C \backslash \{c_1, \cdots, c_l\}$ and corresponding $W' = (w', t')$ make $F(w'^{\beta} c'^{t'} \bmod n^2) \equiv v (\bmod\ n)$ with a non-negligible advantage. This paper constructs the following simulator $\mathcal{B}$ to break the es-RSA hypothesis with a non-negligible advantage.

Initialization. $\mathcal{B}$ runs the initialization algorithm and gets the relevant system parameters, the accumulated values v and witnesses of l elements $L^* : \{c_1, \cdots, c_l\} \subset C$, the adversary request, and $\mathcal{A}$ requests $L^* : \{c_1, \cdots, c_l\} \subset C$, v and $\{W_1, \cdots, W_l\}$ from $\mathcal{B}$.

Query 1. The adversary $\mathcal{A}$ selects the set of elements $L^{\pm}(L^{\pm} \subset C)$ to be added or deleted and sends them to $\mathcal{B}$. $\mathcal{B}$ runs the corresponding algorithm to complete the addition or revocation of the user certificate, and gets the new accumulated value v', the new auxiliary information a_c,a_u and the corresponding witness $W_i^{\pm} = (w_i^{\pm}, t_i^{\pm})$. Return to the adversary.

Query 2. The adversary $\mathcal{A}$ selects a set $L'(L' \subset L^*)$ of updated users to send to the $\mathcal{B}$. .. runs the algorithm to update the user certificate and returns the update witness $W_i' = (w_i', t_i)$ corresponding to the relevant user.

Challenge: After performing Query 1 and Query 2 several times, the adversary $\mathcal{A}$ selects $L : \{c_1, \cdots, c_m\} \subset C$ queries $\mathcal{B}$ for corresponding v and $\{W_1, \cdots, W_m\}$, then forges element $c'(c' \in C \backslash L)$ and its corresponding witness $W' = (w', t')$ and sends them to $\mathcal{B}$. $\mathcal{B}$ runs the algorithm and verifies if the element c' has been accumulated in v. If the algorithm *Verify* outputs Yes with a non-negligible advantage, which means $\mathcal{B}$ can break the es-RSA assumption with a non-negligible advantage.

$\mathcal{B}$ calculates v and $\{W_1, \cdots, W_m\}$ corresponding to $L : \{c_1, \cdots, c_m\}$ and therefore exist $F(w_i^{\beta} c_i^{t_i} \bmod n^2) \equiv v (\bmod\ n)$, $(i = 1, \cdots, m)$, which means that:

$$\exists k \in Z, \frac{w_i^{\beta} c_i^{t_i} \bmod n^2 - 1}{n} = kn + v \tag{5}$$

Therefore,

$$w_i^{\beta} c_i^{t_i} \equiv (vn + 1) (\bmod n^2) \tag{6}$$

Also, we have

$$w_i^{\beta} \equiv (vn + 1) c_i^{-t_i} (\bmod\ n^2), c_i^{t_i} \equiv (vn + 1) w_i^{-\beta} (\bmod\ n^2) \tag{7}$$

So there are m triplets (c_i, w_i, t_i), c_i, w_i and t_i can be calculated from Eqs. (6) and (7).

Since v is calculated by adding a random element each time an element is added or revoked, and t_i is the randomly selected, the probability distributions of v, w_i, t_i and a_u are consistent, so Query 1, 2 does not help $\mathcal{A}$ forging. If $\mathcal{B}$ breaks the es-RSA hypothesis with a non-negligible advantage, he can get a different triplet (c', w', t') make (6) true with a non-negligible advantage. At this time, we have:

$$w'^{\beta} \equiv (vn + 1) c'^{-t'} (\bmod n^2) \Rightarrow w'^{\beta} \equiv \left((vn + 1)^{-\frac{1}{7}} c' \right)^{-t'} (\bmod n^2) \tag{8}$$

If $y = w'^{\beta}$, $x = (vn + 1)^{-\frac{1}{7}} c'$, $s = -t'$, that is $y \equiv x^s (\bmod n^2)$.

Obviously, if the adversary $\mathcal{A}$ can forge a triple (c', w', t'), it can resolve Eq. (8), which is equivalent to solving $y \equiv x^s (\bmod n^2)$. This means that the es-RSA assumption is broken. This contradicts with Lemma 1. So, it can be concluded that no enemy can win the security game with obvious advantages. The scheme can defend against CEA. According to the previous security model, our scheme can prevent adversary from forging witnesses and identities.

Theorem 2. The scheme can resist Sybil attack.

Proof. Sybil attack refers to the creation of multiple account identities in one malicious node. The adversary $\mathcal{A}$ can control most of the network with few nodes to achieve refusal to deal, fork, double payment and so on. In this paper, the user's network address and the user's identity are bound, and the joining of the new node needs to be authenticated by the supervisory node, so that $\mathcal{A}$ cannot create multiple identities in one node, so the scheme can resist Sybil attack.

7 Analysis and Comparison

7.1 Efficiency Analysis

The overhead of this scheme is mainly divided into storage overhead and computational overhead, and communication overhead is not considered. For storage overhead, the user node only needs to store its own witness and the accumulated value can realize the identity verification. The miner node must retain the certificate data (c_i, W_i, pk_i, v) of the entire node group and maintain the relevant information of the accumulator (auxiliary information a_u, a_c, A_l etc.). Supervisory node is only responsible for identity information and transaction auditing, no need to store relevant information. Since the magnitude of the witness and the accumulated value are small and constant, that is, the size of the dynamic accumulator is small, the full node and the light node can complete the corresponding identity authentication at any time only by updating regularly, and the space complexity of the corresponding storage overhead is $O(1)$, which improves the efficiency of certification.

The computation overhead mainly includes requests for registration, deletion, and update of certificates. Let Md be the cost of modular operation, E be the cost of exponential operation. For a group with m initial member, the calculation cost of the scheme mainly includes:

Compute initial key: Md; Generate initial parameters: $2(m+1)E + (3m+5)Md$; Generation of each certificate: $E + Md$. Verification a certificate: $2E + 3Md$. k Certificate registration: $(3k+2)E + (4k+6)Md$. k Certificate revocation: $2(k+1)E + (3k+6)Md$. Update a certificate: $2Md$.

7.2 Scheme Comparison

The comparison between this paper and related PKI schemes is shown in Table 2. Certcoin proposed in [3] builds a PKI model based on blockchain, and uses the offline key to protect the online key. At the same time, the certificate is efficiently managed by means of RSA accumulator and distributed hash table. Aucoin is a decentralized PKI scheme [5]. The scheme uses a flexible challenge response mechanism for verification and authentication when issuing a public key, thereby reducing illegal occupancy and Sybil attack. The IKP scheme proposed in [8] uses smart contracts to reward detectors that report illegal certificates, impose financial penalties on CAs that issue illegal certificates, and to motivate CAs that work correctly to ensure proper certification. Cercoin in [12] proposed a set of rules based on the Bitcoin system to verify the validity of the certificate and the consistency of ownership, and to provide a method of identity assignment. At the same time, the scheme improves the Merkle Patricia tree to achieve efficient management of certificates, including efficient retrieval and verification of certificates.

Table 2. Comparison of this article and other PKI schemes

Scheme	CAs	Certcoin [3]	Aucoin [5]	IKP [8]	Cercoin [12]	Our scheme
Update	√	√	×	×	√	√
Revocation	√	√	×	×	√	√
Multiple certificates	√	×	√	√	√	√
Single point of failure	×	√	×	√	√	√
Resist Sybil attack	–	√	√	×	×	√
Preemptive registration	×	×	√	×	√	√
Certificate transparency	×	√	√	√	√	√
Batch update	–	×	–	–	–	√
Resist replay attack	×	×	√	√	×	√

8 Conclusion

This paper proposes a one-stop efficient PKI authentication service model based on blockchain. Firstly, we divide the node group into five different participating entities: user, miner node, supervisory node, certificate blockchain and third-party service providers, and propose a new blockchain based PKI model that resolves the single point of failure problem and can resist Sybil attack. In addition, this paper uses the witness generated by the dynamic accumulator to replace the role of certificates in traditional PKI, and proposes new user certificate management (registration, revocation and update) algorithms based on the dynamic accumulator, which improves efficiency of authentication. This paper also builds an authentication interaction model between the certificate blockchain and the third-party service providers. This model can provide a one-stop authentication service for users and third-party service providers, which will facilitate the deployment of PKI on blockchain. Finally, Security and efficiency analyses show that our scheme can effectively resist the Chosen Element Attacking, and improve the identity verification efficiency.

However, there still exists improvement spaces in our scheme. Because this article uses the network address to ensure the identity authentication, once the user's network address is changed, he must carry out the corresponding revoke and add a new certificate, this will bring inconvenience to users and improve the system overhead. In addition, the dynamic accumulator used in this paper exists much modular arithmetic, which brings high computational overhead. We will improve the dynamic accumulator to increase the calculation efficiency in future work, and further improve the model to avoid frequent certificate revocation and adding transactions in some cases.

Acknowledgment. This work is supported by the National Science Foundation of China (No. 61462060, No. 61762060).

References

1. Lin, J.Q., Jing, J.W., Zhang, Q.L.: Recent advances in PKI technologies. J. Cryptol. Res. **27** (1), 487–496 (2015)
2. Yuan, Y., Wang, F.Y.: Blockchain: the state of the art and future trends. Acta Automatica Sinica **42**, 481–494 (2016)
3. Fromknecht, C., Velicanu, D., Yakoubov, S.: CertCoin: a NameCoin based decentralized authentication system 6.857 class project. Unpublished class project (2014)
4. Fromknecht, C., Velicanu, D., Yakoubov, S.: A decentralized public key infrastructure with identity retention. IACR Cryptol. ePrint Arch. **2014**, 803 (2014)
5. Leiding, B., Cap, C.H., Mundt, T., Rashidibajgan, S.: Authcoin: validation and authentication in decentralized networks. arXiv preprint arXiv:1609.04955 (2016)
6. Muftic, S.: Bix certificates: cryptographic tokens for anonymous transactions based on certificates public ledger. Ledger **1**, 19–37 (2016)
7. Longo, R., Pintore, F., Rinaldo, G., Sala, M.: On the security of the blockchain BIX protocol and certificates. In: 2017 9th International Conference on Cyber Conflict (CyCon), pp. 1–16. IEEE (2017)
8. Matsumoto, S., Reischuk, R., M.: IKP: turning a PKI around with decentralized automated incentives. In: 2017 IEEE Symposium on Security and Privacy (SP), pp. 410–426. IEEE (2017)
9. Wan, Z., Guan, Z., Zhuo, F., Xian, H.: BKI: towards accountable and decentralized public-key infrastructure with blockchain. In: Lin, X., Ghorbani, A., Ren, K., Zhu, S., Zhang, A. (eds.) SecureComm 2017. LNICST, vol. 238, pp. 644–658. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-78813-5_33
10. Syta, E., Tamas, I., Visher, D.: Keeping authorities "honest or bust" with decentralized witness cosigning. In: 2016 IEEE Symposium on Security and Privacy (SP), pp. 526–545. IEEE (2016)
11. Dykcik, L., Chuat, L., Szalachowski, P., Perrig, A.: BlockPKI: an automated, resilient, and transparent public-key infrastructure. In: 2018 IEEE International Conference on Data Mining Workshops (ICDMW), pp. 105–114. IEEE (2018)
12. Qin, B., Huang, J., Wang, Q., Luo, X., Liang, B., Shi, W.: Cecoin: a decentralized PKI mitigating MitM attacks. Future Gener. Comput. Syst. (2017)
13. Benaloh, J., de Mare, M.: One-way accumulators: a decentralized alternative to digital signatures. In: Helleseth, T. (ed.) EUROCRYPT 1993. LNCS, vol. 765, pp. 274–285. Springer, Heidelberg (1994). https://doi.org/10.1007/3-540-48285-7_24
14. Camenisch, J., Lysyanskaya, A.: Dynamic accumulators and application to efficient revocation of anonymous credentials. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 61–76. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-45708-9_5
15. Wang, P., Wang, H., Pieprzyk, J.: A new dynamic accumulator for batch updates. In: Qing, S., Imai, H., Wang, G. (eds.) ICICS 2007. LNCS, vol. 4861, pp. 98–112. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-77048-0_8