A POST-QUANTUM PRIVACY-ENHANCING BLOCKCHAIN-BASED TRANSACTION FRAMEWORK WITH ACCESS CONTROL

LingyunLi[1, 2, 3*], XianhuiLu[1, 3], and KunpengWang[1, 3]

[1]State Key Laboratory of Information Security,
Institute of Information Engineering, Chinese Academy of Sciences, China
[2]School of Computer Science, Liaocheng University, China
[3]School of Cyber Security, University of Chinese Academy of Sciences, China

ABSTRACT

Protecting the transaction from address-based tracking is one of the core issues in blockchain privacy preservation. In this paper, we propose a transaction framework through which the trader of a transaction organization transacts on the blockchain public chain with privacy-enhancing; meanwhile, the manager gets access to the trader's transaction with access control based on cryptography. In the proposed framework, the hash-based one-time address is utilized to protect transactions from unauthorized tracking; furthermore, the hash-based one-time signature is creatively being used twice to verify and track the transactions safely in the semi-honest model; through access control, the authorized managers can obtain transaction information within their authorities. Compared with the standard Bitcoin transaction system, the proposed system achieves privacy-enhancing and post-quantum security.

KEYWORDS

Post-quantum, Privacy-enhancing, Blockchain, Security, Hash-based Signature, Security

1. INTRODUCTION

Blockchain is one of the key technologies to realize decentralized peer-to-peer transactions based on distributed storage and computation. On the one hand, the blockchain technique protects the privacy of traders to a certain extent through the anonymity of transaction; on the other hand, compared with the traditional centralized transaction mechanism, the transaction mechanism based on the blockchain exposes transactions to all the nodes in the whole system to reach consensus, analysts can obtain some transaction information of trader through public transaction information, thus violating trader's privacy. Take Bitcoin with typical blockchain mechanism as an example, which utilizes the hash value of the trader's public key as the transaction ID, the attacker can track the transaction ID in the global ledger, to obtain the fund balance and where its fund flows to of a specific account, all the transaction records of a specific coin, all related transactions of a specific transaction ID, as well as the transaction information and rules between different transaction IDs, etc. Combined with the attack techniques of the network layer such as tracking IP and propagation path, and social engineering attack techniques of application layer, etc., attackers can obtain some private transaction information of the trader. Currently, a variety of privacy preservation techniques for blockchain have been proposed, such as restricted access[1], malicious node detection and shielding[2,3], anonymous communication network[4],data distortion[5], cryptographic techniques like encryption, group signature and ring

David C. Wyld et al. (Eds): NLPML, AIAP, SIGL, CRIS, COSIT, DMA -2023
pp. 99-111, 2023. CS & IT - CSCP 2023 DOI: 10.5121/csit.2023.130809

signature technology[7], etc. However, the above techniques can only partially solve the privacy preservation issue in blockchain, and they all have corresponding drawbacks.

1.1. Motivation

Existing techniques can just partly solve the privacy preservation issues in blockchain, and few of them achieve post-quantum security. This, therefore, motivates us to propose a blockchain transaction framework, which achieves post-quantum security, and enhance the privacy based on access control to address the urgent security needs in the blockchain transactions. Specifically, imagine the following scenario, considering the high security of transactions in the public blockchain, a sales company demands its traders to directly transact through the public blockchain, avoiding the information related transactions to be obtained by others without authorities; meanwhile, in order to evaluate employees through the value, amount and other indicators of the transactions by the company, traders need to submit their own transaction information that later can be verified to prove that certain transactions are completed by themselves. Therefore, the above data needs to be real, effective and verifiable.

1.2. Contribution

We propose a transaction framework in which a transaction organization with the traders transacts on the public chain of blockchain and managers manage its traders with access control based on cryptography.

- The proposed system transacts on public chain of blockchain directly, and utilizes one-time addresses to protect transactions from unauthorized tracking.
- Through access control, the proposed system realizes that the authorized managers can track transactions within their authorities.
- The proposed system creatively uses the hash-based one-time signature twice to verify and track the transactions.
- Compared with the standard Bitcoin transaction system, the proposed system achieves privacy-enhancing and post-quantum security in the semi-honest model.

2. RELATED WORK

Hash-based signature and lattice-based encryption are two of the key post-quantum cryptosystems which resist to both classical and quantum attacks and attract more and more concern in current cryptography research. The former one, hash-based signature, is used to authenticate message and identity of the entity, and designed bycombining one-time signature [8, 9] or few-time signature (FTS) [10,11] with public key authentication hash tree, its security is only based on the security assumptions of the underlying hash function, rather than that of number theory. Most of the existing one-time signature schemes can achieve EUCMA [12]. Typical schemes of limited number and stateful hash-based signature are Merkle hash-based signature, XMSS, Leighton- LMS, etc.; typical Schemes of unlimited number and stateful hash-based signature are GMSS, $XMSS^{MT}$, HSS, etc.; typical Schemes of unlimited number and stateless hash-based signature are SPHINCS, SPHINCS+, Gravity-SPHINCS, etc [13]. The latter one, lattice-based encryption is considered to be secure under the assumption that certain well-studied computational lattice problems, with typical scheme of GGH encryption scheme based on the closest vector problem [14], NTRUEncrypt with shortest vector problem [15], Gentry's fully homomorphic encryption scheme[16], Brakerski-Gentry-Vaikuntanathan fully homomorphic encryption[17], etc.

A great deal of recent research into blockchain has focused on anonymity and other privacy protection issues, which could be classified as solutions implemented in different layers of the blockchain system. In the network layer, clustering based on the behavior patterns of the nodes has been studied extensively to position the malicious node in blockchain network. In application layer, the encryption device cold wallet, software Tor, etc., encrypt data through cryptographic approach[18, 19]. More attention has focused on the security technology in transaction layer. Coin mixing is a typical technology with centralized and decentralized models to prevent transactions from being tracked. In [20], a trustless coin mixing service called MixEth is proposed for Turing-complete blockchains, which implements without relying on a trusted setup. Mixcoin[21], Blindcoin[22], Dash MixEth are the typical centralized Coin mixing services with trusted-third party whereas Coinshuffle and Tumble are decentralized ones[23, 24]. More coin mixing technologies are proposed in application. Other privacy protection technologies based on homomorphic encryption, secure multi-party computation and other cryptographic techniques are discussed to improve the privacy of the application system combined with blockchain[25, 26].

3. PROPOSED SYSTEM

3.1. Objectives of the System

Our goal is to build a blockchain based transaction system, through which traders of a hierarchical transaction organization can transact through the public blockchain. By using the hash of one-time public key of hash-based signature as the transaction address, which is secretly associated with a specific trader, and the relevant one-time private key is only used to sign a unique transaction in the blockchain, the resultant transaction system can be prevented from unauthorized tracking. Meanwhile, authorized managers in the transaction organization can access and track the transaction within their authority by making the trader to utilize the hash-based one-time private key to sign twice, and the proposed system can realize post-quantum security.

3.2. Participators

The participators of the proposed transaction system have hierarchical authorities. According to their different authorities, participators can be divided into the following different identities.

- Managers with authorities at all levels. The authorities of the managers are divided into different levels. Based on the tree-based hierarchy, The managers with upper authority directly manage the ones with authority of one level lower within their authorities, and the managers with bottom-level authorities directly manage the traders participating in the transaction within their authorities. The Authority Table of the proposed system are shown in Figure 1. If a manager or a trader A is in any sub-tree with a manager B as the root node, A is managed within B's authority. For instance, $Trader_{LV3,6}$ and $Trader_{LV3,7}$ are managed within Manager $_{LV2,3}$'s authority, $Trader_{LV3,6}$ to $Trader_{LV3,10}$, Manager $_{LV2,3}$ and Manager $_{LV2,4}$ are all within $Manager_{LV2,3}$'s authority.
- Trader. The trader is the direct participator of the transaction and at the bottom of the authority hierarchy, and his signature and verification keys are used to sign and verify the transactions they participate in.
- Authority Center. Authority Center is a participator trusted by all the participators of the organization. It is responsible for verifying, recording, and managing the transaction information of traders; when managers request the transaction information of traders within their authority, Authority Center verifies the identity and authority of the requested manager, then returns to him the requested transaction information.

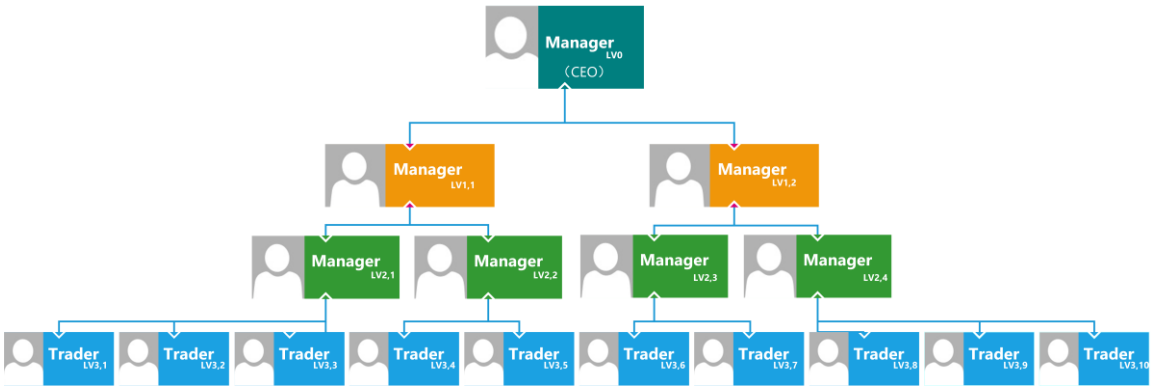

Figure 1. Authority table

3.3. Proposed System

1. FWS-HBS-

KeyGen($1^{\lambda} \rightarrow (SK_{OT,manager_{LV_{i,j},k}}, PK_{OT,manager_{LV_{i,j},k}})$&$(SK_{OT,trader_{i,k}}, PK_{OT,trader_{i,k}})$ &$(SK_{OT,CA_k}, PK_{OT,CA_k} PK_{OT,CA_k})$): The FWS-HBS-KeyGen algorithm is executed by each participator in the system to generate the forward-secure one-time private keys and public keys of hash-based signature. On input the security parameter 1^{λ}, output the one-time private key $SK_{OT,manager_{LV_{i,j},k}}$ and public key $PK_{OT,manager_{LV_{i,j},k}}$ for each manager, where *i*, *j* and *k* represent the authority level from top to bottom, the index in each layer from left to right of the manager in the authority table, and the index of one-time key pair of hash-based signature of the manager, separately; the one-time private key $SK_{OT,trader_{i,k}}$ and public key $PK_{OT,trader_{i,k}}$ for each trader, where *i* and *k* represent the index of the trader from left to right in the bottom layer of the authority table, and the index of one-time key pair of hash-based signature of the trader, separately; the one-time private key SK_{OT,CA_k} and public key PK_{OT,CA_k} for Authority Center, where *k* represents the index of one-time key pair of hash-based signature of Authority Center. Additionally, function *P* is a PRF, function *f* is used to generate one-time public key from one-time private key according to specific key generation schemes of the one-time signature.

Algorithm 1 HBS-KeyGen

Input: security parameter 1^{λ}.
Output: $(SK_{OT,1}, PK_{OT,1})$, …, $(SK_{OT,t}, PK_{OT,t})$.
Participator chooses $SEED_{fws}$ from seed space uniformly at random;
$seed_0 = SEED_{fws}$,
for *i*=1 to t, do
  for *j* =1 to *k*, do
  $seed_j = P_{seed_{j-1}}(0)$
  $sk_j = P_{seed_j}(1)||P_{seed_j}(2)||\cdots$.
  end
$SK_{OT,i}=sk_1||\ldots||sk_k$,
$PK_{OT,i}=f(sk_1)||\ldots||f(sk_k)$.
end
return $(SK_{OT,1}, PK_{OT,1})$, …, $(SK_{OT,t}, PK_{OT,t})$

FWS-HBS-KeyGen is run as shown in Algorithm 1. Without ambiguity, we use

$(SK_{OT,i}, PK_{OT,i})$ to represent the *i*th one-time key pair of the participator for simplicity. Specially, in case of a one-time signature (private) key is leaked, forward security can protect the antecedent signature keys used in the key-evolving signature scheme from being revealed without authority. In the proposed system, each participator utilizes a PRF *P* to iteratively generate seed sequence on inputting a uniform $SEED_{fws}$ chosen from the seed space, and each element $seed_i$ in seed sequence is used to generate each component of a hash-based one-time private key, it has been proven that, the resultant signature scheme can achieve forward security when generating one-time private keys in this way [27].

Typically, the number of the one-time key pairs of each manager is set to power of 2.

2. **HBS-RT-PKeyGen** $((PK_{OT,1}, \ldots, PK_{OT,t}) \rightarrow PK_{RT})$. The HBS-RT-PKeyGen algorithm is executed by the manager in the system and Authority Center to generate their root public key of hash-based signature. On input the one-time public keys of the above participator, output the root public key $PK_{RT,manager_{LV_{i,j}}}$ of each manager, where *i* and *j* represent the authority level from top to bottom and the index in each layer from left to right of the manager in the authority table, separately;the root public key $PK_{RT,CA}$ of Authority Center.

   Each of the above participators establishes his own public key authentication hash tree to compress all his one-time public keys to his root public key by Merkle tree as shown in Figure 2 and Algorithm 2. Without ambiguity, we use $PK_{OT,1}, \ldots, PK_{OT,t}$ and PK_{RT} to represent one-time public keys and root public key of each participator above for simplicity in Algorithm 2. Function *h* is used to compress two children nodes to their parent node on one layer higher.

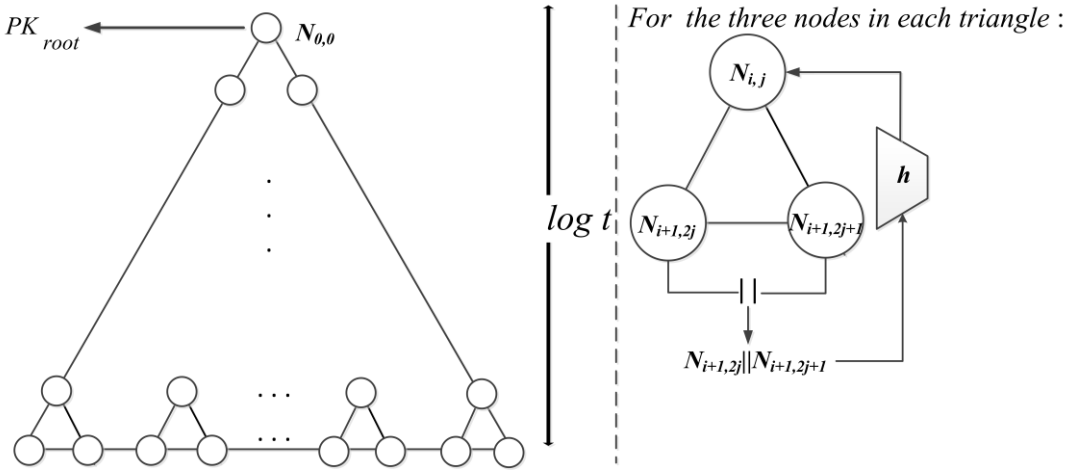

Figure 2. Public key authentication hash tree

Algorithm 2 HBS-RT-PKeyGen

Input: $PK_{OT,1}, \ldots, PK_{OT,t}$.
Output: PK_{RT}.
for i = 0 to t-1, do
$N_{log\,t,i} = h(PK_{OT,i})$.
end
for *i*= log*t* -1 to 0, do
  for *j* = 0 to 2^i-1, do

$$N_{i,j} = h(N_{i+1,2j}, N_{i+1,2j+1})$$

  end
end
Return $N_{0,0}$.

The authorized $PK_{RT,CA}$ is distributed to each participator.

3. **LBE-KeyGen(** $1^{\lambda} \rightarrow$ ($SK_{LBE,manager_{LV_{i.j}}}$, $PK_{LBE,manager_{LV_{i.j}}}$) &($SK_{LBE,trader\ i}$, $PK_{LBE,trader\ i}$)&($SK_{LBE,CA}$, $PK_{LBE,CA}$)): The LBE-KeyGen algorithm is executed by each participator in the system to generate his private key and public key of lattice-based encryption. On input the security parameter 1^{λ} , output the private key $SK_{LBE,manager_{LV_{i.j}}}$ and public key $PK_{LBE,manager_{LV_{i.j}}}$ of lattice-based encryption of each manager, where i and j represent the authority level from top to bottom, the index in each layer from left to right of the manager in the authority table, the private key $SK_{LBE,trader\ i}$ and public key $PK_{LBE,trader\ i}$ of each trader, where i represents the index of the trader in bottom layer from left to right of authority table, the private key $SK_{LBE,CA}$ and public key $PK_{LBE,CA}$ of lattice-based encryption of Authority Center.
The authorized $PK_{LBE,CA}$ is distributed to each participator.

4. **Auth-PKs**
($(ID_{P_i}$, $PROOF_{ID_{P_i}}$, PK_{RT,P_i} , PK_{LBE,P_i} , PK_{OT,P_i-1}, PK_{OT,P_i-1}, $SK_{LBE,CA}$, $PK_{LBE,C}) \rightarrow$ Tab_{PKs}): The Auth-PKs algorithm is executed between managers and their manager of one level higher (if exists), as well as managers (except the one with the authority of the top level) and Authority Center, to authenticate and store the root public key of hash-based signature and the public key of lattice-based encryption of the managers by Authority Center. On input $Manager_i$'s identity $ID_{Manager_i}$ and its proof $PROOF_{ID_{Manager_i}}$, the root public key $PK_{RT,Manager_i}$ of hash-based signature, the public key $PK_{LBE,Manager_i}$ of lattice-based encryption; $Manager_{i-1}$'s one-time private key $SK_{OT,Manager_{i-1}}$ and public key $PK_{OT,Manager_{i-1}}$ of hash-based signature where $Manager_{i-1}$ is the direct manager of manager i; Authority Center's private key $SK_{LBE,CA}$ and public key $PK_{LBE,CA}$ of lattice-based encryption, output $\left(ID_{Manager_i}, PK_{RT,Manager_i}, PK_{LBE,Manager_i}\right)$ as a record for $Manager_i$ in public key record table Tab_{PKS}. Additionally, S_{PKS,P_i} denotes the complete hash-based signature of $(ID_{Manager_i}||PK_{RT,Manager_i}||PK_{LBE,Manager_i})$, $PATH_{AUTH,Manager_i}$ denotes the authentication path from $PK_{OT,Manager_{i-1}}$ to $Manager_{i-1}$'s root public key $PK_{RT,Manager_{i-1}}$ in $Manager_{i-1}$'s public key authentication hash tree; function $S-OTS_a(b)$ is the hash-based one-time signature function to sign b by private key a , $E-LBE_a(b)$ is the lattice-based encryption function to encrypt b by public key a whereas $D-LBE_a(b)$ is the relevant inverse decryption function; assuming $PATH_{AUTH,P_i} = (ph_{logt}, \dots, ph_1)$ from the bottom up, function PH is used to calculate the root value $PK_{RT,Manager_{i-1}}$ of P_{i-1} from the node $PK_{OT,Manager_{i-1}}$ in the bottom of $Manager_{i-1}$'s public key authentication hash tree along the authentication path $PATH_{AUTH,Manager_i}$ in such a way that, $N_{logt} = PK_{OT,Manager_{i-1}}$, For i= logt -1 to 0, $N_i = h(N_{i+1}, ph_{i+1})$. Return N_0.

Algorithm 3 Auth-PKs

Input: $ID_{Manager_i}, PROOF_{ID_{Manager_i}}, PK_{RT,Manager_i}, PK_{LBE,Manager_i}$

Output: record of $\left(ID_{P_i}, PK_{RT,Manager_i}, PK_{LBE,Manager_i}\right)$ in Tab_{PKs}.

1. All the manager $Manager_i$ except the one with top-level authority send $ID_{Manager_i}||PROOF_{ID_{Manager_i}}||PK_{RT,Manager_i}||PK_{LBE,Manager_i}$ to their manager $Manager_{i-1}$ of one lever higher through security channel.

2. $Manager_{i-1}$ verifies whether $PROOF_{ID_{Manager_i}}$ matches with $ID_{Manager_i}$,

if yes, then

$$S_{PKs,Manager_i} = \begin{pmatrix} (S-OTS_{SK_{OT,Manager_{i-1}}}(ID_{Manager_i}||PK_{RT,Manager_{i-1}}||PK_{LBE,Manager_{i-1}}), \\ PK_{OT,P_i-1}, PATH_{AUTH,Manager_i} \end{pmatrix};$$

sends $S_{PKs,Manager_i}$ to $Manager_i$.

else, requests to $Manager_i$ for identity and public keys information and repeats stage 1 to 2.

End

3. $Manager_i$ verifies

$$\begin{pmatrix} S-OTS_{SK_{OT,Manager_{i-1}}}\left(ID_{Manager_i}\left||PK_{RT,Manager_i}\right|\right|PK_{LBE,Manager_i}), \\ PK_{OT,Manager_{i-1}} \end{pmatrix} \stackrel{?}{=} ture$$

if yes, then

$$C_{PKs,Manager_i} = E-LBE_{PK_{LBE,CA}}\begin{pmatrix} ID_{Manager_i}||PK_{RT,Manager_i}|| \\ PK_{LBE,Manager_i}||S_{PKs,Manager_i} \end{pmatrix}$$

sends C_{PKs,P_i} to Authority Center,

else, repeats stage 1 to 3.

end

4. The public keys of the manager with top-level authority are sent to the authoritative center through security channel.
5. Authority Center processes ciphertexts received from all the managers following the order of their authorities from top to bottom:

for each $C_{PKs,Manager_i}$ received from $Manager_i$,

Authority Center computes

$$ID_{Manager_i}||PK_{RT,Manager_i}||PK_{LBE,Manager_i}||S_{PKs,Manager_i} = D-LBE_{SK_{LBE,CA}}(C_{PKs,Manager_i}),$$

verifies

$$\begin{pmatrix} S-OTS_{SK_{OT,Manager_{i-1}}}\left(ID_{Manager_i}\left||PK_{RT,Manager_i}\right|\right|PK_{LBE,Manager_i}), \\ PK_{OT,Manager_{i-1}} \end{pmatrix} \stackrel{?}{=} ture$$

if yes, then

verifies $PH(PK_{OT,Manager_{i-1}}, PATH_{AUTH,Manager_i}) \stackrel{?}{=} PK_{RT,Manager_{i-1}}$;

if yes, then saves a record $(ID_{Manager_i}, PK_{RT,Manager_i}||PK_{LBE,Manager_i})$ in the public key record table Tab_{PKs};

else requests to $Manager_i$ andrepeats stage 3 and 5;

end

else requests to $Manager_i$ andrepeats stage 3 and 5;

end

stores the Tab_{PKs}.

5. Trans. Trans is the transaction algorithm executed by the users of the public chain of blockchain to complete transactions. Transaction mode similar to that of Bitcoin is used by the proposed system, with the only difference that signature in the transaction is realized by utilizing hash-based one-time signature. The structure of a transaction with *m* inputs and *n* outputs is shown as follows.

Input : Payer
Input 0:
Previous tx: the hash value of the previous transaction where the coins come from.

Index: the specific output in the referenced transaction. scriptSig: contains a hash-based one-time signature of this payer over the hash value ofa simplified version of the current transaction and the relevant one-time public key.
…
Input m-1: Previous tx Index scriptSig
Output : Payee
Output 0: Value: transaction value transferred to the following scriptPubKey. scriptPubKey: the hash value of the public key of current payee, which is used as his addressto receive transaction value.
…
Output n-1: Value scriptPubKey
OP_EQUALVERIFYOP_CHECKSIG

The verification (OP_EQUALVERIFYOP_CHECKSIG) is done in the following way. Firstly, compare the hash of one-time public key contained in scriptSig of each input with the scriptPubKey contained in the output with index value Index of the previous transaction Previous tx pointing to, if matches, the identity of this payer is true; else, aborts. Secondly, utilize the one-time public key to verify the hash-based one-time signature for each input, if it is passed, the signature is valid; else, aborts. After the verification, the transaction will be compressed and contained in the public chain as in the Bitcoin.

6. Trans-Re

$\left(ID_{Trader_i}, SK_{OT,trader\ _{i,k}}, PK_{OT,trader\ _{i,k}}, SK_{LBE,CA}, PK_{LBE,CA}\right) \rightarrow \left(ID_{Trader_i}, \left(PK_{OT,trader\ _{i,k}} || Inf_{PK_{OT,trader\ _{i,k}}}\right)\right)$: The Trans-Reg algorithm is executed between the traders and Authority Center to authenticate and register the public key address of the transactions of the traders by Authority Center according to a certain time interval or transaction frequency. On input trader's identity ID_{Trader_i} , his one-time public keys $PK_{OT,trader\ _{i,i_1}}$ to $PK_{OT,trader\ _{i,i_p}}$ used in the current time interval or transaction frequency and the relevant private keys $SK_{OT,trader\ _{i,i_1}}$ to $SK_{OT,trader\ _{i,i_p}}$ of the hash-based signature, Authority Center's private key$SK_{LBE,CA}$ and public key$PK_{LBE,CA}$ of lattice-based encryption, time stamp value $timestamp$, output a record $\left(ID_{Trader_i}, \left(PK_{OT,trader\ _{i,k}} || Inf_{PK_{OT,trader\ _{i,k}}}\right)\right)$ in the transaction regedit table $Tab_{Trans-Reg}$.

Algorithm 4 Trans-Reg
Input: $ID_{Trader_i}, SK_{OT,trader\ _{i,k}}, PK_{OT,trader\ _{i,k}}, SK_{LBE,CA}, PK_{LBE,CA}, timestamp$
Output: record $\left(ID_{Trader_i}, \left(PK_{OT,trader\ _{i,k}} || Inf_{PK_{OT,trader\ _{i,k}}}\right)\right)$ in $Tab_{Trans-Reg}$.
1. for $k = i_1$ to i_p, the trader ID_{Trader_i} do searches $PK_{OT,trader\ _{i,k}}$ in the inputs and $H(PK_{OT,trader\ _{i,k}})$ in the outputs of the transactions in the public chain of blockchain; gets x eligible transactions with txhashs TXH_1 to TXH_x.

end
for $k = i_1$ to i_p, the trader ID_{Trader_i} do
computes
$$S_{PK-evds,Trader_{i,k}} = \begin{pmatrix} (S-OTS_{SK_{OT,Trader_{i,k}}}(ID_{Trader_i}||TXH_1||\dots||TXH_x||timestamp), \\ ID_{Trader_i}, timestamp, PK_{OT,Trader_{i,k}} \end{pmatrix},$$
end
the trader ID_{Trader_i} computes
$C_{PK-evds,trader_i} = E-LBE_{PK_{LBE,CA}}(S_{PK-evds,trader_{i,1}}||\cdots||S_{PK-evds,trader_{i,k}})$,
sends $C_{PK-evds,trader_i}$ to Authority Center.

2. Authority Center computes
$S_{PK-evds,trader_{i,1}}||\cdots||S_{PK-evds,trader_{i,k}} = D-LBE_{SK_{LBE,CA}}(C_{PK-evds,trader_i})$,
for $k = i_1$ to i_p,
searches $PK_{OT,trader_{i,k}}$ in the inputs and $H(PK_{OT,trader_{i,k}})$ in the outputs of the transactions in the public chain of blockchain,
gets x eligible transactions with txhashsTXH_1 to TXH_x.
end
verifies $\begin{pmatrix} (S-OTS_{SK_{OT,Trader_{i,k}}}(ID_{Trader_i}||TXH_1||\dots||TXH_x||timestamp), \\ ID_{Trader_i}, TXH_1, \dots, TXH_x, timestamp, PK_{OT,Trader_{i,k}} \end{pmatrix} \stackrel{?}{=} ture$,
if yes, then saves a record
$\left(ID_{Trader_i}, \left(PK_{OT,trader_{i,i_1}}||Inf_{PK_{OT,trader_{i,i_1}}}\right), \cdots PK_{OT,trader_{i,i_p}}||Inf_{PK_{OT,trader_{i,i_p}}}\right)$
in $Tab_{Trans-Reg}$;
else, inform trader ID_{Trader_i} transaction information of $PK_{OT,trader_{i,k}}$ is invalid.
end

What should be mentioned here is that, unlike in the hash based signature, the private key of one-time signature is only used for signing once. Here, the private keys$SK_{OT,trader_{i,i_1}}$ to $SK_{OT,trader_{i,i_p}}$ have been used to sign transactions once in phase Trans, and they are used here to generate a secondary signature. We give the security reduction in section IV.

7. Trans-track $((ID_{Manager_i}, RQT, timestamp, SK_{LBE,CA}, PK_{LBE,CA}, SK_{OT,Manager_i}, PK_{OT,Manager_i},) \rightarrow Info_{trans-RQST})$. The Trans-track algorithm is executed between the manager and Authority Center based on access control to ensure the manager accessible to the transaction information of traders within their authority. It is run as shown in Algorithm 5.

On input the identity $ID_{Manager_i}$ of the manager i who requests to get access to the transaction information, his one-time private key $SK_{OT,Manager_i}$ and public key $PK_{OT,Manager_i}$ of hash-based signature, the request RQT, Authority Center's private key$SK_{LBE,CA}$ and public key$PK_{LBE,CA}$ of lattice-based encryption, Authority Center's hash-based one-time private key$SK_{OT,CA}$ and public key$PK_{OT,CA}$, time stamp value $timestamp$, Authority Center replies to the manager i with the transaction information $Info_{trans-RQST}$he requests after verification. Additionally, $S_{RQT,Manager_i}$ denotes the complete hash-based signature of $(ID_{Manager_i}||RQT||timestamp)$, $PATH_{AUTH,Manager_i}$ denotes the authentication path from $PK_{OT,Manager_i}$ to the root public key $PK_{RT,Manager_i}$ of hash-based signature in the public key authentication hash tree of manager i.

Algorithm 5 Trans-Track

Input: $ID_{Manager_i}$, $SK_{OT,Manager_i}$,$PK_{OT,Manager_i}$, $SK_{LBE,CA}$, $PK_{LBE,CA}$,RQT, $timestamp$

Output: $Info_{trans-RQST}$.

1. The manager $ID_{Manager_i}$ computes

$$S_{RQT,Manager_i} = \begin{pmatrix} (S-OTS_{SK_{OT,Manager_i}}(ID_{Manager_i}||RQT||timestamp), \\ ID_{Manager_i}||RQT||timestamp, PK_{OT,Manager_i}, PATH_{AUTH,Manager_i} \end{pmatrix}.$$

If confidentiality of request is needed, then

the manager $ID_{Manager_i}$ computes

$$C_{RQST,Manager_i} = E-LBE_{PK_{LBE,CA}}(S_{RQST,Manager_i}).$$

end

sends $S_{RQST,Manager_i}$ or $C_{RQST,Manager_i}$ to Authority Center.

2. Authority Centercomputes

$C_{RQST,Manager_i} = D-LBE_{SK_{LBE,CA}}(C_{RQST,Manager_i})$,

verifies RQT matches with the access authority of $ID_{Manager_i}$;

if yes, then

verifies $\begin{pmatrix} (S-OTS_{SK_{OT,Manager_i}}(ID_{Manager_i}||RQT||timestamp), \\ ID_{Manager_i}||RQT||timestamp, PK_{OT,Manager_i} \end{pmatrix} \stackrel{?}{=} ture,$

if yes, verifies $PH(PK_{OT,Manager_i}, PATH_{AUTH,Manager_i}) \stackrel{?}{=} PK_{RT,Manager_i}$;

if yes, computes

$$S_{Info_{trans-RQST},Manager_i} = \begin{pmatrix} (S-OTS_{SK_{OT,CA}}(Info_{trans-RQST}), \\ Info_{trans-RQST}, PK_{OT,CA}, PATH_{AUTH,CA} \end{pmatrix}$$

$C_{Info_{trans-RQST},Manager_i} = E-LBE_{PK_{LBE,Manager_i}}(S_{Info_{trans-RQST},Manager_i})$;

sents$C_{Info_{trans-RQST},Manager_i}$ to manager $ID_{Manager_i}$;

else abort.

end

else abort.

end

3. The manager $ID_{Manager_i}$ computes

$S_{Info_{trans-RQST},Manager_i} = D-LBE_{SK_{LBE,Manager_i}}(S_{Info_{trans-RQST},Manager_i})$

verifies $(S-OTS_{SK_{OT,CA}}(Info_{trans-RQST}), Info_{trans-RQST}, PK_{OT,CA}) \stackrel{?}{=} ture,$

if yes, verifies $PH(PK_{OT,CA}, PATH_{AUTH,CA}) \stackrel{?}{=} PK_{RT,CA}$

if yes, obtains the authenticated$Info_{trans-RQST}$;

else abort.

end

else abort.

end

4. SECURITY

Theorem.*The proposed system is post-quantum forward-secure,EUCMA and IND-CPA secure in thesemi-honest model if*

P is a post-quantum PRF;

h is a post-quantum one-way hash function;

the underlying hash-based one-time signature is post-quantum EUCMA (*with query time set to one*);

the underlying lattice-based encryption is post-quantum IND-CPA.

5. CONCLUSIONS

We propose a transaction framework by which a transaction organization with multiple traders transacting on the public blockchain realizes access control based on cryptography. Compared with the classical blockchain which protects the privacy of traders through the anonymity of the transaction address, the proposed system essentially prevents the transaction from unauthorized tracking to enhance privacy preservation by using a one-time transaction address;furthermore, the proposed system creatively utilizes hash-based one-time private key to generate two signatures, one for authentication of transaction, the other for transaction tracking to realizes access control; the proposed system enhances privacy and achieves post-quantum security in the semi-honest model. Compared with the classical cryptographic tools used in blockchain, the limitations of usage of hash-based signature technology are the increase in the key size, and the consequent additional cost of key management. In the following work, we will consider the ways to reduce the size of the private key, and the possible applications of other post-quantum cryptographic technologies in the blockchain.

ACKNOWLEDGEMENTS

This work is supported by the National Natural Science Foundation of China (No. 61972391) and the Open Project Program of State Key Laboratory of Cryptology (MMKFKT201810).

REFERENCES

[1] Sukhodolskiy and S. Zapechnikov. “A blockchain-based access control system for cloud storage, ” 2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus). IEEE, 2018, pp.1575-1578.

[2] W. She, Q. Liu, et al. “Blockchain trust model for malicious node detection in wireless sensor networks. ” IEEE Access. vol.7, pp. 38947-38956, 2019.

[3] M. Almaiah. “A new scheme for detecting malicious attacks in wireless sensor networks based on blockchain technology,” Artificial Intelligence and Blockchain for Future Cybersecurity Applications. Springer, Cham, 2021, pp.217-234.

[4] R. Henry, A. Herzberg and A. Kate. “Blockchain access privacy: Challenges and directions, ” IEEE Security & Privacy vol.16.4, p.p. 38-45, 2018.

[5] A. H. Mohsin, A. A. Zaidan, B. B. Zaidan, , et al. “PSO–Blockchain-based image steganography: towards a new method to secure updating and sharing COVID-19 data in decentralised hospitals intelligence architecture, ” Multimedia tools and applications vol.80.9, pp. 14137-14161, 2021.

[6] Y. Rahulamathavan, Phan, M. Rajarajan, et al. “Privacy-preserving blockchain based IoT ecosystem using attribute-based encryption, ” 2017 IEEE International Conference on Advanced Networks and Telecommunications Systems. IEEE, 2017, pp. 1-6.

[7] W. Fang, W. Cheng, W. Zhang , et al. “Digital signature scheme for information non-repudiation in blockchain: a state of the art review, ” EURASIP Journal on Wireless Communications and Networking vol.1, pp.1-15, 2020.

[8] R. C. Merkle, Secrecy, “Authentication and Public Key Systems, ” PhD thesis, Stanford, 1979.

[9] D. Naor, A. Shenhav, and A. Wool, “One-time signatures revisited: Have they become practical? ” IACR Cryptol. ePrint Arch., 2005.

[10] A. Perrig, “The BiBa one-time signature and broadcast authentication protocol,” Proceedings of the 8th ACM Conference on Computer and Communications Security. 2001, pp.28-37.

[11] L. Reyzin and N. Reyzin, “Better than BiBa: Short one-time signatures with fast signing and verifying,” Australasian Conference on Information Security and Privacy, Springer, 2002, pp. 144-153.

[12] S. Halevi, and H. Krawczyk. “Strengthening Digital Signatures Via Randomized Hashing,” CRYPTO 06, 2006, pp. 41-59.

[13] L. Li, X. Lu, and K. Wang. “Hash-based signature revisited,” Cybersecurity vol.5.1, pp.1-26, 2022.

[14] S.Garg, C. Gentry, and S. Halevi. “Candidate multilinear maps from ideal lattices,” Annual International Conference on the Theory and Applications of Cryptographic Techniques. Springer, Berlin, Heidelberg, 2013, pp.1-17.

[15] J. Hoffstein, N Howgrave-Graham, J Pipher, et al. “Practical lattice-based cryptography: NTRUEncrypt and NTRUSign,” The LLL Algorithm. Springer, Berlin, Heidelberg, pp.349-390, 2009.

[16] C. Gentry. “Fully homomorphic encryption using ideal lattices,” Proceedings of the forty-first annual ACM symposium on Theory of computing. 2009, pp.169-178.

[17] Z. Brakerski, C. Gentry, and V. Vaikuntanathan. "(Leveled) fully homomorphic encryption without bootstrapping,” ACM Transactions on Computation Theory vol.6.3, pp.1-36, 2014

[18] Bentov and R. Kumaresan, “How to use Bitcoin to design fair protocols,” Lecture Notes in Computer Science, vol. 8617, pp. 421–439, 2017.

[19] Mck. Khalilov and A. Levi. “A survey on anonymity and privacy in bitcoin-like digital cash systems,” IEEE Communications Surveys & Tutorials vol.20.3 2543-2585, 2018.

[20] I.A. Seres, D.A. Nagy, C. Buckland, et al. “Mixeth: efficient, trustless coin mixing service for ethereum,” Cryptology ePrint Archive, 2019.

[21] Bonneau, A. Narayanan, A. Miller et al., “Mixcoin: Anonymity for bitcoin with accountable mixes,” International Conference on Financial Cryptography and Data Security. Springer, Berlin, Heidelberg, 2014, pp.486-504.

[22] Valenta and B. Rowan, “Blindcoin: Blinded, accountable mixes for bitcoin,” in Financial Cryptography and Data Security. Heidelberg, Germany: Springer, 2015, pp. 112–126.

[23] T. Ruffifing, P. Moreno-Sanchez, and A. Kate, “CoinShufflfle: Practical decentralized coin mixing for Bitcoin,” in Computer Security—ESORICS 2014. Cham, Switzerland: Springer, 2014, pp. 345–364.

[24] E. Heilman, L. Alshenibr, F. Baldimtsi, A. Scafuro, and S. Goldberg, “TumbleBit: An untrusted bitcoin-compatible anonymous payment hub,” IACR Cryptol. ePrint Archive, Rep. 2016/575, Jun. 2016.

[25] Singh, Parminder, et al. "Blockchain and homomorphic encryption-based privacy-preserving data aggregation model in smart grid." Computers & Electrical Engineering 93 (2021): 107209.

[26] Alper, Handan Kılınç, and AlptekinKüpçü. "Optimally efficient multi-party fair exchange and fair secure multi-party computation." ACM Transactions on Privacy and Security 25.1 (2021): 1-34.

[27] H. Krawczyk. “Simple forward-secure signatures from any signature scheme,” Proceedings of the 7th ACM Conference on Computer and Communications Security. 2000, pp. 108-115.

AUTHORS

Lingyun Li received the B.S. and M.S. degrees in information security from Shandong University, China. She is currently a Ph.D. student in Institute of Information Engineering, Chinese Academy of Sciences. Her research interests include privacy preservation, post-quantum cryptography, and network security.

Xianhui Lu Received his Ph.D. in information security from Southwest Jiaotong University. He is currently a Full Professor and Ph.D. Supervisor with the Institute of Information Engineering, Chinese Academy of Sciences, China. His research interests include provable Security, post-quantum cryptography and homomorphic encryption.

Kunpeng Wang received his Ph.D. from Tsinghua University. He is currently a Professor and Ph.D. Supervisor with the Institute of Information Engineering, Chinese Academy of Sciences, China. His research interests include post-quantum cryptography and homomorphic encryption.