Received: Added at production | Revised: Added at production | Accepted: Added at production

DOI: xxx/xxxx

ARTICLE TYPE

Privacy-aware PKI Model with Strong Forward Security

Fengyin Li[1] | **Zhongxing Liu**[1] | **Tao Li**[1] | **Hongwei Ju**[2] | **Hua Wang**[1] | **Huiyu Zhou***[3]

[1]School of Information Science and Engineering, Qufu Normal University, Rizhao, China
[2]Experimental Teaching and Equipments Management Center, Qufu Normal University, Rizhao, China
[3]School of Informatics, University of Leicester, Leicester, United Kingdom

Correspondence
*Huiyu Zhou, University of Leicester. Email: hz143@leicester.ac.uk

Summary

With the development of network technology, privacy protection and users anonymity become a new research hotspot. The existing blockchain privacy-aware PKI (Public Key Infrastructure) model can ensure the privacy of users in the authentication process to a certain extent, but there are still problems of the storage and leakage of users' keys. This paper first proposes a strong forward-secure ring signature scheme based on RSA, which ensures the anonymity of the signing users and the forward-backward security of the keys. Then, by introducing the ring signature technology into the privacy-aware PKI model, this paper proposes a privacy-aware PKI model with strong forward security based on block chains, which not only ensures the users' identity privacy, but also solves the problem of the storage and leakage of the users' keys, greatly improving the success rate and security of the users' identity authentication. Finally, this paper applies the proposed PKI model to anonymous transactions, designs a privacy-aware anonymous transaction model with strong forward security, realizing anonymous transactions without relying on trusted third parties, and implementing users' privacy protection.

KEYWORDS:
blockchain, RSA, ring signature, anonymous transactions, PKI

1 | INTRODUCTION

With the development of network technology, the world has entered the information age, Internet of Things, social Internet of Things and other network products emerge in endlessly in people's vision, and the research on these technologies has greatly promoted the development of human society[1,2,3,4]. And these products all depend on the underlying Public Key Infrastructure(PKI). Traditional PKI is an identity authentication method relies on a centralized certificate Authority(CA). Users send identity information to CA, then CA verifies their identities. If the authentication is successful, the user is issued with an identity certificate. So, the user can perform various activities requiring positive identity. Although this method is simple and fast, it has the problem of centralized CA[5,6]. Once the CA is compromised, the users' identity information will be disclosed. Even if CA has no vulnerability and user s' identity information is secret, there is no guarantee that the CA will be honest. Until the development of blockchain technology bring light to the improvement of identity authentication mechanism[7].

Blockchain is a decentralized accounting method, it has immutability, anonymity and other features. Therefore, the blockchain has spawned a large number of applications[8,9,10,11,12,13], such as solving the problem of users' identity authentication in the field of information security. By applying the blockchain technology to the identity authentication mechanism, we can get the blockchain-based PKI. Compared with the traditional PKI, it can remove the centralized mechanism like CA and realize the decentralized identity authentication mechanism, namely, the identity information is published to the blockchain.

However, due to the transparent and immutability of the blockchain, with the development of the blockchain, some people found that the real identity of users could be discovered by tracking transactions on the chain. Therefore, the identity privacy disclosure problem appears in the blockchain-based PKI, which is a fatal blow to the blockchain-based PKI. To solve this problem, Louise Axon et al. proposed a blockchain-based privacy-aware PKI model in 2017[14], which can realize the concealment of users' real identities through the iteration of public and private keys. At

the same time, the use of pseudonym mechanism makes transaction traceability only obtain the pseudonym of the user, but not the real identity of the user, which greatly improves the anonymity of users' authentication. But with the further study, Olamide Omolola et al. found that common modulus attack and other security problems still existed in PKI models with privacy perception [15]. Whereupon, the ring signature technology is introduced into the privacy-awareness PKI, and the identity key is signed and authenticated through the authentication of the registered key, which ensures the correctness and security of the identity key published in the blockchain. Because the signature key used for its ring signature is the registration key, there is a security problem of the registration key. Once the registration key is disclosed, the security of the current signature and user identity will be threatened. This paper introduces a strong forward-secure ring signature based on forward-secure ring signature [16] and applies it into privacy-aware PKI to solve the security problems of registration key and users' authentication. Strong forward security here refers to forward security and backward security. Forward security means that even if an adversary obtains the users' key for the current phase, it cannot calculate the key for the previous phase. Similarly, backward security means that even if an adversary obtains the users' key for the current stage, it cannot calculate the key for the later stage.

The main contributions of this thesis are as follows:

(1) This paper presents a strong forward-secure ring signature algorithm based on RSA. And this algorithm not only guarantees the anonymity of the signing user, but also realizes the forward and backward security of the key.

(2) By introducing the proposed ring signature algorithm into the PKI mechanism, this paper designs a privacy-aware PKI model with strong forward security based on blockchain. The model guarantees the correctness of key iteration and strong forward security.

(3) Based on the privacy-aware PKI model with strong forward security proposed in this paper, an anonymous transaction model is designed in this paper. The model realizes the anonymity and privacy protection in the user transaction process.

In this paper, section 2 introduces the basic concepts; section 3 introduces the strong forward-secure ring signature based on RSA; section 4 introduces the detailed content and security analysis of our PKI model; section 5 describes the presentation of applying the model to anonymous transactions; section 6 summarizes this paper.

2 | PRELIMINARIES

2.1 | Blockchain

As Satoshi Nakamoto came up with the concept of Bitcoin, the blockchain technology developed rapidly. As the core technology of Bitcoin, blockchain is a decentralized accounting system. All users share the same "account book", so it has immutability and unforgeability. As a form of bookkeeping, blockchain naturally has staff involved in bookkeeping, which we call miners, and the behavior of miners to book data blocks is called mining. The blockchain rewards miners for doing the right thing. Blockchain works correctly as long as most miners are honest. Each block of the blockchain contains the hash value of the previous block, block is connected in turn to form a chain structure. All users can trace the data in the chain structure. Blockchain can be divided into public chain, private chain and alliance chain according to the restrictions on participants. Public chain is a blockchain that everyone can participate in, so it also has the function of anonymity. The users' identity can use the public key as the account to participate in transactions, and a user can create multiple accounts, which greatly improves the security of anonymity. The private chain is a blockchain that can only participate after authentication has passed. It does not need to guarantee anonymity and is suitable for small-scale bookkeeping. And the alliance chain is formed by combining different groups as participants, which is applicable to the accounting method between companies.

2.2 | PKI

Public Key Infrastructure (PKI) is a system that provides public key encryption and digital signature. By identifying and verifying the users' identity information correctly, the PKI issues a certificate to the user, which binds the key and the users' identity. The traditional public key infrastructure is a system that relies on a certificate authority (CA). Users send identity information to CA, then CA verify and authenticate identities. If the authentication is successful, the user is issued with an identity certificate. So, the user can perform various activities requiring positive identity. Although this method is simple and fast, it has the problem of CA centralization. Once the CA vulnerability is stolen, the users' identity information will be disclosed. Even if users' identity information is secret, there is no guarantee that the CA will be honest. Hence the emergence of decentralized authentication, namely public key infrastructure based on blockchain. In this way, the centralized CA is replaced by blockchain. The users' identity information is verified by most people, and once passed, it will be put into the blockchain by miners. Due to the immutability and unforgeability characteristics of the blockchain, the security and reliability of identity authentication can be guaranteed.

2.3 | Ring Signature

Ring signature was first proposed in 2001 by Rivest, Shamir and Tauman [17], and is named for the fact that it can form a ring structure according to certain rules. The signature verifier can determine that the signer is from a member of the ring, but not the identity of the real signer. Unlike group signatures, ring signatures can choose any member as a possible signer. There is no group building process. It doesn't need administrators of group and it can guarantee the unconditional anonymity of the identity of the signer. These characteristics make the ring signature have a wide range of applications in electronic cash, electronic voting, ADHOC anonymous authentication fields. The forward security ring signature adds the forward security on the basis of the ring signature, which divides the signature cycle into several time periods. The private key is iteratively updated in stages, while the public key remains unchanged, so that the adversary cannot forge the previous ring signature even if he obtains the users' current private key.

2.4 | Privacy-aware PKI Model

Privacy-aware in blockchain-based PKI is an improvement on Certcoin [18] by Louise Axon. On the basis of the framework of Certcoin, the privacy awareness part was added, that is, the identity of registered user was hidden through the iteration of public and private keys in the key update process, and the identity leakage problem was further solved through the authentication of pseudonyms to ensure that the real identity *id* of user would not be disclosed in the key update process. The process is also divided into two phases: the registration phase and the update phase. The most important difference between PB-PKI and Certcoin is in the key update phase. PB-PKI needs to update the online key iteratively with the newly generated pair of offline keys. The iteration process is shown in Figure 1 below.

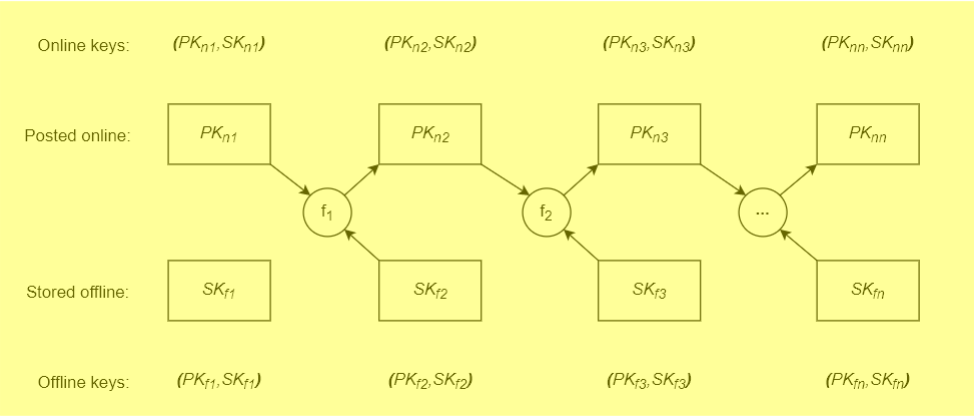

FIGURE 1 Key update phase for PB-PKI

In this process, the function "f" is the key evolution function: $pk_{n_n} = pk_{n_{n-1}} \times sk_{f_n} \bmod N$. According to this function, PB-PKI can obtain a new pair of online keys: (pk_{n_n}, sk_{n_n}). At the same time, the relationship between online public and private keys which satisfying the formula $pk_{n_n} \times sk_{n_n} \equiv 1 \bmod N$ is guaranteed. Finally, the user will publish the updated publicly available information to the blockchain to authenticate the pseudonym.

3 | STRONG FORWARD-SECURE RING SIGNATURE BASED ON RSA

Based on literature [16], this paper proposes a new strong forward-secure ring signature scheme based on RSA, which is introduced in det ail in this section:

(1) Key generation: According to the key generation algorithm in RSA encryption algorithm, user $U_i (i = 1, \cdots, n)$ randomly selects two large prime numbers p_i, q_i and calculates $N_i = p_i \cdot q_i; \varphi(N_i) = (p_i - 1) \cdot (q_i - 1)$. Then U_i selects two larger integers e_i and y_i which co-prime with $\varphi(N_i)$ as the public keys. Finally U_i calculates the inverse elements d_i and x_i of the public keys according to the formula: $e_i \cdot d_i \equiv 1 \bmod \varphi(N_i), y_i \cdot x_i \equiv 1 \bmod \varphi(N_i)$ as the private keys. In this way, a pair of public and private keys $(d_i, e_i), (x_i, y_i)$ are obtained. Then U_i chooses an anti-collision hash function $H_i : \{0,1\}^* \rightarrow Z_{N_i}$ and forms its own signature public and private key pair: $SK = (d_i, x_i, p_i, q_i), PK = (e_i, y_i, N_i, H_i)$.

(2) Key update: The validity period of the signature key is divided into T periods (T is known by all members as the system parameter). Then, in the j($1 \le j \le T$) time period, $SK_i = (d_{i,j}, x_{i,j})$ is the user U_i in the j time period private key. When the time enters in the j-th time period, we get the equation 1 and 2 :

$$d_{i,j} = (d_{i,j-1})^2 mod\ \varphi(N_i) = d_i^{2^j}\ mod\ \varphi(N_i) \tag{1}$$

$$x_{i,j} = (x_{i,j+1})^2 mod\ \varphi(N_i) = x_i^{2^{T-j}}\ mod\ \varphi(N_i) \tag{2}$$

(3) Signature: Let the public key set of the ring members selected by the user U_k to be signed be $L = \{(e_1, y_1, N_1, H_1), \cdots, (e_n, y_n, N_n, H_n)\}$. Then in the j-th time period, U_k randomly selects $r_k, r_k^{'} \in Z_{N_k}^*$ and calculates $c_{k+1} = H_{k+1}(L, m, r_k, r_k^{'})$, where H is the anti-collision hash function $H_i : \{0,1\}^* \to Z_{N_i}$.

For $i = k+1, k+2, \cdots, n-1, n, 1, 2, \cdots, k-1$, U_k selects the random numbers $s_k, s_k^{'} \in Z_{N_k}^*$, and then calculates the equation 3 :

$$c_{i+1} = H_{i+1}(L, m, c_i + s_i^{(e_i)^{2^j}}\ mod\ N_i, c_i + s_i^{'(e_i)^{2^{T-j}}}\ mod\ N_i) \tag{3}$$

Notice: If $i = n$, let $c_{i+1} = c_1, H_{i+1} = H_1$.

Finally, U_k calculates s_k and $s_k^{'}$: $s_k = (r_k - c_k)^{d_{k,j}} mod\ N_k, s_k^{'} = (r_k^{'} - c_k)^{x_{k,j}} mod\ N_k$, and output the signature: $\sigma_m = (j, c_1, s_1, \cdots, s_n, s_1^{'}, \cdots, s_n^{'})$.

(4) Verification: After the verifier receives the signature σ_m, the message m and the set L of public keys, for $i = 1, 2, \cdots, n$, in turn, calculates the equation 4 :

$$r_i = c_i + s_i^{(e_i)^{2^j}}\ mod\ N_i, r_i^{'} = c_i + s_i^{'(e_i)^{2^{T-j}}}\ mod\ N_i \tag{4}$$

Then the verifier calculates $c_{i+1} = H_{i+1}(L, m, r_i, r_i^{'})$ and finally verifies whether $c_1 = H_1(L, m, r_n, r_n^{'})$ holds. If it is established, the signature is correct. Otherwise, the signature is invalid.

(5) Correctness: According to the key update formula 1 and 2 , we get the equation 5 and 6 :

$$r_k = c_k + s_k^{(e_k)^{2^j}}\ mod\ N_k = c_k + ((r_k - c_k)^{d_{k,j}})^{e_k^{2^j}}\ mod\ N_k = r_k \tag{5}$$

$$r_k^{'} = c_k + s_k^{'(y_k)^{2^{T-j}}}\ mod\ N_k = c_k + ((r_k^{'} - c_k)^{x_{k,j}})^{y_k^{2^{T-j}}}\ mod\ N_k = r_k^{'} \tag{6}$$

(6) Strong forward security: Forward security is achieved by the iterative update of the private key of the first part of the user U_i: $d_{i,j} = (d_{i,j-1})^2 mod\ \varphi(N_i) = d_i^{2^j} mod\ \varphi(N_i)$. And backward security is achieved by the update of the private key of the second part of the user U_i: $x_{i,j} = (x_{i,j+1})^2 mod\ \varphi(N_i) = x_i^{2^{T-j}} mod\ \varphi(N_i)$.

Among them,$\varphi(N_i)$ is obtained by calculating two large prime numbers p_i, q_i randomly selected by the user U_i through the calculation of $\varphi(N_i) = (p_i - 1) \cdot (q_i - 1)$. According to the intractability of "finding the square root in the case of a large composite number" , it is difficult to obtain the previous private key from the latter private key.

(7) Security: We prove the security of the proposed strong forward-secure ring signature by the following theorem.

Theorem. If the factorization of large numbers is difficult, the strong forward-secure ring signature based on RSA proposed in this paper is provably secure in the EU-CMA security model.

Proof. We construct a simulator B, and then B interacts with a probabilistic polynomial time (PPT) adversary A in the following steps. A can make q_H inquiries to hash oracles $H_i (i = 1, \ldots, n)$ and q_S inquiries to signature oracle (SO) at most. If A can forge a valid signature in time τ with a non-negligible probability $\epsilon > 1/Q(k)$ on the message *m* and a set *L* of public keys, and the signature satisfies $V(m, L, \sigma) = 1$ (where *Q* is a polynomial function and *k* is a sufficiently large security parameter),

$$Pr(A(L) \to (m, \sigma) : V(m, L, \sigma) = 1) > 1/Q(k) \tag{7}$$

there is a PPT algorithm, which can solve the large numbers factorization problem with a non-negligible probability $\frac{1}{n(q_H + q_S)Q(k)}$ within time τ. This is contradictory to the difficult problem of large numbers factorization. Therefore, the hypothesis is not true, that is, it is impossible to have a PPT adversary A, who can forge an effective signature in time τ with a non-negligible probability $\epsilon > 1/Q(k)$.

Setup. Let $L^{'} = \{pk_1, pk_2, \cdots, pk_n\} = \{(e_1, y_1, N_1, H_1), (e_2, y_2, N_2, H_2), \cdots, (e_n, y_n, N_n, H_n)\}$. A is a PPT adversary and can make q_H inquiries to hash oracles $H_i (i = 1, \ldots, n)$ and q_S inquiries to signature oracle (SO) at most. In addition to repeated queries, the independent random oracles H_i output random results, SO can also query the oracles H_i and be consistent with the query output of A.

Query. At this stage, the adversary A can make hash queries and signature queries to the oracles H_i and SO, respectively. Simulator B prepares a hash list and a signature list to record all queries and responses as follows, where both lists are empty at the beginning. When the adversary uses L_i, m_j as input for a hash query, B first checks to see if the query record exists in the hash list and, if so, returns the hash value $H(L_i, m_j, z_i, z_i^{'})$ of the query directly. Otherwise, according to the input L_i, m_j, B selects the random numbers $s_i, s_i^{'}$ and calculates $z_i = c_i + s_i^{(e_i)^{2^j}} \bmod N_i$ and $z_i^{'} = c_i + s_i^{'(y_i)^{2^{T-j}}} \bmod N_i$. According to the input and calculation results, B calculates its hash value $H(L_i, m_j, z_i, z_i^{'})$ stored in the hash list and returns the value to A.

Similarly, when the adversary uses L_i, m_j as input for a signature query, B first checks whether there is a record of the query in the signature list, and if there is a record, it directly returns the signature $\sigma_{m_j}(L_i) = (c_1, s_1, \cdots, s_n, s_1^{'}, \cdots, s_n^{'})$ of the query. Otherwise, B calculates the signature $\sigma_{m_j}(L_i)$ according to the input L_i, m_j and returns it to A, and stores the signature in the signature list.

Forgery. By running adversary A, simulator B is able to simulate a random oracle and get a consistent answer with each hash function H_i and signature oracle SO. For any message *m*, any public key set $L \subseteq L^{'}$, B simulates SO, does not use any private key, and only controls hash function *H* to generate a valid signature as follows:

(1) B randomly selects $i \in \{1, 2, \cdots, n\}, c_i \in Z_{N_i}^*$;

(2) For $i = k, k+1, \cdots, n, 1, \cdots, k-1$, Brandomly selects $s_i, s_i^{'} \in Z_{N_i}^*$, and calculates $z_i = c_i + s_i^{(e_i)^{2^j}} \bmod N_i$, $z_i^{'} = c_i + s_i^{'(y_i)^{2^{T-j}}} \bmod N_i$, $c_{i+1} = H_{i+1}(L, m, z_i, z_i^{'})(i \neq k-1)$;

(3) Let $H_k(L, m, z_{k-1}, z_{k-1}^{'}) = c_k$;

(4) Output the signature $(c_1, s_1, \cdots, s_n, s_1^{'}, \cdots, s_n^{'}))$.

Notice: Let $c_{i+1} = c_1, H_{i+1} = H_1$ when $i = n$. SO returns the signature for the U_k as the actual signer.

A returns a forged signature with a probability not less than $\frac{1}{Q(k)} - \frac{1}{q - q_H - q_S}$ for *n* random oracles used for validation. Where the *q* is the number of possible responses of all the oracles. Since $\frac{1}{q - q_H - q_S}$ is negligible, A returns a forged signature with a probability not less than $1/Q(k)$ for all random oracles used for verification. Therefore, when A forges a valid signature, it must ask n queries to hash oracles H_i that are consistent with the verification equation. So, Let's call these n queries $X_{i_1}, X_{i_2}, \cdots, X_{i_n}, 1 \leq i_1 < \cdots < i_n$.

Let $X_{i_1}, X_{i_2}, \cdots, X_{i_n}$ be the n queries that satisfy the verification for the first time, and let *k* satisfy formula 8.

$$X_{i_k} \rightarrow H_k(L, m, c_{k-1} + s_{k-1}^{(e_{k-1})^{2^j}} mod\ N_{k-1}, c_{k-1} + s_{k-1}^{'(y_{k-1})^{2^{T-j}}} mod\ N_{k-1}) \tag{8}$$

That is, X_{i_k} corresponds to the query to H_k in validation, and *k* is called the gap of ring signature σ. If $i_1 = \ell$, the forged signature σ is denoted as (ℓ, k)-σ. In other words, the first query associated with all verifications is the ℓ query and the gap is *k*.

At the beginning of the simulation, B selects a pair (ℓ, k), then B can ensure that (ℓ, k) satisfies a successfully forged signature (ℓ, k)-σ with a probability not less than $\frac{1}{n(q_H + q_S)Q(k)}$ and receive $X_{i_k} \rightarrow H_k(L, m, z_{k-1}, z_{k-1}^{'})$, $X_{i_{k+1}} \rightarrow H_{k+1}(L, m, z_k, z_k^{'})$.

When the query X_{i_k} occurs (and the query $X_{i_{k+1}}$ has also occurred), B returns c_k as the value of $H_k(L, m, z_{k-1}, z_{k-1}^{'})$. Where the $X_{i_{k+1}}$ satisfies the formula 9.

$$X_{i_{k+1}} \rightarrow H_{k+1}(L, m, c_k + s_k^{(e_k)^{2^j}} mod\ N_k, c_k + s_k^{'(y_k)^{2^{T-j}}} mod\ N_k) \rightarrow H_{k+1}(L, m, z_k, z_k^{'}) \tag{9}$$

At this point, since $c_k, z_k, z_k^{'}$ are known, if A can successfully forge the ring signature, the s_k and $s_k^{'}$ satisfying the relational $z_k = c_k + s_k^{(e_k)^{2^j}} \bmod N_k$ and $z_k^{'} = c_k + s_k^{'(y_k)^{2^{T-j}}} \bmod N_k$ will be output. That is, A can get the s_k and $s_k^{'}$ from $s_k^{(e_k)^{2^j}} \bmod N_k$ and $s_k^{'(y_k)^{2^{T-j}}} \bmod N_k$ without knowing the private key, which is contradictory to the difficult problem of large numbers factorization. So, the signature cannot be forged.

This completes the proof of the theorem. □

4 | PRIVACY-AWARE PKI MODEL WITH STRONG FORWARD SECURITY

4.1 | Model Architecture

In this paper, the privacy-aware model proposed in literature [15] is adopted as the model framework. During the key update process, a new pair of online keys is generated each time, rather than by iteratively updating the online keys using the offline key. In identity authentication, the newly generated online key is authenticated through the ring signature of the users' registration key released in the blockchain to ensure the correctness and security of the users' online key. The frame structure is shown in Figure 2.

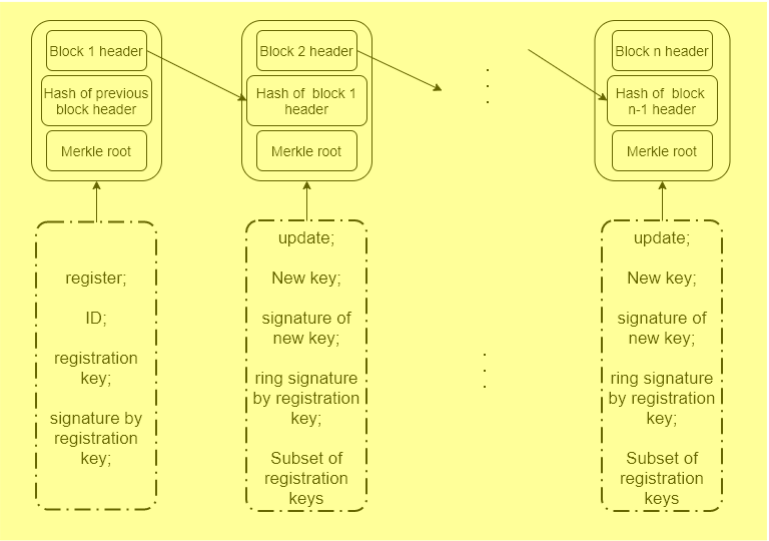

FIGURE 2 The model frame structure

4.2 | A Privacy-aware PKI Model with Strong Forward Security

This paper introduces the strong forward-secure ring signature into the PKI model. Based on the RSA algorithm, the privacy-aware PKI model with strong forward security is proposed to update the users' registration key, which solves the security hidden danger of the registration key loss and ensures the forward security of the registration key. The model of this paper is divided into two parts: registration stage and key update stage. The model of this paper is divided into two parts: the registration stage and the key update stage. The registration process authenticates the real identity of users through the block chain structure, while the update process takes place after the registration process. At the end of the registration process, the registered user carries on the key update operation to generate the new pseudonym and identity key, and uses the ring signature technology to prove himself as the valid registered user, so as to realize the purpose of hiding his real identity and achieving anonymity. In the following sections we will introduce the two parts in detail.

4.2.1 | Registration

The registration stage is mainly to bind the real identity *id* of the user to the registration key obtained by the RSA key generation algorithm, and publish it to the blockchain to complete the registration of users' identity, the specific registration process is shown in Figure 3 below.

At the registration stage, the user first generates the registration key and master key according to the RSA key generation algorithm. The registration key is bound to the users' identity, and then participates in the users' key update process as the signature key. The function of the key pair is to prove his identity when an adversary impersonates himself. The specific algorithms for generating registration key are as follows:

(1) The user randomly selects two large prime numbers p_0 and q_0, and calculates $N_0 = p_0 \cdot q_0$. Then, the user calculates $\varphi(N_0) = (p_0-1)\cdot(q_0-1)$ according to Euler formula.

(2) The user selects two large integers pk_0 and $pk_0^{'}$ that are mutually prime with $\varphi(N_0)$ as the public key.

(3) Then the user calculates the inverse elements sk_0 and $sk_0^{'}$ of the public keys as the private keys according to the formula $pk_0 \cdot sk_0 \equiv 1 \bmod \varphi(N_0), pk_0^{'} \cdot sk_0^{'} \equiv 1 \bmod \varphi(N_0)$, thus a set of registered key pairs is obtained: $PK_0 = (pk_0, pk_0^{'}), SK_0 = (sk_0, sk_0^{'})$.

Then the master key pair is generated: similarly, p_m and q_m are taken, and $N_m = p_m \cdot q_m, \varphi(N_m) = (p_m - 1) \cdot (q_m - 1)$ is calculated. Then according to the formual $mpk \cdot msk \equiv 1 \bmod \varphi(N_m)$, we can get the master key pair (mpk, msk).

At this point, after the two pairs of keys required for user registration are generated, the user needs to sign the identity(*id*) according to the RSA signature scheme to get $\sigma_0, \sigma_0^{'}$ and σ_m: $\sigma_0 = id^{sk_0} \bmod N_0, \sigma_0^{'} = id^{sk_0^{'}} \bmod N_0, \sigma_m = id^{msk} \bmod N_m$.

By this time, all the information needed for user registration is ready. Next, the user needs to package this information to get a standard format packet: $(id, register, T_0, PK_0 = (pk_0, pk_0^{'}), \sigma_0, \sigma_0^{'}, \sigma_m)$ and publishes it to the blockchain. Where *id* represents the real identity of the user, register represents this operation as a user registration operation, T_0 represents the current timestamp, PK_0 represents the generated registration public key, σ_0 and $\sigma_0^{'}$ represent the signature of the registration private key to the identity, and the intention is to associate

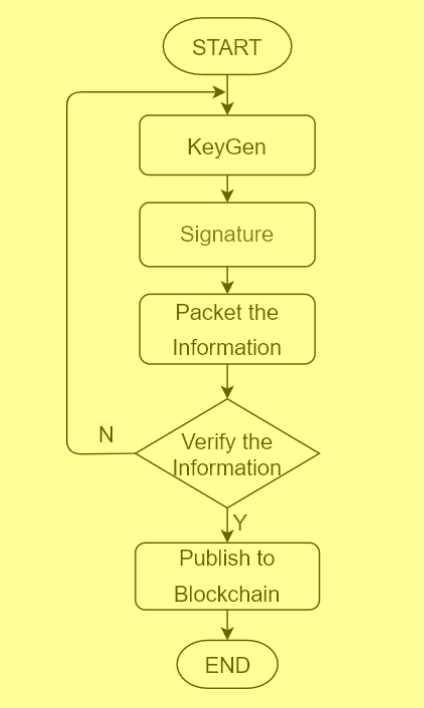

FIGURE 3 Flow chart of registration

the registered public key with the identity, the last σ_m indicates the signature of the master private key to the identity, and the intention is that if someone impersonates the users' identity in the future, the user can prove his identity by providing the master key and its signature. The rest of the information includes that the current registered private key $SK_0 = (sk_0, sk_0^{'})$ and the master key pair (mpk, msk) are stored locally.

At the end of the registration phase, the members of the blockchain verify the information published by the user. First, they will verify whether the registered *id* and registered public key $pk_0, pk_0^{'}$ are registered for the first time, and then verify the correctness of the signature according to the RSA signature algorithm. That is, equation 10 and 11 :

$$id = \sigma_0^{pk_0} mod\ N_0 = (id^{sk_0})^{pk_0} mod\ N_0 = id^{sk_0 \cdot pk_0} mod\ N_0 \tag{10}$$

$$id = \sigma_0^{'pk_0^{'}} mod\ N_0 = (id^{sk_0^{'}})^{pk_0^{'}} mod\ N_0 = id^{sk_0^{'} \cdot pk_0^{'}} mod\ N_0 \tag{11}$$

If the signature is correct, the user publishes a block containing this information into the blockchain.

4.2.2 | Key Update

The key update stage is the focus of this model, in which we hide the real identity of the user through the evolution of user's public keys. After the registration process, in order to hide their real identity, registered users will perform the key update process to generate new pseudonyms and keys, and prove themselves as registered users through ring signature technology, while hiding the relationship between pseudonyms and real *id*. The specific process is shown in Figure 4 below.

In the update phase, based on the RSA key generation algorithm, registered user first generates a new key pair: $PK_n = (pk_n, pk_n^{'}), SK_n = (sk_n, sk_n^{'})$. And then the user publishes the ring signature of the registration key: $PK_0 = (pk_0, pk_0^{'}), SK_0 = (sk_0, sk_0^{'})$, so as to prove that the user's identity is indeed a registered member.

According to the flow chart, we can describe the process of key update in detail as below:

(1) Key iteration: The user generates a random number $R_n \in \{0,1\}^*$ as the pseudonym of user's identity at this stage. Next, according to the RSA key generation algorithm, a new identity key pair: $PK_n = (pk_n, pk_n^{'}), SK_n = (sk_n, sk_n^{'})$ is generated. Then, the strong forward-secure ring signature scheme proposed in Section 3 is used to perform the ring signature operation on pseudonym R_n, and the signature σ_{r_n} is obtained:

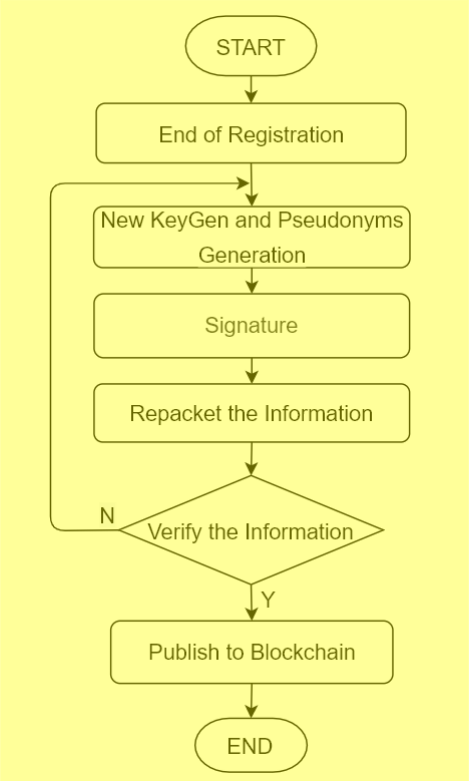

FIGURE 4 Flow chart of keys update phase

a) Key generation: Assume that the user is $U_k (1 \leq k \leq n)$. Set the registration key for user U_k as $PK_{0,k} = (pk_{0,k}, pk'_{0,k})$, $SK_{0,k} = (sk_{0,k}, sk'_{0,k})$, where the module used is $N_{0,k}$.

b) Key update: The registration key's validity period of user U_k is divided into *T* time periods, and when the time enters the j-th time period, U_k calculates the equation 12 and equation 13.

$$sk_{0,k,j} = (sk_{0,k,j-1})^2 mod\ \varphi(N_{0,k}) = sk_{0,k}^{2^j} mod\ \varphi(N_{0,k}) \tag{12}$$

$$sk'_{0,k,j} = (sk'_{0,k,j+1})^2 mod\ \varphi(N_{0,k}) = (sk'_{0,k})^{2^{T-j}} mod\ \varphi(N_{0,k}) \tag{13}$$

c) Signature: Let the public key set of the ring members selected by the user U_k be $L = \{(pk_{0,1}, pk'_{0,1}, N_{0,1}, H_1), \cdots, (pk_{0,n}, pk'_{0,n}, N_{0,n}, H_n)\}$. Then in the j-th time period, U_k randomly selects $r_k, r'_k \in Z^*_{N_{0,k}}$ and calculates $c_{k+1} = H_{k+1}(L, R_n, r_k, r'_k)$, where *H* is the anti-collision hash function $H_k : \{0,1\}^* \to Z_{N_{0,k}}$.

For $i = k+1, k+2, \cdots, n-1, n, 1, 2, \cdots, k-1$, U_k selects the random numbers $s_i, s'_i \in Z^*_{N_{0,k}}$, and then calculates the equation 14.

$$c_{i+1} = H_{i+1}(L, R_n, c_i + s_i^{(pk_{0,i})^{2^j}} mod\ N_{0,i}, c_i + s'_i{}^{(pk'_{0,i})^{2^{T-j}}} mod\ N_{0,i}) \tag{14}$$

When $i = n$,let $c_{i+1} = c_1, H_{i+1} = H_1$.

Finally, U_k calculates s_k and s'_k: $s_k = (r_k - c_k)^{sk_{0,k,j}} \bmod N_{0,k}$, $s'_k = (r'_k - c_k)^{sk'_{0,k,j}} \bmod N_{0,k}$, and then outputs the signature: $\sigma_{r_n} = (j, c_1, s_1, \cdots, s_n, s'_1, \cdots, s'_n)$.

(2) Information collation: The user collates all the corresponding information and packages them into the following format: $(update, T_n, PK_n = (pk_n, pk'_n), R_n, \sigma_{r_n})$. Where *update* indicates that the current operation is an update operation, T_n indicates the current timestamp, and R_n is the pseudonym generated this time.

(3) Information verification and release: Finally, the verifier of blockchain verifies that the updated key has not been registered before, and uses a verification algorithm of strong forward-secure ring signature to verify the signature and ensure the validness of the signature. After the verification, the blockchain members will publish the packets sorted out in step (2) to the blockchain.

4.3 | Security Analysis

The security of the strong forward-secure PKI model proposed in this paper is mainly analyzed into two aspects: one is the security of RSA scheme itself, and the other is the security of the model.

4.3.1 | RSA Security

The security of RSA algorithm depends on the principle of large number decomposition, that is, given the value of an integer, and that it is the product of two prime numbers and requires the calculation of the value of two prime numbers. When n is large enough, factorization of large integers is computationally difficult, and there is no general effective algorithm. The key point for cryptanalysts to attack RSA systems is how to break down n. If the decomposition succeeds in making $n = p \cdot q$, then $\varphi(n) = (p-1) \cdot (q-1)$ can be calculated, and then the private key d can be obtained from the public key e. So, if p and q are large enough prime numbers for RSA Security, the analyst cannot decompose n in the effective time (polynomial time). Of course, the theory has not proved that the RSA algorithm must need large number factorization, and there may be some decoding algorithms without large number factorization. In this case, we will not go deep into this problem. The more complex and secure the module we select by default.

4.3.2 | Model Security

The security of the privacy-aware PKI model with strong forward security proposed in this paper is mainly in two aspects: one is to ensure the security of master key and identity key generated by the users; the other is the security of the signature scheme used in the model. The security of the ring signature scheme used in this paper has been demonstrated in section 3, which focuses on the security of master key and identity key generated by the users. The security objective of the model in this paper is to completely distinguish the real identity of the user from the updated key, that is, the updated key cannot be traced back to the real identity of the user. For this security goal, we analyze the following key security situations.

(1) When only the master key is lost, the adversary can only analyze the user's *id* by using the master key, but will not associate the *id* with the pseudonym the user is using.

(2) When only the identity key is lost, the user can update the identity key in time, or revoke the identity key directly by issuing the master key, which can prevent the adversary from committing illegal acts by posing as himself.

(3) When the master key and identity key are lost at the same time, the adversary obtains the user's complete identity information and identity proof credentials, and the user can only revoke his/her identity with the master key and then re-register.

5 | THE PRIVACY-AWARE ANONYMOUS TRADING MODEL WITH STRONG FORWARD SECURITY

Now that there is an anonymous public key infrastructure, we can then apply it to anonymous transactions[19]. This paper divides anonymous transactions into two forms, one is the direct connection between the two parties to the transaction, through their respective anonymity to ensure the anonymity of the transaction; the second is to set up an anonymous transaction model: such as joining a third party: proxy[20].

For the first form, the PKI model based on blockchain design adds identity anonymity that can be controlled by users and can be directly applied to anonymous transactions. Using the public key and pseudonym generated by the model itself to participate in the transaction, can prevent the enemy from tracing back to the real identity of the user, plus the characteristics of the blockchain itself to the center, to trust, can completely guarantee the anonymity of the transaction.

For the second form, we use the form of adding proxy to further enhance transaction anonymity.

5.1 | Anonymous Trading Architecture

The role of proxy in the anonymous transaction model can cut off the relationship between the two parties and further enhance the anonymity of the transaction.

Based on the anonymous transaction model [16], the third party: proxy "I" is added between "A" and "B" to form the framework of the transaction model. See Figure 5 below.

Through this simple model, we can use proxy "I" to confuse transactions users A and B, destroy the connection between A and B, so that user A and B transactions will not be traced back, and greatly improve the anonymity and security of transactions.

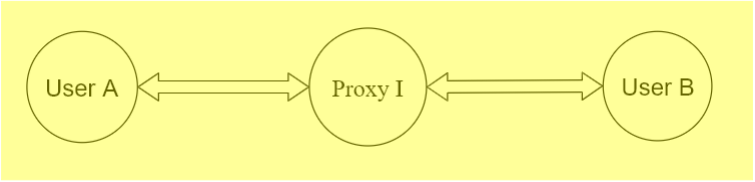

FIGURE 5 Anonymous transaction framework

5.2 | Privacy-aware Anonymous Trading Model

For the anonymous transaction model, the trader and the proxy use the key generated by the PKI model as the transaction account and set up four transaction algorithms: $T_{offer}(EG \rightarrow V)$, $T_{offer}(V \rightarrow EG)$, $T_{fulfill}(EG \rightarrow V)$ and $T_{fulfill}(V \rightarrow EG)$.

(1) $T_{offer}(EG \rightarrow V)$: Algorithm for exchange transaction voucher V using Electronic Currency (E-Gold) EG;

(2) $T_{offer}(V \rightarrow EG)$: Algorithm of Using transaction vouchers V exchange electronic currency EG transaction;

(3) $T_{fulfill}(EG \rightarrow V)$: To the electronic currency EG exchange transaction vouchers V the transaction confirmation;

(4) $T_{fulfill}(V \rightarrow EG)$: To the transaction vouchers V exchange electronic currency EG transaction confirmation.

Through the confirmation mechanism of the blockchain itself, the four transaction algorithms are sorted and published, that is, the proxy "I" first adds the transaction voucher V to the blockchain to exchange the electronic currency EG transaction $T_{offer}(V \rightarrow EG)$ through the release block; Then the transaction party A to add transaction $T_{offer}(EG \rightarrow V)$ of electronic currency EG exchange voucher V to the blockchain by issuing blocks ;finally, transactions $T_{fulfill}(EG \rightarrow V)$, $T_{fulfill}(V \rightarrow EG)$ are issued simultaneously by the counterparty A and the proxy "I" in a block, for confirmation of the first two transactions. The specific process is shown in Figure 6 below. Thus, the confirmation mechanism of the blockchain

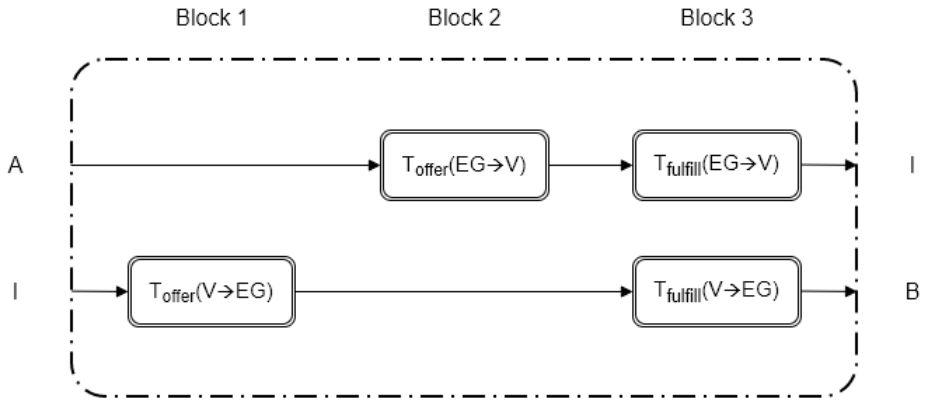

FIGURE 6 Process of transaction execution

itself can ensure that the transaction party A I the transaction currency to the proxy, and the proxy "I" carry out the transaction simultaneously with the transaction currency to the transaction party, so as to prevent the transaction from going wrong. At the same time, a transaction party A can be set up to trade a part of the currency to the proxy "I" to encourage the proxy "I" to perform the operation of anonymous transactions. In this way, under the premise of the anonymity of its own account, the anonymous mechanism of proxy is added, and the anonymity of the transaction is greatly improved by double guarantee.

5.3 | Performance Analysis of Privacy-aware Anonymous Trading Model

The proposed privacy-aware anonymous transaction model consists of two parts: the PKI model and the anonymous transaction model. Firstly, the PKI model ensures that only the user's public and private keys and pseudonyms are used in the whole transaction model, and the user's real identity

is not connected, which greatly improves the anonymity of both parties. Secondly, the proxy mechanism of the anonymous transaction model only use the proxy as the communication hub of the two parties, which can cut off the relationship between the two parties of the transaction, further improving the anonymity of the transaction. The security of the trading model is guaranteed by the four functions set in the model. The four functions are published in three blocks: first, the currency exchange voucher transaction block initiated by user A is published to proxy "I" : $T_{offer}(EG \rightarrow V)$, and then the block in which proxy I publishes the voucher exchange currency transaction $T_{offer}(V \rightarrow EG)$ to user B. Finally, two trades:$T_{fulfill}(EG \rightarrow V)$, $T_{fulfill}(V \rightarrow EG)$ are published simultaneously in the third block to confirm the two previously published trades. In this way, even if the proxy is malicious, the proposed model can ensure the correct transaction, greatly improving the security of the anonymous transactions.

6 | CONCLUSION

Firstly, this paper proposes a strong forward security loop signature scheme based on RSA, which not only guarantees the anonymity of the user, but also ensures the forward and backward security of the signed user key. then, by introducing the ring signature technology into the privacy-aware PKI model, this paper proposes a forward security privacy-aware PKI model based on blockchain. while ensuring user identity privacy, it solves the problem of user key storage and disclosure, and greatly improves the success rate and security of user identity authentication. Finally, this paper applies forward security privacy perception PKI model to anonymous transactions, and designs a privacy perception anonymous transaction model, which realizes anonymous transactions without relying on trusted third parties and protects the privacy of user.

ACKNOWLEGEMENT

This work was partly funded by EU Horizon 2020 DOMINOES Project (Grant Number: 771066) and CERNET Innovation Project(NGII20181201).

References

1. Jiang N, Chen J, Zhou R, et al. PAN: Pipeline assisted neural networks model for data-to-text generation in social internet of things. *Inf. Sci.* 2020; 530: 167–179. doi: 10.1016/j.ins.2020.03.080
2. Jiang N, Xu D, Zhou J, Yan H, Wan T, Zheng J. Toward optimal participant decisions with voting-based incentive model for crowd sensing. *Inf. Sci.* 2020; 512: 1–17. doi: 10.1016/j.ins.2019.09.068
3. Chen X, Li C, Wang D, et al. Android HIV: A Study of Repackaging Malware for Evading Machine-Learning Detection. *IEEE Transactions on Information Forensics and Security* 2020; 15: 987–1001.
4. Sun N, Zhang J, Rimba P, Gao S, Zhang LY, Xiang Y. Data-Driven Cybersecurity Incident Prediction: A Survey. *IEEE Communications Surveys and Tutorials* 2019; 21(2): 1744–1772.
5. Ellison C, Schneier B. Ten risks of PKI: What you're not being told about Public Key Infrastructure. *Computer Security Journal* 2000; 16. doi: 10.1201/9780203498156.ch23
6. Brands SA. Rethinking Public Key Infrastructures and Digital Certificates: Building in Privacy. 2000.
7. Nakamoto S. Bitcoin: A Peer-to-Peer Electronic Cash System. *Cryptography Mailing list at https://metzdowd.com* 2009.
8. Meiklejohn S, Orlandi C. Privacy-Enhancing Overlays in Bitcoin. 2015: 127-141. doi: 10.1007/978-3-662-48051-9_10
9. Miers I, Garman C, Green M, Rubin A. Zerocoin: Anonymous Distributed E-Cash from Bitcoin. *Proceedings - IEEE Symposium on Security and Privacy* 2013: 397-411. doi: 10.1109/SP.2013.34
10. LIN J, Primicerio K, Squartini T, Decker C, Tessone C. Lightning Network: a second path towards centralisation of the Bitcoin economy. *New Journal of Physics* 2020. doi: 10.1088/1367-2630/aba062
11. Ron D, Shamir A. Quantitative Analysis of the Full Bitcoin Transaction Graph. 2012. doi: 10.1007/978-3-642-39884-1_2

12. Ruffing T, Moreno-Sanchez P, Kate A. CoinShuffle: Practical Decentralized Coin Mixing for Bitcoin. 2014; 8713. doi: 10.1007/978-3-319-11212-1_20

13. Singla A, Bertino E. Blockchain-Based PKI Solutions for IoT. 2018: 9-15. doi: 10.1109/CIC.2018.00-45

14. Axon L, Goldsmith M. PB-PKI: A Privacy-aware Blockchain-based PKI. 2017: 311-318. doi: 10.5220/0006419203110318

15. Plessing P, Omolola O. Revisiting Privacy-aware Blockchain Public Key Infrastructure. 2020: 415-423. doi: 10.5220/0008947104150423

16. Wansu B, Yun W. A Forward Secure Ring Signature Scheme. 2008: 215-218. doi: 10.1109/IIH-MSP.2008.106

17. Rivest R, Shamir A, Tauman Y. How to Leak a Secret. *LNCS* 2001; 2248: 552-565. doi: 10.1007/3-540-45682-1_32

18. Miers I, Garman C, Green M, Rubin A. Zerocoin: Anonymous Distributed E-Cash from Bitcoin. *Proceedings - IEEE Symposium on Security and Privacy* 2013: 397-411. doi: 10.1109/SP.2013.34

19. Meiklejohn S, Pomarole M, Jordan G, et al. A Fistful of Bitcoins: Characterizing Payments among Men with No Names. *Communications of the Acm* 2016; 59(4): 86-93.

20. Heilman E, Baldimtsi F, Goldberg S. Blindly Signed Contracts: Anonymous On-Blockchain and Off-Blockchain Bitcoin Transactions. 2016; 9604: 43-60. doi: 10.1007/978-3-662-53357-4_4