Hindawi
Security and Communication Networks
Volume 2021, Article ID 7266564, 14 pages
https://doi.org/10.1155/2021/7266564

Research Article

Linkable Ring Signature Scheme Using Biometric Cryptosystem and NIZK and Its Application

Xuechun Mao, Lin You, Chengtang Cao, Gengran Hu, and Liqin Hu

College of Cybersecurity, Hangzhou Dianzi University, Hangzhou, China

Correspondence should be addressed to Lin You; mryoulin@gmail.com

Received 16 April 2021; Revised 16 September 2021; Accepted 27 October 2021; Published 15 December 2021

Academic Editor: David W. Chadwick

Biometric encryption, especially based on fingerprint, plays an important role in privacy protection and identity authentication. In this paper, we construct a privacy-preserving linkable ring signature scheme. In our scheme, we utilize a fuzzy symmetric encryption scheme called symmetric keyring encryption (SKE) to hide the secret key and use non-interactive zero-knowledge (NIZK) protocol to ensure that we do not leak any information about the message. Unlike the blind signature, we use NIZK protocol to cancel the interaction between the signer (the prover) and the verifier. The security proof shows that our scheme is secure under the random oracle model. Finally, we implement it on a personal computer and analyze the performance of the constructed scheme in practical terms. Based on the constructed scheme and demo, we give an anonymous cryptocurrency transaction model as well as mobile demonstration.

1. Introduction

With the advantages of decentralized control and anonymous payment, cryptocurrency is gradually replacing the traditional payment mode. However, the anonymity provided by bitcoin has been questioned in the sense that it offers pseudonymity instead of real anonymity. The research work [1] has shown that attackers can improperly obtain the actual identity of a bitcoin's owner or even other users through proxy addresses. In order to improve anonymity, researchers have proposed various privacy protection schemes, such as Dash based on the mixed coins protocol, Monero based on the CryptoNote protocol, and Zerocoin [2] based on the Zero-Knowledge Succinct Non-interactive Arguments of Knowledge (zk-SNARK) protocol [3], etc.

In 2015, Noether [4] improved Monero's original CryptoNote protocol by using a variant of linkable ring signature, which was called Ring Confidential Transactions (Ring CT). In the Ring CT protocol, Noether improved the "one-time ring signature" to linkable ring signature as the core cryptoprimitive to provide anonymity, which could not only meet the actual transaction needs, but also prevent the occurrence of double spending in transactions. Meanwhile, Monero also used stealth address to hide the recipient's identity. Sasson et al. [2] proposed Zerocash, which used zk-SNARK protocol to construct the anonymous electronic cash system to protect the privacy of users and transaction amounts.

Along with the development of cryptocurrency applications, privacy protection has gradually become an important issue. How to authenticate users while ensuring their anonymity has always been an important challenge in the information age. Biometric encryption technology which combines the cryptographic schemes with biometrics is an important branch of biometric protection technology. It is designed to protect secrets by binding/retrieving secrets with biometrics rather than using passwords or tokens in conventional cryptographic systems. Compared with passwords or tokens, biometrics such as fingerprints are more convenient, stable, and unforgettable. Nowadays, many biometric encryption algorithms have been proposed [5, 6]. In a word, combining biometric encryption technology and

NIZK protocol with ring signatures will provide a great potential advantage for the protection of users' privacy.

1.1. Related Work

Linkable Ring Signature Schemes. In 2004, linkability property was first introduced in a ring signature scheme by Liu et al. [7]. Later, Franklin and Zhang [8] proposed a general framework for linkable ring signatures. Nowadays, there were many variants of linkable ring signatures based on different features. Deng et al. [9] presented a new identity-based linkable ring signature scheme which avoided certificate management. Sun et al. [10] formalized the syntax of Ring CT protocol and then put forward a new efficient Ring CT protocol (Ring CT 2.0) which could save significant space.

Non-Interactive Zero-Knowledge (NIZK) Protocol. In 1988, Blum et al. [11] firstly studied the NIZK proof system and presented the common reference string model which is generally applied to Zerocash. This construction is a NIZK range proof system based on a number theoretic assumption related to factoring. Using the Fiat-Shamir heuristic, Groth [12] suggested a NIZK argument for correctness of an approval vote. In order to optimize the size of NIZK proofs, Gentry et al. [13] constructed a fully homomorphic hybrid encryption scheme to minimize the communication cost. In 2019, Tsai et al. [14] proposed a new non-interactive ZKRP scheme to maintain high flexible range form.

Biometric Encryption Schemes. In 1994, combining with fingerprint recognition, Tomoko firstly proposed the concept of biometric encryption and applied it for patents. Since then, various biometric cryptographic algorithms have been proposed. Juels and Sudan [15] put forward the concept of fuzzy vault whose security is based on the hardness of the polynomial reconstruction. However, Osadchy and Dunkelman [16] found that many of the existing schemes do not consider the privacy and security aspects of the feature extraction and binarization processes which have a huge risk for user privacy. Therefore, Lai et al. [5] proposed a novel biometric cryptosystem for vectorial biometrics called symmetric keyring encryption (SKE) by using an index of maximum hashed vectors, simple filtering mechanism, and Shamir's secret-sharing scheme. They also formalized and analyzed the threat model for SKE, which involved four major security attacks.

1.2. Contributions.

In this work, we firstly use a simplistic biometric secret-binding scheme called SKE to encrypt the secret key which can protect user's secret key and authenticating user's identity at the same time. Second, we utilize a NIZK protocol to provide anonymity for the message. Unlike the blind signatures [17, 18] which have similar property, no interaction is required during the signing and validation process of our scheme. The security analysis shows that our proposed scheme is provably secure under the random oracle model. Third, we analyze the performance of the proposed scheme and also implement it based on the fingerprint model. The encouraging results indicate that our scheme is practicable. Finally, we propose an anonymous cryptocurrency transaction model with a corresponding mobile demo.

1.3. Structure of the Paper.

The rest of this paper is organized as follows. In Section 2, some notations are introduced, and SKE and NIZK protocols are described. System framework and security model are presented in Section 3. We describe the signature scheme in Section 3. And its security analysis is provided in Section 5. In Section 6, some experimental results are given. The anonymous cryptocurrency transaction model is given in Section 7. Finally, the last section gives the conclusion.

2. Preliminary

First, we give some notations in Table 1 which are used in the rest of this paper.

2.1. SKE Model.

The SKE model is a novel simplistic biometric secret-binding scheme for vectorial biometrics which is based on the notion of symmetric key cryptosystems [5]. First, the SKE model uses IoM hashing [19] which can generate abundant IoM hashed entries as genuine entries without being restricted by the original biometric vector size. We use $x \in \mathbb{Z}_q^d$ as an enrolled biometric vector and N as a random projection matrix. Given x, m, N, the IoM hashing operations as follows: $\Omega^{\mathrm{IoM}}(x, m, N) \longrightarrow \phi_x \in \mathbb{Z}_q^m$.

During enrollment, there is a user with an enrolled biometric vector x, parameter m, two different nonces $N, \widehat{N}$, and a finite field polynomial $f(\cdot) \in \mathcal{F}_q^m$ of order to generate $\phi_{\mathbf{x}}$ and $\widehat{\varphi}_x$. Then, we perform polynomial projection to generate $f(\phi_x)$ and yield a public secure sketch $ss = (\widehat{\varphi}_{x(1)} \oplus f(\varphi_{x(1)}), \ldots, \widehat{\varphi}_{x(m)} \oplus f(\varphi_{x(m)}))$. Finally, we generate authentication tag $\mathrm{TAG} = (\mathrm{tag}_1, \ldots, \mathrm{tag}_m)$, where $\mathrm{tag}_i = H(\varphi_{x(i)} \| ss_i \| \widehat{\varphi}_{x(i)})$ is the one-way hashed output. We store $(N, \widehat{N}, \mathrm{TAG}, ss)$ as the public helper data.

During secret retrieval, given the query biometric vector x' and the public helper data above, we generate $\phi_{x'}$ and $\widehat{\varphi}_{x'}$ as well as tag_i' and $\mathrm{TAG}\prime$ following the steps above.

When $\mathrm{tag}_i' = \mathrm{tag}_i$, we can get a genuine pair $(\varphi_{x(i)}, f(\varphi_{x(i)})) \in G$, where G is a genuine set. If we have a sufficient number of revealed genuine pairs with $|U| = t \geq k$, the secret key y can be retrieved via polynomial interpolation using the unlocking set $U = \left\{(\varphi_{\mathbf{x}(i)}, f(\varphi_{\mathbf{x}(i)}))\right\}_{i=1}^{t}$. A high-level overview of SKE is shown in Figure 1.

2.2. NIZK Protocol.

NIZK protocols can be used to demonstrate the truth of a statement without revealing anything else. We briefly state the NIZK protocol [12] which we will use below. First, we set the system parameters $g, h \in \mathbb{G}$. We also denote $H_1(\cdot)$, $H_2(\cdot)$ to be secure hash functions and individually output $\mu, e \in \mathbb{Z}_q$. Randomizer R and message are prover's input. Then, we simply discuss this NIZK as follows.

Table 1: Notations.

Symbol	Representation
$\mathbb{G}, \mathbb{G}_T$	Cyclic group or prime of prime order q
$\mathbb{Z}_q$	Ring of integers modulo q
$\mathbb{Z}_q^*$	$\mathbb{Z}_q/\{0\}$
$\mathbb{G}^n, \mathbb{Z}_q^n$	Vector spaces of dimension n over $\mathbb{G}$ and Z_q
e	Efficiently computable nondegenerate pairing $\mathbb{G} \times \mathbb{G} \longrightarrow \mathbb{G}_T$
$\mathbb{R}^d$	Continuous domain
$[N]$	$\{1, \ldots, N\}$
$\varepsilon(l)$	The negligible function
$\widehat{\pi}$	Omitting the index π
Y	Random projection matrix composed of q Gaussian row vectors
N	Random projection matrix $(Y_1, \ldots, Y_m)$
$\varphi_{x(i)}$	$\arg\max_i Y_i$ where $i \in 0, 1, \ldots, q$
H_1, H_2, H_3	A hash function returning a value in $\mathbb{Z}_q$
H_p	A map-to-point hash function

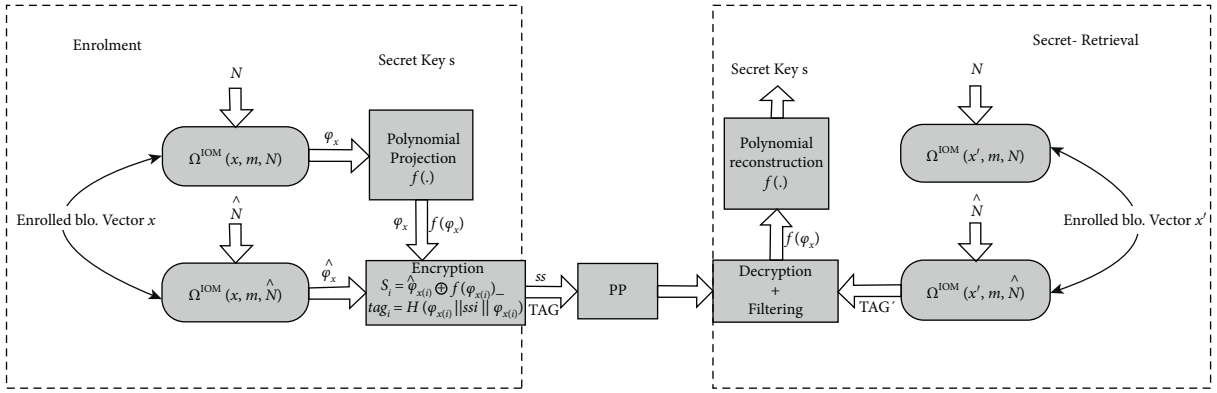

Figure 1: The process of SKE.

(i) Argument:

(1) Compute $H_1(\text{message}) = \mu$ and $C = \mu g + Rh \in \mathbb{G}$.
(2) Randomly choose $R_\mu, R_R \in \mathbb{Z}_q^*$; compute $C_R = R_\mu g + R_R h$ and $e = H_2(C, C_R)$.
(3) Set $\mu' = e\mu + R_\mu$, $R' = eR + R_R$.
(4) The argument is (C_R, μ', R').

(ii) Verifier:

(1) Compute e as above.
(2) Verify whether $eC + C_R = \mu' g + R' h$.

In this instance, it is easy to see that prover can convince the verifier that he/she knows message without revealing knowledge message and interacting with the verifier.

NIZK proofs are usually use in the common reference string (CRS) model, where in a string of a special structure is generated in a setup phase and made available for everyone to prove/verify statements.

Definition 1 (NIZK Argument [20]). *A NIZK argument for an NP relation R consists of a triple of polynomial time algorithms* (Setup, Prove, Verify) *defined as follows:*

(1) $\text{Setup}(1^\kappa)$ takes a security parameter κ and outputs a CRS Σ.

(2) $\text{Prove}(\Sigma, s, w)$ takes as input the CRS Σ, a statement s, and a witness w and outputs an argument π.

(3) $\text{Verify}(\Sigma, s, \pi)$ takes as input the CRS Σ, a statement s, and a proof π and outputs either 1 accepting the argument or 0 rejecting it.

The algorithms above should satisfy the following three properties.

(1) Completeness. For all $\kappa \in \mathbb{N}$, $(s, w) \in R$, $\Pr(\text{Verify}(\Sigma, s, w) = 1) = 1$.

(2) Computational soundness. For all PPT adversaries $\mathscr{A}$, the following probability is negligible in κ:

$$\Pr(\text{Verify}(\Sigma, \widetilde{s}, \widetilde{w}) = 1 \wedge \widetilde{s} \notin L). \tag{1}$$

(3) Zero-knowledge. There exists a PPT simulator $(\mathscr{C}_1, \mathscr{C}_2)$ such that $\mathscr{C}_1$ outputs a simulated CRS Σ and trapdoor τ; $\mathscr{C}_2$ takes as input Σ, a statement s and τ and outputs a simulated proof π'; and, for all PPT adversaries $(\mathscr{A}_1, \mathscr{A}_2)$, the following probability is negligible in κ:

$$\left|\Pr((s, w) \in R \wedge \mathscr{A}_2(\pi, st) = 1) - \Pr\left((s, w) \in R \wedge \mathscr{A}_2(\pi\prime, st) = 1\right)\right|. \tag{2}$$

Definition 2 (NIZK Argument of Knowledge [20]). A NIZK argument of knowledge for a relation R is a NIZK argument for R with the following additional extractability property.

(1) Extraction. For any PPT adversary $\mathcal{A}$, random string $r \longleftarrow \{0,1\}^*$, there exists a PPT algorithm Ext outputting $w\prime$ such that the following probability is negligible in κ:

$$\Pr(\text{Verify}(\Sigma, \tilde{s}, \tilde{\pi}) = 1 \wedge R(\tilde{s}, w\prime) = 0). \quad (3)$$

2.3. (t, u)-Threshold Secret-Sharing Scheme. In this section, we describe a (t, u)-threshold scheme [21] which enables making u shares (distribution) and recovering the secret from t or more shares (recovery) using just XOR operations, for arbitrary threshold t and the number of participants u. We will only use the distribution algorithm which is described in Algorithm 1. In this algorithm, the secret $y \in \{0,1\}^{d(u_p-1)}$ needs to be divided equally into $u_p - 1$ blocks $y_1, \ldots, y_{u_p-1} \in \{0,1\}^d$, where u_p is a prime number, and $d > 0$ denotes the bit-size of every divided piece of the secret. Also, it uses u shares, $y_1, \ldots, y_u$, of a (t, u_p)-threshold scheme to construct a (t, u)-threshold scheme if the desired number of participants u is a composite number (in our scheme, we set $u = u_p$).

These XORed terms are circulated in a specific pattern with t dimensions and do not overlap with each other because the properties of prime numbers are used. By an implementation on a PC, they showed that the proposed scheme is able to make u shares from the secret and recover the secret from t shares more quickly than Shamir's scheme [22] if u is not extremely large.

2.4. UTXO Ledger Model. Bitcoin, the most valuable and popular cryptocurrency, uses a graph-based ledger model built on the concept of UTXOs (unspent transaction outputs). In the UTXO ledger model, individual transactions consist of a list of inputs and a list of outputs. Each of the transactions can merge the bitcoins in the previous multiple accounts and transfer them to another one or more accounts. Figure 2 shows how UTXO model works, where Tx1 contains one input and two outputs, and Tx2 contains three inputs and two outputs.

2.5. Linkable Ring Signature. The biometric cryptosystem can be found in [5] and the NIZK scheme can be found in [12]. Our definitions are in the spirit of [4, 5, 12].

Definition 3. A linkable ring signature scheme based on SKE and NIZK consists of five algorithms:

(1) Setup*: on input of the user's biometric vector x and a finite field polynomial order $k-1$, output public helper data* PP.

(2) KeyGen*: on input of the user's biometric vector x' and public helper data* PP*, output the secret key-vectors y and its corresponding public key-vectors PK.*

(3) Sign*: on input of a message, the parameters g, h, R, G, and the set $L = \{PK_i^j\}_{i \in [n]}^{j \in [u]}$ where (y_j, PK_π^j) is a valid key pair output by* KeyGen *and $PK_\pi^j \in L$, output a* signature Ω.

(4) Verify*: on input of the purported signatures Ω, anyone can verify Ω and output a bit $b \in \{0,1\}$.*

(5) Link*: on input of two messages M_1, M_2 as well as two signatures Ω_1 and Ω_2, output a bit $b \in \{0,1\}$.*

2.6. Complexity Assumptions and Lemma

Definition 4 (Discrete Logarithm (DL) Assumption). *Given a generator g of G^*, where $G^* = \mathbb{G}$ or $\mathbb{G}_T$, and $a \in \mathbb{Z}_q^*$, for every adversary $\mathcal{A}$, $\Pr[\mathcal{A}(g, ag) = a] = \varepsilon(\kappa)$.*

Definition 5 (Decision Diffie-Hellman (DDH) Assumption). *Distinguish the distributions (ag, bg, abg) and (ag, bg, cg) with $a, b, c \longleftarrow \mathbb{Z}_q$ and $g \in \mathbb{G}$. The DDH assumption is the intractability of the problem for any PPT distinguisher.*

Lemma 1 (From Liu et al. [7]). *Let $\mathcal{A}$ be an attacker and $\mathcal{C}$ be a challenger; $\mathcal{C}$ invokes $\mathcal{A}$ to obtain a transcript $\mathcal{T}$; if $\mathcal{T}$ is successful, then $\mathcal{C}$ rewinds $\mathcal{T}$ to a header $\mathfrak{H}$ and resimulates $\mathcal{A}$ to obtain transcript $\mathcal{T}\prime$. If Pr ($\mathcal{T}$ succeeds) = ε, then Pr ($\mathcal{T}\prime$ succeeds) = ε.*

3. Security Model

In consideration of the security, our scheme should satisfy four fundamental properties: unforgeability, anonymity, linkability, and zero-knowledge which are very similar to the definitions given by [4, 7].

Before giving the definition, we give the definitions of the following queries at first. They will be carried out between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$, which together simulate the ability of the adversary.

(1) **Hash functions query:** $\mathcal{A}$ may request the values of the hash functions for any input.

(2) **Key query:** $\mathcal{A}$ requests the key of a user; $\mathcal{C}$ responds with the secret key.

(3) **Signature query:** $\mathcal{A}$ submits a tuple (M, x, L); $\mathcal{C}$ outputs a signature.

3.1. Unforgeability. We give the adversary model about unforgeability, which follows a similar structure as [4, 7].

For any PPT adversary $\mathcal{A}$, the advantage that $\mathcal{A}$ wins the following game can be ignored; then our scheme is said to be unforgeable.

Game I: an adversary $\mathcal{A}$ plays a game with a challenger $\mathcal{C}$ as follows.

Initialization: running the Setup algorithm, $\mathcal{C}$ obtains the public helper data PP and then gives it to $\mathcal{A}$.

Query: $\mathcal{A}$ performs a polynomially bounded number of queries.

Input: $y \in \{0,1\}^{d(u_p-1)}$.
Output: $y = (y_1, \ldots, y_u)$
(1) $s = 0^d$, $s_1 \| \cdots \| s_{u_p-1} = s$.
(2) for $1 \le i \le t-1$ do
(3) for $1 \le j \le u_p$ do
(4) Choose $r_j^i = \mathrm{GEN}(\{0,1\}^d)$;
(5) end
(6) end
(7) for $1 \le i \le u$ do
(8) for $1 \le j \le u_p - 1$ do
(9) Choose $y_{(i,j)} = (\oplus_{h=1}^{t-1} r_{h\cdot i+j}^h) \oplus s_{j-i}$;
(10) end
(11) $y_i = y_{(i,1)} \| \cdots \| y_{(i,u_p-1)}$
(12) end
(13) return $y = (y_1, \ldots, y_u)$.

Algorithm 1: DisThreshold (y, t, u).

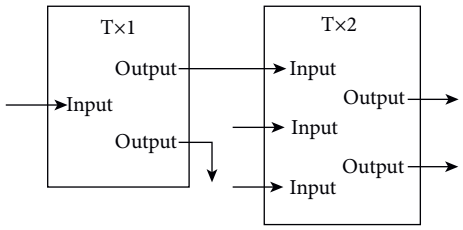

Figure 2: UTXO ledger model.

Forge: $\mathscr{A}$ submits a new tuple $(\Omega^*, x^*, M^*, L^*)$. $\mathscr{A}$ will win if the following conditions hold:

(1) Ω^* is a legal signature.

(2) $\mathscr{A}$ did not query the key of anyone in L^*.

(3) $\mathscr{A}$ did not query the tuple (x^*, M^*).

The advantage of the unforgeability is denoted by

$$\mathrm{Advantage}_{\mathscr{A}}^{\mathrm{Forge}} = \Pr[\mathscr{A} \text{ succeeds}]. \quad (4)$$

If for any PPT algorithm $\mathscr{A}$, the advantage of $\mathrm{Advantage}_{\mathscr{A}}^{\mathrm{Forge}}$ is negligible, we say the scheme is unforgeable.

3.2. Anonymity. Our scheme is said to be signer anonymity if for any PPT adversary $\mathscr{A}$, $\mathrm{Advantage}_{\mathscr{A}}^{\mathrm{Forge}}$ is negligible.

Game II: an adversary $\mathscr{A}$ plays a game with a challenger $\mathscr{C}$ as follows.

Initialization: it is the same as that in Game I.

Query: $\mathscr{A}$ performs a polynomially bounded number of queries.

Challenge: $\mathscr{A}$ outputs a new tuple $(M, x_{i,0}, x_{i,1}, L)$. $\mathscr{C}$ flips a coin $b \in \{0,1\}$ and then returns $\mathscr{A}$ with the signature $\Omega(M, x_{i,b}, L)$.

Guess: $\mathscr{A}$ outputs a bit b'. If $b = b'$, $\mathscr{A}$ is considered to succeed with the probability of $\mathrm{Success}_{\mathscr{A}}^{\mathrm{Anon}}$.

The anonymity advantage of our scheme is denoted by $\mathrm{Advantage}_{\mathscr{A}}^{\mathrm{Anon}} = |\mathrm{Success}_{\mathscr{A}}^{\mathrm{Anon}} - 1/2|$.

3.3. Linkability. If for any PPT adversary $\mathscr{A}$, the advantage of $\mathscr{A}$ is ignorable in the following game, our scheme is said to be linkable.

Game III: an adversary $\mathscr{A}$ plays a game with a challenger $\mathscr{C}$ as follows.

Initialization, Query: it is the same as that in Game I.

Unlink: $\mathscr{A}$ outputs two valid signatures (Ω_1, M_1, L_1) and (Ω_2, M_2, L_2) with respect to secret keys y_1 and y_2, respectively. $\mathscr{A}$ wins if the following conditions hold:

(1) Verify $(\Omega_1, M_1, L_1) = \mathrm{Verify}\ (\Omega_2, M_2, L_2) = 1$.

(2) $y_1 \cap y_2 \neq \varnothing$.

(3) $\mathrm{Link}\ (\Omega_1, \Omega_2) = \mathrm{unlink}$.

The advantage of the linkability is denoted by $\mathrm{Advantage}_{\mathscr{A}}^{\mathrm{Link}} = \Pr[\mathscr{A} \text{ wins}]$.

3.4. NIZK Argument. In ROM, if our scheme is proved to be completeness, computational soundness, and zero-knowledge, our protocol is a NIZK argument where the plaintext space is $\mathbb{Z}_q$.

4. Signature Scheme

4.1. Our Construction. The detailed steps of our scheme are given as follows.

Setup. (x)

On input of $\mathbf{x} \in \mathbb{Z}_q^d$, two random nonces $N, \widehat{N} \in \mathbb{Z}_q^{mq \times d}$, parameter $m \in \mathbb{Z}_q$, a finite field polynomial $f(\cdot)$ which encodes the secret key y, and a one-way hash function H: $\{0,1\}^* \longrightarrow \{0,1\}^l$, the Setup algorithm does as follows:

(1) Run IoM (x) to generate vectors $\varphi_x \longleftarrow \Omega^{\mathrm{IoM}}(x, m, N)$ and $\widehat{\varphi}_x \longleftarrow \Omega^{\mathrm{IoM}}(x, m, \widehat{N})$, where $\varphi_x = (\varphi_{x(1)}, \ldots, \varphi_{x(m)})$, $\widehat{\varphi}_x = (\widehat{\varphi}_{x(1)}, \ldots, \widehat{\varphi}_{x(m)})$.

(2) Compute $f(\varphi_x) = (f(\varphi_{x(1)}), \ldots, f(\varphi_{x(m)}))$.

(3) For $i \in [m]$, compute $ss_i = \widehat{\varphi}_{x(i)} \oplus f(\varphi_{x(i)})$, such that the secure sketch $ss = (s_1, \ldots, s_m)$.

(4) For $i \in [m]$, compute $\mathrm{tag}_i = H(\varphi_{x(i)} \| ss_i \| \widehat{\varphi}_{x(i)})$, such that $\mathrm{TAG} = (\mathrm{tag}_1, \ldots, \mathrm{tag}_m)$.

(5) Output the public helper data $PP = \{N, \widehat{N}, ss, \mathrm{TAG}\}$.

A formal description of this algorithm is shown Algorithm 2.

KeyGen. (x')

On input $x' \in \mathbb{Z}_q^d$, $u \in \mathbb{Z}_q$, the public helper data $PP = \{N, \widehat{N}, ss, \mathrm{TAG}\}$ with parameter m, generator $G \in \mathbb{G}$, and the set $U = \varnothing$. The KeyGen algorithm does as follows:

(1) Run $\mathrm{IoM}(x')$ to generate vectors $\varphi_{x'} \longleftarrow \Omega^{\mathrm{IoM}}(x', m, N)$ and $\widehat{\varphi}_{x'} \longleftarrow \Omega^{\mathrm{IoM}}(x', m, \widehat{N}) \in \mathcal{F}_q^m$.

(2) For $i \in [m]$, compute $\mathrm{tag}_i' = H(\varphi_{x'(i)} \| ss_i \| \widehat{\varphi}_{x'(i)})$, such that $\mathrm{TAG}' = (\mathrm{tag}_1', \ldots, \mathrm{tag}_m')$.

(3) For $i \in [m]$, compute $f(\varphi_{x(i)})' = ss_i \oplus \widehat{\varphi}_{x'(i)}$ while $tag_i' = tag_i$, and then add $(\varphi_{x(i)}, f(\varphi_{x(i)})')$ to an unlocking set U.

(4) If $|U| \geq k$, perform polynomial reconstruction with U and output the secret key $y = f(0) \in \mathbb{Z}_q^*$; else repeat step 1.

(5) Run DisThreshold $(y, u-1, u)$ to obtain y, output $y = (y_1, \ldots, y_u)$.

(6) For $j \in [u]$, compute $PK_i = y_i G \in \mathbb{G}$. The $PK = (PK_1, \ldots, PK_u)$ and y are the public key-vectors and the secret key-vectors, respectively.

A formal description of this algorithm is shown as Algorithm 3.

Sign.

On input the message $\in \{0,1\}^l$, generators $g, h, G \in \mathbb{G}$, $R \in \mathbb{Z}_q^*$, and public key-matrix $L = \{PK_1, \ldots, PK_n\}$ $(PK_i = (PK_i^1, \ldots, PK_i^u))$ containing n public key-vectors of length u, user's secret key y corresponding to PK_π. We do as follows:

(1) Let $H_1(\text{message}) = \mu$, choose $R_\mu, R_R \longleftarrow \mathbb{Z}_q^*$, compute $C = \mu g + Rh$; $C_R = R_\mu g + R_R h$; $e = H_2(C, C_R)$.

(2) Compute $M = e\mu + R_\mu$, $R' = eR + R_R$.

(3) For $j \in [u]$, $i = 1, \ldots, \widehat{\pi}, \ldots, n$, choose $s_i^j \longleftarrow \mathbb{Z}_q$; when $i = \pi$, choose $a_\pi^j \longleftarrow \mathbb{Z}_q$.

(4) For $j \in [u]$, compute the key image $I_j = y_j H_p(PK_\pi^j) \in \mathbb{G}$.

(5) When $i = \pi$, for $j \in [u]$, compute $L_\pi^j = a_\pi^j G$; $R_\pi^j = a_\pi^j H_p(PK_\pi^j)$; $c_{\pi+1} = H_3(M, L_\pi^1, R_\pi^1, \ldots, L_\pi^u, R_\pi^u)$.

(6) When $i = \pi+1, \ldots, n, 1, \ldots, \pi-1$, for $j \in [u]$, compute $L_i^j = s_i^j G + c_i PK_i^j$; $R_i^j = s_i^j H_p(PK_i^j) + c_i I_j$; $c_{i+1} = H_3(M, L_i^1, R_i^1, \ldots, L_i^u, R_i^u)$.

(7) When $i = \pi$, for $j \in [u]$, compute $s_\pi^j = a_j - c_\pi y_j \mod q$.

(8) Output $\Omega(M) = (I_j, c_1, s_i^j, C_R, M, R')_{i \in [n], j \in [u]}$.

A formal description of this algorithm is shown as Algorithm 4.

Verify

(1) The verifier computes $e = H_2(C, C_R)$. Then, checking whether $eC + C_R = Mg + R'h$. If the congruence holds, return $''1''$, otherwise $''0''$.

(2) For $i \in [n]$, $j \in [u]$, the verifier regenerates all the L_i^j, R_i^j, c_{i+1} and verifies whether $c_{n+1} = c_1$. If the congruence holds, return $''1''$, otherwise $''0''$.

A formal description of this algorithm is shown Algorithm 5.

Link.

For a fixed set of public key-vectors, given two messages $M, M\prime$, two valid signatures $\Omega(M)$ and $\Omega\prime(M\prime)$, the verifier outputs *link* if the key image $I_j = I_j'(j \in [u])$; otherwise the verifier outputs *unlink*.

5. Security Analysis

In this section, the security proofs of our scheme are given, which follow similarly to the security proofs of [4, 7].

Theorem 1. *In ROM, the scheme is unforgeable if DL problem is hard.*

Proof. (Similar proof to Theorem 6 of [4] and Theorem 1 of [7]) We follow the notation introduced above. Suppose that an adversary $\mathcal{A}$ can forge signature with nonnegligible probability; then we certainly can construct a *PPT* simulator $\mathcal{C}$ which can extract a solution to the DL problem. Given a random instance of the DL problem with security level κ, $\mathcal{C}$ is asked to solve the DL problem in polynomial time.

First of all, $\mathcal{C}$ maintains 4 lists $\ell_1, \ell_2, \ell_3, \ell_4, \ell_5$ in its local storage to store the outputs of *H-oracle*, H_1*-oracle*, H_3*-oracle*, key query, and sign query, initially setting to be empty. The interaction between $\mathcal{A}$ and $\mathcal{C}$ does as follows:

Initialization: Running the *Setup* algorithm; $\mathcal{C}$ gives $\mathcal{A}$ parameters.

Query: $\mathcal{A}$ is allowed to make the following queries (supposed that $\mathcal{A}$ will not initiate repeated queries).

(1) H query: $\mathcal{C}$ maintains list ℓ_1 of tuple (x_i, TAG_i). For a request of x_i, if x_i contains list ℓ_1, $\mathcal{C}$ returns the corresponding tuple to $\mathcal{A}$; otherwise, $\mathcal{C}$ randomly chooses $\mathrm{TAG}_i \in \{0,1\}^l$ and returns it to $\mathcal{A}$; meanwhile, it stores the tuple (x_i, TAG_i) into list ℓ_1.

(2) H_μ query: $\mathcal{C}$ maintains list ℓ_2 of tuple $(\text{message}, M)$. For a request of message, if message contains list ℓ_1, $\mathcal{C}$ returns the corresponding tuple to $\mathcal{A}$; otherwise, $\mathcal{C}$ randomly chooses M and returns it to $\mathcal{A}$; meanwhile, it stores the tuple $(message, M)$ into list ℓ_2.

(3) H_3 query: $\mathcal{C}$ maintains list ℓ_3 of tuple $(M, s_i^j, c_i, L)_{i \in [n], j \in [u]}$. For a request of M, a set L has n public key-vectors and randomly chooses $s_i^j \in \mathbb{Z}_q (i \in [n], j \in [u])$; if (M, L) contains list ℓ_3, $\mathcal{C}$ returns the corresponding tuple to $\mathcal{A}$; otherwise, $\mathcal{C}$ randomly chooses $c_i \in \mathbb{Z}_q$ and returns it to $\mathcal{A}$; meanwhile, it stores the tuple $(M, s_i^j, c_i, L)_{i \in [n], j \in [u]}$ into list ℓ_3.

(4) Key query: for a request of x_i, if the tuple (x_i, TAG_i) contains list ℓ_1, $\mathcal{C}$ performs polynomial reconstruction function to generate $f(\cdot)$ and the secret key

Input: x, m, $f(\cdot)$, N, $\widehat{N}$
Output: PP = $\{N, \widehat{N}, ss, \text{TAG}\}$
(1) Let $\varphi_x = \Omega^{\text{IoM}}(x, m, N)$, $\widehat{\varphi}_x = \Omega^{\text{IoM}}(x, m, \widehat{N})$.
(2) Call $f(\cdot)$ to obtain
(3) $f(\varphi_x) = (f(\varphi_{x(1)}), \ldots, f(\varphi_{x(m)}))$.
(4) for $1 \le i \le m$ do
(5) Compute $ss_i = \widehat{\varphi}_{x(i)} \oplus f(\varphi_{x(i)})$,
(6) $\text{tag}_i = H(\varphi_{x(i)} \| ss_i \| \widehat{\varphi}_{x(i)})$.
(7) end
(8) Let $ss = (s_1, \ldots, s_m)$, $\text{TAG} = (\text{tag}_1, \ldots, \text{tag}_m)$.
(9) return PP = $\{N, \widehat{N}, ss, \text{TAG}\}$.

Algorithm 2: Setup (x).

Input: $x\prime$, u, m, G, U, $PP = \{N, \widehat{N}, ss, \text{TAG}\}$.
Output: PK.
(1) Compute $\varphi_{x_\prime} = \Omega^{\text{IoM}}(x\prime, m, N)$, $\widehat{\varphi}_{x_\prime} = \Omega^{\text{IoM}}(x\prime, m, \widehat{N})$.
(2) Let $U = \varnothing$.
(3) for $1 \le i \le m$ do
(4) Compute $\text{tag}_i' = H(\varphi_{x\prime(i)} \| ss_i \| \widehat{\varphi}_{x\prime(i)})$.
(5) if $\text{tag}_i' == \text{tag}_i$ then
(6) Compute $f(\varphi_{\mathbf{x}(i)})' = ss_i \oplus \widehat{\varphi}_{\mathbf{x}'(i)}$
(7) if $(\varphi_{x(i)}, f(\varphi_{x(i)})\prime) \notin U$ then
(8) Let $U = U \cup (\varphi_{x(i)}, f(\varphi_{x(i)})\prime)$
(9) end
(10) end
(11) end
(12) Let $t = |U|$.
(13) if $t \ge k$ then
(14) Call function lagrange$(\cdot)(0)$ to obtain y
(15) Run DisThreshold$(y, u-1, u)$ to obtain $\mathbf{y}$
(16) for $1 \le i \le u$ do
(17) Compute $PK_i = y_i G$.
(18) end
(19) end
(20) return $PK = \{PK_1, \ldots, PK_u\}$, y.

Algorithm 3: KeyGen $(x\prime)$.

Input: message $\in \{0,1\}^l$, $g, h, G \in \mathbb{G}$, $R \in \mathbb{Z}_q^*$, $L = \{PK_1, \ldots, PK_n\}$ $(PK_i = (PK_i^1, \ldots, PK_i^u))$, y.
Output: $\Omega(M) = (I_j, c_1, s_i^j, C_R, M, R\prime)_{i\in[n], j\in[u]}$.
(1) Let $H_1(message) = \mu$.
(2) Compute $C = \mu g + Rh$, $C_R = R_\mu g + R_R h$ and $e = H_2(C, C_R)$ where
(3) $R_\mu, R_R \longrightarrow \mathbb{Z}_q^*$.
(4) Compute $M = e\mu + R_\mu$, $R\prime = eR + R_R$.
(5) for $1 \le j \le u$ do
(6) Choose $a_\pi^j \leftarrow \mathbb{Z}_q$
(7) Compute $I_j = y_j H_p(PK_\pi^j)$; $L_\pi^j = a_\pi^j G$;
(8) $R_\pi^j = a_\pi^j H_p(PK_\pi^j)$;
(9) end
(10) Compute $c_{\pi+1} = H_3(M, L_\pi^1, R_\pi^1, \ldots, L_\pi^u, R_\pi^u)$.
(11) for $i = \pi+1, \ldots, n, 1, \ldots, \pi-1$ do
(12) for $1 \le j \le u$ do
(13) Choose $s_i^j \leftarrow \mathbb{Z}_q$

Algorithm 4: Continued.

(14) Compute $L_i^j = s_i^j G + c_i PK_i^j$;
(15) $R_i^j = s_i^j H_p(PK_i^j) + c_i I_j$;
(16) end
(17) Compute $c_{i+1} = H_3(M, L_i^1, R_i^1, \ldots, L_i^u, R_i^u)$;
(18) end
(19) for $1 \le j \le u$ do
(20) Compute $s_\pi^j = a_j - c_\pi y_j \bmod q$;
(21) end
(22) return $\Omega(M) = (I_j, c_1, s_i^j, C_R, M, R\prime)_{i\in[n], j\in[u]}$.

Algorithm 4: Sign (message, L, y).

Input: $\Omega(M) = (I_j, c_1, s_i^j, C_R, M, R\prime)_{i\in[n], j\in[u]}$,
$g, h \in \mathbb{G}$, $L = \{PK_1, \ldots, PK_n\}$ $(PK_i = (PK_i^1, \ldots, PK_i^u))$.
Output: 0 or 1.
(1) Compute $e = H_2(C, C_R)$.
(2) if $eC + C_R = Mg + R'h$ then
(3) for $1 \le i \le n$ do
(4) for $1 \le j \le u$ do
(5) Compute $L_i^j = s_i^j G + c_i PK_i^j$;
(6) $R_i^j = s_i^j H_p(PK_i^j) + c_i I_j$
(7) end
(8) Compute $c_{i+1} = H_3(M, L_i^1, R_i^1, \ldots, L_i^u, R_i^u)$;
(9) end
(10) else
(11) 0
(12) end
(13) if $c_{n+1} = c_1$ then 1;
(14) else 0;
(15) return 0 or 1.

Algorithm 5: Verify(Ω, L).

$s_i = f(0)$ and returns it to $\mathcal{A}$. Meanwhile, $\mathcal{C}$ stores the tuple (x_i, s_i) into the ℓ_4.

(5) Sign query: for a request of message, the user's biometric vector x_i, and a set L of n public key-vectors where π is the real secret index, $\mathcal{C}$ generates a signature:

(a) Check list ℓ_4, if there exists x_i, the real secret key $s_\pi = y_\pi$.

(b) Check list ℓ_2, if there exists (message, M), the secret message is M.

(c) For $j \in [u]$, compute $I_j = y_j H_p(PK_\pi^j)$.
(d) For $j \in [u]$, choose $a_\pi^j \longleftarrow \mathbb{Z}_q$ and $c_{\pi+1} \longleftarrow \mathbb{Z}_q$.
(e) When $i = \pi + 1, \ldots, n, 1, \ldots, \pi - 1$:
For $j \in [u]$, choose $s_i^j \longleftarrow \mathbb{Z}_q$; compute $L_i^j = s_i^j G + c_i PK_i^j$; $R_i^j = s_i^j H_p(PK_i^j) + c_j I_j$;
Compute $c_{i+1} = H_3(M, L_i^1, R_i^1, \ldots, L_i^u, R_i^u)$, add $(s_i^j, c_{i+1}, C_R, M, R')_{(i\in[n], j\in[u])}$ into ℓ_5. If collision occurs, repeat steps 3 and 5.
(f) Output $\Omega(M) = (I_j, c_1, s_i^j, C_R, M, R')_{i\in[n], j\in[u]}$ as a signature. It can be seen from the signing process that $\Omega(M)$ is a valid signature.

Forge: $\mathcal{A}$ outputs a forged signature $\Omega(M^*) = (I_j^*, c_1^*, s_i^{j^*}, C_R^*, M^*, R'^*)_{i\in[n], j\in[u]}$. Assume that $\mathcal{A}$ makes no more than $q_1 + q_H + uq_3$ queries to the signing oracles H_1, H, H_3 and $\mathcal{SO}$.

From Lemma 1, for each successful forgery $\mathcal{A}$ completes with transcript $\mathcal{T}$, there are $u_\mathcal{T}$ queries to H_3 matching the n queries used to verify the signature. Let $X_{i_1}, \ldots, X_{i_u}$ denote the i^{th} forgery and let π be the index for the last verification query; we have $X_{i_\ell} = H_3(M, L_{\pi-1}^1, R_{\pi-1}^1, \ldots, L_{\pi-1}^{u_\mathcal{T}}, R_{\pi-1}^{u_\mathcal{T}})$.

If $i_1 = \ell$, $\mathcal{A}$ produces an attempted forgery Ω that is an (ℓ, π)-forgery. By assumption, there exists (ℓ, π) for giving a successful forgery; $\varepsilon_{\ell,\pi}(\mathcal{T})$ satisfies

$$\varepsilon_{\ell,\pi} \ge \frac{1}{u\mathcal{T}(q_1 + q_H + u\mathcal{T}q_3)} \cdot \frac{1}{Q_1(\kappa)} \ge \frac{1}{u(q_1 + q_H + uq_3)} \cdot \frac{1}{Q_1(\kappa)}. \quad (5)$$

Then, $\mathcal{A}$ rewinds $\mathcal{T}$ before the ℓ^{th} query and again attempts a forgery on the same set of keys that $\mathcal{T}'$ satisfies

$$\varepsilon_{\ell,\pi}(\mathcal{T}') \ge \frac{1}{u(q_1 + q_H + uq_3)} \cdot \frac{1}{Q_2(\kappa)}, \quad (6)$$

and also a successful forgery, where Q_2 is a polynomial inputting a security parameter κ.

Therefore, the probability that both $\mathscr{T}$ and $\mathscr{T}'$ correspond to verifying forgeries Ω and Ω' is nonnegligible:

$$\varepsilon_{\ell,\pi}(\mathscr{T}\&\mathscr{T}') \geq \left(\varepsilon_{\ell,\pi}(\mathscr{T})\right)^2. \tag{7}$$

In the way above, we again obtain a forged signature $\Omega(M_i') = (I_j', c_1', s_i^j, C_R', M', R'')_{i\in[n], j\in[u]}$. For each j, we have $s_\pi^j \neq s_\pi^j$, $c_\pi \neq c_\pi'$, and we can solve y_π^j as

$$y_\pi^j = \frac{s_\pi'^j - s_\pi^j}{c_\pi' - c_\pi} \bmod q, \tag{8}$$

which contradicts the DL assumption. □

Theorem 2. *In ROM, the proposed scheme is signer-anonymous under the DDH assumption.*

Proof (similar proof to Theorem 8 of [4]). Assume that the DDH problem is hard in the cyclic group generated by $\mathbb{G}$ and suppose there exists a PPT adversary $\mathscr{A}$ against signer ambiguity. After that, given a set L of n public key-vectors of length u, a set of t biometric vectors $\mathscr{D}_t = \{x_1, \ldots, x_t\}$, and a valid signature Ω on L signed by a user with respect to a key-vector $\overline{PK}$ such that the corresponding biometric vector $\overline{x}_\pi$ satisfies $\overline{x}_\pi \notin \mathscr{D}_t$, then, $\mathscr{A}$ can win the game above with probability

$$\Pr[\mathscr{A} \longrightarrow \pi] \geq \frac{1}{n-t} + \frac{1}{Q_3(\kappa)} \tag{9}$$

for some polynomial $Q_3(\kappa)$. We certainly can construct a PPT simulator $\mathscr{C}$ which takes as inputs a tuple (G, aG, bG, c_iG), where $i \in \{0,1\}$ is randomly chosen and not a priori known to $\mathscr{C}$, $c_1 = ab$, and c_0 is a random scalar; then $\mathscr{C}$ can output i and solve the DDH problem with probability

$$\Pr[\mathscr{C}(G, aG, bG, c_iG) \longrightarrow i] \geq \frac{1}{2} + \frac{1}{Q_4(\kappa)}, \tag{10}$$

for some polynomial $Q_4(\kappa)$.

Inputting scalars a, c, the user's biometric vector x, a set L of n public key-vectors of length u, index π, and message M, we act as follows.

Initialization, Query: it is the same as that in Theorem 1.

Challenge: $\mathscr{A}$ submits a new tuple $(M, x_{i,0}, x_{i,1}, L)$, where the public key-vectors corresponding to $\mathbf{x}_{i,0}, \mathbf{x}_{i,1}$ are in L. $\mathscr{C}$ flips a coin $b \in \{0,1\}$ and then returns $\mathscr{A}$ with the signature $\Omega(M, \mathbf{x}_{i,b}, L)$.

Guess: $\mathscr{A}$ outputs a bit $b' \in \{0,1\}$.

Given a tuple (G, aG, bG, c_iG) where a, b are randomly selected scalars, with $c_1 = ab$, c_0 is a random scalar, $i \in \{0,1\}$, $\mathscr{C}$ takes the following steps to solve the DDH problem with nonnegligible probability.

Firstly, $\mathscr{C}$ grabs a private/public key-vector pair $(\overline{y}, \overline{PK})$ from H_1, H_3 and a random $\gamma$and computes $s = a - \gamma y$. Then, $\mathscr{C}$ performs SIMNIZKP [4] with arbitrarily selected key-vectors $\{\overline{PK}_i\}_{i\in[n]}$ such that $\overline{PK} = \overline{PK}_\pi$, $a \longrightarrow a$, $c_i \longrightarrow c$, some message M, and $\overline{y} \longrightarrow \overline{y}$.

If $i = 1$, then $c = ab$, $\log_G aG = \log_{bG} cG = a$; and due to the fact that $\mathscr{A}$ is assumed to be able to find π with nonnegligible probability, then there is a nonnegligible probability over 1/2 that $\mathscr{A}$ returns 1 (upon which $\mathscr{C}$ returns 1). Meanwhile, if $i = 0$, then $\mathscr{A}$ returns 1 only with probability 1/2, and so for some nonnegligible probability over 1/2, $\mathscr{C}$ returns the same value as $\mathscr{A}$ and thus solves the DDH problem for randomly chosen scalars with nonnegligible probability over 1/2, which is a contradiction. □

Theorem 3. *In ROM, our scheme satisfies linkability.*

Proof (similar proof to Theorem 7 of [4]). Suppose that a PPT adversary $\mathscr{A}$ can produce two unlinkable signatures (Ω_1, M_1, L_1) and (Ω_2, M_2, L_2) with nonnegligible probability, and they signed with respect to public key-matrices L_1 and L_2 such that there exists a public key PK in both L_1 and L_2 that is negligible.

Suppose to the contrary that Ω_1 and Ω_2 both signed with respect to public key-matrices L_1 and L_2 such that there exists a public key PK in both L_1 and L_2, then with overwhelming probability, there exists the indexes π and π' for the public keys in Ω_1 and Ω_2, respectively, such that

$$\begin{aligned} L_\pi^j &= s_\pi^j G + c_\pi PK_\pi^j; R_\pi^j = s_\pi^j H_p\left(PK_\pi^j\right) + c_\pi I_j; \\ L_{\pi'}^{'j} &= s_{\pi'}^{'j} G + c_{\pi'} PK_{\pi'}^{'j}; R_{\pi'}^{'j} = s_{\pi'}^{'j} H_p\left(PK_{\pi'}^{'j}\right) + c_{\pi'} I_{j'}. \end{aligned} \tag{11}$$

Thus, we have

$$\log_G L_\pi^j = \log_{H_p\left(PK_\pi^j\right)} R_\pi^j;\ \log_G L_\pi^{'j} = \log_{H_p\left(PK_\pi^{'j}\right)} R_\pi^{'j}. \tag{12}$$

For I_j and $I_{j'}$, it follows that $I_j = y_j H_p(PK_\pi^j) = y_j H_p(PK)$ and $I_{j'} = y_j H_p(PK)$; the private key is related to the user's biometric vector within $|U| \geq k$. In this way, two signatures Ω_1 and Ω_2 include $I_j = I_{j'}$; since the duplicate key images are rejected, one of them must not verify. □

Theorem 4. *In ROM, our scheme is a NIZK argument of plaintext knowledge.*

Proof. In the above scheme, we have completeness, computational soundness, and zero-knowledge. Proving procedures is as described in [12]. □

6. Efficiency Analysis

In this section, we give a brief efficiency comparison among the scheme of ours [4, 10]. The communication complexity of our scheme is almost $\mathcal{O}(u(n+1))$ where n is the number of public key vectors in each set. As shown in [4], the communication complexity of its scheme is also $\mathcal{O}(u(n+1))$. In contrast, it is $\mathcal{O}(n)$ in [10].

As shown in Table 2, the cost of time and communication in our scheme is larger than [4, 10]. However, our scheme constructs a linkable ring signature scheme based on NIZK and SKE protocols to enhance the status of privacy-preserving, which is the vital improvement of our scheme.

Table 2: Comparison of time and communication cost.

Ref.	Prover	Verifier	Communication
[4]	$2.1un\cdot\ \exp_1 + un\cdot H_q$	$2.2un\cdot\ \exp_1 + un\cdot H_q$	$u\lvert\mathbb{G}\rvert + \lvert H\rvert + un\lvert\mathbb{Z}_q\rvert$
[10]	$0.4(n+1)(u+1)\cdot\ \exp_1 + (n+1)\cdot p + (n+1)\cdot\ \exp_T$	$3.2\lambda(n+1)\cdot\ \exp_q + (1.1\lambda + 0.4u + 4)(n+1)\cdot\ \exp + 3(n+1)\cdot p + 2.2(n+1)\cdot\ \exp_T$	$(\lambda+5)(n+1)\lvert\mathbb{G}\rvert + (3\lambda+6)(n+1)\lvert\mathbb{Z}_p\rvert + 3\lambda(n+1)\lvert\mathbb{Z}_q\rvert + (n+1)\lvert\mathbb{G}_T\rvert$
Ours	$2.5un\cdot\ \exp_1 + un\cdot H_q$	$2.7un\cdot\ \exp_1 + un\cdot H_q$	$(u+1)\lvert\mathbb{G}\rvert + \lvert H\rvert + (un+2)\lvert\mathbb{Z}_q\rvert$

[*] "n": the number of public key-vectors in each set; "u": the length of each public key-vector; "λ": the length of element in $\mathbb{Z}_p$; "$\exp_1$": an exponentiation operation in group $\mathbb{G}_1$; "$\exp_T$": an exponentiation operation in group $\mathbb{G}_T$; "$\exp_q$": an exponentiation operation in group $\mathbb{G}_q \subset \mathbb{Z}_p^*$; "$p$": a bilinear pairing operation; H_p: a map-to-point hash function; $|H|$: the output size of a hash function H; $|\mathbb{G}_1|$: the length of element in group $\mathbb{G}_1$, similarly for $|\mathbb{Z}_p|$, $|\mathbb{Z}_q|$, and $|\mathbb{G}_T|$.

7. Experiments and Discussion

In this section, we present a simple demonstration of our scheme by using *Python*.

7.1. Parameters. In order to get fingerprint vectors, we adopt the idea of fingerprint image conversion to fingerprint vector which is consisted of several sequential stages: getting fingerprint image, preprocessing, and taking attribute values for feature key points. For each user, one of the fingerprint vectors is used for *Setup*, and other fingerprint vectors from the same user are used to retrieve the secret key with the procedure described in Section 5.

In order to preserve the accuracy performance at the same level as in the original fingerprint in [5], we choose the appropriate parameters of SKE algorithm to control FAR, FRR, and EER within a suitable range in our experiments. Meanwhile, these parameters can also get a theoretically favorable level of the computational security to resist the brute-force and false-accept attacks. For all the experiments, we choose $m = 1024$, $q = 251$, $q' = 2^{80}$ for the dataset (FVC 2002 subset DB1, DB2) with $k = 18$.

As for the signing process, we make improvements based on the original Ring CT [4] codes by adding the SKE and NIZK protocols. In this way, we generate a new code for our scheme. Specifically, we use Keccak-256 hash function to generate the secret keys as well as the public keys and set parameters of the elliptic curve as same as Ed25519. We implemented our scheme in Intel Core i 5 2.5 GHz, 4 Gb RAM, MacOS 10.13.6. Each element in $\mathbb{Z}_q$ is represented by 32 bytes.

7.2. Experimental Results. For a small ring size, we repeat the signing and verification process separately with $n = 6, 8, 10, 12$ and $u = 4, 6, 8, 10$. The experimental results are presented in Table 1.

For a relatively large ring size, we repeat the signing and verification process; the experimental results are presented in Figure 3.

As shown in Figure 3, the running time of signature in our scheme increases linearly with the number of group accounts. Meanwhile, the running times of signing and verification process are almost the same. It also can be concluded from the construction of our scheme.

As shown in Table 3, when we use a small ring size in our signature scheme, the simulation experiment can be carried out by using an ordinary personal computer. The result of experiment may have some abnormal deviations for different computer hardware, but overall it represents the results of our scheme where we run the scheme on our personal computer. This is useful for mobile demonstration in section 7. According to the experimental results above, we can see that our scheme can meet the needs of practical applications.

8. Anonymous Cryptocurrency Transaction Model Based on LRS Scheme

Based on the proposed scheme and demonstration above, we propose a relatively private cryptocurrency model. Based on the experimental results on PC and private cryptocurrency model, we design a mobile simulation interface which means it may have a practical application value in mobile phone.

In the transaction schematics (Figure 4), we suppose that Alice sends money to Bob. In this process, Alice's anonymity is protected by ring signature and NIZK protocol. One-time address is used to protect Bob's anonymity. Moreover, we use SKE algorithm to protect secret key on both sides and use linkable tag to prevent double spending.

(1) During the registration phase (Figure 5(a)), while Alice enters the personal information and password as prompted, she also takes the fingerprint photo at the same time. On the input information above, the system runs the biometric encryption algorithm SKE to encrypt Alice's password. In other words, the system runs $Setup$ (x) algorithm and stores the public helper data PP.

(2) In the login page (Figure 5(b)), Alice uses the new fingerprint to unlock the interface. On input of the new fingerprint x' (the query biometric vector in the Section 2 part A), the system runs $KeyGen$ (x') algorithm to recover the password through the public helper data PP in step 1. Similarly, Bob does the same operations.

(3) As shown in Figure 5(c), Alice transfers the money to Bob. In addition, for multilayer ring signature scheme, Alice chooses the number n. Note could be indicated in remarks column, if it has. Finally, Alice clicks on the sign button to sign for this transaction. For system, Alice generates a one-time address for Bob, and no one other than Bob (including Alice) can recover the full signature key. Then, the system selects the $n-1$ addresses to get the equivalent amount of currency. Thus, the system inputs all Alice's previous transactions into a hash function and obtains the hash value. Using the NIZK algorithm stated above to blind the message M in Figure 4(c), the system runs signature algorithm to obtain $\Omega(M)$. Finally, the system generates a transaction Tx and broadcasts Tx throughout the P2P network (Figure 5(d)).

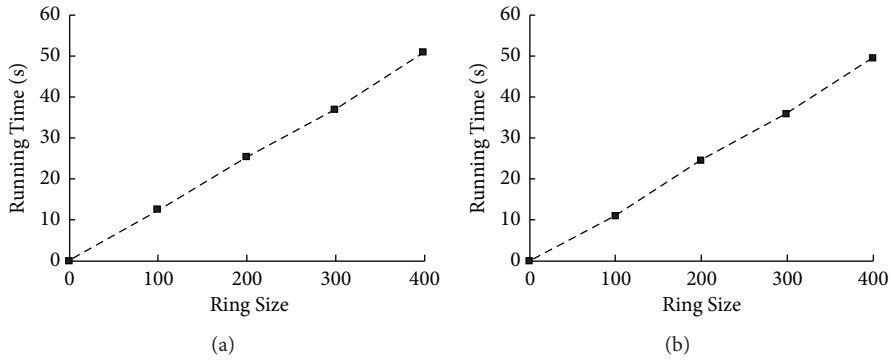

Figure 3: Performance of our shame. (a) Running time of sign. (b) Running time of verify.

Table 3: Average running time of every step for different ring numbers and public keys.

Average running time	$n = 6$		$n = 8$		$n = 10$		$n = 12$	
	Sign	Verify	Sign	Verify	Sign	Verify	Sign	Verify
$u = 4$	2.74	2.32	3.16	3.04	4.08	3.76	4.89	4.72
$u = 6$	2.97	2.77	4.45	4.21	6.53	6.50	8.06	7.81
$u = 8$	4.33	4.18	6.97	6.34	9.13	8.84	11.53	11.10
$u = 10$	6.19	5.94	8.86	8.22	12.17	11.90	15.32	14.95

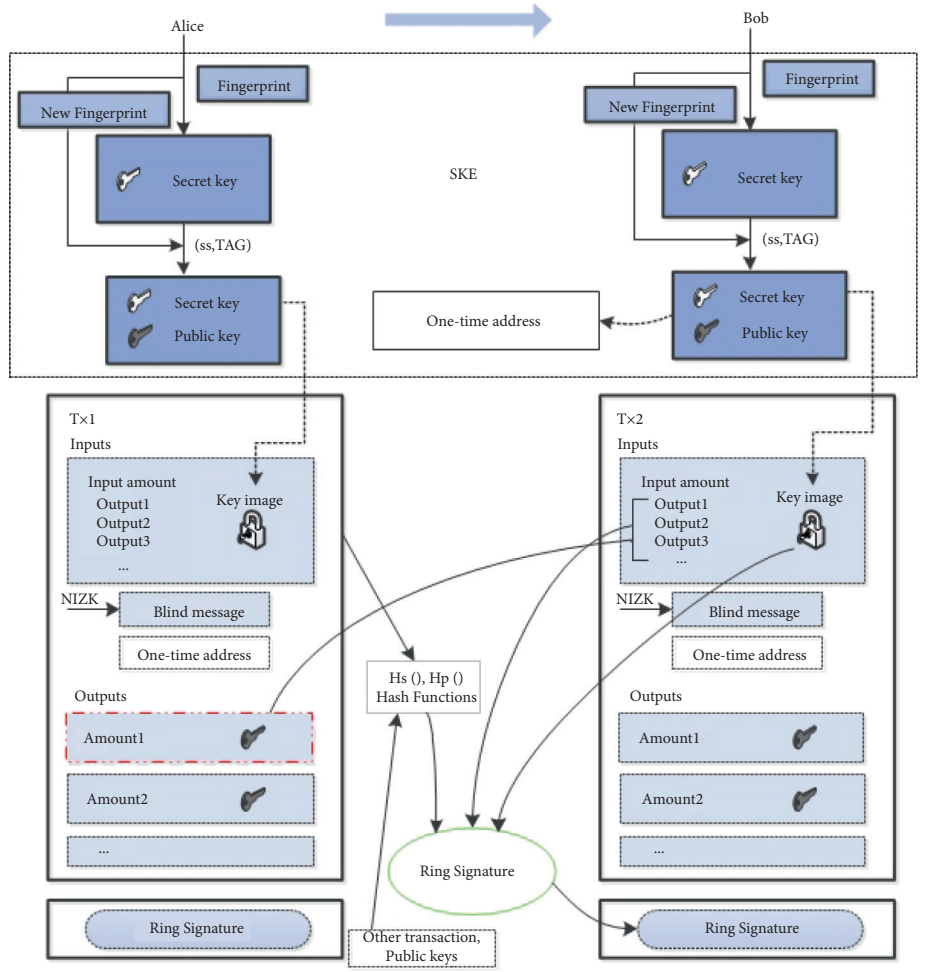

Figure 4: Cryptocurrency transactions model.

(a) (b) (c)

(d) (e)

Figure 5: Mobile demonstration interface.

(4) In the verification phase (Figure 5(e)), everyone in the P2P network can verify our transactions without knowing our plain message.

However, we do not use an authentication mechanism in the registration phase which will reduce the security of cryptocurrency transactions model. To deal with this issue, we consider using the following two-factor or three-factor authentication mechanisms. Stanislaw et al. [23] presented a secure two-factor authentication system based on the possession by the user of a password and a cryptocapable device which can achieve end-to-end security. Qiu et al. [24] put forward a provably secure three-factor protocol for mobile lightweight devices, which can achieve truly three-factor security while providing password change friendliness. Wang et al. [25] advanced a new two-factor authentication mechanism to resolve the various issues arising from user corruption and server compromise. Jiang et al. [26] proposed a cloud-centric three-factor authentication and key agreement protocol integrating passwords, biometrics, and smart cards to ensure secure access to both cloud and autonomous vehicles.

9. Conclusion

In this paper, we construct a linkable ring signature scheme based on NIZK and SKE protocols to enhance the status of privacy-preserving. In our signature scheme, we use the SKE algorithm to protect the secret key. Simultaneously, we also use the NIZK protocol to encrypt the plain message without revealing redundant information of the message in the verification process. With respect to the NIZK part, the NIZK protocol we used satisfies three characteristics: completeness, soundness, and zero-knowledge. At the same time, our scheme holds unforgeability, anonymity, and linkability in ROM. We also demonstrate the practical performance of our scheme on a personal computer. The result provides us with strong confidence in applying our scheme in practice.

However, since we apply the SKE and NIZK protocols to original Ring CT scheme to encrypt the secret key and the message, the running time and communication cost of our scheme have slightly increased. Although our signature

scheme may meet a real need, we will improve our scheme to reduce the running time and communication cost in our future work. In addition, the security of our scheme is based on the hardness of the DL problem, which is vulnerable to quantum attacks. In the future research, we will improve our scheme based on postquantum cryptography.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was partially supported by the Key Program of the Nature Science Foundation of Zhejiang Province of China (no. LZ17F020002) and the National Science Foundation of China (no. 61772166).

References

[1] p. Koshy, D. Koshy, and P. Mcdaniel, "An analysis of anonymity in bitcoin using P2P network traffic," *Financial Cryptography and Data Security*, vol. 8437, pp. 469–485, 2014.

[2] E. B. Sasson, A. Chiesa, and C. Garman, "Zerocash: decentralized anonymous payments from bitcoin (extended version)," in *Proceedings of the 2014 IEEE Symposium on Security and Privacy (SP)*, pp. 459–474, IEEE, Berkeley, CA, USA, May 2014.

[3] B. S. Eli, C. Alessandro, T. Eran, and M. Virza, "Succinct non-interactive zero knowledge for a von Neumann architecture," in *Proceedings of the 23rd USENIX conference on Security Symposium (SEC'14)*, pp. 781–796, USENIX, San Deigo, CA, August 2014.

[4] S. Noether, *Ring Signature Confidential Transactions for Monero*, Cryptology ePrint Archive, 2015, https://eprint.iacr.org/.

[5] Y.-L. Lai, J. Y. Hwang, Z. Jin, S. Kim, S. Cho, and A. B. J. Teoh, "Symmetric keyring encryption scheme for biometric cryptosystem," *Information Sciences*, vol. 502, pp. 492–509, 2019.

[6] D. Chang, S. Garg, M. Hasan, and S. Mishra, "Cancelable multi-biometric approach using fuzzy extractor and novel bit-wise encryption," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3152–3167, 2020.

[7] J. K. Liu, V. K. Wei, and D. S. Wong, "Linkable spontaneous anonymous group signature for ad hoc groups," *Information Security and Privacy*, vol. 3108, pp. 325–335, 2004.

[8] M. Franklin and H. Zhang, "A framework for unique ring signatures," *Financial Cryptography and Data Security*, vol. 7859, pp. 162–170, 2012.

[9] L. Deng, Y. Jiang, and B. Ning, "Identity-based linkable ring signature scheme," *IEEE Access*, vol. 7, pp. 153969–153976, 2019.

[10] S.-F. Sun, M. H. Au, J. K. Liu, and T. H. Yuen, "RingCT 2.0: a compact accumulator-based (linkable ring signature) protocol for blockchain cryptocurrency Monero," *Computer Security - ESORICS*, vol. 10493, pp. 456–474, 2017.

[11] M. Blum, P. Feldman, and S. Micali, "Non-interactive zero-knowledge and its applications," in *Proceedings of the twentieth annual ACM symposium on Theory of computing*, pp. 103–112, ACM, New York, NY, United States, Jan 1988.

[12] J. Groth, "Non-interactive zero-knowledge arguments for voting," in *Proceedings of the Applied Cryptography and Network Security, Third International Conference*, pp. 7–10, Springer, UCLA, USA, June 2005.

[13] C. Gentry, J. Groth, Y. Ishai, C. Peikert, A. Sahai, and A. Smith, "Using fully homomorphic hybrid encryption to minimize non-interative zero-knowledge proofs," *Journal of Cryptology*, vol. 28, no. 4, pp. 820–843, 2015.

[14] Y. C. Tsai, R. Tso, and Z. Y. Liu, "An improved non-interactive zero-knowledge range proof for decentralized applications," in *Proceedings of the 2019 IEEE International Conference on Decentralized Applications and Infrastructures (DAPPCON)*, pp. 129–134, IEEE, Newark, CA, USA, April 2019.

[15] A. Juels and M. Sudan, "A fuzzy vault scheme," in *Proceedings of the IEEE International Symposium on Information Theory*, p. 408, IEEE, Lausanne, Switzerland, July 2002.

[16] M. Osadchy and O. Dunkelman, "It is all in the system's parameters: privacy and security issues in transforming biometric raw data into binary strings," *IEEE Transactions on Dependable and Secure Computing*, vol. 16, no. 5, pp. 796–804, 2019.

[17] H. D. Dung, S. Willy, and T. H. T. Nguyen, "A multivariate blind ring signature scheme," *The Computer Journal*, vol. 8, pp. 1194–1202, 2019.

[18] H. Q. Le, D. H. Duong, and W. Susilo, "A blind ring signature based on the short integer solution problem," *Information Security Applications*, vol. 11897, pp. 92–111, 2020.

[19] Z. Jin, J. Y. Hwang, Y.-L. Lai, S. Kim, and A. B. J. Teoh, "Ranking-based locality sensitive hashing-enabled cancelable biometrics: index-of-max hashing," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 2, pp. 393–407, 2018.

[20] S. Agrawal, C. Ganesh, and P. Mohassel, "Non-interactive zero-knowledge proofs for composite statements," *Lecture Notes in Computer Science*, vol. 10993, pp. 643–673, 2018.

[21] J. Kurihara, S. Kiyomoto, K. Fukushima, and T. Tanaka, "A new (k,n)-Threshold secret sharing scheme and its extension," *Information Security*, vol. 5222, 2012.

[22] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612-613, 1979.

[23] J. Stanislaw, K. Hugo, S. Maliheh, and S. Nitesh, "Two-factor Authentication with end-to-end password security," *PKC*, vol. 10770, pp. 431–461, 2018.

[24] S. Qiu, D. Wang, G. Xu, and S. Kumari, "Practical and provably secure three-factor Authentication protocol based on extended chaotic-maps for mobile lightweight devices," *IEEE Transactions on Dependable and Secure Computing*, 2020.

[25] D. Wang and P. Wang, "Two birds with one stone: two-factor Authentication with security beyond conventional bound," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 4, pp. 708–722, 2016.

[26] Q. Jiang, N. Zhang, J. Ni, J. Ma, X. Ma, and K.-K. R. Choo, "Unified biometric privacy preserving three-factor Authentication and key agreement for cloud-assisted autonomous vehicles," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 9, pp. 9390–9401, 2020.