An Efficient Heterogeneous Ring Signature Scheme

Caixue Zhou
(Corresponding author: Caixue Zhou)
School of Computer and Big Data Science, Jiujiang University
551 Qianjin Donglu, Jiujiang 332005, China
Email: charlesjjjx@126.com
(Received Jan. 18, 2022; Revised and Accepted May 21, 2022; First Online July 3, 2022)

Abstract

A ring signature allows one to sign a message anonymously. However, each of the previously proposed ring signature schemes was constructed under one single type of environment, like the traditional public-key cryptosystem (T-PKC) environment, the identity-based public-key cryptosystem (IB-PKC) environment, or the certificateless public-key cryptosystem (CL-PKC) environment. This paper introduces a heterogeneous ring signature scheme, which allows n users to form a ring under mixed public key environments. In our scheme, we let the T-PKC environment and IB-PKC environment users form a ring and produce a ring signature using bilinear pairings. We give a formal definition and a security model of such a heterogeneous ring signature. Our scheme is proven existentially unforgeable under the computational Diffie-Hellman (CDH) assumption in the random oracle model and can achieve unconditional anonymity. Moreover, the number of pairing computations in the scheme is constant, making it one of the highly efficient schemes compared with other pairing-based ring signature schemes.

Keywords: Bilinear Pairing; Heterogeneous Ring Signature; Insider Corruption Attacks

1 Introduction

The traditional public key cryptosystem is built on the public key infrastructure, in which the private key is produced first, followed by the generation of the corresponding public key. This method of producing the key pair makes the public key a random string. There is no relationship between the public key and the user's identity. Therefore, it needs a trusted third party, i.e., a certificate authority, to issue a certificate to bind the public key with the user's identity. The cost of certificate management is considered to be very high in T-PKC.

The identity-based public key cryptosystem [7] can simplify the public key management. In IB-PKC, the public key is produced first, followed by the generation of the private key. In this way, the public key can be selected according to the user's identity, such as his/her e-mail address or telephone number, etc. Obviously, the cost of the public key management is greatly reduced, because the certificate issuing part has been omitted. The drawback of IB-PKC is that it needs a trusted third party, i.e., a private key generator (PKG), to produce any user's private key, which inevitably brings about the key escrow problem. It is only suitable for the applications where the PKG can be absolutely trusted.

Ring signature [19] is a group-oriented signature scheme where the actual signer spontaneously conscripts other n-1 persons to form a ring, and to generate a signature on behalf of the ring using his private key and others' public keys. The verifier can be convinced that the signature is made by someone in the ring but does not know which one this person is. It is impossible to find out whether two signatures are issued by the same signer, and the other n-1 persons are even unaware that they are included in the ring. Unlike the group signature [2, 10], there is no group manager, setup procedure or coordination in the ring signature. The ring is ad hoc, and the anonymity is irrevocable. Thus, ring signatures can realize signer ambiguity.

Since its introduction, ring signatures have been found in other valuable applications, such as blockchain privacy protection [11], e-voting [17] and anonymous cryptocurrency transaction [15]. In recent years, researchers have mainly focused on the study of ring signatures that are secure against quantum computer attacks, for example, the lattice-based ring signature [9].

A ring signature scheme involves n users' public keys. In practice, users' public keys may be generated independently without any coordination with others, and it is very likely that these public keys are from different signature schemes, such as RSA scheme or Schnorr scheme. In this situation, all above mentioned ring signature schemes are not suitable. In 2002, Abe *et al.* [1] proposed a ring signature scheme that allows mixed use of different flavors of keys at the same time. Their constructions can

use RSA-type keys and DL-type keys to form a ring. In 2003, Liu *et al.* [12] proposed the corresponding threshold ring signature scheme which supports the mix of public keys for any trapdoor one-way type and three-move type signature schemes. In this paper, we carried out a further study on cases where n public keys in a ring can be derived from different environments, such as T-PKC environment and IB-PKC environment. Due to the diversity of the world, we cannot force everyone to use the same public key cryptosystem. In an organization, some people may use T-PKC, while others IB-PKC or CL-PKC. In this situation, schemes [1, 12] are also not suitable, as they are both under T-PKC. So can we conscript n persons under mixed public key environments to form a ring and produce a ring signature? In this paper, we give an affirmative answer and we call this type of ring signature the "heterogeneous ring signature". We give a concrete heterogeneous ring signature scheme where some users are under the T-PKC environment while others, under the IB-PKC environment. If there is zero T-PKC user, our scheme will be reduced to an IB-PKC environment ring signature; if there is zero IB-PKC user, it will be reduced to a T-PKC environment ring signature; and if both of these users exist, it will become a heterogeneous ring signature. We leave the scheme of combining CL-PKC environment with other public key environments to our next-step research work, and do not consider it in this paper.

We give a formal definition and a security model of this kind of heterogeneous ring signature and propose a concrete scheme. Our scheme has the merit of constant pairing computations, while in many others in the literature, the number of bilinear pairings often grows linearly with the ring size n. Then, we prove that our scheme is unforgeable in the random oracle model under the CDH assumption, and that the signer ambiguity satisfies unconditional anonymity. At last, we give an efficiency evaluation on our scheme, which shows it is of high efficiency.

The rest of the paper is organized as follows. In Section 2, we introduce the concept of bilinear pairing, the CDH assumption, and the algorithm constitution and the security model of the heterogeneous ring signature. In Section 3, we propose an efficient heterogeneous ring signature scheme in the random oracle model. In Section 4, we discuss the security and efficiency of the proposed scheme. In Section 5, we give an application example of heterogeneous ring signatures. We conclude the paper in Section 6.

2 Prelimilaries

2.1 Bilinear Pairing

Let $(G_1, +)$ and $(G_2, \cdot)$ be two cyclic groups of prime order q, and g be a generator of G_1. The map $e : G_1 \times G_1 \to G_2$ is said to be an admissible bilinear pairing if the following three conditions hold.

Bilinearity. for all $a, b \in Z_q$, $P, Q \in G_1$, we have $e(P^a, Q^b) = e(P, Q)^{ab}$.

Non-degeneracy. $e(g, g) \neq 1_{G_2}$.

Computability. for all $P, Q \in G_1$, there exists an efficient algorithm to compute $e(P, Q)$.

2.2 Complexity Assumption

Computational Diffie-Hellman (CDH) Problem: Given $(P, aP, bP) \in G_1^3$ for unknown $a, b \in Z_q$, one must compute abP.

The advantage of any probabilistic polynomial time (PPT) algorithm A in solving the CDH problem in G_1 is defined as:

$ADV_A^{CDH} = Pr[A(P, aP, bP) = abP, a, b \in Z_q]$

CDH assumption: for every PPT algorithm A, ADV_A^{CDH} is negligible.

2.3 Definition of Heterogeneous Ring Signature

In this paper, we mainly focus on the ring signatures under the T-PKC environment combined with the IB-PKC environment. For the sake of simplicity, we refer to this type of heterogeneous ring signature as "heterogeneous ring signature" in the following context.

A heterogeneous ring signature scheme consists of the following five algorithms:

$Setup(1^k)$**:** Given a security parameter 1^k, it generates common public parameters $Params$. The PKG in the IB-PKC environment generates a master secret key s and a master public key P_{pub}.

$TPKC - Key - Gen(Params, ID)$**:** On input $Params$, it generates a public/private key pair (PK,SK). This algorithm is run by the user in the T-PKC environment.

$IBPKC - Key - Gen(Params, P_{pub}, s, ID)$**:** On input $Params$, the master public key P_{pub}, the master secret key s and a user's identity ID, it generates a private key D_{ID} for the user ID in the IB-PKC environment. This algorithm is run by the PKG, and the PKG sends D_{ID} to the user securely.

$RSign(Params, m, SK_s \ or \ D_{ID_s}, R)$**:** On input $Params$, the message m, a set of $R = (PK_1, PK_2, ..., PK_{n_1}, ID_1, ID_2, ..., ID_{n_2})$ $(n_1 + n_2 = n)$, and the secret key of one member SK_s $(s \in \{1, 2, ..., n_1\})$ or D_{ID_s} $(s \in \{1, 2, ..., n_2\})$, it generates a ring signature on (m, R).

$RVerify(Params, m, \sigma, R)$**:** On inputs $Params$, the message m, the ring signature σ, and a ring R, the receiver verifies the validity of the ring signature σ. If it is valid, the ring signature is accepted.

For consistency, we require if $\sigma = RSign(Params, m, D_{ID_s}$ or $SK_s, R)$, it must have $RVerify(Params, m, \sigma, R) = true$.

2.4 Security Definitions of Heterogeneous Ring Signature

The security of a heterogeneous ring signature scheme must satisfy both signer ambiguity and unforgeability as a normal ring signature. Informally, the signer ambiguity means no one can identify which signing key is used in the ring signature. This anonymity can be either computational or unconditional. In this paper, we mainly focus on the unconditional anonymity. Unforgeability is the natural extension of the existential unforgeability against adaptive chosen message attack in an ordinary signature scheme. In 2006, Bender *et al.* [3] introduced a strongest security model for T-PKC environment ring signatures by considering insider corruption attacks. In this paper, we modify it to fit our heterogeneous ring signature scheme.

Definition 1. *(Unforgeability) A heterogeneous ring signature scheme is existentially unforgeable against adaptive chosen message attacks (EUF-CMA for short) if no PPT adversary A has a non-negligible advantage in the following game:*

Setup: *Given a security parameter 1^k, the challenger C runs the setup algorithm to generate common public parameters Params. Then he/she runs the $TPKC-Key-Gen$ algorithm to generate $\{PK_i, SK_i\}_{i=1}^{p(k)}$ key pairs, where $p(\cdot)$ represents a polynomial. Then C produces a master secret key s and computes the master public key P_{pub} for the IB-PKC environment users. C gives Params, P_{pub} and the set $S \stackrel{def}{=} \{PK_i\}_{i=1}^{p(k)}$ to A while keeping s and $\{SK_i\}_{i=1}^{p(k)}$ secret.*

Attack: *A can make the following polynomially bounded number of queries adaptively.*

1) *$TPKC-Corrupt$ Queries: A submits an index i $(1 \le i \le p(k))$, and C returns SK_i to A. Let C_1 represent the set of corrupted users.*
2) *$IBPKC-Corrupt$ Queries: A submits an identity ID to query for its secret key. C runs the $IBPKC-Key-Gen$ algorithm to produce a D_{ID} and returns it to A.*
3) *RSign Queries: A produces a message m, and a set of n users' $R = (PK_1, PK_2, ..., PK_{n_1}, ID_1, ID_2, ..., ID_{n_2})$ $(n_1 + n_2 = n)$. C randomly selects an index s from R, and then runs the RSign algorithm to produce a ring signature σ to A. Let $R_1 = (PK_1, PK_2, ..., PK_{n_1})$. We do not require $R_1 \subseteq S$, which means R_1 may contain some adversarially-generated public keys.*

Forgery: *The attacker A outputs a forged ring signature σ^* on message m^*, and n^* $(n^* = n_1^* + n_2^*)$ users' ring $R^* = (PK_1^*, PK_2^*, ..., PK_{n_1^*}^*, ID_1^*, ID_2^*, ..., ID_{n_2^*}^*)$. Let $R_1^* = (PK_1^*, PK_2^*, ..., PK_{n_1^*}^*)$ and $R_2^* = (ID_1^*, ID_2^*, ..., ID_{n_2^*}^*)$. The restrictions are that A must not have made the RSign query of (m^*, R^*, σ^*) before, $R_1^* \subseteq S \backslash C_1$, and that he/she does not ask $IBPKC-Corrupt$ queries of each ID_i in R_2^*. A wins the game if $RVerify(m^*, \sigma^*, R^*) = true$. A's advantage is its probability of victory.*

Note 1. *If we change the restriction of adversary A to that m has never been queried to the ring signature oracle, the scheme will suffer from group-changing attacks and multiple-known-signature existential forgery attacks. If the condition turns into one where (m, R) has never been queried to the ring signature oracle, the scheme will suffer from multiple-known-signature existential forgery attacks. For a more detailed description, please refer to [13].*

Definition 2. *(Signer Ambiguity) A heterogeneous ring signature scheme has unconditional signer ambiguity if for any ring signature $\sigma = RSign(Params, m, SK_s$ or $D_{ID_s}, R)$, where $s \in \{1, 2, ..., n\}$, any verifier A even with unbounded computing resources, cannot identify the actual signer with a probability better than a random guess. In other words, A can only output the actual signer ID_s with a probability no better than $1/n$ $(1/(n-1)$ if A is in the signers set).*

3 An Efficient Heterogeneous Ring Signature Scheme

3.1 Concrete Scheme

Setup. Given a security parameter 1^k, the PKG selects two cyclic groups $(G_1, +)$ and $(G_2, \cdot)$ of prime order q, a generator P of G_1, a bilinear map $e : G_1 \times G_1 \to G_2$, and two hash functions: $H_1 : \{0,1\}^* \to G_1$ and $H_2 : \{0,1\}^* \to Z_q^*$. Then the PKG randomly selects $s \in Z_q^*$ as the master secret key, and sets $P_{pub} = sP$ as the master public key. The system public parameters are $\{e, G_1, G_2, q, P, P_{pub}, H_1, H_2\}$.

TPKC-Key-Gen. A T-PKC environment user randomly selects $x \in Z_q^*$ as his/her private key, and computes the corresponding public key as $PK = xP$.

IBPKC-Key-Gen. Let ID_u be a user's identity. The PKG computes his/her private key as $D_{ID_u} = s \cdot Q_{ID_u}$, where $Q_{ID_u} = H_1(ID_u)$.

RSign. Let $R = \{PK_1, PK_2, ..., PK_{n_1}, Q_{ID_1}, Q_{ID_2}, \cdots, Q_{ID_{n_2}}\}$ $(n_1 + n_2 = n)$.

1) If the actual signer belongs to the T-PKC environment, let his/her private key be x_s, where $1 \le s \le n_1$ and the message $M \in$

$\{0,1\}^*$, and he/she produces the ring signature $\sigma = (U_{11}, U_{12}, ..., U_{1n_1}, U_{21}, U_{22}, ..., U_{2n_2}, V)$ as follows:

a. Randomly chooses $U_{1i} \in G_1$ for $i \in \{1, ..., n_1\}\backslash\{s\}$, $U_{2j} \in G_1$ for $j \in \{1, ..., n_2\}$ and $r \in Z_q^*$.

b. Computes $h_{1i} = H_2(U_{1i}, M, R, P_{pub})$ for $i \in \{1, ..., n_1\}\backslash\{s\}$ and $h_{2j} = H_2(U_{2j}, M, R, P_{pub})$ for $j \in \{1, ..., n_2\}$.

c. Computes $U_{1s} = rP - \sum_{i=1, i\neq s}^{n_1}(h_{1i} \cdot PK_i + U_{1i}) - \sum_{j=1}^{n_2}(h_{2j}Q_{ID_j} + U_{2j})$, $h_{1s} = H_2(U_{1s}, M, R, P_{pub})$ and $V = (r + h_{1s}x_s)P_{pub}$.

2) If the actual signer belongs to the IB-PKC environment, let his/her private key be D_{ID_s}, where $1 \leq s \leq n_2$ and the message $M \in \{0,1\}^*$, and he/she produces the ring signature $\sigma = (U_{11}, U_{12}, ..., U_{1n_1}, U_{21}, U_{22}, ..., U_{2n_2}, V)$ as follows:

a. Randomly chooses $U_{1i} \in G_1$ for $i \in \{1, ..., n_1\}$, $U_{2j} \in G_1$ for $j \in \{1, ..., n_2\}\backslash\{s\}$ and $r \in Z_q^*$.

b. Computes $h_{1i} = H_2(U_{1i}, M, R, P_{pub})$ for $i \in \{1, ..., n_1\}$ and $h_{2j} = H_2(U_{2j}, M, R, P_{pub})$ for $j \in \{1, ..., n_2\}\backslash\{s\}$.

c. Computes $U_{2s} = rQ_{ID_s} - \sum_{i=1}^{n_1}(h_{1i} \cdot PK_i + U_{1i}) - \sum_{j=1, j\neq s}^{n_2}(h_{2j}Q_{ID_j} + U_{2j})$, $h_{2s} = H_2(U_{2s}, M, R, P_{pub})$ and $V = (r + h_{2s})D_{ID_s}$.

RVerify. Given a ring signature $\sigma = (U_{11}, U_{12}, ..., U_{1n_1}, U_{21}, U_{22}, ..., U_{2n_2}, V)$ on (M, R). The verifier does the following.

1) Computes $h_{1i} = H_2(U_{1i}, M, R, P_{pub})$ for $i \in \{1, ..., n_1\}$ and $h_{2j} = H_2(U_{2j}, M, R, P_{pub})$ for $j \in \{1, ..., n_2\}$.

2) Verifies whether $e(P, V) = e(P_{pub}, \sum_{i=1}^{n_1}(h_{1i} \cdot PK_i + U_{1i}) + \sum_{j=1}^{n_2}(h_{2j}Q_{ID_j} + U_{2j}))$.

Note 2. *If $n_1 = 0$ in the above ring signature, it will become an identity-based ring signature scheme. If $n_2 = 0$, it will become a T-PKC environment ring signature scheme. In the latter case, the P_{pub} in the RSign algorithm can be selected by the actual signer as a random element of G_1.*

3.2 Correctness

1) In Case 1 of *RSign* algorithm, $e(P, V) = e(P, (r + h_{1s}x_s)P_{pub})$

$= e(P_{pub}, (r + h_{1s}x_s)P)$

$= e(P_{pub}, U_{1s} + \sum_{i=1, i\neq s}^{n_1}(h_{1i} \cdot PK_i + U_{1i}) + \sum_{j=1}^{n_2}(h_{2j}Q_{ID_j} + U_{2j}) + h_{1s} \cdot PK_s)$

$= e(P_{pub}, \sum_{i=1}^{n_1}(h_{1i} \cdot PK_i + U_{1i}) + \sum_{j=1}^{n_2}(h_{2j}Q_{ID_j} + U_{2j}))$

2) In Case 2 of *RSign* algorithm, $e(P, V) = e(P, (r + h_{2s})D_{ID_s})$

$= e(P_{pub}, (r + h_{2s})Q_{ID_s})$

$= e(P_{pub}, U_{2s} + \sum_{i=1}^{n_1}(h_{1i} \cdot PK_i + U_{1i}) + \sum_{j=1, j\neq s}^{n_2}(h_{2j}Q_{ID_j} + U_{2j}) + h_{2s} \cdot Q_{ID_s})$

$= e(P_{pub}, \sum_{i=1}^{n_1}(h_{1i} \cdot PK_i + U_{1i}) + \sum_{j=1}^{n_2}(h_{2j}Q_{ID_j} + U_{2j}))$

4 Analysis of the Proposed Scheme

4.1 Unforgeability

Theorem 1. *In the random oracle model, if there is a PPT attacker A who acts as a user in the T-PKC environment to win the game of Definition 1 with a non-negligible probability $\varepsilon \geq 7V_{q_{H_2},n}/2^k$ ($V_{q_{H_2},n} = q_{H_2}(q_{H_2} - 1) \cdot ... \cdot (q_{H_2} - n + 1)$) by making a valid ring signature of group size n, in polynomial time T, asking at most q_s ring signature queries, q_{H_1} H_1 queries, q_{H_2} H_2 queries, q_{E_1} $TPKC - Corrupt$ queries, q_{E_2} $IBPKC - Corrupt$ queries, CDH problem can be solved with probability $\varepsilon' \geq (\varepsilon^2/66V_{q_{H_2},n}) \cdot (1 - 1/p(k)) \cdot 1/(p(k) - q_{E_1})$ in time $T' \leq 2T + (2n + 1)t_m + 2t_p$, where $p(k)$ is the total number of public keys generated by the $TPKC - Key - Gen$ algorithm, and t_m and t_p represent the time for a scalar multiplication on G_1 and a pairing operation respectively.*

Proof. Suppose B is given $(P, aP, bP) \in G_1^3$ for random $a, b \in Z_q$. B does not know the values of a and b, and is asked to compute the value of abP. To utilize attacker A, challenger B will simulate the ring signature oracle, $TPKC - Corrupt$ oracle, $IBPKC - Corrupt$ oracle, H_1 oracle and H_2 oracle to provide responses to A's queries. B maintains two lists L_1 and L_2, which are initially empty.

Let $p(\cdot)$ be a polynomial. B randomly selects $s \in \{1, 2, ..., p(k)\}$ and sets $PK_s = aP$. Then B runs the $TPKC - Key - Gen$ algorithm to produce key pairs $\{PK_i, SK_i\}$ ($i \in \{1, 2, ..., p(k)\}\backslash\{s\}$). Let $S = \{PK_i\}_{i=1}^{p(k)}$. B gives A the system parameters with $P_{pub} = bP$ and $S = \{PK_i\}_{i=1}^{p(k)}$. We assume A will ask for $H_1(ID)$ before ID is used in any other queries.

H_1 queries: A supplies an identity ID. B checks list L_1. If an item for that query is found, the same answer will be given to A; otherwise, B randomly selects $x \in Z_q^*$

and repeats the process until x is not in list L_1. B returns xP to A, and stores (ID, x) in list L_1.

H_2 queries: A supplies an item (U, M, R, P_{pub}). B checks list L_2. If an item for that query is found, the same answer will be given to A; otherwise, B randomly selects $h_2 \in Z_q^*$ and repeats the process until h_2 is not in list L_2. B stores the item (U, m, R, P_{pub}, h_2) in list L_2, and returns h_2 to A.

$TPKC-Corrupt$ queries: A submits an index i ($1 \leq i \leq p(k)$). If $i = s$, B outputs failure and aborts; or else B returns SK_i to A.

$IBPKC-Corrupt$ queries: A supplies an identity ID. B returns $x(bP)$ to A.

$RSign$ queries: A supplies a message M and a set of n users' $R = (PK_1, PK_2, \dots, PK_{n_1}, ID_1, ID_2, ..., ID_{n_2})$ $(n_1 + n_2 = n)$. Let $R_1 = (PK_1, PK_2, ..., PK_{n_1})$. B randomly selects an index $t \in \{1, 2, ..., n_1\}$.

$PK_t \neq PK_s$. B produces the ring signature σ as normal because B knows the private key of SK_t.

$PK_t = PK_s$. B produces the ring signature σ as follows.

1) Randomly selects $U_{1i} \in G_1$ for $i \in \{1, ..., n_1\}\backslash\{t\}$, and $U_{2j} \in G_1$ for $j \in \{1, ..., n_2\}$.
2) Computes $h_{1i} = H_2(U_{1i}, M, R, P_{pub})$ for $i \in \{1, ..., n_1\}\backslash\{t\}$ and $h_{2j} = H_2(U_{2j}, M, R, P_{pub})$ for $j \in \{1, ..., n_2\}$.
3) Randomly selects $h_{1t} \in Z_q^*$ and $z \in Z_q^*$, and computes $U_{1t} = zP - h_{1t} \cdot PK_t - \sum_{i=1, i\neq t}^{n_1} (h_{1i} \cdot PK_i + U_{1i}) - \sum_{j=1}^{n_2} (h_{2j} Q_{ID_j} + U_{2j})$.
4) Saves item $(U_{1t}, M, R, P_{pub}, h_{1t})$ to list L_2. If a collision occurs in list L_2, B repeats step (c).
5) Computes $V = z(bP)$ and outputs the signature $\sigma = (U_{11}, U_{12}, ..., U_{1n_1}, U_{21}, U_{22}, ..., U_{2n_2}, V)$.

At last, attacker A outputs a forged ring signature σ^* on message M^*, and n^* $(n^* = n_1^* + n_2^*)$ users' ring $R^* = (PK_1^*, PK_2^*, ..., PK_{n_1^*}^*, ID_1^*, ID_2^*, ..., ID_{n_2^*}^*)$. If the forged ring signature σ^* is valid and A does not violate the restrictions of Definition 1, according to Ring Forking Lemma [8], we can get two valid ring signatures $\sigma^* = (U_{11}, U_{12}, ..., U_{1n_1^*}, U_{21}, U_{22}, ..., U_{2n_2^*}, V^*)$ and $\sigma' = (U_{11}, U_{12}, ..., U_{1n_1^*}, U_{21}, U_{22}, ..., U_{2n_2^*}, V')$ such that $h_{1i}^* \neq h_{1i}'$ for some $i \in \{1, 2, ..., n_1^*\}$, $h_{1j}^* = h_{1j}'$ for all $j \in \{1, 2, ..., n_1^*\}\backslash\{i\}$ and $h_{2t}^* = h_{2t}'$ for all $t \in \{1, 2, ..., n_2^*\}$. If $i = s$, we can solve the CDH problem as follows: $V^* - V' = (h_{1s}^* - h_{1s}')abP$, so $abP = (V^* - V') \cdot (h_{1s}^* - h_{1s}')^{-1}$.

Now we assess the probability of success. In the $TPKC-Corrupt$ queries, the probability of A asking the private key of SK_s is $1/p(k)$. In the forgery stage, the probability of $i = s$ is $1/(p(k) - q_{E_1})$. Combined with the Ring Forking Lemma, the probability of B succeeding is $\varepsilon' \geq (\varepsilon^2/66V_{q_{H_2},n}) \cdot (1 - 1/p(k)) \cdot 1/(p(k) - q_{E_1})$.

The $RSign$ algorithm needs $n+1$ scalar multiplications on G_1, and the $RVerify$ algorithm needs n scalar multiplications on G_1 and 2 pairing operations. Combined with the Ring Forking Lemma, the running time for B is $T' \leq 2T + (2n+1)t_m + 2t_p$. □

Theorem 2. *In the random oracle model, if there is a PPT attacker A who acts as a user in the IB-PKC environment to win the game of Definition 1 with a non-negligible probability $\varepsilon \geq 7V_{q_{H_2},n}/2^k$ by making a valid ring signature of group size n, in polynomial time T, asking at most q_s ring signature queries, q_{H_1} H_1 queries, q_{H_2} H_2 queries, q_{E_1} $TPKC-Corrupt$ queries, q_{E_2} $IBPKC-Corrupt$ queries, CDH problem can be solved with probability $\varepsilon' \geq (\varepsilon^2/66V_{q_{H_2},n}) \cdot (1 - n_2^*/(q_{E_2} + n_2^*))^{(q_{E_2}+n_2^*)} \cdot (n_2^*/q_{E_2})^{n_2^*}$ in time $T' \leq 2T + (2n+1)t_m + 2t_p$, where n_2^* represents the number of IB-PKC environment users in a ring signature.*

Proof. Suppose B is given $(P, aP, bP) \in G_1^3$ for random $a, b \in Z_q$. B does not know the values of a and b, and is asked to compute the value of abP. To utilize attacker A, challenger B will simulate ring signature oracle, $TPKC-Corrupt$ oracle, $IBPKC-Corrupt$ oracle, H_1 oracle and H_2 oracle to provide responses to A's queries. B maintains two lists L_1 and L_2, which are initially empty.

Let $p(\cdot)$ be a polynomial. B runs the $TPKC-Key-Gen$ algorithm to produce key pairs $\{PK_i, SK_i\}$ $(i \in \{1, 2, ..., p(k)\})$. Let $S = \{PK_i\}_{i=1}^{p(k)}$. B gives A the system parameters with $P_{pub} = bP$ and $S = \{PK_i\}_{i=1}^{p(k)}$. We assume A will ask for $H_1(ID)$ before ID is used in any other queries.

H_1 queries: A supplies an identity ID. B checks list L_1. If an item for that query is found, the same answer will be given to A; otherwise, B randomly selects $x \in Z_q^*$ and repeats the process until x is not in list L_1. B then flips a coin $c \in \{0, 1\}$ that yields 0 with probability η and 1 with probability $1 - \eta$. (η will be determined later.) If $c = 0$ then B returns xP to A; or else if $c = 1$ then B returns $x(aP)$ to A. In either case, B stores (ID, x, c) in list L_1.

H_2 queries: A supplies an item (U, M, R, P_{pub}). B checks list L_2. If an item for that query is found, the same answer will be given to A; otherwise, B randomly selects $h_2 \in Z_q^*$ and repeats the process until h_2 is not in list L_2. B stores the item (U, m, R, P_{pub}, h_2) in list L_2, and returns h_2 to A.

$TPKC-Corrupt$ queries: A submits an index i ($1 \leq i \leq p(k)$). B returns SK_i to A.

$IBPKC-Corrupt$ queries: A supplies an identity ID. B searches ID in list L_1. If $c = 0$, then B returns $x(bP)$ to A; or else if $c = 1$ then B outputs failure and aborts.

$RSign$ queries: A supplies a message M and a set of n users' $R = (PK_1, PK_2, \dots, PK_{n_1}, ID_1, ID_2, ..., ID_{n_2})$ $(n_1 + n_2 = n)$. Let $R_2 = (ID_1, ID_2, ..., ID_{n_2})$. B randomly selects an index $t \in \{1, 2, ..., n_2\}$ and searches ID_t in list L_1 for the value of c.

$c = 0$. B produces the ring signature σ as normal because B knows the private key of ID_t.

$c = 1$. B produces the ring signature σ as follows.

1) Randomly selects $U_{1i} \in G_1$ for $i \in \{1, ..., n_1\}$ and $U_{2j} \in G_1$ for $j \in \{1, ..., n_2\}\backslash\{t\}$.
2) Computes $h_{1i} = H_2(U_{1i}, M, R, P_{pub})$ for $i \in \{1, ..., n_1\}$, and $h_{2j} = H_2(U_{2j}, M, R, P_{pub})$ for $j \in \{1, ..., n_2\}\backslash\{t\}$.
3) Randomly selects $h_{2t} \in Z_q^*$ and $z \in Z_q^*$, and computes $U_{2t} = zP - h_{2t} \cdot Q_{ID_t} - \sum_{i=1}^{n_1}(h_{1i} \cdot PK_i + U_{1i}) - \sum_{j=1,j\neq t}^{n_2}(h_{2j}Q_{ID_j} + U_{2j})$.
4) Saves item $(U_{2t}, M, R, P_{pub}, h_{2t})$ to list L_2. If a collision occurs in list L_2, B repeats Step (c).
5) Computes $V = z(bP)$ and outputs the signature $\sigma = (U_{11}, U_{12}, ..., U_{1n_1}, U_{21}, U_{22}, ..., U_{2n_2}, V)$.

At last, attacker A outputs a forged ring signature σ^* on message M^*, and n^* ($n^* = n_1^* + n_2^*$) users' ring $R^* = (PK_1^*, PK_2^*, ..., PK_{n_1^*}^*, ID_1^*, ID_2^*, ..., ID_{n_2^*}^*)$. If the forged ring signature σ^* is valid and A does not violate the restrictions of Definition 1, according to Ring Forking Lemma [8], we can get two valid ring signatures $\sigma^* = (U_{11}, U_{12}, ..., U_{1n_1^*}, U_{21}, U_{22}, ..., U_{2n_2^*}, V^*)$ and $\sigma' = (U_{11}, U_{12}, ..., U_{1n_1^*}, U_{21}, U_{22}, ..., U_{2n_2^*}, V')$ such that $h_{2i}^* \neq h_{2i}'$ for some $i \in \{1, 2, ..., n_2^*\}$, $h_{2j}^* = h_{2j}'$ for all $j \in \{1, 2, ..., n_2^*\}\backslash\{i\}$ and $h_{1t}^* = h_{1t}'$ for all $t \in \{1, 2, ..., n_1^*\}$.

If the corresponding $c = 1$ of ID_i in list L_1, we can solve the CDH problem as follows: $V^* - V' = (h_{2i}^* - h_{2i}')xabP$, so $abP = (V^* - V') \cdot (h_{2i}^* - h_{2i}')^{-1} \cdot x^{-1}$.

Now we assess the probability of success. In the $IBPKC - Corrupt$ queries, the probability that B does not fail is $\eta^{q_{E_2}}$. In the forgery stage, the probability of the corresponding $c = 1$ of ID_i in list L_1 is $(1-\eta)^{n_2^*}$. So the combined probability is $\eta^{q_{E_2}} \cdot (1-\eta)^{n_2^*}$. By differentiation, the maximum of $\eta = q_{E_2}/(q_{E_2} + n_2^*)$ and the maximum of $\eta^{q_{E_2}} \cdot (1-\eta)^{n_2^*} = (1 - n_2^*/(q_{E_2} + n_2^*))^{(q_{E_2}+n_2^*)} \cdot (n_2^*/q_{E_2})^{n_2^*}$. Combined with the Ring Forking Lemma, the probability of B succeeding is $\varepsilon' \geq (\varepsilon^2/66V_{q_{H_2},n}) \cdot (1 - n_2^*/(q_{E_2} + n_2^*))^{(q_{E_2}+n_2^*)} \cdot (n_2^*/q_{E_2})^{n_2^*}$.

The $RSign$ algorithm needs $n+1$ scalar multiplications on G_1, and the $RVerify$ algorithm needs n scalar multiplications on G_1 and 2 pairing operations. Combined with the Ring Forking Lemma, the running time for B is $T' \leq 2T + (2n+1)t_m + 2t_p$. □

4.2 Signer Ambiguity

Theorem 3. *Our scheme has the unconditional signer ambiguity.*

Proof. Our proof is divided into two cases.

1) For the $RSign$ algorithm of the T-PKC environment.

   Since $U_{1i} \in G_1$ for $i \in \{1, ..., n_1\}\backslash\{s\}$, $U_{2j} \in G_1$ for $j \in \{1, ..., n_2\}$, and r are randomly selected, $(U_{11}, U_{12}, ..., U_{1n_1}, U_{21}, U_{22}, ..., U_{2n_2})$ are uniformly distributed. It still has to consider whether $V = (r + h_{1s}x_s)P_{pub}$ leaks information about the actual signer. Let us consider the following equation:

$$e(P, V) = e(P, (r + h_{1s}x_s)P_{pub})$$
$$= e(P_{pub}, (r + h_{1s}x_s)P)$$
$$= e(P_{pub}, rP) \cdot e(P_{pub}, h_{1s} \cdot PK_s)$$
$$= e(P_{pub}, U_{1s} + \sum_{i=1,i\neq s}^{n_1}(h_{1i} \cdot PK_i + U_{1i}) + \sum_{j=1}^{n_2}(h_{2j}Q_{ID_j} + U_{2j})) \cdot e(P_{pub}, h_{1s} \cdot PK_s)$$

   It seems that an attacker can check whether PK_t is the actual signer by checking whether the following equation holds:

$$e(P_{pub}, U_{1t} + \sum_{i=1,i\neq t}^{n_1}(h_{1i} \cdot PK_i + U_{1i}) + \sum_{j=1}^{n_2}(h_{2j}Q_{ID_j} + U_{2j})) = \frac{e(P,V)}{e(P_{pub}, h_{1t} \cdot PK_t)}$$

   However, the equation holds not only when $t = s$, but also when $\forall t \in \{1, 2, ..., n_1\} \backslash\{s\}$, i.e., the signature is symmetric. Let us see the following:

$$e(P_{pub}, U_{1t} + \sum_{i=1,i\neq t}^{n_1}(h_{1i} \cdot PK_i + U_{1i}) + \sum_{j=1}^{n_2}(h_{2j}Q_{ID_j} + U_{2j}))$$
$$= e(P_{pub}, h_{1s} \cdot PK_s + U_{1s} - h_{1t} \cdot PK_t + \sum_{i=1,i\neq s}^{n_1}(h_{1i} \cdot PK_i + U_{1i}) + \sum_{j=1}^{n_2}(h_{2j}Q_{ID_j} + U_{2j}))$$
$$= e(P_{pub}, h_{1s} \cdot PK_s - h_{1t} \cdot PK_t + rP)$$
$$= \frac{e(P_{pub}, h_{1s} \cdot PK_s + rP)}{e(P_{pub}, h_{1t} \cdot PK_t)}$$
$$= \frac{e(P, (h_{1s} \cdot x_s + r) \cdot P_{pub})}{e(P_{pub}, h_{1t} \cdot PK_t)} = \frac{e(P,V)}{e(P_{pub}, h_{1t} \cdot PK_t)}$$

   Thus, V leaks no information about the actual signer. For any fixed message m and fixed ring R, $(U_{11}, U_{12}, ..., U_{1n_1}, U_{21}, U_{22}, ..., U_{2n_2}, V)$ are independent and uniformly distributed no matter who is the actual signer. An attacker has no advantage in identifying the actual signer over random guessing.

2) For the $RSign$ algorithm of the IB-PKC environment.

   The proof is similar to Case 1.

□

4.3 Comparison of Performance

We compare our scheme with other ring signature schemes that use bilinear pairings as shown in Table 1. Pa, SM, Ex and H represent a pairing computation, a scalar multiplication on G_1, an exponentiation on G_2 and a map-to-point hash computation, respectively. $|G_1|$, $|G_2|$ and $|q|$ represent the bit length of group G_1, G_2 and order q, respectively. From Table 1, we can see that the pairing computation of scheme [8] is linear with the ring size n in the $RSign$ and the $Rverify$ algorithms, and thus it needs

Table 1: Comparison of performance

Schemes	RSign	RVerify	Signature-Size
[4]	$(n+1)\cdot SM$	$2\cdot Pa+n\cdot SM$	$(n+1)\cdot\|G_1\|$
[5]	$4\cdot Pa+(n+1)\cdot SM+H$	$3\cdot Pa+n\cdot SM+Ex+H$	$\|G_1\|+\|G_2\|+n\cdot\|q\|$
[8]	$(n+1)\cdot Pa+n\cdot SM$	$2n\cdot Pa+n\cdot SM$	$\|G_1\|+n\cdot\|G_2\|+n\cdot\|q\|$
[16]	$14\cdot SM$	$6\cdot Pa+2\cdot SM$	$6\|G_1\|$
[18]	$(n+1)\cdot SM$	$2\cdot Pa+n\cdot SM$	$n\cdot\|G_1\|+\|G_2\|$
Ours	$(n+1)\cdot SM$	$2\cdot Pa+n\cdot SM$	$(n+1)\cdot\|G_1\|$

a huge amount of computation. The computation of our scheme is the same as those of schemes [4,18] and shorter than those of schemes [5,16]. In the aspect of signature size, scheme [8] has the longest one while scheme [16] has the shortest one.

To provide more direct comparisons, we adopt the experimental results of the scheme [6] (a personal computer with an Intel I7-4510U 2.00 GHz CPU, an 8GB RAM and running Windows 10 operating system). The computation time of each operation is shown in Table 2. According to the scheme [6], the size of q is 512 bits, and if the technique of point compression is used, the size of an element on G1 or G2 is 512 bits, too (a type A pairing $e: G_1\times G_1\to G_2$ defined on the curve $y^2=x^3+x$ over the field F_q with the order r, where $q=3mod4$ is a 512-bit prime and r is a 160-bit prime factor of $q+1$). By combining Tables 1 and 2, and letting ring n=10, we get Table 3. From Table 3, we can conclude that our scheme is one of the highly efficient schemes.

Table 2: Computational time (ms)

Pa	SM	Ex	H
15.500	7.736	0.160	18.673

Table 3: Comparison of performance

Schemes	RSign (ms)	RVerify (ms)	Signature-Size (bits)
[4]	85.096	108.360	5632
[5]	165.769	142.693	6144
[8]	247.860	387.360	10752
[16]	108.304	108.472	3072
[18]	85.096	108.360	5632
Ours	85.096	108.360	5632

5 Application

Our heterogeneous ring signature scheme can be widely applied in the real world. Here we give a concrete example in the wireless body area network (WBAN) to show its superiority. In WBAN, multiple wearable or implanted intelligent physiological sensors will collect various vital signals from a patient in order to monitor his/her health status, and these collected signals will be transmitted wirelessly to a controller (mobile phone or PDA). Then the controller will transmit all this information to a remote health server, so that this information will be shared and processed by multiple doctors. The doctors should only obtain the bio-information of the patient without knowing any of the private information such as name and age, etc. Therefore, anonymity must be provided [14]. At the same time, authentication should also be guaranteed. A ring signature scheme is very suitable for this scenario as it can provide both anonymity and authentication at the same time. But ordinary ring signatures assume that all signers use the same cryptosystem, that is, they all use T-PKC, IB-PKC or CL-PKC, but not the mixed cryptosystems. Due to the dynamic feature of the patient group, they may use different public key cryptosystems. For example, some patients may use T-PKC, while others IB-PKC. Our heterogeneous ring signature scheme can adapt to this situation. It can be illustrated in Figure 1.

The actual signer spontaneously conscripts other n-1 persons to form a ring. In the ring, some patients use T-PKC, while others IB-PKC. The multiple wearable or implanted intelligent physiological sensors collect various vital signals of the patient, and transmit these collected signals to a mobile phone wirelessly. Then the mobile phone uses our heterogeneous ring signature scheme to transmit all this information to a remote health server so that the doctors can share and process this information.

6 Conclusions

In this paper, we introduce a kind of heterogeneous ring signature. It enables T-PKC and IB-PKC users to form a ring and produce a ring signature. We give a concrete scheme and corresponding formal definition and security model. Then we prove our scheme to be existentially unforgeable under the CDH assumption and able to achieve unconditional anonymity. Our scheme has the merit of constant number pairing computations. Compared with other ring signature schemes, it can be considered a highly efficient scheme. Future work is to design schemes for CL-PKC environment combined with other public key environment in the random oracle or in the standard model.

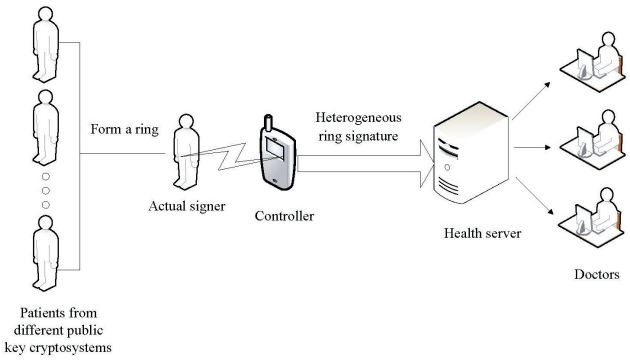

Figure 1: Application of heterogeneous ring signatures in WBAN

Acknowledgments

This study was supported by the Scientific and Technological Research Project of Jiangxi Provincial Education Department of China under Grant GJJ201807 and GJJ201808, the key Project of Jiangxi Provincial Natural Science Foundation of China under Grant 20202ACBL202005. The authors gratefully acknowledge the anonymous reviewers for their valuable comments. We would like to present our thanks to Ms. Yan Di, who checked our manuscript.

References

[1] M. Abe, M. Ohkubo, and K. Suzuki, "1-out-of-n signatures from a variety of keys," in *8th International Conference on the Theory and Application of Cryptology and Information Security (AsiaCrypt 2002)*, pp. 415–432, Queenstown, New Zealand, Dec. 2002.

[2] M. H. Abhilash and B. B. Amberker, "Revocable group signature scheme using ideal lattices," *International Journal of Electronics and Information Engineering*, vol. 13, no. 4, pp. 149–169, 2021.

[3] A. Bender, J. Katz, and R. Morselli, "Ring signatures: stronger definitions, and constructions without random oracles," in *Third Theory of Cryptography Conference (TCC 2006)*, pp. 60–79, New York, USA, Mar. 2006.

[4] S. S. M. Chow, S. M. Yiu, and L. C. K. Hui, "Efficient identity based ring signature," in *Third International Conference of Applied Cryptography and Network Security (ACNS 2005)*, pp. 499–512, New York, USA, Jun. 2005.

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

[6] D. B. He, N. Kumar, S. Zeadally, and H. Q. Wang, "Certificateless provable data possession scheme for cloud-based smart grid data management systems," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 3, pp. 1232–1241, 2018.

[7] J. Y. He, D. Zheng, R. Guo, Y. S. Chen, K. M. Li, and X. L. Tao, "Efficient identity-based proxy re-encryption scheme in blockchain-assisted decentralized storage system," *International Journal of Network Security*, vol. 23, no. 5, pp. 776–790, 2021.

[8] J. Herranz and G. Saez, "New identity-based ring signature schemes," in *6th International Conference of Information and Communications Security (ICICS 2004)*, pp. 27–39, Malaga, Spain, Oct. 2004.

[9] H. Q. Le, B. Vo, D. H. Duong, W. Susilo, N. T. Le, K. Fukushima, and S. Kiyomoto, "Identity-based linkable ring signatures from lattices," *IEEE Access*, vol. 9, pp. 84739–84755, 2021.

[10] C. C. Lee, T. Y. Chang, and M. S. Hwang, "A new group signature scheme based on the discrete logarithm," *Journal of Information Assurance and Security*, vol. 5, no. 1, pp. 54–57, 2010.

[11] X. F. Li, Y. R. Mei, J. Gong, F. Xiang, and Z. X. Sun, "A blockchain privacy protection scheme based on ring signature," *IEEE Access*, vol. 8, pp. 76765–76772, 2020.

[12] J. K. Liu, V. K. Wei, and D. S. Wong, "A separable threshold ring signature scheme," in *6th International Conference of Information Security and Cryptology (ICISC 2003)*, pp. 12–26, Seoul, Korea, Nov. 2003.

[13] J. K. Liu and D. S. Wong, "On the security models of (threshold) ring signature schemes," in *7th International Conference of Information Security and Cryptology (ICISC 2004)*, pp. 204–217, Seoul, Korea, Dec. 2004.

[14] J. W. Liu, Z. H. Zhang, X. F. Chen, and K. S. Kwak, "Certificateless remote anonymous authentication schemes for wireless body area networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 2, pp. 332–342, 2014.

[15] X. C. Mao, L. You, C. T. Cao, G. R. Hu, and L. Q. Hu, "Linkable ring signature scheme using biometric cryptosystem and nizk and its application," *Security and Communication Networks*, vol. 2021, 2021.

[16] X. Peng, K. Gu, Z. L. Liu, and W. B. Zhang, "Traceable identity-based ring signature for protecting mobile iot devices," in *6th International Conference of Data Mining and Big Data (DMBD 2021)*, pp. 158–166, Guangzhou, China, Oct. 2021.

[17] C. Qiu, S. B. Zhang, Y. Chang, X. Huang, and H. Chen, "Electronic voting scheme based on a quantum ring signature," *International Journal of Theoretical Physics*, vol. 60, no. 4, pp. 1550–1555, 2021.

[18] K. A. Shim, "An efficient ring signature scheme from pairings," *Information Sciences*, vol. 300, pp. 63–69, 2015.

[19] J. H. Zhang, W. L. Bai, and Z. T. Jiang, "On the security of a practical constant-size ring signature scheme," *International Journal of Network Security*, vol. 22, no. 3, pp. 394–398, 2020.

Biography

Caixue Zhou received BS degree in Computer Science Department from Fudan University in 1988, Shanghai, China and MS degree in Space College of Beijing University of Aeronautics and Astronautics in 1991, Beijing, China. He is a Professor at the School of Computer and Big Data Science, Jiujiang University, Jiujiang, China and a Supervisor of Postgraduate at the School of Information Technology, Jiangxi University of Finance and Economics, Nanchang, China. He is a member of the CCF (China Computer Federation) and a member of CACR(Chinese Association for Cryptologic Research). His research interests include applied cryptography and network security.