A Provably Secure Certificate Based Ring Signature Without Pairing

Zhiguang Qin[1], Hu Xiong[1,2,3], and Fagen Li[3]

(Corresponding author: Hu Xiong)

School of Computer Science and Engineering, University of Electronic Science and Technology of China[1]
No. 4, North Jianshe Road, Chenghua District, chengdu, Sichuan 610054, China
Key Lab of Network Security and Cryptology, Fujian Normal University[2]
No. 8, Shangsan Road, Cangshan District, Fuzhou, Fujian 350007, China
State Key Laboratory of Information Security, Institute of Software, Chinese Academy of Sciences[3]
No. 19 Yuquan Road, Shijingshan District, Beijing 100190, China
(Email: xionghu.uestc@gmail.com)

(Received Aug. 1, 2012; revised and accepted May 10 & July 5, 2013)

Abstract

In Eurocrypt 2003, Gentry introduced the notion of certificate-based encryption. The merit of certificate-based encryption lies in implicit certificate and no private key escrow. This feature is desirable especially for the efficiency and the real spontaneity of ring signature, which involve a large number of public keys in each execution. In this paper, we propose an efficient certificate-based ring signature scheme which does not require any pairing computation. Furthermore, this scheme is proven secure under the Discrete Logarithm assumption in the random oracle model. To the best of authors' knowledge, this is the first construction of certificate-based ring signature scheme in the literature that has such kind of feature.

Keywords: Certificate-based signature, provable Security, random Oracle, ring signature

1 Introduction

Ring signature, introduced by Rivest, Shamir and Tauman [21], is characterized by two main properties: anonymity and spontaneity. Anonymity in ring signature means 1-out-of-n signer verifiability, which enables the signer to keep anonymous in these "rings" of diverse signers. Spontaneity is a property which makes distinction between ring signatures and group signatures [7]. In group signature schemes, there exists a trusted third party (TTP), usually known as the group manager, who handles the joining of group members by interacting with them. In ring signature schemes, no such trusted party exists and the rest of the $n-1$ members in the ring are totally unaware that they have been included in the ring. These two properties make ring signatures widely applicable to various cryptographic schemes. Taking the example of concurrent signatures [8, 10] which is a partial solution to the fair exchange of signatures without TTPs, anonymity provides the signer-ambiguity of signatures and the spontaneity enables concurrent signature a solution without TTPs. The survey of ring signatures and related applications can be found in [9, 20, 24].

However, the theory of ring signature faces some problems when it comes to reality. The public key of user is usually a "random" string that is unrelated to the identity of the user in traditional public key infrastructure (PKI), so there is a trusted-by-all certificate authority (CA) to assure the relationship between the cryptographic keys and the user. As a result, any verifier of a signature must obtain and verify the user's certificate before checking the validity of the signature. The communication and the validation of a large number of public keys greatly affect the efficiency of the ring signature. Furthermore, the signer cannot spontaneously conscript users who have not registered for a certificate in traditional PKI.

ID-based cryptography which was introduced in 1984 by Shamir [22] solved these problems: the public key of each user is easily computable from a string corresponding to this user's identity, (such as an email address), while the private key associated with that identity is computed and issued secretly to the user by a trusted third party called private key generator (PKG). This property avoids the necessity of certificates, and associates an implicit public key to each person over the world. So ID-based ring signature has rapidly emerged in recent years and been well studied as well. Zhang and Kim proposed the first ID-based ring signature [27]. After that, Lin and Wu [17] gave a more efficient construction, while Awasthi and Lal [2] pointed out and fixed some small inconsistencies in [27] and [17]. An ID-based ring signature scheme for anonymous subsets (i.e. 1-out-of-n-groups instead of 1-out-of-n-individuals)was presented by Herranz and Germ*á*n [14]. Then, Chow and Yiu [11] achieved a constant number of pairing computations and Nguyen [19] also realized a constant size signature. Besides, various research work on its extensions [12, 26] and applications [23, 25] has already appeared. However, an inherent problem of ID-based cryptosystems is key escrow, i.e., the PKG knows users' private key. A malicious PKG can frame an innocent user by forging the user's signature. Due to this inherent problem, ID-based cryptosystems are considered to be suitable only for private networks [22]. Thus, eliminating key escrow in ID-based cryptosystems is essential to make them more applicable in the real world.

To integrate the merits of ID-PKC into traditional PKI, Gentry [13] proposed a paradigm called certificate-based encryption (CBE) in 2003. The CBE combines a traditional public key encryption scheme and an ID-based encryption scheme between a certifier and a user. In CBE, each user generates his own private and public keys and requests a certificate from the CA while the CA uses the key generation algorithm of an ID-based encryption scheme [5] to generate the certificate. The certificate is implicitly used as part of the user decryption key, which is composed of the user-generated private key and the certificate. Although the CA knows the certificate, it does not have the user private key. Thus it cannot decrypt any cipher texts. In addition to CBE, the notion of certificate-based signature (CBS) was first introduced by Kang *et al.* [15]. However, one of their proposed schemes was found insecure against key replacement attack, as pointed out by Li *et al.* [16]. They also proposed a concrete scheme. Furthermore, Au *et al.* [1] proposed the first certificate-based (linkable) ring signature (CBRS) scheme. Unfortunately, all of these CBS and CBRS schemes are derived from bilinear pairings, a powerful but computationally expensive primitive. Furthermore, the pairing has not yet been enjoyed the same exposure to cryptanalytic attacks by experts as other older problems from number theory such as discrete logarithms, factoring and RSA [4, 29]. And the implementations of pairings are much harder than these of exponentiation in a RSA group. Therefore, the construction of certificate-based ring signature scheme without pairing is of great interest in the field of cryptography.

Recently, Liu *et al.* [18] and Zhang *et al.* [28] proposed certificate-based signature scheme without pairings independently. However, to the best of author's knowledge, certificate-based *ring* signature without pairings has not been treated in the literature until now. Our current work is aimed at filling this void. An efficient certificate-based ring signature scheme without pairings is proposed in our paper. Meantime, the proposed scheme is proven to be existential unforgeable using Discrete Logarithm assumption under the random oracle model.

The rest of this paper is organized as follows. A brief review of some preliminaries required throughout the paper in Section 2. A concrete certificate-based ring signature without pairings is proposed in Section 3 and the security of our scheme is analyzed in Section 4. Finally, the conclusions are given in Section 5.

2 Preliminaries

In this section, we will review some fundamental backgrounds required in this paper, namely the certificate-based ring signature definition and complexity assumptions.

2.1 Mathematical Assumption

Definition 1. ***(Discrete Logarithm (DL) Assumption).*** *Given a group $\mathcal{G}$ of prime order q with generator g and elements $A \in \mathcal{G}$, the DL problem in $\mathcal{G}$ is to output $\alpha \in \mathbb{Z}_q$ such that $A = g^\alpha$.*

An adversary $\mathcal{B}$ has at least an ϵ advantage if $\Pr[\mathcal{B}(g, A) = \alpha | A = g^\alpha] \geq \epsilon$. We say that the (ϵ, t)-DL assumption holds in a group $\mathcal{G}$ if no algorithm running in time at most t can solve that DL problem in $\mathcal{G}$ with advantage at least ϵ.

2.2 The Concept of Certificate-based Ring Signature Schemes

A certificate-base ring signature scheme is a tuple (Setup, UserKeyGen, CertGen, Ring-Sign and Verify) of polynomial-time algorithm. The first four algorithms may be randomized but the last one is usually not.

1) Setup: On input 1^k where $k \in \mathbb{N}$ is a security parameter, it generates a master public/secret key pair (mpk, msk). It also outputs a public parameters params which is shared in the system.

2) UserKeyGen: On input the master public key mpk, the system parameter params, and the user identity $ID \in \{0,1\}^*$, it generates a user ID's secret/public key pair (usk_{ID}, upk_{ID}).

3) CertGen: On input master secret key msk, the system parameter params, the identity ID of a user and its public key upk_{ID}, it generates a certificate $cert_{ID}$.

4) Ring-Sign: This algorithm takes as input a message $m \in \{0,1\}^*$, a set of n group members whose identities form the set $L_{ID} = \{ID_1, \ldots, ID_n\}$ and their corresponding public keys form the set $L_{upk} = \{upk_{ID_1}, \ldots, upk_{ID_n}\}$, a public parameters params, a signer ID_s's secret key usk_{ID_s} and its certificate $cert_{ID_s}$ to generate a signature σ. Here ID_s is the s-th group member of L_{ID}.

5) Verify: On input mpk, the system parameter params, the set L_{ID} of the group members' identities and the set L_{upk} of the corresponding public keys of the group members, message m and signature σ, it returns 1 for accept or 0 for reject.

In practice, the CA could be the one who performs Setup and CertGen. The master public key mpk and the system parameter params will then be published and assumed that everyone in the system has gotten a legitimate copy of it. For each user in the system, the user is supposed to be able to carry out UserKeyGen, Ring-Sign, and Verify. Note that a certificate-base ring signature scheme should satisfy the obvious correctness conditions (that a honest signed ring signature should be verified as valid).

2.3 Security Models of Certificate-based Ring Signature Schemes

Combining the security model of certificate-based public key cryptography and security models of ring signature schemes in traditional PKI and ID-PKC, we define the security of certificate-based ring signature scheme as follows. There are two types of security for a certificate-based ring signature scheme, Type-I security and Type-II security, along with two types of adversaries, $\mathcal{A}_1$ and $\mathcal{A}_2$, respectively. Adversary $\mathcal{A}_1$ models a malicious adversary which compromises the user secret key usk_{ID} or replace the user public key upk_{ID}, however, cannot compromise the master secret key msk nor get access to the user certificate $cert_{ID}$[16]. Adversary $\mathcal{A}_2$ models the malicious-but-passive CA who controls the generation of the master public/secret key pair, and that of any user certificate $cert_{ID}$. We define two games, one for $\mathcal{A}_1$ and the other one for $\mathcal{A}_2$.

Game I: Unforgeability of Certificate-based Ring Signature Against Type I Adversary $\mathcal{A}_1$

Setup: Let $\mathcal{S}_1$ be the game simulator/challenger and $k \in \mathbb{N}$ be a security parameter. $\mathcal{S}_1$ first executes Setup(1^k) to get (mpk, msk) and the system parameters params. $\mathcal{S}_1$ then sends params and mpk to the adversary $\mathcal{A}_1$ while keeping the msk as secret. In addition, $\mathcal{S}_1$ will maintain three lists L_1, L_2 and L_3 where

1) L_1 is used to record the identities which have been chosen by $\mathcal{A}_1$ in the **CertGen** queries.

2) L_2 is used to record the identities whose public keys have been replaced by $\mathcal{A}_1$.

3) L_3 is used to record the identities which have been chosen by $\mathcal{A}_1$ in the **Corruption** queries.

All of these three lists are the empty set ϕ at the beginning of the game.

Query: During the simulation, $\mathcal{A}_1$ can adaptively make a polynomially bounded number of queries as defined below:

1) **UserKeyGen**: On input an identity $ID \in \{0,1\}^*$, if ID has already been created, nothing is to be carried out. Otherwise, $\mathcal{S}_1$ generates $(upk_{ID}, usk_{ID}) \leftarrow$ UserKeyGen$(mpk, ID,$ params$)$. In both cases, upk_{ID} is returned.

2) **CertGen**: $\mathcal{A}_1$ can request the certificate of any user whose identity is ID. In respond, $\mathcal{S}_1$ first resets $L_1 = L_1 \cup \{ID\}$, then runs the algorithm CertGen and outputs the certificate $cert_{ID}$ as answer.

3) **ReplaceKey**: On input an identity ID and a user public key upk^*, $\mathcal{S}_1$ searches L_2 for the entry of ID. If it is not found, nothing will be carried. Otherwise, $\mathcal{S}_1$ updates (ID, upk_{ID}) to (ID, upk^*_{ID}) and resets $L_2 = L_2 \cup \{ID\}$.

4) **Corruption**: $\mathcal{A}_1$ can request the secret key of any user whose identity is ID. In respond,

   a. $\mathcal{S}_1$ first checks the set L_2. If $ID \in L_2$ (that is, the public key of the user ID has been replaced), $\mathcal{S}_1$ will return the symbol $\perp$ which means $\mathcal{S}_1$ cannot output the secret key of an identity whose public key has been replaced.

   b. Otherwise, $ID \notin L_2$ and $\mathcal{S}_1$ resets $L_3 = L_3 \cup \{ID\}$. $\mathcal{S}_1$ then runs the algorithm UserKeyGen and outputs the secret key usk_{ID} as answer.

5) **Ring-Sign**: On input a message $m \in \{0,1\}^*$ on behalf of a group whose identities are listed in the set $L_{ID} = \{ID_1, \ldots, ID_n\}$ and the corresponding public keys are in the set $L_{upk} = \{upk_{ID_1}, \ldots, upk_{ID_n}\}$, $\mathcal{S}_1$ outputs a ring signature σ for the message m. It is required that the algorithm Verify will output 1 for the input $(mpk, L_{ID}, L_{upk}, m, \sigma)$.

Forgery: Finally, $\mathcal{A}_1$ is to output $(L^*_{ID}$, L^*_{upk}, m^*, $\sigma^*)$ as the forgery. We say that $\mathcal{A}_1$ wins if Verify $(mpk, L^*_{ID}$, L^*_{upk}, m^*, $\sigma^*) = 1$ for the oracle **Ring-Sign** has never been queried with $(L^*_{ID}$, L^*_{upk}, $m^*)$. One additional restriction is that $L^*_{ID} \cap L_1 \cap L_2 = \phi$ and $L^*_{ID} \cap L_3 = \phi$.

Game II: Unforgeability of Certificate-based Ring Signature Against Type II Adversary $\mathcal{A}_2$

Setup: Let $\mathcal{S}_2$ be the game challenger and $k \in \mathbb{N}$ be a security parameter. $\mathcal{S}_2$ executes $\mathcal{A}_2$ on input 1^k, which returns a master public/secret key pair (mpk, msk) to $\mathcal{A}_2$. Note that $\mathcal{A}_2$ cannot make any query at this stage. $\mathcal{S}_2$ will maintain two lists L_1, L_2 where

1) L_1 is used to record the identities whose public keys have been replaced by $\mathcal{A}_2$.

2) L_2 is used to record the identities which have been chosen by $\mathcal{A}_2$ in the **Corruption** queries.

Query: During this stage of simulation, $\mathcal{S}_2$ can make queries onto oracle **Corruption**, **ReplaceKey** and **Ring-Sign**. $\mathcal{A}_2$ can also make queries to **UserKeyGen**. Note that oracle **CertGen** is not accessible and no longer needed as $\mathcal{A}_2$ has the master secret key.

Forgery: At the end of **Game II**, $\mathcal{A}_2$ is to output a triple $(L^*_{ID}$, L^*_{upk}, m^*, $\sigma^*)$. $\mathcal{A}_2$ wins if Verify $(mpk, L^*_{ID}$, L^*_{upk}, m^*, $\sigma^*) = 1$ for the oracle **Sign** has never been queried with (L^*_{ID}, L^*_{upk}, m^*). One additional restriction is that $L^*_{ID} \cap L_1 = \phi$ and $L^*_{ID} \cap L_2 = \phi$.

Definition 2. *A certificate-based ring signature scheme is existentially unforgeable under adaptively chosen-message attack iff the success probability of any polynomially bounded adversary in the* ***Game I*** *and* ***Game II*** *is negligible.*

Definition 3. *A certificate-based ring signature scheme is said to have the unconditional signer ambiguity if for any group of n users whose identities form the set L_{ID} and their corresponding public keys form the set L_{upk}, any message m and any signature σ =Ring-Sign($L_{ID}, L_{upk}, cert_{ID_s}, usk_{ID_s}, m$); any verifier $\mathcal{V}$ even with unbounded computing resources, cannot identity the actual signer with probability better than a random guess. That is, $\mathcal{V}$ can only output the identity of the actual signer with probability no better than $\frac{1}{n}$ ($\frac{1}{n-1}$ if $\mathcal{V}$ is in the signers group).*

3 A Certificate-based Ring Signature Scheme without Pairing

In this section, we will give the concrete construction of a certificate-based ring signature scheme. In our scheme, we employ some ideas of the certificate-based signature scheme in [18], the ID-based ring signature scheme in [14]. Our certificate-based ring signature consists of the following algorithms:

1) Setup: Let $\mathcal{G}$ be a multiplicative group with order q. The CA selects a random generator $g \in \mathcal{G}$ and randomly chooses $x \in_R \mathbb{Z}^*_q$ as the master secret key. It sets $X = g^x$. Let $H : \{0,1\}^* \rightarrow \mathbb{Z}^*_q$ be a cryptographic hash function. The public parameters are given by params=$(\mathcal{G}, q, g, X, H)$.

2) UserKeyGen: User ID_i selects a secret value $u_i \in \mathbb{Z}^*_q$ as his secret key usk_{ID_i}, and computes his public key $upk_{ID_i} = (g^{u_i}, X^{u_i}, \pi_{u_i})$ where π_{u_i} is the following non-interactive proof-of-knowledge (PoK)[1]:

$$PK\{(u) : U_1 = g^{u_i} \wedge U_2 = X^{u_i}\}. \qquad (1)$$

3) CertGen: Let $\tilde{h}_i = H(upk_{ID_i}, ID_i)$ for user ID_i with public key upk_{ID_i} and binary string ID_i which is used to identify the user. To generate a certificate for user ID_i, the CA randomly chooses $r \in_R \mathbb{Z}^*_q$, computes $R = g^r$ and $k_i = r^{-1}(\tilde{h}_i - xR) \bmod q$. The certificate is (R, k_i). Note that a correctly generated certificate should satisfy the following equality:

$$R^{k_i} X^R = g^{\tilde{h}_i}. \qquad (2)$$

4) Ring-Sign: Suppose there is a group of n users whose identities form the set $L_{ID} = \{ID_1, \ldots, ID_n\}$, and their corresponding public keys form the set $L_{upk} = \{upk_{ID_1}, \ldots, upk_{ID_n}\}$. To sign a message $m \in \{0,1\}^*$ on behalf of the group, the actual signer, indexed by s using the secret key usk_{ID_s} and the certificate $cert_{ID_s}$, performs the following steps.

   a. For each $i \in \{1, \ldots, n\} \setminus \{s\}$, selects $y_i \in_R \mathbb{Z}^*_q$ uniformly at random and computes $Y_i = R^{-y_i}$.
   b. Compute $h_i = H(m \| L_{upk} \| L_{ID} \| Y_i)$ for $i \in \{1, \ldots, n\} \setminus \{s\}$.
   c. Choose $y_s \in_R \mathbb{Z}^*_q$, computes $Y_s = R^{-y_s} \prod_{i \neq s} (g^{u_i})^{h_i \tilde{h}_i} \prod_{i \neq s} (X^{u_i})^{-h_i R}$.
   d. Compute $h_s = H(m \| L_{upk} \| L_{ID} \| Y_s)$.
   e. Compute $z = (\sum_{i=1}^{n} y_i + h_s k_s u_s) \bmod q$.
   f. Output the ring signature on m as $\sigma = \{Y_1, \ldots, Y_n, z\}$.

5) Verify: To verify a ring signature $\sigma = \{Y_1, \ldots, Y_n, z\}$ on a message m with identities in L_{ID} and corresponding public keys in L_{upk}, the verifier performs the following steps.

   a. Check whether π_{u_i} is a valid PoK. If not, outputs $\perp$, Otherwise, run the next step.
   b. Compute $h_i = H(m \| L_{upk} \| L_{ID} \| Y_i)$ and $\tilde{h}_i = H(upk_{ID_i}, ID_i)$ for all $i \in \{1, \ldots, n\}$.
   c. Check whether
   $$\prod_{i=1}^{n} (g^{u_i})^{h_i \tilde{h}_i} \stackrel{?}{=} R^z Y_1 \cdot \ldots \cdot Y_n \prod_{i=1}^{n} (X^{u_i})^{h_i R} \qquad (3)$$
   d. Accept the ring signature as valid and outputs 1 if Equation (3) holds, otherwise, output 0.

4 Analysis of the Proposed Scheme

In this section, we will analyze our proposed scheme in detail.

4.1 Correctness

The correctness of the proposed scheme can be easily verified with the following:

$$\begin{aligned} & \prod_{i=1}^{n} (g^{u_i})^{h_i \tilde{h}_i} \\ = \;& g^{h_s u_s \tilde{h}_s} g^{-h_s u_s x R} \prod_{i \neq s} (g^{u_i})^{h_i \tilde{h}_i} g^{x u_s h_s R} \\ = \;& g^{r h_s u_s r^{-1}(\tilde{h}_s - xR)} \prod_{i \neq s} (g^{u_i})^{h_i \tilde{h}_i} g^{x u_s h_s R} \\ = \;& R^{h_s k_s u_s} \prod_{i \neq s} (g^{u_i})^{h_i \tilde{h}_i} X^{u_s h_s R} \end{aligned}$$

[1] We refer to [6] for a more comprehensive description of notation and implementation of PoK.

$$\begin{aligned} &= R^{\sum_{i=1}^{n} y_i + h_s k_s u_s} \prod_{i \neq s} R^{-y_i} \cdot R^{-y_s} \prod_{i \neq s} (g^{u_i})^{h_i \tilde{h}_i} \\ &\quad \prod_{i \neq s} (X^{u_i})^{-h_i R} \prod_{i=1}^{n} (X^{u_i})^{h_i R} \\ &= R^z Y_1 \cdot \ldots \cdot Y_n \sum_{i=1}^{n} (X^{u_i})^{h_i R} \qquad (4) \end{aligned}$$

4.2 Unconditional Anonymity

Let $\sigma = \{Y_1, \ldots, Y_n, z\}$ be a valid ring signature on a message m on behalf of a group of n members specified by identities in L_{ID} and public keys in L_{upk}. Since all the y_i, $i \in \{1, \ldots, n\} \setminus \{s\}$ are randomly generated, hence all Y_i, $i \in \{1, \ldots, n\} \setminus \{s\}$ are also uniformly distributed. The randomness of y_s chosen by the signer implies $Y_s = R^{-y_s} \prod_{i \neq s} (g^{u_i})^{h_i \tilde{h}_i} \prod_{i \neq s} (X^{u_i})^{-h_i R}$ is also uniformly distributed. So $(Y_1, \ldots, Y_n)$ in the signature reveals no information about the signer.

It remains to consider whether $z = \sum_{i=1}^{n} y_i + h_s k_s u_s$ leaks information about the actual signer. From the construction of z, it is obvious to see that $k_s u_s = h_s^{-1}(z - \sum_{i=1}^{n} y_i)$. To identify whether ID_s is the identity of the actual signer, the only way is to check $R^{k_s u_s} \overset{?}{=} R^{h_s^{-1}(z - \sum_{i=1}^{n} y_i)}$. If ID_s is the identity of the actual signer, it should hold $Y_s = R^{-y_s} \prod_{i \neq s} (g^{u_i})^{h_i \tilde{h}_i} \prod_{i \neq s} (X^{u_i})^{-h_i R}$. It remains to check

$$R^{k_s u_s} \overset{?}{=} (R^z Y_1 \cdot \ldots \cdot Y_n \prod_{i \neq s} (g^{-u_i})^{h_i \tilde{h}_i} \prod_{i \neq s} (X^{u_i})^{h_i R})^{h_s^{-1}} \qquad (5)$$

However, we have for each $j \in \{1, \ldots, n\}$

$$\begin{aligned} &(R^z Y_1 \cdot \ldots \cdot Y_n \prod_{i \neq j} (g^{-u_i})^{h_i \tilde{h}_i} \prod_{i \neq j} (X^{u_i})^{h_i R})^{h_j^{-1}} \\ = &(R^{\sum_{i=1}^{n} y_i + h_s k_s u_s} R^{-\sum_{i=1}^{n} y_i} \prod_{i \neq s} (g^{u_i})^{h_i \tilde{h}_i} \\ &\prod_{i \neq s} (X^{u_i})^{-h_i R} \prod_{i \neq s} (g^{-u_i})^{h_i \tilde{h}_i} \prod_{i \neq s} (X^{u_i})^{h_i R} \\ &g^{-u_s h_s \tilde{h}_s} X^{u_s h_s R} g^{u_j h_j \tilde{h}_j} X^{-u_j h_j R})^{h_j^{-1}} \\ = &(R^{h_s u_s r^{-1}(\tilde{h}_s - xR)} g^{-u_s h_s \tilde{h}_s} X^{u_s h_s R} \\ &g^{u_j h_j \tilde{h}_j} X^{-u_j h_j R})^{h_j^{-1}} \end{aligned}$$

$$\begin{aligned} &= (g^{u_j h_j \tilde{h}_j} X^{-u_j h_j R})^{h_j^{-1}} \\ &= g^{u_j \tilde{h}_j} X^{-u_j R} \\ &= g^{u_j \tilde{h}_j - x u_j R} \\ &= R^{k_j u_j} \qquad (6) \end{aligned}$$

where ID_s is the identity of the actual signer. The fact is that z in the signature does not leak any information about the identity of the actual signer. And hence, the unconditional ambiguity of our proposed ring signature is proved.

4.3 Unforgeability

Assuming that the DL assumption is hard, we now show that the unforgeability of our certificate-based ring signature.

Theorem 1 (Unforgeability against Game I Adversary). *If a probabilistic polynomial-time forger $\mathcal{A}_1$ has an advantage ε in forging a certificate-based ring signature in an attack modeled by **Game I** after running in time t and making q_h queries to hashing queries, q_u queries to the **UserKeyGen** request oracle, q_{cert} queries to the **CertGen** extraction oracle, q_r queries to the **Replace** extraction oracle, q_{cor} queries to the **Corruption** extraction oracle, and q_s queries to the **Ring-Sign** oracle, then the DL problem can be solved with probability $\varepsilon' = (1 - \frac{q_h(q_{cert} + nq_s)}{q})(1 - \frac{1}{q})(\frac{1}{q_h})\varepsilon$ with time $t' < 2(t + q_h T_h + q_u T_u + q_{cert} T_{cert} + q_{cor} T_{cor} + q_r T_r + q_s T_s)$, where T_h(resp. T_u, T_{cert}, T_{cor}, T_s and T_r) is the time cost of an Hash query (resp. **UserKeyGen**, **CertGen**, **Corruption**, **Ring-Sign** and **Replace** query).*

Proof. Here we follow the idea from [3, 18]. Assume there exists a forger $\mathcal{A}_1$. We construct an algorithm $\mathcal{S}_1$ that makes use of $\mathcal{A}_1$ to solve DL problem. $\mathcal{S}_1$ is given a multiplicative group $\mathcal{G}$ with generator g and order q, and a group element $A \in \mathcal{G}$. $\mathcal{S}_1$ is asked to find $\alpha \in \mathbb{Z}_q$ such that $g^\alpha = A$.

Setup.

$\mathcal{S}_1$ chooses a hash function $H : \{0,1\}^* \to \mathbb{Z}_q^*$ which behaves like a random oracle. $\mathcal{S}_1$ is responsible for the simulation of this random oracle. $\mathcal{S}_1$ assigns $X = A$ and outputs the public parameter params$=(\mathcal{G}, q, g, X, H)$ to $\mathcal{A}_1$. $\mathcal{S}_1$ also keeps three lists L_1, L_2, L_3, the functions of these lists are the same as mentioned in **Game I** in Section 2.

UserKeyGen/Corruption.

Query: Whenever receiving a query ID, $\mathcal{S}_1$ generates the secret and public key pair according to the algorithm and stores in the table and outputs the public key. On the corruption query, $\mathcal{S}_1$ sets $L_3 = L_3 \cup \{ID\}$ and returns the corresponding secret key.

CertGen.

Query: $\mathcal{A}_1$ is allowed to make certification query for a public key upk_{ID} with identification string ID. $\mathcal{S}_1$ simulates the oracle as follow. It randomly chooses $a, b \in_R \mathbb{Z}_q^*$ and sets $R = X^a g^b$, $k = -a^{-1} R \bmod q$, and $H(upk_{ID}, ID) = bk \bmod q$. Note that (R, k) generated in this way satisfies Equation (2) in the CertGen algorithm and it is a valid certificate. $\mathcal{S}_1$ sets $L_1 = L_1 \cup \{ID\}$, outputs (R, k) as the certificate of upk_{ID}, ID and store the value of $(R, k, H(upk_{ID}, ID), upk_{ID}, ID)$ in the table for consistency. Later if $\mathcal{A}_1$ queries the H random oracle for (upk_{ID}, ID), $\mathcal{S}_1$ outputs the same value. IF (upk_{ID}, ID) is not found in the table, $\mathcal{S}_1$ executes the **CertGen** oracle simulation, stores the value $(R, k, H(upk_{ID}, ID), upk_{ID}, ID)$ in the table and output $H(upk_{ID}, ID)$ only.

ReplaceKey.

Query: Suppose $\mathcal{A}_1$ makes the query with an input (ID, upk'_{ID}), $\mathcal{S}_1$ first makes a **CertGen** query to obtain an item $(R, k, H(upk_{ID}, ID), upk_{ID}, ID)$, then sets $L_2 = L_2 \cup \{ID\}$, $upk_{ID} = upk'_{ID}$, and updates the item $(R, k, H(upk_{ID}, ID), upk_{ID}, ID)$ to record this replacement.

Ring-Sign.

Query: $\mathcal{A}_1$ chooses a group of n users whose identities form the set $L_{ID} = \{ID_1, \ldots, ID_n\}$, and their corresponding public keys form the set $L_{upk} = \{upk_{ID_1}, \ldots, upk_{ID_n}\}$, and may ask a ring signature on a message m of this group. On receiving a **Ring-Sign** query $RS(m, L_{ID}, L_{upk})$, $\mathcal{S}_1$ creates a ring signature as follow.

1) Choose a random index $s \in \{1, \ldots, n\}$.
2) For all $i \in \{1, \ldots, n\} \setminus \{s\}$, choose $y_i \in \mathbb{Z}_q^*$ uniformly at random, compute $Y_i = R^{-y_i}$.
3) For all $i \in \{1, \ldots, n\} \setminus \{s\}$, compute $h_i = H(m \| L_{upk} \| L_{ID} \| Y_i)$.
4) Choose $h_s \in \mathbb{Z}_q^*$, $z \in \mathbb{Z}_q^*$ at random.
5) Compute $Y_s = \prod_{i=1}^n (g^{u_i})^{h_i \tilde{h}_i} / R^z \prod_{i \neq s} Y_i \sum_{i=1}^n (X^{u_i})^{h_i R}$.
6) Set $h_s = H(m \| L_{upk} \| L_{ID} \| Y_s)$.
7) Return $(m, L_{ID}, L_{upk}, \sigma = (\{Y_1, \ldots, Y_n, z\}))$ as answer.

Forgery: Finally, $\mathcal{A}_1$ outputs a tuple $(m^*, L^*_{ID}, L^*_{upk}, \sigma^*_{(1)} = (Y_1^*, \ldots, Y_n^*, z^*_{(1)}))$ which means σ^* is a ring signature on message m^* on behalf of the group specified by identities in L^*_{ID} and the corresponding public keys in L^*_{upk}. It is required that $\mathcal{S}_1$ does not know the secret key of any member in this group, $L^*_{ID} \cap ((L_1 \cap L_2) \cup L_3) = \perp$ and the ring signature σ^* on message m^* on behalf of the group must be valid. $\mathcal{S}_1$ rewinds $\mathcal{A}_1$ to the point $H(m^* \| L^*_{upk} \| L^*_{ID} \| Y_1^*)$ and supplies with a different value (corresponding to the same input value to the hash query). $\mathcal{A}_1$ outputs another pair of signature $(m^*, L^*_{ID}, L^*_{upk}, \sigma^*_{(2)} = (Y_1^*, \ldots, Y_n^*, z^*_{(2)}))$. $\mathcal{S}_1$ repeats $2n$ times and obtains $\sigma^*_{(3)} = (Y_1^*, \ldots, Y_n^*, z^*_{(3)})$, $\ldots$, $\sigma^*_{(2n+2)} = (Y_1^*, \ldots, Y_n^*, z^*_{(2n+2)})$. Note that for all $i \in \{1, \ldots, n\}$, Y_i^* should be the same every time. We let $(c_1, \ldots, c_{2n+2})$ be the output of the random oracle queries $H(m^* \| L^*_{upk} \| L^*_{ID} \| Y_i^*)$.

We also denote $u_1, \ldots, u_n, r, x, y_1, \ldots, y_n \in \mathbb{Z}_q^*$ such that $g^r = R$, $g^x = X$, $g^{y_i} = Y_i^*$ and $upk_{ID_i} = (g^{u_i}, X^{u_i})$ for $i \in \{1, \ldots, n\}$. From Equation (3), we have $\sum_{i=1}^n c_i u_i H(upk_{ID_i}, ID_i) = r z^*_{(i)} + y_1 + \cdots + y_n + \sum_{i=1}^n x u_i c_i R \bmod q$ for $i \in \{1, \ldots, 2n+2\}$. In these equations, only $u_1, \ldots, u_n, y_1, \ldots, y_n, x, r$ are unknown to $\mathcal{S}_1$. $\mathcal{S}_1$ solves for these values from the above $2n+2$ linear independent equations, and output x as the solution of the DL problem.

Probability Analysis: The simulation of the random oracle fails if the oracle assignment $H(upk_{ID_i}, ID_i)$ causes inconsistency. It happens with probability at most q_h/q. Hence the simulation is successful $q_{cert} + nq_s$ times with probability at least $(1 - \frac{q_h}{q})^{q_{cert}+nq_s} \geq 1 - \frac{q_h(q_{cert}+nq_s)}{q}$. Due to the ideal randomness of the random oracle, there exists a query $H(m^* \| L^*_{upk} \| L^*_{ID} \| Y_i^*)$ with probability at least $1 - 1/q$. $\mathcal{S}_1$ guesses it correctly as the point of rewind, with probability at least $1/q_h$. Thus the overall successful probability is $(1 - \frac{q_h(q_{cert}+nq_s)}{q})(1 - \frac{1}{q})(\frac{1}{q_h})\varepsilon$.

□

Theorem 2 (Unforgeability against Game II Adversary). *If a probabilistic polynomial-time forger $\mathcal{A}_2$ has an advantage ε in forging a certificate-based ring signature in an attack modelled by **Game II** after running in time t and making q_h queries to hashing queries, q_u queries to the **UserKeyGen** request oracle, q_{cor} queries to the **Corruption** extraction oracle, and q_s queries to the **Ring-Sign** oracle, then the DL problem can be solved with probability $\varepsilon' = (1 - \frac{nq_h nq_s}{q})(1 - \frac{1}{q})(\frac{1}{q_h})(\frac{1}{nq_u})\varepsilon$ with time $t' < 2(t + q_h T_h + q_u T_u + q_{cor} T_{cor} + q_s T_s)$, where T_h(resp. T_u, T_{cor}, T_s) is the time cost of an Hash query (resp. **UserKeyGen**, **Corruption**, **Ring-Sign** query).*

Proof. Assume there exists a forger $\mathcal{A}_2$. We construct an algorithm $\mathcal{S}_2$ that makes use of $\mathcal{A}_2$ to solve DL problem. $\mathcal{S}_2$ is given a multiplicative group $\mathcal{G}$ with generator g and order q, and a group element $A \in \mathcal{G}$. $\mathcal{S}_2$ is asked to find $\alpha \in \mathbb{Z}_q$ such that $g^\alpha = A$.

Setup.

$\mathcal{S}_2$ chooses a hash function $H : \{0,1\}^* \to \mathbb{Z}_q^*$ which behaves like a random oracle. $\mathcal{S}_2$ is responsible for the simulation of this random oracle. $\mathcal{S}_2$ randomly chooses $x \in_R \mathbb{Z}_q^*$ and sets $X = g^x$, and outputs the public parameter $\mathsf{params}=(\mathcal{G}, q, g, X, H)$ to $\mathcal{A}_2$. $\mathcal{S}_2$ also keeps two lists L_1, L_2, the functions of these lists are the same as mentioned in **Game II** in Section 2.

UserKeyGen.

Query: $\mathcal{S}_2$ chooses a particular query ID' and assigns the public key $upk_{ID'} = (A, A^u, \pi')$ where π' can be simulated by the control of the random oracle. For the other queries, $\mathcal{S}_2$ generates the secret and public key pair according to the algorithm and stores in the table and outputs the public key.

Corruption.

Query: If the query is not ID', $\mathcal{S}_2$ sets $L_2 = L_2 \cup \{ID\}$ and returns the corresponding secret key from the table. Otherwise, $\mathcal{S}_2$ aborts.

ReplaceKey.

Query: Suppose $\mathcal{A}_2$ makes the query with an input

(ID, upk'_{ID}), $\mathcal{S}_2$ first makes a **UserKeyGen** query to obtain an item $(R, k, H(upk_{ID}, ID), upk_{ID}, ID)$, then sets $L_1 = L_1 \cup \{ID\}$, $upk_{ID} = upk'_{ID}$, and updates the item $(R, k, H(upk_{ID}, ID), upk_{ID}, ID)$ to record this replacement.

Ring-Sign.

Query: It can be simulated in the same way as in **Game I**, which also does not require the knowledge of the secret key.

Forgery: Finally, $\mathcal{A}_2$ outputs a tuple (m^*, L^*_{ID}, L^*_{upk}, $\sigma^*_{(1)} = (Y^*_1$, ..., Y^*_n, $z^*_{(1)}))$ which means σ^* is a ring signature on message m^* on behalf of the group specified by identities in L^*_{ID} and the corresponding public keys in L^*_{upk}. It is required that $\mathcal{S}_2$ does not know the secret key of any member in this group, $L^*_{ID} \cap ((L_1 \cup L_2)) = \perp$ and the ring signature σ^* on message m^* on behalf of the group must be valid. In addition, if $upk_{ID'} \notin L^*_{upk}$, $\mathcal{S}_2$ aborts. Otherwise, $\mathcal{S}_2$ rewinds $\mathcal{A}_2$ to the point $H(m^* \| L^*_{upk} \| L^*_{ID} \| Y^*_1)$ and supplies with a different value (corresponding to the same input value to the hash query). $\mathcal{A}_2$ outputs another pair of signature $(m^*, L^*_{ID}, L^*_{upk}, \sigma^*_{(2)} = (Y^*_1, \ldots, Y^*_n, z^*_{(2)}))$. $\mathcal{S}_2$ repeats $2n - 1$ times and obtains $\sigma^*_{(3)} = (Y^*_1, \ldots, Y^*_n, z^*_{(3)})$, ..., $\sigma^*_{(2n+1)} = (Y^*_1, \ldots, Y^*_n, z^*_{(2n+1)})$. Note that for all $i \in \{1, \ldots, n\}$, Y^*_i should be the same every time. We let $(c_1, \ldots, c_{2n+1})$ be the output of the random oracle queries $H(m^* \| L^*_{upk} \| L^*_{ID} \| Y^*_i)$.

We also denote $u_1, \ldots, u_n, r, y_1, \ldots, y_n \in \mathbb{Z}^*_q$ such that $g^r = R$, $g^{y_i} = Y^*_i$ and $upk_{ID_i} = (g^{u_i}, X^{u_i})$ for $i \in \{1, \ldots, n\}$. From Equation (3), we have $\sum_{i=1}^{n} c_i u_i H(upk_{ID_i}, ID_i) = rz^*_{(i)} + y_1 + \cdots + y_n + \sum_{i=1}^{n} x u_i c_i R \bmod q$ for $i \in \{1, \ldots, 2n+1\}$. In these equations, only $u_1, \ldots, u_n, y_1, \ldots, y_n, r$, where $u' \in \{u_1, \ldots, u_n\}$ are unknown to $\mathcal{S}_2$. $\mathcal{S}_2$ solves for these values from the above $2n+1$ linear independent equations, and output u' as the solution of the DL problem.

Probability Analysis: The simulation of the random oracle fails if the oracle assignment $H(upk_{ID_i}, ID_i)$ causes inconsistency. It happens with probability at most q_h/q. Hence the simulation is successful nq_s times with probability at least $(1 - \frac{q_h}{q})^{nq_s} \geq 1 - \frac{nq_h q_s}{q}$. Due to the ideal randomness of the random oracle, there exists a query $H(m^* \| L^*_{upk} \| L^*_{ID} \| Y^*_i)$ with probability at least $1 - 1/q$. $\mathcal{S}_2$ guesses it correctly as the point of rewind, with probability at least $1/q_h$. In addition, $\mathcal{S}_2$ needs to guess correctly that $upk_{ID'} = upk_{ID_i}$, which happens with probability $1/nq_u$. Thus the overall successful probability is $(1 - \frac{nq_h nq_s}{q})(1 - \frac{1}{q})(\frac{1}{q_h})(\frac{1}{nq_u})\varepsilon$.

□

5 Conclusions

We have proposed in this work a new certificate-based ring signature scheme without pairing. Our scheme is formally proved to be existentially unforgeable under the random oracle model, assuming the hardness of DL problem. According to the current MIRACL implementation, a 512-bit Tate pairing takes 20 ms whereas a 1024-bit prime modular exponentiation takes 8.8 ms. Because the proposed scheme does not need pairing computation, so it is more efficient than those which are constructed from bilinear pairing.

Acknowledgments

The authors would like to acknowledge National Natural Science Foundation of China under Grant Nos. 61003230 and 61133016, the Chongqing Key Lab of Computer Network and Communication Technology and the Key Lab of Network Security and Cryptology, the national key scientific and technological special project of China under Grant No. 2011ZX03002-002-03, the Fundamental Research Funds for the Central Universities under Grant No. ZYGX2011J063, and the National High Technology Joint Research Program of China (863 Program) under Grant No. 2011AA010706. The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] M. H. Au, J. K. Liu, W. Susilo, and T. H. Yuen, "Certificate based (linkable) ring signature," in *3rd International Conference on Information Security Practice and Experience-ISPEC 2007*, pp. 79–92, Hong Kong, China, May 2007.

[2] A. K. Awasthi and S. Lal, "Id-based ring signature and proxy ring signature schemes from bilinear pairings," *International Journal of Network Security*, vol. 4, no. 2, pp. 187–192, 2007.

[3] M. Bellare, C. Namprempre, and G. Neven, "Security proofs for identity-based identification and signature schemes," in *Advances in Cryptology-EUROCRYPT 2004*, pp. 268–286, Interlaken, Switzerland, May 2004.

[4] M. Bellare and G. Neven, "Identity-based multisignatures from rsa," in *Topics in Cryptology-CT-RSA 2007*, pp. 145–162, CA, USA, February 2007.

[5] D. Boneh and M. K. Franklin, "Identity-based encryption from the weil pairing," in *Advances in Cryptology-Crypto '01*, pp. 213–229, California, USA, Aug. 2001.

[6] J. Camenisch and M. Stadler, "Efficient group signature schemes for large groups (extended abstract)," in *Advances in Cryptology-Crypto '97*, pp. 410–424, California, USA, Aug. 1997.

[7] D. Chaum and E. v. Hevst, "Group signature," in *Advances in Cryptology-Eurocrypt '91*, pp. 257–265, Brighton, UK, Apr. 1991.

[8] L. Chen, C. Kudla, and K. G. Paterson, "Concurrent signatures," in *Advances in Cryptology-Eurocrypt '04*, pp. 287–305, Interlaken, Switzerland, May 2004.

[9] S. S. M. Chow, R. W. C. Lui, L. C. K. Hui, and S. M. Yiu, "Identity based ring signature: Why, how and what next," in *EuroPKI '05*, pp. 144–161, Canterbury, UK, June 2005.

[10] S. S. M. Chow and W. Susilo, "Generic construction of (identity-based) perfect concurrent signatures," in *7th International Conference on Information and Communications Security-ICICS '05*, pp. 194–206, Beijing, China, Dec. 2005.

[11] S. S. M. Chow, S. M. Yiu, and L. C. K. Hui, "Efficient identity based ring signature," in *3rd International Conference on Applied Cryptography and Network Security-ACNS '05*, pp. 499–512, NY, USA, June 2005.

[12] L. Fabien and D. Vergnaud, "Multi-designated verifiers signatures," in *6th International Conference on Information and Communications Security-ICICS '04*, pp. 495–507, Malaga, Spain, Oct. 2004.

[13] C. Gentry, "Certificate-based encryption and the certificate revocation problem," in *Advances in Cryptology-Eurocrypt '03*, pp. 272–293, Warsaw, Poland, May 2003.

[14] J. Herranz and G. Sáez, "New identity-based ring signature schemes," in *6th International Conference on Information and Communications Security-ICICS '04*, pp. 27–39, Malaga, Spain, Oct. 2004.

[15] B. G. Kang, J. H. Park, and S. G. Hahn, "A certificate-based signature scheme," in *Topics in Cryptology-CT-RSA '04*, pp. 99–111, CA, USA, Feb. 2004.

[16] J. Li, X. Huang, Y. Mu, W. Susilo, and Q. Wu, "Certificate-based signature: Security model and efficient construction," in *EuroPKI '07*, pp. 110–125, Palma de Mallorca, Spain, June 2007.

[17] C. Y. Lin and T. C. Wu, "An identity-based ring signature scheme from bilinear pairings," *Cryptology ePrint Archive*, vol. Report 2003/117, 2003.

[18] J. K. Liu, J. Baek, W. Susilo, and J. Zhou:, "Certificate-based signature schemes without pairings or random oracles," in *11th International Conference on Information Security-ISC '08*, pp. 285–297, Taipei, Taiwan, Sep. 2008.

[19] L. Nguyen, "Accumulators from bilinear pairings and applications to id-based ring signatures and group membership revocation," in *Topics in Cryptology-CT-RSA 2005*, pp. 275–292, CA, USA, Feb. 2005.

[20] R. L. Rivest, A. Shamir, and Y. Tauman, "How to leak a secret: Theory and applications of ring signatures," in *Theoretical Computer Science, Essays in Memory of Shimon Even.*

[21] R. L. Rivest, A. Shamir, and Y. Tauman, "How to leak a secret," in *Advances in Cryptology-AsiaCrypt '01*, pp. 552–565, Gold Coast, Australia, Dec. 2001.

[22] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Advances in Cryptology-Crypto '84*, pp. 47–53, California, USA, Aug. 1984.

[23] W. Susilo and Y. Mu, "Non-interactive deniable ring authentication," in *5th International Conference on Information and Communications Security-ICICS 2003*, pp. 386–401, Huhehaote, China, Oct. 2003.

[24] L. Wang, G. Zhang, and C. Ma, "A survey of ring signature," *Frontiers of Electrical and Electronic Engineering in China*, vol. 3, no. 1, pp. 10–19, 2008.

[25] H. Xiong, Z. Qin, and F. Li, "An anonymous sealed-bid electronic auction based on ring signature," *International Journal of Network Security*, vol. 8, no. 3, pp. 235–242, 2009.

[26] H. Xiong, Z. Qin, and F. Li, "A certificateless proxy ring signature scheme with provable security," *International Journal of Network Security*, vol. 12, no. 2, pp. 92–106, 2011.

[27] F. Zhang and K. Kim, "Id-based blind signature and ring signature from pairings," in *Advances in Cryptology-Asiacrypt '02*, pp. 533–547, Queenstown, New Zealand, December 2002.

[28] J. Zhang, H. Chen, X. Liu, and C. Liu, "An efficient certificate-based signature scheme without pairings," in *2nd International Workshop on Web-Based Contents Management Technologies-WCMT '10*, pp. 177–188, Jiuzhaigou Valley, China, July 2010.

[29] J. Zhang and J. Mao, "An efficient rsa-based certificateless signature scheme," *The Journal of Systems and Software*, vol. 85, no. 3, pp. 638–642, 2012.

Zhiguang Qin is the dean and professor in the School of Computer Science and Engineering,University of Electronic Science and Technology of China. He received his PH.D. degree from University of Electronic Science and Technology of China in 1996. His research interests include: information security and computer network.

Hu Xiong is an Assistant Professor at University of Electronic Science and Technology of China (UESTC). He received his Ph.D degree from UESTC in 2009. His research interests include: cryptographic protocol, and network security.

Fagen Li received his Ph.D. degree in Cryptography from Xidian University, Xi'an, P.R. China in 2007. He is now a associate professor in the School of Computer Science and Engineering, University of Electronic Science and Technology of China. His recent research interests include cryptography and network security.