Certificateless Ring Signatures

Sherman S.M. Chow[1] and Wun-She Yap[2]

[1] Department of Computer Science
Courant Institute of Mathematical Sciences
New York University, NY 10012, USA
schow@cs.nyu.edu
[2] Centre for Cryptography and Information Security, FIT
Multimedia University, 63100 Cyberjaya, Malaysia
wsyap@mmu.edu.my

Abstract. Ring signature scheme is a cryptographic construct that enables a signer to sign on behalf of a group of n different people such that the verifier can only ensure someone in the group signed, but not exactly whom. Ring signatures are utilized in many security applications.
It is tricky to deploy multi-user cryptographic construct due to the complexity involved by certificates. Specifically, ring signatures working under traditional public key infrastructure requires the transfer and verification of n certificates, making the scheme both space and time inefficient. On the other hand, the key-escrow problem of identity-based solution makes the authenticity of the ring signature in question. This paper studies ring signature in certificateless cryptography, one with neither certificate nor key-escrow.
Designing a certificateless ring signature scheme is not entirely trivial. Many certificateless signatures require public key validity checking. In the context of ring signatures, this means both the signer and the verifier need to deal with the complexity in the verification of n public keys. We propose the first certificateless ring signature scheme, without such public key validity checking.

Key words: certificateless signatures, ring signatures

1 Introduction

Ring signature was introduced by Rivest, Shamir, and Tauman [19]. Ring signature is a group oriented signature with privacy concerns: a user can anonymously signs a message on behalf of a groups of spontaneously conscripted users including the actual signer. Any verifier can be convinced that the message has been signed by one of the members in this group, but the actual signer remains unknown. In traditional public key ring signature, both of the signer and verifier must obtain a copy of the user's certificate and check the validity of the certificate before checking the validity of the signature. Authentication of large numbers of public keys which linearly dependent to the group's size will greatly effect the efficiency of the ring signature scheme.

1.1 Applications of Ring Signatures

Ring signatures are utilized in many security applications. For example, a 2-user ring signature with the verifier in the ring can be viewed as a designated-verifier signature [16] that only the designate verifier can ensure its authenticity, which in turns useful in anti-phishing solution.

In phishing attack, email recipients are lured by an legitimate-looking email to a fraudulent website that appears to be an official one. As a consequence, the victims are likely to leak their credentials to the attacker. One may consider having all email digitally signed to avoid such attack. However, it requires the deployment of public key infrastructure (PKI) and takes away the inherent property of being repudiable from email. In using identity-based [22] designated-verifier signature (e.g. [8]), both the designated-authenticity and the repudiability are ensured, without PKI deployment [1].

1.2 Traditional and Identity-based Ring Signatures

Note that a single ring signature involves n-user, where n is the size of the diversion group associated with a ring signature. Experiences told us that it is tricky to deploy multi-user cryptographic construct in ubiquitous computing environment due to the complexity involved by certificates.

Ring signatures working under traditional PKI requires the transfer and verification of n certificates, making the scheme both space and time inefficient. On the other hand, the key-escrow problem of identity-based solution makes the authenticity of the ring signature in question.

Survey of traditional and identity-based ring signatures can be found at [20] and [10] respectively.

1.3 Certificateless Public Key Cryptography

Certificateless public key cryptography (CL-PKC) was formulated by Al-Riyami and Paterson [2] in 2003 to fill the gap between traditional public key cryptography (PKC) and identity-based cryptography [22] (ID-PKC). The basic concept of CL-PKC is to generate a public/private key pair for a user by using a master key of a Key Generation Center (KGC) with a random secret value selected by the user. Thus, the CL-PKC can be seen as a model that is intermediate between PKC and ID-PKC. Hence, CL-PKC achieves implicit certification (through the ID) while does not suffer from the inherent key escrow problem in ID-PKC (through the user public key).

Research on certificateless signature schemes have been very active of late, to name some [2, 5, 7, 13, 14, 17, 23–25, 27]. Sadly, most of these schemes are proved insecure [2, 5, 13, 14, 17, 24, 25]. One of the reasons is that many schemes are lack of a (good) security model that can capture the real world attack, and some simply proposed without formal security proof.

1.4 Possibility of Black-Box Generic Construction

Recently, a generic approach for building identity-based signature schemes with additional properties (for example, blind signature) from traditional signature schemes has been proposed [12]. However, as noted in [12] since ring signature involves public key of users other than the signer, this approach is not applicable.

On the other hand, generic approach exists for building certificateless signature schemes [14]. Without delving into technicalities, the signature produced is basically a concatenation of a traditional signature and an identity-based signature. Again, since more than one public keys are involved, we see no trivial black-box construction of certificateless ring signature from traditional ring signature and identity-based one. Due to the anonymity properties, no one can tell which secret keys are used in the respective signatures. It is entirely possible that both keys may not constitute a valid certificateless key of the *same* user.

1.5 Our Contributions

This paper studies ring signature in certificateless cryptography, one with neither certificate nor key-escrow. We propose the first certificateless ring signature (CLRS), with detailed framework and security proofs.

This turns out to be more tricky than a simple combination of certificateless signature and ring signature one may consider. Note that most certificateless signature schemes (for examples, [5, 15, 17, 27]) require public key validity checking, i.e. even the scheme is free from certificate, the verifier still needs to pay computational effort to check if the purported public key is a valid one. A naive solution simply means verification of n public key is necessary, which offers us no advantage over PKI-based scheme. On the other hand, the signer should perform the same verification because any invalid public key rules out one possible signer, and hence the anonymity is degraded. Being said, our first certificateless ring signature scheme is free for such public key validity checking.

Our scheme can be seen as extending a recent ring signature scheme proposed by Chow and Wong [11] and a recent certificateless signature scheme proposed by Choi *et al.* [7]. It is provably secure against existential unforgeability under chosen message and identity attack in the random oracle model, based on the intractability of k-collision attack algorithm problem (k-CAA) and modified inverse computational Diffie-Hellman problem.

1.6 Organization.

In Section 2, we review some preliminaries. In Section 3, we present the security model for a CLRS scheme. Section 4 propose our concrete CLRS scheme and prove its security. Finally, we give some concluding remarks in Section 5.

2 Mathematical Settings

Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two groups of order q for some large prime q. Like Boneh and Franklin [4], we make use of a bilinear map $\hat{e}(\mathbb{G}_1, \mathbb{G}_1) \rightarrow \mathbb{G}_2$ between these two groups. The map must satisfy the following properties:

1. Bilinear: We say that a map $\hat{e}(\mathbb{G}_1, \mathbb{G}_1) \rightarrow G_2$ is *bilinear* if $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$ for all $P, Q \in \mathbb{G}_1$.
2. Non-degenerate: For every Q there exists a P, so that $\hat{e}(P, Q) \neq 1_{\mathbb{G}_2}$.
3. Computable: There is an efficient algorithm to compute $\hat{e}(P, Q)$ for any $P, Q \in \mathbb{G}_1$.

A bilinear map satisfying the three properties above is said to be an *admissible* bilinear map. Throughout the paper, we view $\mathbb{G}_1$ as an additive group and $\mathbb{G}_2$ as a multiplicative group.

Definition 1. *The k-Collision Attack Algorithm (k–CAA) Problem in $\mathbb{G}_1$ is defined as follows: For a fixed and known integer k, given a $(2k+2)$-tuple $(t_1, \ldots, t_k, P, Q = sP, \frac{1}{t_1+s}P, \ldots, \frac{1}{t_k+s}P) \in \mathbb{Z}_q^k \times \mathbb{G}_1^{k+2}$, output a pair (A, c) such that $A = \frac{1}{c+s}P$ where $c \in Z_q^* \backslash \{t_1, \ldots, t_k\}$. We say that the (k, τ, ϵ)-CAA assumption holds in $\mathbb{G}_1$ if no τ-time algorithm has advantage at least ϵ in solving the k-CAA problem.*

Definition 2. *The Modified Inverse Computational Diffie-Hellman ($mICDH$) Problem in $\mathbb{G}_1$ is defined as follows: Given b, P and aP for some $a, b \in \mathbb{Z}_q^*$, output $(a+b)^{-1}P$. We say that the mICDH assumption holds in $\mathbb{G}_1$ if no τ-time algorithm has advantage at least ϵ in solving the mICDH problem.*

The k-CAA assumption has been widely used in a number of cryptographic schemes (e.g. all identity-based schemes following Sakai-Kasahara's paradigm [21].) The relation of k-CAA with some other problems can be found in [26]. In particular, it is shown in [26] that k-CAA is equivalent to the k-Strong Diffie-Hellman (k-SDH) problem (or $(k+1)$-exponent problem in the terms of [26]). Those who worried about the k-CAA assumption may find the security analysis of k-SDH problem in [6] useful.

The mICDH assumption and k-CAA assumption are both related to "inversion element", a scalar-point multiplication of the generator where the scalar is an inverse of something related to the problem instance. In an mICDH problem instance, only one $\mathbb{Z}_q^*$ element and no inversion element is included. Besides, the mICDH solution is completely determined by the problem instance, in contrast with the flexibility of k-CAA problem that many possible solutions exist.

3 Framework of Certificateless Ring Signatures

Now, we present the definition and security model of CLRS. In order to maintain the features of ring signature, CLRS scheme must satisfy the following properties:

1. **Anonymity**: Any verifier should not have probability greater than $1/n$ to guess the identity of the actual signer who signed on a message on behalf of a group which consists n members. If the verifier is one of the members in the group, then he/she should not have probability greater than $1/(n-1)$ to guess the identity of the actual signer.
2. **Unforgeability**: Any attacker must have non negligible probability of success in forging a valid signature for some messages m on behalf of a group, even if he knows valid ring signatures for some messages, different from m, that he can adaptively choose.

3.1 Definition of CLRS

A certificateless ring signature scheme consists of the following five algorithms: $\mathcal{SETUP}$, $\mathcal{PKGEN}$, $\mathcal{UKGEN}$, $\mathcal{SIG}$, and $\mathcal{VER}$.

1. $\mathcal{SETUP}$ is a probabilistic algorithm that takes security parameter k as input and returns the system parameters, *params* and master secret key.
2. $\mathcal{PKGEN}$ is a deterministic algorithm that takes *params*, master secret key, and ID as inputs. It returns a partial private key, D_{ID}.
3. $\mathcal{UKGEN}$ is a probabilistic algorithm that takes *params*, D_{ID}, and ID as inputs. The algorithm returns a public/private key pair as R_{ID}, S_{ID}.
4. $\mathcal{SIG}$ is a probabilistic algorithm that accepts a message, $m \in M$, a group of n user IDs, $\bigcup_{i=1}^{n}\{ID_i\}$, *params*, and the private key of one member S_{ID_A} to produce a signature σ on the message m.
5. $\mathcal{VER}$ is a deterministic algorithm that accepts a signature σ, message m, *params*, a group of n user IDs, $\bigcup_{i=1}^{n}\{ID_i\}$, and a group of n user public keys, $\bigcup_{i=1}^{n}\{R_i\}$ to output $\top$ for *true* or $\perp$ for *false*, depending on whether σ is a valid signature signed by a certain member in the group $(\bigcup_{i=1}^{n}\{ID_i\}, \bigcup_{i=1}^{n}\{R_i\})$ on a message m.

3.2 Definition of Security

As defined in [2], there are two types of adversaries with different capabilities. In CLRS, we assume Type I Adversary, $\mathcal{A}_{\mathcal{I}}$ acts as a dishonest user while Type II Adversary, $\mathcal{A}_{\mathcal{II}}$ acts as malicious Key Generation Center (KGC):

1. *CLRS Type I Adversary*: Adversary $\mathcal{A}_{\mathcal{I}}$ does not have access to master secret key. However, $\mathcal{A}_{\mathcal{I}}$ may replace public keys, extract partial private and private keys and make sign queries.
2. *CLRS Type II Adversary*: Adversary $\mathcal{A}_{\mathcal{II}}$ does have access to master secret key, but cannot not replace public keys of entities.

We provide a formal definition of existential unforgeability of CLRS under adaptive chosen message and identity attack (EUF-CLRS-CMIA2) for both two types of adversaries. They are defined using the following game between an adversary $\mathcal{A} \in \{\mathcal{A}_{\mathcal{I}}, \mathcal{A}_{\mathcal{II}}\}$ and a challenger $\mathcal{C}$.

EUF-CLRS-CMIA2 Game for Type I Adversary
`Setup`: The challenger $\mathcal{C}$ takes a security parameter k and runs the $\mathcal{SETUP}$ to generate common public parameters *params* and master secret key s. Then, $\mathcal{C}$ sends *params* to $\mathcal{A_I}$.

`Attack`: The adversary, $\mathcal{A_I}$ can perform a polynomially bounded number of queries described below in an adaptive manner (i.e., each query may depend on the responses to the previous queries).
-`Hash Queries`: $\mathcal{A_I}$ can request the hash values for any input.
-$\mathcal{PKGEN}$: $\mathcal{A_I}$ can request the partial private key, D_{ID} for any ID except those associated with the forgery.
-`Extract-Private-Key`: $\mathcal{A_I}$ can request the private key for any ID except the challenged ID.
-`Request-Public-Key`: $\mathcal{A_I}$ can request the public key for any ID.
-`Replace-Public-Key`: For any ID, $\mathcal{A_I}$ can choose a new secret value, x_{ID} and compute the new public key, R_{ID}, $\mathcal{A_I}$ then sets R_{ID} as ID's new public key.
-$\mathcal{SIG}$: $\mathcal{A_I}$ chooses a group of n user IDs, $\bigcup_{i=1}^{n}\{ID_i\}$, a group of n user public keys, $\bigcup_{i=1}^{n}\{R_i\}$, and any message m. $\mathcal{C}$ outputs a signature σ on the message m.

`Forgery`: The adversary $\mathcal{A_I}$ outputs a signature σ on a message m, a group of n user IDs $\bigcup_{i=1}^{n}\{U_i\}$, and a group of n user public keys, $\bigcup_{i=1}^{n}\{R_i\}$. The only restriction is that $(m, \bigcup_{i=1}^{n}\{ID_i\})$ does not appear in the set of previous $\mathcal{SIG}$ queries and each of the partial signing keys in $\bigcup_{i=1}^{n}\{D_{ID_i}\}$ is never returned by any $\mathcal{PKGEN}$ query. It wins the game if $\mathcal{VER}(\sigma, m, \bigcup_{i=1}^{n}\{ID_i\}, \bigcup_{i=1}^{n}\{R_i\})$ is equal to $\top$. The advantage of $\mathcal{A_I}$ is defined as the probability that it wins.

EUF-CLRS-CMIA2 Game for Type II Adversary
`Setup`: The challenger $\mathcal{C}$ takes a security parameter k and runs the $\mathcal{SETUP}$ to generate common public parameters *params* and master secret key s. Then, $\mathcal{C}$ sends *params* and s to $\mathcal{A_{II}}$.
`Attack`: The adversary, $\mathcal{A_{II}}$ can perform a polynomially bounded number of queries described above (same as *EUF-CLRS-CMIA2 Game for Type I Adversary*) in an adaptive manner (i.e., each query may depend on the responses to the previous queries). However, $\mathcal{A_{II}}$ cannot replace any public key.
`Forgery`: The adversary $\mathcal{A_{II}}$ outputs a signature σ on a message m, a group of n user IDs $\bigcup_{i=1}^{n}\{U_i\}$, and a group of n user public keys, $\bigcup_{i=1}^{n}\{R_i\}$. The only restriction is that $(m, \bigcup_{i=1}^{n}\{ID_i\})$ does not appear in the set of previous $\mathcal{SIG}$ queries. It wins the game if $\mathcal{VER}(\sigma, m, \bigcup_{i=1}^{n}\{ID_i\}, \bigcup_{i=1}^{n}\{R_i\})$ is equal to $\top$. The advantage of $\mathcal{A_{II}}$ is defined as the probability that it wins.

Definition 3. *A certificateless ring signature scheme is said to satisfy the property of existential unforgeability against adaptive chosen message and identity attack (EUF-CLRS-CMIA2 secure) if no adversary has a non-negligible advantage in the EUF-CLRS-CMIA2 game.*

Definition 4. *A certificateless ring signature scheme is said to have the unconditional signer ambiguity if for any group of n users'* ID, $\bigcup_{i=1}^{n}\{U_i\}$, *any group* *of n users' public key, $\bigcup_{i=1}^{n}\{R_i\}$, any message m, and any signature σ; any verifier $\mathcal{A}$ even with unbounded computing resources, cannot identify the actual signer with probability better than a random guess. That is, $\mathcal{A}$ can only output the actual signer indexed by A with probability no better than $1/n$ or $1/(n-1)$ if $\mathcal{A}$ is one member of the signer's group.*

4 Proposed Scheme

In this section, we propose the first non-trivial CLRS, and prove the security of the proposed scheme, based on [7] and [11]. Our partial private key follows from the identity-based user secret key generation of Sakai-Kasahara's identity-based encryption scheme [21], which is subsequently presented as a short signature scheme with other extensions in [26].

4.1 Construction

$\mathcal{SETUP}$: The KGC performs as follows to generate system parameters and master secret key:

1. Generate $(\mathbb{G}_1, \mathbb{G}_2, \hat{e})$ where $\mathbb{G}_1$ and $\mathbb{G}_2$ are cyclic groups of prime order q and $\hat{e}$ is an admissible bilinear map.
2. Choose a random $s \in_R \mathbb{Z}_q^*$ and a generator P of $\mathbb{G}_1$. Compute the corresponding public key $P_{pub} = sP$.
3. Pre-compute $g = \hat{e}(P, P)$.
4. Choose three cryptographic hash functions $H_0 : \{0,1\}^* \to \mathbb{Z}_q^*$, $H_1 : \{0,1\}^* \to \mathbb{Z}_q^*$ and $H_2 : \mathbb{G}_1 \to \mathbb{Z}_q^*$.

   The system parameters are:

$$\{\mathbb{G}_1, \mathbb{G}_2, q, \hat{e}(\cdot,\cdot), H_0(\cdot), H_1(\cdot), H_2(\cdot), g, P, P_{pub}\}.$$

A recent security concern in certificateless paradigm is that a malicious KGC may manipulate these parameters to compromise the security of users [3]. However, this concern can be handled easily by some standard practice like using the outputs of a pseudo-random function (PRF) as the parameters.

Only two group elements are included in the system parameters of our proposed scheme. The discrete logarithm of P_{pub} with respect to P should be known to the KGC for supporting valid partial private key generation query. We only require the KGC to publish the input of one PRF invocation for generation of P, in contrast with scheme like [18], in which a whole bunch of generators in the system parameters should be protected in this way.

$\mathcal{PKGEN}$: The signer with identity $\mathsf{ID} \in \{0,1\}^*$ submits ID to KGC. KGC sets the signer's public key q_{ID} to be $H_0(\mathsf{ID}) \in \mathbb{Z}_q^*$, computes the signer's partial private key D_{ID} by $D_{\mathsf{ID}} = \frac{1}{s+q_{\mathsf{ID}}}P$. Then KGC sends the partial private key to the signer via a secure channel.

Due to the structure of identity-based secret key in Sakai-Kasahara's paradigm [21], we do not need to hash an arbitrary string to a point on elliptic curve, which is a somewhat inefficient operation.

$\mathcal{UKGEN}$: After obtained the partial private key D_{ID} from the KGC, the signer with identity ID performs the following to get his/her key pair.

1. Compute $Q_{\mathsf{ID}} = P_{pub} + H_0(\mathsf{ID})P$.
2. Randomly choose $x_{\mathsf{ID}} \in_R \mathbb{Z}_q^*$.
3. Compute $R_{\mathsf{ID}} = x_{\mathsf{ID}} Q_{\mathsf{ID}}$ and $y_{\mathsf{ID}} = H_2(R_{\mathsf{ID}})$.
4. Compute $S_{\mathsf{ID}} = \frac{1}{x_{\mathsf{ID}}+y_{\mathsf{ID}}} D_{\mathsf{ID}}$.
5. Return public/private key pair as $(R_{\mathsf{ID}}, S_{\mathsf{ID}})$.

$\mathcal{SIG}$: Let $L = \{\mathsf{ID}_1, \mathsf{ID}_2, \cdots, \mathsf{ID}_n\}$ be the set of identities of n users and $R = \{R_{\mathsf{ID}_1}, R_{\mathsf{ID}_2}, \cdots, R_{\mathsf{ID}_n}\}$ be the set of corresponding public keys. The actual signer, indexed by A (i.e. his/her identity ID_A), carries out the following steps to give an certificateless ring signature on behalf of the group L.

1. Compute $y_{\mathsf{ID}_i} = H_2(R_{\mathsf{ID}_i})\ \forall i \in \{1, 2, \cdots, n\}$.
2. Choose $v_{\mathsf{ID}_i} \in_R \mathbb{Z}_q^*$, and compute $V_{\mathsf{ID}_i} = v_{\mathsf{ID}_i} P\ \forall i \in \{1, 2, \cdots, n\} \backslash \{A\}$.
3. Choose $r \in_R \mathbb{Z}_q^*$.
4. Compute $u = g^r \prod_{i \neq A} \hat{e}(V_{\mathsf{ID}_i}, R_{\mathsf{ID}_i} + y_{\mathsf{ID}_i} Q_{\mathsf{ID}_i})$ (the logical step) by $u = g^r \hat{e}(P, \sum_{i \neq A} v_{\mathsf{ID}_i}(R_{\mathsf{ID}_i} + y_{\mathsf{ID}_i} Q_{\mathsf{ID}_i}))$ (the concrete step).
5. Compute $h = H_1(m, u, L, R)$ and $V_{\mathsf{ID}_A} = (h + r)S_{\mathsf{ID}_A}$.
6. Output the signature on m as $\sigma = \{u, \bigcup_{i=1}^{n}\{V_{\mathsf{ID}_i}\}\}$.

$\mathcal{VER}$: A verifier can check the validity of a ring signature $\sigma = \{u, \bigcup_{i=1}^{n}\{V_{\mathsf{ID}_i}\}\}$ on the message m signed on behalf of a set of identities L with corresponding public keys R by checking if $g^{H_1(m,u,L,R)} \cdot u = \prod \hat{e}(V_{\mathsf{ID}_i}, R_{\mathsf{ID}_i} + y_{\mathsf{ID}_i} Q_{\mathsf{ID}_i})$ holds.

4.2 Correctness

$$
\begin{aligned}
& \prod \hat{e}(V_{\mathsf{ID}_i}, R_{\mathsf{ID}_i} + y_{\mathsf{ID}_i} Q_{\mathsf{ID}_i}) \\
&= \hat{e}(V_{\mathsf{ID}_A}, R_{\mathsf{ID}_A} + y_{\mathsf{ID}_A} Q_{\mathsf{ID}_A}) \prod_{i \neq A} \hat{e}(V_{\mathsf{ID}_i}, R_{\mathsf{ID}_i} + y_{\mathsf{ID}_i} Q_{\mathsf{ID}_i}) \\
&= \hat{e}((h+r)S_{\mathsf{ID}_A}, x_{\mathsf{ID}_A}(P_{pub} + H_1(\mathsf{ID}_A)P) + y_{\mathsf{ID}_A}(P_{pub} + H_1(\mathsf{ID}_A)P)) \prod_{i \neq A} \hat{e}(V_{\mathsf{ID}_i}, R_{\mathsf{ID}_i} + y_{\mathsf{ID}_i} Q_{\mathsf{ID}_i}) \\
&= \hat{e}((h+r)S_{\mathsf{ID}_A}, (x_{\mathsf{ID}_A} + y_{\mathsf{ID}_A})(s + H_1(\mathsf{ID}_A))P) \prod_{i \neq A} \hat{e}(V_{\mathsf{ID}_i}, R_{\mathsf{ID}_i} + y_{\mathsf{ID}_i} Q_{\mathsf{ID}_i}) \\
&= \hat{e}((h+r)P, P) \prod_{i \neq A} \hat{e}(V_{\mathsf{ID}_i}, R_{\mathsf{ID}_i} + y_{\mathsf{ID}_i} Q_{\mathsf{ID}_i}) \\
&= g^{h+r} \prod_{i \neq A} \hat{e}(V_{\mathsf{ID}_i}, R_{\mathsf{ID}_i} + y_{\mathsf{ID}_i} Q_{\mathsf{ID}_i}) \\
&= g^h g^r \prod_{i \neq A} \hat{e}(V_{\mathsf{ID}_i}, R_{\mathsf{ID}_i} + y_{\mathsf{ID}_i} Q_{\mathsf{ID}_i}) = g^{H_1(m,u,L,R)} \cdot u
\end{aligned}
$$

4.3 Security Analysis

The security proofs below borrow the proof ideas from [7].

Theorem 1. *Our CLRS scheme is existential unforgeable against the Type I adversary in the random oracle model assuming the k-CAA is hard.*

Proof. Let $\mathcal{A}_\mathcal{I}$ be a forger that breaks the proposed signature scheme under adaptive chosen message and identity attack. We show that how $\mathcal{B}$ can use $\mathcal{A}_\mathcal{I}$ to solve the k-CAA instance $(t_1, \ldots, t_k, P, Q = sP, \frac{1}{t_1+s}P, \ldots, \frac{1}{t_k+s}P)$ where $k \geq q_{H_0}$ (we suppose $\mathcal{A}_\mathcal{I}$ makes at most q_{H_0} queries to H_0 oracle). Its goal is to compute $\frac{1}{s+q_{\mathsf{ID}_A}}P$ for some $q_{\mathsf{ID}_A} \notin \{t_1, \ldots, t_k\}$ and A denotes an arbitrary signer associated with the forgery.

$\mathcal{B}$ sets $g = \hat{e}(P, P)$ and $P_{pub} = sP$ where s is the master secret key, which is unknown to $\mathcal{B}$. $\mathcal{B}$ then gives the system parameters to $\mathcal{A}_\mathcal{I}$. Without loss of generality, we assume that any extraction ($\mathcal{PKGEN}$, `Request-Public-Key`, `Extract-Private-Key`) and $\mathcal{SIG}$ queries are preceded by H_0 query, and the $\mathcal{SIG}$ and `Extract-Private-Key` queries are preceded by `Request-Public-Key` query. $\mathcal{B}$ maintains four lists L_{H_0}, L_{H_1}, and $L_{H_2}, L_K = \langle \mathsf{ID}, R_{\mathsf{ID}}, x_{\mathsf{ID}}, c \in \{0, 1\}\rangle$ which are initially empty.

Adversary $\mathcal{B}$ interacts with $\mathcal{A}_\mathcal{I}$ in the `Attack` phase of the game as follows:

H_0 `Queries`: When $\mathcal{A}_\mathcal{I}$ queries H_0 on ID_i where $1 \leq i \leq q_{H_0}$, $\mathcal{B}$ checks the corresponding L_{H_0} and outputs Q_{ID_i} if such query has already been made. Otherwise, $\mathcal{B}$ picks $j \in \{1, q_{H_0}\}$ at random. If $i = j$ (we let $\mathsf{ID}_i = \mathsf{ID}^*$ at this point), $\mathcal{B}$ returns $q_{\mathsf{ID}^*} = t_0$ where $t_0 \in_R \mathbb{Z}_q^*$ is chosen randomly, otherwise $q_{\mathsf{ID}_i} = t_i$ (t_i are taken from the k-CAA instance). $\mathcal{B}$ then computes $Q_{\mathsf{ID}_i} = P_{pub} + q_{\mathsf{ID}_i}P$ and adds $\langle \mathsf{ID}_i, Q_{\mathsf{ID}_i}, q_{\mathsf{ID}_i}\rangle$ to L_{H_0}.

H_1 `Queries`: When $\mathcal{A}_\mathcal{I}$ issues a query H_1 on $(m_i||u||L = \bigcup_{i=1}^n \mathsf{ID}_i||R = \bigcup_{i=1}^n R_{\mathsf{ID}_i})$, $\mathcal{B}$ checks the corresponding L_{H_1} and outputs h_i if such value is defined. Otherwise, $\mathcal{B}$ picks $h_i \in_R \mathbb{Z}_q^*$ at random. $\mathcal{B}$ then outputs h_i as answer and adds $\langle m_i, u, L, R, h_i\rangle$ to L_{H_1}.

H_2 `Queries`: When $\mathcal{A}_\mathcal{I}$ queries H_2 on input R_{ID_i}, $\mathcal{B}$ checks the corresponding L_{H_2} and outputs y_{ID_i} if such value is defined. Otherwise, $\mathcal{B}$ picks $y_{\mathsf{ID}_i} \in_R Z_q^*$ at random and outputs y_{ID_i} as answer, and adds $\langle R_{\mathsf{ID}_i}, y_{\mathsf{ID}_i}\rangle$ to L_{H_2}.

$\mathcal{PKGEN}(\mathsf{ID}_i)$: When $\mathcal{A}_\mathcal{I}$ queries on input ID_i, $\mathcal{B}$ performs as follows:

1. If $\mathsf{ID}_i = \mathsf{ID}_A$, $\mathcal{B}$ outputs `FAIL` and aborts the simulation.
2. If $\mathsf{ID}_i \neq \mathsf{ID}_A$, $\mathcal{B}$ returns $D_{\mathsf{ID}_i} = \frac{1}{s+t_i}P$.

`Request-Public-Key`(ID_i): When $\mathcal{A}_\mathcal{I}$ queries on input ID_i, if the list L_K contains $\langle \mathsf{ID}_i, R_{\mathsf{ID}_i}, x_{\mathsf{ID}_i,c}\rangle$, $\mathcal{B}$ returns R_{ID_i}. If no such query exists, $\mathcal{B}$ finds $\langle \mathsf{ID}_i, Q_{\mathsf{ID}_i}, q_{\mathsf{ID}_i}, c\rangle$ in L_{H_0}, and picks a random $x_{\mathsf{ID}_i} \in_R \mathbb{Z}_q^*$. $\mathcal{B}$ then returns $R_{\mathsf{ID}_i} = x_{\mathsf{ID}_i}Q_{\mathsf{ID}_i}$ and adds $\langle \mathsf{ID}_i, R_{\mathsf{ID}_i}, x_{\mathsf{ID}_i,1}\rangle$ to L_K.

`Extract-Private-Key`(ID_i): For query on input ID_i, $\mathcal{B}$ performs as follows:

1. If $\mathsf{ID}_i = \mathsf{ID}_A$, $\mathcal{B}$ outputs `FAIL` and aborts the simulation.
2. If $\mathsf{ID}_i \neq \mathsf{ID}_A$, $\mathcal{B}$ finds $\langle \mathsf{ID}_i, R_{\mathsf{ID}_i}, x_{\mathsf{ID}_i}, c\rangle$ in L_K. If $c = 1$, $\mathcal{B}$ performs as follows:
   - If the list L_{H_2} contains $\langle R_{\mathsf{ID}_i}, y_{\mathsf{ID}_i}\rangle$, $\mathcal{B}$ returns $S_{\mathsf{ID}_i} = (x_{\mathsf{ID}_i}+y_{\mathsf{ID}_i})^{-1}\frac{1}{s+q_{\mathsf{ID}_i}}P$.
   - If the list L_{H_2} does not contain $\langle R_{\mathsf{ID}_i}, y_{\mathsf{ID}_i}\rangle$, $\mathcal{B}$ makes query H_2 on input R_{ID_i} and returns $S_{\mathsf{ID}_i} = (x_{\mathsf{ID}_i}+y_{\mathsf{ID}_i})^{-1}\frac{1}{s+q_{\mathsf{ID}_i}}P$.

   Otherwise, if $c = 0$, $\mathcal{B}$ gets additionally information x'_{ID_i} from $\mathcal{A}_{\mathcal{I}}$, $\mathcal{B}$ simulates as in the above case ($c = 1$).

`Replace-Public-Key`$(\mathsf{ID}_i, R'_{\mathsf{ID}_i})$: When $\mathcal{A}_{\mathcal{I}}$ queries on input $(\mathsf{ID}_i, R_{\mathsf{ID}_i})$:

1. If the list L_K contains $\langle \mathsf{ID}_i, R_{\mathsf{ID}_i}, x_{\mathsf{ID}_{i},c}\rangle$, $\mathcal{B}$ sets $R_{\mathsf{ID}_i} = R'_{\mathsf{ID}_i}$ and $c = 0$.
2. If the list L_K does not contain $\langle \mathsf{ID}_i, R_{\mathsf{ID}_i}, x_{\mathsf{ID}_{i},c}\rangle$, $\mathcal{B}$ makes a `Replace-Public-Key` query on ID_i. Then, $\mathcal{B}$ sets $R_{\mathsf{ID}_i} = R'_{\mathsf{ID}_i}$ and $c = 0$.

$\mathcal{SIG}(L, R, m)$: When $\mathcal{A}_{\mathcal{I}}$ queries on input $(L = \bigcup_{i=1}^{n} \mathsf{ID}_i, R = \bigcup_{i=1}^{n} R_{\mathsf{ID}_i}, m)$, $\mathcal{B}$ finds $\langle \mathsf{ID}_i, Q_{\mathsf{ID}_i}, q_{\mathsf{ID}_i}\rangle$ in L_{H_0} and $\langle \mathsf{ID}_i, R_{\mathsf{ID}_i}, x_{\mathsf{ID}_i}, c\rangle$ in L_K for every ID and R_{ID}.

1. If $c = 1$, $\mathcal{B}$ performs as follows:
   - Choose an index $A \in \{1, \ldots, n\}$.
   - Find $\langle R_{\mathsf{ID}_i}, y_{\mathsf{ID}_i}\rangle$ in $L_{H_2} \forall i \in \{1, 2, \cdots, n\}$. Let R_A denote the signer's public key and y_{ID_A} denotes the $H_2(R_{\mathsf{ID}_A})$. If it does not exist, $\mathcal{B}$ picks a random $y_{\mathsf{ID}_i} \in_R \mathbb{Z}_q^*$ and adds $\langle R_{\mathsf{ID}_i}, y_{\mathsf{ID}_i}\rangle$ to L_{H_2}.
   - Choose $v_{\mathsf{ID}_i} \in_R \mathbb{Z}_q^*$, and compute $V_{\mathsf{ID}_i} = v_{\mathsf{ID}_i}P \ \forall i \in \{1, 2, \cdots, n\}\backslash\{A\}$.
   - Choose $r_A, h_A \in_R \mathbb{Z}_q^*$ and compute $V_{\mathsf{ID}_A} = (r_A + h_A)\frac{1}{x_{\mathsf{ID}_A}+y_{\mathsf{ID}_A}}P$.
   - Compute $u = g^{-h_A}\cdot\hat{e}(P, \sum_{i\neq A} v_{\mathsf{ID}_i}(R_{\mathsf{ID}_i} + y_{\mathsf{ID}_i}Q_{\mathsf{ID}_i}))\cdot\hat{e}((r_A{+}h_A)P, Q_{\mathsf{ID}_A})$ and set $h_A = H_1(m, u, L, R)$. ($\mathcal{B}$ outputs `FAIL` and aborts the simulation if the $H_1(m, u, L, R)$ has already been defined in the list L_{H_1}).
   - Return $\sigma = \{u, \bigcup_{i=1}^{n}\{V_{\mathsf{ID}_i}\}\}$.
2. If $c = 0$, $\mathcal{B}$ gets additionally information x'_{ID_i} from $\mathcal{A}_{\mathcal{I}}$, $\mathcal{B}$ simulates as in the above case ($c = 1$).

`Forgery`: The next step of the simulation is to apply the general forking lemma. Let $\langle u^*, L^*, h, \bigcup^{n}\{V_{\mathsf{ID}_i}\},\rangle$ be a forgery of a ring signature on message m^* with respect to a ring containing all uncompromised user. Suppose without loss of generality that a key for one of the ring member is $\langle \mathsf{ID}_A, R_{\mathsf{ID}_A}\rangle$.

$\mathcal{B}$ then replays $\mathcal{A}_{\mathcal{I}}$ with the same random tape but different H_1. Suppose H_1 outputs h and $h' \neq h$ in the first round and the second round respectively. We get another valid forgery $\langle u^*, L^*, \bigcup_{i\neq A}^{n}\{V_{\mathsf{ID}_i}\}, V'_{\mathsf{ID}_A} = (h' + r)S_{\mathsf{ID}_A}\rangle$. $\mathcal{B}$ thus gets $V'_{\mathsf{ID}_A} - V_{\mathsf{ID}_A} = (h' - h)S_{\mathsf{ID}_A}$. k-CAA solution is $\frac{1}{(s+q_{\mathsf{ID}_A})}P = \frac{(x_{\mathsf{ID}_A}+y_{\mathsf{ID}_A})}{(h'-h)}(V'_{\mathsf{ID}_A} - V_{ID_A})$ since $S_{\mathsf{ID}_A} = \frac{1}{(x_{\mathsf{ID}_A}+y_{\mathsf{ID}_A})}\cdot\frac{1}{(s+q_{\mathsf{ID}_A})}P$.

Theorem 2. *Our CLRS scheme is existential unforgeable against the Type II adversary in the random oracle model assuming the mICDH is hard.*

Proof. Let $\mathcal{A}_{\mathcal{II}}$ be a forger that breaks the proposed signature scheme under adaptive chosen message and identity attack. We show that how $\mathcal{B}$ can use $\mathcal{A}_{\mathcal{II}}$ to solve the mICDH instance (P, aP, b) for randomly chosen $a, b \in_R \mathbb{Z}_q^*$ and $P \in \mathbb{G}_1$. Its goal is to compute $(a+b)^{-1}P$.

$\mathcal{B}$ sets $g = \hat{e}(P, P)$ and $P_{pub} = sP$ where s is the master secret key, which is chosen by $\mathcal{B}$. $\mathcal{B}$ then gives the system parameters to $\mathcal{A}_{\mathcal{II}}$. $\mathcal{B}$ randomly selects an index I such that $1 \leq I \leq q_{H_0}$, where q_{H_0} denotes the maximum number of queries to the random oracle H_0. Without loss of generality, we assume that any extraction ($\mathcal{PKGEN}$, `Request-Public-Key`, `Extract-Private-Key`) and $\mathcal{SIG}$ queries are preceded by H_0 query, and the $\mathcal{SIG}$ and `Extract-Private-Key` queries are preceded by `Request-Public-Key` query. $\mathcal{B}$ maintains four initially empty lists L_{H_0}, L_{H_1}, L_{H_2} and $L_K = \langle \mathsf{ID}, R_{ID}, x_{ID} \rangle$.

Adversary $\mathcal{B}$ interacts with $\mathcal{A}_{\mathcal{II}}$ in the `Attack` phase of the game as follows:

H_0 `Queries`: When $\mathcal{A}_{\mathcal{II}}$ queries H_0 on input ID_i, $\mathcal{B}$ checks the corresponding L_{H_0} and outputs q_{ID_i} if such value is defined. Otherwise, $\mathcal{B}$ picks $q_{\mathsf{ID}_i} \in_R \mathbb{Z}_q^*$ at random and outputs q_{ID_i} as answer. $\mathcal{B}$ computes $Q_{\mathsf{ID}_i} = sP + q_{\mathsf{ID}_i}P$ and adds $\langle \mathsf{ID}_i, Q_{\mathsf{ID}_i}, q_{\mathsf{ID}_i} \rangle$ to L_{H_0}.

H_1 `Queries`: When $\mathcal{A}_{\mathcal{II}}$ issues a query H_1 on $(m_i || u || L = \bigcup_{i=1}^{n} \mathsf{ID}_i || R = \bigcup_{i=1}^{n} R_{ID_i})$, $\mathcal{B}$ checks the corresponding L_{H_1} and outputs h_i if such value is defined. Otherwise, $\mathcal{B}$ picks $h_i \in_R \mathbb{Z}_q^*$ at random. $\mathcal{B}$ then outputs h_i as answer and adds $\langle m_i, u, L, R, h_i \rangle$ to L_{H_1}.

H_2 `Queries`: When $\mathcal{A}_{\mathcal{II}}$ queries H_2 on input R_{ID_i}, $\mathcal{B}$ checks the corresponding L_{H_2} and outputs y_{ID_i} if such value is defined. Otherwise, if $R_{\mathsf{ID}_i} = saP + q_{\mathsf{ID}_i}aP$, $\mathcal{B}$ sets $y_{\mathsf{ID}_i} = b$, else picks $y_{\mathsf{ID}_i} \in_R \mathbb{Z}_q^*$ at random. $\mathcal{B}$ then outputs y_{ID_i} as answer and adds $\langle R_{\mathsf{ID}_i}, y_{\mathsf{ID}_i} \rangle$ to L_{H_2}.

`Request-Public-Key`(ID_i): When $\mathcal{A}_{\mathcal{II}}$ queries on input ID_i, $\mathcal{B}$ finds $\langle \mathsf{ID}_i, Q_{ID_i}, q_{ID_i} \rangle$ in L_{H_0}. If no such query exists, $\mathcal{B}$ performs as follows:

1. If $\mathsf{ID}_i = \mathsf{ID}_A$, $\mathcal{B}$ returns $R_{\mathsf{ID}_i} = saP + q_{\mathsf{ID}_i}aP$ and adds $\langle \mathsf{ID}_i, R_{\mathsf{ID}_i}, \perp \rangle$ to L_K.
2. If $\mathsf{ID}_i \neq \mathsf{ID}_A$, $\mathcal{B}$ picks a random $x_{\mathsf{ID}_i} \in_R \mathbb{Z}_q^*$ and returns $R_{\mathsf{ID}_i} = x_{\mathsf{ID}_i} Q_{\mathsf{ID}_i}$. $\mathcal{B}$ adds $\langle \mathsf{ID}_i, R_{\mathsf{ID}_i}, x_{\mathsf{ID}_i} \rangle$ to L_K.

$\mathcal{PKGEN}$(ID_i): Note that at any time during the simulation, equipped with the master secret key s, $\mathcal{A}_{\mathcal{II}}$ is able to generate partial private key for any ID.

`Extract-Private-Key`(ID_i): When $\mathcal{A}_{\mathcal{II}}$ queries on input ID_i:

1. If $\mathsf{ID}_i = \mathsf{ID}_A$, $\mathcal{B}$ outputs `FAIL` and aborts the simulation.
2. If $\mathsf{ID}_i \neq \mathsf{ID}_A$, $\mathcal{B}$ finds $\langle \mathsf{ID}_i, Q_{\mathsf{ID}_i}, q_{\mathsf{ID}_i} \rangle$ in L_{H_0} and $\langle \mathsf{ID}_i, R_{\mathsf{ID}_i}, x_{\mathsf{ID}_i} \rangle$ in L_K. $\mathcal{B}$ performs as follows:
   – If the list L_{H_2} contains $\langle R_{\mathsf{ID}_i}, y_{\mathsf{ID}_i} \rangle$, $\mathcal{B}$ returns $S_{\mathsf{ID}_i} = (x_{\mathsf{ID}_i} + y_{\mathsf{ID}_i})^{-1} \frac{1}{s + q_{\mathsf{ID}_i}} P$.
   – If the list L_{H_2} does not contain $\langle R_{\mathsf{ID}_i}, y_{\mathsf{ID}_i} \rangle$, $\mathcal{B}$ makes query H_2 on input R_{ID_i} and returns $S_{\mathsf{ID}_i} = (x_{\mathsf{ID}_i} + y_{\mathsf{ID}_i})^{-1} \frac{1}{s + q_{\mathsf{ID}_i}} P$.

$\mathcal{SIG}(L, R, m)$: When $\mathcal{A_{II}}$ queries on input $(L = \bigcup_{i=1}^{n} \mathsf{ID}_i, R = \bigcup_{i=1}^{n} R_{\mathsf{ID}_i}, m)$, $\mathcal{B}$ finds $\langle \mathsf{ID}_i, Q_{\mathsf{ID}_i}, q_{\mathsf{ID}_i} \rangle$ in L_{H_0} and $\langle \mathsf{ID}_i, R_{\mathsf{ID}_i}, x_{\mathsf{ID}_i} \rangle$ in L_K for every ID and R_{ID}. $\mathcal{B}$ performs as follows:

1. Choose an index $A \in \{1, \ldots, n\}$.
2. Find $\langle R_{\mathsf{ID}_i}, y_{\mathsf{ID}_i} \rangle$ in $L_{H_2} \forall i \in \{1, 2, \cdots, n\}$. Let R_A denote the signer's public key and y_{ID_A} denotes the $H_2(R_{\mathsf{ID}_A})$. If it does not exist, $\mathcal{B}$ picks a random $y_{\mathsf{ID}_i} \in_R \mathbb{Z}_q^*$ and adds $\langle R_{ID_i}, y_{\mathsf{ID}_i} \rangle$ to L_{H_2}.
3. Choose $v_{\mathsf{ID}_i} \in_R \mathbb{Z}_q^*$, and compute $V_{\mathsf{ID}_i} = v_{\mathsf{ID}_i} P \ \forall i \in \{1, 2, \cdots, n\} \backslash \{A\}$.
4. Choose $r_A, h_A \in_R \mathbb{Z}_q^*$ and compute $V_{\mathsf{ID}_A} = r_A P$.
5. Compute $u = g^{-h_A} \cdot \hat{e}(P, \sum_{i \neq A} v_{\mathsf{ID}_i}(R_{\mathsf{ID}_i} + y_{\mathsf{ID}_i} Q_{\mathsf{ID}_i})) \cdot \hat{e}(R_{\mathsf{ID}_A}, r_A P) \cdot \hat{e}((y_{\mathsf{ID}_A})(s + q_{\mathsf{ID}_A})P, r_A P)$ and set $h_A = H_1(m, u, L, R)$. ($\mathcal{B}$ outputs `FAIL` and aborts the simulation if the $H_1(m, u, L, R)$ has already been defined in the list L_{H_1}).
6. Compute $V_{\mathsf{ID}_A} = r_A P$ and return $\sigma = \{u, \bigcup_{i=1}^{n} \{V_{\mathsf{ID}_i}\}\}$.

`Forgery`: The next step of the simulation is to apply the general forking lemma: Let $\langle u^*, L^*, \bigcup_{i \neq A}^{n} \{V_{\mathsf{ID}_i}\}, V_{\mathsf{ID}_A}(h_A + r_A) S_{ID_A} \rangle$ be a forgery of a signature on message m^* with respect to $\langle \mathsf{ID}_A, R_{\mathsf{ID}_A} \rangle$ that is output by $\mathcal{A_{II}}$ at the end of the attack. If $\mathcal{A_{II}}$ does not output $\mathsf{ID}^* = \mathsf{ID}_A$ as a part of the ring associated with the forgery then $\mathcal{B}$ aborts.

$\mathcal{B}$ then replays $\mathcal{A_{II}}$ with the same random tape but different H_1. Suppose H_1 outputs h and $h' \neq h$ in the first round and the second round respectively. We get another valid forgery $\langle u^*, L^*, \bigcup_{i \neq A}^{n} \{V_{\mathsf{ID}_i}\}, V'_{\mathsf{ID}} = (h' + r) S_{\mathsf{ID}_A} \rangle$. $\mathcal{B}$ thus gets $V'_{\mathsf{ID}_A} - V_{\mathsf{ID}_A} = (h' - h) S_{\mathsf{ID}_A}$. mICDH solution is $\frac{1}{a+b} P = \frac{(s + q_{\mathsf{ID}_A})}{(h' - h)} (V'_{\mathsf{ID}_A} - V_{ID_A})$ since $S_{\mathsf{ID}_A} = \frac{1}{a+b} \cdot \frac{1}{s + q_{\mathsf{ID}_A}} P$.

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

Proof. Each V_{ID_i} is a random element in $\mathbb{G}_1$, even the V component corresponding to the real signer, i.e. V_{ID_A} seems to be in a special form of $(h + r) S_{\mathsf{ID}_A}$. We can always find a r' such that $(h + r') S_{\mathsf{ID}_i} = V_{\mathsf{ID}_i}$ for any other members in the diversion group. Anonymity thus follows.

4.4 Discussion on Anonymity

Our proof gives the anonymity in theory. In practice, the KGC may only willing to generate one partial private key to the user, which means there is always a single valid public key for each user. For real anonymity, the signer should obtain the "correct" copy of the public key that each members in the diversion group is using. Otherwise, one can always repudiate being the signer of a certain ring signature by demonstrating the ability to give a normal signature with the knowledge of the private key that corresponding to a different public key.

One may argue that it essentially introduces some kind of "certificates" back to the system since the signer seems required to get some normal signature from each of the other $n - 1$ diversion group member, essentially $n - 1$ "certificates", to protect his/her anonymity. Our certificateless ring signatures may not posses the real spontaneousity enjoyed by identity-based ring signature. However, this assumption of getting "correct public keys" can be easily realized in other ways.

It is natural for the KGC to maintain a copy of all public keys of the user since all user ought to know how to contact the KGC. This is different from maintaining a certificate repository in traditional PKI setting since only public key but not certificates are stored. Even the KGC does not take this responsibility, that correct public key is generally available from each user, like from his/her personal homepage, since others need to get his/her public key to prepare encrypted messages or to verify the purported signature. However, the signer's retrieval of others public key may need to be made anonymous by other mean (e.g. proxy server providing anonymizing services), or the public key owner can guess who is the real signer of a certain signature.

5 Concluding Remarks

We propose the first non-trivial certificateless ring signature scheme, with detailed framework and security proofs. Our solution removes the high costs to deal with the transfer and verification of n certificates for a n-user ring signature under traditional public key infrastructure; at the same time our solution is free from key-escrow. Removing the complexity about certificates makes the scheme more applicable in ubiquitous computing environment.

A drawback of our system is that the signer needs to get the public keys for each member of the diversion group, which is our cost (beared by the signer) to get rid of certificates and key-escrow. However, it is still more reasonable than having the verifier to verify n certificates as the signer is motivated by his/her own privacy to collect the public keys. It seems this weakness is inherent in certificateless ring signatures. Nevertheless, it is worthy to see if we can achieve get all nice properties (certificateless, escrow-free, real spontaneous) at one shot. Another challenge is to extend the scheme to the security-medicated certificateless setting [9], which is a generalization of the certificateless paradigm that revocation is supported and the adversary can see the partial results generated from the partial private key.

References

1. Ben Adida, Susan Hohenberger and Ronald L. Rivest. Separable Identity-Based Ring Signatures: Theoretical Foundations For Fighting Phishing Attacks. DIMACS Workshop on Theft in E-Commerce: Content, Identity, and Service, 2005.
2. Sattam S. Al-Riyami and Kenneth G. Paterson. Certificateless Public Key Cryptography. In *AsiaCrypt 2003*, volume 2894 of LNCS, pp. 452-473, Springer.
3. Man Ho Au, Jing Chen, Joseph K. Liu, Yi Mu, Duncan S. Wong and Guomin Yang. Malicious KGC Attacks in Certificateless Cryptography. In *ASIACCS 2007*, pp. 302 - 311, ACM, 2007.
4. Dan Boneh and Matthew K. Franklin. Identity-Based Encryption from the Weil Pairing. In *CRYPTO 2001*, volume 2139 of LNCS, pp. 213-229, Springer, 2001.

5. Xuefei Cao, Kenneth G. Paterson and Weidong Kou. An Attack on Certificateless Signature Scheme. Available from `http://eprint.iacr.org/2006/367.pdf` .
6. Jung Hee Cheon. Security Analysis of the Strong Diffie-Hellman Problem. In *EuroCrypt 2006*, volume 4004 of LNCS, pp. 1-11, Springer, 2006.
7. Kyu Young Choi, Jong Hwan Park, Jung Yeon Hwang, and Dong HoonLee. Efficient Certificateless Signature Schemes. In *ACNS 2007*, volume 4521 of LNCS, Springer, 2007.
8. Sherman S.M. Chow. Identity-based Strong Multi-Designated Verifiers Signatures. In *EuroPKI 2006*, volume 4043 of LNCS, pp. 257-259, Springer, 2006.
9. Sherman S.M. Chow, Colin Boyd and Juan Manuel González Nieto. Security Mediated Certificateless Cryptography. In *PKC 2006*, volume 3958 of LNCS, pp. 508-524, Springer, 2006.
10. Sherman S.M. Chow, Richard W.C. Lui, Lucas C.K. Hui, and S.M. Yiu. Identity Based Ring Signature: Why, How and What Next. In *EuroPKI 2005*, volume 3545 of *LNCS*, pp. 144–161. Springer, 2006.
11. Sherman S.M. Chow and Duncan S. Wong. Anonymous Identification and Designated-Verifiers Signatures from Insecure Batch Verification. In *EuroPKI 2007*, volume 4582 of LNCS, pp. 203-219, Springer, 2007.
12. David Galindo, Javier Herranz, and Eike Kiltz. On the Generic Construction of Identity-Based Signatures with Additional Properties. In *AsiaCrypt 2006*, volume 4284 of *LNCS*, pages 178–193. Springer, 2006.
13. M. Choudary Gorantla and Ashutosh Saxena. An Efficient Certificateless Signature Scheme. In *CIS 2005*, volume 3802 of LNAI, pp. 110-116, Springer, 2005.
14. Bessie C. Hu, Duncan S. Wong, Zhenfeng Zhang, and Xiaotie Deng. Key Replacement Attack Against a Generic Construction of Certificateless Signature. In *ACISP 2006*, volume 4058 of LNCS, pp. 235-246, Springer, 2006.
15. Xinyi Huang, Willy Susilo, Yi Mu, and Futai Zhang. On the Security of Certificateless Signature Schemes from Asiacrypt 2003. In *CANS 2005*, volume 3810 of LNCS, pp. 13-25, Springer, 2005.
16. Markus Jakobsson, Kazue Sako and Russell Impagliazzo. Designated Verifier Proofs and Their Applications. In *EuroCrypt 1996*, volume 1070 of LNCS, pp. 143-154, Springer, 1996.
17. X. Li, K. Chen and L. Sun. Certificateless Signature and Proxy Signature Schemes from Bilinear Pairings. *Lithuanian Mathematical Journal*, Vol 45(1), pp. 76-83, Springer, 2005.
18. Joseph K. Liu, Man Ho Au, and Willy Susilo. Self-Generated-Certificate Public Key Cryptography and Certificateless Signature/Encryption Scheme in the Standard Model. In *ASIACCS 2007*, pp. 273-283, ACM, 2007.
19. Ronald L. Rivest, Adi Shamir, and Yael Tauman. How to Leak a Secret. In *AsiaCrypt 2001*, volume 2248 of LNCS, pp. 552-565, Springer, 2001.
20. Ronald L. Rivest, Adi Shamir, and Yael Tauman. How to Leak a Secret: Theory and Applications of Ring Signatures. In *Theoretical Computer Science: Essays in Memory of Shimon Even*, volume 3895 of LNCS, pp. 164-186, Springer, 2006.
21. Ryuichi Sakai and Masao Kasahara. ID based Cryptosystems with Pairing on Elliptic Curve. Cryptology ePrint Archive, Report 2003/054, 2003.
22. Adi Shamir. Identity-Based Cryptosystems and Signature Schemes. In *CRYPTO 1984*, volume 196 of LNCS, pp. 47-53, Springer, 1984.
23. Wun-She Yap, Sherman S.M. Chow, Swee-Huay Heng, and Bok-Min Goi. Security Mediated Certificateless Signatures. In *ACNS 2007*, volume 4521 of LNCS, pp. 459-477, Sprinegr, 2007.

24. Wun-She Yap, Swee-Huay Heng, and Bok-Min Goi. An Efficient Certificateless Signature Scheme. In *EUC 2006*, volume 4097 of LNCS, pp. 322-331, Springer.
25. Dae Hyun Yum and Phil Joong Lee. Generic Construction of Certificateless Signature. In *ACISP 2004*, volume 3108 of LNCS, pp. 200-211, Springer, 2004.
26. Fangguo Zhang, Reihaneh Safavi-Naini, and Willy Susilo. An Efficient Signature Scheme from Bilinear Pairings and Its Applications. In *PKC 2004*, volume 2947 of LNCS, pp. 277-290, Springer, 2004.
27. Zhenfeng Zhang, Duncan S. Wong, Jing Xu, and Dengguo Feng. Certificateless Public Key Signature: Security Model and Efficient Construction. In *ACNS 2006*, volume 3989 of LNCS, pp. 293-308, Springer, 2006.