Secure ID-Based Linkable and Revocable-iff-Linked Ring Signature with Constant-Size Construction

Man Ho Au[a,*], Joseph K. Liu[b], Willy Susilo[a,1], Tsz Hon Yuen[c]

[a] *Centre for Computer and Information Security Research*
School of Computer Science and Software Engineering
University of Wollongong, Australia
[b] *Cryptography and Security Department*
Institute for Infocomm Research
Singapore
[c] *Department of Computer Science*
University of Hong Kong, Hong Kong

Abstract

In this paper, we propose a new ID-based event-oriented linkable ring signature scheme, with an option as revocable-iff-linked. With this option, if a user generates two linkable ring signatures in the same event, everyone can compute his identity from these two signatures. We are *the first* in the literature to propose such a secure construction in ID-based setting. Even compared with other existing non ID-based schemes, we enjoy significant efficiency improvement, including constant signature size and linking complexity.

Our scheme can be also regarded as a normal ID-based ring signature. We are also the first to propose such a scheme with constant signature size *and* enhanced privacy, namely the signer is anonymous even to the PKG who has the master secret key.

We prove the security of our scheme in the random oracle model, using DL, DDL and q-SDH assumptions.

Keywords: Digital Signatures, ID-based cryptography, Anonymity, Ring

*Corresponding author.
Email addresses: aau@uow.edu.au (Man Ho Au), ksliu@i2r.a-star.edu.sg (Joseph K. Liu), wsusilo@uow.edu.au (Willy Susilo), thyuen@cs.hku.hk (Tsz Hon Yuen)

[1] This work is supported by ARC Future Fellowship FT0991397.

1. Introduction

Ring Signature. A ring signature scheme (such as [30, 1, 40, 8, 39, 17, 14]) allows members of a group to sign messages on behalf of the group without any necessity to reveal their identities, *i.e.*, providing signer anonymity. Additionally, it is impossible to decide whether two signatures have been issued by the same group member. In contrast to the notion of a group signature scheme (such as [11, 10, 5]), the group formation in a ring signature is spontaneous and there exists *no* group manager who is responsible for revoking the signer's identity. That is, under the assumption that each user is already associated with a public key of any standard signature scheme, a user can form a group by simply collecting the public keys of all the group members including his own. These diversion group members can be totally unaware of being conscripted into the group.

Applications of ring signature schemes include whistle blowing [30], anonymous membership authentication for ad hoc groups [9], non-interactive deniable ring authentication [32], perfect concurrent signature [33] and multi-designated verifiers signature [22].

A "regular" ring signature is unlinkable. That is, no one can determine whether two ring signatures are generated by the same signer.

Linkable Ring Signature. Linkable ring signatures was first proposed by Liu *et al.* [25] in 2004. In this notion, the identity of the signer in a ring signature remains anonymous, but two ring signatures can be linked if they are signed by the same signer. Linkable ring signatures are suitable in many different practical applications, such as ad-hoc network authentication [25], e-voting [12] and e-cash [36]. Regular ring signatures cannot be used for e-voting since any double votes remain undetectable as they are unlinkable. No one is able to find out whether any two signatures (with two votes) are generated by the same voter or not. Linkable ring signatures provide the remedy to this problem by allowing the public to detect any signer who has produced two or more signatures (*i.e.,* votes).

We note that linkability is compulsorily embedded into the signature instead of voluntarily added in linkable ring signatures. If the signer refuses to add the correct linking information, the whole signature becomes invalid. In other words, linkability is enforced by the verifier. The signer cannot decline

to do so. This is different from voluntarily added linkability. In this case, whether allowing the signature to be linked or not can be decided by the signer. This issue is also explained in [25].

Linkability can only happen within the same event (e.g., a voting event). Two signatures from two different events cannot be linked, even though they are generated by the same signer. Although the earlier schemes such as [25, 26, 27] do not mention about this property, they can be modified trivially to achieve this property.

Revocable Ring Signature. In a revocable ring signature [23], any member of the group is able to revoke the anonymity of the actual signer. This is different from group signature. For group signature, only the group manager has the power to do so. Furthermore, the formation of the group requires the assistance of the group manager. For revokable ring signature, since there is no group manager, the formation of the group still remains spontaneous, as in conventional ring signature. In other words, it explicitly provides all the group members the privilege of identifying the actual signer of any ring signature that has been generated on behalf of their group, while still keeping the actual signer anonymous to all the 'outsiders' (those who do not belong to the group).

Revocable-iff-Linked Ring Signature. A Revocable-iff-Linked ring signature [3, 19, 18] ([19, 18] described the same context as "Traceable Ring Signature") possesses the normal properties of a linkable ring signature. In addition, if two sigantures are linked (generated by the same user), the identity of this user will be revealed in the public. That is, his anonymity will be revoked if and only if he generates two ring signatures. Note that it is different from Revocable Ring Signature. In the latter, the anonymity of the actual user will be revoked *unconditionally* by other users within the ring. For other entities, the anonymity will be preserved in any circumstance.

There are many applications for revocable-iff-linked ring signature. It can be applied in anonymous e-voting scheme [12], unclonable group identification [15, 18] and can be extended to k-times anonymous authentication [34, 18].

The scheme proposed by Au *et al.* [3] is in identity-based setting. However, it was later proven insecure [21]. The other secure schemes [19, 18] are all in public-key based.

IDENTITY-BASED CRYPTOGRAPHY. Identity-based (ID-based) cryptosystem, introduced by Shamir [31], eliminated the need for checking the validity of the certificates. In an ID-based cryptosystem, public key of each user is easily computable from a string corresponding to this user's identity (e.g. an email address, a telephone number, etc.). A private key generator (PKG) then computes the private keys from a master secret for the users. This property avoids the necessity of certificates and associates an implicit public key (user identity) to each user within the system. Signature verification only requires the identity of the signer. The inefficient and costly certificates and the corresponding checking process are eliminated.

1.1. Challenge for ID-based Linkable Ring Signature

ID-based ring signature combines the property of ring signature and ID-based signature. However, we have to take extra care for the design of schemes. While some of the existing schemes provide anonymity *unconditionally*, others are computational only. The Private Key Generator (PKG) itself may have extra advantage in breaking the anonymity since it is in possession of all the private keys. This problem does not sound serious in normal ID-based ring signature scheme because almost all existing schemes is unconditionally anonymous. However, in the case of linkable ring signatures, it is still an open problem to construct one with unconditional anonymity. The inherent constraint is that anonymity in linkable ring signature cannot be *unconditional*. The capability of linking implies some the signature must contain some information about the actual signer. Within the constraint of computational anonymity, it is a great challenge of providing privacy in an ID-based setting to the PKG, as it is in possession of all users' secret keys. Given any ring signature, the PKG can create a set of linkable ring-signatures on behalf of each member in the ring for the same event. Next, the PKG can identify the actual signer of the given ring signature by checking which signature does the given signature is linked to. We require special attention in the design of the scheme. Looking ahead, we alter the semantic of traditional ID-based signature. In our proposal, the user obtains his/her secret key from the PKG via an interactive protocol in which that some part of the key obtained by the user is unknown to the PKG. This is to allow us to provide the proof of anonymity against the PKG under some well-established computational assumptions.

1.2. Contribution

We propose a new ID-based Linkable Ring Signature scheme with Revocable-iff-Linked as an option. Our scheme has the following merits:

1. There are only 2 *secure* ID-based Linkable Ring Signature schemes. When compared to the first one proposed by Chow *et al.* [13], our scheme achieves a higher level of anonymity. In our scheme, even the PKG cannot tell who is the actual signer. On the other side, the PKG in [13] can easily compute the signer of any given signature under any circumstance.
2. When compared to the second ID-based Linkable Ring Signature scheme proposed by Tsang *et al.* [35], our scheme achieves a constant size signature which is independent to the number of users included in the ring.
3. Since the one proposed in [3] is insecure [21], we are the first to propose a secure construction of ID-based Revocable-iff-Linked Ring Signature scheme. We prove the security using DL, DDH and q-SDH assumptions in the random oracle model.
4. When compared to the (non ID-based) Revocable-iff-Linked Ring Signature schemes [19, 18] (as known as traceable ring signature), our scheme achieves constant signature size while the scheme in [19] is with linear signature size and the scheme in [18] can achieve at most sub-linear size.

We summarize our merits in table 1.

1.2.1. Organization.

The rest of the paper is organized as follows: Some preliminaries will be presented in Section 2. We will describe our security model in Section 3. We are going to present our scheme in Section 4, followed by a formal security analysis in Section 5. Finally we conclude our paper in Section 6.

2. Preliminaries

2.1. Notations

Let $\hat{e}$ be a bilinear map such that $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$.

- $\mathbb{G}_1$ and $\mathbb{G}_2$ are cyclic multiplicative groups of prime order p.

Table 1: Comparison of Linkable Ring Signatures

Scheme	Signature Size	ID -based	anony. to PKG	revoke -iff-link	Linking Complexity
Liu *et al.* [24]	$O(n)$	×	N.A.	×	$O(1)$
Tsang and Wei [37]	$O(1)$	×	N.A.	×	$O(1)$
Liu and Wong [26]	$O(n)$	×	N.A.	×	$O(1)$
Au *et al.* [2]	$O(1)$	×	N.A.	×	$O(1)$
Zheng *et al.* [41]	$O(n)$	×	N.A.	×	$O(1)$
Tsang *et al.* [38]	$O(n)$	×	N.A.	×	$O(n^2)$
Tsang *et al.* [35]	$O(n)$	✓	✓	×	$O(n^2)$
Chow *et al.* [13]	$O(1)$	✓	×	×	$O(1)$
Fujisaki and Suzuki [19]	$O(n)$	×	N.A.	✓	$O(n)$
Fujisaki [18]	$O(\sqrt{n})$	×	N.A.	✓	$O(n \log n)$
Our Scheme	$O(1)$	✓	✓	✓	$O(1)$

- each element of $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ has unique binary representation.
- g_0, h_0 are generators of $\mathbb{G}_1$ and $\mathbb{G}_2$ respectively.
- $\psi : \mathbb{G}_2 \to \mathbb{G}_1$ is a computable isomorphism from $\mathbb{G}_2$ to $\mathbb{G}_1$, with $\psi(h_0) = g_0$.
- (Bilinear) $\forall x \in \mathbb{G}_1$, $y \in \mathbb{G}_2$ and a, $b \in \mathbb{Z}_p$, $\hat{e}(x^a, y^b) = \hat{e}(x, y)^{ab}$.
- (Non-degenerate) $\hat{e}(g_0, h_0) \neq 1$.

$\mathbb{G}_1$ and $\mathbb{G}_2$ can be the same or different groups. If the group operation in $\mathbb{G}_1$ and $\mathbb{G}_2$, the isomorphism ψ and the bilinear map $\hat{e}$ are all efficiently computable, the groups ($\mathbb{G}_1$, $\mathbb{G}_2$) are referred to as a bilinear group pair.

2.2. Mathematical Assumptions

Definition 1 (Discrete Logarithm)**.** *The Discrete Logarithm (DL) problem in $\mathbb{G}$ is defined as follows: On input a tuple $(Y, g) \in \mathbb{G}^2$ such that $|\mathbb{G}| = p$ for some prime p, output $a \in \mathbb{Z}_p$ such that $Y = g^a$. The advantage of an algorithm that solves the DL problem is the probability that it outputs a. We say that the (t, ϵ)-DL assumption holds in $\mathbb{G}$ if no t-time algorithm has advantage at least ϵ in solving the DL problem in $\mathbb{G}$.*

Definition 2 (Decisional Diffie-Hellman)**.** *The Decisional Diffie-Hellman (DDH) problem in* $\mathbb{G}$ *is defined as follows: On input a quadruple* $(g, h, g^a, T) \in \mathbb{G}^4$ *such that* $|\mathbb{G}| = p$ *for some prime* p*, output 1 if* $T = h^a$ *and 0 otherwise. The advantage of an algorithm that solves the DDH problem is the probability that it outputs the correct bit. We say that the* (t, ϵ)*-DDH assumption holds in* $\mathbb{G}$ *if no* t*-time algorithm has advantage at least* ϵ *over random guessing in solving the DDH problem in* $\mathbb{G}$*.*

Definition 3 (q-Strong Diffie-Hellman)**.** *The* q*-Strong Diffie-Hellman (*q*-SDH) problem in* $(\mathbb{G}_1, \mathbb{G}_2)$ *is defined as follows: On input a* $(q+2)$*-tuple* $(g_0, h_0, h_0^x, h_0^{x^2}, \cdots, h_0^{x^q}) \in \mathbb{G}_1 \times \mathbb{G}_2^{q+1}$*, output a pair* (A, c) *such that* $A^{(x+c)} = g_0$ *where* $c \in \mathbb{Z}_p^*$*. The advantage of an algorithm that solves the* q*-SDH problem is the probability that it outputs a tuple* (A, c)*. We say that the* (q, t, ϵ)*-SDH assumption holds in* $(\mathbb{G}_1, \mathbb{G}_2)$ *if no* t*-time algorithm has advantage at least* ϵ *in solving the* q*-SDH problem in* $(\mathbb{G}_1, \mathbb{G}_2)$*.*

The q-SDH assumption is shown to be true in the generic group model [7].

3. Security Model

3.1. Definition

The security definitions of ID-Based Linkable Ring Signature and ID-Based Revocable-iff-Linked Ring Signature are very similar. Therefore we describe the security notions of them together, and their differences are specified at appropriate places.

An ID-Based Linkable (or Revocable-iff-Linked) Ring Signature scheme is a tuple of probabilistic polynomial-time (PPT) algorithms and protocol below:

- **Setup**. On input an unary string 1^λ where λ is a security parameter, the algorithm outputs a master secret key s and a list of system parameters param that includes λ and the descriptions of a user secret key space $\mathcal{D}$, a message space $\mathcal{M}$ as well as a signature space Ψ.

- **Extract**. The common input of this interactive protocol between a user and the PKG is the system paratmers param and an identity $ID_i \in \{0,1\}^*$ for a user. The PKG has additional private input the master secret key s. Upon successful completion of the protocol, the

user outputs a secret key $d_i \in \mathcal{D}$. When we say identity ID_i corresponds to user secret key d_i or vice versa, we mean d_i is the output of the **Extract** protocol (at the user side) with respect to common input param and ID_i.

- **Sign**. On input a list param of system parameters, a group size n of length polynomial in λ, a set $\{ID_i \in \{0,1\}^* | i \in [1,n]\}$ of n user identities, a message $m \in \mathcal{M}$, an event identifier $event \in \{0,1\}^*$ and a secret key $\{d_j \in \mathcal{D} | j \in [1,n]\}$, the algorithm outputs an ID-based linkable (or revocable-iff-linked) ring signature $\sigma \in \Psi$.

- **Verify**. On input a list param of system parameters, a group size n of length polynomial in λ, a set $\{ID_i \in \{0,1\}^* | i \in [1,n]\}$ of n user identities, a message $m \in \mathcal{M}$, an event identifier $event \in \{0,1\}^*$ a signature $\sigma \in \Psi$, it outputs either valid or $\mathsf{invalid}$.

- **Link**. On input an event identifier $event \in \{0,1\}^*$ and two signatures $\sigma_1, \sigma_2 \in \Psi$, it outputs either link or unlink.

- **Revoke**. (For ID-based revocable-iff-linked ring signature only.) On input an event identifier $event \in \{0,1\}^*$ and two signatures $\sigma_1, \sigma_2 \in \Psi$ such that $\mathsf{link} \leftarrow \mathbf{Link}(\sigma_1, \sigma_2)$, it outputs ID.

Correctness. An ID-Based Linkable / Revocable-iff-Linked Ring Signature scheme should satisfy:

- *Verification Correctness* – Signatures signed by honest signers are verified to be valid.

- *Linking Correctness* – If two signatures are linked, they must be generated ***from the same secret key*** of the same signer.

- *Revoking Correctness* – For ID-Based Revocable-iff-Linked Ring Signature, it requires that the output of **Revoke** of two linked signatures must be the actual signer.

3.2. Security Requirement of ID-based Linkable Ring Signature

A secure ID-Based Linkable Ring Signature scheme should possess *unforgeability*, *anonymity*, *linkability* and *non-slanderability* which will be defined below.

3.2.1. Unforgeability.

An adversary should not be able to forge any signature just from the identities of the group members. We specify a security model which mainly captures the following two attacks:

1. Adaptive chosen message attack
2. Adaptive chosen identity attack

Adaptive chosen message attack allows an adversary to obtain message-signature pairs on demand during the forging attack. Adaptive chosen identity attack allows the adversary to forge a signature with respect to a group chosen by the adversary. To support adaptive chosen message attack, we provide the adversary the following oracle queries.

- **Extraction oracle** ($\mathcal{EO}$)**:** On input ID_i, $d_i \leftarrow$ **Extract**(param, ID_i, s) is returned . The oracle is stateful, meaning that if $ID_i = ID_j$, then $d_i = d_j$.
- **Signing oracle** ($\mathcal{SO}$)**:** $\mathcal{A}$ chooses a group of n identities $\{ID_i\}_{i\in[1,n]}$, a signer identity ID_j among them, an event identifier *event* and a message m, the oracle outputs a valid ID-based linkable (or revocable-iff-linked) ring signature denoted by $\sigma \leftarrow$ **Sign**($\mathsf{param}, n, \{ID_i | i \in [1, n]\}, m, event, d_j$). The signing oracle may query the extraction oracle during its operation.
- **Hash oracle** ($\mathcal{H}$)**:** $\mathcal{A}$ can ask for the values of the hash functions for any input.

We have the following unforgeability game:

1. A simulator $\mathcal{S}$ takes a sufficiently large security parameter λ and runs **Setup** to generate the public parameters param and master secret key s. The adversary $\mathcal{A}$ is given param.
2. $\mathcal{A}$ can make a polynomial number of oracle queries to $\mathcal{EO}$, $\mathcal{SO}$ and $\mathcal{H}$ adaptively.
3. $\mathcal{A}$ outputs a signature σ^* for message m^*, event $event^*$ and a set of identities L^*.

$\mathcal{A}$ wins the above game if

1. **Verify**($\mathsf{param}, |L^*|, L^*, m^*, event^*, \sigma^*$) = valid;

2. $(L^*, m^*, event^*)$ and σ^* should not be in the set of oracle queries and replies between $\mathcal{A}$ and $\mathcal{SO}$; and
3. $\mathcal{A}$ did not query $\mathcal{EO}$ on any identity $ID \in L^*$.

The advantage of $\mathcal{A}$ is defined as the probability that $\mathcal{A}$ wins.

Definition 4 (Unforgeability)**.** *A scheme is unforgeable if no PPT adversary has non-negligible advantage in winning the above game.*

3.2.2. L-Anonymity.

An adversary should not be able to tell the identity of the signer with a probability larger than $1/n$, where n is the cardinality of the ring. A crucial difference between Anonymity for ring signatures and L-Anonymity for linkable ring signatures is that in the latter, the adversary cannot query signatures of a user who appears in the challenge phase under the same event. The rationale is that if the adversary obtains any signature of user i, it can tell if the challenge signature is generated by this user due to the linking property.

Different from a non-ID-based linkable ring signature scheme, the PKG who knows the master secret key (thus it knows the secret key of every user), may gain advantage on the anonymity of a signature. In order to capture this potential attack, we enhance our model in a way that the adversary is also given the master secret key.

In order to capture the potential attack, we further define the following oracle:

- **Reversed Extraction oracle** ($\mathcal{REO}$)**:** The only difference between $\mathcal{REO}$ and the traditional $\mathcal{EO}$ is that, it is simulated by the adversary instead of the simulator. The initial request can be made by the adversary if the extracted protocol is an interactive one. In this case, the simulator acts as an honest user to provide interactions and the oracle records the necessary transcript of the interaction. Note that this maybe different from the final output of the interaction protocol due to some secret information which is only known to the honest user.

We have the following anonymity game:

1. A simulator $\mathcal{S}$ takes a sufficiently large security parameter λ and runs **Setup** to generate the public parameters param and master secret key s. The adversary $\mathcal{A}$ is given param and s.

2. $\mathcal{A}$ can make a polynomial number of oracle queries to $\mathcal{RE}\mathcal{O}$, $\mathcal{SO}$ and $\mathcal{H}$ adaptively.
3. In the challenge phase, $\mathcal{A}$ picks two identities ID_0^*, ID_1^*, which are not queried to the $\mathcal{SO}$ as a signer. $\mathcal{A}$ also picks a message m^*, an event $event^*$ and a set of n identities L^*. Then $\mathcal{A}$ receives a challenge signature $\sigma^* = \mathbf{Sign}(\mathsf{param}, n+2, L^* \cup \{ID_0^*, ID_1^*\}, m^*, event^*, d_{ID_b^*})$, where $b \in \{0,1\}$.
4. $\mathcal{A}$ can queries oracles $\mathcal{RE}\mathcal{O}$, $\mathcal{SO}$ and $\mathcal{H}$ adaptively, where $(ID_0^*, event^*)$ and $(ID_1^*, event^*)$ are not queried to the $\mathcal{SO}$.
5. Finally $\mathcal{A}$ outputs a guess $b' \in \{0,1\}$.

$\mathcal{A}$ wins the above game if $b = b'$. The advantage of $\mathcal{A}$ is defined as the probability that $\mathcal{A}$ wins minus 1/2.

Definition 5 (Anonymity)**.** *A scheme is anonymous if no PPT adversary has non-negligible advantage in winning the above game.*

Note 1: Although the adversary has the master secret key and it can generate an additional secret key for ID_0^* or ID_1^*, this secret key is different from the one owned by ID_0^* or ID_1^* (generated by $\mathcal{RE}\mathcal{O}$). According to our definition of *Linking Correctness*, those signatures generated by these two secret keys cannot be linked, although they are corresponding to the same identity.

3.2.3. Linkability.

An adversary should not be able to form two signatures ***with the same secret key*** in the same event without being linked by the **Link** protocol. We further generalize the notion that an adversary with k secret keys cannot create $k+1$ signatures in the same event without being linked.

We have the following linkability game:

1. A simulator $\mathcal{S}$ takes a sufficiently large security parameter λ and runs **Setup** to generate the public parameters param and master secret key s. The adversary $\mathcal{A}$ is given param.
2. $\mathcal{A}$ can make a polynomial number of oracle queries to $\mathcal{E}\mathcal{O}$, $\mathcal{SO}$ and $\mathcal{H}$ adaptively.
3. $\mathcal{A}$ outputs an event $event^*$, a positive integer k, a set of signatures σ_i^* for messages m_i^* and sets of identities L_i^* for $i = 0$ to k.

Let C be the set of identities queried to $\mathcal{E}\mathcal{O}$. $\mathcal{A}$ wins the above game if:

- σ_i for $i = 0$ to k are not outputs from $\mathcal{SO}$.
- **Verify**$(\mathsf{param}, |L_i^*|, L_i^*, m_i^*, event^*, \sigma_i^*) = \mathsf{valid}$ for $i = 0$ to k;
- **Link**$(\sigma_i^*, \sigma_j^*) = \mathsf{Unlink}$ for all $i, j \in \{0, \ldots, k\}$ such that $i \neq j$; and
- $|(L_0^* \cup L_1^* \cup \ldots \cup L_k^*) \cap C| \leq k$.

The advantage of $\mathcal{A}$ is defined as the probability that $\mathcal{A}$ wins.

Definition 6 (Linkability)**.** *A scheme is linkable if no PPT adversary has non-negligible advantage in winning the above game.*

3.2.4. Non-slanderability.

Informally speaking, non-slanderability ensure that no adversary, can frame an honest user for signing a signature. That is, an adversary cannot produce a valid signature that is linked to a signature generated by a user. In addition to the above oracles, we define one more:

- **Challenged Signing oracle** ($\mathcal{CSO}$)**:** The only difference between $\mathcal{CSO}$ and the traditional $\mathcal{SO}$ is that, it requires the simulator to use the secret key queried from the $\mathcal{RSO}$ and execute **Sign** algorithm specified in the scheme to generate the signature. $\mathcal{RSO}$ should be queried before if necessary.

Formally it is defined as follows:

1. A simulator $\mathcal{S}$ takes a sufficiently large security parameter λ and runs **Setup** to generate param and master secret key s. $\mathcal{S}$ sends param and s to the adversary $\mathcal{A}$.
2. $\mathcal{A}$ makes a polynomial number of oracle queries to $\mathcal{RSO}$ and $\mathcal{H}$ in an adaptive manner.
3. $\mathcal{A}$ submits a polynomial number of oracle queries to $\mathcal{CSO}$ adaptively for generating challenged signatures.
4. $\mathcal{A}$ outputs a signature σ^* for message m^*, event $event^*$ and ring L^*.

$\mathcal{A}$ wins the game if

- **Verify**$(\mathsf{param}, |L^*|, L^*, m^*, event^*, \sigma^*)$ returns valid.
- σ^* is not an output of any $\mathcal{CSO}$ query.

- $\mathbf{Link}(\sigma^*, \hat{\sigma}) = \mathsf{Link}$ where $\hat{\sigma}$ is any signature outputted from $\mathcal{CSO}$.

Definition 7 (Non-slanderability)**.** *A scheme is non-slanderability if no PPT adversary has non-negligible advantage in winning the above game.*

Note 2: Although the adversary may initialize the query of $\mathcal{REO}$, it cannot get the user secret key since it does not know some secret information which is only known to the honest user (that is, the simulator in this game). Thus it cannot generate a signature by that particular secret key which is linked together with some signatures outputted by $\mathcal{CSO}$. In addition, the remark of Note 1 also applies here.

Theorem 1. *For an ID-based linkable ring signature scheme, if it is linkable and non-slanderable, it implies that it is unforgeable.*

Proof. The proof is by contradiction. Suppose there exists an adversary $\mathcal{A}$ who can win game unforgeability with non-negligible probability. We show how to construct an adversary $\mathcal{A}'$ that can win game linkability or game non-slanderability with blackbox access to $\mathcal{A}$.

In order for $\mathcal{A}$ to win, it must output a valid signature σ^* for message m^*, event $event^*$ and a set of identities L^*. Due to the setting of the game, $\mathcal{A}$ has not issued any $\mathcal{EO}$ query with input $ID \in L^*$. Let $m \neq m^*$ be a random message and $\sigma_i = \mathbf{Sign}(\mathsf{param}, |L^*|, L^*, m, event^*, d_i)$, where $d_i = \mathbf{Extract}(\mathsf{param}, ID_i, s)$. The forger $\mathcal{A}$ can be classified into two types according to the characteristic of σ^* with respect to these σ_i's.

1. Type I. There exists an i such that $\mathsf{link} \leftarrow \mathbf{Link}(\sigma^*, \sigma_i, event^*)$.
2. Type II. For all i $\mathsf{unlink} \leftarrow \mathbf{Link}(\sigma^*, \sigma_i, event^*)$.

For type I forger $\mathcal{A}$, we show how to construct $\mathcal{A}'$ that wins game non-slanderability. For type II forger $\mathcal{A}$, we show how to construct $\mathcal{A}'$ that wins game linkability. The view provided by $\mathcal{A}'$ to $\mathcal{A}$ is indistinguishable in these two cases. Thus, if $\mathcal{A}$ can win with non-negligible probability in game unforfeability, $\mathcal{A}'$ either win game non-slanderability or game linkability with non-negligible probability.

Type I Forger. $\mathcal{A}'$ receives param and s from $\mathcal{S}$ in game non-slanderability. $\mathcal{A}'$ forwards param to $\mathcal{A}$. For each $\mathcal{EO}$ query made by $\mathcal{A}$, $\mathcal{A}'$ uses the master secret key s to answer the query. For each $\mathcal{SO}$ query made by $\mathcal{A}$, $\mathcal{A}'$ made an $\mathcal{REO}$ query.

Finally, $\mathcal{A}$ submits a forgery σ^* for message m^*, event $event^*$ and a set of identities L^*. If σ^* links to any of the output from the previous $\mathcal{REO}$ query, $\mathcal{A}'$ wins the game directly. Otherwise, $\mathcal{A}'$ randomly picks an index $i^* \in \{1, \ldots, |L^*|\}$ and issue an $\mathcal{REO}$ to the simulator on input ID_{i^*}. Next, $\mathcal{A}$ issues a $\mathcal{CSO}$ query on input a random message $m \neq m^*$, event event^*, ring L^* and index i^* to obtain a signature σ. Since $\mathcal{A}$ is a type I forger, with probability $1/|L^*|$, $\mathsf{link} \leftarrow \mathbf{Link}(\sigma^*, \sigma, event^*)$. $\mathcal{A}'$ submits $\sigma^*, m^*, event^*, L^*$ and wins game non-slanderability.

Type II Forger. $\mathcal{A}'$ receives param from $\mathcal{S}$ in game linkability. $\mathcal{A}'$ forwards param to $\mathcal{A}$. For each $\mathcal{EO}$ query made by $\mathcal{A}$, $\mathcal{A}'$ forwards the query to $\mathcal{S}$. Likewise, for each $\mathcal{SO}$ query made by $\mathcal{A}$, $\mathcal{A}'$ forwards the query to $\mathcal{S}$.

Finally, $\mathcal{A}$ submits a forgery σ^* for message m^*, event $event^*$ and a set of identities L^*. $\mathcal{A}'$ randomly picks an index $i^* \in \{1, \ldots, |L^*|\}$ and issue an $\mathcal{EO}$ query to the simulator on input ID_{i^*} to obtain a secret key d_{i^*}. $\mathcal{A}'$ randomly picks a message $m \neq m^*$, computes $\sigma = \mathbf{Sign}(\mathsf{param}, |L^*|, L^*, m, event^*, d_{i^*})$. Since $\mathcal{A}$ is a type II forger, $\mathsf{unlink} \leftarrow \mathbf{Link}(\sigma^*, \sigma, event^*)$. $\mathcal{A}'$ then outputs $k = 1$, $event^*$, $\sigma_0^* = \sigma^*$, $\sigma_1^* = \sigma$, $m_0^* = m^*$, $m_1^* = m$ and win game linkability. □

3.3. Security Requirement of ID-based Revocable-iff-Linked Ring Signature

The definitions of unforgeability and anonymity are the same as ID-based Linkable Ring Signature defined in Section 3.2. We skip here.

3.3.1. Revoke-iff-Linkability.

An adversary should not be able to form two signatures with the same secret key in the same event without being linked by the **Link** protocol or the **Revoke** algorithm outputs an identity outside the ring. We further generalize the notion to the case that an adversary with k secret keys cannot create $k + 1$ signatures in the same event without being linked.

We have the following linkability game:

1. A simulator $\mathcal{S}$ takes a sufficiently large security parameter λ and runs **Setup** to generate the public parameters param and master secret key s. The adversary $\mathcal{A}$ is given param.
2. $\mathcal{A}$ can make a polynomial number of oracle queries to $\mathcal{EO}$, $\mathcal{SO}$ and $\mathcal{H}$ adaptively.
3. $\mathcal{A}$ outputs an event $event^*$, a positive integer k, a set of signatures σ_i^* for messages m_i^* and sets of identities L_i^* for $i = 0$ to k.

Let C be the set of identities queried to $\mathcal{EO}$. $\mathcal{A}$ wins the above game if it fulfils either condition:

1.
   - σ_i for $i = 0$ to k are not outputs from $\mathcal{SO}$.
   - $\mathbf{Verify}(\mathsf{param}, |L_i^*|, L_i^*, m_i^*, event^*, \sigma_i^*) = \mathsf{valid}$ for $i = 0$ to k;
   - $\mathbf{Link}(\sigma_i^*, \sigma_j^*) = \mathsf{Unlink}$ for all $i, j \in \{0, \ldots, k\}$ such that $i \neq j$; and
   - $|(L_0^* \cup L_1^* \cup \ldots \cup L_k^*) \cap C| \leq k$.

   OR

2.
   - there exists $i, j \in \{0, \ldots, k\}$ such that σ_i and σ_j are not outputs from $\mathcal{SO}$.
   - $\mathbf{Verify}(\mathsf{param}, |L_b^*|, L_b^*, m_b^*, event^*, \sigma_b^*) = \mathsf{valid}$ for $b \in \{i, j\}$;
   - $\mathbf{Link}(\sigma_i^*, \sigma_j^*) = \mathsf{Link}$; and
   - $\mathbf{Revoke}(\sigma_i^*, \sigma_j^*, event^*) = ID'$ where $ID' \notin \{L_i^* \cup L_j^*\}$ or ID' has not been inputted to $\mathcal{EO}$.

The advantage of $\mathcal{A}$ is defined as the probability that $\mathcal{A}$ wins.

Definition 8 (Revoke-iff-Linkability)**.** *A scheme is revocable-iff-linked if no PPT adversary has non-negligible advantage in winning the above game.*

3.3.2. Non-slanderability in revocable-iff-linked ring signatures.

Definition of framing attack in ID-based revocable-iff-linked ring signature covers two aspects of framing. The first one is that no attacker should be able to create a signature that is linked to a signature created by an honest signer. The second one is that no attacker should be able to create two signatures so that when they are input to the algorithm **Revoke**, an identity of an honest user will be the output. Formally, non-slanderability of revocable-iff-linked ring signatures includes the one defined above in Section 3.2 (Def. 7) and the definition of Revoke-iff-linkability (Def. 8).

Definition 9 (Non-slanderability for revocable-iff-linked ring signatures)**.** *A revocable-iff-linked ring signatures is non-slanderable if no PPT adversary has non-negligible advantage in winning the games defined in Def. 7 and Def. 8.*

4. Our Proposed Schemes

4.1. Building Blocks

Zero-Knowledge Proof-of-Knowledge. Our construction utilizes extensively the well-established non-interactive zero-knowledge proof-of-knowledge protocols. In a zero-knowledge proof-of-knowledge protocol [20] (ZKPoK), a prover convinces a verifier that some statement is true while the verifier learns nothing except the validity of the statement. We follow the notation introduced by Camenisch and Stadler [10] in which $\mathsf{PK}\{(x) : y = g^x\}$ denotes a ZKPoK protocol that proves the knowledge of an integer x such that $y = g^x$ holds. A large class of ZKPoK protocols, called Σ-protocol, can be transformed into secure signature scheme in the random oracle model. The resulting signature is often called signature of knowledge. Following the same notation, the signature scheme corresponds to the protocol $\mathsf{PK}\{(x) : y = g^x\}$ will be denoted as $\mathsf{SPK}\{(x) : y = g^x\}(M)$. The Σ-protocol is inspired by the protocols described in [4].

Accumulators. An accumulator allows the representation of a set of elements, say, $\mathcal{X} = \{x_1, \ldots, x_n\}$ by a short value v. For each element x in the set $\mathcal{X}$, there exists a corresponding value w_x, called witness. Given v, x and w_x, anyone can verify efficiently whether x is an element in the set that produces the accumulator value v. Accumulator was first introduced in [6]. Dodis et al. [16] illustrates how an accumulator can be used to construct constant-size ring signatures. Roughly, their idea is as follows. The signer produces a short value v, which is the accumulation of the set of public keys in the ring. Next, he produces an non-interactive proof of knowledge of the tuple (x, w_x), which assures the verifier that x is one element of the set of the public keys in the ring. Finally, the signer also creates a non-interactive zero-knowledge proof-of-knowledge of the secret key with respect to that public key x. To verify the ring signature, the verifier checks if v is indeed the result of the accumulation of all public keys in the ring as well as the non-interactive proofs. Ngyuen [28] proposed an accumulator in groups equipped with bilinear maps and construct a constant-size ring signature in the ID-based setting. Our construction also makes use of accumulator to achieve constant-size signature in a similar manner.

4.2. Construction

In this sub-section, we detail our construction.

- **Setup:**

  1. *Init (Common parameter):* Let λ be the security parameter. Let $\hat{e}$ be a bilinear pairing as described in Section 2.1 and $(\mathbb{G}_1, \mathbb{G}_2)$ be a bilinear group pair with computable isomorphism ψ such that $|\mathbb{G}_1| = |\mathbb{G}_2| = p$ for some prime p of λ bits. Also let $\mathbb{G}_p$ be a group of order p where DDH is intractable[2] We define $H : \{0,1\}^* \to \mathbb{Z}_p$, $G_0 : \{0,1\}^* \to \mathbb{G}_p$ and $G_1 : \{0,1\}^* \to \mathbb{G}_p$ be some cryptographic hash functions. Let g_0, g_1, g_2 be generators of $\mathbb{G}_1$ and h_0, h_1, h_2 be generators of group $\mathbb{G}_2$ such that $\psi(h_i) = g_i$ for $i = 0, 1, 2$.
  2. *Init (Accumulator):* Choose a generator h of $\mathbb{G}_2$. Randomly select $q \in_R \mathbb{Z}_p^*$ and compute $q_i = h^{(q^i)}$ for $i = 1 \cdots t_{max}$, where t_{max} is the maximum size of the ring.
  3. *PKG Setup:* The PKG randomly selects $\gamma \in_R \mathbb{Z}_p^*$ and compute $w = {h_0}^{\gamma}$. The master secret is γ while the public parameters are $(H, G_0, G_1, \psi, \hat{e}, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_p, p, g_0, g_1, g_2, h_0, h_1, h_2, h, q_1, \ldots, q_{t_{max}}, w)$.

- **Extract:** User with identity ID_u obtain the corresponding secret key from PKG through the following interactive protocol.

  1. Let $e = H(ID_u)$. User with identity ID_u randomly selects $s', r_s \in_R \mathbb{Z}_p^*$, compute and sends $C' = g_1^{s'} g_2^{r_s}$ to the PKG. The user also conducts the following zero-knowledge proof-of-knowledge protocol with the PKG.

  $$\mathsf{PK}\{(s', r_s) : C' = g_1^{s'} g_2^{r_s}\}$$

  2. If PKG accepts the proof, it randomly selects $s'' \in_R \mathbb{Z}_p^*$ and computes

  $$C = C' g_1^{s''}, \qquad A = (g_0 C)^{\frac{1}{e+\gamma}}$$

  and sends (A, s'') to the user.
  3. User computes $s = s' + s''$ and checks if $\hat{e}(A, w h_0^e) = \hat{e}(g_0 g_1^s g_2^{r_s}, h_0)$. It then stores (A, s, r_s).

We only allow each user to obtain one secret key.

[2]In practice, one could set $\mathbb{G}_p$ to be $\mathbb{G}_T$ or a subgroup of the multiplicative group modulo a prime p' where $p | p' - 1$.

- **Sign(Link Version):** To sign a message m with a set of identities $L = |ID_1, \ldots, ID_n|$, where $n \leq t_{max}$, using the secret key of a user with identity ID_u (where $u \in \{1, n\}$) in an event $event$, first compute $e = H(ID_u)$, $u_0 = G_0(event)$ and

$$v = \psi(h^{\prod_{k=1}^{k=n}\left(q+H(ID_k)\right)}), \qquad v_w = \psi(h^{\prod_{k=1,k\neq u}^{k=n}\left(H(ID_k)+q\right)}), \qquad S = {u_0}^s.$$

Note that although the user does not know q, he can still compute v as

$$\begin{aligned} v &= \psi(h^{\prod_{k=1}^{k=n}\left(q+H(ID_k)\right)}) \\ &= \psi(h^{\left(q+H(ID_1)\right)\ldots\left(q+H(ID_n)\right)}) \\ &= \psi(h^{q^n+c_{n-1}q^{n-1}+\ldots+c_1q+c_0}) \\ &= \psi(h^{q^n}(h^{q^{n-1}})^{c_{n-1}}\ldots(h^q)^{c_1}h^{c_0}) \\ &= \psi(q_n q_{n-1}^{c_{n-1}}\ldots q_1^{c_1}h^{c_0}) \qquad (1) \end{aligned}$$

where $c_0, \ldots, c_{n-1} \in Z_p$ are some coefficients. v_w can be computed in a similar way.

Denote $M = m||L||event$. The user further computes the following non-interactive signature-of-knowledge Π_0 which assures the verifier the following relationship are satisfied.

$$\mathsf{SPK}\left\{ (A, e, s, r_s, v_w) : \begin{array}{rrrcl} & & \hat{e}(A, wh_0^e) & = & \hat{e}(g_0 g_1^s g_2^{r_s}, h_0) \\ & \wedge & \hat{e}(v_w, q_1 h^e) & = & \hat{e}(v, h) \\ & \wedge & S & = & {u_0}^s \end{array} \right\}(M)$$

The SPK Π_0 can be efficiently transformed into a a discrete-log relation SPK that is easily instantiated, by randomly generating some variables $r_1, r_2, r_e \in_R \mathbb{Z}_p^*$ and computing

$$A_1 = {g_1}^e {g_2}^{r_e}, \quad A_2 = A{g_2}^{r_1}, \quad A_3 = v_w {g_2}^{r_2},$$
$$\alpha_1 = r_1 e, \quad \alpha_2 = r_2 e, \quad \alpha_3 = r_1 r_e, \quad \alpha_4 = r_2 r_e.$$

Denote by M' the string $m||L||event||A_1||A_2||A_3$ and let $1_{\mathbb{G}_1}$ be the identity element of $\mathbb{G}_1$. The signer produces the following non-interactive

zero-knowledge proof-of-knowledge Π_1:

$$\mathsf{SPK}\left\{ \begin{pmatrix} r_1, r_2, \\ r_e, e, \\ s, r_s, \\ \alpha_1, \alpha_2, \\ \alpha_3, \alpha_4 \end{pmatrix} : \begin{array}{lrcl} \wedge & A_1 & = & g_1^e g_2^{r_e} \\ \wedge & 1_{\mathbb{G}_1} & = & A_1^{-r_1} g_1^{\alpha_1} g_2^{\alpha_3} \\ \wedge & 1_{\mathbb{G}_1} & = & A_1^{-r_2} g_1^{\alpha_2} g_2^{\alpha_4} \\ \wedge & S & = & u_0^s \\ \wedge & \frac{\hat{e}(A_2,w)}{\hat{e}(g_0,h_0)} & = & \frac{\hat{e}(g_1,h_0)^s \hat{e}(g_2,h_0)^{r_s} \hat{e}(g_2,w)^{r_1} \hat{e}(g_2,h_0)^{\alpha_1}}{\hat{e}(A_2,h_0)^e} \\ \wedge & \frac{\hat{e}(A_3,q_1)}{\hat{e}(v,h)} & = & \hat{e}(g_2,q_1)^{r_2} \hat{e}(g_2,h)^{\alpha_2} \hat{e}(A_3,h)^{-e} \end{array} \right\}(M')$$

The linkable ring signature on message m in event *event* with ring L is parsed as $(S, A_1, A_2, A_3, \Pi_2)$. Note that S is the linkability tag. Π_1 consists of 11 elements of Z_p. Details of Π_1 can be found in Appendix A.

- **Sign(Revocable-iff-Link Version):** It is the same as above except adding an extra component T. Specifically, first create a signature as in the link version $(S, A_1, A_2, A_3, \Pi_1)$. Then compute

  $$R = H(S||A_1||A_2||A_3||\Pi_2||m||L||event),\ \ u_1 = G_1(event),\ \ T = u_0^e (u_1^R)^s$$

  and make a proof that T is correctly formed.

  This can be done via the following non-interactive zero-knowledge proof-of-knowledge Π_2:

  $$\mathsf{SPK}\left\{ (e, r_e, s) : \begin{array}{lrcl} & A_1 & = & g_1^e g_2^{r_e} \\ \wedge & S & = & u_0^s \\ \wedge & T & = & {u_0}^e (u_1^R)^s \end{array} \right\}(R)$$

  The revocable-iff-link ring signature on message m in event *event* with ring L is parsed as $(S, T, A_1, A_2, A_3, \Pi_1, \Pi_2)$. Note Π_1 and Π_2 consists of 11 and 4 elements of Z_p respectively. It is straightforward to combine Π_1 and Π_2 into a single proof consisting of 12 elements. We choose to present them separately for charity. Details of Π_2 can be found in Appendix A.

- **Verify:** First compute $u_0 = G_0(event)$ and v as in equation (1).

  In the link version, parse the signature as $(S, A_1, A_2, A_3, \Pi_1)$. Verify the SPK Π_1.

In the revocable-iff-link version, parse the signature as $(S, T, A_1, A_2, A_3, \Pi_1, \Pi_2)$. Compute $u_1 = G_1(event)$, $R = H(S||A_1||A_2||A_3||\Pi_1||m||L||event)$ and verify the SPK Π_1 and Π_2.

- **Link:** Two valid signatures are linked in the same event if they share the same link tag S.

- **Revoke:** Given two signatures $(S, T, A_1, A_2, A_3, \Pi_1, \Pi_2)$ and $(S, T', A'_1, A'_2, A'_3, \Pi'_1, \Pi'_2)$ for an event *event* on message m from the set of identities L and another message m' from another set of identities L' respectively, compute $R = H(S||A_1||A_2||A_3||\Pi_1||m||L||event)$ and $R' = H(S||A'_1||A'_2||A'_3||\Pi'_1||m'||L'||event)$. Compute $u_0 = G_0(event)$ and $U = (\frac{T^{R'}}{T'^{R}})^{(\frac{1}{R'-R})}$. For each identity ID in $L \cap L'$, output ID if and only if $U = u_0^{H(ID)}$.

5. Security Analysis

We present the security analysis of our constructions in this section. Regarding the security of our schemes, we have the following theorem.

Theorem 2. *Our scheme Link version (resp. Revocalbe-iff-Link version) possesses unforgeability, anonymity, linkability (resp. revocable-iff-linkability) and non-slanderability if the q-SDH assumption holds in $(\mathbb{G}_1, \mathbb{G}_2)$, the DDH and DL assumptions hold in $\mathbb{G}_p$ in the random oracle model.*

The Revocable-iff-Link Version can be regarded as a generalization of the Link Version. Thus we only show the security analysis of the Revocable-iff-Link Version. In rest of this section, "our scheme" refers to the proposed ID-Based Revocable-iff-Link Ring Signature scheme.

To prove Theorem 2, we have to show our scheme satisfies definition 5, 7, 8. Then from Theorem 1, the scheme will be unforgeable and that it will be non-slanderable according to definition 9.

Lemma 1. *Our scheme satisfies definition 5 (anonymous) if the DDH assumption in $\mathbb{G}_p$ holds in the random oracle model.*

Proof. The proof is by reduction. Suppose there exists an adversary $\mathcal{A}$ that has non-negligible advantage in wining game anonymity, we construct an algorithm $\mathcal{A}'$ that solves the DDH problem. Since it is widely believed that

the DDH problem is hard, it implies no algorithm $\mathcal{A}$ exists and this complete the proof of the lemma.

Below we show how to construct algorithm $\mathcal{A}'$. $\mathcal{A}'$ is given an instance of the DDH problem, $(u_0, u_1, u_0^{\tilde{s}}, Z)$ and its task is to determine if $Z \stackrel{?}{=} u_1^{\tilde{s}}$.

$\mathcal{A}'$ creates the public parameters param and the master secret key s according to the setup algorithm. param and s are given to the adversary $\mathcal{A}$.

Suppose $\mathcal{A}$ makes q_H queries to the hash oracle $\mathcal{H}$ with input ID_i for $i = 1$ to q_H. $\mathcal{A}'$ randomly picks an index $i^* \in_R \{1, \ldots, q_H\}$ and all queries related to ID_{i^*} are handled differently. Likewise, assume $\mathcal{A}$ makes q_{G_0} and q_{G_1} hash queries to hash oracle G_0 and G_1 respectively. $\mathcal{A}'$ randomly picks two indexes $j^* \in_R \{1, \ldots, q_{G_0}\}$ and $k^* \in_R \{1, \ldots, q_{G_1}\}$. For input $event_j$, if $j \neq j^*$, $\mathcal{A}'$ returns $(u_0^s)^{\mu_j}$ for some random $\mu_j \in_R \mathbb{Z}_p$. For $j = j^*$, $\mathcal{A}'$ returns $u_1^{\mu_{j^*}}$ for some random $\mu_{j^*} \in_R \mathbb{Z}_p$. Likewise, For input $event_k$, if $k \neq k^*$, $\mathcal{A}'$ returns $(u_0^s)^{\nu_k}$ for some random $\nu_k \in_R \mathbb{Z}_p$. For $k = k^*$, $\mathcal{A}'$ returns $u_1^{\nu_{k^*}}$ for some random $\nu_{k^*} \in_R \mathbb{Z}_p$.

For queries not related to ID_{i^*}, $\mathcal{A}'$ answers the query following the specification. For $\mathcal{RECO}$ query related to ID_{i^*}, $\mathcal{A}'$ picks a random value $C' \in_R \mathbb{G}_1$ and employs the zero-knowledge simulator to simulate proof-of-knowledge. For signature query related to ID_{i^*}, $\mathcal{A}'$ simulates the reply as if the secret key obtained in the $\mathcal{RECO}$ query for ID_{i^*} is (A, e, s, r_s). $\mathcal{A}'$ locates the hash query for $event$ for the signature such that $event = event_j$ and $event = event_k$ for some indexes j and k. $\mathcal{A}'$ proceeds only if $j \neq j^* \wedge k \neq k^*$. $\mathcal{A}'$ sets the value of S and T to be $(u_0^s)^{\mu_j}$ and $u_0^e(u_0^s)^{R\nu_k}$ respectively. Simulation is possible because of the zero-knowledgeness of Π_1 and Π_2. Note that in the view of the adversary, it is entirely correct as for any value s, there exists a value r_s such that $C = C' g_1^{s''} = g_1^s g_2^{r_s}$.

Then at the challenge phase, $\mathcal{A}$ submits two identities ID_0^* and ID_1^*, a message m^*, an event $event^*$ and a set of n identities L^*. $\mathcal{A}'$ picks $b \in_R \{0, 1\}$. If $ID_b^* \neq ID_{i^*}$ $\mathcal{A}'$ aborts and outputs a random guess. In addition, if $event^* \neq event_{j^*}$ or $event^* \neq event_{k^*}$, $\mathcal{A}'$ aborts and outputs a random guess.

In case $\mathcal{A}'$ does not abort. It creates a challenge signature as if the secret key it obtains for ID_{i^*} is is (A, e, s, r_s). $\mathcal{A}'$ sets the value of S and T to be $(Z)^{\mu_{j^*}}$ and $u_0^e(Z)^{R\nu_{k^*}}$ respectively. Note that in the view of the adversary, it is entirely correct as for any value s, there exists a value r_s such that $C = C' g_1^{s''} = g_1^s g_2^{r_s}$. $\mathcal{A}'$ then sets the value A_1, A_2, A_3 as three random values in $\mathbb{G}_1$. Note that for any value of e, there exists a value r_e such that

$A_1 = g_1^e g_2^{r_e}$. Likewise, for any values A and v_w, there exists r_1, r_2 such that $A_2 = A g_2^{r_1}$ and $A_3 = v_w g_2^{r_2}$. Thus, (A_1, A_2, A_3) is correctly distributed in the view of the adversary. $\mathcal{A}'$ then simulates the zero-knowledge proofs Π_1 and Π_2 using the values S, T, A_1, A_2, A_3.

If $\mathcal{A}$ finally outputs $b' = b$, then $\mathcal{A}'$ outputs 1 for the DDH problem. Otherwise, $\mathcal{A}'$ outputs 0.

Note that if $Z = u_1^{\tilde{s}}$, the challenge signature is a correct signature produced by the secret key (A, e, s, r_s) which belongs to ID_b^*. Otherwise, it contains no information about ID_b^* in the view of the adversary and thus based on the success of $\mathcal{A}$, $\mathcal{A}'$ can solve the DDH problem.

□

Lemma 2. *Our scheme satisfies definition 7 (non-slanderable) if the DL assumption holds in $\mathbb{G}_p$ in the random oracle model.*

Proof. The proof is by reduction. Suppose there exists an adversary $\mathcal{A}$ that has non-negligible advantage in wining game non-slanderability, we construct an algorithm $\mathcal{A}'$ that solves the DL problem. Since it is widely believed that the DL problem is hard, it implies no algorithm $\mathcal{A}$ exists and this complete the proof of the lemma.

Below we show how to construct algorithm $\mathcal{A}'$. $\mathcal{A}'$ is given an instance of the DL problem (S^*, u^*) and its task is to output s^* such that $S^* = (u^*)^{s^*}$.

$\mathcal{A}'$ creates the public parameters param and the master secret key s according to the setup algorithm. param and s are given to the adversary $\mathcal{A}$.

Suppose $\mathcal{A}$ makes q_H queries to the hash oracle $\mathcal{H}$ with input ID_i for $i = 1$ to q_H. $\mathcal{A}'$ randomly picks an index $i^* \in_R \{1, \ldots, q_H\}$ and all queries related to ID_{i^*} are handled differently.

For all hash queries to hash oracle G_0 and G_1, $\mathcal{A}'$ randomly picks a number $\mu \in_R \mathbb{Z}_p$ and returns $(u^*)^\mu$ as the hash value.

For queries not related to ID_{i^*}, $\mathcal{A}'$ answers the query following the specification. For $\mathcal{REO}$ query related to ID_{i^*}, $\mathcal{A}'$ picks a random value $C' \in_R \mathbb{G}_1$ and employs the zero-knowledge simulator to simulate proof-of-knowledge.

For $\mathcal{CSO}$ queries related to ID_{i^*}, $\mathcal{A}'$ simulates the reply as if the secret key obtained in the $\mathcal{REO}$ query for ID_{i^*} is (A, e, s, r_s). $\mathcal{A}'$ locates the hash query for *event* for the signature such that $G_0(event) = (u^*)^\mu$ and $G_1(event) = (u^*)^{\mu'}$. $\mathcal{A}'$ sets the value of S and T to be $(S^*)^\mu$ and $u_0^e (S^*)^{R\mu'}$ respectively. Simulation is possible because of the zero-knowledgeness of Π_1 and Π_2. Note that in the view of the adversary, it is entirely correct as for any value s, there exists a value r_s such that $C = C' g_1^{s''} = g_1^s g_2^{r_s}$.

Finally, $\mathcal{A}$ returns a valid signature σ^*, which is not the output from $\mathcal{CSO}$, but is linked to one of them. If it is linked to the signature created from ID_{i^*}, $\mathcal{A}'$ rewinds and extracts the SPK Π_1 to obtain s^*. Then $\mathcal{A}'$ returns s^* as the solution of the DL problem. □

Lemma 3. *Our scheme satisfies definition 8 (revocable-iff-linked) if the $q_{\mathcal{H}}$-SDH assumption in holds in $(\mathbb{G}_1, \mathbb{G}_2)$ in the random oracle model, where $q_{\mathcal{H}} + 1$ is the number of hash queries made by $\mathcal{A}$.*

Proof. The proof is by reduction. Suppose there exists an adversary $\mathcal{A}$ that has non-negligible advantage in wining game revocable-iff-linkability, we construct an algorithm $\mathcal{A}'$ that solves the $q_{\mathcal{H}}$-SDH problem. Under the $q_{\mathcal{H}}$-SDH assumption, this implies no $\mathcal{A}$ exists and this completes the proof of the lemma.

Below we show how to construct algorithm $\mathcal{A}'$. $\mathcal{A}'$ is given an instance of the $q_{\mathcal{H}}$-SDH problem $(g_1', g_2', {g_2'}^x, \ldots, {g_2'}^{x^{q_{\mathcal{H}}}})$ and its task is to output a tuple $(e, {g_1'}^{\frac{1}{x+e}})$.

$\mathcal{A}'$ creates the public parameters param as follows. $\mathcal{A}'$ randomly picks $e_1, \ldots e_{q_{\mathcal{H}}-1} \in_R \mathbb{Z}_p^*$. Denote $f(x)$ as the polynomial $\prod_{i=1}^{q_{\mathcal{H}}-1}(x + e_i)$. Assume $x \neq -e_i$ for all i. If $x = -e_i$ for some i, $\mathcal{A}'$ solves the $q_{\mathcal{H}}$-SDH problem directly. Next, $\mathcal{A}'$ computes the following values.

$$h_0 = {g_2'}^{f(x)}, \qquad w = {g_2'}^{xf(x)}, \qquad g_0 = \psi(h_0).$$

While x is unknown to $\mathcal{A}'$, it possesses its powers in the form of ${g_2'}^{x^i}$ and thus it is possible for $\mathcal{A}'$ to compute the values h_0 and w. Then, $\mathcal{A}'$ picks $e^*, a^*, k^* \in \mathbb{Z}_p^*$ and computes:

$$h_1 = [(w h_0^{e^*})^{k^*} h_0^{-1}]^{1/a^*} = h_0^{\frac{(e^*+x)k^*-1}{a^*}}, \qquad g_1 = \psi(h_1).$$

$\mathcal{A}'$ randomly picks $\mu \in \mathbb{Z}_p^*, h \in \mathbb{G}_2$, sets $g_2 = g_0^{\mu}$. Other public parameters are created following the specification. The public parameters param is given to the adversary $\mathcal{A}$.

For $i = 1$ to $q_{\mathcal{H}} - 1$, $\mathcal{A}'$ computes

$$A_i = g_0^{1/x+e_i} = \psi({g_2'}^{f(x)/x+e_i}).$$

WLOG, we denote $e^* = e_{q_{\mathcal{H}}}$, $A_{q_{\mathcal{H}}} = g_0^{k^*}$ and $e_0 \in_R \mathbb{Z}_p^*$. The set $\{(A_i, e_i)_{i=1}^{q_{\mathcal{H}}}\}$ and e_0 are keep private to $\mathcal{A}'$ and are used to answer the oracle queries.

Recall that we assume $\mathcal{A}$ makes $q_{\mathcal{H}}+1$ queries to the hash oracle $\mathcal{H}$. For each hash query to oracle $\mathcal{H}$, $\mathcal{A}'$ chooses an index $i \in \{0,1,\ldots,q_{\mathcal{H}}\}$ and returns e_i.

For each $\mathcal{EO}$ query on identity ID, we assume there exists an index $i \in \{0,\ldots,q_{\mathcal{H}}\}$ and a hash query such that $e_i = H(ID)$. Otherwise $\mathcal{A}'$ made the hash query itself. If $i=0$, $\mathcal{A}'$ aborts. Otherwise, $\mathcal{A}'$ extracts the value (s', r_s) from C'. This is possible due to the soundness of the zero-knowledge proof protocol. For $i \neq q_{\mathcal{H}}$, $\mathcal{A}'$ randomly picks $s'' \in \mathbb{Z}_p^*$ and computes:

$$\begin{aligned}
A &= (g_0 C g_2^{r_s})^{1/x+e_i} \\
&= (g_0^{1+r_s\mu+\frac{(s'+s'')[(e^*+x)k^*-1]}{a^*}})^{1/x+e_i} \\
&= A_i^{1+r_s\mu-\frac{(s'+s'')}{a^*}} g_0^{\frac{(s'+s'')k^*(e^*+x)}{a^*(e_i+x)}} \\
&= A_i^{(1+r_s\mu-\frac{(s'+s'')}{a^*})} \big(g_0^{\frac{(s'+s'')k^*}{a^*}}\big)^{(1-\frac{e_i-e^*}{e_i+x})} \\
&= A_i^{(1+r_s\mu-\frac{(s'+s'')}{a^*}-\frac{(s'+s'')k^*(e_i-e^*)}{a^*})} \big(g_0^{\frac{(s'+s'')k^*}{a^*}}\big)
\end{aligned}$$

$\mathcal{A}'$ returns (A, e_i, s'') to $\mathcal{A}$.

For $i = q_{\mathcal{H}}$, $\mathcal{A}'$ returns $(A_{q_{\mathcal{H}}} = g_0^{k^*}, e_{q_{\mathcal{H}}}, s'' = a^* - s')$ to $\mathcal{A}$. The value $s = s' + s''$ in each of these queries are known to $\mathcal{A}'$. Denote the value s as s_i and r_s as r_{s_i} in the query with index i. $\mathcal{A}'$ stores (A_i, e_i, s_i, r_{s_i}) alongside i.

For $\mathcal{SO}$ query, $\mathcal{A}'$ answers these queries using the zero-knowledge simulator of Π_1 and Π_2.

Finally, $\mathcal{A}$ returns $k+1$ valid signatures σ_j^* on message m_j^* and ring L_j^* for event $event^*$ for $j=0$ to k, where k is a positive integer. Let L^* be the union of the rings L_j^*.

Under the soundness of the SPK, each signature σ_j^* corresponds to a set of witnesses satisfying the following relationships.

1. $A^{e+x} = g_0 g_1^s g_2^{r_s}$
2. $v_w^{e+q} = v$
3. $S = {u_0}^s$
4. $T = {u_0}^e u_1^{Rs}$

Assume $\mathcal{A}$ wins by condition 1 of definition 8, then $\mathcal{A}$ is in possession of at most k secret keys in L^*. Under the assumption that the accumulator is secure [29], $e = H(ID)$ for some $ID \in L^*$. Furthermore, since all these $k+1$ signatures are not linked, they have different values in s. Under the discrete

logarithm assumption, the value of $g_0 g_1^s g_2^{r_s} \neq g_0 g_1^{s'} g_2^{r'_s}$ if $s \neq s'$. This implies there exists $k+1$ distinct pairs of (A, e) in these $k+1$ valid signatures. $\mathcal{A}'$ randomly picks an index j and extracts the witnesses (A, e, s, r_s, v_w) from the SPK Π_1, Π_2 in σ_j^*. Let $e = H(ID)$. With probability at least $1/(k+1)$, one of the following two cases is true.

- Case 1: ID has not been input to $\mathcal{EO}$. With probability $1/(q_{\mathcal{H}}+1)$, $e = e_0$. If not, $\mathcal{A}'$ aborts. Otherwise, $\mathcal{A}'$ computes:

$$B = \left(A g_0^{-\frac{sk^*}{a^*}}\right)^{\frac{a^*}{a^* + r_s \mu a^* - s - sk^*(e_0 - e^*)}}$$

  Note that since σ_j^* is a valid signature,

$$\begin{aligned} A^{e_0+x} &= g_0 g_1^s g_2^{r_s} = g_0^{1 + r_s\mu + \frac{s[(e^*+x)k^*-1]}{a^*}} \\ A &= \left(g_0^{\frac{a^* + r_s \mu a^* - s}{a^*(e_0+x)}}\right)\left[\left(g_0^{\frac{sk^*}{a^*}}\right)^{(1 - \frac{e_0 - e^*}{e_0+x})}\right] \\ \therefore B &= g_0^{\frac{1}{x+e_0}} \end{aligned}$$

- Case 2: ID has been queried to $\mathcal{EO}$ and that there exists i such that $e = e_i$ but $A \neq A_i$. With probability $1/(q_{\mathcal{H}}+1)$, $e = e^*$. If not, $\mathcal{A}'$ aborts. Otherwise, $\mathcal{A}'$ computes:

$$B = \left(A g_0^{-\frac{sk^*}{a^*}}\right)^{\frac{a^*}{a^* + r_s \mu a^* - s}}$$

  Note that since σ_j^* is a valid signature,

$$\begin{aligned} A^{e^*+x} &= g_0 g_1^s g_2^{r_s} = g_0^{1 + r_s\mu + \frac{s[(e^*+x)k^*-1]}{a^*}} \\ A &= \left(g_0^{\frac{a^* + r_s \mu a^* - s}{a^*(e_0+x)}}\right)\left(g_0^{\frac{sk^*}{a^*}}\right) \\ \therefore B &= g_0^{\frac{1}{x+e^*}} \end{aligned}$$

In both cases, $\mathcal{A}'$ obtains a tuple (B, e) such that $B = g_0^{\frac{1}{x+e}}$. $\mathcal{A}'$ computes the polynomial $a(x)$ of degree $q_{\mathcal{H}} - 2$ such that $f(x) = (x+e)a(x) + b$ for some $b \neq 0 \in \mathbb{Z}_p$ (since $e \notin \{e_1, \ldots, e_{q_{\mathcal{H}}-1}\}$). Recall that $g_0 = \psi(g_2'^{f(x)}) = g_1'^{f(x)}$. Thus, $B = g_0^{\frac{1}{x+e}} = g_1'^{a(x)} g_1'^{\frac{b}{x+e}}$. $\mathcal{A}'$ outputs $(e, (B g_1'^{-a(x)})^{\frac{1}{b}})$ as the solution to the $q_{\mathcal{H}}$-SDH problem.

Assume $\mathcal{A}$ wins by condition 2 of definition 8. WLOG, we let the two signatures outputted by $\mathcal{A}$ and linked together are $\dot{\sigma}^*$ and $\ddot{\sigma}^*$ on event $event^*$ and that $ID' = \textbf{Revoke}(\dot{\sigma}^*, \ddot{\sigma}^*, event^*)$. Due to the soundness of the SPK, the two signatures are the proof-of-knowledge of two sets of witnesses $(\dot{A}, \dot{e}, \dot{s}, \dot{r_s}, \dot{v_w})$ and $(\ddot{A}, \ddot{e}, \ddot{s}, \ddot{r_s}, \ddot{v_w})$ respectively. If $\dot{e} = \ddot{e}$, there exists i such that $\dot{e} = \ddot{e} = e_i$ (since $\dot{e}, \ddot{e}$ must be the output of the random oracle $\mathcal{H}$). Due to the soundness of the SPK and the correctness of the **Revoke** algorithm, $e_i = H(ID')$. Under the assumption that the accumulator is secure [29], $ID' \in L^*$. In that case, $\mathcal{A}$ wins only if ID' has not been submitted to $\mathcal{EO}$. With probability $1/(q_{\mathcal{H}} + 1)$, $e_i = e_0$. $\mathcal{A}'$ solves the $q_{\mathcal{H}}$-SDH problem as in case 1 in the previous condition.

Next, consider the case when $\dot{e} \neq \ddot{e}$. Let $\dot{e} = H(\dot{ID})$ and $\ddot{e} = H(\ddot{ID})$ respectively. If $\dot{ID}$ (or $\ddot{ID}$) has not been submitted to $\mathcal{EO}$, with probability $1/(q_{\mathcal{H}} + 1)$, $\dot{e} = e_0$ (or $\ddot{e} = e_0$). In that case, $\mathcal{A}'$ solves the $q_{\mathcal{H}}$-SDH problem as in case 1 in the previous condition.

Finally, consider the case that both $\dot{ID}$ and $\ddot{ID}$ have been submitted to the $\mathcal{EO}$ oracle. In that case, either Let $\dot{i}$ and $\ddot{i}$ be the indexes such that $e_{\dot{i}} = H(\dot{ID})$ and $e_{\ddot{i}} = H(\ddot{ID})$. Since $\dot{\sigma}^*$ and $\ddot{\sigma}^*$ are linked, $\dot{s} = \ddot{s}$. On the other hand, the values $s_{\dot{i}}$ and $s_{\ddot{i}}$ in the corresponding $\mathcal{EO}$ queries are different with overwhelming probability. The reason is that the value s_i in the an $\mathcal{EO}$ query depends on the random number chosen by $\mathcal{A}'$ and that in the query with index $q_{\mathcal{H}}$, the value s^* is hidden in and information-theoretic manner before the query. WLOG, assume $s_{\dot{i}} \neq \dot{s}$. Under the discrete logarithm assumption, if $s_{\dot{i}} \neq \dot{s}$, the value of $g_0 {g_1}^{s_{\dot{i}}} {g_2}^{r_{s_{\dot{i}}}} \neq g_0 {g_1}^{\dot{s}} {g_2}^{\dot{r_s}}$. Thus, $\dot{A} \neq A_{\dot{i}}$. With probability $1/(q_{\mathcal{H}} + 1)$, $\dot{e} = e_{q_{\mathcal{H}}}$. In that case, $\mathcal{A}'$ solves the $q_{\mathcal{H}}$-SDH problem as in case 1 in the previous condition. □

6. Conclusion

In this paper, we proposed a new construction of ID-based Linkable Ring Signature, with an option as Revocable-iff-Linked. Taking this option, it is the first secure construction in the literature. When compared to other non ID-based schemes, we enjoy a significant efficiency advantages: Both the signature size and the linking complexity are constant, independent to the number of users in the ring. When compared to other non revocable ID-based linkable ring signature schemes, we also achieve a higher level of anonymity: The PKG cannot tell the actual signer of any signature, despite it has the secret key of every user. We prove the security of our scheme in

the random oracle model. It is still an open problem to construct a scheme with such desirable features and can be proven secure in the standard model.

References

[1] Masayuki Abe, Miyako Ohkubo, and Koutarou Suzuki. 1-out-of-n Signatures from a Variety of Keys. In *ASIACRYPT 2002*, volume 2501 of *Lecture Notes in Computer Science*, pages 415–432. Springer, 2002.

[2] Man Ho Au, Sherman S. M. Chow, Willy Susilo, and Patrick P. Tsang. Short linkable ring signatures revisited. In *EuroPKI*, volume 4043 of *Lecture Notes in Computer Science*, pages 101–115. Springer, 2006.

[3] Man Ho Au, Joseph K. Liu, Willy Susilo, and Tsz Hon Yuen. Constant-size id-based linkable and revocable-iff-linked ring signature. In *INDOCRYPT*, volume 4329 of *Lecture Notes in Computer Science*, pages 364–378. Springer, 2006.

[4] Man Ho Au, Willy Susilo, and Yi Mu. Constant-Size Dynamic k-TAA. In Roberto De Prisco and Moti Yung, editors, *SCN*, volume 4116 of *Lecture Notes in Computer Science*, pages 111–125. Springer, 2006.

[5] Mihir Bellare, Daniele Micciancio, and Bogdan Warinschi. Foundations of Group Signatures: Formal Definitions, Simplified Requirements, and a Construction Based on General Assumptions. In *EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 614–629. Springer, 2003.

[6] Josh Cohen Benaloh and Michael de Mare. One-way accumulators: A decentralized alternative to digital sinatures (extended abstract). In *EUROCRYPT*, pages 274–285, 1993.

[7] Dan Boneh and Xavier Boyen. Short signatures without random oracles. In *EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 56–73. Springer, 2004.

[8] Dan Boneh, Craig Gentry, Ben Lynn, and Hovav Shacham. Aggregate and Verifiably Encrypted Signatures from Bilinear Maps. In *EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 416–432. Springer, 2003.

[9] Emmanuel Bresson, Jacques Stern, and Michael Szydlo. Threshold ring signatures and applications to ad-hoc groups. In Moti Yung, editor, *CRYPTO 2002*, volume 2442 of *Lecture Notes in Computer Science*, pages 465–480. Springer, 2002.

[10] Jan Camenisch and Markus Stadler. Efficient Group Signature Schemes for Large Groups (Extended Abstract). In *CRYPTO 97*, volume 1294 of *Lecture Notes in Computer Science*, pages 410–424. Springer, 1997.

[11] David Chaum and Eugène van Heyst. Group signatures. In Donald W. Davies, editor, *EUROCRYPT 1991*, volume 547 of *Lecture Notes in Computer Science*, pages 257–265. Springer, 1991.

[12] Sherman S. M. Chow, Joseph K. Liu, and Duncan S. Wong. Robust receipt-free election system with ballot secrecy and verifiability. In *NDSS*. The Internet Society, 2008.

[13] Sherman S. M. Chow, Willy Susilo, and Tsz Hon Yuen. Escrowed linkability of ring signatures and its applications. In *VIETCRYPT*, volume 4341 of *Lecture Notes in Computer Science*, pages 175–192. Springer, 2006.

[14] Sherman S. M. Chow, Siu-Ming Yiu, and Lucas Chi Kwong Hui. Efficient Identity Based Ring Signature. In *ACNS 2005*, volume 3531 of *Lecture Notes in Computer Science*, pages 499–512, 2005. Also available at Cryptology ePrint Archive, Report 2004/327.

[15] Ivan Damgård, Kasper Dupont, and Michael Østergaard Pedersen. Unclonable group identification. In *EUROCRYPT*, volume 4004 of *Lecture Notes in Computer Science*, pages 555–572. Springer, 2006.

[16] Y. Dodis, A. Kiayias, A. Nicolosi, and V. Shoup. Anonymous Identification in Ad Hoc Groups. In *EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 609–626. Springer, 2004.

[17] Yevgeniy Dodis, Aggelos Kiayias, Antonio Nicolosi, and Victor Shoup. Anonymous Identification in Ad Hoc Groups. In *EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 609–626. Springer, 2004.

[18] Eiichiro Fujisaki. Sub-linear size traceable ring signatures without random oracles. In *CT-RSA*, volume 6558 of *Lecture Notes in Computer Science*, pages 393–415. Springer, 2011.

[19] Eiichiro Fujisaki and Koutarou Suzuki. Traceable ring signature. In *Public Key Cryptography*, volume 4450 of *Lecture Notes in Computer Science*, pages 181–200. Springer, 2007.

[20] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM J. Comput.*, 18(1):186–208, 1989.

[21] Ik Rae Jeong, Jeong Ok Kwon, and Dong Hoon Lee. Analysis of revocable-iff-linked ring signature scheme. *IEICE Transactions*, 92-A(1):322–325, 2009.

[22] Fabien Laguillaumie and Damien Vergnaud. Multi-designated Verifiers Signatures. In *ICICS 2004*, volume 3269 of *Lecture Notes in Computer Science*, pages 495–507, Malaga, Spain, October 2004. Springer.

[23] Dennis Y. W. Liu, Joseph K. Liu, Yi Mu, Willy Susilo, and Duncan S. Wong. Revocable ring signature. *J. Comput. Sci. Technol.*, 22(6):785–794, 2007.

[24] J. K. Liu, V. K. Wei, and D. S. Wong. Linkable spontaneous anonymous group signature for ad hoc groups (extended abstract). In *ACISP'04*, volume 3108 of *LNCS*, pages 325–335. Springer, 2004.

[25] Joseph K. Liu, Victor K. Wei, and Duncan S. Wong. Linkable Spontaneous Anonymous Group Signature for Ad Hoc Groups (Extended Abstract). In *ACISP '04*, volume 3108 of *LNCS*, pages 325–335. Springer, 2004.

[26] Joseph K. Liu and Duncan S. Wong. Linkable ring signatures: Security models and new schemes. In *ICCSA (2)*, volume 3481 of *Lecture Notes in Computer Science*, pages 614–623. Springer, 2005.

[27] Joseph K. Liu and Duncan S. Wong. Enhanced security models and a generic construction approach for linkable ring signature. *Int. J. Found. Comput. Sci.*, 17(6):1403–1422, 2006.

[28] Lan Nguyen. Accumulators from bilinear pairings and applications. In Alfred Menezes, editor, *CT-RSA*, volume 3376 of *Lecture Notes in Computer Science*, pages 275–292. Springer, 2005.

[29] Lan Nguyen. Accumulators from Bilinear Pairings and Applications. In A. J. Menezes, editor, *CT-RSA 2005*, volume 3376 of *Lecture Notes in Computer Science*, pages 275–292. Springer, 2005.

[30] Ronald L. Rivest, Adi Shamir, and Yael Tauman. How to Leak a Secret. In *ASIACRYPT 2001*, volume 2248 of *Lecture Notes in Computer Science*, pages 552–565. Springer, 2001.

[31] Adi Shamir. Identity-Based Cryptosystems and Signature Schemes. In *CRYPTO 84*, volume 196 of *Lecture Notes in Computer Science*, pages 47–53. Springer, 1984.

[32] Willy Susilo and Yi Mu. Non-Interactive Deniable Ring Authentication. In *ICISC 2003*, volume 2971 of *Lecture Notes in Computer Science*, pages 386–401. Springer, 2004.

[33] Willy Susilo, Yi Mu, and Fangguo Zhang. Perfect Concurrent Signature Schemes. In *ICICS 2004*, volume 3269 of *Lecture Notes in Computer Science*, pages 14–26. Springer, October 2004.

[34] Isamu Teranishi, Jun Furukawa, and Kazue Sako. k-times anonymous authentication (extended abstract). In *ASIACRYPT*, volume 3329 of *Lecture Notes in Computer Science*, pages 308–322. Springer, 2004.

[35] Patrick P. Tsang, Man Ho Au, Joseph K. Liu, Willy Susilo, and Duncan S. Wong. A Suite of Non-Pairing ID-Based Threshold Ring Signature Schemes with Different Levels of Anonymity. In *ProvSec 2010*, volume 6402 of *Lecture Notes in Computer Science*, pages 166–183. Springer, 2010.

[36] Patrick P. Tsang and Victor K. Wei. Short linkable ring signatures for e-voting, e-cash and attestation. In *ISPEC 2005*, volume 3439 of *Lecture Notes in Computer Science*, pages 48–60. Springer, 2005.

[37] Patrick P. Tsang and Victor K. Wei. Short linkable ring signatures for e-voting, e-cash and attestation. In *ISPEC 2005*, volume 3439 of *Lecture Notes in Computer Science*, pages 48–60. Springer, 2005.

[38] Patrick P. Tsang, Victor K. Wei, Tony K. Chan, Man Ho Au, Joseph K. Liu, and Duncan S. Wong. Separable linkable threshold ring signatures. In *INDOCRYPT 2004*, Lecture Notes in Computer Science, pages 384–398. Springer, 2004.

[39] Duncan S. Wong, Karyin Fung, Joseph K. Liu, and Victor K. Wei. On the rs-code construction of ring signature schemes and a threshold setting of rst. In *ICICS 2003*, volume 2836 of *Lecture Notes in Computer Science*, pages 34–46. Springer, 2003.

[40] Fangguo Zhang and Kwangjo Kim. ID-Based Blind Signature and Ring Signature from Pairings. In *ASIACRYPT 2002*, volume 2501 of *Lecture Notes in Computer Science*, pages 533–547. Springer, 2002.

[41] Dong Zheng, Xiangxue Li, Kefei Chen, and Jianhua Li. Linkable ring signatures from linear feedback shift register. In *EUC Workshops*, volume 4809 of *Lecture Notes in Computer Science*, pages 716–727. Springer, 2007.

Appendix A. Details of SPK Π_1 and Π_2

Details of Π_1. The signer randomly generates ρ_{r_1}, ρ_{r_2}, ρ_{r_e}, ρ_{r_s}, ρ_e, ρ_s, ρ_{α_1}, ρ_{α_2}, ρ_{α_3}, $\rho_{\alpha_4} \in_R \mathbb{Z}_p$ and computes

$$\begin{aligned}
\mathfrak{T}_1 &= g_1^{\rho_e} g_2^{\rho_{r_e}} \\
\mathfrak{T}_2 &= A_1^{-\rho_{r_1}} g_1^{\rho_{\alpha_1}} g_2^{\rho_{\alpha_3}} \\
\mathfrak{T}_3 &= A_1^{-\rho_{r_2}} g_1^{\rho_{\alpha_2}} g_2^{\rho_{\alpha_4}} \\
\mathfrak{T}_4 &= u_0^{\rho_s} \\
\mathfrak{T}_5 &= \frac{\hat{e}(g_1, h_0)^{\rho_s} \hat{e}(g_2, h_0)^{\rho_{rs}} \hat{e}(g_2, w)^{\rho_{r_1}} \hat{e}(g_2, h_0)^{\rho_{\alpha_1}}}{\hat{e}(A_2, h_0)^{\rho_e}} \\
\mathfrak{T}_6 &= \frac{\hat{e}(g_2, q_1)^{\rho_{r_2}} \hat{e}(g_2, h)^{\rho_{\alpha_2}}}{\hat{e}(A_3, h)^{\rho_e}}
\end{aligned}$$

Next, the signer computes $c = H(\mathfrak{T}_1||\mathfrak{T}_2||\mathfrak{T}_3||\mathfrak{T}_4||\mathfrak{T}_5||\mathfrak{T}_6||M)$. Finally, the

signer computes

$$\begin{array}{rclrcl} z_{r_1} & = & \rho_{r_1} - cr_1, & z_{r_2} & = & \rho_{r_2} - cr_2 \\ z_{r_e} & = & \rho_{r_e} - cr_e, & z_{r_s} & = & \rho_{r_s} - cr_s \\ z_e & = & \rho_e - ce, & z_s & = & \rho_s - cs \\ z_{\alpha_1} & = & \rho_{\alpha_1} - c\alpha_1, & z_{\alpha_2} & = & \rho_{\alpha_2} - c\alpha_2 \\ z_{\alpha_3} & = & \rho_{\alpha_3} - c\alpha_3, & z_{\alpha_4} & = & \rho_{\alpha_4} - c\alpha_4 \end{array}$$

Output Π_1 as $(c,\ z_{r_1},\ z_{r_2},\ z_{r_e},\ z_{r_s},\ z_e,\ z_s,\ z_{\alpha_1},\ z_{\alpha_2},\ z_{\alpha_3},\ z_{\alpha_4})$.
To verify Π_1, the verifier computes the following:

$$\begin{aligned} \mathfrak{T}'_1 &= A_1^c g_1^{z_e} g_2^{z_{r_e}} \\ \mathfrak{T}'_2 &= A_1^{-z_{r_1}} g_1^{z_{\alpha_1}} g_2^{z_{\alpha_3}} \\ \mathfrak{T}'_3 &= A_1^{-z_{r_2}} g_1^{z_{\alpha_2}} g_2^{z_{\alpha_4}} \\ \mathfrak{T}'_4 &= S^c u_0^{z_s} \\ \mathfrak{T}'_5 &= \Big(\frac{\hat{e}(A_2, w)}{\hat{e}(g_0, h_0)}\Big)^c \frac{\hat{e}(g_1, h_0)^{z_s} \hat{e}(g_2, h_0)^{z_{r_s}} \hat{e}(g_2, w)^{z_{r_1}} \hat{e}(g_2, h_0)^{z_{\alpha_1}}}{\hat{e}(A_2, h_0)^{z_e}} \\ \mathfrak{T}'_6 &= \Big(\frac{\hat{e}(A_3, q_1)}{\hat{e}(v, h)}\Big)^c \frac{\hat{e}(g_2, q_1)^{z_{r_2}} \hat{e}(g_2, h)^{z_{\alpha_2}}}{\hat{e}(A_3, h)^{z_e}} \end{aligned}$$

The verifier then outputs 1 if $c = H(\mathfrak{T}'_1||\mathfrak{T}'_2||\mathfrak{T}'_3||\mathfrak{T}'_4||\mathfrak{T}'_5||\mathfrak{T}'_6||M)$ and 0 otherwise.

Details of Π_2. The signer randomly generates $\rho_e,\ \rho_{r_e},\ \rho_s \in_R \mathbb{Z}_p$ and computes

$$\begin{aligned} \mathfrak{T}_1 &= g_1^{\rho_e} g_2^{\rho_{r_e}} \\ \mathfrak{T}_2 &= u_0^{\rho_s} \\ \mathfrak{T}_3 &= u_0^{\rho_e} (u_1^R)^{\rho_s} \end{aligned}$$

Next, the signer computes $c = H(\mathfrak{T}_1||\mathfrak{T}_2||\mathfrak{T}_3||R)$. Finally, the signer computes

$$z_e = \rho_e - ce, \quad z_{r_e} = \rho_{r_e} - cr_e, \quad z_s = \rho_s - cs$$

Output Π_2 as $(c,\ z_e,\ z_{r_e},\ z_s)$.
To verify Π_2, the verifier computes the following:

$$\begin{aligned} \mathfrak{T}'_1 &= A_1^c g_1^{z_e} g_2^{z_{r_e}} \\ \mathfrak{T}'_2 &= S^c u_0^{z_s} \\ \mathfrak{T}'_3 &= T^c u_0^{z_e} (u_1^R)^{z_s} \end{aligned}$$

The verifier then outputs 1 if $c = H(\mathfrak{T}'_1||\mathfrak{T}'_2||\mathfrak{T}'_3||R)$ and 0 otherwise.