2013

Secure ID-based linkable and revocable-iff-linked ring signature with constant-size construction

Man Ho Au
University of Wollongong, aau@uow.edu.au

Joseph K. Liu
Institute for Infocomm Research, Singapore

Willy Susilo
University of Wollongong, wsusilo@uow.edu.au

Tsz Hon Yuen
University of Hong Kong, thy738@uow.edu.au

Publication Details
Secure ID-based linkable and revocable-iff-linked ring signature with constant-size construction

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 an 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

Keywords
revocable, linkable, linked, id, i, secure

Disciplines
Engineering | Science and Technology Studies

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

Man Ho Au\textsuperscript{a,}\textsuperscript{*}, Joseph K. Liu\textsuperscript{b}, Willy Susilo\textsuperscript{a,1}, Tsz Hon Yuen\textsuperscript{c}

\textsuperscript{a}Centre for Computer and Information Security Research
School of Computer Science and Software Engineering
University of Wollongong, Australia

\textsuperscript{b}Cryptography and Security Department
Institute for Infocomm Research
Singapore

\textsuperscript{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 \textit{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 \textit{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.

\textit{Keywords:} Digital Signatures, ID-based cryptography, Anonymity, Ring

\textsuperscript{*}Corresponding author.

Email addresses: aaau@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)

\textsuperscript{1}This work is supported by ARC Future Fellowship FT0991397.
1. Introduction

**Ring Signature.** A ring signature scheme (such as \[30, 140, 8, 39, 17, 14\]) 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 revocable 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 signatures 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

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

- each element of \(G_1, G_2\) and \(G_T\) has unique binary representation.
- \(g_0, h_0\) are generators of \(G_1\) and \(G_2\) respectively.
- \(\psi : G_2 \rightarrow G_1\) is a computable isomorphism from \(G_2\) to \(G_1\), with \(\psi(h_0) = g_0\).
- (Bilinear) \(\forall x \in G_1, y \in G_2\) and \(a, b \in \mathbb{Z}_p\), \(\hat{e}(x^a, y^b) = \hat{e}(x, y)^{ab}\).
- (Non-degenerate) \(\hat{e}(g_0, h_0) \neq 1\).

\(G_1\) and \(G_2\) can be the same or different groups. If the group operation in \(G_1\) and \(G_2\), the isomorphism \(\psi\) and the bilinear map \(\hat{e}\) are all efficiently computable, the groups \((G_1, G_2)\) are referred to as a bilinear group pair.

2.2. Mathematical Assumptions

**Definition 1 (Discrete Logarithm).** The Discrete Logarithm (DL) problem in \(G\) is defined as follows: On input a tuple \((Y, g)\) \(\in G^2\) such that \(|G| = p\) 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 \((t, \epsilon)\)-DL assumption holds in \(G\) if no \(t\)-time algorithm has advantage at least \(\epsilon\) in solving the DL problem in \(G\).
Definition 2 (Decisional Diffie-Hellman). The Decisional Diffie-Hellman (DDH) problem in $G$ is defined as follows: On input a quadruple $(g, h, g^a, T) \in G^4$ such that $|G| = p$ 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 $(t, \epsilon)$-DDH assumption holds in $G$ if no $t$-time algorithm has advantage at least $\epsilon$ over random guessing in solving the DDH problem in $G$.

Definition 3 ($q$-Strong Diffie-Hellman). The $q$-Strong Diffie-Hellman ($q$-SDH) problem in $(G_1, G_2)$ is defined as follows: On input a $(q + 2)$-tuple $(g_0, h_0, h_0^2, \ldots, h_0^q) \in G_1 \times G_2^{q+1}$, 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 $(q, t, \epsilon)$-SDH assumption holds in $(G_1, G_2)$ if no $t$-time algorithm has advantage at least $\epsilon$ in solving the $q$-SDH problem in $(G_1, G_2)$.

The $q$-SDH assumption is shown to be true in the generic group model \cite{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 $1^\lambda$ where $\lambda$ is a security parameter, the algorithm outputs a master secret key $s$ and a list of system parameters $\text{param}$ that includes $\lambda$ and the descriptions of a user secret key space $D$, a message space $M$ as well as a signature space $\Psi$.

- **Extract.** The common input of this interactive protocol between a user and the PKG is the system parameters $\text{param}$ 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 $\lambda$, 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 $\lambda$, 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 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 link $\leftarrow \text{Link}(\sigma_1, \sigma_2)$, 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 \(ID_i, d_i \leftarrow \text{Extract}(\text{param}, ID_i, s)\) 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 \(\sigma \leftarrow \text{Sign}(\text{param}, n, \{ID_i | i \in [1, n]\}, m, event, d_j)\). 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 \(\lambda\) and runs **Setup** to generate the public parameters \(\text{param}\) and master secret key \(s\). The adversary \(\mathcal{A}\) is given \(\text{param}\).
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 \(\sigma^*\) for message \(m^*\), event \(event^*\) and a set of identities \(L^*\).

\(\mathcal{A}\) wins the above game if

1. \(\text{Verify}(\text{param}, |L^*|, L^*, m^*, event^*, \sigma^*) = \text{valid}\);
2. \((L^*, m^*, \text{event}^*)\) and \(\sigma^*\) should not be in the set of oracle queries and replies between \(A\) and \(SO\); and
3. \(A\) did not query \(EO\) on any identity \(ID \in L^*\).

The advantage of \(A\) is defined as the probability that \(A\) 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 \(EO\) 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 may be 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 \(S\) takes a sufficiently large security parameter \(\lambda\) and runs 
   **Setup** to generate the public parameters \(\text{param}\) and master secret key \(s\). The adversary \(A\) is given \(\text{param}\) and \(s\).
2. $\mathcal{A}$ can make a polynomial number of oracle queries to $\mathcal{REO}$, $\mathcal{SO}$ 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 $\sigma^* = \text{Sign}(\text{param}, n + 2, L^* \cup \{ID_0^*, ID_1^*, m^*, event^*, d_{ID_b^*}\}$, where $b \in \{0, 1\}$.

4. $\mathcal{A}$ can queries oracles $\mathcal{REO}$, $\mathcal{SO}$ 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 $\mathcal{REO}$). 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 $\lambda$ and runs $\text{Setup}$ to generate the public parameters $\text{param}$ and master secret key $s$. The adversary $\mathcal{A}$ is given $\text{param}$.

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 $\sigma_i^*$ 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:
\(\sigma_i\) for \(i = 0\) to \(k\) are not outputs from \(\mathcal{SO}\).

- \(\text{Verify}(\text{param}, |L^*_i|, L^*_i, m^*_i, \text{event}^*, \sigma^*_i) = \text{valid}\) for \(i = 0\) to \(k\);
- \(\text{Link}(\sigma^*_i, \sigma^*_j) = \text{Unlink}\) 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 \(\mathcal{REO}\) and execute \(\text{Sign}\) algorithm specified in the scheme to generate the signature. \(\mathcal{REO}\) should be queried before if necessary.

Formally it is defined as follows:

1. A simulator \(\mathcal{S}\) takes a sufficiently large security parameter \(\lambda\) and runs \(\text{Setup}\) to generate \(\text{param}\) and master secret key \(s\). \(\mathcal{S}\) sends \(\text{param}\) and \(s\) to the adversary \(\mathcal{A}\).
2. \(\mathcal{A}\) makes a polynomial number of oracle queries to \(\mathcal{REO}\) 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 \(\sigma^*\) for message \(m^*\), event \(\text{event}^*\) and ring \(L^*\).

\(\mathcal{A}\) wins the game if

- \(\text{Verify}(\text{param}, |L^*|, L^*, m^*, \text{event}^*, \sigma^*) = \text{valid}\).
- \(\sigma^*\) is not an output of any \(\mathcal{CSO}\) query.
• \( \text{Link}(\sigma^*, \hat{\sigma}) = \text{Link} \) 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 \( \sigma^* \) for message \( m^* \), event \( \text{event}^* \) 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 \( \sigma_i = \text{Sign}(\text{param}, |L^*|, L^*, m, \text{event}^*, d_i) \), where \( d_i = \text{Extract}(\text{param}, ID_i, s) \). The forger \( \mathcal{A} \) can be classified into two types according to the characteristic of \( \sigma^* \) with respect to these \( \sigma_i \)’s.

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

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 unforgeability, \( \mathcal{A}' \) either win game non-slanderability or game linkability with non-negligible probability.

**Type I Forger.** \( \mathcal{A}' \) receives \( \text{param} \) and \( s \) from \( \mathcal{S} \) in game non-slanderability. \( \mathcal{A}' \) forwards \( \text{param} \) 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 \( \sigma^* \) for message \( m^* \), event \( \text{event}^* \) and a set of identities \( L^* \). If \( \sigma^* \) 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 \( i^* \in \{1, \ldots, \vert L^* \vert \} \) 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 \( \text{event}^* \), ring \( L^* \) and index \( i^* \) to obtain a signature \( \sigma \). Since \( \mathcal{A} \) is a type I forger, with probability \( 1/\vert L^* \vert \), \( \text{link} \leftarrow \text{Link}(\sigma^*, \sigma, \text{event}^*) \). \( \mathcal{A}' \) submits \( \sigma^*, m^*, \text{event}^*, L^* \) and wins game non-slanderability.

**Type II Forger.** \( \mathcal{A}' \) receives \( \text{param} \) from \( \mathcal{S} \) in game linkability. \( \mathcal{A}' \) forwards \( \text{param} \) 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 \( \sigma^* \) for message \( m^* \), event \( \text{event}^* \) and a set of identities \( L^* \). \( \mathcal{A}' \) randomly picks an index \( i^* \in \{1, \ldots, \vert L^* \vert \} \) 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 \( \sigma = \text{Sign}(\text{param}, \vert L^* \vert, L^*, m, \text{event}^*, d_{i^*}) \). Since \( \mathcal{A} \) is a type II forger, \( \text{unlink} \leftarrow \text{Link}(\sigma^*, \sigma, \text{event}^*) \). \( \mathcal{A}' \) then outputs \( k = 1 \), \( \text{event}^*, \sigma_0^* = \sigma^*, \sigma_1^* = \sigma, m_0^* = m^*, m_1^* = m \) and win game linkability. \( \Box \)

### 3.3. Security Requirement of ID-based Revocable-if-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. Revocable-if-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 \( \text{Link} \) protocol or the \( \text{Revoke} \) 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 \( \lambda \) and runs \texttt{Setup} to generate the public parameters \( \text{param} \) and master secret key \( s \). The adversary \( \mathcal{A} \) is given \( \text{param} \).
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 \( \text{event}^* \), a positive integer \( k \), a set of signatures \( \sigma_i^* \) for messages \( m_i^* \) and sets of identities \( L_i^* \) for \( i = 0 \) to \( k \).
Let $C$ be the set of identities queried to $\mathcal{EO}$. $A$ wins the above game if it fulfills either condition:

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

OR

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

The advantage of $A$ is defined as the probability that $A$ 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 \textbf{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 noninteractive 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 \( \text{PK}\{ (x) : y = g^x \} \) 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 \( \Sigma \)-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 \( \text{PK}\{ (x) : y = g^x \} \) will be denoted as \( \text{SPK}\{ (x) : y = g^x \}(M) \). The \( \Sigma \)-protocol is inspired by the protocols described in \[4\].

Accumulators. An accumulator allows the representation of a set of elements, say, \( X = \{ x_1, \ldots, x_n \} \) by a short value \( v \). For each element \( x \) in the set \( X \), 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 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 noninteractive proofs. Nguyen \[28\] 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 \( \lambda \) be the security parameter. Let \( \hat{e} \) be a bilinear pairing as described in Section 2.1 and \((G_1, G_2)\) be a bilinear group pair with computable isomorphism \( \psi \) such that \( |G_1| = |G_2| = p \) for some prime \( p \) of \( \lambda \) bits. Also let \( \mathbb{G}_p \) be a group of order \( p \) where DDH is intractable\(^2\). We define \( H : \{0, 1\}^* \to \mathbb{Z}_p, \quad G_0 : \{0, 1\}^* \to \mathbb{G}_p \) and \( G_1 : \{0, 1\}^* \to \mathbb{G}_p \) be some cryptographic hash functions. Let \( g_0, g_1, g_2 \) be generators of \( G_1 \) 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 \( q \in \mathbb{Z}_p^* \) and compute \( q_i = h^{(q^i)} \) for \( i = 1 \cdots t_{\text{max}} \), where \( t_{\text{max}} \) is the maximum size of the ring.

3. **PKG Setup:** The PKG randomly selects \( \gamma \in \mathbb{Z}_p^* \) and compute \( w = h_0^{\gamma} \). The master secret is \( \gamma \) while the public parameters are \((H, G_0, G_1, \psi, \hat{e}, G_1, G_2, \mathbb{G}_p, p, g_0, g_1, g_2, h_0, h_1, h_2, h, q_1, \ldots, q_{t_{\text{max}}}, w)\).

• 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 \( s', r_s \in \mathbb{Z}_p^* \), 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.

\[
\text{PK}\{ (s', r_s) : C' = g_1^{s'} g_2^{r_s} \}
\]

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

\[
C = C' g_1^{s''}, \quad A = (g_0 C)^{\frac{1}{e+\gamma}}
\]

and sends \((A, s'')\) to the user.

3. User computes \( s = s' + s'' \) and checks if \( \hat{e}(A, w h_0^s) = \hat{e}(g_0 g_1^s g_2^{r_s}, h_0) \).

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 \( n \leq t_{\text{max}} \), using the secret key of a user with identity \( ID_u \) (where \( u \in \{1, n\} \)) in an event \( \text{event} \), first compute \( e = H(ID_u) \), \( u_0 = G_0(\text{event}) \) and

\[
v = \psi(h^k_{k=1}(q+H(ID_k))), \quad v_w = \psi(h^{k=1,k\neq u}(H(ID_k)+q)), \quad S = u_0^s.
\]

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

\[
v = \psi(h^k_{k=1}(q+H(ID_k))) = \psi(h^{(q+H(ID_1)) \ldots (q+H(ID_n))}) = \psi(h^{q+c_n-1q^{-1}+\ldots+c_1q+c_0}) = \psi(h^{q+c_n-1 \ldots (h^q c_1 h^c_0)}) = \psi(q^c_{n-1} \ldots q^c_1 h^c_0)
\]

where \( c_0, \ldots, c_{n-1} \in \mathbb{Z}_p \) are some coefficients. \( v_w \) can be computed in a similar way.

Denote \( M = m||L||\text{event} \). The user further computes the following non-interactive signature-of-knowledge \( \Pi_0 \) which assures the verifier the following relationship are satisfied.

\[
\text{SPK} \left\{ \begin{array}{l}
(A, e, s, r, v_w) : \\
\hat{e}(A, wh_0^e) = \hat{e}(g_0^{g_1^e} g_2^{r_1}, h_0) \\
\hat{e}(v_w, q_1 h^e) = \hat{e}(v, h) \\
S = u_0^s
\end{array} \right\} (M)
\]

The SPK \( \Pi_0 \) 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

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

Denote by \( M' \) the string \( m||L||\text{event}||A_1||A_2||A_3 \) and let \( 1_{G_1} \) be the identity element of \( G_1 \). The signer produces the following non-interactive
zero-knowledge proof-of-knowledge $\Pi_1$:

$$\begin{align*}
&\left\{ \begin{array}{l}
(r_1, r_2, r_e, e, s, r_s, \alpha_1, \alpha_2, \alpha_3, \alpha_4) : \\
A_1 = g_1^e g_2^{r_e} \\
1_{G_1} = A_1^{r_1} \alpha_1^2 g_3^2 \\
1_{G_1} = A_1^{r_2} \alpha_2^2 g_2^2 \\
S = u_0^s \\
T = u_0^e (u_1^R)^s
\end{array} \right. \\
\end{align*}$$

SPK $(M')$

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. $\Pi_1$ consists of 11 elements of $Z_p$. Details of $\Pi_1$ 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

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

and make a proof that $T$ is correctly formed.

This can be done via the following non-interactive zero-knowledge proof-of-knowledge $\Pi_2$:

$$\begin{align*}
&\left\{ \begin{array}{l}
(e, r_e, s) : \\
A_1 = g_1^e g_2^{r_e} \\
S = u_0^s \\
T = u_0^e (u_1^R)^s
\end{array} \right. \\
\end{align*}$$

SPK $(R)$

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 $\Pi_1$ and $\Pi_2$ consists of 11 and 4 elements of $Z_p$ respectively. It is straightforward to combine $\Pi_1$ and $\Pi_2$ into a single proof consisting of 12 elements. We choose to present them separately for charity. Details of $\Pi_2$ 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 $\Pi_1$. 

19
In the revocable-iff-link version, parse the signature as \((S, T, A_1, A_2, A_3, \Pi_1, \Pi_2)\). Compute \(u_1 = G_1(\text{event})\), \(R = H(S||A_1||A_2||A_3||\Pi_1||m||L||\text{event})\) and verify the SPK \(\Pi_1\) and \(\Pi_2\).

- **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 \((S, T', A'_1, A'_2, \Pi'_1, \Pi'_2)\) for an event \(\text{event}\) on message \(m\) from the set of identities \(L\) and another message \(m'\) from another set of identities \(L'\) respectively, compute \(R = H(S||A_1||A_2||A_3||\Pi_1||m||L||\text{event})\) and \(R' = H(S||A'_1||A'_2||A'_3||\Pi'_1||m'||L'||\text{event})\). Compute \(u_0 = G_0(\text{event})\) and \(U = \frac{(T^R)}{(T^{R'})} \left( \frac{1}{r - r'} \right)^{\frac{1}{r - r'}} \). 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 \((G_1, G_2)\), the DDH and DL assumptions hold 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 \(G_p\) holds in the random oracle model.

**Proof.** The proof is by reduction. Suppose there exists an adversary \(A\) that has non-negligible advantage in winning game anonymity, we construct an algorithm \(A'\) that solves the DDH problem. Since it is widely believed that
the DDH problem is hard, it implies no algorithm $A$ exists and this complete the proof of the lemma.

Below we show how to construct algorithm $A'$. $A'$ is given an instance of the DDH problem, $(u_0, u_1, u_0^?, Z)$ and its task is to determine if $Z = u_1^?$. $A'$ creates the public parameters $\text{param}$ and the master secret key $s$ according to the setup algorithm. $\text{param}$ and $s$ are given to the adversary $A$.

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

For queries not related to $ID_{i^*}$, $A'$ answers the query following the specification. For $\text{REO}$ query related to $ID_{i^*}$, $A'$ picks a random value $C' \in_R G_1$ and employs the zero-knowledge simulator to simulate proof-of-knowledge. For signature query related to $ID_{i^*}$, $A'$ simulates the reply as if the secret key obtained in the $\text{REO}$ query for $ID_{i^*}$ is $(A, e, s, r_s)$. $A'$ locates the hash query for $\text{event}$ for the signature such that $\text{event} = \text{event}_j$ and $\text{event} = \text{event}_k$ for some indexes $j$ and $k$. $A'$ proceeds only if $j \neq j^* \land k \neq k^*$. $A'$ sets the value of $S$ and $T$ to be $(u_0^j)^{\mu_j}$ and $u_0^j(u_0^j)^{R_{j-k}}$ respectively. Simulation is possible because of the zero-knowledgeness of $\Pi_1$ and $\Pi_2$. 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 $C = C'g_1^{\nu_j} = g_1^{\nu_j}g_2^{r_s}$.

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

In case $A'$ does not abort. It creates a challenge signature as if the secret key it obtains for $ID_{i^*}$ is $(A, e, s, r_s)$. $A'$ sets the value of $S$ and $T$ to be $(Z)^{\mu_{j^*}}$ and $u_0^j(Z)^{R_{j-k^*}}$ 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 $C = C'g_1^{\nu_j} = g_1^{\nu_j}g_2^{r_s}$. $A'$ then sets the value $A_1, A_2, A_3$ as three random values in $G_1$. Note that for any value of $e$, there exists a value $r_e$ such that
\[ A_1 = g_1^r g_2^{r_2}. \] Likewise, for any values \( A \) and \( v_w \), there exists \( r_1, r_2 \) such that
\[ A_2 = Ag_2^{r_1} \quad \text{and} \quad A_3 = v_w g_2^{r_2}. \] Thus, \((A_1, A_2, A_3)\) is correctly distributed in the view of the adversary. \( A' \) then simulates the zero-knowledge proofs \( \Pi_1 \) and \( \Pi_2 \) using the values \( S, T, A_1, A_2, A_3 \).

If \( A \) finally outputs \( b' = b \), then \( A' \) outputs 1 for the DDH problem. Otherwise, \( A' \) outputs 0.

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

**Lemma 2.** Our scheme satisfies definition (non-slanderable) if the DL assumption holds in \( G_p \) in the random oracle model.

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

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

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

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

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

For queries not related to \( ID_i \), \( A' \) answers the query following the specification. For \( \text{REO} \) query related to \( ID_i \), \( A' \) picks a random value \( C'' \in_R G_1 \) and employs the zero-knowledge simulator to simulate proof-of-knowledge.

For \( \text{CSO} \) queries related to \( ID_i \), \( A' \) simulates the reply as if the secret key obtained in the \( \text{REO} \) query for \( ID_i \) is \((A, e, s, r_s)\). \( A' \) locates the hash query for \text{event} for the signature such that \( G_0(\text{event}) = (u^\ast)^\mu \) and \( G_1(\text{event}) = (u^\ast)^{\mu'} \). \( A' \) sets the value of \( S \) and \( T \) to be \((S^\ast)^\mu \) and \( u_0^\ast(S^\ast)^{\mu'} \) respectively. Simulation is possible because of the zero-knowledgeness of \( \Pi_1 \) and \( \Pi_2 \). 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, \( A \) returns a valid signature \( \sigma^* \), which is not the output from \( CSO \), but is linked to one of them. If it is linked to the signature created from \( ID_{i^*} \), \( A' \) rewinds and extracts the SPK \( \Pi_1 \) to obtain \( s^* \). Then \( A' \) returns \( s^* \) as the solution of the DL problem.

**Lemma 3.** Our scheme satisfies definition (revocable-iff-linked) if the \( q_{H} \)-SDH assumption holds in \((G_1, G_2)\) in the random oracle model, where \( q_{H} + 1 \) is the number of hash queries made by \( A \).

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

Below we show how to construct algorithm \( A' \). \( A' \) is given an instance of the \( q_{H} \)-SDH problem \((g'_1, g'_2, g'_2^x, \ldots, g'_2^{x^{q_{H}}})\) and its task is to output a tuple \((e, g'_1^x)^{e^*}\).

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

\[
\begin{align*}
    h_0 &= g'_2^{f(x)}, \quad w = g'_2^{x f(x)}, \quad g_0 = \psi(h_0).
\end{align*}
\]

While \( x \) is unknown to \( A' \), it possesses its powers in the form of \( g'_2^{x^i} \) and thus it is possible for \( A' \) to compute the values \( h_0 \) and \( w \). Then, \( A' \) picks \( e^*, a^*, k^* \in \mathbb{Z}_p^* \) and computes:

\[
\begin{align*}
    h_1 &= [(w h_0^{e^*})^{k^*} h_0^{-1}]^{1/a^*} = h_0^{(e^* + x)k^* - 1}, \quad g_1 = \psi(h_1).
\end{align*}
\]

\( A' \) randomly picks \( \mu \in \mathbb{Z}_p^* \), \( h \in G_2 \), sets \( g_2 = g_0^\mu \). Other public parameters are created following the specification. The public parameters \( \text{param} \) is given to the adversary \( A \).

For \( i = 1 \) to \( q_{H} - 1 \), \( A' \) computes

\[
A_i = g_0^{1/x + e_i} = \psi(g_2^*(f(x)/x + e_i)).
\]

WLOG, we denote \( e^* = e_{q_{H}}, A_{q_{H}} = g_0^{e^*} \) and \( e_0 \in_R \mathbb{Z}_p^* \). The set \( \{(A_i, e_i)_{i=1}^{q_{H}}\} \) and \( e_0 \) are keep private to \( A' \) and are used to answer the oracle queries.
Recall that we assume $\mathcal{A}$ makes $q_H + 1$ queries to the hash oracle $H$. For each hash query to oracle $H$, $\mathcal{A}'$ chooses an index $i \in \{0, 1, \ldots, q_H\}$ and returns $e_i$.

For each $EO$ query on identity ID, we assume there exists an index $i \in \{0, \ldots, q_H\}$ 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 $i \neq q_H$, $\mathcal{A}'$ randomly picks $s'' \in \mathbb{Z}_p^*$ and computes:

$A = (g_0Cg_0^s)^{1/x+e_i}$

$= (g_0^{1+r_{s}\mu+(s'+s'')(e^*+e)x_{i}})^{1/x+e_i}$

$= A_i^{1+r_{s}\mu-(s'+s'')(e^*+e)x_{i}}g_0^{(s'+s'')(e^*+e)x_{i}}$

$= A_i^{(1+r_{s}\mu-(s'+s'')(e^*+e)x_{i})}(g_0^{(s'+s'')(e^*)/(1-e_{i+e})})$

$= A_i^{(1+r_{s}\mu-(s'+s'')(e^*+e)x_{i})}(g_0^{(s'+s'')(e^*+e)x_{i}})$

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

For $i = q_H$, $\mathcal{A}'$ returns $(A_{q_H} = g_0^{k^*}, e_{q_H}, s'' = a^* - s')$ 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 $SO$ query, $\mathcal{A}'$ answers these queries using the zero-knowledge simulator of $\Pi_1$ and $\Pi_2$.

Finally, $\mathcal{A}$ returns $k + 1$ valid signatures $\sigma_j^*$ 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 $\sigma_j^*$ corresponds to a set of witnesses satisfying the following relationships.

1. $A_{i}^{e+x} = g_0 g_{i}^{s} g_{2}^{r_{s}}$
2. $v_{w}^{e+q} = v$
3. $S = u_{0}^{s}$
4. $T = u_{0}^{e} u_{1}^{r_{s}}$

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 \( g_0^s g_1^{s'} \neq g_0^s g_2^{s'} \) if \( s \neq s' \). This implies there exists \( k+1 \) distinct pairs of \((A,e)\) in these \( k+1 \) valid signatures. \( A'\) randomly picks an index \( j \) and extracts the witnesses \((A,e,s,r,v,w)\) from the SPK \( \Pi_1, \Pi_2 \) in \( \sigma^* \). 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 \( 1/(q_H+1) \), \( e = e_0 \). If not, \( A' \) aborts. Otherwise, \( A' \) computes:

\[
B = \left( Ag_0^{\frac{sk^s}{a^s}} \right)^{\frac{1}{a^s + r_s \mu + \frac{2[(e^s + x)h^s - 1]}{a}}}
\]

Note that since \( \sigma^*_j \) is a valid signature,

\[
A^{e_0 + x} = g_0^s g_1^{r_s} g_2^{s'} = g_0^1 + r_s \mu + \frac{2[(e^s + x)h^s - 1]}{a}
\]

\[
A = \left( g_0^{\frac{a^s + r_s \mu a^s}{a}} \right)^{(e_0^s + x)(1-e_0^s - e^s)}
\]

\[
\therefore B = g_0^1
\]

- **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 \( 1/(q_H+1) \), \( e = e^* \). If not, \( A' \) aborts. Otherwise, \( A' \) computes:

\[
B = \left( Ag_0^{\frac{sk^s}{a^s}} \right)^{\frac{1}{a^s + r_s \mu + e^s}}
\]

Note that since \( \sigma^*_j \) is a valid signature,

\[
A^{e^* + x} = g_0^s g_1^{r_s} g_2^{s'} = g_0^1 + r_s \mu + \frac{2[(e^s + x)h^s - 1]}{a}
\]

\[
A = \left( g_0^{\frac{a^s + r_s \mu a^s}{a}} \right)^{(e_0^s + x)}
\]

\[
\therefore B = g_0^1
\]

In both cases, \( A' \) obtains a tuple \((B,e)\) such that \( B = \frac{1}{g_0^1} \). \( A' \) computes the polynomial \( a(x) \) of degree \( q_H - 2 \) such that \( f(x) = (x + e)a(x) + b \) for some \( b \neq 0 \in \mathbb{Z}_p \) (since \( e \notin \{e_1, \ldots, e_{q_H-1}\} \)). Recall that \( g_0 = \psi(g_2^{f(x)}) = g_1^{f(x)} \). Thus, \( B = \frac{1}{g_0^1} = g_1^{a(x)} g_1^{\frac{1}{x+e}} \). \( A' \) outputs \((e, (Bg_1 - a(x))^\frac{1}{b})\) as the solution to the \( q_H \)-SDH problem.
Assume $A$ wins by condition 2 of definition 8. WLOG, we let the two signatures outputted by $A$ and linked together are $\dot{\sigma}^*$ and $\ddot{\sigma}^*$ on event $\mathit{event}^*$ and that $ID' = \mathsf{Revoke}(\dot{\sigma}^*, \ddot{\sigma}^*, \mathit{event}^*)$. Due to the soundness of the SPK, the two signatures are the proof-of-knowledge of two sets of witnesses $(\dot{A}, \dot{\mathit{e}}, \dot{s}, \dot{r}, \dot{v})$ and $(\ddot{A}, \ddot{\mathit{e}}, \ddot{s}, \ddot{r}, \ddot{v})$ respectively. If $\dot{\mathit{e}} = \ddot{\mathit{e}}$, there exists $i$ such that $\dot{\mathit{e}} = \ddot{\mathit{e}} = e_i$ (since $\dot{\mathit{e}}, \ddot{\mathit{e}}$ must be the output of the random oracle $H$). Due to the soundness of the SPK and the correctness of the $\mathsf{Revoke}$ algorithm, $e_i = H(ID')$. Under the assumption that the accumulator is secure \cite{29}, $ID' \in L^*$. In that case, $A$ wins only if $ID'$ has not been submitted to $EO$. With probability $1/(q_H + 1)$, $e_i = e_0$. $A'$ solves the $q_H$-SDH problem as in case 1 in the previous condition.

Next, consider the case when $\dot{\mathit{e}} \neq \ddot{\mathit{e}}$. Let $\dot{\mathit{e}} = H(\dot{ID})$ and $\ddot{\mathit{e}} = H(\ddot{ID})$ respectively. If $\dot{ID}$ (or $\ddot{ID}$) has not been submitted to $EO$, with probability $1/(q_H + 1)$, $\dot{e} = e_0$ (or $\ddot{e} = e_0$). In that case, $A'$ solves the $q_H$-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 $EO$ oracle. In that case, either Let $i$ and $\bar{i}$ be the indexes such that $e_i = H(\dot{ID})$ and $e_{\bar{i}} = H(\ddot{ID})$. Since $\dot{\sigma}^*$ and $\ddot{\sigma}^*$ are linked, $\dot{s} = \ddot{s}$. On the other hand, the values $s_i$ and $s_{\bar{i}}$ in the corresponding $EO$ queries are different with overwhelming probability. The reason is that the value $s_i$ in the an $EO$ query depends on the random number chosen by $A'$ and that in the query with index $q_H$, the value $s^*$ is hidden in and information-theoretic manner before the query. WLOG, assume $s_i \neq \dot{s}$. Under the discrete logarithm assumption, if $s_i \neq \dot{s}$, the value of $g_0 g_1^{s_i} g_2^{e_i}$, $g_0 g_1^{s_{\bar{i}}} g_2^{e_{\bar{i}}}$. Thus, $A \neq A'$. With probability $1/(q_H + 1)$, $\dot{e} = e_{q_H}$. In that case, $A'$ solves the $q_H$-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


Appendix A. Details of SPK $\Pi_1$ and $\Pi_2$

Details of $\Pi_1$. The signer randomly generates $\rho_1, \rho_2, \rho_r, \rho_s, \rho_e, \rho_s, \rho_{\alpha_1}, \rho_{\alpha_2}, \rho_{\alpha_3}, \rho_{\alpha_4} \in \mathbb{Z}_p$ and computes

$$\begin{align*}
\bar{\xi}_1 &= g_1^{\rho_e} g_2^{\rho_r} \\
\bar{\xi}_2 &= A_1^{-\rho_1} g_1^{\rho_{\alpha_1}} g_2^{\rho_{\alpha_3}} \\
\bar{\xi}_3 &= A_1^{-\rho_2} g_1^{\rho_{\alpha_2}} g_2^{\rho_{\alpha_4}} \\
\bar{\xi}_4 &= u_0^{\rho_e} \\
\bar{\xi}_5 &= \hat{e}(g_1, h_0)^{\rho_s} \hat{e}(g_2, h_0)^{\rho_{\alpha_1}} \hat{e}(g_2, w)^{\rho_r} \hat{e}(g_2, h_0)^{\rho_{\alpha_2}} \\
\bar{\xi}_6 &= \hat{e}(g_2, q_1)^{\rho_{\alpha_3}} \hat{e}(g_2, h)^{\rho_{\alpha_4}} \\
\end{align*}$$

Next, the signer computes $c = H(\bar{\xi}_1||\bar{\xi}_2||\bar{\xi}_3||\bar{\xi}_4||\bar{\xi}_5||\bar{\xi}_6||M)$. Finally, the
The signer computes
\[
\begin{align*}
z_{r_1} &= \rho_{r_1} - cr_1, \\
z_{r_e} &= \rho_{r_e} - cr_e, \\
z_e &= \rho_e - ce, \\
z_{\alpha_1} &= \rho_{\alpha_1} - c\alpha_1, \\
z_{\alpha_3} &= \rho_{\alpha_3} - c\alpha_3.
\end{align*}
\]

Output \(\Pi\) as \((c, z_{r_1}, z_{r_e}, z_e, z_{\alpha_1}, z_{\alpha_3}, z_s, z_{\alpha_4})\).

To verify \(\Pi\), the verifier computes the following:
\[
\begin{align*}
\mathcal{X}_1 &= A_1^c g_1^{z_e} g_2^{z_{r_e}} \\
\mathcal{X}_2 &= A_1^{-z_{r_1}} g_1^{z_{\alpha_1}} g_2^{z_{\alpha_3}} \\
\mathcal{X}_3 &= A_1^{-z_{r_e}} g_1^{z_{\alpha_2}} g_2^{z_{\alpha_4}} \\
\mathcal{X}_4 &= S^c u_0^{z_e} \\
\mathcal{X}_5 &= (\frac{\hat{e}(A_2, w)}{\hat{e}(g_1, h_0)}) c \hat{e}(g_2, h_0)^{z_e} \hat{e}(g_2, w)^{z_{r_e}} \hat{e}(g_2, h_0)^{z_{\alpha_1}} \\
\mathcal{X}_6 &= (\frac{\hat{e}(A_3, q_1)}{\hat{e}(v, h)}) c \hat{e}(g_2, q_1)^{z_{r_e}} \hat{e}(g_2, h)^{z_{\alpha_2}} \\
\end{align*}
\]

The verifier then outputs 1 if \(c = H(\mathcal{X}_1 || \mathcal{X}_2 || \mathcal{X}_3 || \mathcal{X}_4 || \mathcal{X}_5 || \mathcal{X}_6 || M)\) and 0 otherwise.

Details of \(\Pi_2\). The signer randomly generates \(\rho_e, \rho_{r_e}, \rho_s \in \mathbb{Z}_p\) and computes
\[
\begin{align*}
\mathcal{X}_1 &= g_1^{\rho_e} g_2^{\rho_{r_e}} \\
\mathcal{X}_2 &= u_0^{\rho_s} \\
\mathcal{X}_3 &= u_0^{\rho_e} (u_1^R)^{\rho_s}
\end{align*}
\]

Next, the signer computes \(z_e = \rho_e - ce, \quad z_{r_e} = \rho_{r_e} - cr_e, \quad z_s = \rho_s - cs\)

Output \(\Pi_2\) as \((c, z_e, z_{r_e}, z_s)\).

To verify \(\Pi_2\), the verifier computes the following:
\[
\begin{align*}
\mathcal{X}_1' &= A_1^c g_1^{z_e} g_2^{z_{r_e}} \\
\mathcal{X}_2' &= S^c u_0^{z_s} \\
\mathcal{X}_3' &= T^c u_0^{z_e} (u_1^R)^{z_s}
\end{align*}
\]

The verifier then outputs 1 if \(c = H(\mathcal{X}_1' || \mathcal{X}_2' || \mathcal{X}_3' || M)\) and 0 otherwise.