1-1-2013

Efficient linkable and/or threshold ring signature without random oracles

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

Joseph K. Liu
Institute for Infocomm Research

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

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

Jianying Zhou
Institute for Infocomm Research Singapore, jyzhou@i2r.a-star.edu.sg

Recommended Citation

Yuen, Tsz Hon; Liu, Joseph K.; Au, Man Ho; Susilo, Willy; and Zhou, Jianying, "Efficient linkable and/or threshold ring signature without random oracles" (2013). *Faculty of Engineering and Information Sciences - Papers: Part A*. 696.
https://ro.uow.edu.au/eispapers/696

Efficient linkable and/or threshold ring signature without random oracles

Abstract

Linkable ring signatures have found many attractive applications. One of the recent important extensions is a linkable threshold ring signature (LTRS) scheme. Unfortunately, the existing LTRS schemes are only secure in the random oracle model (ROM). In this paper, we make the following contributions. First, we construct the first LTRS scheme that is secure without requiring the ROM. Further, we enhance the security of a threshold ring signature (for both linkable or non-linkable) by providing a stronger definition of anonymity. This strengthened notion makes threshold ring signature schemes more suitable in real life. Finally, we provide efficient schemes that outperform the existing schemes in the literature. Our scheme is particularly suitable for electronic commerce or electronic government where anonymity and accountability are the most concerned factors. © 2012 The Author.

Keywords

era2015

Disciplines

Engineering | Science and Technology Studies

Publication Details

Yuen, T. Hon., Liu, J. K., Au, M. Ho., Susilo, W. & Zhou, J. (2013). Efficient linkable and/or threshold ring signature without random oracles. The Computer Journal, 56 (4), 407-421.

Efficient Linkable and/or Threshold Ring Signature without Random Oracles

Tsz Hon Yuen[1], Joseph K. Liu[2], Man Ho Au[3], Willy Susilo[3] and Jianying Zhou[2]

[1] University of Hong Kong, Hong Kong
[2] Institute for Infocomm Research, Singapore
[3] University of Wollongong, Australia
Email: thyuen@cs.hku.hk, ksliu@i2r.a-star.edu.sg, {aau,wsusilo}@uow.edu.au, jyzhou@i2r.a-star.edu.sg

Linkable ring signatures have found many attractive applications. One of the recent important extensions is a linkable threshold ring signature scheme (LTRS). Unfortunately, the existing LTRS schemes are only secure in the random oracle model. In this paper, we make the following contributions. First, we construct *the first* LTRS scheme that is secure without requiring the random oracle model. Further, we enhance the security of a threshold ring signature (for both linkable or non-linkable) by providing a stronger definition of anonymity. This strengthened notion makes threshold ring signature schemes more suitable in the real life. Finally, we provide efficient schemes that outperform the existing schemes in the literature. Our scheme is particularly suitable for electronic commerce or electronic government where anonymity and accountability are the most concerned factors.

Keywords: Threshold; Linkable; Ring Signature; Random Oracles

1. INTRODUCTION

Ring Signature. A ring signature scheme (such as [1, 2, 3, 4, 5, 6, 7]) 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 [8, 9, 10]), 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 [1], anonymous membership authentication for ad hoc groups [11], non-interactive deniable ring authentication [12], smart grid systems [13], perfect concurrent signature [14] and multi-designated verifiers signature [15].

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. [16] 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 [16], e-voting [17] and e-cash [18]. 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 [16].

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 [16, 19, 20] do not mention about this property, they can be modified trivially to achieve this property.

All previous linkable ring signature schemes (e.g., [21, 18, 19, 20, 22, 23, 24, 25, 26]) except the recent work by Fujisaki [27] are only proven secure in the random oracle model.

We also remark that Wang and Zhao [28] made some cryptanalysis to a number of previous linkable ring signature schemes. They also claimed that *"To design secure linkable ring signature scheme is still an open problem."*. In Appendix Appendix A, we demonstrate that their cryptanalysis is indeed invalid and hence, their claim is also incorrect.

Linkable Threshold Ring Signature. A (d, n)-linkable threshold ring signature (LTRS) has the similar notion to the (1-out-of-n) linkable ring signature. A (d, n)-LTRS scheme requires at least d signers to work collaboratively to generate a signature. Those d participating signers can select any set of n entities including themselves without getting any consent from those diversion group members. Linkability for threshold ring signatures is diversified into *individual-linkability* and *coalition-linkability*. For individual-linkability, two signatures are linked if they share at least one common signer even though the two identity sets are different. On the other hand, two signatures are coalition-linked if the signer sets are exactly the same.

There are only two LTRS schemes in the literature. The first one was given by Tsang *et al.* in [21] which allows a separate type of public key. Another LTRS scheme was presented in [26] in ID-based setting. All of them rely on random oracles for proving their security.

Applications. In addition to the applications described above for the non-threshold linkable ring signature, the threshold variant can be useful in the following situation. Assume there is an election for a company chairman. All management committee members are eligible to vote. Before the voting, each candidate should have at least d nominations within the management committee members. The nomination process should be anonymous. One committee member can only nominate one candidate. Otherwise, the nomination becomes invalid. In this case, threshold linkable ring signature can be deployed as it fulfills all the requirement:

- It provides anonymous to each nominator.
- It can make sure each nominatee to have at least d nominators.
- If a committee member nominates more than one candidate, those two nominations can be linked and will become invalid.

In the reality, the Hong Kong Chief Executive election uses the same mode. There are a small group of 1200 election committee members that can vote for the Chief Executive. Each candidate should get at least 150 committee members for the nomination before becoming a candidate. The process should be anonymous. Threshold linkable ring signature can be fully suitable in this kind of nomination.

Our Contribution. The contribution of our paper can be classified into the following area:

1. We propose the first d-out-of-n Linkable Threshold Ring Signature (LTRS) scheme provable secure without random oracles. All previous LTRS schemes can be only proven secure in the random oracle model.
2. We enhance the security of threshold ring signature (linkable or non-linkable) by giving a stronger definition of anonymity, called *Anonymity under Full Key Exposure and Insider Attack*. Under this stronger notion, the adversary is not only given secret keys of the target users, it is also allowed to *interact* with some honest users. All previous threshold ring signature definition only allows the adversary to have user secret keys.
3. We achieve better efficiency when compared to other schemes (even with those rely on random oracles):
   (a) When compared with other LTRS schemes, they require $O(n^2)$ for the linking complexity while we can achieve $O(d \log d)$.
   (b) Our scheme can be seen as a normal linkable ring signature when we set the threshold value d to 1. When compared to the Fujisaki scheme [27], which is the only linkable ring signature scheme that can be proven secure without random oracles, the linking complexity of our scheme is $O(1)$ while the Fujisaki scheme requires $O(n \log n)$.
   (c) Our scheme can be easily modified to achieve a regular threshold ring signature (*i.e.,* without linkability), by using a different event tag every time. When compared to other threshold ring signature schemes, our signature size is $O(d\sqrt{n})$ while all other schemes require at least $O(n)$. When $d < \sqrt{n}$, our signature size is smaller than all previous schemes. If the threshold value d is a small integer such as 2 or 3 while the value n is very large, our signature size is very short when compared to others.

2. PRELIMINARIES

Pairings. We make use of bilinear groups of composite order. Let N be a composite number with factorization $N = pq$. $\mathbb{G}$ is a multiplicative cyclic group of order N. $\mathbb{G}_p$ is its cyclic order-p subgroup, and $\mathbb{G}_q$ is its cyclic order-q subgroup. $\mathbb{G}_T$ is a multiplicative group of order N. g is a generator of $\mathbb{G}$. Then $\hat{e}$ is a bilinear map such that $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ with the following properties:

- *Bilinearity*: For all $u, v \in \mathbb{G}$, and $a, b \in \mathbb{Z}$, $\hat{e}(u^a, v^b) = \hat{e}(u, v)^{ab}$.
- *Non-degeneracy*: $\langle \hat{e}(g, g) \rangle = \mathbb{G}_T$ whenever $\langle g \rangle = \mathbb{G}$.
- *Computability*: It is efficient to compute $\hat{e}(u, v)$ for all $u, v \in \mathbb{G}$.

The group operations on $\mathbb{G}$ and $\mathbb{G}_T$ can be performed efficiently. Bit strings corresponding to elements of $\mathbb{G}$ and of $\mathbb{G}_T$ can be recognized efficiently.

Mathematical Assumptions. For our scheme, we assume three problems are difficult to solve in the setting described above.

Definition 2.1 (Q-Strong Diffie-Hellman (SDH) in $\mathbb{G}_p$[29]). *Given the tuple* $(g_p, g_p^{\alpha}, g_p^{\alpha^2}, \ldots, g_p^{\alpha^Q})$, *where* $g_p \in_R \mathbb{G}_p$, *and* $\alpha \in_R \mathbb{Z}_p$, *compute and output* $g_p^{\frac{1}{\alpha+r}} \in \mathbb{G}_p$ *and* $r \in \mathbb{Z}_p$.

Definition 2.2 (Subgroup Decision in $\mathbb{G}_q$ [30]). *Given w selected at random either from $\mathbb{G}$ (with probability $1/2$) or from $\mathbb{G}_q$ (with probability $1/2$), decide whether w is in $\mathbb{G}_q$. For this problem one is given the description of $\mathbb{G}$, but not given the factorization of N.*

Definition 2.3 (Q'-Decisional Diffie-Hellman Inversion (DDHI) [27]). *Given the tuple* $(g, A_1, A_2, \ldots, A_{Q'})$, *where*

$$A_i = \begin{cases} g^{\frac{1}{\alpha+i}}, & \text{if } i \neq \tau, \\ R & \text{if } i = \tau. \end{cases}$$

$\alpha \in_R \mathbb{Z}_N$, $R \in \mathbb{G}$, $\tau \in \{1, \ldots, Q'\}$, *decide whether* $R = g^{\frac{1}{\alpha+\tau}}$.

These assumptions are formalized by measuring an adversary's success probability for SDH problem and an adversary's guessing advantage for the subgroup decision problem and the DDHI problem.

Boneh-Boyen Signature. Boneh and Boyen [29] proposed a short signature without random oracles. This BB-signature will be used in our construction. We briefly review the BB-signature here.

- Setup: On input the security parameter λ, it generates a pairing $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$, where $\mathbb{G}, \mathbb{G}_T$ are cyclic groups of order p. Let g is a generator of $\mathbb{G}$. It randomly picks $\alpha \in_R \mathbb{Z}_p$. The public key is $(p, \mathbb{G}, \mathbb{G}_T, \hat{e}, g, g^{\alpha})$ and the secret key is α.
- Sign: On input the message m and the secret key α, the signer computes the signature $\sigma = g^{\frac{1}{\alpha+m}}$.
- Verify: On input the message m, the signature σ and the public key, output accept if $\hat{e}(g^{\alpha} g^m, \sigma) = \hat{e}(g, g)$.

The BB signature is existentially unforgeable under the *weak chosen message attack* if the SDH assumption holds in $\mathbb{G}$.

3. SECURITY MODEL

We give our security model and define relevant security notions.

3.1. Syntax of linkable threshold ring signature

A *linkable threshold ring signature*, (LTRS) scheme, is a tuple of five algorithms (Setup, KeyGen, Sign, Verify and Link).

- param ← Setup(λ) is a PPT algorithm which, on input a security parameter λ, outputs the set of security parameters param which includes λ. We denote by $\mathcal{EID}$, $\mathcal{M}$ and Σ the domains of event-id, messages and signatures, resp.
- (sk_i, pk_i) ← KeyGen(param) is a PPT algorithm which, on input a security parameter $\lambda \in \mathbb{N}$, outputs a private/public key pair (sk_i, pk_i). We denote by $\mathcal{SK}$ and $\mathcal{PK}$ the domains of possible secret keys and public keys, resp. When we say that a public key corresponds to a secret key or vice versa, we mean that the secret/public key pair is an output of KeyGen.
- σ ← Sign$(e, n, d, \mathcal{Y}, \mathcal{X}, M)$ which, on input event-id e, group size n, threshold $d \in \{1, \ldots, n\}$, a set $\mathcal{Y}$ of n public keys in $\mathcal{PK}$, a set $\mathcal{X}$ of d private keys whose corresponding public keys are all contained in $\mathcal{Y}$, and a message M, produces a signature σ.
- accept/reject ← Verify$(e, n, d, \mathcal{Y}, M, \sigma)$ which, on input event-id e, group size n, threshold $d \in \{1, \ldots, n\}$, a set $\mathcal{Y}$ of n public keys in $\mathcal{PK}$, a message-signature pair (M,σ) returns accept or reject. If accept, the message-signature pair is *valid*.
- linked/unlinked ← Link $(e, d, n_1, n_2, \mathcal{Y}_1, \mathcal{Y}_2, M_1, M_2,, \sigma_1, \sigma_2)$ which, on input event-id e, group size n_1, n_2 (assume $n_1 \leq n_2$), threshold $d \in \{1, \ldots, n_1\}$, two sets $\mathcal{Y}_1, \mathcal{Y}_2$ of n_1, n_2 public keys respectively, two valid signature and message pairs $(M_1, \sigma_1, M_2, \sigma_2)$, outputs linked or unlinked.

Remark: According to [18, 31], linkability for threshold ring signatures is diversified into *individual-linkability* and *coalition-linkability*, our definition belongs to the former type. That is, in our definition **two signatures are linked if there exists at least one common signer**, while the later linkability only represents whether two signatures share the exact same set of common signers even though the two public key sets are different.

Correctness. LTRS schemes must satisfy:

- (Verification Correctness.) Signatures signed according to specification are accepted during verification.
- (Linking Correctness.) If two signatures are signed for the same event according to specification, then they are linked if and only if the two signatures share at least one common signer.

3.2. Notions of Security of Linkable Threshold Ring Signature

Security of LTRS schemes has four aspects: unforgeability, anonymity, linkability and non-slanderability. Before giving their definition, we consider the following oracles which together model the ability of the adversaries in breaking the security of the schemes.

- $pk_i \leftarrow \mathcal{JO}(\perp)$. The *Joining Oracle*, on request, adds a new user to the system. It returns the public key $pk \in \mathcal{PK}$ of the new user.
- $sk_i \leftarrow \mathcal{CO}(pk_i)$. The *Corruption Oracle*, on input a public key $pk_i \in \mathcal{PK}$ that is a query output of $\mathcal{JO}$, returns the corresponding secret key $sk_i \in \mathcal{SK}$.
- $\sigma' \leftarrow \mathcal{SO}(e, n, d, \mathcal{Y}, \mathcal{V}, M)$. The *Signing Oracle*, on input an event-id e, a group size n, a threshold $d \in \{1, \ldots, n\}$, a set $\mathcal{Y}$ of n public keys, a signer subset $\mathcal{V}$ of $\mathcal{Y}$ with $|\mathcal{V}| = d$, and a message M, returns a valid signature σ'.

1. Unforgeability. Unforgeability for LTRS schemes is defined in the following game between the Simulator $\mathcal{S}$ and the Adversary $\mathcal{A}$ in which $\mathcal{A}$ is given access to oracles $\mathcal{JO}$, $\mathcal{CO}$ and $\mathcal{SO}$:

   (a) $\mathcal{S}$ generates and gives $\mathcal{A}$ the system parameters param.
   (b) $\mathcal{A}$ may query the oracles according to any adaptive strategy.
   (c) $\mathcal{A}$ gives $\mathcal{S}$ an event-id $e \in \mathcal{EID}$, a group size $n \in \mathbb{N}$, a threshold $d \in \{1, \ldots, n\}$, a set $\mathcal{Y}$ of n public keys in $\mathcal{PK}$, a message $M \in \mathcal{M}$ and a signature $\sigma \in \Sigma$.

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

   (1) Verify$(e, n, d, \mathcal{Y}, M, \sigma)$=accept
   (2) All of the public keys in $\mathcal{Y}$ are query outputs of $\mathcal{JO}$
   (3) At most $(d-1)$ of the public keys in $\mathcal{Y}$ have been input to $\mathcal{CO}$
   (4) σ is not a query output of $\mathcal{SO}$.

   We denote by

$$\mathbf{Adv}_{\mathcal{A}}^{unf}(\lambda) = \Pr[\mathcal{A} \text{ wins the game }]$$

   Definition 3.1 (unforgeability). *A LTRS scheme is unforgeable if for all PPT adversary $\mathcal{A}$, $\mathbf{Adv}_{\mathcal{A}}^{unf}(\lambda)$ is negligible.*

2. Linkable-Anonymity.
   Anonymity for LTRS schemes is defined in the following game between the Simulator $\mathcal{S}$ and the Adversary $\mathcal{A}$ in which $\mathcal{A}$ is given access to oracles $\mathcal{JO}$, $\mathcal{CO}$ and $\mathcal{SO}$:

   (a) $\mathcal{S}$ generates and gives $\mathcal{A}$ the system parameters param.
   (b) $\mathcal{A}$ may query the oracles according to any adaptive strategy. Suppose $\mathcal{A}$ makes a total number of v queries to $\mathcal{CO}$. The restriction is that: $v < n - d$.
   (c) $\mathcal{A}$ gives $\mathcal{S}$ event-id e, group size n, threshold $d \in \{1, \ldots, n\}$, message M, and a set $\mathcal{Y}$ of n public keys all of which are query outputs of $\mathcal{JO}$. $\mathcal{S}$ picks randomly a subset $\mathcal{V}$ of $\mathcal{Y}$ with $|\mathcal{V}| = d$, such that $\mathcal{V}$ is not contained in any of the queries to $\mathcal{SO}$ and $\mathcal{CO}$. Let $\mathcal{X}$ be a set of secret keys with $|\mathcal{X}| = d$ and whose corresponding public keys are all contained in $\mathcal{V}$. $\mathcal{S}$ computes $\sigma' =$ Sign $(e, n, d, \mathcal{Y}, \mathcal{V}, \mathcal{X}, M)$.
   (d) $\mathcal{A}$ queries the oracles adaptively. Suppose $\mathcal{A}$ makes a total number of v' queries to $\mathcal{CO}$. The restriction is that: $v' < n - d - v$. If any of the queries to $\mathcal{CO}(pk)$ contains a public key pk such that $pk \in \mathcal{V}$, or to $\mathcal{SO}(e, \cdot, \cdot, \cdot, \mathcal{V}', \cdot)$ such that $\mathcal{V} \cap \mathcal{V}' \neq \emptyset$, $\mathcal{S}$ halts.
   (e) $\mathcal{A}$ outputs an index $\hat{\pi}$.

   We denote by

$$\mathbf{Adv}_{\mathcal{A}}^{Anon}(\lambda) = \Pr[\hat{\pi} \in \mathcal{V}] - \frac{d}{n - (v + v')}$$

   Definition 3.2 (Linkable-anonymity). *A LTRS scheme is linkably-anonymous if for any PPT adversary $\mathcal{A}$, $\mathbf{Adv}_{\mathcal{A}}^{Anon}(\lambda)$ is negligible.*

   Note: We will further show how to enhance the security into *Insider Security for Anonymity* in Section 5.2.

3. Linkability.
   Linkability for LTRS schemes is compulsory, that is, it should be infeasible for a set of signers to generate two signatures such that they are determined to be unlinked using `LTRS.Link`. The following definition/game essentially captures a scenario that an adversary tries to generate two LTRS signatures, say a (d_1, n_1)-threshold linkable ring signature and a (d_2, n_2)-threshold linkable ring signature, using strictly fewer than $d_1 + d_2$ user secret keys, so that these two signatures are determined to be unlinked using `LTRS.Link`. If the LTRS scheme is unforgeable (as defined above), then these signatures can only be generated if at least d_1 and d_2 user secret keys are known,

respectively. If less than d_1+d_2 user secret keys are known, then there must be at least one common signer to both of the signatures. Therefore, this model can effectively capture the definition of *individual-linkability* given in [26, 21]. Note that it is different from the definition of *coalition-linkability* for LTRS schemes.

Linkability for LTRS scheme is defined in the following game between the Simulator $\mathcal{S}$ and the Adversary $\mathcal{A}$ in which $\mathcal{A}$ is given access to oracles $\mathcal{JO}$, $\mathcal{CO}$ and $\mathcal{SO}$:

(a) $\mathcal{S}$ generates and gives $\mathcal{A}$ the system parameters param.
(b) $\mathcal{A}$ may query the oracles according to any adaptive strategy.
(c) $\mathcal{A}$ gives $\mathcal{S}$ an event-id $e \in \mathcal{EID}$, group sizes $n_1, n_2 \in \mathbb{N}$ (w.l.o.g. we assume $n_1 \leq n_2$), thresholds $d_1 \in \{1, \ldots, n_1\}$, $d_2 \in \{1, \ldots, n_2\}$, sets $\mathcal{Y}_1$ and $\mathcal{Y}_2$ of public keys in $\mathcal{PK}$ of sizes n_1 and n_2 resp., messages $M_1, M_2 \in \mathcal{M}$ and signatures $\sigma_1, \sigma_2 \in \Sigma$.

$\mathcal{A}$ wins the game if

(1) All public keys in $\mathcal{Y}_1 \cup \mathcal{Y}_2$ are query outputs of $\mathcal{JO}$
(2) $\mathsf{Verify}(e, n_i, d_i, \mathcal{Y}_i, M_i, \sigma_i) = \mathsf{accept}$ for $i = 1, 2$ such that σ_i are not outputs of $\mathcal{SO}$
(3) $\mathcal{CO}$ has been queried less than $d_1 + d_2$ times
(4) $\mathsf{Link}(\sigma_1,\ \sigma_2) = \mathsf{unlinked}$.

We denote by

$$\mathbf{Adv}_{\mathcal{A}}^{Link}(\lambda) = \Pr[\mathcal{A} \text{ wins the game }]$$

Definition 3.3 (Individual-Linkability). *A LTRS scheme is individually-linkable if for all PPT adversary $\mathcal{A}$, $\mathbf{Adv}_{\mathcal{A}}^{Link}$ is negligible.*

4. Non-Slanderability.
Non-slanderability ensures that no signer can generate a signature which is determined to be linked by `LTRS.Link` with another signature which is not generated by the signer. In other words, it prevents adversaries from framing honest users.

Non-Slanderability for LTRS schemes is defined in the following game between the Simulator $\mathcal{S}$ and the Adversary $\mathcal{A}$ in which $\mathcal{A}$ is given access to oracles $\mathcal{JO}$, $\mathcal{CO}$ and $\mathcal{SO}$:

(a) $\mathcal{S}$ generates and gives $\mathcal{A}$ the system parameters param.
(b) $\mathcal{A}$ may query the oracles according to any adaptive strategy.
(c) $\mathcal{A}$ gives $\mathcal{S}$ an event e, group size n, threshold d, a set of n public keys $\mathcal{Y}$, a set of d insiders $\mathcal{V} \subseteq \mathcal{Y}$, a message M. No member of $\mathcal{V}$ has been queried to $\mathcal{CO}$ or has been included in the insider set of any query to $\mathcal{SO}$. $\mathcal{S}$ uses the secret keys $\mathcal{X}$ corresponding to $\mathcal{V}$ to run $\mathsf{Sign}(e, n, d, \mathcal{Y}, \mathcal{X}, M)$ and to produce a signatures σ'.
(d) $\mathcal{A}$ queries oracles with arbitrary interleaving. Except at most $d-1$ members of $\mathcal{V}$ can be queries to $\mathcal{CO}$, or included in the insider set of any query to $\mathcal{SO}$. In particular, $\mathcal{A}$ is allowed to query any public keys which is not in $\mathcal{V}$ to $\mathcal{CO}$.
(e) $\mathcal{A}$ delivers group size n^*, threshold d^*, a set of n^* public keys $\mathcal{Y}^*$, a message M^* and a signature $\sigma^* \neq \sigma'$.

$\mathcal{A}$ wins the game if

(1) $\mathsf{Verify}(e, n^*, d^*, \mathcal{Y}^*, M^*, \sigma^*) = \mathsf{accept}$
(2) σ^* is not an output of $\mathcal{SO}$
(3) All of the public keys in $\mathcal{Y}^*, \mathcal{Y}$ are query outputs of $\mathcal{JO}$
(4) None of the public keys in $\mathcal{V}$ have been input to $\mathcal{CO}$
(5) $\mathsf{Link}(\sigma^*,\ \sigma') = \mathsf{linked}$.

We denote by

$$\mathbf{Adv}_{\mathcal{A}}^{NS}(\lambda) = \Pr[\mathcal{A} \text{ wins the game }]$$

Definition 3.4 (Non-Slanderability). *A LTRS scheme is* non-slanderable *if for all PPT adversary $\mathcal{A}$, $\mathbf{Adv}_{\mathcal{A}}^{NS}$ is negligible.*

Summarizing we have:

Definition 3.5 (Security of LTRS Schemes). *A LTRS scheme is* secure *if it is unforgeable, linkably-anonymous, linkable and non-slanderable.*

4. OUR PROPOSED THRESHOLD AND LINKABLE RING SIGNATURE SCHEME

Intuition. We modify the Fujisaki traceable ring signature [27] which is inherited from [32, 33, 34] and turn it into Linkable Threshold Ring Signature Scheme. We make use of the following observations:

- We do not need the traceable property of the Fujisaki traceable ring signature. Therefore, we do not need the n linkability tags of the Fujisaki traceable ring signature. We also drop the NIZK and NIWI proofs related to the n linkability tags. We only need one linkability tag for our linkability property. Therefore our signature is much shorter than the Fujisaki traceable ring signature.
- Note that ***the modification is not trivial!*** We need to re-design the linking tag in order to make it secure and more efficient. Therefore the approach of our linking tag is totally different from the Fujisaki scheme. Although we take the same assumptions as the Fujisaki scheme, the security proof of our scheme is also different from their scheme.

- Our Link algorithm is more efficient than the Trace algorithm of the Fujisaki traceable ring signature. It is because the Fujisaki traceable ring signature has n times more linkability tags than our proposal. We also give a new method for efficient comparison of linkability tags. Our linking complexity is $O(d \log d)$, which becomes a constant if we set the threshold value d to 1, while Fujisaki scheme requires $O(n \log n)$. Detailed comparison result is presented in Section 5.3.
- We observe that linkable ring signature implies threshold ring signature. It is because if there is d linkable ring signatures for the same message and linkable tag which are not linked with each other, then it implies that there are d distinct signers out of the n public keys. However, we still need to connect all linkable ring signatures together. Otherwise, the linkable ring signature of a signer can be extracted and used to generate another threshold ring signature (which breaks the non-slander security). It can be done by each signer signing on the one-time verification keys from d signers.

Construction. We give our linkable threshold ring signatures as follows:

- Setup: The setup algorithm runs the bilinear group generator $(N = pq, \mathbb{G}, \mathbb{G}_T, \hat{e}) \leftarrow \mathcal{G}(\lambda)$. Suppose the group generator $\mathcal{G}$ also gives the generators $g, h, h' \in \mathbb{G}$.
  The event ID space is $\mathcal{EID}$, the message space is $\mathcal{M}$ and the signature space is Σ. Let (OTGen, OTSign, OTVerify) be a secure one-time signature scheme[4] [35, 36, 37, 38, 39, 40, 41]. The message space of the OTSign is $(\mathcal{EID} \times \mathcal{M} \times \mathbb{N} \times |pk|^n \times |vk|^d \times (7 + 6\sqrt{n})\mathbb{G})$, where $|pk|$ is size of a public key, $|vk|$ is the size of a one-time verification key and n is the number of public keys in the ring signature. Let $H : \mathcal{EID} \to \mathbb{Z}_K$ be a collision resistant hash function, where K is a security parameter. The public parameters are $(N, \mathbb{G}, \mathbb{G}_T, \hat{e}, g, h, h', H)$.
- KeyGen: For user i, he picks a random $x_i, d_i \in \mathbb{Z}_N$. His public key is $y_i = g^{x_i} h^{d_i}$ and his secret key is (x_i, d_i).
- Sign: On input $(e, n, d, \mathcal{Y}, \mathcal{X}, M)$, suppose $\mathcal{Y} = \{pk_1, \ldots, pk_n\}$ is the user ring. $\mathcal{X}$ is the set of private keys of d participating signers, who cooperate to generate a (d, n) linkable threshold ring signature for the message M with an event-id e. We arrange $\mathcal{Y}$ as a $\nu \times \nu$ matrix, where $\nu = \sqrt{n}$.[5] Denote $pk_{(j,k)}$ as the public key at the j-th row and the k-th column.

  Each signer i generates $(vk_i, sk_i) \leftarrow \mathsf{OTGen}(1^\lambda)$, which is a pair of verification key and signing key for a one-time signature scheme. We assume vk can be represented by an element in $\mathbb{Z}_N$. All signers firstly publish their own vk_i.

  For each signer i with private key (x_i, d_i), we denote (j', k') as the index of pk_i in the matrix. He computes the following:

  1. Compute $\tau = H(e)$. Calculate the linkability tag $\sigma_T = g^{\frac{1}{x_i+\tau}}$.
  2. Calculate the BB signature $\sigma_{vk} = g^{\frac{1}{x_i+vk_i}}$ and give some NIWI proofs π_{LTRS} as follows
     (a) Pick a random $r, s \in \mathbb{Z}_N$ and calculate
     $$C = y_i h^r, \quad L = \sigma_{vk} h^s$$
     $$\pi_1 = g^{(vk_i+x_i)s} g^{\frac{d_i+r}{x_i+vk_i}} h^{(d_i+r)s}.$$
     (Note that π_1 is a NIWI proof that σ_{vk} is a BB signature.)
     (b) Calculate
     $$\pi_2 = g^{\frac{d_i+r}{x_i+\tau}}, \quad \pi_3 = h'^{\frac{1}{x_i+\tau}}.$$
     (Note that π_2 is a NIWI proof that σ_T is a BB signature. π_3 is the proof that the signer cannot make another $\sigma'_T \neq \sigma_T$ such that $(\sigma'_T)^q = (\sigma_T)^q$.)
     (c) Pick some random $r_l \leftarrow \mathbb{Z}_N$ and calculate
     $$C_l = h^{r_l}, \quad \pi_l^C = (g^{-1} h^{r_l})^{r_l}$$
     for $1 \leq l \leq \nu, l \neq j'$. Then set
     $$r_{j'} = -\sum_{l \neq j'} r_l, \qquad C_{j'} = g h^{r_{j'}},$$
     $$\text{and } \pi_{j'}^C = (g h^{r'_j})^{r'_j}.$$
     (Note that $C_{j'}$ is a commitment to g while other C_l are commitments to 1. The proofs π_l^C are NIWI proofs that C_l are commitments to g or 1.)
     (d) Pick some random $s_m \leftarrow \mathbb{Z}_N$ and calculate
     $$B_m = pk_{(j',m)} h^{s_m}, \; \pi_m^B = g^{-s_m} \prod_{l=1}^{\nu} \left(pk_{(l,m)}\right)^{r_l}$$
     $$\text{for } 1 \leq m \leq \nu.$$
     (Note that B_m are commitments to the public keys in row j' of the matrix. The proofs π_m^B are NIWI proofs that B_m are commitments to the public keys in row j'.)

[4] A one-time signature is a signature that its unforgeability is under a *one-time* chosen-message attack. That is, the adversary is only allowed to make at most one query to its signing oracle.

[5] Without loss of generality, we assume n is a square. If n is not a square, we simply repeat pk_1 sufficiently many times until n is a square.

(e) Pick some random $t_m \leftarrow \mathbb{Z}_N$ and calculate

$$D_m = h^{t_m}, \quad \pi_m^D = (g^{-1}h^{t_m})^{t_m}$$

$$\text{for } 1 \le m \le \nu, m \neq k'.$$

Then set $t_{k'} = -\sum_{m \neq k'} t_m$, $D_{k'} = gh^{t_{k'}}$ and $\pi_{k'}^D = (gh^{t_k'})^{t_k'}$.
(Note that $D_{k'}$ is a commitment to g while other D_m are commitments to 1. The proofs π_m^D are NIWI proofs that D_m are commitments to g or 1.)

(f) Calculate

$$\pi_C = g^{s_{k'}-r} \prod_{m=1}^{\nu} \left(pk_{(j',m)} h^{s_m} \right)^{t_m}.$$

(Note that π_C is a NIWI proof that

$$\prod_{m=1}^{\nu} \hat{e}(B_m, D_m)$$

is a commitment to $pk_{j',k'}$.)

(g) Output $\pi_{LTRS} = (C, L, \pi_1, \pi_2, \pi_3, \{C_i, \pi_i^C, B_i, \pi_i^B, D_i, \pi_i^D\}_{1 \le i \le \nu}, \pi_C)$

3. Produce the one-time signature $\sigma_{OT} = \mathsf{OTSign}_{sk_i}(e, M, d, \mathcal{Y}, \{vk_1, \ldots, vk_d\}, \sigma_T, \pi_{LTRS})$
4. Output the signature $(vk_i, \sigma_T, \pi_{LTRS}, \sigma_{OT})$ on the message M with respect to *event* and $\mathcal{Y}$.

For the d participating signers, they all generate their own linkable ring signature on the same message M and event e. Therefore, the final ring signature includes $\{vk_\ell, \sigma_{T,\ell}, \pi_{LTRS,\ell}, \sigma_{OT,\ell}\}$ for $1 \le \ell \le d$. Notice that $\sigma_{T,\ell} \neq \sigma_{T,\ell'}$ for all $\ell \neq \ell'$. The signature size is $O(d\sqrt{n})$.

- Verify: On input $(e, n, d, \mathcal{Y}, M, \sigma)$, first compute $\tau = H(e)$ and parse $\sigma = \{vk_\ell, \sigma_{T,\ell}, \pi_{LTRS,\ell}, \sigma_{OT,\ell}\}$ for $1 \le \ell \le d$. Output accept if all of the following holds

$$\sigma_{T,\ell} \neq \sigma_{T,\ell'} \text{ for all } \ell \neq \ell',$$
$$\mathsf{accept} \leftarrow \mathsf{OTVerify}_{vk_\ell}(\sigma_{OT,\ell}, e, M, d, \mathcal{Y}, \{vk_1, \ldots, vk_d\}, \sigma_{T,\ell}, \pi_{LTRS,\ell}) \text{ for some } 1 \le \ell \le d,$$
$$\pi_{LTRS,\ell} \text{ is a valid NIWI proof for } 1 \le \ell \le d.$$

The details of checking each

$$\pi_{LTRS,\ell} = (C, L, \pi_1, \pi_2, \pi_3, \{C_i, \pi_i^C, B_i, \pi_i^B, D_i, \pi_i^D\}_{1 \le i \le \nu}, \pi_C)$$

is as follows:

1. Verify that

$$\hat{e}(g^{vk_\ell}C, L) = \hat{e}(g, g) \cdot \hat{e}(h, \pi_1)$$

2. Compute $\tau = H(e)$. Verify that

$$\hat{e}(g^\tau C, \sigma_T) = \hat{e}(g, g) \cdot \hat{e}(h, \pi_2) \qquad \text{and}$$

$$\hat{e}(\sigma_T, h') = \hat{e}(g, \pi_3)$$

3. Verify that

$$\hat{e}(C_l, C_l g^{-1}) = \hat{e}(h, \pi_l^C)$$

for all $1 \le l \le \nu$ and $\prod_{l=1}^{\nu} C_l = g$.

4. Compute

$$A_m = \prod_{l=1}^{\nu} \hat{e}(C_l, pk_{l,m})$$

and verify that

$$A_m = \hat{e}(g, B_m)\hat{e}(h, \pi_m^B)$$

for all $1 \le m \le \nu$.

5. Verify that

$$\hat{e}(D_m, D_m g^{-1}) = \hat{e}(h, \pi_m^D)$$

for all $1 \le m \le \nu$ and $\prod_{m=1}^{\nu} D_m = g$.

6. Compute

$$A = \prod_{m=1}^{\nu} \hat{e}(B_m, D_m)$$

and verify that

$$A = \hat{e}(g, C)\hat{e}(h, \pi_C)$$

7. Return accept if all of the above steps verify correctly. Otherwise, output reject.

- Link: On input two signatures σ_1, σ_2 for two messages M_1, M_2 and the same event *event*, suppose $\sigma_{T,\ell}^{(1)}$ and $\sigma_{T,\ell}^{(2)}$ are from σ_1 and σ_2 respectively, for $1 \le \ell \le d$. We use $\sigma_{T,\ell}^{(1)}$ to build a binary search tree. The average and worst case complexity is $O(d \log d)$. Then for each $\sigma_{T,\ell}^{(2)}$, we look up the binary search tree to search for duplicate entry. Each lookup operation has complexity $O(\log d)$. If there is any duplicate, output linked. Otherwise, output unlinked. The overall complexity is $O(d \log d)$.

Remark: The complexity of the Link protocol in Fujisaki ring signature is $O(n^2)$. Even if we apply the binary search tree method to their scheme, the complexity is still $O(n \log n)$.

Theorem 4.1. *The threshold linkable ring signature scheme is unforgeable against insider corruption under the subgroup decision assumption in $\mathbb{G}_q$, the $q_s + K$-SDH assumption in $\mathbb{G}_p$ and the unforgeability of one-time signature.*

Theorem 4.2. *The linkable threshold ring signature scheme is linkably-anonymous if the K-DDHI assumption holds in $\mathbb{G}_N$.*

Theorem 4.3. *The threshold linkable ring signature scheme is individually-linkable if the subgroup decision assumption holds in $\mathbb{G}_p$ and $\mathbb{G}_q$, and the threshold linkable ring signature scheme is unforgeable.*

Theorem 4.4. *The threshold linkable ring signature scheme is non-slanderable if the subgroup decision assumption holds in $\mathbb{G}_q$, the SDH assumption holds in $\mathbb{G}_p$ and the one-time signature scheme is unforgeable.*

The proofs of the theorems are given in the Appendix.

5. FURTHER ANALYSIS AND COMPARISON

5.1. Linkable or Threshold Ring Signatures

Our linkable threshold ring signature is constructed from a linkable ring signature, and uses the linkability tag for threshold signing. Therefore, our proposal can also be used to construct *linkable ring signature* (with single signer) and *threshold ring signature* (which is unlinkable).

- To turn our construction into a linkable ring signature, we can simply set $d = 1$ in our construction.
- To turn our construction into a threshold ring signature which is unlinkable, we can use a nonce to replace the event-id. This nonce can be chosen by an arbitrary signer, and this number should not be re-used. By the linkability property of our construction, we can achieve the threshold property since all signers calculate a linkability tag on the same nonce for each Sign operation. By the non-slander property of our construction, different runs of the Sign protocol give threshold ring signatures which are not linked. It is because the signatures are signed on different nonce (event-id).

Therefore we have threshold and/or linkable ring signature.

5.2. Insider Security for Anonymity

For the security model of anonymity for threshold ring signatures (either linkable or not) in the literature, we only consider the anonymity similar to the ring signatures with a single signer. It means that the adversary is given a challenge signature of a ring of n signers, the adversary guesses who is one of the d real signers. It is known as the *basic anonymity* [42]. If all secret keys of the ring are known to the adversary, it is called the *anonymity against full key exposure* [42]. Our scheme is proved under this security model.

However, anonymity for threshold ring signatures is more complicated than the single signer case. We have to consider if the communication between the d signers is known to the adversary during the generation of the challenge signature. We also have to consider the case that the adversary can participate in the generation of the challenge signature, by acting as some of the d signers. To the best of the author's knowledge, no threshold ring signatures in the literature considered this insider attack from the adversary. Their security model implies a secure channel between the signers and all signers are honest during the generation of threshold ring signatures [43]. However, it may not be true in the real world. Therefore, we propose a new security model for *anonymity against full key exposure and insider attack*. The security game is modified as follows.

Linkable-Anonymity. Anonymity for LTRS schemes is defined in the following game between the Simulator $\mathcal{S}$ and the Adversary $\mathcal{A}$ in which $\mathcal{A}$ is given access to oracles $\mathcal{JO}$, $\mathcal{CO}$ and $\mathcal{SO}$:

1. $\mathcal{S}$ generates and gives $\mathcal{A}$ the system parameters param.
2. $\mathcal{A}$ may query the oracles according to any adaptive strategy. Suppose $\mathcal{A}$ makes a total number of v queries to $\mathcal{CO}$. The restriction is that: $v < n - d$.
3. $\mathcal{A}$ gives $\mathcal{S}$ event-id e, group size n, threshold $d \in \{1, \dots, n\}$, message M, a set $\mathcal{Y}$ of n public keys all of which were query outputs of $\mathcal{JO}$, and a subset $\mathcal{D} \subset \mathcal{Y}$ of $d' < d$ public keys all of which were query inputs to $\mathcal{CO}$. $\mathcal{S}$ picks randomly a subset $\mathcal{V}$ of $\mathcal{Y}$ with $|\mathcal{V}| = d - d'$, such that $\mathcal{V}$ is not contained in any of the queries to $\mathcal{SO}$ and $\mathcal{CO}$. (Note that $\mathcal{D} \cap \mathcal{V} = \emptyset$).
   Let $\bar{\mathcal{V}}$ and $\bar{\mathcal{D}}$ be a set of secret keys whose corresponding public keys are all contained in $\mathcal{V}$ and $\mathcal{D}$ respectively. $\mathcal{S}$ runs the $\mathsf{Sign}(e, n, d, \mathcal{Y}, \bar{\mathcal{V}} \cup \bar{\mathcal{D}}, M)$ protocol with $\mathcal{A}$, with $\mathcal{S}$ runs as the signers in $\mathcal{V}$ and $\mathcal{A}$ runs as the signers in $\mathcal{D}$. Eventually, the challenge signature σ' is outputted to $\mathcal{A}$.
4. $\mathcal{A}$ queries the oracles adaptively. Suppose $\mathcal{A}$ makes a total number of v' queries to $\mathcal{CO}$. The restriction is that: $v' < n - d - v$. If any of the queries to $\mathcal{CO}(pk)$ contains a public key pk such that $pk \in \mathcal{V}$, or to $\mathcal{SO}(e, \cdot, \cdot, \cdot, \mathcal{V}', \cdot)$ such that $\mathcal{V} \cap \mathcal{V}' \neq \emptyset$, $\mathcal{S}$ halts.
5. $\mathcal{A}$ outputs an index $\hat{\pi}$.

We denote by

$$\mathbf{Adv}_{\mathcal{A}}^{Anon}(\lambda) = \Pr[\hat{\pi} \in \mathcal{V}] - \frac{d - d'}{n - (v + v')}.$$

Definition 5.1 (Linkable-anonymity under Full Key Exposure and Insider Attack). *A LTRS scheme is linkably-anonymous against full key exposure and insider attack if for any PPT adversary $\mathcal{A}$, $\mathbf{Adv}_{\mathcal{A}}^{Anon}(\lambda)$ is negligible.*

We can define the *anonymity under full key exposure and insider attack* for ring signatures (which is unlinkable) similarly.

Theorem 5.1. *Our linkable threshold ring signature scheme is linkably-anonymous under full key exposure and insider attack if the K-DDHI assumption holds in $\mathbb{G}_N$.*

The proof of this theorem is very similar to the proof of theorem 4.2. For the challenge signature, we can see that all communications between signers are the NIWI proofs. Therefore, the adversary cannot use it to break the anonymity of the scheme.

5.2.1. Insider Security of Other Threshold Ring Signatures.

For most threshold ring signature schemes, there exists a party who is responsible to generate the "partial" signatures (part of the final signatures) for the non-participating members in the ring. These "partial" signatures are generated from the public key / identities of those non-participating members. And these "partial" signatures are indistinguishable from the partial signatures generated from the real signers (which require the secret key of those signers). Examples can be found in [11, 21, 44, 45, 26, 43]. Therefore, this party knows who are the real signers. Previous security model does not capture the attack that the adversary is acting as this party.

5.3. Comparison

Linkable Threshold Ring Signatures. Tsang *et al.* [21] proposed the first linkable threshold ring signature scheme based on the strong RSA problem and the Decisional Diffie-Hellman (DDH) problem. Tsang *et al.* [26] proposed another linkable threshold ring signature scheme which has similar complexity and is secure under the same assumptions. These two schemes are secure in the random oracle model (ROM) only.

Our proposal is secure in the *standard model* under the SDH assumption, the subgroup decision assumption, the DDHI assumption and the security of the one-time signature (OTS) scheme. Our proposal is more efficient than the previous schemes if $d < \sqrt{n}$. It is possible in the real world applications, when the signers are concern about their anonymity. For example, a 10-out-of-1000 threshold ring signature appears to be "more anonymous" to a 10-out-of-100 threshold ring signature. The size of the ring n is much larger the number of the actual signers d, then our proposal is more efficient.

Moreover, the size of all linkability tags is $O(d)$ in our scheme, while it is $O(n)$ in other schemes. Therefore, the running time of the Link protocol is smaller in our scheme. Since we propose an optimization of the Link protocol, the running time is further reduced to $O(d \log d)$. We summerize the comparison in Table 1.

Threshold Ring Signatures. Bresson *et al.* [11] proposed the threshold ring signature scheme based on the RSA problem. However, this scheme is not efficient. Liu *et al.* [45] proposed a threshold ring signature using Shamir secret sharing. The security of the scheme is based on the discrete logarithm (DL) and the RSA-collision problem. Tsang *et al.* [26] proposed the first identity-based threshold ring signature, whose security is based on the computational Diffie-Hellman (CDH) problem. Tsang *et al.* also pointed out a flaw in the security proof of the identity-based threshold ring signature by Han *et al.* [46]. Tsang *et al.* [26] proposed a threshold ring signature, whose security is based on the strong RSA problem and DDH problem.

Melchor *et al.* [47] and Dallot and Vergnaud [48] proposed threshold ring signatures based on coding theory. The security of the scheme of Melchor *et al.* [47] is based on the Minimum Distance problem, while the security of the scheme of Dallot and Vergnaud [48] is based on the Goppa Parameterized Bounded Decoding (GPBD) problem and Goppa Code Distinguisher (GD).

Cayrel *et al.* [49] proposed a lattice-based threshold ring signatures. The security is based on the Short Integer Solution problem.

Yuen *et al.* [43] proposed the first threshold ring signatures in the standard model. The security is based on the CDH problem. We summerize the comparison in Table 2.

Linkable Ring Signatures. Liu *et al.* [16] proposed the first linkable ring signature scheme based on the DL problem and the DDH problem. Tsang and Wei [18] proposed a linkable ring signature, whose security is based on the link decisional RSA (LD-RSA) problem and the DDH. Liu and Wong [19] proposed a linkable ring signature, whose security is based on the DL problem and the DDH. Au *et al.* [23] improved the scheme [21] by showing the new scheme is secure in a stronger security model, under the LD-RSA assumption, the DDH assumption and the strong RSA assumption. Zheng *et al.* [25] proposed a linkable ring signatures from linear feedback shift register. Its security is based on the state-based DL (S-DL) assumption and the state-based decisional product Diffie-Hellman (S-DPDH) assumption. Recently, Fujisaki [27] proposed the first linkable ring signature scheme without random oracles which relies on various assumptions. We summerize the comparison in Table 3.

6. CONCLUSION

In this paper, we presented the first linkable threshold ring signature (LTRS) without random oracles. When compared to previous LTRS schemes (which are only secure in the random oracle model), we also enjoy significant efficiency improvement. Our scheme can be regarded as a "regular" threshold ring signature scheme (*i.e.,* without linkability). The signature size of our scheme is shorter than all previous threshold ring

TABLE 1. Comparison of (d,n)-Linkable Threshold Ring Signatures

Scheme	Signature Size	Security	Model	Linking Complexity	Sign Computation[a]	Verify Computation[a]
Tsang *et al.* [21]	$O(n)$	strong RSA, DDH	ROM	$O(n^2)$	$2(n+d)E+2(n-d)M$	$3nM$
Tsang *et al.* [26]	$O(n)$	strong RSA, DDH	ROM	$O(n^2)$	$(n+4d)E+4nM$	$5nM$
Our scheme	$O(d\sqrt{n})$	SDH, Subgp, DDHI, OTS	standard	$O(d\log d)$	$(8d+4d\sqrt{n})E+$ $(4d+2d\sqrt{n})M+d\mathsf{OTS}$	$2dE+8d(1+$ $\sqrt{n})P+d\mathsf{OTV}$

[a] When we come across the computation of sign and verify, we use E to represent an exponentiation, M to represent a multi-bases exponentiation which is equal to the cost of approximate 1.3 exponentation, P to represent a pairing, OTS to represent a one-time signature signing and OTV to represent a one-time signature verification.

TABLE 2. Comparison of (d,n)-Threshold Ring Signatures

Scheme	Signature Size	Security	Model	Sign Computation[a]	Verify Computation[a]
Bresson *et al.* [11]	$O(n\log n)$	RSA	ROM	$(d2^d\log n)C+dC$ $+(n2^d\log n)E+nE$	$(d2^d\log n)C$ $+(d2^d\log n)E$
Liu *et al.* [45]	$O(n)$	DL, RSA-Collision	ROM	$dE+(n-d)M$ (DL Ver.) $dE+(n-d)E$ (RSA Ver.)	nM (DL Ver.) nE (RSA Ver.)
Chow *et al.* [44]	$O(n)$	CDH	ROM	$nE+nM$	$nE+(n+1)P$
Melchor *et al.* [47]	$O(n)$	Minimum Distance	ROM	$O(n^2)$ matrix op.[b]	$O(n^2)$ matrix op.[b]
Dallot and Vergnaud [48]	$O(n)$	GPBD, GD	ROM	n matrix op. $+nC$	n matrix op. $+nC$
Tsang *et al.* [26, 21]	$O(n)$	strong RSA, DDH	ROM	$2(n+d)E+2(n-d)M$ ([21]) $(n+4d)E+4nM$ ([26])	$3nM$ ([21]) $5nM$ ([26])
Cayrel *et al.* [49]	$O(n)$	Short Integer Solution	ROM	$O(n\log n)$ matrix op.[c]	$O(n\log n)$ matrix op.[c]
Yuen *et al.* [43]	$O(n)$	CDH	standard	$(n+d+1)E+(n+d)M$	$(3n+3)P+E$
Our scheme	$O(d\sqrt{n})$	SDH, Subgp, DDHI, OTS	standard	$(8d+4d\sqrt{n})E+$ $(4d+2d\sqrt{n})M+d\mathsf{OTS}$	$2dE+8d(1+$ $\sqrt{n})P+d\mathsf{OTV}$

[a] Same as Table 1, when we come across the computation of sign and verify, we use E to represent an exponentiation, M to represent a multi-bases exponentiation and P to represent a pairing. In addition, we use C to represent a symmetric cipher operation.

[b] The scheme in [47] is based on coding theory. The signing and verification processes require $140m^2n$ matrix operations as described in their paper. The size of the matrix is $m\times(m-k)$ where m,k are some security parameters.

[c] The scheme in [49] is also based on coding theory. The authors stated that if $n=100$, the required number of runs of matrix operation is about 111 for signing and verification for an acceptable security level.

TABLE 3. Comparison of $(1,n)$-Linkable Ring Signatures

Scheme	Signature Size	Security	Model	Linking Complexity	Sign Computation[a]	Verify Computation[a]
Liu *et al.* [16]	$O(n)$	DL, DDH	ROM	$O(1)$	$2(n-1)M+3E$	$2nM$
Tsang and Wei [18]	$O(1)$	LD-RSA, DDH	ROM	$O(1)$	$(2+n)E+7M$	$7M$
Liu and Wong [19]	$O(n)$	DL, DDH	ROM	$O(1)$	$E+2M$	$2M$
Au *et al.* [23]	$O(1)$	LD-RSA, DDH, strong RSA	ROM	$O(1)$	$(2+n)E+7M$	$7M$
Zheng *et al.* [25]	$O(n)$	S-DL, S-DPDH	ROM	$O(1)$	$(14n+2)$ seq. op.[b]	$(14n+2)$ seq. op.[b]
Tsang *et al.* [26, 21]	$O(n)$	strong RSA, DDH	ROM	$O(n^2)$	$2(n+1)E$ $+2(n-1)M$ ([21]) $(n+4)E$ $+4nM$ ([26])	$3nM$ ([21]) $5nM$ ([26])
Fujisaki [27]	$O(\sqrt{n})$	SDH, Subgp, DDHI, OTS	standard	$O(n\log n)$	$(6+13\sqrt{n})E$ $+(5+n+2\sqrt{n})M$ $+2\sqrt{n}P+\mathsf{OTS}$	nM $+(8+2n+12\sqrt{n})P$ $+\mathsf{OTV}$
Our scheme	$O(\sqrt{n})$	SDH, Subgp, DDHI, OTS	standard	$O(1)$	$(8+4\sqrt{n})E+$ $(4+2\sqrt{n})M+\mathsf{OTS}$	$2E+8(1+$ $\sqrt{n})P+1$

[a] Same as Table 1, when we come across the computation of sign and verify, we use E to represent an exponentiation, M to represent a multi-bases exponentiation and P to represent a pairing.

[b] The scheme in [25] relies on Linear Feedback Shift Register (LFSR) and the computations are sequence operations.

signature schemes if $d < \sqrt{n}$. Furthermore, if we set the threshold value to be 1, our scheme is a (1-out-of-n) linkable ring signature scheme. The linking complexity of our scheme is much faster than the Fujisaki scheme, which is the only linkable ring signature scheme secure in the standard model.

We also enhanced the security model of threshold ring signature scheme, by allowing the adversary to be any insider interacting with other participating signers. We claim that this security model should be more practical in the real life.

REFERENCES

[1] Rivest, R. L., Shamir, A., and Tauman, Y. (2001) How to Leak a Secret. *ASIACRYPT 2001*, Lecture Notes in Computer Science, **2248**, pp. 552–565. Springer.

[2] Abe, M., Ohkubo, M., and Suzuki, K. (2002) 1-out-of-n Signatures from a Variety of Keys. *ASIACRYPT 2002*, Lecture Notes in Computer Science, **2501**, pp. 415–432. Springer.

[3] Zhang, F. and Kim, K. (2002) ID-Based Blind Signature and Ring Signature from Pairings. *ASIACRYPT 2002*, Lecture Notes in Computer Science, **2501**, pp. 533–547. Springer.

[4] Boneh, D., Gentry, C., Lynn, B., and Shacham, H. (2003) Aggregate and Verifiably Encrypted Signatures from Bilinear Maps. *EUROCRYPT 2003*, Lecture Notes in Computer Science, **2656**, pp. 416–432. Springer.

[5] Wong, D. S., Fung, K., Liu, J. K., and Wei, V. K. (2003) On the rs-code construction of ring signature schemes and a threshold setting of rst. *ICICS 2003*, Lecture Notes in Computer Science, **2836**, pp. 34–46. Springer.

[6] Dodis, Y., Kiayias, A., Nicolosi, A., and Shoup, V. (2004) Anonymous Identification in Ad Hoc Groups. *EUROCRYPT 2004*, Lecture Notes in Computer Science, **3027**, pp. 609–626. Springer.

[7] Chow, S. S. M., Yiu, S.-M., and Hui, L. C. K. (2005) Efficient Identity Based Ring Signature. *ACNS 2005*, Lecture Notes in Computer Science, **3531**, pp. 499–512. Also available at Cryptology ePrint Archive, Report 2004/327.

[8] Chaum, D. and van Heyst, E. (1991) Group signatures. In Davies, D. W. (ed.), *EUROCRYPT 1991*, Lecture Notes in Computer Science, **547**, pp. 257–265. Springer.

[9] Camenisch, J. and Stadler, M. (1997) Efficient Group Signature Schemes for Large Groups (Extended Abstract). *CRYPTO 97*, Lecture Notes in Computer Science, **1294**, pp. 410–424. Springer.

[10] Bellare, M., Micciancio, D., and Warinschi, B. (2003) Foundations of Group Signatures: Formal Definitions, Simplified Requirements, and a Construction Based on General Assumptions. *EUROCRYPT 2003*, Lecture Notes in Computer Science, **2656**, pp. 614–629. Springer.

[11] Bresson, E., Stern, J., and Szydlo, M. (2002) Threshold ring signatures and applications to ad-hoc groups. In Yung, M. (ed.), *CRYPTO 2002*, Lecture Notes in Computer Science, **2442**, pp. 465–480. Springer.

[12] Susilo, W. and Mu, Y. (2004) Non-Interactive Deniable Ring Authentication. *ICISC 2003*, Lecture Notes in Computer Science, **2971**, pp. 386–401. Springer.

[13] Liu, J. K., Yuen, T. H., and Zhou, J. (2011) Forward secure ring signature without random oracles. *ICICS*, Lecture Notes in Computer Science, **7043**, pp. 1–14. Springer.

[14] Susilo, W., Mu, Y., and Zhang, F. (2004) Perfect Concurrent Signature Schemes. *ICICS 2004*, October, Lecture Notes in Computer Science, **3269**, pp. 14–26. Springer.

[15] Laguillaumie, F. and Vergnaud, D. (2004) Multi-designated Verifiers Signatures. *ICICS 2004*, Malaga, Spain, October, Lecture Notes in Computer Science, **3269**, pp. 495–507. Springer.

[16] Liu, J. K., Wei, V. K., and Wong, D. S. (2004) Linkable Spontaneous Anonymous Group Signature for Ad Hoc Groups (Extended Abstract). *ACISP*, Lecture Notes in Computer Science, **3108**. Springer.

[17] Chow, S. S. M., Liu, J. K., and Wong, D. S. (2008) Robust receipt-free election system with ballot secrecy and verifiability. *NDSS*. The Internet Society.

[18] Tsang, P. P. and Wei, V. K. (2005) Short linkable ring signatures for e-voting, e-cash and attestation. *ISPEC 2005*, Lecture Notes in Computer Science, **3439**, pp. 48–60. Springer.

[19] Liu, J. K. and Wong, D. S. (2005) Linkable ring signatures: Security models and new schemes. *ICCSA (2)*, Lecture Notes in Computer Science, **3481**, pp. 614–623. Springer.

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

[21] Tsang, P. P., Wei, V. K., Chan, T. K., Au, M. H., Liu, J. K., and Wong, D. S. (2004) Separable linkable threshold ring signatures. *INDOCRYPT 2004* Lecture Notes in Computer Science, pp. 384–398. Springer.

[22] Au, M. H., Liu, J. K., Susilo, W., and Yuen, T. H. (2006) Constant-size ID-based linkable and revocable-iff-linked ring signature. *INDOCRYPT 2006*, Lecture Notes in Computer Science, **4329**, pp. 364–378. Springer.

[23] Au, M. H., Chow, S. S. M., Susilo, W., and Tsang, P. P. (2006) Short linkable ring signatures revisited. *EuroPKI*, Lecture Notes in Computer Science, **4043**, pp. 101–115. Springer.

[24] Au, M. H., Liu, J. K., Susilo, W., and Yuen, T. H. (2007) Certificate based (linkable) ring signature. *ISPEC 2007*, Lecture Notes in Computer Science, **4464**, pp. 79–92. Springer.

[25] Zheng, D., Li, X., Chen, K., and Li, J. (2007) Linkable ring signatures from linear feedback shift register. *EUC Workshops*, Lecture Notes in Computer Science, **4809**, pp. 716–727. Springer.

[26] Tsang, P. P., Au, M. H., Liu, J. K., Susilo, W., and Wong, D. S. (2010) A suite of non-pairing id-based threshold ring signature schemes with different levels of anonymity. *ProvSec 2010*, Lecture Notes in Computer Science, **6402**, pp. 166–183. Springer.

[27] Fujisaki, E. (2011) Sub-linear size traceable ring signatures without random oracles. *CT-RSA 2011*,

Lecture Notes in Computer Science, **6558**, pp. 393–415. Springer.

[28] Wang, H. and Zhao, S. (2010) Cryptanalysis of several linkable ring signature schemes. *NSWCTC 2010*, pp. 302–305. IEEE Computer Society.

[29] Boneh, D. and Boyen, X. (2004) Short signatures without random oracles. *EUROCRYPT 2004*, Lecture Notes in Computer Science, **3027**, pp. 56–73. Springer.

[30] Boneh, D., Goh, E.-J., and Nissim, K. (2005) Evaluating 2-dnf formulas on ciphertexts. *TCC*, Lecture Notes in Computer Science, **3378**, pp. 325–341. Springer.

[31] Tsang, P. P. (2005). Cryptography in privacy-preserving applications. Master Thesis, The Chinese University of Hong Kong.

[32] Chandran, N., Groth, J., and Sahai, A. (2007) Ring signatures of sub-linear size without random oracles. *ICALP 2007*, Lecture Notes in Computer Science, **4596**, pp. 423–434. Springer.

[33] Groth, J., Ostrovsky, R., and Sahai, A. (2006) Non-interactive zaps and new techniques for nizk. *CRYPTO*, Lecture Notes in Computer Science, **4117**, pp. 97–111. Springer.

[34] Groth, J., Ostrovsky, R., and Sahai, A. (2006) Perfect non-interactive zero knowledge for np. *EUROCRYPT*, Lecture Notes in Computer Science, **4004**, pp. 339–358. Springer.

[35] Lamport, L. (1979). Constructing digital signatures from a one-way function. Technical Report SRI-CSL-98, SRI International Computer Science Laboratory.

[36] Merkle, R. C. (1987) A digital signature based on a conventional encryption function. *CRYPTO*, Lecture Notes in Computer Science, **293**, pp. 369–378. Springer.

[37] Merkle, R. C. (1989) A certified digital signature. *CRYPTO*, Lecture Notes in Computer Science, **435**, pp. 218–238. Springer.

[38] Bleichenbacher, D. and Maurer, U. M. (1996) Optimal tree-based one-time digital signature schemes. *STACS*, Lecture Notes in Computer Science, **1046**, pp. 363–374. Springer.

[39] Bleichenbacher, D. and Maurer, U. M. (1996) On the efficiency of one-time digital signatures. *ASIACRYPT*, Lecture Notes in Computer Science, **1163**, pp. 145–158. Springer.

[40] Perrig, A. (2001) The biba one-time signature and broadcast authentication protocol. *ACM Conference on Computer and Communications Security*, pp. 28–37. ACM.

[41] Reyzin, L. and Reyzin, N. (2002) Better than biba: Short one-time signatures with fast signing and verifying. *ACISP*, Lecture Notes in Computer Science, **2384**, pp. 144–153. Springer.

[42] Bender, A., Katz, J., and Morselli, R. (2006) Ring signatures: Stronger definitions, and constructions without random oracles. *TCC*, Lecture Notes in Computer Science, **3876**, pp. 60–79. Springer.

[43] Yuen, T. H., Liu, J. K., Au, M. H., Susilo, W., and Zhou, J. (2011) Threshold Ring Signatures without Random Oracles. *ASIACCS 2011*, pp. 261–267. ACM Press.

[44] Chow, S. S., Hui, L. C., and Yiu, S. (2004) Identity based threshold ring signature. *ICISC 2004*, Seoul, Korea, Lecture Notes in Computer Science, **3506**, pp. 218–232. Springer. Also available at Cryptology ePrint Archive, Report 2004/179.

[45] Liu, J. K., Wei, V. K., and Wong, D. S. (2003) A separable threshold ring signature scheme. *ICISC 2003*, Lecture Notes in Computer Science, **2971**, pp. 352–369. Springer.

[46] Han, J., Xu, Q., and Chen, G. (2008) Efficient id-based threshold ring signature scheme. *EUC (2)*, pp. 437–442. IEEE Computer Society.

[47] Melchor, C. A., Cayrel, P.-L., and Gaborit, P. (2008) A new efficient threshold ring signature scheme based on coding theory. *PQCrypto*, Lecture Notes in Computer Science, **5299**, pp. 1–16. Springer.

[48] Dallot, L. and Vergnaud, D. (2009) Provably secure code-based threshold ring signatures. *IMA Int. Conf.*, Lecture Notes in Computer Science, **5921**, pp. 222–235. Springer.

[49] Cayrel, P.-L., Lindner, R., Rückert, M., and Silva, R. (2010) A lattice-based threshold ring signature scheme. *LATINCRYPT*, Lecture Notes in Computer Science, **6212**, pp. 255–272. Springer.

APPENDIX A. ON THE "ATTACK" OF LINKABLE RING SIGNATURE BY WANG AND ZHAO [45]

In [28], the authors claimed all existing linkable ring signature schemes suffer from linkability attack. Specifically, it is claimed that a malicious signer can generate two ring signatures that are not linked together. This break the requirement of linkability. It is also claimed that a malicious signer can generate a signature that is linked to another signature generated by an honest signer. This break the requirement of non-slanderability.

A careful look into the paper reveals that the attacks are carried out in a different model which, we think is just too strong and unreasonable.

Using the terminology of the paper, the malicious signer is in fact holding two secret keys, x_π and x_s, of the members in the ring. In the attack against linkability, the malicious signer first obtains a signature σ which originates from singer whose secret key is x_s. Next, the malicious signer uses a seperate secret key x_π to generate another signature σ_1. The authors go on to claim that this malicious signer has successfully broken the linkability property since σ and σ_1 does not linked together. This claim is unreasonable since someone holding two secret keys should be allowed to generate two ring signatures that are not linked together.

The so-called attack against non-slanderability works in a similar fashion. In the attack, the malicious signer obtains a signature σ from the signing oracle where the secret key of its signer is x_s. Strangely, it is assumed that, in this attack, the value of x_s is known to this malicious signer. This attacker simply generates another signature σ' using the knowledge of x_s. Of course, σ and σ' links together and the authors claimed that this malicious singer has successfully broken the non-slanderability property. This claim is unreasonable since the attacker knows the secret key of the victim and thus it is entirely possible for him/her to slander the victim.

APPENDIX B. SECURITY PROOF

Appendix B.1. Proof of Theorem 4.1

Proof. Let G_0 be the original unforgeability game. Let G_1 be the same as G_0, except that h is selected from the sub-group $\mathbb{G}_q$ instead of $\mathbb{G}$. If the subgroup decision assumption holds, the adversary $\mathcal{A}$ cannot distinguish G_0 and G_1. Now we simulate G_1 as follows.

Setup. The simulator $\mathcal{S}$ runs the bilinear group generator $(N = pq, \mathbb{G}, \mathbb{G}_T, \hat{e}) \leftarrow \mathcal{G}(1^\lambda)$. Suppose $\mathcal{D}$ is the challenger of the BB signature in $\mathbb{G}_p$ and $\mathcal{S}$ tries to forge a BB signature.

Firstly, $\mathcal{S}$ runs $(vk_i, sk_i) \leftarrow \mathsf{OTGen}(\lambda)$ for $1 \leq i \leq q_s$. According to the weak unforgeability model of the BB signature, $\mathcal{S}$ gives $\{1, \ldots, K\}, \{vk_1, \ldots, vk_{q_s}\}$ as the "message" of the BB signature. $\mathcal{D}$ gives the public parameter g_p and the public key g_p^α of the BB signature to $\mathcal{S}$. $\mathcal{D}$ also gives the BB signatures for all these messages to $\mathcal{S}$.

Denote $g = g_p^{1/q}$. $\mathcal{S}$ can calculate $g^\alpha = (g_p^\alpha)^{1/q}$. $\mathcal{S}$ picks a random $\beta \in \mathbb{Z}_N$ and $h \in \mathbb{G}_q$ using p. $\mathcal{S}$ sets $h' = g^\beta$. Finally, $\mathcal{S}$ randomly chooses a collision resistant hash function $H : \mathcal{EID} \to \mathbb{Z}_K$. Then $\mathcal{S}$ gives the public parameters $(N, \mathbb{G}, \mathbb{G}_T, \hat{e}, g, h, h', H)$ to the adversary $\mathcal{A}$.

Assume that $\mathcal{A}$ makes q_j query to $\mathcal{JO}$. $\mathcal{S}$ picks τ^* as the challenge signer. For $i = 1, \ldots, q_j$, $\mathcal{S}$ picks random $x_i, d_i \in \mathbb{Z}_N$ and sets:

$$pk_i = \begin{cases} g^{x_i} h^{d_i} & \text{if } i \neq \tau^*, \\ (g^\alpha) h^{d_i} & \text{if } i = \tau^*. \end{cases}$$

$\mathcal{S}$ stores the set of public keys $\{pk_i\}_{i=1}^n$.

Oracle Simulation. $\mathcal{S}$ simulates the oracles as follows:

- $\mathcal{JO}$: on the i-th query, $\mathcal{S}$ returns pk_i.
- $\mathcal{CO}(pk_i)$: If $i = \tau^*$, $\mathcal{S}$ declares failure and exits. Otherwise, $\mathcal{S}$ returns (x_i, d_i).
- $\mathcal{SO}(e, n, d, \mathcal{Y}, \mathcal{V}, M)$: On input a message M, a set of n public keys $\mathcal{Y} = \{pk'_i\}_{i=1}^n$, and a set of d signers $\mathcal{V}$, $\mathcal{S}$ calculates all $\{vk_\ell, \sigma_{T,\ell}, \pi_{LTRS,\ell}, \pi_{OT,\ell}\}$ according to the Sign algorithm, for all $pk_\ell \in \mathcal{V}$ and $\ell \neq \tau^*$. If $pk_{\tau^*} \in \mathcal{V}$, $\mathcal{S}$ calculates $\tau = H(e) \in \{1, \ldots, K\}$. $\mathcal{S}$ retrieves the BB signatures σ_{vk_i} and σ_T for messages vk_i and τ, respectively. $\mathcal{S}$ uses σ_{vk_i} and d_{τ^*} to compute:

  $$L = \sigma_{vk_i} h^s, \quad \pi_1 = (g^{vk_i} g^\alpha)^s \sigma_{vk_i}^{d_{\tau^*}+r} h^{(d_{\tau^*}+r)s}.$$

  $\mathcal{S}$ uses σ_T and d_{τ^*} to compute:

  $$\pi_2 = \sigma_T^{d_{\tau^*}+r}, \quad \pi_3 = \sigma_T^\beta.$$

  Finally, $\mathcal{S}$ calculates the rest of the signature according to the Sign algorithm, using the one-time signing key sk_i.

Output. $\mathcal{A}$ returns $(e^*, n^*, d^*, \mathcal{Y}^*, M^*, \sigma^*)$. If $pk_{\tau^*} \notin \mathcal{Y}^*$, then $\mathcal{S}$ aborts.

For each $\{vk_\ell, \sigma_{T,\ell}, \pi_{LTRS,\ell}, \sigma_{OT,\ell}\}$, where $1 \leq \ell \leq d$, Denote $\pi_{LTRS,\ell} = (C, L, \pi_1, \pi_2, \pi_3, \{C_i, \pi_i^C, B_i, \pi_i^B, D_i, \pi_i^D\}_{1 \leq i \leq \nu}, \pi_C)$.

By raising $\hat{e}(C_l, C_l g^{-1}) = \hat{e}(h, \pi_l^C)$ to the q-th power, we can see that either C_l or $C_l g^{-1}$ has order p. Since $\prod_{l=1}^\nu C_l = g$, we have one C_l committed to g and the others are committed to 1. Denote such l as j^*. By raising $\hat{e}(D_m, D_m g^{-1}) = \hat{e}(h, \pi_m^D)$ to the q-th power, we can see that either D_m or $D_m g^{-1}$ has order p. Since $\prod_{m=1}^\nu D_m = g$, we have one D_m committed to g and the others are committed to 1. Denote such m as k^*.

Since $\prod_{l=1}^\nu \hat{e}(C_l, vk_{l,m}) = \hat{e}(g, B_m)\hat{e}(h, \pi_m^B)$ for all $1 \leq m \leq \nu$, we can raise it to the q-th power. We have $\hat{e}(g, pk_{j^*,m})^q = \hat{e}(g, B_m)^q$. Since $\prod_{m=1}^\nu \hat{e}(B_m, D_m) = \hat{e}(g, C_{\ell^*})\hat{e}(h, \pi^C)$, we can raise it to the q-th power. We have $\hat{e}(g, C)^q = \hat{e}(B_{k^*}, g)^q = \hat{e}(g, pk_{j^*,k^*})^q$. Therefore C is committed to pk_{j^*,k^*}.

If $pk_{j^*,k^*} = pk_{\tau^*}$, $\mathcal{S}$ raises $\hat{e}(g^{vk}C, L) = \hat{e}(g, g) \cdot \hat{e}(h, \pi_1)$ to the q-th power. We have

$$\hat{e}(g^{vk} g^\alpha, L)^q = \hat{e}(g, g)^q.$$

If L is not a previous signing oracle output, then $\mathcal{S}$ returns L^q as the forgery to the BB signature in $\mathbb{G}_p$ for the "message" vk.

Otherwise (L is the same as the previous signing oracle output), check if the corresponding $\sigma_{OT,\ell}$ is the same as the previous signing oracle output. If they are not the same, it implies that $\sigma_{OT,\ell}$ is a forgery of the one-time signature for the verification key vk_ℓ. If they are the same, observe that $\sigma_{OT,\ell}$ is a one-time signature on all verification keys $\{vk_1, \ldots, vk_d\}$. Since the signature outputted by $\mathcal{A}$ cannot be the same as the previous signing oracle output, then $\mathcal{A}$ must forge a the one-time signature for some verification key vk_i, where $i = 1, \ldots, d$.

If such $pk_{j^*,k^*} \neq pk_{\tau^*}$ for all $\pi_{LTRS,\ell}$, then $\mathcal{S}$ aborts.

Analysis. The probability of not asking pk_τ in the corruption oracle is $1 - \frac{q_c}{q_j}$. The probability of $pk_{\tau^*} \in \mathcal{Y}^*$ and $pk_{j^*,k^*} = pk_{\tau^*}$ for some $\pi_{LTRS,\ell}$ in the output phase is $\frac{d^*}{n^*}$. Note that the BB signature is weakly unforgeable if the SDH assumption holds. Therefore $\mathcal{S}$ solves the $q_s + K$-SDH problem, the subgroup decision problem or forges a one-time signature with probability $\frac{d^*}{n^*}(1 - \frac{q_c}{q_j})$, where q_s, q_c, q_j is the number of $\mathcal{SO}$, $\mathcal{CO}$ and $\mathcal{JO}$ respectively. □

Appendix B.2. Proof of Theorem 4.2

Proof. Setup. The simulator $\mathcal{S}$ is given the K-DDHI problem instance $(N, \mathbb{G}, \mathbb{G}_T, \hat{e}, g, A_1, A_2, \ldots, A_K)$, where

$$A_i = \begin{cases} g^{\frac{1}{x+i}} & \text{if } i \neq \bar{\tau}, \\ g^{\frac{1}{x+\bar{\tau}}} \text{ or } \alpha \in_R \mathbb{G} & \text{if } i = \bar{\tau}. \end{cases}$$

$\mathcal{S}$ is asked to determine whether $A_{\bar{\tau}} = g^{\frac{1}{x+\bar{\tau}}}$ or not. $\mathcal{S}$ randomly picks $\beta, \gamma \in \mathbb{Z}_N$ and sets

$$h = g^{\beta}, \quad h' = g^{\gamma}.$$

Finally, $\mathcal{S}$ randomly chooses a collision resistant hash function $H : \mathcal{EID} \times \{0,1\}^* \to \mathbb{Z}_K$. Then $\mathcal{S}$ gives the public parameters $(N, \mathbb{G}, \mathbb{G}_T, \hat{e}, g, h, h', H)$ to the adversary $\mathcal{A}$.

Assume that $\mathcal{A}$ makes $q_j = K$ query to $\mathcal{JO}$. $\mathcal{S}$ picks τ^* as the challenge signer. For $i = 1, \ldots, q_j$, $\mathcal{S}$ picks random $x_i, d_i, \bar{x} \in \mathbb{Z}_N$ and sets:

$$pk_i = \begin{cases} g^{x_i} h^{d_i} & \text{if } i \neq \tau^*, \\ g^{\bar{x}} & \text{if } i = \tau^*. \end{cases}$$

$\mathcal{S}$ stores the set of public keys $\{pk_i\}_{i=1}^n$. In the later part, we will implicitly set $pk_i = g^x h^{d_i}$, where x is unknown from the problem instance. It implies that $d_i = (\bar{x} - x)/\beta$, which is not known as well. We will simulate the NIWI proofs without using x and d_i.

Oracle Simulation. $\mathcal{S}$ simulates the oracles as follows:

- $\mathcal{JO}$: on the i-th query, $\mathcal{S}$ returns pk_i.
- $\mathcal{CO}(pk_i)$: If $i = \tau^*$, $\mathcal{S}$ declares failure and exits. Otherwise, $\mathcal{S}$ returns (x_i, d_i).
- $\mathcal{SO}(e, n, d, \mathcal{Y}, \mathcal{V}, M)$: On input a message M, a set of n public keys $\mathcal{Y} = \{pk'_i\}_{i=1}^n$, and a set of d signers $\mathcal{V}$, $\mathcal{S}$ calculates all $\{vk_\ell, \sigma_{T,\ell}, \pi_{LTRS,\ell}, \pi_{OT,\ell}\}$ according to the Sign algorithm, for all $pk_\ell \in \mathcal{V}$ and $\ell \neq \tau^*$.

  If $pk_{\tau^*} \in \mathcal{V}$, $\mathcal{S}$ runs $(vk, sk) \leftarrow \mathsf{OTGen}(\lambda)$ and $\tau = H(e)$. If $\tau = \bar{\tau}$, then $\mathcal{S}$ declares failure and exits. Otherwise, $\mathcal{S}$ uses A_τ from the problem instance and the trapdoor β to complete the NIZK proof. $\mathcal{S}$ picks $r, \bar{L} \in \mathbb{Z}_N$ and computes:

  $$\sigma_T = A_\tau, \quad C = g^{\bar{x}} h^r, \quad \pi_2 = (g^{-1} A_\tau^{\tau+\bar{x}+\beta r})^{1/\beta},$$

  $$L = g^{\bar{L}}, \quad \pi_1 = (g^{(vk+\bar{x}+\beta r)\bar{L}-1})^{1/\beta}, \quad \pi_3 = \sigma_T^{\gamma}.$$

  It is easy to check that π_1, π_2, π_3 can pass the verification. Finally, $\mathcal{S}$ calculates the rest of the signature according to the Sign algorithm.

Challenge. At some point, $\mathcal{A}$ outputs a message M^*, an event-id e^*, a set of n^* public keys $\mathcal{Y}^*$ and a threshold d^*. $\mathcal{S}$ picks a random subset $\mathcal{V}^*$ of $\mathcal{Y}^*$ with $|\mathcal{V}^*| = d^*$, such that

- $pk_{\tau^*} \in \mathcal{V}^*$;
- any public key in $\mathcal{V}^*$ is not contained in any query to $\mathcal{CO}$;
- there was no query to $\mathcal{SO}$ with input $(e^*, \cdot, \cdot, \cdot, \mathcal{V}, \cdot)$, where $\mathcal{V} \cap \mathcal{V}^* \neq \emptyset$.

$\mathcal{S}$ calculates all $\{vk^*_\ell, \sigma^*_{T,\ell}, \pi^*_{LTRS,\ell}, \pi^*_{OT,\ell}\}$ according to the Sign algorithm, for all $pk_\ell \in \mathcal{V}^*$ and $\ell \neq \tau^*$.

$\mathcal{S}$ runs $(vk, sk) \leftarrow \mathsf{OTGen}(\lambda)$ and $\tau = H(e^*)$. If $\tau \neq \bar{\tau}$, then $\mathcal{S}$ declares failure and exits. Otherwise, $\mathcal{S}$ uses $A_{\bar{\tau}}$ from the problem instance and the trapdoor β to complete the Sign algorithm similar to the simulation of the signing oracle. Finally, $\mathcal{S}$ returns the whole challenge signature to $\mathcal{A}$.

Output. If $\mathcal{A}$ can correct guess the index $\hat{\pi} = \tau^*$, then $\mathcal{S}$ outputs $A_{\bar{\tau}} = g^{\frac{1}{x+\bar{\tau}}}$. Otherwise, $\mathcal{S}$ outputs $A_{\bar{\tau}} \in \mathbb{G}$.

Analysis. In the challenge signature $\{vk^*_\ell, \sigma^*_{T,\ell}, \pi^*_{LTRS,\ell}, \pi^*_{OT,\ell}\}$, vk^*_ℓ and $\pi^*_{OT,\ell}$ are from the one-time signature and hence do not contain any information about the signer's secret key. $\pi^*_{LTRS,\ell}$ is a NIWI proof such that the commitments are perfectly binding and the proofs are witness indistinguishable. The remaining part of the signature is $\sigma^*_{T,\ell}$. Note that pk_{τ^*} is a random element in the set $\mathcal{V}^*$. If $\mathcal{A}$ correctly guesses $pk_{\tau^*} \in \mathcal{V}^*$ with probability greater than $\frac{d^*}{n^*-(q_c+q_s)}$, then $\mathcal{S}$ can solve the K-DDHI problem.

Note that the probability of not to abort during the simulation is $\frac{1}{K}(1-\frac{1}{K})^{q_s} \approx \frac{1}{K}(1-\frac{q_s}{K})$, where q_s is the number of signing oracle query. The value K should be chosen in a way that K is large enough that $H : \mathcal{EID} \to \mathbb{Z}_K$ is a collision resistant hash function, and K is small enough that the abort probability of $\mathcal{S}$ is small. In addition, if K is smaller, the K-DDHI assumption is weaker. For example, we can set $K = 2q_s$. The probability of not to abort is $\frac{1}{2q_s}$. The size of $\mathcal{EID}$ is restricted such that $H : \mathcal{EID} \to \mathbb{Z}_K$ is collision resistant. □

Appendix B.3. Proof of Theorem 4.3

Proof. This theorem can be proven by a sequence of games.

- Let G_0 be the original individual-linkability game.
- Let G_1 be the same as G_0, except that h is selected from the sub-group $\mathbb{G}_q$ instead of $\mathbb{G}$.
- Let G_2 be the same as G_1, except that $\mathcal{A}$ wins if $\mathcal{A}$ wins in G_1 and $\mathcal{A}$ can output a pair (σ_T, σ'_T) such that $\sigma_T \neq \sigma'_T$ and $(\sigma_T)^q = (\sigma'_T)^q$, where (σ_T, σ'_T) are from the signatures σ_1, σ_2 returned by $\mathcal{A}$ respectively.
- Let G_3 be the same as G_2, except that h is selected from $\mathbb{G}$ instead of the sub-group $\mathbb{G}_q$.
- Let G_4 be the same as G_3, except that h' is selected from the sub-group $\mathbb{G}_p$ instead of $\mathbb{G}$.

If the subgroup decision assumption holds in $\mathbb{G}_q$, the adversary $\mathcal{A}$ cannot distinguish between G_0 and G_1, between G_2 and G_3, and between G_3 and G_4.

Lemma B.1. *The game G_1 and G_2 are indistinguishable if the threshold linkable ring signature scheme is unforgeable.*

Proof. Setup. We first simulate G_1 as follows. The simulator $\mathcal{S}$ runs the bilinear group generator $(N = pq, \mathbb{G}, \mathbb{G}_T, \hat{e}) \leftarrow \mathcal{G}(1^\lambda)$. $\mathcal{S}$ picks a random $\beta \in \mathbb{Z}_N$ and $h \in \mathbb{G}_q$ using p. $\mathcal{S}$ sets $h' = g^\beta$. Finally, $\mathcal{S}$ randomly chooses a collision resistant hash function $H : \mathcal{EID} \to \mathbb{Z}_K$. Then $\mathcal{S}$ gives the public parameters $(N, \mathbb{G}, \mathbb{G}_T, \hat{e}, g, h, h', H)$ to the adversary $\mathcal{A}$.

Oracle Simulation. $\mathcal{S}$ honestly simulates the oracles using the user secret keys.

Output. $\mathcal{A}$ returns $(e, n_1, n_2, d_1, d_2, \mathcal{Y}_1, \mathcal{Y}_2, M_1, M_2, \sigma_1, \sigma_2)$. Denote $\tau = H(e)$ and $\sigma_b = \{\cdot, \sigma_{T,b,\ell}, \pi_{LTRS,b,\ell}, \cdot\}$, where $1 \leq \ell \leq d_b$, $b = 1, 2$. Denote $\pi_{LTRS,b,\ell} = (C_{b,\ell}, \cdot, \cdot, \pi_{2,b,\ell}, \pi_{3,b,\ell}, \cdot, \cdot)$.

Similar to the proof of unforgeability, we can show that $C_{b,\ell}$ is a commitment to some public key $g^{x_i} h^{d_i} \in \mathcal{Y}_b$. $\mathcal{S}$ raises $\hat{e}(g^\tau C_{b,\ell}, \sigma_{T,b,\ell}) = \hat{e}(g, g) \cdot \hat{e}(h, \pi_{2,b,\ell})$ to the q-th power. We have

$$\hat{e}(g^\tau g^{x_i}, \sigma_{T,b,\ell})^q = \hat{e}(g, g)^q.$$

Therefore $(\sigma_{T,b,\ell})^q = (g^{\frac{1}{\tau + x_i}})^q$.

Note that $\mathcal{A}$ is only given $q_c < d_1 + d_2$ secret keys, and $\sigma_{T,b,\ell}$ cannot be the output of $\mathcal{SO}$ using the same event e. If $\mathcal{A}$ wins the game, it means that all tags $\sigma_{T,b,\ell}$ are distinct. By the unforgeability property, there exists at least a pair (σ_T, σ'_T) such that $\sigma_T \neq \sigma'_T$ and $(\sigma_T)^q = (\sigma'_T)^q$. (For example, if $\sigma_T = g^{\frac{1}{\tau + x_i}}$, where σ_T is not the output from $\mathcal{SO}$ and x_i is not the output from $\mathcal{CO}$, it means that σ_T is a forgery.)

Therefore, we can see that if the threshold linkable ring signature scheme is unforgeable, then $\mathcal{A}$ can win in both game G_1 and G_2. Therefore the extra winning condition in G_2 does not affect $\mathcal{A}$'s probability of winning. □

Lemma B.2. *There is no adversary who can win G_4 with non-negligible probability if the subgroup decision assumption holds in $\mathbb{G}_p$.*

Proof. Setup. We simulate G_4 as follows. The simulator $\mathcal{S}$ is given the subgroup decision problem instance $(N, \mathbb{G}, \mathbb{G}_T, \hat{e}, g, h')$. $\mathcal{S}$ picks a random $\beta \in \mathbb{Z}_N$. $\mathcal{S}$ sets $h = g^\beta$. Finally, $\mathcal{S}$ randomly chooses a collision resistant hash function $H : \mathcal{EID} \to \mathbb{Z}_K$. Then $\mathcal{S}$ gives the public parameters $(N, \mathbb{G}, \mathbb{G}_T, \hat{e}, g, h, h', H)$ to the adversary $\mathcal{A}$.

Oracle Simulation. $\mathcal{S}$ honestly simulates the oracles using the user secret keys.

Output. If $\mathcal{A}$ wins the game G_4, it means that there exists at least a pair (σ_T, σ'_T) such that $\sigma_T \neq \sigma'_T$ and $(\sigma_T)^q = (\sigma'_T)^q$. (Although such q is now unknown to $\mathcal{S}$.) Denote g_p and g_q as the generators of $\mathbb{G}_p$ and $\mathbb{G}_q$ respectively. We can write

$$\sigma_T = g_p^a g_q^r, \quad \sigma'_T = g_p^a g_q^{r'},$$

where a, r, r' are some unknown integers in $\mathbb{Z}_N$ and $r \neq r'$.

From the verification, we also have $\hat{e}(\sigma_T, h') = \hat{e}(g, \pi_3)$ and $\hat{e}(\sigma'_T, h') = \hat{e}(g, \pi'_3)$, where π_3 and π'_3 are the corresponding NIWI proof for σ_T and σ'_T respectively. Therefore we have

$$\hat{e}(g_p^a g_q^r, , h') = \hat{e}(g, \pi_3), \quad \hat{e}(g_p^a g_q^{r'}, h') = \hat{e}(g, \pi'_3).$$

If $h' \in \mathbb{G}_p$, then $h' = g_p^\gamma$ for a unknown $\gamma \in \mathbb{Z}_N$. Since $\hat{e}(g_p, g_q) = 1$ and $\hat{e}(g_p, g_p) \neq 1$, we have

$$\hat{e}(g, \pi_3) = \hat{e}(g_p^a g_q^r, g_p^\gamma) = \hat{e}(g_p^a, g_p^\gamma) = \hat{e}(g_p^a g_q^{r'}, g_p^\gamma) = \hat{e}(g, \pi'_3).$$

Therefore $\pi_3 = \pi'_3$.

If $h' \in \mathbb{G}$, then $h' = g_p^\gamma g_q^\delta$ for some unknown $\gamma, \delta \in \mathbb{Z}_N$. Since $\hat{e}(g_p, g_q) = 1$, $\hat{e}(g_p, g_p) \neq 1$ and $\hat{e}(g_q, g_q) \neq 1$, we have

$$\hat{e}(g, \pi_3) = \hat{e}(g_p^a g_q^r, g_p^\gamma g_q^\delta) = \hat{e}(g_p^a, g_p^\gamma) \cdot \hat{e}(g_q^r, g_q^\delta),$$
$$\hat{e}(g, \pi'_3) = \hat{e}(g_p^a g_q^{r'}, g_p^\gamma g_q^\delta) = \hat{e}(g_p^a, g_p^\gamma) \cdot \hat{e}(g_q^{r'}, g_q^\delta).$$

Since $r \neq r'$, $\pi_3 \neq \pi'_3$.

If $\mathcal{A}$ wins the game G_4, then $\mathcal{S}$ compares π_3 with π'_3. If $\pi_3 = \pi'_3$ for some signatures in $\mathcal{A}$'s output σ_1 and σ_2, then $\mathcal{S}$ outputs $h' \in \mathbb{G}_q$; if $\pi_3 \neq \pi'_3$ for all signatures in $\mathcal{A}$'s output σ_1 and σ_2, then $\mathcal{S}$ outputs $h' \in \mathbb{G}$.

Therefore if the subgroup decision assumption holds in $\mathbb{G}_p$, there is no adversary who can win G_4 with non-negligible probability. □

Following the sequence of games, the theorem is proven. □

Appendix B.4. Proof of Theorem 4.4

Proof. Let G_0 be the original unforgeability game. Let G_1 be the same as G_0, except that h is selected from the sub-group $\mathbb{G}_q$ instead of $\mathbb{G}$. If the subgroup decision assumption holds, the adversary $\mathcal{A}$ cannot distinguish G_0 and G_1.

Setup. Now we simulate G_1 as follows. The simulator $\mathcal{S}$ runs the bilinear group generator $(N = pq, \mathbb{G}, \mathbb{G}_T, \hat{e}) \leftarrow \mathcal{G}(1^\lambda)$. $\mathcal{S}$ picks a random $\beta \in \mathbb{Z}_N$ and $h \in \mathbb{G}_q$ using p. $\mathcal{S}$ sets $h' = g^\beta$. Finally, $\mathcal{S}$ randomly chooses a collision resistant hash function $H : \mathcal{EID} \to \mathbb{Z}_K$. Then $\mathcal{S}$ gives the public parameters $(N, \mathbb{G}, \mathbb{G}_T, \hat{e}, g, h, h', H)$ to the adversary $\mathcal{A}$.

Oracle Simulation. $\mathcal{S}$ honestly simulates the oracles using the user secret keys.

Challenge. $\mathcal{A}$ gives $\mathcal{S}$ an event e, group size n, threshold d, a set of n public keys $\mathcal{Y}$, a set of d insiders $\mathcal{V} \subseteq \mathcal{Y}$, a message M. $\mathcal{S}$ uses the corresponding user secret keys to calculate the challenge signature σ'.

Output. $\mathcal{A}$ returns $(n^*, d^*, \mathcal{Y}^*, M^*, \sigma^*)$. Suppose σ^* is linked to σ' by some

$$\{vk^*, \sigma_T, \pi^*_{LTRS}, \sigma^*_{OT}\} \text{ in the signature } \sigma^*,$$
$$\{vk', \sigma_T, \pi'_{LTRS}, \sigma'_{OT}\} \text{ in the signature } \sigma'.$$

Denote $\pi^*_{LTRS} = (C, L, \pi_1, \pi_2, \pi_3, \{C_i, \pi_i^C, B_i, \pi_i^B, D_i, \pi_i^D\}_{1 \leq i \leq \nu}, \pi_C)$. Similar to the proof of unforgeability, $\mathcal{S}$ can use q to show that C is committed to some $pk = g^{x^*} h^d$. After that, $\mathcal{S}$ raises $\hat{e}(g^\tau C, \sigma_T) = \hat{e}(g, g) \cdot \hat{e}(h, \pi_2)$ to the q-th power. We have

$$\hat{e}(g^\tau g^{x^*}, \sigma_T)^q = \hat{e}(g, g)^q.$$

Therefore $(\sigma_T{}^{\frac{1}{\tau + x^*}})^q = g^q$. Note that $\mathcal{S}$ honestly generate π'_{LTRS}. In particular, $\sigma_T{}^{\frac{1}{\tau + x'}} = g$. Therefore

$$(g^{\frac{\tau + x'}{\tau + x^*}})^q = g^q.$$

It implies that $x' = x^*$.

Note that π^*_{LTRS} is a NIWI proof of witness x^* if the SDH assumption holds (the proof is the same as that in unforgeability). Since x^* is not outputted by the $\mathcal{CO}$, $\mathcal{A}$ can only win by setting $\pi^*_{LTRS} = \pi'_{LTRS}$. Since vk^* is involved in the NIWI proof, it forces $vk^* = vk'$. If the one-time signature scheme is unforgeable, $\mathcal{A}$ cannot output a $\sigma^*_{OT} \neq \sigma'_{OT}$.

Observe that $d^*, M^*, \{vk_1^*, \ldots, vk_{d^*}^*\}, \mathcal{Y}^*$ (which implies n^*) are included as part of the message of the one-time signature. Therefore, it implies that $d = d^*$, $M = M^*$, $vk_i = vk_i^*$ for $1 \leq i \leq d^*$ and $\mathcal{Y} = \mathcal{Y}^*$. It means all the one-time verification keys are the same. If $\sigma^* \neq \sigma'$, then it means that $\mathcal{A}$ has forged a one-time signature for some verification key vk_i. □