Separable Linkable Threshold Ring Signatures*

Patrick P. Tsang[1], Victor K. Wei[1], Tony K. Chan[1], Man Ho Au[1], Joseph K. Liu[1], and Duncan S. Wong[2]

[1] Department of Information Engineering
The Chinese University of Hong Kong
Shatin, Hong Kong
{pktsang3,kwwei,klchan3,mhau3,ksliu9}@ie.cuhk.edu.hk
[2] Department of Computer Science
The City University of Hong Kong
Hong Kong
duncan@cityu.edu.hk

Abstract. A ring signature scheme is a group signature scheme with no group manager to setup a group or revoke a signer. A linkable ring signature, introduced by Liu, et al. [20], additionally allows anyone to determine if two ring signatures are signed by the same group member (a.k.a. they are *linked*). In this paper, we present the first separable linkable ring signature scheme, which also supports an efficient thresholding option. We also present the security model and reduce the security of our scheme to well-known hardness assumptions. In particular, we introduce the security notions of *accusatory linkability* and *non-slanderability* to linkable ring signatures. Our scheme supports "event-oriented" linking. Applications to such linking criterion is discussed.

1 Introduction

Ring Signatures. A ring signature scheme [22] is a group signature scheme [10, 2] with no group manager to setup a group or revoke a signer's identity. Formation of a group is *spontaneous* in a way that diversion group members can be totally unaware of being conscripted to the group. It allows members to *anonymously* sign messages on behalf of their group. Applications include leaking secrets [22] and anonymous identification/authentication for ad hoc groups [6, 13].

Threshold Ring Signatures. Threshold cryptography [12] allows n parties to share the ability to perform a cryptographic operation (e.g., creating a digital signature). Any d parties can perform the operation jointly, whereas it is infeasible for at most $d-1$ to do so. In a (d, n)-threshold ring signature scheme, the generation of a ring signature for a group of n members requires the involvement of at least d members/signers, and yet the signature reveals nothing about the identities of the signers. Schemes in the literature include [6, 19, 24].

* An extended abstract was in Indocrypt'04. This version updates the security model and results concerning anonymity and non-slanderability.

Linkable Ring Signatures. The notion of linkable ring signatures was introduced by Liu, et al. [20]. They are ring signatures, but with added linkability: such signatures allow anyone to determine if two signatures are signed by the same group member (in which case the two signatures are said to be "*linked*"). If a user signs only once on behalf of a group, the user still enjoys anonymity similar to that in conventional ring signature schemes. If the user signs multiple times, anyone can tell that these signatures have been generated by the same group member. Applications include leaking sequences of secrets and e-voting [20].

Linkable Threshold Ring Signatures. In [20], a (d, n)-threshold extension to its original linkable ring signature scheme is constructed by concatenating d linkable ring signatures. We note that the construction, though simple and trivial, is not efficient. In particular, the space and time complexities are both $O(dn)$. We give in this paper a construction with time and space complexities both being $O(n)$.

Separability. In [8], Camenisch, et. al. diversified the concept of separability of cryptographic protocols into *perfect separability*, *strong separability* and *weak separability* when describing the users' ability to choose their own cryptographic primitive and system parameters. Separability is of particular importance for ring signature schemes as there is no group manager to coordinate the choice of signature primitive and system parameters for each user. For instance, a ring signature scheme that is only weak separable is not practical at all as it is unlikely to have all group members using the same primitive, system parameters and security parameters. The RSA implementation of [22, 1, 19, 24, 20] are strongly separable while the DL implementation of [1, 19, 20] are only weakly separable.

Event-Oriented Linkability. In [20], one can tell if two ring signatures are linked or not if and only if they are signed on behalf of the same group of members. We call this "*group-oriented*" linkability. We present a new linking criterion that we call "*event-oriented*" linkability in which one can tell if two signatures are linked if and only if they are signed for the same event, despite the fact that they may be signed on behalf of different groups. Event-oriented linkable ring signatures are comparatively more flexible in application. E.g., group settings keep changing frequently in ad-hoc group and most of the ring signatures are signed on behalf of different groups, thus render group-oriented linkability virtually useless. Consider another scenario: The CEOs of a company vote for business decisions. Using linkable ring signatures, they can vote anonymously by ring-signing their votes. However, as the group is fixed throughout the polls, votes among polls can be linked by anybody and information can be derived which means anonymity is in jeopardy. This can be prevented when an event-oriented scheme is used.

1.1 Contributions

Our main contributions include:

- We give the first separable linkable ring signature. It also the first linkable ring signature of the CDS-type ([11]).
- We present a security model for linkable threshold ring signature, and reduce the security of our scheme to well-known hard problem assumptions.

- Our scheme supports bandwidth-efficient threshold signing. The signature size in [20] is $O(dn)$ while ours is $O(n)$, where n is the number of users and d is the threshold. However, our scheme is *interactive*: insiders interact collaboratively to generate the signature.
- We introduce new security notions to linkable ring signatures: (1) *Non-accusatory linkability* only detects the presence of two "linked" signatures, while *accusatory linkability* additionally outputs the identity of the suspected "double-signer". (2) Strong *non-slanderability* means no coalition can generate signatures accusatorily linked to a targeted victim.
- We present a new linking criterion that is "*event-oriented*". Under such linkability, one can tell if two signatures are linked if and only if they are signed for the same event, despite the fact that they may be signed on behalf of different groups.

1.2 Organization

The paper is organized as follows: In Sec. 2, we give some preliminaries. In Sec. 3, we describe the building blocks used in our construction. Then we define our separable linkable threshold signatures in Sec. 4. A construction and its security analysis are presented in Sec. 5. We conclude in Sec. 6.

2 Preliminaries

2.1 Notations and Mathematical Assumptions

Definition 1. *A function $f(\lambda)$ is negligible if for all polynomials $p(\lambda)$, $f(\lambda) < 1/p(\lambda)$ holds for all sufficiently large λ. A function is non-negligible if it is not negligible.*

Definition 2 (Strong RSA Assumption [7, 15, 16]). *Given a safe prime product N, and $z \in QR(N)$, it is infeasible to find $u \in \mathbb{Z}_N^*$ and $e > 1$ such that $u^e = z (\bmod N)$, in time polynomial in the size of N.*

Definition 3 (Decisional Diffie-Hellman (DDH) over QR(N) Assumption). *Given a generator g of a cyclic group $QR(N)$, where N is a composite of two primes, the distribution ensembles (g^x, g^y, g^z) and (g^x, g^y, g^{xy}), where $x, y, z \in_R [1, ord(g)]$, are computationally indistinguishable by all PPT algorithm in time polynomial in the size of N.*

2.2 Honest-Verifier Zero-Knowledge (HVZK) Proof of Knowledge Protocols (PoKs)

Every HVZK proof can be turned into a signature scheme by setting the challenge to the hash value of the commitment together with the message to be signed [14]. Such a scheme is proven secure by [21] against existential forgery under adaptively chosen message attack [17] in the random oracle model [4]. Following [9], we call these signature schemes "signatures based on proofs of knowledge", SPK for short. Note that there always exists a corresponding HVZK PoK protocol for every SPK.

3 Basic Building Blocks

In this section, we describe some three-move interactive HVZK PoK protocols that we will use as basic building blocks for our event-oriented linkable threshold ring signature scheme. These protocols all work in finite cyclic groups of quadratic residues modulo safe prime products. For each $i = 1, \ldots, n$, let N_i be a safe-prime product and define the group $G_i \doteq QR(N_i)$ such that its order is of length $\ell_i - 2$ for some $\ell_i \in \mathbb{N}$. Also let g_i, h_i be generators of G_i such that their relative discrete logarithms are not known.

Let $1 < \epsilon \in \mathbb{R}$ be a parameter and let $\mathcal{H} : \{0,1\}^* \to \mathbb{Z}_q$ be a strong collision-resistant hash function, where q is a κ-bit prime for some security parameter $\kappa \in \mathbb{N}$. Define $\mathcal{N} \doteq \{1, \ldots, n\}$ and $\Gamma_i \doteq \{-2^{\ell_i} q, \ldots, (2^{\ell_i} q)^\epsilon\}$.

3.1 Proving the Knowledge of Several Discrete Logarithms

This protocol is a straightforward generalization of the protocol for proving the knowledge of a discrete logarithm over groups of unknown order in [7]. This allows a prover to prove to a verifier the knowledge of n discrete logarithms $x_1, \ldots, x_n \in \mathbb{Z}$ of elements $y_1, \ldots, y_n$ respectively and to the bases $g_1, \ldots, g_n$ respectively. Using the notation in [9], the protocol is denoted by:

$$PK\{(\alpha_1, \ldots, \alpha_n) : \bigwedge_{i=1}^{n} y_i = g_i^{\alpha_i}\}.$$

A prover $\mathcal{P}$ knowing $x_1, \ldots, x_n \in \mathbb{Z}$ such that $y_i = g_i^{x_i}$ for all $i = 1, \ldots, n$ can prove to a verifier $\mathcal{V}$ his/her knowledge as follows.

- (Commit.) $\mathcal{P}$ chooses $r_i \in_R \mathbb{Z}_{(2^{\ell_i} q)^\epsilon}$ and computes $t_i \leftarrow g_i^{r_i}$ for all $i = 1, \ldots, n$. $\mathcal{P}$ sends $(t_1, \ldots, t_n)$ to $\mathcal{V}$.
- (Challenge.) $\mathcal{V}$ chooses $c \in_R \mathbb{Z}_q$ and sends it to $\mathcal{P}$.
- (Response.) $\mathcal{P}$ computes, for all $i = 1, \ldots, n$, $s_i \leftarrow r_i - cx_i$ (in $\mathbb{Z}$). $\mathcal{P}$ sends $(s_1, \ldots, s_n)$ to $\mathcal{V}$.

$\mathcal{P}$ verifies by checking, for all $i = 1, \ldots, n$, if $t_i \stackrel{?}{=} g_i^{s_i} y_i^c$.

Theorem 1. *If the Strong RSA assumption holds, the protocol is an HVZK PoK protocol.*

Proof. We omit the proof as it is a straightforward extension of the proof of Lemma 1 in [7]. □

As noted before, the protocol can be turned into a signature scheme by replacing the challenge by the hash of the commitment together with the message M to be signed: $c \leftarrow \mathcal{H}((g_1, y_1)||\ldots||(g_n, y_n)||t_1||\ldots||t_n||M)$. In this case, the signature is $(c, s_1, \ldots, s_n)$ and the verification becomes:

$$c \stackrel{?}{=} \mathcal{H}((g_1, y_1)||\ldots||(g_n, y_n)||g_1^{s_1} y_1^c||\ldots||g_n^{s_n} y_n^c||M).$$

Following [9], we denote this signature scheme by:

$$SPK\{(\alpha_1, \ldots, \alpha_n) : \bigwedge_{i=1}^{n} y_i = g_i^{\alpha_i}\}(M).$$

3.2 Proving the Knowledge of d Out of n Equalities of Discrete Logarithms

This protocol is constructed using the techniques described in [11], by combining the PoK for discrete logarithm in [7] and the secret sharing scheme due to Shamir [23]. This allows a prover to prove to a verifier his/her knowledge of some d out of n integers $x_1, \ldots, x_n$, where $x_i = \log_{g_i} y_i = \log_{h_i} v_i$ for all $i = 1, \ldots, n$. The protocol is denoted by:

$$PK\left\{(\alpha_1, \ldots, \alpha_n) : \bigvee_{\mathcal{J} \subseteq \mathcal{N}, |\mathcal{J}|=d} \left(\bigwedge_{i \in \mathcal{J}} y_i = g_i^{\alpha_i} \wedge v_i = h_i^{\alpha_i} \right)\right\}.$$

A prover $\mathcal{P}$ knowing, for all $i \in \mathcal{I}$, $x_i \in \mathbb{Z}$ such that $y_i = g_i^{x_i}$ and $v_i = h_i^{x_i}$, where $\mathcal{I}$ is some subset of $\mathcal{N}$ such that $|\mathcal{I}| = d$, can prove his/her knowledge to a verifier $\mathcal{P}$ as follows.

– (Commit.) $\mathcal{P}$ does the following: For $i \in \mathcal{N} \backslash \mathcal{I}$, select $c_i \xleftarrow{R} \mathbb{Z}_q$. For all $i \in \mathcal{N}$, select $r_i \xleftarrow{R} \mathbb{Z}_{(2^{\ell_i} q)^{\epsilon}}$. Compute

$$t_i \leftarrow \begin{cases} g_i^{r_i}, & i \in \mathcal{I}; \\ g_i^{r_i} y_i^{c_i}, & i \in \mathcal{N} \backslash \mathcal{I}, \end{cases} \quad \text{and } T_i \leftarrow \begin{cases} h_i^{r_i}, & i \in \mathcal{I}; \\ h_i^{r_i} v_i^{c_i}, & i \in \mathcal{N} \backslash \mathcal{I}. \end{cases}$$

  $\mathcal{P}$ sends $(t_1, \ldots, t_n, T_1, \ldots, T_n)$ to $\mathcal{V}$.
– (Challenge.) $\mathcal{V}$ chooses $c \in_R \mathbb{Z}_q$ and sends it to $\mathcal{P}$.
– (Response.) $\mathcal{P}$ does the following: Compute a polynomial f of degree $\leq n-d$ over $\mathbb{Z}_q$ such that $f(0) = c$ and $f(i) = c_i$ for all $i \in \mathcal{N} \backslash \mathcal{I}$. Compute $c_i \leftarrow f(i)$ for all $i \in \mathcal{I}$. Set

$$s_i \leftarrow \begin{cases} r_i - c_i x_i, & i \in \mathcal{I}; \\ r_i, & i \in \mathcal{N} \backslash \mathcal{I}. \end{cases}$$

  $\mathcal{P}$ sends $(f, s_1, \ldots, s_n)$ to $\mathcal{V}$.

$\mathcal{P}$ verifies by checking if (1) f is a polynomial of degree $\leq n - d$ over $\mathbb{Z}_q$, (2) $f(0) \stackrel{?}{=} c$, and (3) $t_i \stackrel{?}{=} y_i^{f(i)} g_i^{s_i}$ and $T_i \stackrel{?}{=} v_i^{f(i)} h_i^{s_i}$, for all $i = 1, \ldots, n$.

Theorem 2. *If the Strong RSA assumption holds, the protocol is an HVZK PoK protocol.*

(*Proof Sketch*) To prove the theorem, it suffices to show that the protocol is correct, sound and statistical HVZK.

– (Correctness.) Straightforward.

– (Soundness.) It suffices to show how a witness can be extracted if given two valid protocol conversations with the same commitment but different challenges. Denoting the two conversation transcripts by $\langle\ (t_1,\ldots,t_n,T_1,\ldots,T_n),$ $(c), (f, s_1,\ldots,s_n)\ \rangle$ and $\langle (t_1,\ldots,t_n,T_1,\ldots,T_n),(c'),(f',s'_1,\ldots,s'_n)\rangle$, we have $c \neq c'$ and thus $f(0) \neq f'(0)$. As the degrees of f and f' are at most $n-d$, there are at least d distinct values $\pi_1,\ldots,\pi_d \in \{1,\ldots,n\}$ such that $f(\pi_i) \neq f'(\pi_i)$ for all $i = 1,\ldots,d$. Using arguments in [7], $f(\pi) - f'(\pi)$ divides $s'_\pi - s_\pi$ and therefore an integer $\hat{x}$ such that $y_\pi = g_\pi^{\hat{x}\pi}$ and $v_\pi = h_\pi^{\hat{x}\pi}$ can be computed as: $\hat{x}_\pi \leftarrow (s_\pi - s'_\pi)/(f'(\pi) - f(\pi))$.
Hence a witness $(\hat{x}_{\pi_1},\ldots,\hat{x}_{\pi_d})$ can be computed from two such transcripts.
– (Statistical HVZK.) To simulate a transcript, a simulator $\mathcal{S}$ first chooses uniformly at random a polynomial f' of degree $n-d$ over $\mathbb{Z}_q$. For all $i = 1,\ldots,n$, $\mathcal{S}$ picks uniformly at random $s'_i \in_R \mathbb{Z}_{(2^{\ell_i}q)^\epsilon}$ and computes $t'_i \leftarrow g_i^{s'_i} y_i^{f'(i)}$. The simulated transcript is: $\langle (t'_1,\ldots,t'_n,T'_1,\ldots,T'_n),(f'(0)),(f',s'_1,\ldots,s'_n)\rangle$.
To prove that the simulation is statistical indistinguishable from real protocol conservations, one should consider, for each $i = 1,\ldots,n$, the probability distribution $P_{S_i}(s_i)$ of the responses of the prover and the probability distribution $P_{S'_i}(s'_i)$ according to which $\mathcal{S}$ chooses s'_i. The statistical distance between the two distributions can be computed to be at most: $2(2^{\ell_i})(q-1)/(2^{\ell_i}q)^\epsilon$ $\leq 2/(2^{\ell_i}q)^{\epsilon-1}$. The result follows.

□

The protocol can be turned into a signature scheme by replacing the challenge by the hash of the commitment together with the message M to be signed:

$$c \leftarrow \mathcal{H}((g_1,y_1,h_1,v_1)||\ldots||(g_n,y_n,h_n,y_n)||t_1||\ldots||t_n||T_1||\ldots||T_n||M).$$

In this case, the signature is $(f, s_1,\ldots,s_n)$ and step (3) of the verification becomes:

$$\begin{aligned} c \stackrel{?}{=} \mathcal{H}(\ &(g_1,y_1,h_1,v_1)||\ldots||(g_n,y_n,h_n,y_n)|| \\ &y_1^{c_1}g_1^{s_1}||\ldots||y_n^{c_n}g_n^{s_n}||v_1^{c_1}h_1^{s_1}||\ldots||v_n^{c_n}h_n^{s_n}||M). \end{aligned}$$

We denote this signature scheme by:

$$SPK\left\{(\alpha_1,\ldots,\alpha_n) : \bigvee_{\mathcal{J}\subseteq\mathcal{N},|\mathcal{J}|=d}\left(\bigwedge_{i\in\mathcal{J}} y_i = g_i^{\alpha_i} \wedge v_i = h_i^{\alpha_i}\right)\right\}(M).$$

4 Security Model

We give our security model and define relevant security notions.

4.1 Syntax

A *linkable threshold ring signature*, (LTRS) scheme, is a tuple of five algorithms (Key-Gen, Init, Sign, Verify and Link).

- $(sk_i, pk_i) \leftarrow \mathsf{Key\text{-}Gen}(1^{\lambda_i})$ is a PPT algorithm which, on input a security parameter $\lambda_i \in \mathbb{N}$, 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 $\mathsf{Key\text{-}Gen}$.
- $\mathsf{param} \leftarrow \mathsf{Init}(\lambda)$ is a PPT algorithm which, on input a security parameter λ, outputs the set of security parameters param which includes λ.
- $\sigma'=(e,n,d,\mathcal{Y},\sigma)\leftarrow \mathsf{Sign}(e, n, d, \mathcal{Y}, \mathcal{X}, M)$ 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'.
- $1/0 \leftarrow \mathsf{Verify}(M, \sigma')$ is an algorithm which, on input a message-signature pair (M,σ') returns 1 or 0 for accept or reject, resp. If accept, the message-signature pair is *valid*.
- $1/0 \leftarrow \mathsf{Link}$ (σ'_1, σ'_2) is an algorithm which, upon input two valid signature pairs, outputs 0 or 1 for linked or $\mathsf{unlinked}$. In case of linked it additionally outputs the public key pk^* of the suspected "double-signer".

Remark: Our linkability is *accusatory* meaning it outputs the public key of the suspected "double signer". The linkability in [20] is not accusatory – it only outputs linked or $\mathsf{unlinked}$ without suspect identity.

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 a common signer. In the case of linked, the suspect output by Link is exactly the common signer.

4.2 Notions of Security

Security of LTRS schemes has three aspects: unforgeability, anonymity and linkability. 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 $sk_i \in \mathcal{SK}$.
- $\sigma' \leftarrow \mathcal{SO}(e, n, d, \mathcal{Y}, \mathcal{V}, \mathcal{X}, M)$. 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 subset $\mathcal{V}$ of $\mathcal{Y}$ with $|\mathcal{V}| = d$, a set of secret keys $\mathcal{X}$ whose corresponding public keys are all contained in $\mathcal{V}$, and a message M, returns a valid signature σ'.

Remark: An alternative approach to specify the $\mathcal{SO}$ is to exclude the signer set $\mathcal{V}$ from the input and have $\mathcal{SO}$ select it according to suitable random distribution. We do not pursue that alternative further.

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 $\mathcal{SO}$:

1. $\mathcal{S}$ generates and gives $\mathcal{A}$ the system parameters param.
2. $\mathcal{A}$ may query the oracles according to any adaptive strategy.
3. $\mathcal{A}$ 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) $\mathsf{Verify}(M,\sigma')$=1, (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 $\mathcal{CO}$, and (4) σ is not a query output of $\mathcal{SO}$ on any input containing M. We denote by $\mathbf{Adv}_{\mathcal{A}}^{unf}(\lambda)$ the probability of $\mathcal{A}$ winning the game.

Definition 4 (unforgeability). *An LTRS scheme is unforgeable if for all PPT adversary $\mathcal{A}$, $\mathbf{Adv}_{\mathcal{A}}^{unf}(\lambda)$ is negligible.*

Linkable Anonymity. Anonymity for LTRS schemes is defined in the following game:

Game LA

1. (*Initialization Phase*) $\mathcal{S}$ generates and gives $\mathcal{A}$ the system parameters param.
2. (*Probe-1 Phase*) $\mathcal{A}$ may query the oracles according to any adaptive strategy.
3. (*Gauntlet Phase*) $\mathcal{A}$ gives $\mathcal{S}$ event-id e_g, group size n_g, threshold $d_g \in \{1, \ldots, n_g\}$, message M_g, a set $\mathcal{Y}_g$ of n public keys all of which are query outputs of $\mathcal{JO}$, a subset $\mathcal{V}_g$ of $\mathcal{Y}_g$ with $|\mathcal{V}_g| = d_g$, a set of secret keys $\mathcal{X}_g$ with $|\mathcal{X}_g| = d_g - 1$ and whose corresponding secret keys are all contained in $\mathcal{V}_g$. The lone public key $y_g \in \mathcal{V}_g$ whose corresponding secret key is not contained in $\mathcal{X}_g$ has never been queried to $\mathcal{CO}$ and has been included in the insider set $\mathcal{V}$ in any query to Signing Oracle $\mathcal{SO}$.
   Then $\mathcal{S}$ flips a fair coin to select $b \in \{\mathsf{real}, \mathsf{ideal}\}$. Case b=real: $\mathcal{S}$ queries $\mathcal{CO}$ with y_g to obtain its corresponding secret key x_g, and computes $\sigma'_g = \mathsf{Sign}$ (e_g, n_g, d_g, $\mathcal{Y}_g$, $\mathcal{X}_g \cup \{x_g\}$, M_g), Case b=ideal: $\mathcal{S}$ computes $\sigma'_g = \mathcal{SO}$ (e_g, n_g, d_g, $\mathcal{Y}_g$, $\mathcal{V}_g$, $\mathcal{X}_g$, M_g).
   $\mathcal{S}$ sends σ'_g to $\mathcal{A}$.
4. (*Probe-2 Phase*) $\mathcal{A}$ queries the oracles adaptively, except that y_g cannot be queried to $\mathcal{CO}$ or included in the insider set $\mathcal{V}$ of any query to $\mathcal{SO}$.
5. (*End Game*) $\mathcal{A}$ delivers an estimate $\hat{b} \in \{\mathsf{real}, \mathsf{ideal}\}$ of b.

$\mathcal{A}$ wins the game if $\hat{b} = b$. Define the *advantage* of $\mathcal{A}$ as

$$\mathbf{Adv}_{\mathcal{A}}^{Anon}(\lambda) = \mathsf{Pr}[\mathcal{A} \text{ wins}] - 1/2.$$

Definition 5 (Linkable-anonymity). *An LTRS scheme is linkably-anonymous if for any PPT adversary $\mathcal{A}$, $\mathbf{Adv}_{\mathcal{A}}^{Anon}(\lambda)$ is negligible.*

Remark: Linkable anonymity is a form of computational zero-knowledge: the attacker cannot computationally distinguish the real world from the ideal world. Note that the anonymity notions in [3, 5, 18] appear to be also computational zero-knowledge. Our attacker model is not a fully active attacker: queries relevant to the gauntlet public key, y_g, are ruled out. The anonymity in [20] is also with respect to the above model. We note that [3], p.623, argued that anonymity and linkability cannot coexist in their security model.

Linkability. Linkability 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.
3. $\mathcal{A}$ gives $\mathcal{S}$ an event-id $e \in \mathcal{EID}$, group sizes $n_1, n_2 \in \mathbb{N}$, 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) Verify(M_i,σ'_i)=1 for $i = 1, 2$, (3) $\mathcal{CO}$ has been queried at most $(d_1 + d_2 - 1)$ times, and (4) Link(σ'_1, σ'_2)=0. We denote by $\mathbf{Adv}_{\mathcal{A}}^{Link}$ the probability of $\mathcal{A}$ winning the game.

Definition 6 (Linkability). *An LTRS scheme is linkable if for all PPT adversary $\mathcal{A}$, $\mathbf{Adv}_{\mathcal{A}}^{Link}$ is negligible.*

Non-Slanderability. 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 $\mathcal{SO}$:

Game NS

1. (*Initialization Phase*) $\mathcal{S}$ generates and gives $\mathcal{A}$ the system parameters param.
2. (*Probe-1 Phase*) $\mathcal{A}$ may query the oracles according to any adaptive strategy.
3. (*Gauntlet Phase*) $\mathcal{A}$ gives $\mathcal{S}$ an event e, group size n, threshold d, a set of n public keys $\mathcal{Y}_g$, a set of d insiders $\mathcal{V}_g \subseteq \mathcal{Y}_g$, a message M. No member of $\mathcal{V}_g$ has been queried to $\mathcal{CO}$ or has been included in the insider set of any query to $\mathcal{SO}$. $\mathcal{S}$ queries all members of $\mathcal{V}_g$ to $\mathcal{CO}$ to obtain the corresponding secret keys $\mathcal{X}_g$, and invoke Sign to produce a signatures $\sigma' = (e, n, d, \mathcal{Y}_g, \sigma)$.
4. (*Probe-2 Phase*) $\mathcal{A}$ queries oracles with arbitrary interleaving. Except no member of $\mathcal{V}_g$ can be queries to $\mathcal{CO}$, or included in the insider set of any query to $\mathcal{SO}$. In parcitular, $\mathcal{A}$ is allowed to query any public keys which is not in $\mathcal{V}_g$ to $\mathcal{CO}$.

5. (*End Game.*) $\mathcal{A}$ delivers a valid signature $\hat{\sigma}$ which is not an $\mathcal{SO}$ query output to $\mathcal{S}$.

$\mathcal{A}$ wins *Game NS* if $\mathsf{Link}(\hat{\sigma},\ \sigma') = 1$. The Adverary $\mathcal{A}$'s *advantage* is his probability of winning.

Definition 7 (Non-Slanderability). *An* LTRS *scheme is* non-slanderable *if no PPT adversary $\mathcal{A}$ has a non-negligible advantage in Game NS.*

Security. Summarizing we have:

Definition 8 (Security of LTRS Schemes). *An* LTRS *scheme is* secure *if it is unforgeable, linkably-anonymous, linkable and non-slanderable.*

5 Our Construction

5.1 An Linkable Threshold Ring Signature Scheme

In this section, we give a concrete construction of an LTRS scheme. We then show that such a construction is secure under the security model defined in the previous section.

- Key-Gen. On input a security parameter ℓ_i, the algorithm randomly picks two distinct primes p_i, q_i of the form $p_i = 2p'_i + 1$ and $q_i = 2q'_i + 1$, where p'_i, q'_i are both $((\ell_i - 2)/2)$-bit primes, and sets $N_i \leftarrow p_i q_i$. It then picks a random generator g_i of $QR(N_i)$ and a random $x_i \in_R \mathbb{Z}_{p'_i q'_i}$ and computes $y_i \leftarrow g_i^{x_i}$. It picks a strong collision-resistant hash function $H_i : \{0,1\}^* \rightarrow \{h | \langle h \rangle = QR(N_i)\}$. It sets the public key to $pk_i \leftarrow (\ell_i, N_i, g_i, y_i, H_i)$, and the secret key to $sk_i \leftarrow (p_i, q_i, x_i)$. Finally it outputs (sk_i, pk_i).
- Init. On input security parameters $\ell \in \mathbb{N}$, $1 < \epsilon \in \mathbb{R}$ and $\kappa \in \mathbb{N}$, the algorithm randomly picks a κ-bit prime q and a strong collision-resistant hash function $H : \{0,1\}^* \rightarrow \mathbb{Z}_q$. It outputs the system parameters $\mathsf{param} = (\ell, \epsilon, \kappa, q, H)$.
- Sign. On input the system parameters $\mathsf{param} = (\ell, \epsilon, \kappa, q, H)$, an event-id $e \in \{0,1\}^*$, a group size $n \in \mathbb{N}$, a threshold $d \in \{1, \ldots, n\}$, a public key set $\mathcal{Y} = \{pk_1, \ldots, pk_n\}$, where each $pk_i = (\ell_i, N_i, g_i, y_i, H_i)$ is s.t. $\ell_i \geq \ell$, a private key set $\mathcal{X} = \{sk_{\pi_1}, \ldots, sk_{\pi_d}\}$, where each $sk_{\pi_i} = (p_{\pi_i}, q_{\pi_i}, x_{\pi_i})$ corresponds to $pk_{\pi_i} \in \mathcal{Y}$, and a message $M \in \{0,1\}^*$, Define $\mathcal{N} = \{1, \ldots, n\}$ and $\mathcal{I} = \{\pi_1, \ldots, \pi_d\} \subseteq \mathcal{N}$, the algorithm does the following:
  1. For all $i \in \mathcal{N}$, compute $h_{i,e} \leftarrow H_i(\mathsf{param}, pk_i, e)$ and the tags

$$\tilde{y}_{i,e} \leftarrow \begin{cases} h_{i,e}^{x_i},\ i \in \mathcal{I}; \\ h_{i,e}^{a_i},\ i \in \mathcal{N} \backslash \mathcal{I},\ a_i \stackrel{R}{\leftarrow} \mathbb{Z}_{\lfloor N_i/4 \rfloor}. \end{cases}$$

2. Compute a signature $(f, s_1, \ldots, s_n)$ for

$$SPK\left\{(\alpha_1, \ldots, \alpha_n) : \bigvee_{\mathcal{J}\subseteq\mathcal{N},|\mathcal{J}|=d} \left(\bigwedge_{i\in\mathcal{J}} y_i = g_i^{\alpha_i} \wedge \tilde{y}_{i,e} = h_{i,e}^{\alpha_i}\right)\right\}(M).$$

   In particular, this requires the knowledge of $x_{\pi_1}, \ldots, x_{\pi_d}$. We will refer to this signature scheme as SPK_1.
3. Compute a signature $(c, s'_1, \ldots, s'_n)$ for

$$SPK\left\{(\beta_1, \ldots, \beta_n) : \bigwedge_{i=1}^{n} \tilde{y}_{i,e} = h_{i,e}^{\beta_i}\right\}(M).$$

   In particular, this requires the knowledge of x_i for all $i \in \mathcal{I}$ and a_i for all $i \in \mathcal{N}\backslash\mathcal{I}$. We will refer to this signature scheme as SPK_2.
4. The signature is

$$\sigma \leftarrow \langle((\tilde{y}_{1,e}, \ldots, \tilde{y}_{n,e}), (f, s_1, \ldots, s_n), (c, s'_1, \ldots, s'_n)\rangle.$$

   Note that a signature is composed of three parts: the tags, a signature for SPK_1 and a signature for SPK_2.

– Verify. On input a tuple $(\mathsf{param}, e, n, d, \mathcal{Y}, M, \sigma)$, the algorithm parses param into $(\ell, \epsilon, \kappa, q, H)$, $\mathcal{Y}$ into $\{pk_1, \ldots, pk_n\}$, where $pk_i = (\ell_i, N_i, g_i, y_i, H_i)$, and σ into $\langle((\tilde{y}_1, \ldots, \tilde{y}_n), (f, s_1, \ldots, s_n), (c, s'_1, \ldots, s'_n)\rangle$. If any $\ell_i < \ell$, the algorithm returns with 0. Otherwise it does the following:
  1. For $i \in \mathcal{N}$, compute $h_{i,e} \leftarrow H_i(\mathsf{param}, pk_i, e)$.
  2. Verify if $(f, s_1, \ldots, s_n)$ is a correct signature for SPK_1.
  3. Verify if $(c, s'_i, \ldots, s'_n)$ is a correct signature for SPK_2.

– Link. On input a tuple $(\mathsf{param}, e, (n_1, d_1, \mathcal{Y}_1, M_1, \sigma_1), (n_2, d_2, \mathcal{Y}_2, M_2, \sigma_2))$ s.t., for $j = 1, 2$, $\mathsf{Verify}(M_j,\sigma'_j)$=1, the algorithm first parses, for $j = 1, 2$, $\mathcal{Y}_j$ into $\mathcal{Y}_j = \{pk_1^{(j)}, \ldots, pk_{n_j}^{(j)}\}$ and σ_j into

$$\langle((\tilde{y}_{1,e}^{(j)}, \ldots, \tilde{y}_{n,e}^{(j)}), (f^{(j)}, s_1^{(j)}, \ldots, s_n^{(j)}), (c^{(j)}, s_1'^{(j)}, \ldots, s_n'^{(j)})\rangle.$$

  If there exists $\pi_1 \in \{1, \ldots, n_1\}$ and $\pi_2 \in \{1, \ldots, n_2\}$ s.t. $pk_{\pi_1}^{(1)} = pk_{\pi_2}^{(2)}$ and $\tilde{y}_{\pi_1,e}^{(1)} = \tilde{y}_{\pi_2,e}^{(2)}$, it returns 1 and additionally $pk_{\pi_1}^{(1)}$. Otherwise it returns 0.

Correctness. Straightforward.

5.2 Security

We state the security theorems here and provide proof sketches.

Theorem 3 (Unforgeability). *Our construction is unforgeable under the Strong RSA assumption in the random oracle model.*

(*Proof Sketch*) Roughly speaking, similarly constructed ring signatures [19] already has unforgeability, and that implies unforgeability with linkable ring signatures. □

Theorem 4 (Linkable-anonymity). *Our construction is anonymous under the Strong RSA assumption and DDH over QR(N) assumption in the random oracle model.*

(*Proof Sketch*) Simulating Signing Oracle, $\mathcal{SO}$: Upon input (e, n, d, $\mathcal{Y}$, $\mathcal{V}$, $\mathcal{X}$, M), generate a valid signature as follows: For each $i \in \mathcal{Y} \backslash \mathcal{V}$, randomly generate a_i and compute $\tilde{y}_{i,e} = h_{i,e}^{a_i}$. For each $i \in \mathcal{V}$, randomly generate a_i and backpatch the random oracle to $h_{i,e} = H_i(\mathsf{param}, pk_i, e) = g_i^{a_i}$ and compute $\tilde{y}_{i,e} = y^{a_i}$. Ensure consistency with other oracles from the beginning. Generate $c_0, \cdots, c_n$ such that they interpolate a polynomial f with degree $\leq n-d$ and $f(i) = c_i$ for $0 \leq i \leq n$. For each i, simulate the corresponding 3-move conversation in Step (2) of Sign with randomly generated responses $s_1, \cdots, s_n$ to produce the commitments. Backpatch the random oracle so that the commitments are hashed to c_0. This completes up to Step (2) in Sign. The rest is easy: Randomly generate challenge c, simulate the SPK in Step (3) of Sign with randomly generate responses s_1', $\cdots$, s_n'.

Setting up the gauntlet for solving DDH: Similar to proof of anonymity in [20]. Let Q_J be the number of $\mathcal{JO}$ queries. Denote the Gauntlet DDH Problem as ($\hat{N}$, $\hat{g}$, $\hat{g}^\alpha$, $\hat{g}^\beta$, $\hat{g}^\gamma$) where $\gamma = \alpha\beta$ with probability 1/2. In the Gauntlet Phase, Simulator $\mathcal{S}$ sets up the witness extraction mechanism as follows: Randomly select $i^* \in \{1, \cdots, Q_J\}$. Return $pk^* \leftarrow (\hat{l}, \hat{N}, \hat{g}, \hat{g}^\alpha, \hat{H})$ in the i^*-th $\mathcal{JO}$ query, backpatch Random Oracle $\mathcal{HO}_{i^*}$ to $h_{i,e} = \hat{g}^\beta$. There is a non-negligible probability that $pk^* = y_g$, the gauntlet public key. Generate the Gauntlet signature σ_g' with $\tilde{y}_{i,e} = \hat{g}^\gamma$ and simulate the SPK's. With 1/2 probability, $\alpha\beta = \gamma$ and it can be shown that the gauntlet signature is indistinguishable from one generated using Sign. Otherwise, with 1/2 probability, $\alpha\beta \neq \gamma$ and it can be shown that σ_g' is indistinguishable from one generated using $\mathcal{SO}$.

If $\mathcal{A}$ returns $\hat{b} = 1$, $\mathcal{S}$ answers Yes to the DDH question. Otherwise, $\mathcal{S}$ answers No. $\mathcal{S}$'s advantage in DDH equals $\mathcal{A}$'s advantage in winning Game LA. □

Theorem 5 (Linkability). *Our construction is linkable under the Strong RSA assumption in the random oracle model.*

(*Proof Sketch*) Similar to proof of linkability in [20]. If Adversary can produce two unlinked signatures, then he is rewound twice to produce two sets of witnesses of set-size d_1 and d_2 respectively. If the two sets overlap, then the threshold signatures should have already been linked. If the two sets do not overlap, then we would have obtained a total of $d_1 + d_2$ witnesses while Adversary only corrupted at most $d_1 + d_2 - 1$ witnesses. □

Theorem 6 (Non-Slanderability). *Our construction is non-slanderable under the Strong RSA assumption in the random oracle model.*

(*Proof Sketch*) The non-slanderability is protected by Step (3) of the signature. Given a signature from $\mathcal{SO}$, Adversary does not know the discrete logarithm of any $\tilde{y}_i$, and therefore cannot produce a signature containing some $\tilde{y}_j$ and prove knowledge of logarithm of $\tilde{y}_j$ as in Sign's Step (3). □

Summarizing, we have

Theorem 7 (Security). *Our construction is a secure LTRS scheme.*

Note the linkable ring signature in [20] is also secure in our security model.

5.3 Discussions

Separable Linkable Ring Signatures. We achieved separable linkable ring signatures where individual users choose their own safe RSA modulus N_i. In our construction, individual user's key pair are constrained to reside in Discret Logarithm (DL) over a composite moduli. In fact, our method can be easily modified to allow user key pairs from DL over a prime modulus, i.e. (sk, pk)= (x, $y = g^x \pmod{P_i}$). Therefore, our signatures can be easily modified to support a mixture of composite DL and prime DL.

RST-type ring signature. Although our construction utilizes the CDS-type structure, meaning the structure from Cramer, et al. [11], the technique can be easily adapted to construct the first separable linkable ring signature of the RST-type, meaning the structure from Rivest, et al. [22]. Simply follow [20] but use different $\tilde{y}_i$ for different users i instead using a single $\tilde{y}$. $\tilde{y}_i = h^{a_i}$ with randomly generated a_i except $\tilde{y}_s = h^{x_s}$ with signer s. Then simulate the Proof-of-Knowledge $\{(x_i) : y_i = g^{x_i} \wedge \tilde{y}_i = h^{x_i}\}$ along the *ring*, computing $Hash(commitments_i) = challenge_{i+1}$ and simulating, except for the actual signer. The resulting linkable ring signature is separable, supporting a mixture of composite DL and prime DL key pairs.

Bandwidth Efficiency. The length of our signature is $O(n)$ (n being the group size). This improves upon [20] whose length if $O(nd)$. However, our scheme is not non-interactive while [20] is.

Event-IDs. Event-ids should be chosen carefully to according specific applications. We give two examples here. (1) When an event-oriented linkable (threshold) ring signature scheme is used to leak sequences of secrets, the whistle-blower should choose a unique event-id when leaking the first secret and stick to using the same in the sequel. This makes sure that the sequence of secrets cannot be linked to other sequences. (2) When used in electronic voting, it is usually the voting organizer (e.g. the government) who decides on an event-id. Each eligible voter should therefore, before they cast a vote, make sure that the event-id has not been used in any previous voting event, so as to secure the intended privacy.

Linkability in Threshold Ring Signatures. Linkability in threshold ring signatures requires a more precise definition. In particular, there are two possible flavors: two signatures are linked if and only if (1) they are signed by exactly the same set of signers, or (2) they involve a common signer. We call signatures of the former type "*coalition-linkable*" while those of the latter type "*individual-linkable*".

In a coalition-linkable scheme, users are able to sign multiple times without their signatures being linked, as long as they are not collaborating with exactly the same set of signers again. However, in an individual-linkable scheme, a user signing more than once will have the signatures linked, no matter who other collaborating signers are. The scheme we present in this paper falls into the later category.

6 Conclusion

We have given in this paper the first separable linkable ring signature scheme, which also supports an efficient thresholding option. We have also presented the security model and reduce the security of our scheme to well-known hardness assumptions. In particular, we have introduced the security notions of *accusatory linkability* and *non-slanderability* to linkable ring signatures. Applications to event-oriented ring-signing has been discussed.

References

1. M. Abe, M. Ohkubo, and K. Suzuki. 1-out-of-n signatures from a variety of keys. In *ASIACRYPT 2002*, pages 415–432. Springer-Verlag, 2002.
2. G. Ateniese, J. Camenisch, M. Joye, and G. Tsudik. A practical and provably secure coalition-resistant group signature scheme. In *CRYPTO 2000*, pages 255–270. Springer-Verlag, 2000.
3. M. Bellare, D. Micciancio, and B. Warinschi. Foundations of group signatures: formal definitions, simplified requirements and a construction based on general assumptions. In *EUROCRYPT'03*, volume 2656 of *LNCS*. Springer-Verlag, 2003.
4. M. Bellare and P. Rogaway. Random oracles are practical: a paradigm for designing efficient protocols. In *Proceedings of the 1st ACM conference on Computer and communications security*, pages 62–73. ACM Press, 1993.
5. M. Bellare, H. Shi, and C. Zhang. Foundations of group signatures: the case of dynamic groups. Cryptology ePrint Archive, Report 2004/077, 2004. http://eprint.iacr.org/.
6. E. Bresson, J. Stern, and M. Szydlo. Threshold ring signatures and applications to ad-hoc groups. In *Crypto'02*, volume 2442 of *LNCS*, pages 465–480. Springer-Verlag, 2002.
7. J. Camenisch and M. Michels. A group signature scheme based on an RSA-variant. rs RS-98-27, brics, 1998.
8. J. Camenisch and M. Michels. Separability and efficiency for generic group signature schemes. In *Crypto'99*, pages 413–430. Springer-Verlag, 1999.
9. J. Camenisch and M. Stadler. Efficient group signature schemes for large groups (extended abstract). In *CRYPTO'97*, pages 410–424. Springer-Verlag, 1997.
10. D. Chaum and E. van Heyst. Group signatures. In *EUROCRYPT'91*, volume 547 of *LNCS*, pages 257–265. Springer-Verlag, 1991.
11. R. Cramer, I. Damgard, and B. Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In *CRYPTO'94*, pages 174–187. Springer-Verlag, 1994.

12. Y. Desmedt and Y. Frankel. Threshold cryptosystems. In *CRYPTO '89*, volume 435 of *LNCS*, pages 307–315. Springer-Verlag, 1990.
13. Y. Dodis, A. Kiayias, A. Nicolosi, and V. Shoup. Anonymous identification in ad hoc groups. In *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 609–626. Springer-Verlag, 2004.
14. A. Fiat and A. Shamir. How to prove yourself: Practical solution to identification and signature problems. In *CRYPTO'86*, volume 263 of *LNCS*, pages 186–194. Springer-Verlag, 1987.
15. E. Fujisaki and T. Okamoto. Statistical zero knowledge protocols to prove modular polynomial relations. In *CRYPTO'97*, pages 16–30. Springer-Verlag, 1997.
16. E. Fujisaki and T. Okamoto. A practical and provably secure scheme for publicly verifiable secret sharing and its applications. In *Eurocrypt '98*, volume 1403 of *LNCS*, pages 32–46. Springer-Verlag, 1998.
17. S. Goldwasser, S. Micali, and R. L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM J. Comput.*, 17(2):281–308, 1988.
18. A. Kiayias and M. Yung. Group signatures: provable security, efficient constructions, and anonymity from trapdoor-holders. Cryptology ePrint Archive, Report 2004/076, 2004. http://eprint.iacr.org/.
19. J. K. Liu, V. K. Wei, and D. S. Wong. A separable threshold ring signature scheme. In *ICISC 2003*, volume 2971 of *LNCS*, pages 12–26. Springer-Verlag, 2003.
20. J. K. Liu, V. K. Wei, and D. S. Wong. Linkable spontaneous anonymous group signature for ad hoc groups (extended abstract). In *ACISP'04*, volume 3108 of *LNCS*, pages 325–335. Springer-Verlag, 2004.
21. D. Pointcheval and J. Stern. Security proofs for signature schemes. In *EUROCRYPT'96*, volume 1070 of *LNCS*, pages 387–398. Springer-Verlag, 1996.
22. R. L. Rivest, A. Shamir, and Y. Tauman. How to leak a secret. In *ASIACRYPT 2001*, pages 552–565. Springer-Verlag, 2001.
23. A. Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979.
24. D. S. Wong, K. Fung, J. K. Liu, and V. K. Wei. On the RS-code construction of ring signature schemes and a threshold setting of RST. In *ICISC 2003*, volume 2971 of *LNCS*, pages 34–46. Springer-Verlag, 2003.