A Suite of Non-Pairing ID-Based Threshold Ring Signature Schemes with Different Levels of Anonymity

Patrick P. Tsang[1], Man Ho Au[2], Joseph K. Liu[3], Willy Susilo[2], and Duncan S. Wong[4]

[1] Department of Computer Science
Dartmouth College
Hanover NH 03755, USA
patrick@cs.dartmouth.edu

[2] Centre for Computer and Information Security (CCISR)
School of Computer Science and Software Engineering
University of Wollongong, Australia
{aau, wsusilo}@uow.edu.au

[3] Cryptography and Security Department
Institute for Infocomm Research, Singapore
ksliu@i2r.a-star.edu.sg

[4] Department of Computer Science
City University of Hong Kong, Hong Kong
duncan@cityu.edu.hk

Abstract. Since the introduction of Identity-based (ID-based) cryptography by Shamir in 1984, numerous ID-based signature schemes have been proposed. In 2001, Rivest et al. introduced ring signature that provides irrevocable signer anonymity and spontaneous group formation. In recent years, ID-based ring signature schemes have been proposed and almost all of them are based on bilinear pairings. In this paper, we propose the first ID-based threshold ring signature scheme that is not based on bilinear pairings. We also propose the first ID-based threshold 'linkable' ring signature scheme. We emphasize that the anonymity of the actual signers is maintained even against the private key generator (PKG) of the ID-based system. Finally we show how to add identity escrow to the two schemes. Due to the different levels of signer anonymity they support, the schemes proposed in this paper actually form a suite of ID-based threshold ring signature schemes which is applicable to many real-world applications with varied anonymity requirements.

1 Introduction

As the number of applications on the Internet continues to grow, more and more traditional human interactions have been converted to their electronic counterparts: messaging, voting, payments, commerce, etc. The increase in reliance on the Internet potentially erodes personal privacy, the right of the individual to be let alone [58], or the right to determine the amount of personal information which should be available to others [59]. Privacy is important for many reasons, such as impersonation and fraud. As more identity information is collected, correlated, and sold, it becomes easier for criminals to commit fraud. But privacy is more than that, it also concerns about the secrecy of which websites we visited, the candidates we voted for, etc.

Anonymity is one important form of privacy protection. In practice, anonymity diversifies into various forms with different levels of anonymity. For example, look at how anonymous remailers [35] have evolved over time – from type 0 to type I to type II, every successor provides a higher level of anonymity, at the cost of lower efficiency and higher resource consumption. On the other side, for some applications, too high a level of anonymity can do more harm than good. For example, while unconditional anonymity provides maximum protection to users which can be useful for scenarios such as secret leaking [53]. However, unconditional anonymity may not be desirable for some other applications. For instance, in some scenarios one would like to have a trusted third party to have the capability to trace users after the fact that the users have misbehaved, such as tracing double-spenders in an e-cash system.

Designing secure cryptographic schemes with unconditional anonymity is undoubtedly challenging. However, designing schemes with a carefully adjusted level of anonymity is sometimes even more challenging. It is also very rewarding due to the fact that these schemes find many applications in practice. For example, a ring signature scheme [53] allows a signer to generate a signature on behalf of a group of signers such that everyone can be sure that the signature is generated by one of the group members yet no one can tell who the real signer is. Different from group signature, there is no group manager, no member revocation, and it is spontaneous (setup-free). While a *linkable* ring signature [46] allows anyone to tell whether two signatures are generated by the same signer while still maintaining the anonymity of the real signer as a conventional ring signature scheme in the way that no one can revoke the real signer's anonymity.

1.1 Background and Related Work

Identity-based Cryptography. In 1984, Shamir [55] introduced the notion of Identity-based (ID-based) cryptography to simplify certificate management. The unique feature of ID-based cryptography is that a user's public key can be any arbitrary string. Since then, many other ID-based signature schemes have been proposed, despite the fact that the first practical ID-based encryption appeared only until 2001 [13]. In 2004, Bellare et al. [10] developed a framework to analyze the security of ID-based signature schemes and they proved the security (or insecurity) of 14 schemes found in the literature. As in the case of standard signature, there are also blind signature [63], proxy signature [61], proxy blind signature [32], proxy ring signature [6, 63], and proxy signcryption [44] in the paradigm of ID-based cryptography.

Group-oriented Cryptography. This type of schemes has a group of users involved, e.g. secret sharing schemes, group signature schemes, etc. In some of them, group members participate equally well in all the processes and therefore, there is no concern of anonymity. In some other schemes, however, the participation of only one or a proper subset of members is required to complete a process, while the remaining members are not involved in (and are possibly unaware of) the process. Such a distinction between participants and non-participants gives anonymity a meaning. Specifically, a participant may prefer to be indistinguishable from the whole group of members, thus maintaining his privacy in participating the process. According to the level of anonymity the group-oriented cryptographic schemes provide, they can be categorized as follows.

No Anonymity means the identities of the participating users are known to everyone. Privacy is simply not a concern here. For example, in a multi-signature scheme [41, 48], everyone can identify who has contributed in the signing process.

Anonymity means not everyone should be able to identify participating users. A good example is ring signature [53], in which besides the actual signer, no one can identify the actual signer of a signature among a group of possible signers. There have been many different schemes proposed [1, 31, 54, 21] since the first appearance of ring signature in 1994 [30] and the formal introduction of it in 2001 [53]. The first ID-based ring signature was proposed in 2002 [62]. Two constructions in the standard model were proposed [5]. Their first construction was discovered to be flawed [33], while the second construction is only proven secure in a weaker model, namely, selective-ID model. The first scheme claimed to be secure in the standard model is [39] under the trusted setup assumption. However, their proof is wrong and it is unknown whether their scheme is secure or not.[5] Other existing ID-based ring signatures includes [23, 7, 64, 28, 26, 50, 40]. Threshold variant of ID-based ring signatures includes [24, 29, 39]. To the best of the authors' knowledge, all the existing ID-based ring signature schemes are pairing-based except the one in [40] which is RSA-based.

Revocable Anonymity can be summarized as "no anonymity to an authority, but anonymity to anybody else". In schemes with revocable anonymity, there is always an authority who is capable of revoking the anonymity, e.g., under dispute or court order. The authority is often assumed to be trusted not to abuse power. Users are anonymous to everybody other than this authority. Group signature schemes [22, 9, 12] provide revocable anonymity. Many credential systems [16–18] also provide revocable anonymity.

[5] We explicitly point out the flaw in Appendix A.

Linkable Anonymity is “anonymity with a condition”. Schemes with linkable anonymity give maximal anonymity to users who succeeded in satisfying the condition and take away a certain degree of anonymity from users who failed as a punishment. Let us illustrate the idea using a linkable ring signature scheme. In this scheme, users are assumed to sign only once, in which case they enjoy anonymity in full. However, if a user signs twice (or k times, in general), anyone can tell if two signatures are produced by the same user or not, thus resulting in a reduced level of anonymity. Linkable ring signature was introduced in [46]. [57] gave a separable construction that supports threshold. The first constant-size linkable ring signature was proposed in [56]. Linkable group signature first appeared in [49]. Escrowed linkable ring signature was proposed in [27]. The first constant-size linkable ring signature (and revocable if and only if linked variant) was proposed in [4]. The construction, however, was flawed as shown in [42]. A practical application of linkable ring signature is e-voting [25].

A technical difficulty in constructing an ID-based linkable ring signature is that there exists a Private Key Generator (PKG) in the system responsible for issuing users' secret keys yet linkable anonymity should be maintained, even against the PKG. Our construction solves this by modifying the key extraction algorithm such that user's secret key is co-generated by the PKG and the user. This idea is reminiscent to the idea of self-certified keys [37]. It also allows the users in our ID-based linkable signature scheme to refute any framing attacks launched by the PKG through generating another signature which is unlinked to the forged signature.

1.2 Our Contributions and Motivations

- We propose the first ID-based threshold ring signature scheme that is *not* based on bilinear pairings. We show its security under the Strong RSA and DDH Assumption, in the random oracle model [11]. In particular, anonymity of the ring signers is maintained even against the PKG.
- By extending on our basic construction, we propose the first ID-based *linkable* threshold ring signature scheme. All previously proposed linkable ring signature schemes except [27] are not ID-based. [6]
- We show the method of adding identity escrow in both of our schemes. With identity escrow, some trusted authority can revoke the anonymity of a ring signature when it becomes necessary. The ability of revoking the real signer can help prevent the signature scheme from being abused by misbehaving users. The schemes, plus their identity-escrowed counterparts, form a suite of ID-based signature schemes applicable to a wide variety of scenarios with different anonymity requirements. Note that even with identity escrow, the scheme is not the same as a group signature scheme due to the spontaneity property of the ring signature scheme.

Our Motivations. As we have seen many constructions of threshold ring signature schemes [30, 14, 60, 45, 47, 57, 24, 29, 39] proposed recently, there are only few of them [24, 29, 39] under the setting of ID-based cryptography. ID-based ring signature schemes have similar applications to that of conventional public key setting, but with the key escrow property. Applications include whistleblowing [53] and ad hoc group authentication [14]. All ID-based threshold ring signature schemes proposed are pairing based. Also it is obvious to further extend them to a ***linkable*** variant, especially it needs to be secure under the security models we define in this paper. Therefore, the work presented in this paper is mainly motivated by the following two aspects.

1. As of theoretical interest, we target to propose an identity-based scheme which does not rely on security assumptions related to pairings, for example, Gap Diffie-Hellman Problem.
2. All current ID-based threshold ring signature schemes do not allow us to extend it to an ID-based ***linkable*** threshold ring signature scheme. We target to construct a scheme which can be extended so that we can construct a linkable variant.

[6] We note that although the linkable ring signature scheme in [4] is ID-based, it is later proven insecure in [42].

1.3 Comparison

We compare our scheme with other ID-based threshold ring signature schemes [24, 29, 39] in Table 1.

	Signature size (group elements)	Number of pairing in verification	Mathematical Assumption	Security model	Extend to linkable
[24]	$\mathcal{O}(n)$	$\mathcal{O}(n)$	GDH	ROM	No
[29]	$\mathcal{O}(n^2)$	$\mathcal{O}(n^2)$	ECDL, BPI	ROM	No
[39]	$\mathcal{O}(n)$	$\mathcal{O}(n)$	SGH, CDH	Unknown	No
Our schemes	$\mathcal{O}(n)$	0	Strong RSA, DDH	ROM	Yes

Table 1. Comparison of different ID-based threshold ring signature schemes

In the table, n is the number of users included in the ring. The assumptions mentioned include:

- GDH: Gap Diffie-Hellman problem
- ECDL: Elliptic Curve Discrete Logarithm problem
- BPI: Bilinear Pairings Identity problem
- SGH: Subgroup Decision problem
- DDH: Decisional Diffie-Hellman problem

Note that each group element of our scheme is about 1024 bits, while a group element of other pairing-based schemes is about 160 bits.

We also note that although the authors of the scheme in [39] claimed that their scheme is secure in the standard model, we find out a flaw in the proof. It is unknown whether their scheme is secure or not, at least in the standard model. We present the flaw in Appendix A.

Paper Organization. We give some preliminaries in Sec. 2 and define a security model in Sec. 3. We then propose an ID-based threshold ring signature scheme in Sec. 4 and an ID-based linkable variant in Sec. 5. In Sec. 6, we show how to add identity escrow to our schemes.

2 Preliminaries

A safe prime p is a prime such that $(p-1)/2$ is also prime[7]. Denote by $QR(N)$ the group of quadratic residues modulo the safe prime product N. For positive real numbers $a \leq b$, $\lfloor a \rfloor$ denotes the greatest integer less than or equal to a; $[a, b]$ denotes the set $\{x \in \mathbb{Z} | \lfloor a \rfloor \leq x \leq \lfloor b \rfloor\}$ and $S(a, b)$ denotes $[\lfloor a \rfloor - \lfloor b \rfloor + 1, \lfloor a \rfloor + \lfloor b \rfloor - 1]$. If S is a set, $\wp(S)$ denotes the power set of S and $\wp_t(S)$ denotes the set of elements in $\wp(S)$ of size t, i.e. $\wp_t(S) \doteq \{s \in \wp(S) | \ |s| = t\}$. A *negligible* function $\nu(\lambda)$ is a function such that for all polynomial poly and sufficiently large λ, $\nu(\lambda) < 1/\mathsf{poly}(\lambda)$. When G is a finite cyclic group, define $\mathcal{G}(G)$ to be the set of generators of G, i.e. $\{g \in G | \langle g \rangle = G\}$.

2.1 Mathematical Assumptions

Definition 1 (Strong RSA [8, 36]). *Let $n = pq$ be an RSA modulus. Let G be a cyclic subgroup of $\mathbb{Z}_n^*$ of order u. Given n and $z \in_R G$, the* Strong RSA Problem *is to find $x \in G$ and $e \in \mathbb{Z}_{>1}$ such that $z = x^e \mod n$. The* Strong RSA Assumption *says that there exists no PPT algorithm that can solve the Strong RSA Problem, in time polynomial in the size of $|u|$.*

In our schemes, we need to make restriction to safe primes for p and q in the Strong RSA assumption. However, it is easy to see that the Strong RSA assumption without this restriction implies the Strong RSA assumption with this restriction, assuming that safe primes are sufficiently dense.

[7] Although it has never been proven, it is widely conjectured and amply supported by empirical evidence, that safe primes are sufficiently dense.

Definition 2 (Decisional Diffie-Hellman (DDH) [11]). *Let G be a cyclic group generated by g of order u. The* DDH Problem *is to distinguish between the distributions (g, g^a, g^b, g^c) and (g, g^a, g^b, g^{ab}), with $a, b, c \in_R \mathbb{Z}_u$. The* DDH Assumption *says there exists no PPT algorithm solve the DDH Problem, in time polynomial in the size of $|u|$.*

2.2 Signature of Knowledge

A Σ-protocol for an **NP**-relation R is a 3-round two-party protocol, such that for every input $(x, y) \in R$ to a prover $\mathcal{P}$ and y to a verifier $\mathcal{V}$, the first $\mathcal{P}$-round yields a commitment t, the subsequent $\mathcal{V}$-round replies with a challenge c, and the last $\mathcal{P}$-round concludes by sending a response s. At the end of a run, $\mathcal{V}$ outputs a 0/1 value, functionally dependent on y and the transcript $\pi \doteq (t, c, s)$ only. A transcript is valid if the output of the honest verifier is 1. Additionally, we require a Σ-protocol to satisfy:

- *(Special Soundness.)* There exists a computable function $\mathcal{K}$ (Knowledge Extractor) that on input y in the domain of the second component of R and a pair of valid transcripts (t, c, s) and (t, c', s'), with the same commitment, outputs x such that $(x, y) \in R$.
- *(Special Honest-Verifier Zero-Knowledge (Special HVZK).)* There exists an efficient algorithm $\mathcal{S}$ (Simulator) that on input y in the domain of the second component of R and a challenge c, outputs a pair of commitment/response messages t, s, such that the transcript $\pi \doteq (t, c, s)$ is valid, and it is distributed according to the distribution $(\mathcal{P}(x, y) \leftrightarrow \mathcal{V}(y))$.

A signature of knowledge allows a signer to prove the knowledge of a secret with respect to some public information non-interactively. Following [20], we call this type of signatures "a signature based on proofs of knowledge", SPK for short. A HVZK Σ-protocol can be turned into a SPK by setting the challenge to the hash value of the commitment together with the message to be signed [34]. Such schemes can be proven secure against existential forgery under chosen-message attack [38] in the random oracle model using the proofing technique introduced in [51].

3 Definitions and Security Models

3.1 ID-TRS (ID-based Threshold Ring Signature)

An ID-Based Threshold Ring Signature (ID-TRS) scheme is defined as a tuple of four probabilistic polynomial-time (PPT) algorithms:

- `ID-TRS.Setup`. On input 1^λ where $\lambda \in \mathbb{N}$ is a security parameter, it outputs a master secret key s and a system parameter set $\mathsf{param} = (1^\lambda, \mathcal{S}, \mathcal{M}, \Psi)$, where $\mathcal{S}$ is the user secret key space, $\mathcal{M}$ the message space, and Ψ the signature space.
- `ID-TRS.Extract`. On input param, an identity $\mathsf{ID}_i \in \{0,1\}^*$ for a user and the master secret key s, it outputs a user secret key $s_i \in \mathcal{S}$ for the user.
- `ID-TRS.Sign`. On input param, an integer n as the ring size, a threshold $t \in [1, n]$, an identity set $\{\mathsf{ID}_i \in \{0,1\}^* \mid i \in [1, n]\}$, a message $m \in \mathcal{M}$, and a t-element user secret key set $\{s_j \in \mathcal{S} \mid j \in \Pi\}$ where $\Pi \in \wp_t([1, n])$, it outputs an ID-based (t, n)-threshold ring signature $\sigma \in \Psi$.
- `ID-TRS.Verify`. On input param, ring size n, threshold t, identity set $\{\mathsf{ID}_i \in \{0,1\}^* \mid i \in [1, n]\}$, message $m \in \mathcal{M}$ and signature $\sigma \in \Psi$, it outputs either valid or $\mathsf{invalid}$.

Correctness. An ID-TRS scheme defined above satisfies **verification correctness** if for any $(s, \mathsf{param}) \leftarrow$ `ID-TRS.Setup`(1^λ), $n \in \mathbb{N}$, $t \in [1, n]$, $L = \{\mathsf{ID}_i \in \{0,1\}^* \mid i \in [1, n]\}$, $\Pi \in \wp_t([1, n])$, $\{s_i \leftarrow$ `ID-TRS.Extract`$(\mathsf{param}, \mathsf{ID}_i, s) \mid i \in [1, n]\}$ and $m \in \mathcal{M}$, if $\sigma \leftarrow$ `ID-TRS.Sign`$(\mathsf{param}, n, t, L, m, \{s_j \mid j \in \Pi\})$, then $\mathsf{valid} \leftarrow$ `ID-TRS.Verify`$(\mathsf{param}, n, t, L, m, \sigma)$.

A secure ID-TRS scheme should be ***unforgeable*** and ***anonymous***. Specific to ID-based setting, our security model captures the *adaptive chosen ID attacks*. Let $\mathcal{A}$ be an adversary. The capabilities of $\mathcal{A}$ is modeled by making queries to the following oracles:

Hash: [8] $\mathcal{A}$ can ask for hash values of any finite length strings.
Key: On input ID_i, $s_{\mathsf{ID}_i} \leftarrow \texttt{ID-TRS.Extract}(\mathsf{param}, \mathsf{ID}_i, s)$ is returned. The oracle is stateful, meaning that if $\mathsf{ID}_i = \mathsf{ID}_j$, then $s_i = s_j$.
Signature: On input an identity set $L = \{\mathsf{ID}_i\}_{i\in[1,n]}$, a signer set $\Pi \in \wp_t([1,n])$ and a message m, the oracle returns $\sigma \leftarrow \texttt{ID-TRS.Sign}(\mathsf{param}, n, t, L, m, \{s_i \mid i \in \Pi\})$.

Definition 3 (Unforgeability). *We consider the following game.*

- (Initialization Phase.) *Challenger $\mathcal{C}$ generates $(s, \mathsf{param}) \leftarrow \texttt{ID-TRS.Setup}(1^\lambda)$ and sends param to $\mathcal{A}$.*
- (Probing Phase.) *$\mathcal{A}$ makes queries to any of the oracles.*
- *$\mathcal{A}$ outputs $L^* = \{ID_i \in \{0,1\}^* \mid i \in [1,n]\}$, $m^* \in \mathcal{M}$ and an ID-based (t,n)-threshold ring signature $\sigma^* \in \Psi$.*

*Restrictions are: (1) (m^*, L^*) should not be queried to **Signature**; (2) strictly less than t users in L^* are queried to oracle **Key**.*

An ID-TRS is unforgeable (i.e. existentially unforgeable against adaptive chosen-message-and-ID attacks) or EUF-IDTR-CMIA secure if for all sufficiently large λ and any PPT adversary, the probability that $\mathsf{valid} \leftarrow \texttt{ID-TRS.Verify}(\mathsf{param}, n, t, L^*, m^*, \sigma^*)$ is negligible. The probability is taken over the coin tosses of $\mathcal{C}$ and $\mathcal{A}$.

On anonymity, we emphasize that although the key escrow property of ID-based cryptography is inherent, the anonymity of the actual signers should still be protected against the PKG. Indeed, our model below captures the scenario that the PKG is an adversary which tries to find out who the actual signers are.

Definition 4 (Anonymity).

- (Initialization Phase.) *Challenger $\mathcal{C}$ generates $(s, \mathsf{param}) \leftarrow \texttt{ID-TRS.Setup}(1^\lambda)$ and sends **both** param and s to $\mathcal{A}$.*
- (Probing Phase.) *Same as that in Unforgeability definition.*
- (Challenge Phase.) *$\mathcal{A}$ gives $\mathcal{C}$ an identity set $L = \{ID_i \mid i \in [1,n]\}$, $t \in [1,n]$ and $m \in \mathcal{M}$. $\mathcal{C}$ picks randomly $\Pi \in_R \wp_t([1,n])$ and returns $\sigma \leftarrow \texttt{ID-TRS.Sign}(\mathsf{param}, n, t, L, m, \{s_i \mid i \in \Pi\})$.*
- *$\mathcal{A}$ continues making queries to any of the oracle. Finally, $\mathcal{A}$ outputs $\hat{\pi} \in [1,n]$.*

An ID-TRS scheme is anonymous (i.e. signer indistinguishable against adaptive chosen-message-and-ID attacks) or IND-IDTR-CMIA secure if for all sufficiently large λ and any PPT adversary, the probability that $\hat{\pi} \in \Pi$ is negligibly greater than $\frac{t}{n}$. The probability is taken over the coin tosses of $\mathcal{C}$ and $\mathcal{A}$.

3.2 ID-LTRS (ID-based Linkable Threshold Ring Signature)

As introduced at the beginning of this paper, ID-LTRS (ID-based ***Linkable*** Threshold Ring Signature) scheme is a variant of ID-TRS. In the following, we give the formal definition and specify the security requirements.

- `ID-LTRS.Setup`. Same as `ID-TRS.Setup`, except: (1) it has an additional input $k \in \mathbb{N}$ which represents the maximum number of events that the system supports, and (2) param additionally includes an event-ID space $\mathcal{E}$. We have $|\mathcal{E}| = k$.
- `ID-LTRS.Extract Protocol`. User with identity ID_i engage with PKG in the protocol with common input param. After the protocol, the user is obtained a user secret key $s_i \in \mathcal{S}$.
- `ID-LTRS.Sign,Verify`. Same as `ID-TRS.Sign,Verify`, except they both additionally have an input event-ID $e \in \mathcal{E}$.
- `ID-LTRS.Link`. On input param, $e \in \mathcal{E}$, two ring sizes n_1, n_2, two thresholds $t_1 \in [1, n_1]$ and $t_2 \in [1, n_2]$, two identity sets $\mathcal{Y}_j = \{\mathsf{ID}_i^{(j)} \mid i \in [1, n_j]\}$ for $j = 1, 2$, two messages $m_1, m_2 \in \mathcal{M}$, and two signatures $\sigma_1, \sigma_2 \in \Psi$ such that $\mathsf{valid} \leftarrow \texttt{ID-LTRS.Verify}(\mathsf{param}, e, n_j, t_j, \mathcal{Y}_j, m_j, \sigma_j)$ for $j = 1, 2$, the algorithm returns either linked or $\mathsf{unlinked}$.

[8] The hash oracle is only needed in the random oracle model.

Note that we require an interactive extract protocol (between the PKG and user) instead of the normal extract algorithm here. The purpose is to prevent the PKG from learning the identity of the actual signer from the additional linking tag.

Correctness. Besides **verification correctness** (which is defined similarly to that for ID-TRS), an ID-LTRS scheme also satisfies **linking correctness** if

$$\mathsf{linked} \leftarrow \texttt{ID-LTRS.Link}(\mathsf{param}, e, n_1, n_2, t_1, t_2, \mathcal{Y}_1, \mathcal{Y}_2, m_1, m_2, \sigma_1, \sigma_2)$$

for any $(s, \mathsf{param}) \leftarrow \texttt{ID-LTRS.Setup}(1^\lambda, k)$, $n_1, n_2 \in \mathbb{N}$, $t_j \in [1, n_j]$, $\mathcal{Y}_j = \{\mathsf{ID}_i^{(j)} \mid i \in [1, n_j]\}$, $\Pi_j \in \wp_{t_j}([1, n_j])$, $\{s_i^{(j)} \leftarrow \texttt{ID-LTRS.Extract Protocol} \mid i \in [1, n_j]\}$, $m_1, m_2 \in \mathcal{M}$ such that $\sigma_j \leftarrow \texttt{ID-LTRS.Sign}(\mathsf{param}, n_j, t_j, \mathcal{Y}_j, m_j, \{s_i^{(j)} \mid i \in \Pi_j\})$, for $j = 1, 2$ and $\Pi_1 \cap \Pi_2 \neq \emptyset$.

Remark: According to [56], linkability for threshold ring signatures is diversified into *individual-linkability* and *coalition-linkability*, our definition belongs to the former type. That is, two signatures are linked if they share at least one common signer even though the two identity sets are different. The definition of linkability affects directly the level of anonymity due to the additional access to `ID-LTRS.Link` by the adversary.

The security requirements of ID-LTRS schemes include ***Unforgeability***, ***Anonymity***, ***Linkability*** and ***Non-slanderability***.

The definition of Unforgeability for ID-LTRS is the same as that for ID-TRS schemes. For anonymity, a crucial difference between ID-LTRS and ID-TRS is that in the former, the adversary cannot query signatures of a user who appears in the challenge phase. The reason is that if the adversary has obtained some signature of user i in ID-LTRS, it can tell if the signature for challenge is generated by this user due to the linking property. Also note that in the game below, we equip the adversary with the master secret key. This implies that we require an ID-LTRS to be anonymous (as defined below) even when the adversary colludes with the PKG. It also simulates the situation that an outside attacker somehow steals the master secret key of the PKG. *Note that we do not model the case of a* ***malicious PKG*** *[3] where the adversary acts as a malicious PKG who generates all public parameters instead of just given the secret key.*

Definition 5 (L-Anonymity).

- (Initialization Phase.) *$\mathcal{C}$ runs $(\mathsf{param}, s) \leftarrow \texttt{ID-LTRS.Setup}(1^\lambda, k)$ and sends (param, s) to $\mathcal{A}$.*
- (Probing Phase I.) *$\mathcal{A}$ makes queries to any of the oracles. Suppose $\mathcal{A}$ makes a total of v queries to* ***Key***. *The restriction is that $v < n - t$.*
- (Challenge Phase.) *$\mathcal{A}$ gives $\mathcal{C}$ a ring size n, a threshold $t \in [1, n]$, an identity set $L = \{\mathsf{ID}_i \mid \in [1, n]\}$ and a message $m \in \mathcal{M}$. $\mathcal{C}$ picks randomly an index set $\Pi \in_R \wp_t([1, n])$ such that every element in Π is not contained in any of the queries to* ***Signature*** *and* ***Key***. *$\mathcal{C}$ computes $\sigma \leftarrow \texttt{ID-TRS.Sign}(\mathsf{param}, n, t, L, m, \{s_i | i \in \Pi\})$.*
- (Probing Phase II.) *As in Probing Phase I, $\mathcal{A}$ makes queries to the oracles. Suppose $\mathcal{A}$ makes a total of v' queries to* ***Key*** *in this phase. The restriction is that $v' < n - t - v$. If any of the queries to* ***Signature*** *or* ***Key*** *contains an identity d such that $d \in \Pi$, $\mathcal{C}$ halts.*
- *$\mathcal{A}$ outputs an index $\hat{\pi}$.*

An ID-LTRS scheme is signer indistinguishable against adaptive chosen-message-and-identity attacks (or IND-IDLTR-CMIA secure) if for all sufficiently large λ and any PPT adversary, the probability that $\hat{\pi} \in \Pi$ is negligibly greater than $\frac{t}{n-(v+v')}$.

Linkability for ID-LTRS schemes is compulsory, that is, it should be infeasible for a signer to generate two signatures such that they are determined to be $\mathsf{unlinked}$ using `ID-LTRS.Link`. The following definition/game essentially captures a scenario that an adversary tries to generate two ID-LTRS signatures, say an ID-based (t_1, n_1)-threshold linkable ring signature and an ID-based (t_2, n_2)-threshold linkable ring signature, using strictly fewer than $t_1 + t_2$ user secret keys, so that these two signatures are determined to be $\mathsf{unlinked}$ using `ID-LTRS.Link`. If the ID-LTRS scheme is unforgeable (as defined above), then these signatures can only be generated if at least t_1 and t_2 user secret keys are known, respectively. If strictly fewer than $t_1 + t_2$ user secret keys are known, then there must be at least one user which is in common to both of the signatures. Therefore, this model can effectively capture the definition of linkability for ID-LTRS schemes.

Definition 6 (Linkability).

- (Initialization Phase.) *$\mathcal{C}$ runs $(\mathsf{param}, s) \leftarrow$ `ID-LTRS.Setup`$(1^\lambda, k)$ and sends param to $\mathcal{A}$.*
- (Probing Phase.) *$\mathcal{A}$ makes queries to any of the oracles.*
- *$\mathcal{A}$ outputs two ring sizes n_1, n_2, an event-ID $e \in \mathcal{E}$, two thresholds $t_1 \in [1, n_1]$ and $t_2 \in [1, n_2]$, two identity sets $\mathcal{Y}_1 = \{\mathsf{ID}_i \mid i \in [1, n_1]\}$ and $\mathcal{Y}_2 = \{\mathsf{ID}_i \mid i \in [1, n_2]\}$, two messages $m_1, m_2 \in \mathcal{M}$, an ID-based (t_1, n_1)-linkable threshold ring signature σ_1 and an ID-based (t_2, n_2)-linkable threshold ring signature σ_2. The restrictions are: (1) $(m_1, \mathcal{Y}_1)$ and $(m_2, \mathcal{Y}_2)$ have never been queried to **Signature**; (2) strictly fewer than $t_1 + t_2$ secret keys of $\mathcal{Y}_1 \cup \mathcal{Y}_2$ have been obtained from **Key**.*

An ID-LTRS scheme is linkable (or IDLTR-LINK secure) if for all sufficiently large λ and any PPT adversary, it is negligible to have all the following conditions hold.

- *$\mathsf{valid} \leftarrow$ `ID-LTRS.Verify`$(\mathsf{param}, e, n_j, t_j, \mathcal{Y}_j, m_j, \sigma_j)$, for $j = 1, 2$.*
- *$\mathsf{unlinked} \leftarrow$ `ID-LTRS.Link`$(\mathsf{param}, e, n_1, n_2, t_1, t_2, \mathcal{Y}_1, \mathcal{Y}_2, m_1, m_2, \sigma_1, \sigma_2)$*

Non-slanderability ensures that no signer can generate a signature which is determined to be linked by `ID-LTRS.Link` with another signature which is not generated by the signer. In other words, it prevents adversaries from framing honest users. Also note that we require that even the PKG cannot frame an honest user. This is modeled by equipping the adversary with the master secret key.

Definition 7 (Non-slanderability).

- (Initialization Phase.) *$\mathcal{C}$ runs $(\mathsf{param}, s) \leftarrow$ `ID-LTRS.Setup`$(1^\lambda, k)$ and sends (param, s) to $\mathcal{A}$.*
- (Probing Phase I.) *$\mathcal{A}$ makes queries to any of the oracles.*
- (Challenge Phase.) *$\mathcal{A}$ gives $\mathcal{C}$ a ring size n, an event-ID $e \in \mathcal{E}$, a threshold $t \in [1, n]$, an identity set $\mathcal{Y} = \{\mathsf{ID}_i \mid i \in [1, n]\}$, a t-element set of insider identities $\mathcal{V} \subseteq \mathcal{Y}$, and a message $m \in \mathcal{M}$. $\mathcal{C}$ returns $\sigma \leftarrow$ `ID-LTRS.Sign`$(\mathsf{param}, n, t, \mathcal{Y}, m, \{s_i \mid i \in \mathcal{V}\})$*
- (Probing Phase II.) *Same as Probing Phase I.*
- (End Game Phase.) *$\mathcal{A}$ outputs a ring size n', a threshold t', an identity set $\mathcal{Y}'$, a message m' and a signature σ'.*

An ID-LTRS scheme is non-slanderable (or IDLTR-NON-SLAND secure) if for all sufficiently large λ and any PPT adversary, it is negligible to have all the following conditions hold.

- *$\mathsf{valid} \leftarrow$ `ID-LTRS.Verify`$(\mathsf{param}, e, n', t', \mathcal{Y}', m', \sigma')$*
- *$\mathsf{linked} \leftarrow$ `ID-LTRS.Link`$(\mathsf{param}, e, n, n', t, t', \mathcal{Y}, \mathcal{Y}', m, m', \sigma, \sigma')$*

*The restrictions of the game above are: (1) σ' is not returned by oracle **Signature**; (2) none of the user secret keys corresponding to elements in $\mathcal{V}$ has been returned by oracle **Key**.*

4 Our ID-TRS Scheme

We first give an overview of our construction. For an identity ID, the corresponding secret key is (a, x), with $x > 1$, such that $a^x \equiv H_{id}(\mathsf{ID}) \pmod{N}$, where $H_{id} : \{0,1\}^* \rightarrow QR(N)$ is some hash function. The modulus N is a product of two equal-length safe primes with factorization only known to the PKG.

A user proves the knowledge of his secret key by running the Σ-protocol given by:

$$PK\{(a, x) : y \equiv a^x \wedge x \in \Gamma\}$$

for $y = H_{id}(\mathsf{ID})$ and some suitable range Γ. An ID-based signature scheme is readily available after carrying out the Fiat-Shamir transformation on the Σ-protocol:

$$SPK_1\{(a, x) : y \equiv a^x \wedge x \in \Gamma\}(m). \tag{1}$$

Now, to extend the IBS scheme construction above into a threshold ring setting, we implement the following signature of knowledge (SPK):

$$SPK_2 \left\{ (\alpha_i, \chi_i)_{i=1}^{n} : \bigvee_{\mathcal{J}\in\wp_t([1,n])} \bigwedge_{i\in\mathcal{J}} y_i \equiv \alpha_i^{\chi_i} \wedge \chi_i \in \Gamma \right\} (m) \tag{2}$$

with $y_i = H_{id}(\mathsf{ID}_i)$ for all $i \in [1, n]$. This SPK proves that there exists d identities in $\{\mathsf{ID}_1, \cdots, \mathsf{ID}_n\}$ such that the prover knows the secret keys corresponding to these identities. To implement SPK_2, we incorporate the polynomial interpolation technique [30] into SPK_1.

We now describe the details of our ID-based (t, n)-threshold ring signature scheme.

- `ID-TRS.Setup`. On input a security parameter λ, the algorithm randomly generates a safe prime product $N = pq = (2p' + 1)(2q' + 1)$, where $|p'| = |q'| = \lambda$. It then selects two cryptographic hash functions $H_{id} : \{0,1\}^* \to QR(N)$ and $H_{sig} : \{0,1\}^* \to \mathbb{Z}_{2^\kappa}$. For security analysis, we consider them to behave as random oracles. It also randomly picks $g_1, g_2, g_3 \in QR(N)$ that are generators of $QR(N)$.
  To implement H_{id} using a conventional string-based hash function, we need to randomly choose another generator g of $QR(N)$ and define H_{id} as $\mathsf{ID} \to \mathsf{g}^{\mathsf{h}(\mathsf{ID})} \bmod N$, where $\mathsf{h} : \{0,1\}^* \to \{0,1\}^{2\lambda+\theta}$ is a hash function. The parameter $\theta > 0$ defines the quality of the hash output of H_{id}. A good construction of H_{id} should have the hash value distributed uniformly on $QR(N)$. It can be seen that the construction above can yield a good distribution when θ is large enough. In practice, we may consider setting θ to 8.
  Let $\kappa, \gamma_1, \gamma_2 \in \mathbb{N}$ and $1 < \epsilon \in \mathbb{R}$ be further security parameters such that $\gamma_1 - 2 > \epsilon(\gamma_2 + \kappa) > 2\lambda$. Define $\Gamma' \doteq S(2^{\gamma_1}, 2^{\gamma_2})$, and $\Gamma \doteq S(2^{\gamma_1}, 2^{\epsilon(\gamma_2+\kappa)})$. The master secret key is set to $\mathsf{msk} := (p, q)$. The list of system parameters is $\mathsf{param} := (\lambda, \kappa, \epsilon, N, H_{id}, H_{sig}, g_1, g_2, g_3, \Gamma', \Gamma)$.
  To achieve security comparable to the standard 1024-bit RSA signature, $\lambda = 512$, $\kappa = 160$, $\epsilon = 1.1$, $\gamma_1 = 1080$, $\gamma_2 = 800$ can be used as the security parameters. For security analysis, we require that all these security parameters to be sufficiently large. It is also important for the generators $\mathsf{g}, g_1, g_2, g_3$ are generated independently, that is, their relative discrete logarithm should not be known to anyone. This is to prevent the secret keys of users from being known from the auxiliary commitments which is defined below and make sure that the proper implementation of H_{id} described above.
- `ID-TRS.Extract`. On input a new user ID ID_i, the algorithm computes $y_i := H_{id}(\mathsf{ID}_i)$, picks a prime $x_i \in_R \Gamma'$, and then solves $a_i^{x_i} \equiv y_i \pmod N$ for a_i using the master secret key msk. It finally returns the user's secret key $sk_i := (a_i, x_i)$. An entry $\langle \mathsf{ID}_i, y_i, a_i, x_i \rangle$ is recorded. On input an old user ID, the algorithm retrieve the corresponding entry to maintain consistency.
- `ID-TRS.Sign`. On input the list of system parameters param, a group size $n \in \mathbb{N}$ of size polynomial in λ, a threshold $t \in [1, n]$, a set of n IDs $\mathcal{Y} = \{\mathsf{ID}_1, \cdots, \mathsf{ID}_n\}$, a list of t secret keys $\mathcal{X} = \{sk_{\pi_1}, \cdots, sk_{\pi_t}\}$ such that the corresponding public key ID_{π_i} of each $sk_{\pi_i} = (a_{\pi_i}, x_{\pi_i})$ is contained in $\mathcal{Y}$, a message $m \in \{0,1\}^*$, the algorithm first sets $\Pi := \{\pi_1, \cdots, \pi_t\} \subseteq [1, n]$, computes $y_i := H_{id}(\mathsf{ID}_i)$ for all $i \in [1, n]$ and then does the following:
  1. *(Auxiliary commitment.)* For all $i \in \Pi$, pick $u_i \in_R \pm\{0,1\}^{2\lambda}$ and compute $w_i := u_i x_i$. Compute in modulo N:
     $$A_{i,1} := g_1^{u_i}, \ A_{i,2} := a_i g_2^{u_i}, \ A_{i,3} := g_1^{x_i} g_3^{u_i}.$$
     For all $i \in [1, n] \backslash \Pi$, pick $A_{i,1}, A_{i,2}, A_{i,3} \in_R QR(N)$.
  2. *(Commitment.)* For all $i \in \Pi$, pick $r_{i,x} \in_R \pm\{0,1\}^{\epsilon(\gamma_2+\kappa)}$, $r_{i,u} \in_R \pm\{0,1\}^{\epsilon(2\lambda+\kappa)}$, $r_{i,w} \in_R \pm\{0,1\}^{\epsilon(\gamma_1+2\lambda+\kappa+1)}$. Compute in modulo N:
     $$T_{i,1} := g_1^{r_{i,u}}, \ T_{i,2} := g_1^{r_{i,x}} g_3^{r_{i,u}}, \ T_{i,3} := A_{i,1}^{r_{i,x}} g_1^{-r_{i,w}}, \ T_{i,4} := A_{i,2}^{r_{i,x}} g_2^{-r_{i,w}}.$$
     For all $i \in [1, n] \backslash \Pi$, pick $c_i \in_R \mathbb{Z}_{2^\kappa}$, $s_{i,u} \in_R \pm\{0,1\}^{\epsilon(2\lambda+\kappa)}$, $s_{i,x} \in_R \pm\{0,1\}^{\epsilon(\gamma_2+\kappa)}$, $s_{i,w} \in_R \pm\{0,1\}^{\epsilon(\gamma_1+2\lambda+\kappa+1)}$. Compute in modulo N:
     $$\begin{aligned} T_{i,1} &:= g_1^{s_{i,u}} A_{i,1}^{c_i}, & T_{i,2} &:= g_1^{s_{i,x} - c_i 2^{\gamma_1}} g_3^{s_{i,u}} A_{i,3}^{c_i}, \\ T_{i,3} &:= A_{i,1}^{s_{i,x} - c_i 2^{\gamma_1}} g_1^{-s_{i,w}}, & T_{i,4} &:= A_{i,2}^{s_{i,x} - c_i 2^{\gamma_1}} g_2^{-s_{i,w}} y_i^{c_i}. \end{aligned}$$

3. *(Challenge.)* Compute

$$c_0 := H_{sig}(\mathsf{param}, n, d, (y_i, A_{i,1}, A_{i,2}, A_{i,3})_{i=1}^n, (T_{i,1}, \cdots, T_{i,4})_{i=1}^n, m).$$

4. *(Response.)* Generate a polynomial f over $GF(2^\kappa)$ of degree at most $(n-t)$ such that $c_0 = f(0)$ and $c_i = f(i)$ for all $i \in [1,n] \backslash \Pi$. For all $i \in \Pi$, compute $c_i := f(i)$, and compute in $\mathbb{Z}$:

$$s_{i,u} := r_{i,u} - c_i u_i, \;\; s_{i,x} := r_{i,x} - c_i(x_i - 2^{\gamma_1}), \;\; s_{i,w} := r_{i,w} - c_i w_i.$$

5. *(Signature.)* Set $\sigma' := (f, (s_{i,u}, s_{i,x}, s_{i,w})_{i=1}^n)$.
6. *(Output.)* Return the signature as: $\sigma := ((A_{i,1}, A_{i,2}, A_{i,3})_{i=1}^n, \sigma')$.
Remark: step 2 to 4 together contribute to the signing algorithm of:

$$SPK_3 \left\{ \begin{pmatrix} u_i, \\ x_i, \\ w_i \end{pmatrix}_{i=1}^{n} : \bigvee_{\mathcal{J} \in \wp_t([1,n])} \bigwedge_{i \in \mathcal{J}} \begin{array}{l} A_{i,1} \equiv g_1^{u_i} \wedge A_{i,3} \equiv g_1^{x_i} g_3^{u_i} \wedge \\ A_{i,1}^{x_i} \equiv g_1^{w_i} \wedge A_{i,2}^{x_i} \equiv g_2^{w_i} y_i \wedge \\ x_i \in \Gamma \end{array} \right\} (m), \tag{3}$$

which is an instantiation of SPK_2. The signature of SPK_3 is σ' in step 5.

- `ID-TRS.Verify`. On input param, a group size n of length polynomial in λ, a threshold $t \in [1,n]$, a set $\{\mathsf{ID}_i \in \{0,1\}^* | i \in [1,n]\}$ of n user identities, a message $m \in \mathcal{M}$, a signature $\sigma \in \Psi$, the algorithm computes $y_i := H_{id}(\mathsf{ID}_i)$ for all $i \in [1,n]$ and then does the following.
  1. Check if f is a polynomial over $GF(2^\kappa)$ of degree at most $(n-t)$.
  2. For all $i \in [1,n]$, compute $c_i := f(i)$ and compute in modulo N:

$$\begin{array}{ll} T'_{i,1} := g_1^{s_{i,u}} A_{i,1}^{c_i}, & T'_{i,2} := g_1^{s_{i,x} - c_i 2^{\gamma_1}} g_3^{s_{i,u}} A_{i,3}^{c_i}, \\ T'_{i,3} := A_{i,1}^{s_{i,x} - c_i 2^{\gamma_1}} g_1^{-s_{i,w}}, & T'_{i,4} := A_{i,2}^{s_{i,x} - c_i 2^{\gamma_1}} g_2^{-s_{i,w}} y_i^{c_i}. \end{array}$$

  3. Check if the following statements hold: $s_{i,u} \overset{?}{\in} \{0,1\}^{\epsilon(2\lambda+\kappa)+1}$, $s_{i,x} \overset{?}{\in} \{0,1\}^{\epsilon(\gamma_2+\kappa)+1}$, $s_{i,w} \overset{?}{\in} \{0,1\}^{\epsilon(\gamma_1+2\lambda+\kappa+1)+1}$, for all $i \in [1,n]$, and

$$f(0) \overset{?}{=} H_{sig}(\mathsf{param}, n, t, (y_i, A_{i,1}, A_{i,2}, A_{i,3})_{i=1}^n, (T'_{i,1}, \cdots, T'_{i,4})_{i=1}^n, m).$$

  4. Accept if all checks pass and reject otherwise.

  Remark: The above verification actually verifies SPK_3.

The proof for correctness is straightforward. We show its security in Appendix C.

5 ID-Based Linkable Threshold Ring Signature

In this section, we propose the *first* ID-based linkable threshold ring signature (ID-LTRS) and present its security analysis.

5.1 Our Proposed Construction

The key idea is to include a tag to the original ID-TRS signature for the purpose of linking. Such a tag is a one-way and unique image of the signer's secret signing key. To prevent PKG from learning the signer identity from the tag, we modify the extract protocol so that the secret signing key is co-generated by signer and PKG. The signature, besides proving the knowledge of a secret signing key, now also proves that the tag is formed correctly. To test whether two signatures are linked, one simply checks if the two signatures contain the same tag. Below is our construction.

- `ID-LTRS.Setup`. Same as `ID-TRS.Setup`, except it additionally picks $e_i \in_R \mathcal{G}(QR(N))$ for all $i \in [1,k]$ and sets $\mathcal{E} := \{e_i | i \in [1,k]\}$. It also picks one more generator $h \in_R \mathcal{G}(QR(N))$. Define λ_1, λ_2 such that $\gamma_2 > \lambda_1 + 2$, $\lambda_1 > \epsilon(\lambda_2 + \kappa)$ and $\lambda_2 > 2\lambda$. Define $\tilde{\Lambda}' =]0, 2^{\lambda_2}[$, $\Lambda' = S(2^{\lambda_1}, 2^{\lambda_2})$ and $\Lambda = S(2^{\lambda_1}, 2^{\epsilon(\lambda_2+\kappa)})$

- `ID-LTRS.Extract Protocol`. User i with ID ID_i engage with PKG in the following protocol.
  1. User randomly generates $\tilde{d_i} \in_R \tilde{\Lambda}'$, a random $\tilde{r} \in_R \pm\{0,1\}^{2\lambda}$ and sends $C_1 = g_1^{\tilde{d_i}} g_2^{\tilde{r}}$, together with knowledge of representation of C_1 with respect to g_1 and g_2 to PKG. It also sends ID_i together.
  2. PKG checks that the proof is valid and randomly selects $\alpha, \beta \in_R \tilde{\Lambda}'$ and sends α, β to user.
  3. User computes $d_i = 2^{\lambda_1} + (\alpha\tilde{d_i} + \beta \bmod 2^{\lambda_2})$ and sends $C_2 = h^{d_i}$ together with the proof of validity to PKG. This can be done by $SPK\{(u, v, w) : C_1^{\alpha} g_1^{\beta} = g_1^u g_1^{2^{\lambda_2} v} g_2^w \wedge C_2 = h^u \wedge u \in \Lambda'\}(m)$
  4. PKG checks if the proof is valid, and picks a prime $x_i \in_R \Gamma'$, and then solves $a_i^{x_i} \equiv y_i C_2 \pmod N$ for a_i using the master secret key msk, where $y_i = H(\mathsf{ID}_i)$. Return (a_i, x_i) to user and record the entry $\langle \mathsf{ID}_i, y_i, a_i, x_i \rangle$.
  5. User checks if $a_i^{x_i} = y_i h^{d_i} \pmod N$

  We remark that this structure is used by the ACJT group signature [2].
- `ID-LTRS.Sign`. For an event with event-ID $e \in \mathcal{E}$, compute $\tau_i := e^{d_i} \bmod N$ for all $i \in \Pi$ and $\tau_i := e^{t_i} \bmod N$ with $t_i \in_R \Lambda'$ for all $i \in [1, n] \backslash \Pi$. The algorithm is subsequently modified from `ID-TRS.Sign` to also prove that the τ_i's are correctly formed. Specifically, the algorithm now implements:

$$SPK_4 \left\{ (a_i, x_i, d_i)_{i=1}^n : \bigvee_{\mathcal{J} \in \wp_t([1,n])} \bigwedge_{i \in \mathcal{J}} y_i h^{d_i} \equiv a_i^{x_i} \wedge \tau_i \equiv e^{d_i} \wedge d_i \in \Lambda, x_i \in \Gamma \right\} (m) \qquad (4)$$

  which is instantiated as:

$$SPK_5 \left\{ (u_i, x_i, w_i)_{i=1}^n : \bigvee_{\mathcal{J} \in \wp_t([1,n])} \bigwedge_{i \in \mathcal{J}} \begin{array}{ccc} A_{i,1} \equiv g_1^{u_i} \;\wedge & A_{i,3} \equiv g_1^{x_i} g_3^{u_i} & \wedge \\ A_{i,1}^{x_i} \equiv g_1^{w_i} \;\wedge & A_{i,2}^{x_i} \equiv g_2^{w_i} y_i h^{d_i} & \wedge \\ \tau_i \equiv e^{d_i} \;\wedge \; x_i \in \Gamma \wedge d_i \in \Lambda & & \end{array} \right\} (m). \qquad (5)$$

  The actual steps implementing the SPK_5 above follow closely those implementing SPK_3 in `ID-TRS.Sign` and are thus not verbosely enumerated . Denote by σ_5 the signature output of SPK_5. Note that it includes $\tau_1, \cdots, \tau_n$.

  In addition, generate a signature σ_6 for the following SPK using the knowledge of x_i's for $i \in \Pi$ and t_i's for $i \in [1, n] \backslash \Pi$:

$$SPK_6 \left\{ (\alpha_i)_{i=1}^n : \bigwedge_{i=1}^{n} \tau_i \equiv e^{\alpha_i} \right\} (m). \qquad (6)$$

  The detailed implementation of the above SPK is given in Appendix B.

  Finally the signature is output as $\sigma := (\sigma_5, \sigma_6)$.
- `ID-LTRS.Verify`. Given a signature $\sigma = (\sigma_5, \sigma_6)$, verify the validity of σ_5 with respect to SPK_5 and that of σ_6 with respect to SPK_6. Again we omit the verification algorithm for SPK_5 as it can be adapted in a straightforward manner from `ID-TRS.Verify`. Verification for SPK_6 is given in Appendix B.
- `ID-LTRS.Link`. On input the list of system parameters param, an event-ID $e \in \mathcal{E}$, two group sizes $n_1, n_2 \in \mathbb{N}$ of length polynomial in the security parameter λ, two thresholds $t_1 \in [1, n_1]$ and $t_2 \in [1, n_2]$, two identity sets $\mathcal{Y}_j = \{\mathsf{ID}_i^{(j)} \in \{0,1\}^* | i \in [1, n_j]\}$ for $j = 1, 2$, two messages $m_1, m_2 \in \mathcal{M}$, and two signatures $\sigma_1, \sigma_2 \in \Psi$ such that $\mathsf{valid} \leftarrow \mathtt{Verify}(\mathsf{param}, e, n_j, t_j, \mathcal{Y}_j, m_j, \sigma_j)$ for $j = 1, 2$, the algorithm parses σ_1 for the tags $(\tau_1^{(1)}, \cdots, \tau_{n_1}^{(1)})$ and σ_2 for the tags $(\tau_1^{(2)}, \cdots, \tau_{n_2}^{(2)})$. If there exists a tag from the first set and a tag from the second set such that the two tags are equal in value, the algorithm outputs linked. Otherwise it returns $\mathsf{unlinked}$.

Correctness of our scheme is straightforward and we show its security in Appendix C.1.

6 Identity Escrow

As mentioned earlier, the anonymity provided by ring signatures can be undesirably strong in some situations. Authorities prefer providing only revocable anonymity to their users. Their ability of revocation serves as a mechanism that prevents them from being suffered from the presence of misbehaving users. Introducing a trusted authority who can reveal the true identity of the user under certain circumstances is formally known as identity escrow [43].

To add identity escrow to ring signature schemes, one could variably encrypt any information sufficient for identifying the signer, and then include in the signature the resulting ciphertext plus a proof that it is correctly formed. In fact, verifiable encryption [15, 19] has been frequently used (though sometimes implicitly) to achieve revocable anonymity. For instance, the generic constructions of group signatures [9, 12]. As a concrete example, in [2], part of the user's secret key [9] is ElGamal encrypted under the public key of an authority. The unforgeability of the signature scheme implies that valid signatures are actually proofs of the fact that encryption was done according to specification.

Our Construction. We use the same technique as in [2] to add identity escrow to the two schemes proposed above. The resulting schemes are virtually the same as their respective original schemes without identity escrow, except that in `Setup`, g_2 is not generated randomly. Instead it is generated in a way such that the revocation manager knows the discrete logarithm of g_2 in base g_1, i.e. he knows an integer s such that $g_2 \equiv g_1^s \pmod N$. Assume the revocation manager is trusted not to abuse his knowledge of s in the sense that he does not collude with any adversary and only uses s when trying to revoke the anonymity of a signature with eligible reasons, e.g. under court orders. Then the two schemes with identity escrow still enjoy all the security notions we proved for original schemes.

To see how the anonymity can be revoked, the revocation manager can compute from a signature a part of the secret key (a_i, x_i), namely a_i, of all participating users by computing $A_{i,2}/A_{i,1}^s \bmod N$ for all $i \in [1, n]$. The unforgeability of the signature scheme forces at least d pairs of $A_{i,1}$ and $A_{i,2}$ to be formed correctly. These pairs are exactly those belonging to the participating users. The remaining a_i could just be some random numbers. All n a_i's are passed to the key issuing manager, whom can then look up in his database the identity of the user possessing a_i as a part of his secret key, for each $i \in [1, n]$. In this way, the d actual signers can be identified.

The revocation manager cannot frame a user if he is required to prove (in zero-knowledge of s) the statement $g_2 \equiv g_1^s \wedge A_{i,2} \equiv a_i A_{i,1}^s$. The key issuing manager cannot frame a user as well if he is required to prove (in zero-knowledge of x_i) the statement $a_i^{x_i} \equiv y_i$, where $y_i = H_{id}(\mathsf{ID}_i)$.

7 Performance and Conclusion

The computation complexity and the signature size of our construction are both linear to the ring size. This is the major tradeoff of our schemes as they achieve different levels of anonymity. To improve their efficiency, especially on constructing an efficient ID-based linkable threshold ring signature scheme, will be our next research work.

In this paper, we proposed the first ID-based threshold ring signature construction that is *not* based on bilinear pairings. We formally proved the security of the construction under well-known mathematical assumptions in the RO model. Based on the construction, we then proposed the first ID-based linkable (threshold) ring signature scheme. We argued the security of all the constructions. Finally we showed how to add identity escrow to the two schemes. All the ID-based threshold ring signature schemes proposed in this paper form a suite of schemes applicable to many real world applications with varied anonymity requirements.

[9] Also known as the user's signing certificate in the context of group signatures.

Memorial

This paper is dedicated to the first author, Patrick P. Tsang, who was a PhD student in the Computer Science program at Dartmouth College, has passed away on October 27, 2009 as a victim to cancer. He was 28 years old.

References

1. M. Abe, M. Ohkubo, and K. Suzuki. 1-out-of-n signatures from a variety of keys. In *ASIACRYPT 2002*, volume 2501 of *Lecture Notes in Computer Science*, pages 415–432. Springer, 2002.
2. G. Ateniese, J. Camenisch, M. Joye, and G. Tsudik. A practical and provably secure coalition-resistant group signature scheme. In *CRYPTO 2000*, volume 1880 of *Lecture Notes in Computer Science*, pages 255–270. Springer, 2000.
3. M. Au, J. Chen, J. Liu, Y. Mu, D. Wong, and G. Yang. Malicious KGC attacks in certificateless cryptography. In *ASIACCS 2007*, pages 302–311. ACM Press, 2007.
4. M. H. Au, J. K. Liu, W. Susilo, and T. H. Yuen. Constant-size id-based linkable and revocable-iff-linked ring signature. In *INDOCRYPT*, volume 4329 of *Lecture Notes in Computer Science*, pages 364–378. Springer, 2006.
5. M. H. Au, J. K. Liu, T. H. Yuen, and D. S. Wong. Id-based ring signature scheme secure in the standard model. In *IWSEC*, volume 4266 of *Lecture Notes in Computer Science*, pages 1–16. Springer, 2006.
6. A. K. Awasthi and S. Lal. Id-based ring signature and proxy ring signature schemes from bilinear pairings. Cryptology ePrint Archive, Report 2004/184, 2004. http://eprint.iacr.org/.
7. A. K. Awasthi and S. Lal. Id-based ring signature and proxy ring signature schemes from bilinear pairings. *CoRR*, abs/cs/0504097, 2005.
8. N. Barić and B. Pfitzmann. Collision-free accumulators and fail-stop signature schemes without trees. In *EUROCRYPT 1997*, volume 1233 of *Lecture Notes in Computer Science*, pages 480–494, 1997.
9. M. Bellare, D. Micciancio, and B. Warinschi. Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In *EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 614–629. Springer, 2003.
10. M. Bellare, C. Namprempre, and G. Neven. Security proofs for identity-based identification and signature schemes. In *EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 268–286. Springer, 2004.
11. M. Bellare and P. Rogaway. Random oracles are practical: a paradigm for designing efficient protocols. In *Proc. of the 1st ACM conference on Computer and communications security*, pages 62–73. ACM Press, 1993.
12. M. Bellare, H. Shi, and C. Zhang. Foundations of group signatures: The case of dynamic groups. In *CT-RSA 2005*, volume 3376 of *Lecture Notes in Computer Science*, pages 136–153. Springer, 2005.
13. D. Boneh and M. K. Franklin. Identity-based encryption from the weil pairing. In *CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 213–229. Springer, 2001.
14. E. Bresson, J. Stern, and M. Szydlo. Threshold ring signatures and applications to ad-hoc groups. In *CRYPTO 2002*, volume 2442 of *Lecture Notes in Computer Science*, pages 465–480. Springer, 2002.
15. J. Camenisch and I. Damgård. Verifiable encryption, group encryption, and their applications to separable group signatures and signature sharing schemes. In *ASIACRYPT 2000*, volume 1976 of *Lecture Notes in Computer Science*, pages 331–345. Springer, 2000.
16. J. Camenisch and A. Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In *EUROCRYPT 2001*, volume 2045 of *Lecture Notes in Computer Science*, pages 93–118. Springer, 2001.
17. J. Camenisch and A. Lysyanskaya. Dynamic accumulators and application to efficient revocation of anonymous credentials. In *CRYPTO 2002*, volume 2442 of *Lecture Notes in Computer Science*, pages 61–76, 2002.
18. J. Camenisch and A. Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In *CRYPTO 2004*, volume 3152, pages 56–72, 2004.
19. J. Camenisch and V. Shoup. Practical verifiable encryption and decryption of discrete logarithms. In *CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 126–144. Springer, 2003.
20. J. Camenisch and M. Stadler. Efficient group signature schemes for large groups (extended abstract). In *CRYPTO 1997*, volume 1294 of *Lecture Notes in Computer Science*, pages 410–424. Springer, 1997.
21. N. Chandran, J. Groth, and A. Sahai. Ring signatures of sub-linear size without random oracles. In *ICALP*, volume 4596 of *Lecture Notes in Computer Science*, pages 423–434. Springer, 2007.

22. D. Chaum and E. van Heyst. Group signatures. In *EUROCRYPT 1991*, volume 547, pages 257–265, 1991.
23. H.-Y. Chien. Highly efficient id-based ring signature from pairings. In *APSCC*, pages 829–834, 2008.
24. S. S. M. Chow, L. C. K. Hui, and S.-M. Yiu. Identity based threshold ring signature. In *ICISC 2004*, volume 3506 of *Lecture Notes in Computer Science*, pages 218–232. Springer, 2004.
25. S. S. M. Chow, J. K. Liu, and D. S. Wong. Robust receipt-free election system with ballot secrecy and verifiability. In *NDSS*. The Internet Society, 2008.
26. S. S. M. Chow, R. W. C. Lui, L. C. K. Hui, and S.-M. Yiu. Identity based ring signature: Why, how and what next. In *EuroPKI*, volume 3545 of *Lecture Notes in Computer Science*, pages 144–161. Springer, 2005.
27. S. S. M. Chow, W. Susilo, and T. H. Yuen. Escrowed linkability of ring signatures and its applications. In *VIETCRYPT*, volume 4341 of *Lecture Notes in Computer Science*, pages 175–192. Springer, 2006.
28. S. S. M. Chow, S.-M. Yiu, and L. C. K. Hui. Efficient identity based ring signature. In *ACNS 2005*, volume 3531 of *Lecture Notes in Computer Science*, pages 499–512. Springer, 2005.
29. Y.-F. Chung, Z. Y. Wu, F. Lai, and T.-S. Chen. A novel id-based threshold ring signature scheme competent for anonymity and anti-forgery. In *CIS*, volume 4456 of *Lecture Notes in Computer Science*, pages 502–512. Springer, 2006.
30. R. Cramer, I. Damgård, and B. Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In *CRYPTO 1994*, volume 839 of *Lecture Notes in Computer Science*, pages 174–187. Springer, 1994.
31. Y. Dodis, A. Kiayias, A. Nicolosi, and V. Shoup. Anonymous identification in ad hoc groups. In *EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 609–626. Springer, 2004.
32. Z. Dong, H. Zheng, K. Chen, and W. Kou. ID-based proxy blind signature. In *AINA (2)*, pages 380–383, 2004.
33. A. L. Ferrara, M. Green, S. Hohenberger, and M. Ø. Pedersen. Practical short signature batch verification. In *CT-RSA*, volume 5473 of *Lecture Notes in Computer Science*, pages 309–324. Springer, 2009. Full version appeared in http://eprint.iacr.org/2008/015.
34. A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *CRYPTO 1986*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194. Springer, 1986.
35. S. Fischer-Hübner. *IT-Security and Privacy - Design and Use of Privacy-Enhancing Security Mechanisms*, volume 1958 of *Lecture Notes in Computer Science*. Springer, 2001.
36. E. Fujisaki and T. Okamoto. Statistical zero knowledge protocols to prove modular polynomial relations. In *CRYPTO 1997*, volume 1294 of *Lecture Notes in Computer Science*, pages 16–30, 1997.
37. M. Girault. Self-certified public keys. In *EUROCRYPT 1991*, pages 490–497. Springer, 1991. Lecture Notes in Computer Science 547.
38. 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.
39. J. Han, Q. Xu, and G. Chen. Efficient id-based threshold ring signature scheme. In *EUC (2)*, pages 437–442. IEEE Computer Society, 2008.
40. J. Herranz. Identity-based ring signatures from RSA. *Theor. Comput. Sci.*, 389(1-2):100–117, 2007.
41. K. Itakura and K. Nakamura. A public key cryptosystem suitable for digital multisignatures. *NEC Research & Development*, 71:1–8, 1983.
42. I. R. JEONG, J. O. KWON, and D. H. LEE. Analysis of revocable-iff-linked ring signature scheme. *IEICE Transactions*, 92-A(1):322–325, 2009.
43. J. Kilian and E. Petrank. Identity escrow. In *CRYPTO 1998*, volume 1462 of *Lecture Notes in Computer Science*, pages 169–185. Springer, 1998.
44. X. Li and K. Chen. Identity based proxy-signcryption scheme from pairings. In *IEEE SCC*, pages 494–497, 2004.
45. J. K. Liu, V. K. Wei, and D. S. Wong. A separable threshold ring signature scheme. In *ICISC 2003*, volume 2971 of *Lecture Notes in Computer Science*, pages 12–26. Springer, 2003.
46. J. K. Liu, V. K. Wei, and D. S. Wong. Linkable spontaneous anonymous group signature for ad hoc groups (extended abstract). In *ACISP 2004*, volume 3108 of *Lecture Notes in Computer Science*, pages 325–335. Springer, 2004.
47. J. K. Liu and D. S. Wong. On the security models of (threshold) ring signature schemes. In *ICISC 2004*, Lecture Notes in Computer Science. Springer, 2005.
48. S. Micali, K. Ohta, and L. Reyzin. Accountable-subgroup multisignatures: extended abstract. In *CCS '01: Proc. of the 8th ACM conf. on Computer and Communications Security*, pages 245–254. ACM Press, 2001.

49. T. Nakanishi, T. Fujiwara, and H. Watanabe. A linkable group signature and its application to secret voting. *Trans. of Information Processing Society of Japan*, 40(7):3085–3096, 1999.
50. L. Nguyen. Accumulators from bilinear pairings and applications. In *CT-RSA 2005*, volume 3376 of *Lecture Notes in Computer Science*, pages 275–292. Springer, 2005.
51. D. Pointcheval and J. Stern. Security proofs for signature schemes. In *EUROCRYPT 1996*, volume 1070 of *Lecture Notes in Computer Science*, pages 387–398, 1996.
52. D. Pointcheval and J. Stern. Security arguments for digital signatures and blind signatures. *J. Cryptology*, 13(3):361–396, 2000.
53. R. L. Rivest, A. Shamir, and Y. Tauman. How to leak a secret. In *ASIACRYPT 2001*, volume 2248 of *Lecture Notes in Computer Science*, pages 552–565. Springer, 2001.
54. H. Shacham and B. Waters. Efficient ring signatures without random oracles. In *Public Key Cryptography*, volume 4450 of *Lecture Notes in Computer Science*, pages 166–180. Springer, 2007.
55. A. Shamir. Identity-based cryptosystems and signature schemes. In *CRYPTO 1984*, volume 196 of *Lecture Notes in Computer Science*, pages 47–53. Springer, 1984.
56. P. P. Tsang and V. K. Wei. Short linkable ring signatures for e-voting, e-cash and attestation. In *ISPEC 2005*, volume 3439 of *Lecture Notes in Computer Science*, pages 48–60. Springer, 2005.
57. P. P. Tsang, V. K. Wei, T. K. Chan, M. H. Au, J. K. Liu, and D. S. Wong. Separable linkable threshold ring signatures. In *INDOCRYPT 2004*, volume 3348 of *Lecture Notes in Computer Science*, pages 384–398. Springer, 2004.
58. S. D. Warren and L. D. Brandeis. The right to privacy. *Harvard Law Review*, IV(5):193–220, 1890.
59. A. F. Westin. Privacy and freedom. Atheneum, 1970.
60. 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 *ICICS 2003*, volume 2836 of *Lecture Notes in Computer Science*, pages 34–46. Springer, 2003.
61. J. Xu, Z. Zhang, and D. Feng. Id-based proxy signature using bilinear pairings. Cryptology ePrint Archive, Report 2004/206, 2004. http://eprint.iacr.org/.
62. F. Zhang and K. Kim. Id-based blind signature and ring signature from pairings. In *ASIACRYPT 2002*, volume 2501 of *Lecture Notes in Computer Science*, pages 533–547. Springer, 2002.
63. F. Zhang and K. Kim. Efficient id-based blind signature and proxy signature from bilinear pairings. In *ACISP*, volume 2727 of *Lecture Notes in Computer Science*, pages 312–323. Springer, 2003.
64. J. Zhang. An efficient identity-based ring signature scheme and its extension. In *ICCSA (2)*, volume 4706 of *Lecture Notes in Computer Science*, pages 63–74. Springer, 2007.

A Analysis of the Proof of [39]

We point out a flaw in the security proof of [39]. While in the security model, the attacker is allowed to query secret key of any identity of his choice (private key query), However, in the security proof of anonymity and unforgeability, no description is given on how this query is handled.

Indeed, this flaw in the security proof leads to the following theoretical error. Recalled that the secret key of an identity ID is $H(ID)^a$, where H is some collision-resistant hash function and a is the master secret key of the PKG. This is in fact a very common key structure in identity-based encryption or signature [13], and is well-known to be secure under the CDH assumption in the random oracle model.

However, in the standard model where the hash function is only required to be collision-resistant, it is entirely possible for an attacker to obtain the secret key of identity ID_1 by issuing private key queries on a set of identities $\{ID_2, \ldots, ID_k\}$ such that $H(ID_1) = \prod H(ID_i)$.

Thus, it is very doubtful, to say the least, that [39] can be proven secure in the standard model when H is only modelled as collision-resistant hash function. The claim that the scheme in [39] is secure *in the standard model* is not accurate. One could, however, possibly simulates the ID query in the random oracle model.

B Implementations of SPK_6

SPK_6. To sign a signature for SPK_6, do the following:

1. *(Commitment.)* Pick $\rho_i \in_R \pm\{0,1\}^{\epsilon(\lambda_2+\kappa)}$ and compute $T_i := g^{\rho_i} \bmod N$ for all $i \in [1, n]$.

2. *(Challenge.)* Compute $c := H_{sig}(\mathsf{param}, n, g, (\tau_1, T_1)_{i=1}^n, m)$.
3. *(Response.)* Compute $s_i := \rho_i - cx_i$ for all $i \in \mathcal{I}$ and $s_i := \rho_i - ct_i$ for all $i \in [1,n]\backslash \mathit{\Pi}$.

The signature for SPK_6 is thus $\sigma_6 := (c, s_1, \ldots, s_n)$.

Verification for $\sigma_6 = (c, s_1, \ldots, s_n)$ is done by first computing $T_i' := g^{s_i} \tau_i^c \bmod N$ for all $i \in [1,n]$ and then checking if $s_i \overset{?}{\in} \{0,1\}^{\epsilon(\lambda_2+\kappa)+1}$ for all $i \in [1,n]$, and $c \overset{?}{=} H_{sig}(\mathsf{param}, n, g, (\tau_1, T_1')_{i=1}^n, m)$.

C Security Proofs

Theorem 1 (Unforgeability). *Under the condition that both λ and κ are sufficiently large, the ID-TRS scheme proposed in Sec. 4 is existential unforgeable against chosen-message-and-identity attacks (EUF-IDTR-CMIA secure) under the Strong RSA Assumption, in the Random Oracle Model.*

Proof. Suppose the challenger $\mathcal{C}$ receives a random instance (Y, N) of the Strong RSA problem, where N is a product of two equal-length safe primes and $Y \in_R QR(N)$, and is to compute x, e such that $x^e = Y \bmod N$. $\mathcal{C}$ runs $\mathcal{A}$ and acts as $\mathcal{A}$'s challenger in Game Unforgeability. During the game, $\mathcal{C}$ simulates answers to H_{sig}, H_{id} and Key queries made by $\mathcal{A}$. These answers are randomly generated accordingly with consistency maintained and collision avoided. To do so, $\mathcal{C}$ keeps track of all the previous queries and answers. Due to the random oracle assumption, we assume that $\mathcal{A}$ has queried for $H_{id}(\mathsf{ID})$ before ID is used. In the game, $\mathcal{C}$ randomly picks $g_1, g_2, g_3 \in QR(N)$ such that they are generators of $QR(N)$ and chooses $\gamma_1, \gamma_2 \in \mathbb{N}$ and $1 < \epsilon \in \mathbb{R}$ accordingly. $\mathcal{C}$ gives $\mathcal{A}$ the list param of system parameters. In the following, we give more details on how the H_{id} queries and Signature queries are simulated.

H_{id} queries: Besides maintaining consistency and avoiding collision, for each H_{id} query, $\mathcal{C}$ randomly generates a prime x and a number a of suitable range, and returns $a^x \bmod N$. There is one exception: in the game, $\mathcal{C}$ also randomly chooses one of the H_{id} queries and sets the answer as $H_{id}(\mathsf{ID}^*) = Y$, where ID^* is the value of the query. Since Y is an random instance of the strong RSA problem, it does not affect the randomness of simulated H_{id}. However, a Key query on identity ID^* will make $\mathcal{C}$ fail.

Signature queries: $\mathcal{A}$ chooses a group $\{\mathsf{ID}_i\}_{i \in [1,n]}$ of n identities, a threshold value t where $t \in [1,n]$, a set $\mathcal{S} \in \wp_t([1,n])$ and a message $m \in \{0,1\}^*$, and asks for a signature. If $\mathsf{ID}^* \notin \mathcal{S}$, $\mathcal{C}$ is in possession of all secret keys correspond to identities in $\mathcal{S}$ and can simulate a signature accordingly. Otherwise, $\mathcal{C}$ generates the signature by following the steps below. Without loss of generality, we assume $\mathcal{S} = [1,t]$ and $\mathsf{ID}_t = \mathsf{ID}^*$.

1. *(Auxiliary commitment.)* For all $i \in [1, t-1]$, pick $u_i \in_R \pm\{0,1\}^{2\lambda}$ and compute $w_i := u_i x_i$. Compute in modulo N: $A_{i,1} := g_1^{u_i}$, $A_{i,2} := a_i g_2^{u_i}$, $A_{i,3} := g_1^{x_i} g_3^{u_i}$. For all $i \in [t,n]$, randomly pick $A_{i,1}, A_{i,2}, A_{i,3} \in_R QR(N)$.
2. *(Commitment.)* For all $i \in [1, t-1]$, pick $r_{i,x} \in_R \pm\{0,1\}^{\epsilon(\gamma_2+\kappa)}$, $r_{i,u} \in_R \pm\{0,1\}^{\epsilon(2\lambda+\kappa)}$, $r_{i,w} \in_R \pm\{0,1\}^{\epsilon(\gamma_1+2\lambda+\kappa+1)}$. Compute in modulo N:

   $$T_{i,1} := g_1^{r_{i,u}},\ T_{i,2} := g_1^{r_{i,x}} g_3^{r_{i,u}},\ T_{i,3} := A_{i,1}^{r_{i,x}} g_1^{-r_{i,w}},\ T_{i,4} := A_{i,2}^{r_{i,x}} g_2^{-r_{i,w}}.$$

   For all $i \in [t,n]$, pick $c_i \in_R \{0,1\}^{\kappa}$, $s_{i,x} \in_R \pm\{0,1\}^{\epsilon(\gamma_2+\kappa)}$, $s_{i,u} \in_R \pm\{0,1\}^{\epsilon(2\lambda+\kappa)}$, $s_{i,w} \in_R \pm\{0,1\}^{\epsilon(\gamma_1+2\lambda+\kappa+1)}$. Compute in modulo N:

   $$T_{i,1} := g_1^{s_{i,u}} A_{i,1}^{c_i},\ T_{i,2} := g_1^{s_{i,x} - c_i 2^{\gamma_1}} g_3^{s_{i,u}} A_{i,3}^{c_i},$$

   $$T_{i,3} := A_{i,1}^{s_{i,x} - c_i 2^{\gamma_1}} g_1^{-s_{i,w}},\ T_{i,4} := A_{i,2}^{s_{i,x} - c_i 2^{\gamma_1}} g_2^{-s_{i,w}} y_i^{c_i}.$$

3. *(Challenge.)* Generate a polynomial f over $GF(2^{\kappa})$ of degree at most $(n-t)$ such that and $c_i = f(i)$ for all $i \in [t,n]$ and set $H_{sig}(\mathsf{param}, n, t, (y_i, A_{i,1}, A_{i,2}, A_{i,3})_{i=1}^n, (T_{i,1}, \cdots, T_{i,4})_{i=1}^n, m) = f(0)$.

4. *(Response.)* For all $i \in [1, t-1]$, compute $c_i := f(i)$, and compute in $\mathbb{Z}$:
$$s_{i,u} := r_{i,u} - c_i u_i,\ s_{i,x} := r_{i,x} - c_i(x_i - 2^{\gamma_1}),\ s_{i,w} := r_{i,w} - c_i w_i.$$
5. *(Signature and Output.)* Set $\sigma := ((A_{i,1}, A_{i,2}, A_{i,3})_{i=1}^{n},\ f,\ (s_{i,u}, s_{i,x}, s_{i,w})_{i=1}^{n})$.

When $\mathcal{A}$ outputs a forged ID-based (t, n)-threshold ring signature for a group $\mathcal{Y}$ such that $\mathsf{ID}^* \in \mathcal{Y}$, and $\mathcal{A}$ only issues up to $t-1$ key queries corresponding the identities in $\mathcal{Y} \setminus \{\mathsf{ID}^*\}$, the following will be carried out by $\mathcal{C}$ for solving the Strong RSA problem. Otherwise, $\mathcal{C}$ fails.

It follows from the forking lemma [52] that if $\mathcal{A}$ is a sufficiently efficient forger in the above interaction, we can construct a Las Vegas machine $\mathcal{A}'$ that outputs two signatures:

$$\sigma = ((A_{i,1}, A_{i,2}, A_{i,3})_{i=1}^{n},\ f,\ (s_{i,u}, s_{i,x}, s_{i,w})_{i=1}^{n}),$$
$$\sigma' = ((A_{i,1}, A_{i,2}, A_{i,3})_{i=1}^{n},\ f',\ (s'_{i,u}, s'_{i,x}, s'_{i,w})_{i=1}^{n}).$$

$\mathcal{C}$ achieves this result by keeping all the random tapes in two invocations of $\mathcal{A}$ the same except c_0 returned by H_{sig} of the forged message.

Next we consider the probability that ID^* is the chosen target of forgery. Let π be the index of ID^* in $\mathcal{Y}$. Since $f(0) \neq f'(0)$, and the degree of f and f' is at most $n-t$, there are at least t values $k_1, k_2, \cdots, k_t$ such that $f(k_i) \neq f'(k_i)$. With probability at least $1/n$, $k_i = \pi$.

Given σ and σ', $\mathcal{C}$ solves the Strong RSA problem as follows. Denote $f(\pi)$ and $f'(\pi)$ by c_π, c'_π. For clarity, we drop the subscript π, thus A_1 denotes $A_{\pi,1}$, s_u denotes $s_{\pi,u}$, etc. Since ${A_1}^{c}{g_1}^{s_u} = {A_1}^{c'}{g_1}^{s'_u}$, it follows that ${g_1}^{s_u - s'_u} = {A_1}^{c'-c}$. Let $d_u = \gcd(s_u - s'_u, c' - c)$, that is, there exists α_u, β_u such that $\alpha_u(s_u - s'_u) + \beta_u(c' - c) = d_u$. Hence,

$$g_1 = g_1^{\frac{\alpha_u(s_u - s'_u) + \beta_u(c'-c)}{d_u}} = (A_1^{\alpha_u}{g_1}^{\beta_u})^{\frac{c'-c}{d_u}}$$

Under the strong RSA assumption, $c' - c = d_u$ (otherwise the $\frac{c'-c}{d_u}$-th root of g_1 is computed). This implies $(s_u - s'_u) = \hat{u}(c' - c)$ such that ${g_1}^{\hat{u}} = A_1$. Next consider ${A_3}^{c}{g_1}^{s_x - c2^{\gamma_1}}{g_3}^{s_u} = {A_3}^{c'}{g_1}^{s'_x - c'2^{\gamma_1}}{g_3}^{s'_u}$, it follows that ${g_1}^{s_x - s'_x}{g_3}^{s_u - s'_u} = (A_3{g_1}^{-2^{\gamma_1}})^{c'-c}$. By $(s_u - s'_u) = \hat{u}(c' - c)$, $(\frac{A_3}{{g_1}^{2^{\gamma_1}} g_3^{\hat{u}}})^{c'-c} = {g_1}^{s_x - s'_x}$. Under the strong RSA assumption and similar argument as above, we have $s_x - s'_x = \tilde{x}(c' - c)$ such that $(\frac{A_3}{{g_1}^{2^{\gamma_1}} g_3^{\hat{u}}}) = {g_1}^{\tilde{x}}$. That is, $A_3 = g_3^{\hat{u}} g_1^{(\tilde{x} + 2^{\gamma_1})}$. Denote $\hat{x} = \tilde{x} + 2^{\gamma_1}$. Then consider ${A_1}^{(s_x - c2^{\gamma_1})}{g_1}^{-s_w} = {A_1}^{(s'_x - c'2^{\gamma_1})}{g_1}^{-s'_w}$, it follows that ${A_1}^{s_x - s'_x}{A_1}^{(c'-c)2^{\gamma_1}} = {g_1}^{s_w - s'_w}$. By $s_x - s'_x = \tilde{x}(c' - c)$, $({A_1}^{\hat{x}})^{c'-c} = {g_1}^{s_w - s'_w}$. Under the strong RSA assumption and similar argument as above, we have $s_w - s'_w = \hat{w}(c' - c)$ such that $A_1^{\hat{x}} = g_1^{\hat{w}}$. This implies ${g_1}^{\hat{u}\hat{x}} = g_1^{\hat{w}}$ and $\hat{w} = \hat{u}\hat{x}$. Finally, consider ${A_2}^{(s_x - c2^{\gamma_1})}{g_2}^{-s_w}y^{c} = {A_2}^{(s'_x - c'2^{\gamma_1})}{g_2}^{-s'_w}y^{c'}$, it follows that ${A_2}^{s_x - s'_x}{A_2}^{(c'-c)2^{\gamma_1}}{g_2}^{s'_w - s_w} = y^{c'-c}$. By $s_x - s'_x = \tilde{x}(c' - c)$ and $s_w - s'_w = \hat{w}(c' - c)$, we have $({A_2}^{\hat{x}}{g_2}^{-\hat{w}})^{c'-c} = y^{c'-c}$. It follows that $(\frac{A_2}{{g_2}^{\hat{u}}})^{\hat{x}} = y$.

$\mathcal{C}$ returns $(\frac{A_2}{{g_2}^{\hat{u}}},\ \hat{x})$ as the solution to the Strong RSA problem.

The success probability of $\mathcal{C}$ is computed as follows. For $\mathcal{C}$ to succeed, key query on ID^* should never be issued (i.e. ID^* is not corrupted) and the corresponding probability is $\frac{q_{H_{id}} - q_{Key}}{q_{H_{id}}}$, where $q_{H_{id}}$ and q_{Key} are the number of H_{id} queries and Key queries, respectively. Suppose n_a identities in the group $\mathcal{Y}$ of the forged signatures are corrupted using key queries. Here $0 \leq n_a \leq t-1$. With probability $\frac{n - n_a}{q_{H_{id}} - q_{Key}}$, ID^* is in $\mathcal{Y}$, given that ID^* is not corrupted. $\mathcal{C}$ can compute at least t out of n secret keys in the group since there are at least t values $k_1, k_2, \cdots, k_t$ such that $f(k_i) \neq f'(k_i)$. Suppose n_b secret keys corresponding to uncorrupted identities in $\mathcal{Y}$ are computed. Here $1 \leq n_b \leq t$. With probability $\frac{n_b}{n - n_a}$, the secret key of ID^* is computed. Combining all the events, the success probability of $\mathcal{C}$ is given by $\frac{q_{H_{id}} - q_{Key}}{q_{H_{id}}} \frac{n - n_a}{q_{H_{id}} - q_{Key}} \frac{n_b}{n - n_a}$ which is at least $\frac{1}{q_{H_{id}}}$. □

Theorem 2 (Anonymity). *Under the condition that both λ and κ are sufficiently large, the ID-TRS scheme proposed in Sec. 4 is signer indistinguishable against adaptive chosen-message-and-identity attacks (IND-IDTR-CMIA secure) under the DDH Assumption in the random oracle model.*

Proof. Suppose the challenger $\mathcal{C}$ receives a random instance of the DDH problem in the group $QR(N)$: $(g, g^\alpha, g^\beta, g^\gamma)$ and is to decide if $\gamma = \alpha\beta \bmod ord(g)$. $\mathcal{C}$ runs $\mathcal{A}$ and acts as $\mathcal{A}$'s challenger in Game Anonymity. $\mathcal{C}$ sets $g_1 = g$, $g_2 = g^k$ and $g_3 = g^\beta$ where k is randomly generated. It chooses $\gamma_1, \gamma_2 \in \mathbb{N}$ and $1 < \epsilon \in \mathbb{R}$ accordingly, and gives $\mathcal{A}$ the list param of system parameters. During the game, $\mathcal{C}$ answers $\mathcal{A}$'s queries similar to that described in the simulation of Game Unforgeability above. In particular, consistency should be maintained and collision should be avoided. Similarly, we assume that $\mathcal{A}$ has asked for $H_{id}(\mathsf{ID})$ before ID is used.

Challenge Phase: In the challenge phase of Game Anonymity, $\mathcal{A}$ gives $\mathcal{C}$ a group size n, a threshold t, a set $\{\mathsf{ID}_i\}_{i\in[1,n]}$ of identities and a message m. $\mathcal{C}$ picks randomly $\mathit{\Pi} \in_R \wp_t([1, n])$. Without loss of generality, we assume $\mathit{\Pi} = [1, t]$ and $\mathcal{C}$ computes σ as follows.

1. *(Auxiliary commitment.)* For all $i \in [1, t-1]$, pick $u_i \in_R \pm\{0,1\}^{2\lambda}$ and compute $w_i := u_i x_i$. Compute in modulo N: $A_{i,1} := g_1^{u_i}$, $A_{i,2} := a_i g_2^{u_i}$, $A_{i,3} := g_1^{x_i} g_3^{u_i}$. For $i = t$, set $A_{i,1} = g^\alpha$, $A_{i,2} = a_i (g^\alpha)^k$, $A_{i,3} = {g_1}^{x_i} g^\gamma$. For all $i \in [t+1, n]$, pick $A_{i,1}, A_{i,2}, A_{i,3} \in_R QR(N)$.
2. *(Commitment.)* For all $i \in [t-1]$, pick $r_{i,x} \in_R \pm\{0,1\}^{\epsilon(\gamma_2+\kappa)}$, $r_{i,u} \in_R \pm\{0,1\}^{\epsilon(2\lambda+\kappa)}$, $r_{i,w} \in_R \pm\{0,1\}^{\epsilon(\gamma_1+2\lambda+\kappa+1)}$. Compute in modulo N:

   $$T_{i,1} := g_1^{r_{i,u}},\ T_{i,2} := g_1^{r_{i,x}} g_3^{r_{i,u}},\ T_{i,3} := A_{i,1}^{r_{i,x}} g_1^{-r_{i,w}},\ T_{i,4} := A_{i,2}^{r_{i,x}} g_2^{-r_{i,w}}.$$

   For all $i \in [t, n]$, pick $c_i \in_R \{0,1\}^\kappa$, $s_{i,x} \in_R \pm\{0,1\}^{\epsilon(\gamma_2+\kappa)}$, $s_{i,u} \in_R \pm\{0,1\}^{\epsilon(2\lambda+\kappa)}$, $s_{i,w} \in_R \pm\{0,1\}^{\epsilon(\gamma_1+2\lambda+\kappa+1)}$. Compute in modulo N:

   $$T_{i,1} := g_1^{s_{i,u}} A_{i,1}^{c_i},\ T_{i,2} := g_1^{s_{i,x}-c_i 2^{\gamma_1}} g_3^{s_{i,u}} A_{i,3}^{c_i},$$

   $$T_{i,3} := A_{i,1}^{s_{i,x}-c_i 2^{\gamma_1}} g_1^{-s_{i,w}},\ T_{i,4} := A_{i,2}^{s_{i,x}-c_i 2^{\gamma_1}} g_2^{-s_{i,w}} y_i^{c_i}.$$

3. *(Challenge.)* Generate a polynomial f over $GF(2^\kappa)$ of degree at most $(n-t)$ such that and $c_i = f(i)$ for all $i \in [t, n]$ and set $H_{sig}(\mathsf{param}, n, t, (y_i, A_{i,1}, A_{i,2},\ A_{i,3})_{i=1}^n,\ (T_{i,1}, \cdots, T_{i,4})_{i=1}^n, m) = f(0)$.
4. *(Response.)* For all $i \in [1, t-1]$, compute $c_i := f(i)$, and compute in $\mathbb{Z}$:

   $$s_{i,u} := r_{i,u} - c_i u_i,\ s_{i,x} := r_{i,x} - c_i(x_i - 2^{\gamma_1}),\ s_{i,w} := r_{i,w} - c_i w_i.$$

5. *(Signature and Output.)* Set $\sigma := ((A_{i,1}, A_{i,2}, A_{i,3})_{i=1}^n,\ f,\ (s_{i,u}, s_{i,x}, s_{i,w})_{i=1}^n)$.

When $\mathcal{A}$ outputs an index $\hat{\pi}$, $\mathcal{C}$ returns that $(g, g^\alpha, g^\beta, g^\gamma)$ is a valid DDH-tuple if $\hat{\pi} = t$. Otherwise, with half of the chances, $\mathcal{C}$ returns that it is a valid DDH-tuple, and with the other half, $\mathcal{C}$ returns that it is not a DDH-tuple.

Now we evaluate the winning probability of $\mathcal{C}$. Suppose the winning probability of $\mathcal{A}$ in a real Game Anonymity is $t/n + \epsilon_{\mathcal{A}}$ for some non-negligible $\epsilon_{\mathcal{A}}$. There are three cases that $\mathcal{C}$ will win. Case 1: $\mathcal{A}$ outputs $\hat{\pi} = t$ and the challenge is a valid DDH-tuple. Case 2: $\mathcal{A}$ outputs $\hat{\pi} \neq t$ and $\mathcal{C}$'s wild guess is correct. Since half of the chances, the challenge is a valid DDH-tuple, the probability that $\mathcal{A}$ outputs $\hat{\pi} \in [1, t]$ given that the challenge is a valid DDH-tuple is $\epsilon_{\mathcal{A}}$. As the value of t is also randomly chosen, the probability of case 1 is $1/2n + \epsilon_{\mathcal{A}}/2t$. For case 2, there are two sub-cases. In the first sub-case, the challenge is a valid DDH-tuple. Since $\mathcal{C}$ simply makes wild guess in this sub-case, the probability of winning for $\mathcal{C}$ in this sub-case is therefore $\frac{1}{4}(1 - (\frac{1}{n} + \frac{\epsilon_{\mathcal{A}}}{t}))$. The second sub-case is when the challenge is not a DDH-tuple. From the steps of simulating signature σ above, we can see that $(A_{d,1}, A_{d,2}, A_{d,3})$ has no difference from $(A_{i,1}, A_{i,2}, A_{i,3})$ for $i \in [t+1, n]$, i.e. same as those non-signers. Hence the probability of the second sub-case is equal to one minus the probability that $\hat{\pi} = t$ and the challenge is not a DDH-tuple. The probability of $\hat{\pi} = t$ given that the challenge is not a DDH-tuple is $\psi = (1 - (t/n + \epsilon_{\mathcal{A}}))/(n - t + 1)$. Hence the probability of winning for $\mathcal{C}$ in the second sub-case is $\frac{1}{4}(1 - \psi) = \frac{1}{4} - \frac{1-t/n-\epsilon_{\mathcal{A}}}{4(n-t+1)}$. Combining all cases, we have the winning probability of $\mathcal{C}$ to be at least $\frac{1}{2} + \frac{\epsilon_{\mathcal{A}}}{4t}$. □

C.1 Security Arguments of ID-LTRS

Unforgeability: it can be proved in a similar manner as in the case of ID-TRS. The signature using a random number as the tag (i.e., using a random number instead of e^{d_i}) can still be simulated using standard techniques. Distinguishing a random number from a correctly formed tag require solving the DDH problem.

Anonymity: a signature for ID-LTRS is different from a signature for ID-TRS as the former includes tags. The same signer will always produce the same tag. If the signer signs only once, distinguishing the actual signer solves a DDH instance of (g_1,e,${g_1}^{d_i}$, e^{d_i}). Thus, signers who signed only once won't reveal their identity under the DDH assumption.

Linkability: due to the soundness of the SPK, a signer is forced to use a correct tag for yielding a valid a signature. If an adversary can produce two distinct tag using one secret key, it is able to compute $H(ID) = {a_1}^{e_1} h^{-d_1} = {a_2}^{e_2} h^{-d_2}$ for some distinct d_1, d_2. With this, it is easy to set up a simulator to solve the Strong RSA problem and thus linkability is ensured under the Strong RSA assumption.

Non-slanderability: in order to slander, an adversary must produce a valid signature with a same tag of the person-to-be-slandered. Due to the soundness of SPK, the adversary must know the secret key of that person.

We outline how to simulate the key queries in the proofs of ID-LTRS.

Given a random instance (Y,N) of the strong RSA problem, randomly chooses $x_k \in_R \Gamma'$ for $k = [1, q_k] \setminus \{j\}$ for some $j \in [1, q_k]$, where q_k is the number of key queries. Also chooses $d_k \in_R \Lambda'$ for $k = [1, q_k]$.

The public key h is set to be $Y^{\Pi x_k}$. For the ith, $i \neq j$ key query, set $H(\mathsf{ID}_i) = h^{r_i}$ for $r_i \in_R \Lambda'$. Upon receiving C_1 , perform a rewind simulation and obtain $\tilde{d_i}, \tilde{r_i}$. Choose α, β such that $2^{\lambda_1} + (\alpha \tilde{d_i} + \beta \bmod 2^{\lambda_2}) = d_i$. Compute $A_i = Y^{(d_i + r_i) \Pi_{k \neq i} d_i}$. The secret key is (A_i, x_i).

For the jth query, set $H(\mathsf{ID}_j) = A_j^{x_j} / h^{d_i}$ for some $A_j = h^{r_j}$ where $r_j \in_R \Lambda'$. The secret key is (A_j, x_j).

In fact, for fixed ID_i, it is possible to simulate the key query and generate different secret keys using different C_1 as follow. $H(\mathsf{ID}_i)$ is of the form h^t where $t = r_i$ or $r_j x_j - d_i$. Additional secret keys on ID_i can be generated by unused x_k as $(A_i = Y^{(t + r_i) \Pi_{l \neq k} x_l}, x_k)$.

However, from practical point of view, a PKG should not allow users to obtain different secret keys for the same ID_i.