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

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

[1] Department of Information Engineering
The Chinese University of Hong Kong
Shatin, Hong Kong
{mhau3,pktsang3}@ie.cuhk.edu.hk
[2] Department of Computer Science
University of Bristol
Bristol, U.K.
liu@cs.bris.ac.uk
[3] Department of Computer Science
The 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 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 proposed two additional variants of our basic construction. These two variants achieve different levels of signer anonymity. In addition, one of them is also the first ID-based threshold 'linkable' ring signature. Finally we show how to add identity escrow to all the three schemes. Due to their different levels of signer anonymity supported, the schemes proposed in this paper form a suite of ID-based threshold ring signature schemes which can be applicable to many real-world applications with varied anonymity requirements.

Keywords: ID-Based Cryptography, Ring Signature, Anonymity, Linkability

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 [41], or the right to determine the amount of personal information which should be available to others [43]. Privacy is important for many reasons, such as impersonation and fraud. As more information about us is collected, correlated, and sold, it becomes easier for criminals to commit fraud. But privacy is more than that, it also concerns the secrecy of websites we visit, the content of our messages, the candidate we vote for, etc.

Anonymity is one important form of privacy protection. In practice, anonymity diversifies into various forms with different levels of anonymity. This is mainly due to two reasons. First, it gets harder when one strives for a higher level of anonymity. Technology required for privacy-preserving applications is often either not available, or not efficient enough. In all cases, preserving privacy comes with an extra cost and efficiency overhead. Consequently, the best is to achieve a level of anonymity just enough for an application. For example, look at how anonymous remailers have

evolved over time – from type 0 to type I to type II[4], every successor provides a higher level of anonymity, at the cost of lower efficiency and more resource consumption.

The second reason is that too high a level of anonymity can do more harm than good. For example, while total anonymity provides maximum protection to users which could be useful in scenarios in which anonymity is of prime concern, such as leaking secrets [36], being totally anonymous is often too strong a protection which possibly renders some other parties not properly protected and makes such level of anonymity unsuitable for many real life applications. For instance, in many scenarios one would like to have a trusted third party capable of tracing users in order to deter them from misbehaving irresponsibly, such as tracing double-spenders in an e-cash system.

Designing secure cryptographic schemes with maximum 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 application in the real world such as in e-commerce.

A good example is ring signature. A ring signature scheme allows a signer to generate a signature on behalf of a group of users such that everyone can be sure that the signature is generated by one of the group members yet no one can tell who the actual signer is. Different from group signature, there is no group manager, no member revocation, and it is spontaneous (setup-free). Linkable ring signature allows anyone to tell whether two signatures are generated by the same signer or not. However, the verifier still has no way to revoke the anonymity of the actual signer.

1.1 Background and Related Work

Identity-based Cryptography. In 1984, Shamir [37] 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 [9]. In 2004, Bellare et al. [6] developed a framework to analyze the security of ID-based signature schemes and they proved the security (or insecurity) of 14 schemes exist in the literature implicitly or explicitly. As in the case of standard signature, there are also blind signature [46], proxy signature [44], proxy blind signature [22], proxy ring signature [3, 46], and proxy signcryption [29] in the paradigm of ID-based cryptography.

Group-oriented Cryptography. Group-oriented cryptographic schemes are schemes in which more than one user is involved, e.g. secret sharing schemes, group signature schemes, etc. In some of these schemes, every member in a specified group always participates equally well in the cryptographic process and the concern of anonymity simply does not apply. In some other schemes, however, the participation of only one or a proper subset of members among a specified group is required to complete the process, while the remaining members are not involved in (and are possibly even unaware of) the process. Such a distinction between participants and non-participants gives anonymity a meaning. Specifically, a participant may prefer himself to be indistinguishable from the whole group of members, thus maintaining his privacy in participating the process.

Anonymity requirements vary from scheme to scheme. According to the level of anonymity they provide, group-oriented cryptographic schemes can be categorized as follows.

No anonymity means the identities of the participating users are known to everyone, probably by examining the output of a cryptographic process. Privacy is usually not a concern for the participants in schemes with no anonymity. For example, in a multi-signature scheme [27, 32], everyone can identify who has actually contributed in the signing process.

Anonymity, on the other hand, means not everyone can identify participating users. A good example is ring signature [36], 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, 21] since the first appearance of ring signature in 1994 [20] and the formal introduction

[4] Readers are referred to [24] for anonymous remailers and other privacy enhancing technologies.

of it in 2001 [36]. The first ID-based ring signature was proposed in in 2002 [45]. To the best of the authors' knowledge, all the of existing ID-based ring signature schemes are pairing-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 [19, 5, 8] provide revocable anonymity. Many credential systems [14–16] also provide revocable anonymity.

Linkable Anonymity is anonymity with a condition. Schemes with linkable anonymity give 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 the two signatures are produced by the same user or not, thus resulting in a reduced level of anonymity.

Linkable ring signature was introduced in [30]. [40] gave a separable construction that supports thresholding. The first constant-sized linkable ring signature was proposed in [39]. Linkable group signature first appeared in [33]. Restricted multi-show credential systems [31] and direct anonymous attestation [11] also provide linkable anonymity.

Revocable-iff-Linked Anonymity can be viewed as somewhat in between linkable anonymity and revocable anonymity. On one hand, there is no authority capable of revoking the anonymity. On the other hand, everyone can revoke the anonymity if a user failed to meet the condition (e.g. signing more than once). Revocable-iff-linked anonymity takes the good from both – behaving users need not worry about the unjustified revocation of anonymity by the overpowered revocation manager, at the same time, it is also guaranteed that misbehaving users who failed in satisfying the condition can be identified for actions such as penalty to be taken.

There are protocols that provide revocable-iff-linked anonymity. Brands's e-cash system [10] retains the privacy of honest users in payment. Users who double spend, however, can be identified. Recently, a new e-cash system [13], a new authentication protocol [38] and a new group signature [42] were proposed. They all have the property of revocable-iff-linked anonymity and incorporate Brand's technique into their constructions.

1.2 Our Contributions

- We propose the first construction of the ID-based (threshold) ring signature scheme that is not based on bilinear pairings. We show its security under the Strong RSA Assumption and the DDH Assumption, in the Random Oracle model [7].
- Based on our basic construction, we construct two more ID-based threshold ring signature (or authentication) schemes, each of which provides a different level of anonymity. They are:
  - The *first* ID-based *Linkable* Threshold Ring Signature. All linkable ring signatures mentioned above are not ID-based. We propose the first in the literature.
  - ID-based *Revocable-iff-linked* Threshold Ring Authentication.
- We propose how to add identity escrow in all the three schemes. With identity escrow, some trusted authority can revoke the anonymity of a ring signature when it becomes necessary. The ability of revocation of the real signer of a ring signature can help prevent the signature scheme from being abused by misbehaving users. The three 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.

Paper Organization. We give some preliminaries in Sec. 2 and specify a security model in Sec. 3. We then propose an ID-based threshold ring signature scheme in Sec. 4. In Sec. 5, an ID-based linkable threshold ring signature scheme and an ID-based threshold ring authentication scheme are then proposed. In Sec. 6, we show how to add identity escrow to our schemes. The paper is concluded in Sec. 7.

2 Preliminaries

A safe prime p is a prime such that $(p-1)/2$ is also prime. Although it has never been proven, it is widely conjectured and amply supported by empirical evidence, that safe primes are sufficiently dense. 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, $\nu(\lambda) < 1/\mathsf{poly}(\lambda)$. A function is non-negligible if it is not negligible. 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 [4, 25]). *Let $n = pq$ be an RSA modulus. Let G be a cyclic subgroup of $\mathbb{Z}_n^*$ of order $u = \#G$, $\lceil \log_2(\#G) \rceil = \ell_G$. Given n and $z \in G$, 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.

Definition 2 (Decisional Diffie-Hellman (DDH) [7]). *Let G be a cyclic group generated by g of order $u = \#G$. 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 noninteractively. The signer can also tie his knowledge of a secret to a message being signed. Following [18], we call this type of signatures "a signature based on proofs of knowledge", SPK for short. A HVZK Σ-protocol can also be turned into a signature scheme by setting the challenge to the hash value of the commitment together with the message to be signed [23]. Such schemes can be proven secure against existential forgery under chosen-message attack [26] in the random oracle model [7] using the proofing technique introduced in [34].

3 The Security Model of an ID-Based Threshold Ring Signature

An identity-based threshold ring signature (ID-TRS) scheme is a tuple of the following probabilistic polynomial-time (PPT) algorithms:

- `ID-TRS.Setup`. On input an unary string 1^λ where λ is a security parameter, the algorithm outputs a master secret key s and a list of system parameters param that includes λ and the descriptions of a user secret key space $\mathcal{S}$ a message space $\mathcal{M}$ as well as a signature space Ψ.
- `ID-TRS.Extract`. On input a list param of system parameters, an identity $\mathsf{ID}_i \in \{0,1\}^*$ for a user and the master secret key s, the algorithm outputs the user's secret key $s_i \in \mathcal{S}$. When we say identity ID corresponds to user secret key s_i or vice versa, we mean the pair (ID_i, s_i) is an input-output pair of `ID-TRS.Extract` with respect to param and s.
- `ID-TRS.Sign`. On input a list param of system parameters, 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 $m \in \mathcal{M}$, and a set $\{s_j \in \mathcal{S} | j \in \Pi\}$ of t user secret keys with some $\Pi \in \wp_t([1,n])$, the algorithm outputs an ID-based (t,n) threshold ring signature $\sigma \in \Psi$.
- `ID-TRS.Verify`. On input a list param of system parameters, 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 $\sigma \in \Psi$, it outputs either valid or $\mathsf{invalid}$.

Correctness. An ID-based threshold ring signature must satisfy the *verification correctness* – signatures signed by honest signers are verified to be invalid with negligible probability.

3.1 Security Definitions

The security requirements of an ID-TRS scheme include *Unforgeability* and *Anonymity* which will be defined in the similar approach to that of a traditional threshold ring signature scheme, but will be a little bit stronger. In a security definition for a traditional threshold ring signature scheme, it is usually defined to have a set of keys initialized by the game simulator and the adversary can select keys to corrupt under a constraint that those keys are initialized by the simulator. In our security definitions for an ID-TRS scheme, the adversary can choose any identity and corrupt the corresponding key without being constrained to any pre-determined set of identities. Let $\mathcal{A}$ be an adversary. The capabilities of $\mathcal{A}$ is modeled by accessing various kinds of oracles via queries.

- Hash queries: $\mathcal{A}$ can ask for hash values of any finite length strings.
- Key queries: For a query input ID. The oracle returns $S_{\mathsf{ID}} \leftarrow \texttt{ID-TRS.Extract}(\mathsf{param}, \mathsf{ID}, s)$.
- Signature queries: $\mathcal{A}$ chooses a group of n identities $\{\mathsf{ID}_i\}_{i\in[1,n]}$, a threshold value t where $t \in [1,n]$, a set $\mathcal{S} \in \wp_t([1,n])$ and a message m, the oracle outputs a valid ID-based (t,n)-threshold ring signature $\sigma \leftarrow \texttt{ID-TRS.Sign}(\mathsf{param}, n, t, \{\mathsf{ID}_i | i \in [1,n]\}, m, \{s_i | i \in \mathcal{S}\})$.

Definition 3 (Game Unforgeability).

- (Initialization Phase.) *The challenger $\mathcal{C}$ takes a sufficiently large security parameter λ and runs the `ID-TRS.Setup` to generate the list param of system parameters and master secret key s. $\mathcal{C}$ keeps s to himself and sends param to $\mathcal{A}$.*
- (Probing Phase.) *$\mathcal{A}$ makes a polynomial number of oracle queries in an adaptive manner.*
- (End Game Phase.) *$\mathcal{A}$ outputs a group size n of length polynomial in the security parameter λ, a threshold $t \in [1,n]$, a set $\{\mathsf{ID}_i \in \{0,1\}^* | i \in [1,n]\}$ of n identities, a message $m \in \mathcal{M}$ and an ID-based (t,n)-threshold ring signature $\sigma \in \Psi$. The only restriction is that $(m, \{\mathsf{ID}_i\})$ should not appear in any of the previous signature queries and less than t secret keys of $\{\mathsf{ID}_i\}$ are returned by key queries.*

$\mathcal{A}$ wins the game if $\texttt{ID-TRS.Verify}(\mathsf{param}, n, t, \{\mathsf{ID}_i\}, m, \sigma)$ returns accept. The advantage of $\mathcal{A}$ is defined as the probability that $\mathcal{A}$ wins.

Definition 4 (EUF-IDTR-CMIA). *An ID-based threshold ring signature scheme is existential unforgeable against adaptive chosen-message-and-identity attacks (or EUF-IDTR-CMIA secure) if no PPT adversary has a non-negligible advantage in Game Unforgeability above.*

Definition 5 (Game Anonymity).

- (Initialization Phase.) $\mathcal{C}$ *takes a sufficiently large security parameter* λ *and runs* `ID-TRS.Setup` *to generate* param *and master secret key* s. $\mathcal{C}$ *keeps* s *to himself and sends* param *to* $\mathcal{A}$.
- (Probing Phase I.) $\mathcal{A}$ *makes a polynomial number of oracle queries in an adaptive manner.*
- (Challenge Phase.) $\mathcal{A}$ *gives* $\mathcal{C}$ *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 *identities and a message* $m \in \mathcal{M}$. $\mathcal{C}$ *picks randomly an index set* $\Pi \in_R \wp_t([1,n])$ *and computes* $\sigma \leftarrow$ `ID-TRS.Sign`$(\mathsf{param}, n, t, \{\mathsf{ID}_i\}, m, \{s_i | i \in \Pi\})$, *where each* s_i *is the user secret key that corresponds to* ID_i.
- (Probing Phase II.) $\mathcal{A}$ *makes a polynomial number of oracle queries in an adaptive manner.*
- (End Game Phase.) $\mathcal{A}$ *outputs an index* $\hat{\pi}$.

$\mathcal{A}$ *wins the game if* $\hat{\pi} \in \Pi$. *The advantage of* $\mathcal{A}$ *is defined as the probability that* $\mathcal{A}$ *wins minus* $\frac{t}{n}$.

Definition 6 (IND-IDTR-CMIA). *An ID-based threshold ring signature scheme is signer indistinguishable against adaptive chosen-message-and-identity attacks (or IND-IDTR-CMIA secure) if no PPT adversary has a non-negligible advantage in Game Anonymity above.*

Definition 7 (Security). *An ID-based threshold ring signature scheme is secure if it is EUF-IDTR-CMIA secure and IND-IDTR-CMIA secure.*

4 The 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 some group master. The knowledge of the factorization gives the group master the ability to extract the secret keys from any identity.

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):

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

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 [20] into SPK_1.

We now describe the details of our ID-based (d, 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 $H_{id} : \{0,1\}^* \rightarrow QR(N)$ and $H_{sig} : \{0,1\}^* \rightarrow \mathbb{Z}_{2^\kappa}$. 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 $\mathsf{ID} \rightarrow \mathsf{g}^{\mathsf{h}(\mathsf{ID})} \bmod N$, where $\mathsf{h} : \{0,1\}^* \rightarrow \{0,1\}^{2\lambda+\theta}$ 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 $d \in [1, n]$, a set of n IDs $\mathcal{Y} = \{\mathsf{ID}_1, \cdots, \mathsf{ID}_n\}$, a list of d secret keys $\mathcal{X} = \{sk_{\pi_1}, \cdots, sk_{\pi_d}\}$ 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 $\mathcal{I} := \{\pi_1, \cdots, \pi_d\} \subseteq [1, n]$, computes $y_i := H_{id}(\mathsf{ID}_i)$ for all $i \in [1, n]$ and then does the following:
  1. *(Auxiliary commitment.)* For all $i \in \mathcal{I}$, 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 $i \in [1, n] \backslash \mathcal{I}$, pick $A_{i,1}, A_{i,2}, A_{i,3} \in_R QR(N)$.
  2. *(Commitment.)* For all $i \in \mathcal{I}$, 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 $i \in [1, n] \backslash \mathcal{I}$, 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 $(n-d)$ such that $c_0 = f(0)$ and $c_i = f(i)$ for all $i \in [1, n] \backslash \mathcal{I}$. For all $i \in \mathcal{I}$, 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:

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

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 $(n - d)$.
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: $s_{i,u} \stackrel{?}{\in} \{0,1\}^{\epsilon(2\lambda+\kappa)+1}$, $s_{i,x} \stackrel{?}{\in} \{0,1\}^{\epsilon(\gamma_2+\kappa)+1}$, $s_{i,w} \stackrel{?}{\in} \{0,1\}^{\epsilon(\gamma_1+2\lambda+\kappa+1)+1}$, for all $i \in [1, n]$, and

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

4. 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 B.

5 Other Members in the Suite

In this section, we propose two more schemes, each of which has a different level of anonymity. They are the *first* ID-based linkable threshold ring signatures and ID-based revocable-iff-linked threshold ring authentication.

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

An ID-based linkable threshold ring signature (ID-LTRS) scheme is a tuple of five PPTs:

– `ID-LTRS.Setup`. Same as `ID-TRS.Setup`, except: (1) it additionally gets as input $k \in \mathbb{N}$ of length polynomial in the security parameter λ, and (2) the list of system parameters param additionally includes an event-ID space $\mathcal{E}$.
– `ID-LTRS.Extract`. Exactly the same as `ID-TRS.Extract`.
– `ID-LTRS.Sign,Verify`. Same as `ID-TRS.Sign,Verify`, except they both additionally get as input an event-ID $e \in \mathcal{E}$.
– `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 $\mathsf{valid} \leftarrow$ `Verify`(param, e, n_j, t_j, $\mathcal{Y}_j$, m_j, σ_j) for $j = 1, 2$, the algorithm returns either linked or $\mathsf{unlinked}$.

Correctness. In addition to verification correctness as for ID-TRS schemes, an ID-LTRS scheme must also satisfy the *linking correctness* – signatures signed by the same signer are unlinked with negligible probability and those signed by different signers are linked with negligible probability.

Remark: According to [39], linkability for threshold ring signatures is diversified into *individual-linkability* and *coalition-linkability*, our construction belongs to the former type.

Security Model. The security requirements of ID-LTRS schemes include *Unforgeability*, *Anonymity*, *Linkability* and *Non-slanderability*. The definition of Unforgeability for ID-LTRS is virtually the same as that for ID-TRS schemes. A crucial difference between Anonymity for ID-LTRS and Anonymity for ID-TRS schemes is that in the former, the adversary cannot query signatures of a user who appears in the challenge phase. Linkability means that an adversary with adaptive hash, key and signature queries cannot produce two valid but unlinked signatures given that he has only corrupted at most one user. Non-slanderability says that an adversary with same queries cannot produce a valid signature that is linked to a signature generated by a user, without corrupting that user. Due to the lack of space, we omit the detailed definition of the security notions above.[5]

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. 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 $\mathcal{E} := \{e_i | i \in [1,k]\}$.
- `ID-LTRS.Extract`. Exactly the same as `ID-LTRS.Extract`.
- `ID-LTRS.Sign`. Compute $\tau_i := e^{x_i} \bmod N$ for all $i \in \mathcal{I}$ and $\tau_i := e^{t_i} \bmod N$ with $t_i \in_R \Gamma'$ for all $i \in [1,n] \backslash \mathcal{I}$. 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:

$$SPK_4 \left\{ (a_i, x_i)_{i=1}^n : \bigvee_{\mathcal{J} \in \wp_d([1,n])} \bigwedge_{i \in \mathcal{J}} y_i \equiv a_i^{x_i} \wedge \tau_i \equiv e^{x_i} \wedge x_i \in \Gamma \right\} (M) \tag{4}$$

  which is instantiated as:

$$SPK_5 \left\{ (u_i, x_i, w_i)_{i=1}^n : \bigvee_{\mathcal{J} \in \wp_d([1,n])} \bigwedge_{i \in \mathcal{J}} \begin{array}{c} A_{i,1} \equiv g_1^{u_i} \wedge A_{i,3} \equiv g_1^{x_i} g_3^{u_i} \wedge \\ A_{i,1}^{x_i} \equiv g_1^{w_i} \wedge A_{i,2}^{x_i} \equiv g_2^{w_i} y_i \wedge \\ \tau_i \equiv e^{x_i} \wedge x_i \in \Gamma \end{array} \right\} (M). \tag{5}$$

  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 $i \in \mathcal{I}$ and t_i's for $i \in [1,n] \backslash \mathcal{I}$:

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

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

[5] Readers may refer to [30, 40, 39] for more details although they are not in an ID-based setting.

– `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 $\mathsf{valid} \leftarrow$ `Verify`(param, e, n_j, t_j, $\mathcal{Y}_j$, m_j, σ_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 $\mathsf{unlinked}$.

Security Arguments. *Unforgeability*: it can be proved in a similar manner as it the case for ID-TRS. The signature using a random number as the tag (i.e., using a random number instead of e^{x_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 only sign once, distinguishing the actual signer solves a DDH instance of (g_1,e,${g_1}^{x_i}$, e^{x_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 $H(ID) = {a_1}^{e_1} = {a_2}^{e_2}$ for some distinct e_1, e_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.

5.2 ID-based Revocable-iff-Linked Threshold Ring Authentication (ID-RiLTRA)

We show how our ID-LTRS construction can be adapted to construct an ID-RiLTRA. The technique employed comes from Brand's approach for identifying double-spenders in an e-cash system [10]. In [13, 38], the technique is also used for achieving revocable-iff-linked anonymity. We construct our ID-RiLTRA in two stages: (1) we modify our ID-LTRS to an ID-based Revocable-iff-Linked Threshold Ring Signature (ID-RiLTRS); (2) then we convert the ID-RiLTRS to an authentication protocol.

Our Construction of ID-RiLTRS. An ID-RiLTRS scheme is a tuple of six polynomial-time algorithms: `Setup`, `Extract`, `Sign`, `Verify`, `Link` and `Open`.

– `ID-RiLTRS.Setup`. Same as `ID-LTRS.Setup`, except: (1) it additionally picks $\tilde{e}_i \in_R \mathcal{G}(QR(N))$ for all $i \in [1, k]$, (2) $\mathcal{E}$ is set to be $\{(e_i, \tilde{e}_i) | i \in [1, k]\}$ instead, (3) and it additionally picks $b \in_R \mathcal{G}(QR(N))$, defines the nonce space $\mathcal{L} := \{0,1\}^{2\lambda}$, and includes them in the list of system parameters param.
– `ID-RiLTRS.Extract`. Exactly the same as `ID-LTRS.Extract`.
– `ID-RiLTRS.Sign`. A nonce $\ell \in \mathcal{L}$ is also input to the algorithm in addition to those input to `ID-LTRS.Sign`. The input event-ID is now of the form $(e, \tilde{e}) \in \mathcal{E}$. The algorithm computes $\tau_i := e^{x_i} \bmod N$ and $\tilde{\tau}_i := (b^\ell \tilde{e})^{x_i} \bmod N$ for all $i \in \mathcal{I}$, and $\tau_i := e^{t_i} \bmod N$ and $\tilde{\tau}_i := (b^\ell \tilde{e})^{t_i} \bmod N$ with $t_i \in_R \Gamma'$ for all $i \in [1, n] \backslash \mathcal{I}$.
The algorithm is subsequently modified from `ID-LTRS.Sign` to also prove that the τ_i's and $\tilde{\tau}_i$'s are correctly formed. Specifically, the algorithm now implements:

$$SPK_7 \left\{ (a_i, x_i)_{i=1}^n : \bigvee_{\mathcal{J} \in \wp_d([1,n])} \bigwedge_{i \in \mathcal{J}} \begin{array}{l} y_i \equiv a_i^{x_i} \quad \wedge \tau_i \equiv e^{x_i} \wedge \\ \tilde{\tau}_i \equiv (b^\ell \tilde{e})^{x_i} \ \wedge x_i \in \Gamma \end{array} \right\} (M). \tag{7}$$

which is instantiated as:

$$SPK_8\left\{(u_i,x_i,w_i)_{i=1}^n : \bigvee_{\mathcal{J}\in\wp_d([1,n])}\bigwedge_{i\in\mathcal{J}}\begin{array}{ll} A_{i,1}\equiv g_1^{u_i} & \wedge A_{i,3}\equiv g_1^{x_i}g_3^{u_i}\wedge \\ A_{i,1}^{x_i}\equiv g_1^{w_i} & \wedge A_{i,2}^{x_i}\equiv g_2^{w_i}y_i\wedge \\ \tau_i\equiv e^{x_i} & \wedge\tilde{\tau}_i\equiv(b^\ell\tilde{e})^{x_i}\wedge \\ x_i\in\Gamma & \end{array}\right\}(M). \tag{8}$$

As in the case of SPK_5, the actual steps implementing SPK_8 above follow closely `ID-TRS.Sign` and are thus not explicitly enumerated. We denote by σ_8 the signature output by SPK_8. Note that it includes $\tau_1,\cdots,\tau_n$ and $\tilde{\tau}_1,\cdots,\tilde{\tau}_n$.
In addition, generate a signature σ_9 for the following SPK:

$$SPK_9\left\{(\beta_i)_{i=1}^n : \bigwedge_{i=1}^n \tau_i\equiv e^{\beta_i}\wedge\tilde{\tau}_i\equiv(b^\ell\tilde{e})^{\beta_i}\right\}(M). \tag{9}$$

The detailed implementation of the above SPK is given in Appendix A.
Finally the signature is output as $\sigma := (\sigma_8,\sigma_9)$.

- `ID-RiLTRS.Verify`. Given a signature $\sigma=(\sigma_8,\sigma_9)$, the algorithm verifies the validity of σ_8 w.r.t. SPK_8 and that of σ_9 w.r.t. SPK_9. Again we omit the verification algorithm for SPK_8 as it can be adapted in a straightforward manner from `ID-TRS.Verify`. Verification for SPK_9 is given in Appendix A.
- `ID-RiLTRS.Link`. Exactly the same as `ID-LTRS.Link`.
- `ID-RiLTRS.Open`. On input the list of system parameters param, an event-ID $(e,\tilde{e})\in\mathcal{E}$, 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 nonces $\ell_1,\ell_2\in\mathcal{L}$, two messages $m_1,m_2\in\mathcal{M}$, and two signatures $\sigma_1,\sigma_2\in\Psi$ such that $\mathsf{linked}\leftarrow$ `Link`$(\mathsf{param},(e,\tilde{e}),n_j,t_j,\mathcal{Y}_j,\ell_j,m_j,\sigma_j)$ for $j=1,2$, the algorithm returns a user identity $\mathsf{ID}^*\in\mathcal{Y}_1\cap\mathcal{Y}_2$.
To identify the double signer from two linked signatures, let $\tilde{\tau}^{(1)}\equiv(b^{\ell_1}\tilde{e})^x \pmod N$ and $\tilde{\tau}^{(2)}\equiv(b^{\ell_2}\tilde{e})^x \pmod N$, we compute $\tilde{\tau}^*:=\tilde{\tau}^{(1)}/\tilde{\tau}^{(2)}\equiv(a^x)^{\ell_1-\ell_2} \pmod N$, go through all $y_i=H(\mathsf{ID}_i)$, and report ID_i if $y_i^{\ell_1-\ell_2}\equiv\tau^* \pmod N$.

From ID-RiLTRS to ID-RiLTRA. As long as the same signer signs for the same event-ID but for different nonces, the resulting signatures can be linked and from these signatures the signer can be identified. However, ID-RiLTRS itself is by no means secure for real application because no one can guarantee that a signer will use a different nonce every time he signs. Signatures produced by a cheating signer who always uses the same nonce can only be linked but not opened. Fortunately, in the case of authentication in which Bob wants to authenticate to Alice, Alice can enforce the rule to make no nonce is alike by asking Bob to sign for a nonce chosen by Alice.

Our ID-RiLTRA goes as follows. Alice sets up an ID-RiLTRS by invoking `ID-RiLTRS.Setup`. She then publishes the list of system parameters param. Every time Bob wants to authenticate he interacts with Alice according to the following.

1. Bob sends $(n,d,\mathcal{Y},(e,\tilde{e}))$ as a request for authentication, where $n\in\mathbb{N}$ of length polynomial in the security parameter λ is the group size, $d\in[1,n]$ is the threshold value, $\mathcal{Y}$ is a set of n user identities, and $(e,\tilde{e})\in\mathcal{E}$ is an event-ID.
2. Alice sends back a random $\ell\in_R\mathcal{L}$ and a random message (challenge) $M\in_R\{0,1\}^m$ for some predetermined $m\in\mathbb{N}$ of length polynomial in λ.
3. Bob produces an ID-RiLTRS signature σ by invoking `ID-RiLTRS.Sign` on input the tuple $(\mathsf{param},(e,\tilde{e}),\ell,n,d,\mathcal{Y},\mathcal{X},M)$, where $\mathcal{X}$ is a set of d secret keys with their corresponding public keys in $\mathcal{Y}$.
4. Alice accepts if the signature verifies, i.e. `ID-RiLTRS.Verify` returns valid on input the tuple $(\mathsf{param},(e,\tilde{e}),\ell,n,d,\mathcal{Y},M,\sigma)$, and σ is not linked to any signature for the same event in the database. She rejects otherwise. Alice records the input tuple in the database if she accepts.

In the above, in case a signature σ is linked some other signature σ' in the database, Alice can invoke `ID-RiLTRS.Open` to identify the one who "double-authenticated".

Security Argument. The inclusion of proving that the $\tilde{\tau}_i$'s are correctly formed does not affect unforgeability, anonymity, linkability and non-slanderability. The soundness of the SPK ensure that $\tilde{\tau}$ is correctly formed and make sure it is openable when linked.

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 [28].

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 [12, 17] has been frequently used (though sometimes implicitly) to achieve revocable anonymity. For instance, the generic constructions of group signatures [5, 8]. As a concrete example, In [2], part of the user's secret key [6] 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 three 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 three 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 Conclusion

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 suggested two more ID-based threshold ring signatures with different levels of anonymity, including the first ID-based linkable (threshold) ring signature scheme. We argued the security of all the constructions.

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

Finally we showed how to add identity escrow to the three 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.

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, 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 *LNCS*, pages 255–270. Springer-Verlag, 2000.
3. 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/.
4. N. Barić and B. Pfitzmann. Collision-free accumulators and fail-stop signature schemes without trees. In *EUROCRYPT 1997*, volume 1233 of *LNCS*, pages 480–494, 1997.
5. 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 *LNCS*, pages 614–629. Springer-Verlag, 2003.
6. M. Bellare, C. Namprempre, and G. Neven. Security proofs for identity-based identification and signature schemes. In *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 268–286, 2004.
7. 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.
8. M. Bellare, H. Shi, and C. Zhang. Foundations of group signatures: The case of dynamic groups. In *CT-RSA 2005*, volume 3376 of *LNCS*, pages 136–153. Springer-Verlag, 2005.
9. D. Boneh and M. K. Franklin. Identity-based encryption from the weil pairing. In *CRYPTO 2001*, volume 2139 of *LNCS*, pages 213–229, 2001.
10. S. Brands. Untraceable off-line cash in wallets with observers (extended abstract). In *CRYPTO 1993*, volume 773 of *LNCS*, pages 302–318. Springer-Verlag, 1993.
11. E. Brickell, J. Camenisch, and L. Chen. Direct anonymous attestation. In *CCS '04: Proc. of the 11th ACM conference on Computer and communications security*, pages 132–145. ACM Press, 2004.
12. 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 *LNCS*, pages 331–345, 2000.
13. J. Camenisch, S. Hohenberger, and A. Lysyanskaya. Compact e-cash. In *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 302–321, 2005.
14. J. Camenisch and A. Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In *EUROCRYPT 2001*, volume 2045 of *LNCS*, pages 93–118. Springer-Verlag, 2001.
15. J. Camenisch and A. Lysyanskaya. Dynamic accumulators and application to efficient revocation of anonymous credentials. In *CRYPTO 2002*, volume 2442 of *LNCS*, pages 61–76, 2002.
16. J. Camenisch and A. Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In *CRYPTO 2004*, volume 3152, pages 56–72, 2004.
17. J. Camenisch and V. Shoup. Practical verifiable encryption and decryption of discrete logarithms. In *CRYPTO 2003*, volume 2729 of *LNCS*, pages 126–144, 2003.
18. J. Camenisch and M. Stadler. Efficient group signature schemes for large groups (extended abstract). In *CRYPTO 1997*, volume 1294 of *LNCS*, pages 410–424. Springer-Verlag, 1997.
19. D. Chaum and E. van Heyst. Group signatures. In *EUROCRYPT 1991*, volume 547, pages 257–265, 1991.
20. R. Cramer, I. Damgård, and B. Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In *CRYPTO 1994*, volume 839 of *LNCS*, pages 174–187. Springer-Verlag, 1994.
21. 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.
22. Z. Dong, H. Zheng, K. Chen, and W. Kou. ID-based proxy blind signature. In *AINA (2)*, pages 380–383, 2004.

23. A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *CRYPTO 1986*, volume 263 of *LNCS*, pages 186–194. Springer-Verlag, 1986.
24. S. Fischer-Hübner. *IT-Security and Privacy - Design and Use of Privacy-Enhancing Security Mechanisms*, volume 1958 of *LNCS*. Springer, 2001.
25. E. Fujisaki and T. Okamoto. Statistical zero knowledge protocols to prove modular polynomial relations. In *CRYPTO 1997*, volume 1294 of *LNCS*, pages 16–30, 1997.
26. 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.
27. K. Itakura and K. Nakamura. A public key cryptosystem suitable for digital multisignatures. *NEC Research & Development*, 71:1–8, 1983.
28. J. Kilian and E. Petrank. Identity escrow. In *CRYPTO 1998*, volume 1462 of *LNCS*, pages 169–185. Springer-Verlag, 1998.
29. X. Li and K. Chen. Identity based proxy-signcryption scheme from pairings. In *IEEE SCC*, pages 494–497, 2004.
30. 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 *LNCS*, pages 325–335. Springer-Verlag, 2004.
31. J. K. Liu and D. S. Wong. A restricted multi-show credential system and its application on e-voting. In *ISPEC 2005*, volume 3439 of *LNCS*, pages 268–279, 2005.
32. 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.
33. 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.
34. D. Pointcheval and J. Stern. Security proofs for signature schemes. In *EUROCRYPT 1996*, volume 1070 of *LNCS*, pages 387–398, 1996.
35. D. Pointcheval and J. Stern. Security arguments for digital signatures and blind signatures. *J. Cryptology*, 13(3):361–396, 2000.
36. R. L. Rivest, A. Shamir, and Y. Tauman. How to leak a secret. In *ASIACRYPT 2001*, volume 2248 of *LNCS*, pages 552–565. Springer-Verlag, 2001.
37. A. Shamir. Identity-based cryptosystems and signature schemes. In *CRYPTO 1984*, volume 196 of *LNCS*, pages 47–53, 1984.
38. I. Teranishi, J. Furukawa, and K. Sako. k-times anonymous authentication (extended abstract). In *ASIACRYPT 2004*, volume 3329 of *LNCS*, pages 308–322. Springer-Verlag, 2004.
39. P. P. Tsang and V. K. Wei. Short linkable ring signatures for e-voting, e-cash and attestation. In *ISPEC 2005*, volume 3439 of *LNCS*, pages 48–60. Springer-Verlag, 2005.
40. 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 *LNCS*, pages 384–398. Springer-Verlag, 2004.
41. S. D. Warren and L. D. Brandeis. The right to privacy. *Harvard Law Review*, IV(5):193–220, 1890.
42. V. K. Wei. Tracing-by-linking group signatures. Cryptology ePrint Archive, Report 2004/370, 2004. http://eprint.iacr.org/.
43. A. F. Westin. Privacy and freedom. Atheneum, 1970.
44. 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/.
45. F. Zhang and K. Kim. Id-based blind signature and ring signature from pairings. In *ASIACRYPT 2002*, volume 2501 of *LNCS*, pages 533–547. Springer-Verlag, 2002.
46. F. Zhang and K. Kim. Efficient id-based blind signature and proxy signature from bilinear pairings. In *ACISP*, pages 312–323, 2003.

A Implementations of SPK_6 and SPK_9

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

1. *(Commitment.)* Pick $\rho_i \in_R \pm\{0,1\}^{\epsilon(\gamma_2+\kappa)}$ 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 $i \in [1,n] \backslash \mathcal{I}$.

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 $s_i \stackrel{?}{\in} \{0,1\}^{\epsilon(\gamma_2+\kappa)+1}$ for all $n \in [1, n]$, and $c \stackrel{?}{=} H_{sig}(\mathsf{param}, n, g, (\tau_1, T_1')_{i=1}^n, M)$.

***SPK*$_9$.** To sign a signature for SPK_9, do the following:

1. *(Commitment.)* Pick $\rho_i \in_R \pm\{0,1\}^{\epsilon(\gamma_2+\kappa)}$ and compute $T_{1,i} := g^{\rho_i} \bmod N$ and $T_{2,i} := (b^\ell \tilde{g})^{\rho_i} \bmod N$ for all $i \in [1, n]$.
2. *(Challenge.)* Compute $c := H_{sig}(\mathsf{param}, n, g, b, \tilde{g}, \ell, (\tau_i, \tilde{\tau}_i, T_{1,i}, T_{2,i})_{i=1}^n, M)$.
3. *(Response.)* Compute $s_i := \rho_i - cx_i$ for all $i \in \mathcal{I}$ and $s_i := \rho_i - ct_i$ for all $i \in [1, n] \backslash \mathcal{I}$.

The signature for SPK_9 is thus $\sigma_9 := (c, s_1, \ldots, s_n)$.

Verification for $\sigma_9 = (c, s_1, \ldots, s_n)$ is done by first computing $T_{1,i}' := g^{s_i}\tau_i^c \bmod N$ and $T_{2,i}' := (b^\ell \tilde{g})^{s_i}\tilde{\tau}_i^c \bmod N$ for all $i \in [1, n]$ and then checking if $s_i \stackrel{?}{\in} \{0,1\}^{\epsilon(\gamma_2+\kappa)+1}$ for all $n \in [1, n]$, and $c \stackrel{?}{=} H_{sig}(\mathsf{param}, n, g, b, \tilde{g}, \ell, (\tau_1, \tilde{\tau}_1, T_{1,i}', T_{2,i}')_{i=1}^n, M)$.

B 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 d where $d \in [1, n]$, a set $\mathcal{S} \in \wp_d([1, n])$ 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 $\mathcal{S} = [1, d]$ and $\mathsf{ID}_d = \mathsf{ID}^*$.

1. *(Auxiliary commitment.)* For all $i \in [1, d-1]$, 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 $i \in [d, n]$, randomly pick $A_{i,1}, A_{i,2}, A_{i,3} \in_R QR(N)$.

2. *(Commitment.)* For all $i \in [1, d-1]$, 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 $i \in [d, n]$, 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 $(n-d)$ such that and $c_i = f(i)$ for all $i \in [d, n]$ and set $H_{sig}(\mathsf{param}, n, d, (y_i, A_{i,1}, A_{i,2}, A_{i,3})_{i=1}^n, (T_{i,1}, \cdots, T_{i,4})_{i=1}^n, M) = f(0)$.
4. *(Response.)* For all $i \in [1, d-1]$, 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 (d, n)-threshold ring signature for a group $\mathcal{Y}$ such that $\mathsf{ID}^* \in \mathcal{Y}$, and $\mathcal{A}$ only issues up to $d-1$ 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 [35] 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 $n-d$, there are at least d values $k_1, k_2, \cdots, k_d$ 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 $0 \leq n_a \leq d-1$. 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 d out of n secret keys in the group since there are at least d values $k_1, k_2, \cdots, k_d$ such that $f(k_i) \neq f'(k_i)$. Suppose n_b secret keys corresponding to uncorrupted identities in $\mathcal{Y}$ are computed. Here $1 \leq n_b \leq d$. 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 d, a set $\{\mathsf{ID}_i\}_{i\in[1,n]}$ of identities and a message m. $\mathcal{C}$ picks randomly $\Pi \in_R \wp_d([1,n])$. Without loss of generality, we assume $\Pi = [1,d]$ and $\mathcal{C}$ computes σ as follows.

1. *(Auxiliary commitment.)* For all $i \in [1, d-1]$, 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 $i = d$, 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 $i \in [d+1, n]$, pick $A_{i,1}, A_{i,2}, A_{i,3} \in_R QR(N)$.
2. *(Commitment.)* For all $i \in [d-1]$, 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 $i \in [d, n]$, 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 $(n-d)$ such that and $c_i = f(i)$ for all $i \in [d, n]$ and set $H_{sig}(\mathsf{param}, n, d, (y_i, A_{i,1}, A_{i,2}, A_{i,3})_{i=1}^n,\ (T_{i,1}, \cdots, T_{i,4})_{i=1}^n, M) = f(0)$.
4. *(Response.)* For all $i \in [1, d-1]$, 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 $\hat{\pi} = d$. 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 $d/n + \epsilon_{\mathcal{A}}$ for some non-negligible $\epsilon_{\mathcal{A}}$. There are three cases that $\mathcal{C}$ will win. Case 1: $\mathcal{A}$ outputs $\hat{\pi} = d$ and the challenge is a valid DDH-tuple. Case 2: $\mathcal{A}$ outputs $\hat{\pi} \neq d$ 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 $\hat{\pi} \in [1, d]$ given that the challenge is a valid DDH-tuple is $\epsilon_{\mathcal{A}}$. As the value of d is also randomly chosen, the probability of case 1 is $1/2n + \epsilon_{\mathcal{A}}/2d$. 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 $\frac{1}{4}(1 - (\frac{1}{n} + \frac{\epsilon_{\mathcal{A}}}{d}))$. 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 $i \in [d+1, n]$, i.e. same as those non-signers. Hence the probability of the second sub-case is equal to one minus the probability that $\hat{\pi} = d$ and the challenge is not a DDH-tuple. The probability of $\hat{\pi} = d$ given that the challenge is not a DDH-tuple is $\psi = (1 - (d/n + \epsilon_{\mathcal{A}}))/(n - d + 1)$. Hence the probability of winning for $\mathcal{C}$ in the second sub-case is $\frac{1}{4}(1 - \psi) = \frac{1}{4} - \frac{1 - d/n - \epsilon_{\mathcal{A}}}{4(n-d+1)}$. Combining all cases, we have the winning probability of $\mathcal{C}$ to be at least $\frac{1}{2} + \frac{\epsilon_{\mathcal{A}}}{4d}$. □

Corollary 1 (Security). *Under the condition that both λ and κ are sufficiently large, the ID-TRS scheme proposed in Sec. 4 is secure under the Strong RSA Assumption and the DDH Assumption, in the random oracle model.*

Proof. A trivial implication from the two theorems above. □