Solutions to Key Exposure Problem in Ring Signature

Joseph K. Liu[1], and Duncan S. Wong[2]

[1] Department of Computer Science
University of Bristol
Bristol, U.K.
liu@cs.bris.ac.uk
[2] Department of Computer Science
The City University of Hong Kong
Hong Kong
duncan@cityu.edu.hk

Abstract. In this paper, we suggest solutions to the key exposure problem in ring signature. In particular, we propose the first forward secure ring signature scheme and the first key-insulated ring signature schemes. Both constructions allow a (t, n)-threshold setting. That is, even t secret keys are compromised, the validity of all forward secure ring signatures generated in the past is still preserved. In the other way, the compromise of up to all secret keys does not allow any adversary to generate a valid key-insulated ring signature for the remaining time periods.
All our proposed schemes are proven secure in the random oracle model.

Keywords: Forward Secure, Key-Insulated, Ring Signature

1 Introduction

Ring Signatures. A ring signature scheme [21, 9, 4, 24, 8, 18, 14] allows members of a group to sign messages on behalf of the group without revealing their identities. Different from a group signature scheme [11, 10, 6], the formation of a group is spontaneous and there is no group manager to revoke the identity of the signer. The anonymity of the actual signer is protected unconditionally or computationally. Under the assumption that each user is already associated with a public key of some standard signature scheme, a user can form a group by simply collecting the public keys of all the group members including his own. These diversion group members can be totally unaware of being conscripted into the group.

Ring signatures could be used for whistle blowing [21], anonymous membership authentication for ad hoc groups [9] and many other applications which do not want complicated group formation stage but require signer anonymity. For example, in the whistle blowing scenario, a whistleblower gives out a secret as well as a ring signature of the secret to the public. From the signature, the public can be sure that the secret is indeed given out by a group member while cannot figure out who the whistleblower is. At the same time, the whistleblower does not need any collaboration of other users who have been conscripted by him into the group of members associated with the ring signature. Hence the anonymity of the whistleblower is ensured and the public is also certain that the secret is indeed leaked by one of the group members associated with the ring signature.

In 2002, Bresson et al. [9] extended the notion of ring signature schemes to a threshold setting and proposed the first threshold ring signature scheme. Later on, some other threshold ring signature schemes [23, 18] have been proposed. A t-out-of-n threshold ring signature scheme is defined as a ring signature scheme of which at least t corresponding private keys of n public keys are needed to produce a signature. The setup-free and signer anonymity properties of a conventional ring signature scheme are preserved in the threshold setting.

Key Exposure Problem in Ring Signature. Ordinary digital signatures have a fundamental limitation: If the secret key of a signer is compromised, all the signatures of that signer become worthless. This may become quite a realistic threat since if the secret key is compromised, any message can be forged. All future signatures are invalidated as a result of such a compromise, and more importantly, no previously issued signatures can be trusted. Once a leakage has been identified, there may exist some key revocation mechanism to be involved immediately in order to prevent the generation of any signature using the compromised secret key. However, this does not solve the problem of forgeability for past signatures. It is not possible to ask the signer to re-issue all previous signatures due to many physical and practical limitations.

The problem of key exposure in a ring signature scheme is more serious. In ring signature schemes, if a user's secret key is exposed to an adversary, the adversary can generate not only odinary digital signature for any documents, but he can also sign any documents on behalf of the group. More worse, the group can be defined by the adversary due to the spontaneity property of ring signature schemes. The exposure of one user's secret key not only requires changing the public key pairs for the whole group, but also renders all previously obtained ring signatures invalid, because one cannot distinguish whether a signature is generated by an adversary after it has obtained one of the secret keys or by the legitimate user before the adversary got the secret key.

Solutions:

1. **Forward Secure Signature.**
   Forward-secure signature schemes are designed to resolve the key explosure fundamental limitation of digital signature. The goal of a forward-secure signature scheme is to preserve the validity of past signatures even if the current secret key has been compromised. The concept was first suggesed by Anderson [5], and solutions were designed by Bellare and Miner [7]. The idea is that even a compromise of the present secret key does not enable an

adversary to forge signatures pertaining to the past. This can be achieved by the key evolution paradigm: by dividing the total time of the validity of the public key into T time periods, and using a different secret key in each time period while the public key remains the same. Each subsequent secret key is computed from the current secret key via an update algorithm, while any past secret key cannot be computed by the current one. The time period during which a message is signed becomes part of the signature as well. Forward security property means that even if the current secret key is compromoised, a forger cannot forge signatures for past time periods. In other words, the forger can only forge signatures for documents pertaining to time periods after the exposure but not before. The integrity of documents signed before the exposure remains intact.

Forward Secure Ring Signature. We propose to use the concept of *forward security* to reduce the damage of exposure of any secret key of users in ring signature. That is, even when a secret key is compromised, previously generated ring signatures remain valid and do not need to be re-generated. We are the first in the literature to propose the concept of forward secure ring signature.

2. **Key-Insulated Signature.** The notion of key-insulated cryptosystems, which was first introduced in [12], generalizes the concept of forward-secure cryptography. Similiar to forward security, in key-insulated cryptosystems, lifetime of secret keys is also divided into discrete periods. In the case of signature, they are supposed to be generated under an insecure environment. In the model of key-insulated signature, the secret associated with a public key is shared between the user and a physically secure device. At the beginning of each time period, the user obtains a partial secret key from the device. By combining this partial secret key with the secret key for the previous period, the user renews the secret key for the current time period. Exposure of the secret key at a given period will not enable any adversary to derive the secret key for the remaining period, since the adversary is not able to break into the physically secure device. Thus he cannot renew the secret key for the next period.

   Key-Insulated Ring Signature. In addition to forward secure ring signature, we also propose the concept of *key-insulated* in ring signature. It reduces the risk of key exposure in ring signature. That is, even a secret key is comproised, the adversary cannot generate any valid ring signature in all future time period using the compromised secret key. Thus it is not necessary to renew public key pairs of all users even some of the corresponding secret key is compromised. We are also the first in the literature to propose this concept in ring signature.

1.1 Our Contributions

We suggest solutions to the key exposure problem in ring signature. We propose two new concepts, namely *forward secure ring signature* and *key-insulated ring signature*. We give rigorous security model and concrete implementations on these two concepts respectively. Both our constructions allow a threshold setting. That is, a valid signature is generated by t-out-of-n users of a spontaneously formed group while the anonymity of the t acutal signers is preserved. Both schemes are proven secure in the random oracle model. We are also the first in the literature to propose forward secure ring signature scheme and key-insulated ring signature scheme.

1.2 Organization

The paper is organized as follows: Some related works will be given in Sec. 2. We define our security mode in Sec. 3. A construction of forward secure ring signature scheme is presented in Sec. 4. It is followed by another construction of key-insulated ring signature scheme in Sec. 5. We conclude the paper in Sec. 6.

2 Related Works

The concept of forward secure signatures was first proposed by Anderson [5] for traditional signatures. It was formalized by Bellare and Miner [7]. The basic idea is to extend a standard digital signature algorithm with a key update algorithm, so that the secret key can be changed frequently while the public key stays the same. The resulting scheme is forward secure if the knowledge of the secret key at some point in time does not help forge signatures relative to some previous time period. The challenge is to design and efficient scheme of this concept. In particular the size of the secret key, public key and signatures should not be dependent on the number of time period during the lifetime of the public key. Several schemes have been proposed by traditional signatures and threshold signatures that satisfy this efficiency property in [2, 17, 1, 16, 19]. In addition, a forward secure group signature scheme is proposed in [22].

The notion of key-insulated cryptosystems was first introduced by Dodis et al. [12], in the context of public key encryption. Later they proposed a key-insulated signature scheme in [13]. A more efficient scheme was proposed in [15]. In their scheme, the key length is constant and independnt of the number of insulated time periods.

3 Security Model

We give our security model and define relevant security notions.

3.1 Definition of Forward Secure Threshold Ring Signature Scheme

Syntax. A *forward secure (threshold) ring signature*, (FSRS) scheme, is a tuple of five algorithms (Key-Gen, Init, Sign, Verify and Update).

- $(sk_{i,0}, pk_i) \leftarrow \mathsf{Key\text{-}Gen}(1^{\lambda_i})$ is a PPT algorithm which, on input a security parameter $\lambda_i \in \mathbb{N}$, outputs a private/public key pair $(sk_{i,0}, pk_i)$ such that the private key is valid for time $t = 0$.[3] We denote by $\mathcal{SK}$ and $\mathcal{PK}$ the domains of possible secret keys and public keys, respectively. When we say that a public key corresponds to a secret key or vice versa, we mean that the secret/public key pair is an output of Key-Gen.
- $\mathsf{param} \leftarrow \mathsf{Init}(\lambda)$ is a PPT algorithm which, on input a security parameter λ, outputs the set of security parameters param which includes λ.
- $sk_{i,t+1} \leftarrow \mathsf{Update}(sk_{i,t}, t)$ is a deterministic algorithm which, on input a private key for a certain time period t and t, outputs a new private key for the time period $t + 1$.
- σ'_t=(n,d,$\mathcal{Y}$,σ)$\leftarrow \mathsf{Sign}(t, n, d, \mathcal{Y}, \mathcal{X}, M)$ is a PPT algorithm which, on input a certain time period t, group size n, threshold $d \in \{1, \ldots, n\}$, a set $\mathcal{Y}$ of n public keys in $\mathcal{PK}$, a set $\mathcal{X}$ of d private keys whose corresponding public keys are all contained in $\mathcal{Y}$, and a message M, produces a signature σ'_t.
- $1/0 \leftarrow \mathsf{Verify}(M, \sigma'_t, t)$ is a deterministic algorithm which, on input a message-signature pair (M,σ'_t) and a time t returns 1 or 0 for accept or reject, resp. If accept, the message-signature pair is *valid*.

Notions of Security. Security of FSRS schemes has three aspects: correctness, forward security and anonymity. Before giving their definition, we consider the following oracles which together model the ability of the adversaries in breaking the security of the schemes.

- $pk_i \leftarrow \mathcal{JO}(\perp)$. The *Joining Oracle*, on request, adds a new user to the system. It returns the public key $pk \in \mathcal{PK}$ of the new user.
- $sk_{i,t} \leftarrow \mathcal{CO}(pk_i, t)$. The *Corruption Oracle*, on input a public key $pk_i \in \mathcal{PK}$ that is a query output of $\mathcal{JO}$ and a time t, returns the corresponding secret key $sk_{i,t} \in \mathcal{SK}$ for the time t.
- $\sigma'_t \leftarrow \mathcal{SO}(t, n, d, \mathcal{Y}, \mathcal{V}, M)$. The *Signing Oracle*, on input a time t, a group size n, a threshold $d \in \{1, \ldots, n\}$, a set $\mathcal{Y}$ of n public keys, a subset $\mathcal{V}$ of $\mathcal{Y}$ with $|\mathcal{V}| = d$, and a message M, returns a valid signature σ'_t for time t.

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

Correctness. Signatures signed according to specification are accepted during verification.

[3] We denote $sk_{i,t}$ to be the secret key of user i at time t.

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

1. $\mathcal{S}$ generates and gives $\mathcal{A}$ the system parameters param.
2. $\mathcal{A}$ chooses a time t, a group size $n \in \mathbb{N}$, a threshold $d \in \{1, \ldots, n\}$, a set $\mathcal{Y}$ of n public keys in $\mathcal{PK}$ and a message M.
3. $\mathcal{A}$ may query the oracles according to any adaptive strategy.
4. $\mathcal{A}$ outputs a signature σ_t.

$\mathcal{A}$ wins the game if: (1) $\mathsf{Verify}(M,\sigma_t,t)$=1, (2) all of the public keys in $\mathcal{Y}$ are query outputs of $\mathcal{JO}$, (3) at most $(d-1)$ of the public keys in $\mathcal{Y}$ have been input to $\mathcal{CO}$ with time $t' < t$ to be the time input parameter, (4) unlimited query to $\mathcal{CO}$ with time $t'' \geq t$ to be the time input parameter, and (5) σ_t is not a query output of $\mathcal{SO}$ on any input containing M. We denote by $\mathbf{Adv}_{\mathcal{A}}^{fs}(\lambda)$ the probability of $\mathcal{A}$ winning the game.

Remarks. In this game, we do not limit the number of queries made to $\mathcal{CO}$ that are corresponding to the public keys in $\mathcal{Y}$. We only require the number of queries to $\mathcal{CO}$ of the public keys in $\mathcal{Y}$ with time input parameter less than t, should be at most $(d-1)$.

Definition 1 (forward-secure). *An FSRS scheme is forward-secure if for all PPT adversary $\mathcal{A}$, $\mathbf{Adv}_{\mathcal{A}}^{fs}(\lambda)$ is negligible.*

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

1. $\mathcal{S}$ generates and gives $\mathcal{A}$ the system parameters param.
2. $\mathcal{A}$ may query the oracles according to any adaptive strategy.
3. $\mathcal{A}$ gives $\mathcal{S}$ a time t, group size n, threshold $d \in \{1, \ldots, n\}$, message M, a set $\mathcal{Y}$ of n public keys all of which are query outputs of $\mathcal{JO}$, and none of which has been queried to $\mathcal{CO}$.
   $\mathcal{S}$ randomly selects a subset $\mathcal{V} \subset \mathcal{Y}$, $|\mathcal{V}| = d$, to obtain the d corresponding secret keys by quering $\mathcal{CO}$. $\mathcal{S}$ signs with these secret keys and gives the signature to $\mathcal{A}$.
4. $\mathcal{A}$ queries the oracles adaptively, except that any member public key of $\mathcal{Y}$ cannot be queried to $\mathcal{CO}$.
5. $\mathcal{A}$ gives $\mathcal{S}$ a publc key $\widetilde{pk} \in \mathcal{Y}$.

$\mathcal{A}$ wins the game if $\widetilde{pk} \in \mathcal{Y}$. Define the *advantage* of $\mathcal{A}$ as

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

for security parameter λ.

Definition 2 (FS-Anonymity). *A FSRS scheme is anonymous if for any PPT adversary $\mathcal{A}$, $\mathbf{Adv}_{\mathcal{A}}^{FS-Anon}(\lambda)$ is zero.*

Remarks. For anonymity, we require *unconditional anonymous.*

3.2 Definition of Key-Insulated Threshold Ring Signature Scheme

Syntax. A *Key-Insulated secure (threshold) ring signature*, (KIRS) scheme, is a tuple of six algorithms (Key-Gen, Init, Sign, Verify, Device-Update and User-Update).

- $(msk_i, usk_{i,0}, pk_i) \leftarrow \mathsf{Key\text{-}Gen}(1^{\lambda_i})$ is a PPT algorithm which, on input a security parameter $\lambda_i \in \mathbb{N}$, outputs a public key pk_i, a master secret key msk_i, and a user's initial secret key $usk_{i,0}$ such that this key is valid for time $t = 0$.[4] We denote by $\mathcal{PK}$, $\mathcal{MSK}$ and $\mathcal{USK}$ the domains of possible public keys, master secret keys and user secret keys, respectively.
- $\mathsf{param} \leftarrow \mathsf{Init}(\lambda)$ is a PPT algorithm which, on input a security parameter λ, outputs the set of security parameters param which includes λ.
- $psk_{i,t} \leftarrow \mathsf{Device\text{-}Update}(msk_i, t)$ is a deterministic algorithm which, on input the master secret key msk_i and the index of the current time period t, outputs the a partial secret key $psk_{i,t}$ for the time period t.
- $(usk_{i,t}, sk_{i,t}) \leftarrow \mathsf{User\text{-}Update}(usk_{i,t-1}, t)$ is a deterministic algorithm which, on input the user secret key $usk_{i,t-1}$ and the partial secret key $psk_{i,t-1}$ for a certain time period $t-1$ and the index of the current time period t, outputs the user secret key $usk_{i,t}$ and the secret key $sk_{i,t}$ for the time period t.
- σ'_t=$(n$,d,$\mathcal{Y}$,$\sigma)$$\leftarrow$ $\mathsf{Sign}(t, n, d, \mathcal{Y}, \mathcal{X}, M)$ which, on input a certain time period t, group size n, threshold $d \in \{1, \ldots, n\}$, a set $\mathcal{Y}$ of n public keys in $\mathcal{PK}$, a set $\mathcal{X}$ of d private keys whose corresponding public keys are all contained in $\mathcal{Y}$, and a message M, produces a signature σ'_t.
- $1/0 \leftarrow \mathsf{Verify}(M, \sigma'_t, t)$ is an algorithm which, on input a message-signature pair $(M$,$\sigma'_t)$ and a time t returns 1 or 0 for accept or reject, resp. If accept, the message-signature pair is *valid.*

Notions of Security. Security of KIRS schemes has three aspects: correctness, key-insulated and anonymity. Before giving their definition, we consider the following oracles which together model the ability of the adversaries in breaking the security of the schemes.

- $pk_i \leftarrow \mathcal{JO}(\perp)$. The *Joining Oracle*, on request, adds a new user to the system. It returns the public key $pk \in \mathcal{PK}$ of the new user.
- $sk_{i,t} \leftarrow \mathcal{KEO}(pk_i, t)$. The *Key Exposure Oracle*, on input a public key $pk_i \in \mathcal{PK}$ that is a query output of $\mathcal{JO}$ and a time t, returns the corresponding user secret key $usk_{i,t} \in \mathcal{USK}$ and the secret key $sk_{i,t} \in \mathcal{SK}$ for the time t.
- $\sigma'_t \leftarrow \mathcal{SO}(t, n, d, \mathcal{Y}, \mathcal{V}, M)$. The *Signing Oracle*, on input a time t, a group size n, a threshold $d \in \{1, \ldots, n\}$, a set $\mathcal{Y}$ of n public keys, a subset $\mathcal{V}$ of $\mathcal{Y}$ with $|\mathcal{V}| = d$, and a message M, returns a valid signature σ'_t for time t.

Correctness. Signatures signed according to specification are accepted during verification.

[4] We denote $usk_{i,t}$ to be the user secret key of user i at time t.

Key-Insulated. Key-Insulated for KIRS schemes is defined in the following game between the Simulator $\mathcal{S}$ and the Adversary $\mathcal{A}$ in which $\mathcal{A}$ is given access to oracles $\mathcal{JO}$, $\mathcal{KEO}$ and $\mathcal{SO}$:

1. $\mathcal{S}$ generates and gives $\mathcal{A}$ the system parameters param.
2. $\mathcal{A}$ chooses a time t, a group size $n \in \mathbb{N}$, a threshold $d \in \{1, \ldots, n\}$, a set $\mathcal{Y}$ of n public keys in $\mathcal{PK}$ and a message M.
3. $\mathcal{A}$ may query the oracles according to any adaptive strategy.
4. $\mathcal{A}$ outputs a signature σ_t.

$\mathcal{A}$ wins the game if: (1) $\mathsf{Verify}(M,\sigma_t,t)$=1, (2) all of the public keys in $\mathcal{Y}$ are query outputs of $\mathcal{JO}$, (3) at most $(d-1)$ of the public keys in $\mathcal{Y}$ have been input to $\mathcal{KEO}$ with time t to be the time input parameter, and (4) σ_t is not a query output of $\mathcal{SO}$ on any input containing M. We denote by $\mathbf{Adv}^{KI}_{\mathcal{A},\tau}(\lambda)$ the probability of $\mathcal{A}$ winning the game, for security parameter λ, if $\mathcal{A}$ is allowed to submit at most τ key exposure requests.

Definition 3 (Key-Insulated). *An KIRS scheme is key-insulated if for all PPT adversary $\mathcal{A}$, $\mathbf{Adv}^{KI}_{\mathcal{A},\tau}(\lambda)$ is negligible.*

Strong Key-Insulated. It is also possible for an adversary to compromise the physical secure device completely. In this case, the adversary does not query the key exposure oracle here in our model, but the adversary master is allowed to choose at most $d-1$ master secret keys, denoted by $\mathcal{MSK}_{d-1}$ which is simply given to him instead. We denote by $\mathbf{Adv}^{SKI}_{\mathcal{A},\tau}(\lambda, \mathcal{MSK}_{d-1})$ the probability of $\mathcal{A}$ winning the game.

Definition 4 (Strong Key-Insulated). *An KIRS scheme is strong key-insulated if for all PPT adversary $\mathcal{A}$, $\mathbf{Adv}^{SKI}_{\mathcal{A},\tau}(\lambda, \mathcal{MSK}_{d-1})$ is negligible.*

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

1. $\mathcal{S}$ generates and gives $\mathcal{A}$ the system parameters param.
2. $\mathcal{A}$ may query the oracles according to any adaptive strategy.
3. $\mathcal{A}$ gives $\mathcal{S}$ a time t, group size n, threshold $d \in \{1, \ldots, n\}$, message M, a set $\mathcal{Y}$ of n public keys all of which are query outputs of $\mathcal{JO}$, and none of which has been queried to $\mathcal{KEO}$.
   $\mathcal{S}$ randomly selects a subset $\mathcal{V} \subset \mathcal{Y}$, $|\mathcal{V}| = d$, to obtain the d corresponding secret keys by quering $\mathcal{KEO}$. $\mathcal{S}$ signs with these secret keys and gives the signature to $\mathcal{A}$.
4. $\mathcal{A}$ queries the oracles adaptively, except that any member public key of $\mathcal{Y}$ cannot be queried to $\mathcal{CO}$.
5. $\mathcal{A}$ gives $\mathcal{S}$ a publc key $\widetilde{pk} \in \mathcal{Y}$.

$\mathcal{A}$ wins the game if $\widetilde{pk} \in \mathcal{Y}$. Define the *advantage* of $\mathcal{A}$ as

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

for security parameter λ.

Definition 5 (KI-Anonymity). *A* KIRS *scheme is anonymous if for any PPT adversary* $\mathcal{A}$*,* $\mathbf{Adv}_{\mathcal{A}}^{KI-Anon}(\lambda)$ *is zero.*

Remarks. For anonymity, we require *unconditional anonymous.*

4 A Forward Secure Threshold Ring Signature Scheme

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

- $\mathsf{Key\text{-}Gen}$. First, we assume that the public key pairs are valid into T time periods and makes the time intervals public. For user i, where $i = 1, \ldots, n$, on input a security parameter k_i, ℓ_i, the algorithm randomly picks two distinct primes p_i, q_i such that $p_i = 3 \bmod 4$, $q_i = 3 \bmod 4$, $2^{k_i - 1} \le (p_i - 1)(q_i - 1)$ and $p_i q_i < 2^{k_i}$. Sets $N_i \leftarrow p_i q_i$. Let Q_i denote the set of non-zero quadratic residues modulo N_i.

  It then picks random generators $s_{i,0} \in_R \mathbb{Z}_{N_i}^*$ and computes $u_i \leftarrow 1/s_{i,0}^{2^{\ell_i(T+1)}} \bmod N_i$. It sets the public key to $pk_i \leftarrow (N_i, u_i, T)$, and the secret key to $sk_{i,0} \leftarrow (N_i, T, 0, s_{i,0})$. Finally it outputs $(sk_{i,0}, pk_i)$.

  Let ρ be twice the bit length of the largest N_i, for $1 \le i \le n$ and let $G : \{0,1\}^* \to \{0,1\}^{\rho}$ and $H_i : \{0,1\}^* \to \{0,1\}^{\ell_i}$, for $i = 1, \ldots, n$, be some hash functions which behave like a random oracle.
- Update. On input a secret key $sk_{i,j} = (N_i, T, j, s_{i,j})$ for time period j, output the secret key for time period $j+1$ as $sk_{i,j+1} \leftarrow (N_i, T, j+1, s_{i,j}^{2^{\ell_i}} \bmod N_i)$ if $j < T$, otherwise output $\perp$ meaning the secret key has expired.
- Sign. On input a group size $n \in \mathbb{Z}$, security parameters $(k_1, \ell_1, \ldots, k_n, \ell_n)$, a time period j, a threshold $d \in \{1, \ldots, n\}$, a public key set $\mathcal{L} = \{pk_1, \ldots, pk_n\}$, where each $pk_i = (N_i, u_i, T)$, a private key set $\mathcal{X} = \{sk_{\pi_1,j}, \ldots, sk_{\pi_d,j}\}$, where each $sk_{\pi_i,j} = (N_{\pi_i}, T, j, s_{\pi_i,j})$ (for time period j) corresponds to $pk_{\pi_i} \in \mathcal{L}$, $1 \le \pi_1, \ldots, \pi_d \le n$, and a message $m \in \{0,1\}^*$. Define $\mathcal{N} = \{1, \ldots, n\}$ and $\mathcal{I} = \{\pi_1, \ldots, \pi_d\} \subseteq \mathcal{N}$, the algorithm does the following:
  1. For $i \in \mathcal{N} \setminus \mathcal{I}$, pick $c_i \in_R \{0,1\}^{\rho}$ and $z_i \in_R \mathbb{Z}_{N_i}$. Compute

     $$y_i = z_i^{2^{\ell_i(T+1-j)}} u_i^{H_i(c_i)} \bmod N_i$$

  2. For $i \in \mathcal{I}$, pick $r_i \in_R \mathbb{Z}_{N_i}$ and compute

     $$y_i = r_i^{2^{\ell_i(T+1-j)}} \bmod N_i$$

3. Compute $c_0 = G(\mathcal{L}, d, j, m, y_1, \cdots, y_n)$ and construct a polynomial f over $GF(2^\rho)$ such that $\deg(f) = n - d$, $f(0) = c_0$ and $f(i) = c_i$, for $i \in \mathcal{N} \setminus \mathcal{I}$.
4. For $i \in \mathcal{I}$, compute $c_i = f(i)$ and
$$z_i = r_i s_{i,j}^{H_i(c_i)} \bmod N_i$$
5. Output the d-out-of-n forward secure threshold ring signature for message m, time period j and a public key list $\mathcal{L}$ as $\sigma = (z_1, \cdots, z_n, f, j)$.

- Verify. On input a message m, a list of public key $\mathcal{L}$, a signature σ, the algorithm runs as follow:
  1. Check if $\deg(f) = n - d$. If yes, proceed. Otherwise, reject.
  2. For $i = 1, \cdots, n$, compute $c_i = f(i)$ and
  $$y'_i = z_i^{2^{\ell_i(T+1-j)}} u_i^{H_i(c_i)} \bmod N_i$$
  3. Check whether $f(0) \overset{?}{=} G(\mathcal{L}, d, , j, m, y'_1, \cdots, y'_n)$. If yes, accept. Otherwise, reject.

Security Analysis.

Theorem 1. *The scheme proposed in this section is unconditional anonymous.*

Proof. The polynomial f, with degree n–t, is determined by $c_{d+1}, \cdots, c_n$ and c_0. $c_{d+1}, \cdots, c_n$ are randomly generated and c_0 is the output of the random oracle G. Thus f can be considered as a function chosen randomly from the collection of all polynomials over $GF(2^\rho)$ with degree $n - d$. Then the distributions of $c_1, \cdots, c_d$ are also uniform over the underlying range.

For $i = d + 1, \cdots, n$, z_i are chosen independently and distributed uniformly over $\mathbb{Z}_{N_i}$. For $i = 1, \cdots, t$, r_i are chosen independently and distributed uniformly over $\mathbb{Z}_{N_i}$. Since r_i are independent of c_i and the private keys, z_i, $1 \le i \le d$, are also uniformly distributed.

In addition, for any fixed message m and fixed set of public keys $\mathcal{L}$, we can see that $(z_1, \cdots, z_n)$ has exactly
$$\prod_{1 \le i \le n} N_i$$
possible solutions. Since the distribution of these possible solutions are independent and uniformly distributed no matter which t participating signers are, an adversary, even has all the private keys and unbound computing resources, has no advantage in identifying any one of the participating signers over random guessing. □

Theorem 2. *Let $\mathcal{A}$ be a PPT forger. For some message m and a set of n public keys L corresponding to n signers, suppose $\mathcal{A}$ on inputs the security parameter k_i, ℓ_i, for $1 \leq i \leq n$, the private keys of any $t-1$ signers among the n signers, queries a signing oracle $\mathcal{SO}$ for q_S times, random oracle G for q_G times and random oracles $\{H_i\}_{1 \leq i \leq n}$ for q_H times combined, and outputs a forged signature σ (i.e. $1 \leftarrow \mathcal{V}_{t,n}(L, m, \sigma)$), with non-negligible probability ϵ. Then we can factorize Blum integer with probability at least ϵ' in polynomial time, where*

$$\epsilon' = \frac{\left(\epsilon - q_S(q_H 2^{3-k} + q_G/2^{\rho})\right)^2}{2nT^2 q_G} - \frac{\epsilon - q_S(q_H 2^{3-k} + q_G/2^{\rho})}{2^{\ell+1} nT}$$

where $k = min\{k_1, \ldots, k_n\}$, $\rho = 2max\{k_1, \ldots, k_n\}$, $\ell = max\{\ell_1, \ldots, \ell_n\}$.

The proof is in Appendix A.

5 A Key-Insulated Threshold Ring Signature Scheme

In this section, we are going to propsoe a Key-Insulated Threshold Ring Signature scheme, in which a user is associated with a tamper-resistance device such that key updating process can be only taken place inside this device together with the secret input from the owner of this device.

- Key-Gen. For $i = 1, \ldots, n$, on input security parameters k_i, ℓ_i, the algorithm randomly picks two distinct safe primes p'_i, q'_i and compute $p_i = 2p'_i + 1, q_i = 2q'_i + 1, N_i = p_i q_i$ such that N_i is a k_i-bit modulus. Choose another $(\ell_i + 1)$-bit prime number v_i. Let $\kappa = min\{k_1, \ldots, k_n\}$. We assume that the public key pairs are valid into T time periods, where $T = \xi(\kappa)$ for some polynomial ξ, and makes the time intervals public.

  Device i randomly chooses $s_i, t_i, u_i \in_R \mathbb{Z}^*_{N_i}$, such that $s_i^2 \neq s_i^{2^{8+1}} \bmod N_i$, $t_i^2 \neq t_i^{2^{8+1}} \bmod N_i$, $u_i^2 \neq u_i^{2^{8+1}} \bmod N_i$. Compute $\alpha_i = s_i^{-v_i} \bmod N_i$, $\beta_i = t_i^{-v_i} \bmod N_i$ and $\gamma_i = u_i^{-v_i} \bmod N_i$ and sets and outputs the public key to $pk_i \leftarrow (\alpha_i, \beta_i, \gamma_i, v_i, N_i)$. It also computes $\delta_i = s_i^2 \bmod N_i$, $\mu_i = t_i^2 \bmod N_i$ and sets the master secret key $msk_i \leftarrow (\delta_i, \mu_i)$. It also computes $\psi_{i,0} = u_i^{2^{0+1}} \bmod N_i$ and sets and outputs the user's secret key $usk_{i,0} \leftarrow \psi_{i,0}$. Then it deletes ψ_i from its memory.

  Let ρ be twice the bit length of the largest N_i, for $1 \leq i \leq n$ and let $G : \{0,1\}^* \rightarrow \{0,1\}^{\rho}$ and $H_i : \{0,1\}^* \rightarrow \{0,1\}^{\ell_i}$ be some hash functions that behave like a random oracle.
- Device-Update. Device i, on input master secret key $msk_i = (\delta_i, \mu_i)$, computes the partial secret key for the j-th time period as follow:

$$psk_{i,j} = \delta_i^{2^j} \cdot \mu_i^{2^{T-j}} \bmod N_i = s_i^{2^{j+1}} \cdot t_i^{2^{T+1-j}} \bmod N_i$$

- User-Update. User i, on input user's secret key $usk_{i,j-1}$ for time period $j-1$, he computes the user's secret key for time period j as follow:

$$usk_{i,j} = \psi_{i,j-1}^2 \bmod N_i = u_i^{2^{j+1}} \bmod N_i$$

and the corresponding secret key $sk_{i,j}$ to be

$$\psi'_{i,j} = psk_{i,j} \cdot usk_{i,j} \bmod N_i = s_i^{2^{j+1}} \cdot t_i^{2^{T+1-j}} \cdot u_i^{2^{j+1}} \bmod N_i$$

- Sign. On input a group size $n \in \mathbb{Z}$, security parameters $(k_1, \ell_1, \ldots, k_n, \ell_n)$, a time period j, a threshold $d \in \{1, \ldots, n\}$, a public key set $\mathcal{L} = \{pk_1, \ldots, pk_n\}$, where $pk_i = (\alpha_i, \beta_i, \gamma_i, v_i)$, a private key set $\mathcal{X} = \{sk_{\pi_1,j}, \ldots, sk_{\pi_d,j}\}$ (for time period j) corresponds to $pk_{\pi_i} \in \mathcal{L}$, $1 \le \pi_1, \ldots, \pi_d \le n$, where $sk_{i,j} = \psi'_i$, and a message $m \in \{0,1\}^*$. Define $\mathcal{N} = \{1, \ldots, n\}$ and $\mathcal{I} = \{\pi_1, \ldots, \pi_d\} \subseteq \mathcal{N}$, the algorithm does the following:
  1. For $i \in \mathcal{N} \setminus \mathcal{I}$, pick $c_i \in_R \{0,1\}^\rho$ and $z_i \in_R \mathbb{Z}_{N_i}$. Compute
  $$y_i = z_i^{v_i} (\alpha_i^{2^{j+1}} \beta_i^{2^{T+1-j}} \gamma_i^{2^{j+1}})^{H_i(c_i)} \bmod N_i$$
  2. For $i \in \mathcal{I}$, pick $r_i \in_R \mathbb{Z}_{N_i}$ and compute
  $$y_i = r_i^{v_i} \bmod N_i$$
  3. Compute $c_0 = G(\mathcal{L}, d, j, m, y_1, \cdots, y_n)$ and construct a polynomial f over $GF(2^\rho)$ such that $\deg(f) = n - d$, $f(0) = c_0$ and $f(i) = c_i$, for $i \in \mathcal{N} \setminus \mathcal{I}$.
  4. For $i \in \mathcal{I}$, compute $c_i = f(i)$ and
  $$z_i = r_i (\psi'_{i,j})^{H_i(c_i)} \bmod N_i$$
  5. Output the d-out-of-n forward secure threshold ring signature for message m, time period j and a public key list $\mathcal{L}$ as $\sigma = (z_1, \cdots, z_n, f, j)$.
- Verify. On input a message m, a list of public key $\mathcal{L}$, a signature σ, the algorithm runs as follow:
  1. Check if $\deg(f) = n - d$. If yes, proceed. Otherwise, reject.
  2. For $i = 1, \cdots, n$, compute $c_i = f(i)$ and
  $$y'_i = z_i^{v_i} (\alpha_i^{2^{j+1}} \beta_i^{2^{T+1-j}} \gamma_i^{2^{j+1}})^{H_i(c_i)} \bmod N_i$$
  3. Check whether $f(0) \stackrel{?}{=} G(\mathcal{L}, d, j, m, y'_1, \cdots, y'_n)$. If yes, accept. Otherwise, reject.

Security Analysis.

Theorem 3. *The scheme proposed in this section is unconditional anonymous.*

The proof is similar to the proof of Theorem 1 and we skip it.

Theorem 4. *Let $\mathcal{A}$ be a PPT forger. For some message m and a set of n public keys L corresponding to n signers, suppose $\mathcal{A}$ on inputs the security parameter k, all n master secret keys and any $t-1$ user secret keys among the n signers, queries a signing oracle $\mathcal{SO}$ for q_S times, random oracle G for q_G times and random oracles $\{H_i\}_{1 \le i \le n}$ for q_H times combined, and outputs a forged signature σ (i.e. $1 \leftarrow \mathcal{V}_{t,n}(L, m, \sigma)$), with non-negligible probability ϵ. Then we can solve the strong RSA problem with non-negligible probability in polynomial time.*

The proof is in Appendix B.

6 Conclusion

In this paper, we have suggested a some solutions to the key exposure problem in ring signature. We propose the first forward secure ring signature scheme and the first key-insulated ring signature scheme. Both of them allow a (t, n) threshold setting. We have proven their security in the random oracle model.

However, the size of the signature in both scheme grows linear with the number of users. It is an interesting open problem to construct a forward secure ring signature scheme or key-insulated ring signature scheme with a constant size to the number of users.

References

1. M. Abdalla, S. Miner, and C. Namprempre. Forward-secure threshold signature schemes. In *CT-RSA 2001*, volume 2020 of *LNCS*, pages 441–456. Springer-Verlag, 2001.
2. M. Abdalla and L. Reyzin. A new forward-secure digital signature scheme. In *Asiacrypt '00*, volume 1976 of *LNCS*, pages 116–129. Springer-Verlag, 2000.
3. M. Abdalla and L. Reyzin. A new forward-secure digital signature scheme. Cryptology ePrint Archive, Report 2000/002, 2000. http://eprint.iacr.org/ (extended abstract is in Asiacrypt 00).
4. M. Abe, M. Ohkubo, and K. Suzuki. 1-out-of-n signatures from a variety of keys. In *ASIACRYPT 2002*, pages 415–432. Springer-Verlag, 2002.
5. R. Anderson. Two remarks on public-key cryptology. Manuscript, Sep. 2000. Relevant material presented by the author in an invited lecture at the Fourth ACM Conference on Computer and Communications Security, 1997.
6. M. Bellare, D. Micciancio, and B. Warinschi. Foundations of group signatures: formal definitions, simplified requirements and a construction based on general assumptions. In *EUROCRYPT'03*, volume 2656 of *LNCS*. Springer-Verlag, 2003.
7. M. Bellare and S. Miner. A forward-secure digital signature scheme. In *Crypto'99*, volume 1666 of *LNCS*, pages 431–448. Springer-Verlag, 1999.
8. D. Boneh, C. Gentry, B. Lynn, and H. Shacham. Aggregate and verifiably encrypted signatures from bilinear maps. In *EUROCRYPT 2003*, LNCS, pages 416–432. Springer-Verlag, 2003.
9. E. Bresson, J. Stern, and M. Szydlo. Threshold ring signatures and applications to ad-hoc groups. In *Crypto'02*, volume 2442 of *LNCS*, pages 465–480. Springer-Verlag, 2002.
10. J. Camenisch and M. Stadler. Efficient group signature schemes for large groups (extended abstract). In *CRYPTO'97*, pages 410–424. Springer-Verlag, 1997.
11. D. Chaum and E. van Heyst. Group signatures. In *EUROCRYPT'91*, volume 547 of *LNCS*, pages 257–265. Springer-Verlag, 1991.
12. Y. Dodis, J. Katz, S. Xu, and M. Yung. Strong key-insulated public key cryptosystems. In *EUROCRYPT 02*, volume 2332 of *LNCS*, pages 65–82. Springer-Verlag, 2002.
13. Y. Dodis, J. Katz, S. Xu, and M. Yung. Strong key-insulated signature schemes. In *PKC 2003*, volume 2567 of *LNCS*, pages 130–144. Springer-Verlag, 2003.
14. 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.

15. N. Gonzalez-Deleito, O. Markowitch, and E. Dall'Olio. A new key-insulated signature scheme. In *ICICS 2004*, volume 3269 of *LNCS*, pages 465–479. Springer-Verlag, 2004.
16. G. Itkis and L. Reyzin. Forward-secure signatures with optimal signing and verifying. In *CRYPTO '01*, pages 332–354. Springer, 2001. Lecture Notes in Computer Science No. 2139.
17. H. Krawczyk. Simple forward-secure signatures from any signature scheme. In *Proceedings of the 7th ACM conference on Computer and communications Security*, pages 108–115. ACM Press, 2000.
18. J. K. Liu, V. K. Wei, and D. S. Wong. A separable threshold ring signature scheme. In *ICISC 2003*, volume 2971 of *LNCS*, pages 12–26. Springer-Verlag, 2003.
19. T. Malkin, D. Micciancio, and S. Miner. Efficient generic forward-secure signatures with an unbounded number of time periods. In *Eurocrypt '02*, volume 2332 of *LNCS*, pages 400–417. Springer-Verlag, 2002.
20. K. Ohta and T. Okamoto. On concrete security treatment of signatures derived from identification. In *Crypto '98*, pages 354–369. Springer-Verlag, 1998. Lecture Notes in Computer Science No. 1462.
21. R. L. Rivest, A. Shamir, and Y. Tauman. How to leak a secret. In *ASIACRYPT 2001*, pages 552–565. Springer-Verlag, 2001.
22. D. X. Song. Practical forward secure group signature schemes. In *Proceedings of the 8th ACM conference on Computer and communications Security*, pages 225–234. ACM Press, 2001.
23. D. S. Wong, K. Fung, J. K. Liu, and V. K. Wei. On the RS-code construction of ring signature schemes and a threshold setting of RST. In *ICISC 2003*, volume 2971 of *LNCS*, pages 34–46. Springer-Verlag, 2003.
24. F. Zhang and K. Kim. ID-Based blind signature and ring signature from pairings. In *ASIACRYPT 2002*, pages 533–547. Springer-Verlag, 2002. Lecture Notes in Computer Science No. 2501.

A Proof of Theorem 2

Proof. Let $\mathcal{A}$ be a PPT adversary who can forge signatures with non-negligible probability at least ϵ when given , n public keys and strictly less than t of the corresponding private key. Assume $\mathcal{A}$ makes q_G queries to G, q_S queries to the signing oracle $\mathcal{SO}$, and a total of q_H queries to $H_1, \cdots, H_n$ combined. We constrct another PPT $\mathcal{M}$ from $\mathcal{A}$ to factorize a given a Blum integer N.

In order to factor its input N, $\mathcal{M}$ randomly selects $x \in \mathbb{Z}_\eta^*$, computes $v = x^2 \bmod N$, and attempt to use $\mathcal{A}$ to find a square root y of v. Because v has four square roots and x is random, with probability $1/2$ we have $x \neq \pm y \bmod N$, then $\mathcal{M}$ is able to find a factor of N by computing the gcd of $x - y$ and N.

We define $\mathcal{T}$ to be the $\mathsf{breakin}$ period such that $\mathcal{A}$ is allowed to query $\mathcal{SO}$ to obtain at most $d - 1$ private key with time input parameter $t' < \mathcal{T}$ while there is no limitation for time input parameter $t'' \geq \mathcal{T}$. $\mathcal{A}$ is also allowed to choose any $\mathcal{T} \leq T$. $\mathcal{M}$ provides the corresponding private key as a reply to the query to the $\mathcal{SO}$ made by $\mathcal{A}$.

$\mathcal{M}$ needs to guess the $\mathsf{breakin}$ period $\mathcal{T}$ chosen by $\mathcal{A}$. $\mathcal{M}$ randomly chooses t, $1 < t \leq T$, hoping that the $\mathsf{breakin}$ period falls at t or later, so that the forgery will be for a time period earlier than t.

$\mathcal{M}$ also needs to assign N to be the public key of one of the n users and provide all public key to $\mathcal{A}$. $\mathcal{M}$ just randomly chooses $\pi \in_R \{1, \ldots, n\}$ and sets $u_\pi \leftarrow 1/v^{2^{\ell_\pi (T+1-t)}}$, $N_\pi \leftarrow N$. The other $n-1$ public keys are generated in the normal way. $\mathcal{M}$ provides these n public keys to $\mathcal{A}$.

Besides, $\mathcal{M}$ also simulates $\mathcal{A}$'s point of view by constructing the random oracle G and the signing oracle $\mathcal{SO}$. We first describe the construction of the signing oracle $\mathcal{SO}$. On input a time b, a group size n, a threshold $d \in \{1, \ldots, n\}$, a set $\mathcal{Y}$ of n public keys, a subset $\mathcal{V}$ of $\mathcal{Y}$ with $|\mathcal{V}| = d$, and a message M, the answer is simulated as follow:

1. Randomly generate $c_0, c_{d+1}, \cdots, c_n \in_R \{0,1\}^\rho$.
2. Construct f over $GF(2^\rho)$ such that $\deg(f) = n - d$ and $f(0) = c_0, f(i) = c_i$, for $i = d+1, \cdots, n$.
3. Compute $c_1 = f(1), \cdots, c_d = f(d)$.
4. Randomly generate $z_i \in_R \mathbb{Z}_{N_i}$ for $i = 1, \cdots, n$.
5. Compute $y_i = z_i^{2^{\ell_i (T+1-j)}} u_i^{H_i(c_i)} \bmod N_i$ for $i = 1, \cdots, n$.
6. Assign c_0 as the value of $G(\mathcal{L}, d, j, , m, y_1, \cdots, y_n)$.
7. Output $(z_1, \cdots, z_n, f, b)$.

The simulation fails if step 6 causes collision, that is, the value of c_0 has been assigned before. This happens with probability at most $q_G/2^\rho$ where q_G is the number of times that the random oracle G is queried by $\mathcal{A}$.

Let Θ, Ω be the random tapes given to the signing oracle and $\mathcal{A}$ such that $\mathcal{A}$ outputs a forged signature. Notice that the success probability of $\mathcal{A}$ is taken over the space defined by Θ, Ω and the random oracle G.

Assume $\mathcal{A}$ chooses a breakin period $\mathcal{T} \geq t$. That is, the forged signature σ_j is valid for time period $j < t$. The forged signature $\sigma_j = (z_1, \cdots, z_n, f, j)$ contains a polynomial f where $f(0) = G(\mathcal{L}, d, j, m, y_1, \cdots, y_n)$ for $y_i = z_i^{2^{\ell_i (T+1-j)}} u_i^{H_i(c_i)} \bmod N_i, 1 \leq i \leq n$. With probability at least $1 - 2^{-\rho}$, there exists a query $G(\mathcal{L}, d, j, , m, y_1, \cdots, y_n)$ due to the assumption of ideal randomness of G. Split G as (G^-, c_0) where G^- corresponds to the answers to all G-queries except for c_0. Rewind $\mathcal{A}$ to this particular point and by invoking $\mathcal{A}$ with (Θ, Ω, G^-) and randomly chosen another value c_0' $(\neq c_0)$ as the reply to the random oracle query, $\mathcal{A}$ outputs at least one forged signature $\sigma_j' = (z_1', \cdots, z_n', f', j)$ with non-negligible probability, due to the heavy-row lemma [20].

Since the random tape is the same for both forged signature, we have y_π in σ_j should be equal to y_π' in σ_j'. That is,

$$\begin{aligned} z_\pi^{2^{\ell_i (T+1-j)}} u_\pi^{H_\pi(f(\pi))} &\equiv {z'}_\pi^{2^{\ell_i (T+1-j)}} u_\pi^{H_\pi(f'(\pi))} \pmod{N_\pi} \Rightarrow \\ \left(v^{-2^{\ell_\pi (T+1-t)}} \right)^{H_\pi(f(\pi)) - H_\pi(f'(\pi))} &\equiv (z_\pi'/z_\pi)^{2^{\ell_\pi (T+1-j)}} \pmod{N_\pi} \Rightarrow \\ v^{H_\pi(f(\pi)) - H_\pi(f'(\pi))} &\equiv (z_\pi/z_\pi')^{2^{\ell_\pi (t-j)}} \pmod{N_\pi} \end{aligned}$$

By applying Lemma A.1 in [3], $\mathcal{M}$ can easily compute a square root of v, by stting $\alpha = H_\pi(f(\pi)) - H_\pi(f'(\pi))$, $X = z_\pi/z_\pi'$ and $\lambda = \ell_\pi(t-j)$. We stay the lemma below, without proof.

Lemma 1. *Given $\alpha \neq 0$, $\lambda > 0$, $v \in Q_\pi$ and $X \in \mathbb{Z}^*_{N_\pi}$ such that $v^\alpha = X^{2^\lambda} \bmod N_\pi$ and $\alpha < 2^\lambda$, one can easily compute y such that $v = y^2 \bmod N_\pi$.*

Next we are going to analysis the successfuly probability.

First we consider the probability that collision to the hash query occurs. Let $Q = \min\{|Q_1|, \ldots, |Q_n|\}$. The probability of collision occur in the same execution of $\mathcal{A}$ is at most $q_H/Q + q_G/2^\rho$. Thus, the probability of $\mathcal{M}$ failure to simulate the signing oracle is at most $q_S(q_H/Q + q_G/2^\rho) \leq q_S(q_H 2^{3-k} + q_G/2^\rho)$, where $k = \min\{k_1, \ldots, k_n\}$. Let

$$\delta = \epsilon - q_S(q_H 2^{3-k} + q_G/2^\rho)$$

Let ϵ_t be the probability that $\mathcal{A}$ produces a successful forgery such that the break-in query occurs in time period t. Observe that $\delta = \sum_{t=1}^{T} \epsilon_t$. Assume $\mathcal{M}$ picked a specific t as the time period for v. The probability is $1/T$.

Let $p_{h,t}$ be the probability that, in one run, $\mathcal{A}$ produces a valid forgery based on hash query number h after break-in query in time period t. We have

$$\epsilon_t = \sum_{h=1}^{q_G} p_{h,t}$$

The probability that $\mathcal{A}$ produces a valid forgery based on the hash query number h after break-in query in time period t in both runs is $p^2_{h,t}$. This is reduced to $p_{h,t}(p_{h,t} - 2^{-\ell})$ due to the collision probability.

At this stage, we have to apply another lemma, lemma A.2 from [3]:

Lemma 2. *Let $a_1, \ldots, a_\lambda$ be real numbers. Let $a = \sum_{\mu=1}^{\lambda} a_\mu$. Let $s = \sum_{\mu=1}^{\lambda} a^2_\mu$. Then $s \leq a^2/\lambda$.*

Now we have the probability that $\mathcal{A}$ outputs a valid forgery based on the same hash query both times and that the hash query was answered differently in the second run and the break-in query occured in time period t to be

$$\sum_{h=1}^{q_G} p^2_{h,t} - \sum_{h=1}^{q_G} 2^{-\ell} p_{h,t} \geq \frac{\epsilon_t^2}{q_G} - \sum_{h=1}^{q_G} 2^{-\ell} p_{h,t} = \frac{\epsilon_t^2}{q_G} - 2^{-\ell}\epsilon_t$$

(by using Lemma 2.)

Finally, we sum up all time period t to obtain

$$\epsilon' \geq \frac{1}{T} \sum_{t=1}^{T} \left(\frac{\epsilon_t^2}{q_G} - 2^{-\ell}\epsilon_t \right) \geq \frac{\delta^2}{T^2 q_G} - \frac{\delta}{2^\ell T}$$

(by using Lemma 2.)

Finally we divide the result by 2 because only half of the choice for x that can be used to factorize N and also divide by n since $\mathcal{M}$ has to guess which of the user that $\mathcal{A}$ is going to participate for the forgery. □

B Proof of Theorem 4

Proof. Let $\mathcal{A}$ be a PPT adversary who can forge signatures with non-negligible probability at least ϵ when given , n public keys, n corresponding master secret keys and strictly less than t of the corresponding user private keys. Assume $\mathcal{A}$ makes q_G queries to G, q_S queries to the signing oracle $\mathcal{SO}$, and a total of q_H queries to $H_1, \cdots, H_n$ combined. We constrct another PPT $\mathcal{M}$ from $\mathcal{A}$ to solve the strong RSA problem. That is, given a number N, which is the product of two primes, and a number $\lambda \in \mathbb{Z}_N^*$, outputs $\phi \in \mathbb{Z}_N^*$ and $\omega > 1$ such that $\phi^\omega = \lambda \bmod N$.

$\mathcal{M}$ needs to assign N to be the public key of one of the n users and provide all public key to $\mathcal{A}$. $\mathcal{M}$ just randomly chooses $\pi \in_R \{1, \ldots, n\}$ and does the following: $\alpha_\pi, \beta_\pi, \gamma_\pi, v_\pi$ such that

1. Choose $\alpha_\pi, \beta_\pi, \gamma_\pi$ such that $\lambda = \alpha_\pi \cdot \beta_\pi \cdot \gamma_\pi$.
2. Randomly choose $j \in_R \{1, T\}$.
3. Choose v_π such that $\gcd(v_\pi, 2^{j+1} - 1, 2^{T+1-j} - 1) = v_\pi$. That is, $2^{j+1} - 1 = v_\pi K_1$ and $2^{T+1-j} - 1 = v_\pi K_2$ for some integers K_1, K_2.
4. If no such v_π exists, repeat step 3.

The total running time should be in polynomial of κ, the system security parameter, since $T = \xi(\kappa)$ for some polynomial ξ.

The other $n-1$ public keys and n master secret keys are generated in the normal way. $\mathcal{M}$ provides these n public keys to $\mathcal{A}$.

Besides, $\mathcal{M}$ also simulates $\mathcal{A}$'s point of view by constructing the random oracle G and the signing oracle $\mathcal{SO}$. We first describe the construction of the signing oracle $\mathcal{SO}$. On input a time b, a group size n, a threshold $d \in \{1, \ldots, n\}$, a set $\mathcal{Y}$ of n public keys, a subset $\mathcal{V}$ of $\mathcal{Y}$ with $|\mathcal{V}| = d$, and a message M, the answer is simulated as follow:

1. Randomly generate $c_0, c_{d+1}, \cdots, c_n \in_R \{0,1\}^\rho$.
2. Construct f over $GF(2^\rho)$ such that $\deg(f) = n - d$ and $f(0) = c_0, f(i) = c_i$, for $i = d+1, \cdots, n$.
3. Compute $c_1 = f(1), \cdots, c_d = f(d)$.
4. Randomly generate $z_i \in_R \mathbb{Z}_{N_i}$ for $i = 1, \cdots, n$.
5. Compute $y_i' = z_i^{v_i}(\alpha_i^{2^{j+1}} \beta_i^{2^{T+1-j}} \gamma_i^{2^{j+1}})^{H_i(c_i)} \bmod N_i$.
6. Assign c_0 as the value of $G(\mathcal{L}, b, j, , m, y_1, \cdots, y_n)$.
7. Output $(z_1, \cdots, z_n, f, b)$.

The simulation fails if step 6 causes collision, that is, the value of c_0 has been assigned before. This happens with probability at most $q_G/2^\rho$ where q_G is the number of times that the random oracle G is queried by $\mathcal{A}$.

Let Θ, Ω be the random tapes given to the signing oracle and $\mathcal{A}$ such that $\mathcal{A}$ outputs a forged signature. Notice that the success probability of $\mathcal{A}$ is taken over the space defined by Θ, Ω and the random oracle G.

Assume $\mathcal{A}$ chooses a period j. That is, the forged signature σ_j is valid for time j. The forged signature $\sigma_j = (z_1, \cdots, z_n, f, j)$ contains a polynomial f where

$f(0) = G(\mathcal{L}, d, j, , m, y_1, \cdots, y_n)$ for $y_i = z_i^{v_i}(\alpha_i^{2^{j+1}} \beta_i^{2^{T+1-j}} \gamma_i^{2^{j+1}})^{H_i(c_i)} \bmod N_i$, $1 \le i \le n$. With probability at least $1 - 2^{-\rho}$, there exists a query $G(\mathcal{L}, d, j, m, y_1, \cdots, y_n)$ due to the assumption of ideal randomness of G. Split G as (G^-, c_0) where G^- corresponds to the answers to all G-queries except for c_0. Rewind $\mathcal{A}$ to this particular point and by invoking $\mathcal{A}$ with (Θ, Ω, G^-) and randomly chosen another value c_0' $(\neq c_0)$ as the reply to the random oracle query, $\mathcal{A}$ outputs at least one forged signature $\sigma_j' = (z_1', \cdots, z_n', f', j)$ with non-negligible probability, due to the heavy-row lemma [20].

Since the random tape is the same for both forged signature, we have y_π in σ_j should be equal to y_π' in σ_j'. That is,

$$\begin{aligned} z_\pi^{v_\pi}(\alpha_\pi^{2^{j+1}} \beta_\pi^{2^{T+1-j}} \gamma_\pi^{2^{j+1}})^{H_\pi(f(\pi))} &= {z'}_\pi^{v_\pi}(\alpha_\pi^{2^{j+1}} \beta_\pi^{2^{T+1-j}} \gamma_\pi^{2^{j+1}})^{H_\pi(f'(\pi))} \\ &\pmod{N_\pi} \Rightarrow \\ \left(\frac{z_\pi}{z_\pi'}\right)^{v_\pi} &= (\alpha_\pi^{2^{j+1}} \beta_\pi^{2^{T+1-j}} \gamma_\pi^{2^{j+1}})^{H_\pi(f'(\pi)) - H_\pi(f(\pi))} \\ &\pmod{N_\pi} \end{aligned}$$

Since v_π is a prime number, we have $\gcd\Big(v_\pi,\ H_\pi(f'(\pi)) - H_\pi(f(\pi))\Big) = 1$. Thus we can find two integers a and b such that

$$a v_\pi + b\Big(H_\pi(f'(\pi)) - H_\pi(f(\pi))\Big) = 1$$

and compute

$$\begin{aligned} \alpha_\pi^{2^{j+1}} \beta_\pi^{2^{T+1-j}} \gamma_\pi^{2^{j+1}} &= (\alpha_\pi^{2^{j+1}} \beta_\pi^{2^{T+1-j}} \gamma_\pi^{2^{j+1}})^{a v_\pi} \\ &\quad \cdot (\alpha_\pi^{2^{j+1}} \beta_\pi^{2^{T+1-j}} \gamma_\pi^{2^{j+1}})^{b(H_\pi(f'(\pi)) - H_\pi(f(\pi)))} \pmod{N_\pi} \Rightarrow \\ &= (\alpha_\pi^{2^{j+1}} \beta_\pi^{2^{T+1-j}} \gamma_\pi^{2^{j+1}})^{a v_\pi} \\ &\quad \cdot \left(\frac{z_\pi}{z_\pi'}\right)^{b(H_\pi(f'(\pi)) - H_\pi(f(\pi)))} \pmod{N_\pi} \Rightarrow \\ &= \left((\alpha_\pi^{2^{j+1}} \beta_\pi^{2^{T+1-j}} \gamma_\pi^{2^{j+1}})^{a} \cdot \left(\frac{z_\pi}{z_\pi'}\right)^{b}\right)^{v_\pi} \pmod{N_\pi} \qquad (1) \end{aligned}$$

Let

$$\lambda' = \alpha_\pi^{2^{j+1}} \beta_\pi^{2^{T+1-j}} \gamma_\pi^{2^{j+1}} \qquad (2)$$

From equation (2), equation (1) becomes

$$\lambda' = \left(\lambda'^a \cdot \left(\frac{z_\pi}{z_\pi'}\right)^{b}\right)^{v_\pi} \pmod{N_\pi} \qquad (3)$$

From equation (2), we also have

$$\begin{aligned}\lambda' &= (\alpha_\pi \alpha_\pi^{2^{j+1}-1})(\beta_\pi \beta_\pi^{2^{T+1-j}-1})(\gamma_\pi \gamma_\pi^{2^{j+1}-1})\\ &= (\alpha_\pi \beta_\pi \gamma_\pi)(\alpha_\pi^{2^{j+1}-1}\beta_\pi^{2^{T+1-j}-1}\gamma_\pi^{2^{j+1}-1})\\ &= \lambda(\alpha_\pi^{2^{j+1}-1}\beta_\pi^{2^{T+1-j}-1}\gamma_\pi^{2^{j+1}-1})\\ \therefore \lambda &= \frac{\lambda'}{\alpha_\pi^{2^{j+1}-1}\beta_\pi^{2^{T+1-j}-1}\gamma_\pi^{2^{j+1}-1}}\end{aligned}$$

We multiply $\frac{1}{\alpha_\pi^{2^{j+1}-1}\beta_\pi^{2^{T+1-j}-1}\gamma_\pi^{2^{j+1}-1}}$ to both sides of equation (3), to get

$$\begin{aligned}\lambda &= \frac{\left(\lambda'^a \cdot \left(\frac{z_\pi}{z'_\pi}\right)^b\right)^{v_\pi}}{\alpha_\pi^{2^{j+1}-1}\beta_\pi^{2^{T+1-j}-1}\gamma_\pi^{2^{j+1}-1}} \pmod{N_\pi}\\ &= \frac{\left(\lambda'^a \cdot \left(\frac{z_\pi}{z'_\pi}\right)^b\right)^{v_\pi}}{\alpha_\pi^{K_1 v_\pi}\beta_\pi^{K_2 v_\pi}\gamma_\pi^{K_1 v_\pi}} \pmod{N_\pi}\\ &= \frac{\left(\lambda'^a \cdot \left(\frac{z_\pi}{z'_\pi}\right)^b\right)^{v_\pi}}{(\alpha_\pi^{K_1}\beta_\pi^{K_2}\gamma_\pi^{K_1})^{v_\pi}} \pmod{N_\pi}\\ &= \left(\frac{\lambda'^a \cdot \left(\frac{z_\pi}{z'_\pi}\right)^b}{\alpha_\pi^{K_1}\beta_\pi^{K_2}\gamma_\pi^{K_1}}\right)^{v_\pi} \pmod{N_\pi} \qquad (4)\end{aligned}$$

From equation (4), by letting $\phi = \frac{\lambda'^a \cdot \left(\frac{z_\pi}{z'_\pi}\right)^b}{\alpha_\pi^{K_1}\beta_\pi^{K_2}\gamma_\pi^{K_1}}$ and $\omega = v_\pi$, $\mathcal{M}$ can solve the strong RSA problem for λ. □