A k-out-of-n Ring Signature with Flexible Participation for Signers

Takeshi Okamoto[1], Raylin Tso[2], Michitomo Yamaguchi[3], and Eiji Okamoto[4]

[1] Graduate School of Technology and Sciences, Tsukuba University of Technology, Japan ken@cs.k.tsukuba-tech.ac.jp
[2] Department of Computer Science, National Chengchi University, Taiwan raylin@cs.nccu.edu.tw
[3] Graduate School of Technology and Sciences, Tsukuba University of Technology, Japan ninjin.3san@gmail.com
[4] University of Tsukuba, Japan okamoto@risk.tsukuba.ac.jp

Abstract. A k-out-of-n ring signature is a kind of anonymous signature that can be performed by any member in a group. This signature allows the creation of valid signatures if and only if actual signers more than or equal to k sign the message among n possible signers. In this paper, we present a new k-out-of-n ring signature. Our signature has a remarkable property: When the signature is updated from k-out-of-n to $(k+\alpha)$-out-of-n, the previous signers do not need to sign a message again. Our scheme can "reuse" the old signature, whereas the previous schemes revoke it and create a signature from scratch. We call this property "flexibility" and formalize it rigorously. Our signature scheme has a multiple ring structure, each ring of which is based on 1-out-of-n ring signature. The structure of our scheme is completely different from that of conventional schemes, such as a secret-sharing type. The signers' keys are mostly independent of each user, thanks to a part of keys which use a special hash function. We give the results of provable security for our scheme.

Keywords: anonymity · ring signature · k-out-of-n property · flexible participation.

1 Introduction

1.1 Background

Anonymity is a great concern in many cryptographic applications. Ring signature [19] is a primitive which provides the signer's anonymity. In the framework of ring signature, a verifier can be convinced that a signature was produced by an anonymous signer among n possible signers.

A k-out-of-n ring signature [7] is a extended type from the conventional ring signature in [19]. This satisfies the following properties.

1. **Anonymity.** For a given signature, it is hard to identify an actual signer whereas the signer exists in n members that each have keys.
2. ***k*-out-of-*n* property.** A verifier can only be convinced that a signature was produced by the collaboration of at least k anonymous signers among n possible signers.

For this reason, we assume that ring signature is the same as 1-out-of-n ring signature in this paper.

Possible applications of this signature are anonymous petition, voting [12] and whistle blowing [19]. In that application, the signers can provide the evidence that how many people approve the message, while the signers preserve anonymity.

Note that in the case of k-out-of-n ring signature, the information related to the number for actual signers is important. For example, if someone applies this to the prosecution in a court case using this signature as evidence, the verifier can confirm that "at least k signers approve the message". Therefore, such information gives the prosecution accuracy and reliability.

1.2 Previous Work

Informal notions of ring signatures were discussed simultaneously with the appearance of group signatures by [10], [11]. Bresson et al. [7] extend the ring signature scheme into a k-out-of-n threshold ring signature scheme.

k-out-of-n signature has been widely studied. A separable threshold ring signature was proposed by Liu et al. [14] and Abe et al. [2]. It enables possible signers to use variety of keys such as RSA and Schnorr. An individual-linkability for threshold ring signature was proposed by Tsang et al. [24]. In the scheme, anyone can find out if two ring signatures are signed with the help of the same signer. Fujisaki and Suzuki proposed some variation of the k-out-of-n threshold ring signature scheme [12]. The verifier can be convinced that the signatures were produced by the collaboration of at most/at least or exact k anonymous signers. They also proposed a traceable ring signature, which restricts excessive anonymity [13].

A threshold ring signature without random oracle was proposed by Yuen et al. [27]. This is the threshold extension of the Shacham-Waters ring signature [22], thus it needs a setup algorithm in their protocol. A multivariate based threshold ring signature was proposed by Petzoldt et al. [18]. They extended the work of Sakumoto [20] into a threshold ring signature.

Subsequently, different types of setting or construction have been proposed such as ID-based [23], certificateless-based [9], code-based [3] and lattice-based [8].

1.3 Motivation

In some applications, the number of signers quite naturally increases over time gradually. Unfortunately, if the signatures in the previous works are applied for such applications, the following serious problem occurs.

Update Problem. Let us give a scenario to explain the context of our problem clearly. Suppose some people who live in a city, want to express their approval of the city's proposed regulation. They create a k-out-of-n ring signature and send it to the city's office. Here n is the number of people who belong to a community. One month later, the supporters have increased by α people. This time they want to update the signature from k-out-of-n to $(k+\alpha)$-out-of-n ring signatures. In this case, all the people, including previous signers who had signed one month ago, must join to create a new signature. Can we realize such a trial in the real world?

Consequently, the existing schemes in Section 1.2, lead to the same problem. We would like to emphasize that the condition of $(k+\alpha)$-out-of-n property is not satisfied even in the case in which the signers create k-out-of-n and α-out-of-n signatures independently.

For this reason, in case of previous schemes, the signers are obliged to discard the previous signature and re-create a $(k+\alpha)$-out-of-n ring signature from scratch. The participants in this case include previous signers who created a k-out-of-n ring signature. This means that some applications by the previous works, lead to fatal failure under the circumstances in our scenario.

1.4 Our Contribution

In this paper, we introduce a new k-out-of-n ring signature. This study is based on our previous work [17], [25], [26]. The framework of our signature solves the problem described in Section 1.3.

Our Solution. To create an updated $(k+\alpha)$-out-of-n signature, we "reuse" the previous k-out-of-n ring signature. In the generation phase for the updated signature, the previous signer no longer needs to participate. The entity in the phase consists of the signers who will sign for α. An updated signature in this case, is created by the sum of both (i.e., k-out-of-n and α-out-of-n) signatures.

We call this property "flexibility". Our framework contains named the "dealer". She is in contact with the all signers and creates some parameters as a part of a signature. Such a dealer similarly exists in previous schemes described in Section 1.2.

Our signature can be updated at any time or any times. Hence, this is appropriate for using in the following situation: The number of signers increase gradually in progress of time. We beliebe that such a situation is very common in the real world. Our results are summarized as follows.

Flexible *k*-out-of-*n* Ring Signature. We focus on the update problem which exists previous signatures, and give its solution. To the best of our knowledge, this is the first approach to solve the problem. The flexibility enables possible signers to sign a message gradually. Our scheme does not have to determine k at the beginning of signing. As for the flexibility, we define and formalize it rigorously.

Proposal of Our Scheme. We also propose a practical scheme that satisfies the flexible property. Our scheme is anonymous even against a computationally

unbounded adversary, and unforgeable against chosen-subring attacks [6]. We then suggest a specific construction based on a multiple 1-out-of-n ring signature, which makes it possible to increase the threshold value gradually and specify the number of signers who actually join the signing protocol. We also prove the security of our proposal in a random oracle model.

Short-Term Key with Special Hash. To keep a k-out-of-n property using multiple ring structure, the following condition should be satisfied: A signer can sign a message only at once among k rings. In our scheme, we introduce "short-term public key" z_i, and embed it as a part of our signature. To make more concretely, let us explain the structure of z_i. A hash function in z_i, looks like one in the former paper, e.g. [12]. However we also use a parameter $g_i^{t_i}$ in z_i. Consequently, the order of z_i is fixed for each user i, and this leads to satisfy the following two desirable properties: (1) Each 1-out-of-n ring signature scheme must be signed by a different signer in a ring; and (2) Our scheme has "independent key parameter" property, that is, each user in a ring can generate her keys with independent parameters. We believe this technique would contribute to another cryptographic construction.

Security Framework. We define our security framework from flexible k-out-of-n ring signature point of view. Since our scheme is based on a multiple 1-out-of-n ring signature, we can prove in the same manner as [1, 6] that each 1-out-of-n ring signature is unforgeable, respectively. Moreover, we must prove the k-out-of-n property for multiple ring structure and our flexibility. In particular, as regarding flexibility, we needs signer not to sign on the same message with respect to the same ring even if they know her (long-term) secret key and all of the short-term secret keys corresponding to the signature. Hence, we define new security game to capture it. We prove our specified unforgeability with the game in random oracle model. We realize it by using short-term keys and the property of the random oracle.

1.5 Outline of our Paper

This paper is organized as follows. We first provide preliminary materials such as the notation and the necessary definitions in Section 2. Section 3 describes our idea and approach. Section 4 gives some definitions related to flexible k-out-of-n ring signature. We propose our scheme in Section 5, and the security proofs are given in Section 6. In Section 7, we compare the efficiency of existing k-out-of-n ring signature schemes, and discuss them. We finally conclude this paper in Section 8.

2 Preliminaries

2.1 Notation

We give some notations to be used through this paper. We denote the security parameter by λ, by PPT (Probabilistic Polynomial-Time), and by DPT (Deterministic Polynomial-Time). For a set $\mathbf{S}$, we write $x \xleftarrow{\$} \mathbf{S}$ to denote that x is

sampled uniformly and randomly from $\mathbf{S}$. We write $y \leftarrow A(x)$ to indicate that y is the output of an algorithm A when running on input x. We write $z \leftarrow x \circ y$ to indicate that z is the output of an operation $\circ$ when running on input x and y. We write $y \leftarrow x$ to indicate that y is assigned by x. We denote by $|\mathbf{S}|$ the size of the set. We denote by $\langle g \rangle$ the subgroup generated by g.

2.2 k-out-of-n ring signature

Our notation on a ring will basically follows [6]. Hence, hereafter we represent "signer" by "prover". We refer to an ordered list $\mathbf{R} = (PK_1, \ldots, PK_n)$ of public keys as a ring, and let $\mathbf{R}[i] = PK_i$. We will also freely use set notation, and say, e.g., that $PK \in \mathbf{R}$ if there exists an index i such that $\mathbf{R}[i] = PK$. In an analogous way, we will say that $\mathbf{PK} \subset \mathbf{R}$ if there exists a set $\mathbf{PK}$ of public keys such that every elements is in $\mathbf{R}$. We will always assume, without loss of generality, that the keys in a ring are ordered.

Provers can gather public keys in a system to choose a proper ring. Let $\mathbf{R}_\kappa$ and $\mathbf{R}_{\kappa'}$ be distinct rings. We can denote them as $\mathbf{R}_\kappa = \{PK_{(\kappa,1)}, \ldots, PK_{(\kappa,n_\kappa)}\}$ and $\mathbf{R}_{\kappa'} = \{PK_{(\kappa',1)}, \ldots, PK_{(\kappa',n_{\kappa'})}\}$. To avoid complicated suffixes, we describe a public key with a suffix relative to $\mathbf{R}$ in the current context. So, $PK_1 \in \mathbf{R}_\kappa$ and $PK_1 \in \mathbf{R}_{\kappa'}$ could differ. In analogous way, we simply describe a size of a ring as n but $|\mathbf{R}_\kappa|$ and $|\mathbf{R}_{\kappa'}|$ could differ.

Our k-out-of-n ring signature consists of multiple 1-out-of-n ring signature. Here, we define two functions: $f(\cdot)$ and $g(\cdot,\cdot)$. $f(\cdot)$ is inputted our k-out-of-n ring signature and returns the number of our 1-out-of-n ring signatures in it. $g(\cdot,\cdot)$ is inputted our k-out-of-n ring signatures and returns 1 if there exists a signature which is inputted as a first argument such that every elements is in a signature which is inputted as a second argument, otherwise returns 0. For example, let $\sigma = (\sigma_1, \sigma_2)$, $\sigma^* = (\sigma_1, \sigma_2, \sigma_3)$, $\sigma^{**} = (\sigma_2, \sigma_3)$ be our k-out-of-n ring signatures and $\sigma_1, \sigma_2, \sigma_3$ be our 1-out-of-n ring signatures for the same ring. In this case, $2 \leftarrow f(\sigma)$, $3 \leftarrow f(\sigma^*)$, $0 \leftarrow g(\sigma^*, \sigma)$, $1 \leftarrow g(\sigma, \sigma^*)$ and $0 \leftarrow g(\sigma, \sigma^{**})$.

In signing and verifying algorithms, we denote by "round" the order of generating/checking 1-out-of-n signatures. A subscript i means the i-th element in a ring and j means the j-th round in generating/checking signatures. For example, $\triangle_{(i,j)}$ means the i-th element of the j-th round. We show other denotations as follows:

- $\mathbf{U}$: A set of users in a ring. $\mathbf{U} = \{1, 2, \ldots, n\}$.
- U_i: A user corresponding to the i-th element in a ring. We sometimes denote this as user i. A tuple of (PK_i, SK_i) is her valid public and private key-pair.
- $\mathbf{P}$: A set of provers in a ring. $\mathbf{P} \subseteq \mathbf{U}$.
- P_i: A prover corresponding to the i-th element in a ring.
- $\mathbf{PK_S}$: A set of public keys for a set of $\mathbf{S}$. For example, we denote $\mathbf{PK_P}$ for a set of $\mathbf{P}$ of provers in a ring.
- $\mathbf{SK_S}$: A set of secret keys for a set of $\mathbf{S}$. For example, we denote $\mathbf{SK_P}$ for a set of $\mathbf{P}$ of provers in a ring.

To avoid complicated suffixes, we also describe $\mathbf{U}, \mathbf{P}$ with a suffix relative to $\mathbf{R}$ in the current context.

2.3 Complexity Assumptions

The security of our system is based on the discrete logarithm(DL) assumption. We assumes the discrete logarithm problem is hard to compute in prime order groups, which is the same assumption in [21]. Let p, q be large primes. $\langle g \rangle$ denote a prime subgroup of order q in $\mathbb{Z}_p^*$ generated by g. Pick $g \xleftarrow{\$} \langle g \rangle$ and $x \xleftarrow{\$} \mathbb{Z}_q^*$, then compute $y = g^x$. An adversary $\mathcal{A}$ has a advantage ϵ in solving the discrete logarithm problem if the condition

$$\Pr\left[x = x' | x' \leftarrow \mathcal{A}(\langle g \rangle, p, q, g, y)\right] \geq \epsilon(\lambda)$$

is satisfied.

3 Our Basic Idea

3.1 Short-Term Key with Hash Function

Obviously, it is impossible to construct a k-out-of-n ring signature only by using k items of 1-out-of-n ring signature. Therefore, we must make a structure such that each user can form a 1-out-of-n ring signature only once. For the purpose of it, we introduce short-term public key with hash function under the random oracle model in [5]. Let p_i, q_i be large primes. $\langle g_i \rangle$ denote a prime subgroup of order q_i in $\mathbb{Z}_{p_i}^*$ generated by g_i. Let $\bar{H}_i : \{0,1\}^* \rightarrow \langle g_i \rangle$ be cryptography hash function. We define short-term public key z_i as

$$z_i = \frac{g_i^{t_i}}{\bar{H}_i(M||w_i||\mathbf{R})} \bmod p_i,$$

where $M \in \{0,1\}^*$, $t_i \xleftarrow{\$} \mathbb{Z}_{q_i}$ and $w_i \xleftarrow{\$} \{0,1\}^\lambda$ such that $\exists i, i', i \neq i'$ then $w_i \neq w_{i'}$. A short-term secret key is t_i.

We embed short-term keys in each 1-out-of-n ring signature to satisfy k-out-of-n property. Let $\mathcal{S}$ be its signing algorithm, $\mathcal{V}$ be its verifying one and $\mathcal{D}$ be a dealer which generates and deals out short-term keys to $\mathbf{P}$. Let $\sigma_j = (\delta_{(j,0)}, \delta_{(j,1)})$ be a signature generated by our proposal at round j, where $\delta_{(j,0)}$ is a part which depends on a message M and $\delta_{(j,1)}$ is a part which does not depend on it. We assume a prover P_ℓ generates σ_j. Here, we do not consider flexibility of the threshold. P_ℓ receive short-term keys from $\mathcal{D}$ in the following steps.

$$(z_\ell, t_\ell, w_\ell) \leftarrow \mathcal{D}(M, \mathbf{R}, \ell)$$
$$\text{For } i \in \mathbf{U} \backslash P_\ell,\ z_i \xleftarrow{\$} \langle g_i \rangle$$

P_ℓ computes signature in the following steps.

$$\begin{aligned}
\delta_{(j,1)} &\leftarrow w_\ell \\
PK_i &\leftarrow (PK_i, z_i) \\
SK_\ell &\leftarrow (SK_\ell t_\ell) \\
\mathbf{R} &\leftarrow \{PK_i\}_{i\in\mathbf{U}} \\
\delta_{(j,0)} &\leftarrow \mathcal{S}_{SK_\ell}(M||\delta_{(j,1)}, \mathbf{R})
\end{aligned}$$

w_ℓ is for public but it is owner ambiguous value since $\bar{H}$ is a random oracle and t_ℓ is in secret. A verifier checks the signature σ_j in the following steps.

$$\begin{aligned}
(\delta_{(j,0)}, \delta_{(j,1)}) &\leftarrow \sigma_j \\
1/0 &\leftarrow \mathcal{V}_{\mathbf{R}}(\delta_{(j,0)}, M||\delta_{(j,1)})
\end{aligned}$$

We would like to modify the above protocol to satisfy k-out-of-n property with some hardness. We can realize it by expanding the protocol into multiple prover and appending one process at verifying. That is, we check whether $\exists j, j', j \neq j'$ and $\delta_{(j,1)} \stackrel{?}{=} \delta_{(j',1)}$. If the equation is correct, we reject the signature. In intuition, we consider that the protocol has a k-out-of-n property for the following reasons:

- If adversaries reuse t to form a ring more than once, they must find a collision w' such that $\bar{H}(M||w||\mathbf{R}) = \bar{H}(M||w'||\mathbf{R})$. Since $\bar{H}_i$ is a random oracle, it is difficult for any adversaries to break its one-wayness.
- If adversaries pick a proper $t'(\neq t)$, they must find a pre-image w' for $z^{-1} \cdot g^{t'}$ to form a ring more than once. Since $\bar{H}_i$ is a random oracle, it is difficult for any adversaries to break its collision resistant properties.

3.2 Flexibility

We describe the role of the dealer $\mathcal{D}$ especially in the case of flexibility. Dealer generates short-term keys for provers at signing in Section 3.1. We modify it as

$$\{(z_i, t_i, w_i)\}_{i\in\mathbf{U}} \leftarrow \mathcal{D}(M, \mathbf{R}).$$

However, the dealer keeps $\{(t_i, w_i)\}_{i\in\mathbf{U}\setminus\mathbf{P}}$ in secret and sends $\{t_i, w_i\}_{i\in\mathbf{P}}$ to provers in a ring, respectively. One of such z_i is indistinguishable from $z_i \stackrel{\$}{\leftarrow} \langle g_i \rangle$ without the knowledge of (t_i, w_i). If some users in the same ring (that is, new provers $\tilde{\mathbf{P}}$) want to sign on the same message, dealer can respectively send $\{(t_i, w_i)\}_{i\in\tilde{P}}$ to them, which have already embedded in z_i. Hence, new provers can sign the same message and append their 1-out-of-n ring signature to the existing k-out-of-n ring signature with respect to the same ring.

Remarks. The existence of the dealer is not special situation in k-out-of-n ring signature. Most of the existing schemes require it explicitly or implicitly.

They describe it as a "cooperation among provers", "leader of provers" and so on. Moreover, our dealer and most of it in other schemes do not need **Setup** algorithm because they generate only short-term keys at signing.

4 Flexible k-out-of-n Ring Signature

4.1 Functional Definition

We present the functional definition of our flexible k-out-of-n ring signature scheme. Our flexibility means that some new advocate for one message can sign on the message additionally without the help of others.

Definition 1 (Flexible k-out-of-n Ring Signature) *A flexible k-out-of-n ring signature scheme consists of three PPT and one DPT algorithms. We define $\mathbf{\Sigma} = (\mathbf{Gen}, \mathbf{Sign}, \mathbf{FSign}, \mathbf{Vrfy})$, where those algorithms mean generate keys for a user, sign a message, sign flexibly with respect to the number of provers and verify the signature of a message, respectively.*

$\mathbf{Gen}(1^\lambda)$. This algorithm outputs a public key PK_i and secret key SK_i for a user i. All users in this system can run this algorithm, respectively.

$\mathbf{Sign}_{\mathbf{P},\mathbf{SK_P}}(M, \mathbf{R})$. Let $\mathbf{P}$ be provers and $\mathbf{SK_P}$ be corresponding secret keys. This algorithm outputs a signature σ on the message M with respect to the ring $\mathbf{R}$. The signature is k-out-of-n ring signature, where $k = |\mathbf{P}|$. We assume the following:

1. The index i in the ring is known to provers.
2. $(\mathbf{R}[i], SK_i)$ is valid key-pair output by **Gen**.
3. Each public key in the ring is distinct.
4. The condition $1 \leq k \leq n$ is satisfied.

Note that in the case of $n = 1$, this is the special case as a ring signature scheme, which has no anonymity.

$\mathbf{FSign}_{\tilde{\mathbf{P}},\mathbf{SK}_{\tilde{\mathbf{P}}}}(\sigma, M, \mathbf{R})$. Let $\tilde{\mathbf{P}}$ be new provers and $\mathbf{SK}_{\tilde{\mathbf{P}}}$ be corresponding secret keys. Let $\mathbf{P}$ be provers who have already signed on the message and $\mathbf{SK_P}$ be corresponding secret keys. This algorithm checks whether the inputted signature σ is valid or not on the message M with respect to the ring $\mathbf{R}$ by **Vrfy** algorithm. If it is invalid, **FSign** outputs $\perp$. Otherwise, it updates and outputs a signature σ on the message M with respect to the ring $\mathbf{R}$. And it assigns values to variables as $\mathbf{P} \leftarrow \mathbf{P} \cup \tilde{\mathbf{P}}$ and $k \leftarrow |\mathbf{P} \cup \tilde{\mathbf{P}}|$. Note that this algorithm needs $\mathbf{P} \cap \tilde{\mathbf{P}} = \emptyset$ to generate a valid signature. If $\mathbf{P} \cap \tilde{\mathbf{P}} \neq \emptyset$, it should be difficult for **FSign** to generate a valid signature to satisfy the requirement for k-out-of-n property.

$\mathbf{Vrfy}_{\mathbf{R}}(M, \sigma)$. This algorithm outputs a single bit indicating validity(1) or invalidity(0) of a purported signature σ on a message M with respect to the ring $\mathbf{R}$.

■

In the signing algorithms, we will generally omit the input "$\mathbf{P}$" for simplicity. We require the following correctness condition. For any λ, any $\{(PK_i, SK_i)\}_{i=1}^{n}$ output by $\mathbf{Gen}(1^\lambda)$, any $\mathbf{P}, \tilde{\mathbf{P}} \subseteq \mathbf{U}$, any $\mathbf{SK_P} = \{SK_i | i \in \mathbf{P}, PK_i \in \mathbf{R}\}$, any $\mathbf{SK_{\tilde{P}}} = \{SK_i | i \in \tilde{\mathbf{P}}, PK_i \in \mathbf{R}\}$ and any M, we have

$$\begin{aligned} &(\mathbf{Vrfy_R}\left(M, \mathbf{Sign_{SK_P}}(M, \mathbf{R})\right) = 1) \\ \wedge &\left(\mathbf{Vrfy_R}\left(\mathbf{FSign_{SK_{\tilde{P}}}}(\mathbf{Sign_{SK_P}}(M, \mathbf{R}), M, \mathbf{R})\right) = 1\right), \end{aligned}$$

where $\mathbf{P} \cap \tilde{\mathbf{P}} = \emptyset$ and $\mathbf{R} = (PK_1, \ldots, PK_n)$.

The security of ring signature has two aspects: anonymity and unforgeability. Bender et al. [6] formalize them for 1-out-of-n ring signature. We modify their formulation to apply for our k-out-of-n ring signature.

4.2 Anonymity

The anonymity condition requires, informally, that an adversary should not be able to tell which members of ring generated a particular signature. We define a prover anonymity through a game between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$.

Definition 2 (Anonymity) *A flexible k-out-of-n ring signature scheme is anonymous against any computationally unbounded adversary $\mathcal{A}$, if for a polynomial $n(\cdot)$, the probability that $\mathcal{A}$ wins in the following game is exactly k/n:*

1. **Initialization.** $\mathcal{C}$ generates key pairs $\{(PK_i, SK_i)\}_{i=1}^{n(\lambda)}$ using $\mathbf{Gen}(1^\lambda)$ and sends the set of public keys $\mathcal{R} := \{PK_i\}_{i=1}^{n(\lambda)}$ to $\mathcal{A}$.
2. **Query to an oracle OSign.** $\mathcal{A}$ is given access to an oracle $\mathbf{OSign}(\cdot,\cdot,\cdot)$ such that $\mathbf{OSign}(\mathbf{P},M,\mathbf{R})$ returns $\mathbf{Sign_{SK_P}}(M,\mathbf{R})$, where we require $\mathbf{R} \subseteq \mathcal{R}$, $\mathbf{PK_P} \subseteq \mathbf{R}$ and $|\mathbf{R}| > 1$.
3. **Query to an oracle OFSign.** $\mathcal{A}$ is given access to an oracle $\mathbf{OFSign}(\cdot,\cdot,\cdot,\cdot)$ such that $\mathbf{OFSign}(\sigma,\tilde{\mathbf{P}},M,\mathbf{R})$ returns $\mathbf{FSign_{\tilde{P}}}(\sigma,M,\mathbf{R})$, where σ is generated by **OSign** or **OFSign**. We require $\mathbf{R} \subseteq \mathcal{R}$, $\mathbf{PK_{\tilde{P}}} \subseteq \mathbf{R}$, $\mathbf{P} \cap \tilde{\mathbf{P}} = \emptyset$ and $|\mathbf{R}| > 1$.
4. **Challenge.** $\mathcal{A}$ outputs a message M^*, distinct same size sets $\mathbf{P}_0^*, \mathbf{P}_1^*$, and a ring for which $\mathbf{R}_0^*, \mathbf{R}_1^* \subseteq \mathcal{R}$, where $|\mathbf{R}_0^*|, |\mathbf{R}_1^*| > 1$. $\mathcal{C}$ randomly chooses a bit $b \in \{0, 1\}$ and sends the signature $\sigma^* \leftarrow \mathbf{Sign_{SK_{P_b^*}}}(M^*, \mathbf{R}_b^*)$ to $\mathcal{A}$.
5. **Answer.** $\mathcal{A}$ outputs a bit b' and wins if $b' = b$. $\mathcal{C}$ outputs 1 if $\mathcal{A}$ wins this game, otherwise outputs 0.

■

4.3 Unforgeability

The unforgeability means any adversary can not forge any new valid signature. We begin by defining a security game with respect to each 1-out-of-n ring signature. We utilize it to prove that it is difficult for any computationally bounded

adversary to forge a part of our 1-out-of-n ring signature without the knowledge of secret keys corresponding to our complexity assumption. Moreover, we define another security game to prove the k-out-of-n property. That is , the adversaries can not sign on the same message and ring more than once even if they know secret keys in our scheme. With the two games, we will prove our proposal security in Section 6.2. We now define them formally.

Definition 3 (Unforgeability) *A flexible k-out-of-n ring signature scheme is unforgeable against chosen-subring attacks [6] if for any PPT adversary $\mathcal{A}$ and for any polynomial $n(\cdot)$, the probability that $\mathcal{A}$ wins in the following two games is negligible:*

Game0.

1. **Initialization.** $\mathcal{C}$ generates key pairs $\{(PK_i, SK_i)\}_{i=1}^{n(\lambda)}$ using $\mathbf{Gen}(1^\lambda)$ and sends the set of public keys $\mathcal{R} := \{PK_i\}_{i=1}^{n(\lambda)}$ to $\mathcal{A}$.
2. **Query to an oracle OSign.** $\mathcal{A}$ is given access to an oracle $\mathbf{OSign}(\cdot,\cdot,\cdot)$ such that $\mathbf{OSign}(\mathbf{P},M,\mathbf{R})$ returns $\mathbf{Sign}_{\mathbf{SK_P}}(M,\mathbf{R})$, where we require $\mathbf{R} \subseteq \mathcal{R}$ and $\mathbf{PK_P} \subseteq \mathbf{R}$.
3. **Query to an oracle OFSign.** $\mathcal{A}$ is given access to an oracle $\mathbf{OFSign}(\cdot,\cdot,\cdot,\cdot)$ such that $\mathbf{OFSign}(\sigma,\tilde{\mathbf{P}},M,\mathbf{R})$ returns $\mathbf{FSign}_{\tilde{\mathbf{P}}}(\sigma,M,\mathbf{R})$, where σ is generated by **OSign** or **OFSign**. We require $\mathbf{R} \subseteq \mathcal{R}$, $\mathbf{PK}_{\tilde{\mathbf{P}}} \subseteq \mathbf{R}$ and $\mathbf{P} \cap \tilde{\mathbf{P}} = \emptyset$.
4. **Answer.** $\mathcal{A}$ outputs $(\sigma^*,M^*,\mathbf{R}^*)$ and wins if $\mathbf{R}^* \subseteq \mathcal{R}$, $\mathbf{Vrfy}_{\mathbf{R}^*}(M^*,\sigma^*) = 1$, and one of the following conditions are satisfied: (i) $\mathcal{A}$ never queried $(M^*, \mathbf{R}^*)$ to the oracles (ii) $f(\sigma^*) > f(\sigma) \wedge g(\sigma, \sigma^*) = 1$. (Recall the Section 2.1.) $\mathcal{C}$ outputs 1 if $\mathcal{A}$ wins this game, otherwise outputs 0.

Game1.

1. **Initialization.** This is the same as 1 in the Game0.
2. **Query to an oracle OSign.** This is the same as 2 in the Game0.
3. **Query to an oracle OFSign.** This is the same as 3 in the Game0.
4. **Challenge.** $\mathcal{A}$ outputs $(P_i^*, M^*, \mathbf{R}^*)$, and $\mathcal{A}$ is never queried $(M^*, \mathbf{R}^*)$. $\mathcal{C}$ returns a signature σ^{**} corresponding to it. Finally, $\mathcal{C}$ corrupts $SK_{P_i^*}$ and random numbers which are used at signing to $\mathcal{A}$.
5. **Query to an oracle OSign.** This is the same as 2 in this game except that $\mathcal{A}$ never queried $(\cdot, M^*, \mathbf{R}^*)$.
6. **Query to an oracle OFSign.** This is the same as 3 in this game except that $\mathcal{A}$ never queried $(\sigma^{**}, \cdot, M^*, \mathbf{R}^*)$.
7. **Answer.** $\mathcal{A}$ outputs $(\sigma^*,M^*,\mathbf{R}^*)$ and wins if $\mathbf{R}^* \subseteq \mathcal{R}$, $\mathbf{Vrfy}_{\mathbf{R}^*}(M^*,\sigma^*) = 1$. $\mathcal{C}$ outputs 1 if $\mathcal{A}$ wins this game, otherwise outputs 0.

■

5 Our Scheme

5.1 k-out-of-n Ring Signature

We begin by constructing our k-out-of-n ring signature scheme except a flexible sign algorithm for readability.

Gen(1^λ)**.** Let p_i,q_i be large primes. Let $\langle g_i \rangle$ denote a prime subgroup of $\mathbb{Z}_{p_i}^*$ generated by g_i whose order is q_i. Let $H_i : \{0,1\}^* \to \mathbb{Z}_{q_i}$ and $\bar{H}_i : \{0,1\}^* \to \langle g_i \rangle$ be cryptography hash functions. Each user i in this system computes

- $g_i \xleftarrow{\$} \langle g_i \rangle$, $x_i \xleftarrow{\$} \mathbb{Z}_{q_i}$, $H_i \xleftarrow{\$} \{H_i\}$, $\bar{H}_i \xleftarrow{\$} \{\bar{H}_i\}$,
- $y_i \leftarrow g_i^{x_i} \bmod p_i$.

Ring / Public Key / Secret Key. Provers $\mathbf{P}$ gather public keys in this system to decide a ring. We set

$$\begin{aligned} PK_i &\leftarrow (\langle g_i \rangle, g_i, p_i, q_i, y_i, H_i, \bar{H}_i), \\ SK_i &\leftarrow x_i, \\ \mathbf{R} &\leftarrow (PK_1, PK_2, \cdots, PK_n), \\ \mathbf{SK_P} &\leftarrow \{x_i\}_{i \in \mathbf{P}}. \end{aligned}$$

Sign$_{\mathbf{SK_P}}(M, \mathbf{R})$**.** Provers $\mathbf{P}$ send a message M and a ring $\mathbf{R}$ to a dealer. The dealer does as follows.

- For $i \in \mathbf{U}$, $t_i \xleftarrow{\$} \mathbb{Z}_{q_i}^*$.
- For $i \in \mathbf{U}$, $w_i \xleftarrow{\$} \{0,1\}^\lambda$ such that each w_i is different from the others.
- For $i \in \mathbf{U}$, $z_i \leftarrow g_i^{t_i} \cdot (\bar{H}_i(M||w_i||\mathbf{R}))^{-1} \bmod p_i$.
- Output an ordered list $\mathbf{z}_{(M,\mathbf{R})} \leftarrow (z_1, z_2, \ldots, z_n)$ as a part of signature.
- Encrypt a tuple of (t_i, w_i) and send it to each prover $i \in \mathbf{P}$, respectively.
- Save the history of $(M, \mathbf{R}, \{t_i\}_{i \in \mathbf{U}}, \{w_i\}_{i \in \mathbf{U}})$.

Note that non-real prover $i \notin \mathbf{P}$ cannot receive (t_i, w_i) from the dealer.

Let $\ell \in \mathbf{P}$ be a prover who generates j-th 1-out-of-n ring signature. She decrypts the cipher-text from the dealer and parse it as (t_ℓ, w_ℓ).

(Initialization) Prover ℓ computes

- $w_j \leftarrow w_\ell$,
- $r_\ell \xleftarrow{\$} \mathbb{Z}_{q_\ell}$,
- $a_{(\ell,j)} \leftarrow {g_\ell}^{r_\ell} \bmod p_\ell$,
- $c_{(\ell+1,j)} \leftarrow H_{\ell+1}(M||w_j||\mathbf{R}||\mathbf{z}_{(M,\mathbf{R})}||a_{(\ell,j)})$.

(Forward sequence) For $i = \ell+1, \ldots, n, 1, \ldots, \ell-1$, prover ℓ computes the parameters in the j-th round such that

- $s_{(i,j)} \xleftarrow{\$} \mathbb{Z}_{q_i}$,
- $a_{(i,j)} \leftarrow z_i \cdot \bar{H}_i(M||w_j||\mathbf{R})\ {g_i}^{s_{(i,j)}} {y_i}^{c_{(i,j)}} \bmod p_i$,
- $c_{(i+1,j)} \leftarrow H_{i+1}(M||w_j||\mathbf{R}||\mathbf{z}_{(M,\mathbf{R})}||a_{(i,j)})$.

(Forming the ring) Prover ℓ computes

$s_{(\ell,j)} \leftarrow r_\ell - t_\ell - x_\ell c_{(\ell,j)} \bmod q_\ell$.

(Output) Prover ℓ outputs σ_j as his 1-out-of-n ring signature such that

$\sigma_j \leftarrow (c_{(1,j)}, s_{(1,j)}, \ldots, s_{(n,j)}, w_j)$.

Note that each prover can do the above steps independently.

***k*-out-of-*n* Signature.** Provers $\mathbf{P}$ outputs as a k-out-of-n ring signature such that

$\sigma \leftarrow (\{\sigma_j\}_{j \in [1,k]}, \mathbf{z}_{(M,\mathbf{R})})$,

where $k = |\mathbf{P}|$.

Vrfy$_{\mathbf{R}}(M, \sigma)$**.** A verifier checks a signature as follows.

- $(\{\sigma_j\}_{j\in[1,k]}, \mathbf{z}_{(M,\mathbf{R})}) \leftarrow \sigma$.
- $(c_{(1,j)}, s_{(1,j)}, \ldots, s_{(n,j)}, w_j) \leftarrow \sigma_j$.
- $(z_i, z_2, \ldots, z_n) \leftarrow \mathbf{z}_{(M,\mathbf{R})}$.
- If $\exists j, j' \in [1,k]$ such that $w_j = w_{j'}$ where $j \neq j'$, stop this algorithm and reject the signature.
- If $\exists i \in \mathbf{U}$ such that $c_{(i,\cdot)} \notin \langle g_i \rangle$ or $s_{(i,\cdot)} \notin \langle g_i \rangle$, stop this algorithm and reject the signature.
- If $\exists i \in \mathbf{U}$ such that $z_i \notin \langle g_i \rangle$ stop this algorithm and reject the signature.
- For $i = 1, \ldots, n$ and for $j = 1, \ldots, k$, computes
  - $a_{(i,j)} \leftarrow z_i \cdot \bar{H}_i(M||w_j||\mathbf{R})\ {g_i}^{s_{(i,j)}} {y_i}^{c_{(i,j)}} \bmod p_i$,
  - $c_{(i+1,j)} \leftarrow H_{i+1}(M||w_j||\mathbf{R}||\mathbf{z}_{(M,\mathbf{R})}.\ ||a_{(i,j)})$ if $i \neq n$.
- Accept σ as a k-out-of-n ring signature on a message M if and only if $\forall j \in [1,k]$, $c_{(1,j)} = H_1(M||w_j||\mathbf{R}||\mathbf{z}_{(M,\mathbf{R})}||a_{(n,j)})$. Otherwise, return 0.

5.2 Flexible Sign

We assume that $\mathbf{P} \cap \tilde{\mathbf{P}} = \emptyset$. If **FSign** output a signature with $\mathbf{P} \cap \tilde{\mathbf{P}} \neq \emptyset$, **Vrfy** always reject it under the complexity assumption.

$\mathbf{FSign}_{\mathbf{SK}_{\tilde{\mathbf{P}}}}(\sigma, M, \mathbf{R})$. Additional provers $\tilde{\mathbf{P}}$ send $(M, \mathbf{R}, \sigma)$ to the dealer. For all provers $i \in \tilde{\mathbf{P}}$, the dealer does as follows.

- $b \leftarrow \mathbf{Vrfy}_{\mathbf{R}}(M, \sigma)$.
- If $b = 0$, stop this algorithm and return $\perp$.
- Search for a tuple of (t_i, w_i) from the history of $(M, \mathbf{R}, \{t_i\}_{i\in\mathbf{U}}, \{w_i\}_{i\in\mathbf{U}})$.
- Encrypt the tuple of (t_i, w_i) and send it to prover $i \in \mathbf{P}$.

Let $\ell \in \tilde{\mathbf{P}}$ be an additional prover who generates $\tilde{j}$-th($\tilde{j} > k$) 1-out-of-n ring signature. Prover ℓ does the same steps in **Sign** to get $\sigma_{\tilde{j}}$ and updates

- $(\{\sigma\}_{j\in[1,k]}, \mathbf{z}_{(M,\mathbf{R})}) \leftarrow \sigma$,
- $\sigma \leftarrow (\{\sigma\}_{j\in[1,k]}, \sigma_{\tilde{j}}, \mathbf{z}_{(M,\mathbf{R})})$,
- $\mathbf{P} \leftarrow \mathbf{P} \cup \tilde{\mathbf{P}}$,
- $k \leftarrow |\mathbf{P}|$.

All additional provers $\tilde{\mathbf{P}}$ do the above steps, respectively.

***k*-out-of-*n* Signature.** The structure is exactly the same in Section 5.1. Hence, we can use the same **Vrfy**. Note that $\mathbf{P}$ and its size are different since **FSign** updates the signature.

Let ℓ be a prover at round j. For the consistency, it is sufficient to show that the equation $z_\ell \cdot \bar{H}(M||w_j||\mathbf{R}) \cdot g_\ell^{s_{(\ell,j)}} y_\ell^{c_{(\ell,j)}} \bmod p_\ell = {g_\ell}^{r_\ell} \bmod p_\ell$ holds for any j from 1 to n. The correctness can be verified as

$$
\begin{aligned}
z_\ell \ \cdot\ & \bar{H}_\ell(M||w_j||\mathbf{R}) \cdot g_\ell^{s_{(\ell,j)}} y_\ell^{c_{(\ell,j)}} \\
&= g_\ell^{t_\ell} \cdot \bar{H}_\ell(M||w_\ell||\mathbf{R})^{-1} \cdot \bar{H}_\ell(M||w_j||\mathbf{R}) \cdot g_\ell^{s_{(\ell,j)}} y_\ell^{c_{(\ell,j)}} \\
&= g_\ell^{t_\ell} g_\ell^{s_{(\ell,j)}} y_\ell^{c_{(\ell,j)}} \\
&= g_\ell^{t_\ell} g_\ell^{r_\ell - t_\ell - x_\ell c_{(\ell,j)}} y_\ell^{c(\ell,j)} \\
&= g_\ell^{r_\ell} \pmod{p_\ell}.
\end{aligned}
$$

6 Security Analysis

6.1 Anonymity

Theorem 4 (Anonymity of our scheme) *Under the secret of $\{t_i\}_{i\in\mathbf{U}}$, for any computationally unbounded adversary $\mathcal{A}$, our flexible k-out-of-n ring signature scheme is anonymous in the sense of Definition 2.*

Proof of Theorem 4. We first focus on the distribution of our signature scheme $\mathbf{\Sigma}$. Let $\{w\} := \{w | w \in \{0,1\}^\lambda \wedge w' \in \{0,1\}^\lambda \backslash w \wedge w \neq w'\}$ be a set and n be a size of the set. Let $\mathcal{W}$ be a set whose element is $\{w\}$. $\mathbf{\Sigma}$ chooses $\{w_i\}_{i\in\mathbf{U}}$ at **Sign** in the following steps.

$\{w\} \xleftarrow{\$} \mathcal{W}$
For $i \in \mathbf{U}$
  $w_i \xleftarrow{\$} \{w\}$
  $\{w\} \leftarrow \{w\} \backslash w_i$

Therefore, $\{w_i\}_{i\in\mathbf{U}}$ have $\binom{2^\lambda - 1}{n} \cdot n!$ variations with same probability. Since $2^\lambda - 1 >> n$, w_i have no meaning to adversaries to break an anonymity of provers.

Since $t_i \xleftarrow{\$} \mathbb{Z}_{q_i}^*$, each z_i is independent with the others owing to t_i. Moreover, all z_i distribute uniformly over $\mathbb{Z}_{q_i}$ on the same reason. Therefore, $\{z_i\}_{i\in\mathbf{U}}$ have $\mathit{\Pi}_{i\in[1,n]} q_i$ variations with same probability.

We now observe that all $s_{(i,j)}$ are taken randomly from $\mathbb{Z}_{q_i}$ except for $s_{(\ell,j)}$ at the closing point. $s_{(\ell,j)}$ also distributes uniformly over $\mathbb{Z}_{q_i}$ since t_i and r_i are uniformly chosen from $\mathbb{Z}_{q_i}$. Hence, $(s_{(1,j)}, \ldots, s_{(n,j)})$ have $\mathit{\Pi}_{i\in[1,n]} q_i$ variations that are equally likely regardless of the closing point. Remaining $c_{(1,j)}$ in signature is determined uniquely from others. They are independent to each other with respect to i and j. Therefore, $\{(c_{(1,j)}, s_{(1,j)}, \ldots, s_{(n,j)})\}_{j\in[1,k]}$ have $k \cdot \mathit{\Pi}_{i\in[1,n]} q_i$ variations with the same probability.

We recall the game of Definition 2 in this step. $\mathcal{A}$ queries to **OSign** but is not able to retrieve user-specific information from $(c_{(1,j)}, s_{(1,j)}, \ldots, s_{(n,j)}, \{z_i\}_{i\in\mathbf{U}})$. $\mathcal{A}$ is only able to learn information for $P \in \mathbf{P}$ from $\{w_j\}_{j\in[1,k]}$ (and input of specified provers $\mathbf{P}$ for the oracle) but is not able to learn information for $U \in \mathbf{U}\backslash\mathbf{P}$. In analogous way, $\mathcal{A}$ is not able to learn about $U \in \mathbf{U}\backslash(\mathbf{P} \cup \tilde{\mathbf{P}})$ by **OFSign**. Moreover, $\mathbf{\Sigma}$ generates new subgroup $\{w_i\}$ in $\mathbf{U}$ with each function (**O**)**Sign** call, thus, the information of the old $\{w_j\}_{j\in[1,k]}$ has no use. We obtain the proof of Theorem 4. □

6.2 Unforgeability

Theorem 5 (Unforgeability of our scheme) *For any PPT adversary, our flexible k-out-of-n ring signature scheme is unforgeable against chosen-subring attacks [6].*

Proof of Theorem 5. We describe a public key with a suffix relative to $\mathbf{R}$ in the current context. (Recall Section 2.1.) In analogous way, we omit the suffix of $\mathbf{z}_{(M,\mathbf{R})}$. For simplicity, the random oracles $\{H_i\}_{i\in\mathbf{U}}$ and $\{\bar{H}_i\}_{i\in\mathbf{U}}$ are treated as a single oracle H and $\bar{H}$, respectively.

To prove Theorem 5, we define and prove two lemmas. Let $\mathcal{A}_0$ be a $(\tau_0, \epsilon_0, q_s, q_h)$-adversary that requests signing oracles at most q_s times, accesses random oracle H at most q_h times in the Game0 of Section 4.3 and outputs forged $(\sigma, M, \mathbf{R})$ with probability at least ϵ_0 and running time at most τ_0.

Lemma 6 *If there exists $(\tau_0, \epsilon_0, q_s, q_h)$-adversary $\mathcal{A}_0$ for public key set $\mathbf{R}$ of size n, then there exists (η_0, μ_0)-adversary $\mathcal{B}_0$ that computes discrete-logarithm x_i of $(\langle g_i\rangle, p_i, q_i, g_i, y_i(= g_i^{x_i})) \in \mathbf{R}$ for at least one i with probability at least μ_0 within running time η_0 by using $\mathcal{A}_0$. Here, $\eta_0 < (32q_h^2 + 4)\epsilon_0^{-1}\tau_0$ and $\mu_0 > 9/100$ under the condition that $\epsilon_0 > 8q_h^2 q^{-1}$ and $q > 2q_h q_s$ where q is the smallest q_i included in $\mathbf{R}$.*

Proof of Lemma 6. We begin by showing how to simulate the oracles in Game0.

Key generation. $\mathcal{B}_0$ receives tuples of discrete logarithm problems as public key set. Note that they are independent to each other. The remainder of public key is generated exactly as prescribed by the **Gen** algorithm.

Query to $\bar{H}$. $\bar{H}$ takes $\bar{Q}_\kappa = (i, M_\kappa, w_\kappa, \mathbf{R}_\kappa)$ as κ-th query. $\mathcal{B}_0$ computes $\bar{h}_\kappa \xleftarrow{\$} \langle g_i\rangle$ and returns $\bar{h}_\kappa$ that corresponds to $H_i(M_\kappa, w_\kappa, \mathbf{R}_\kappa)$ maintaining consistency against duplicated queries.

Query to H. H takes $Q_\kappa = (i, M_\kappa, w_\kappa, \mathbf{R}_\kappa, \mathbf{z}_\kappa, a_\kappa)$ as κ-th query. $\mathcal{B}_0$ deal with h_κ as follows: If $\mathcal{B}_0$ queries to H to form a ring at signing, chooses h_κ such that it maintains consistency with respect to forming the ring. Otherwise, $\mathcal{B}_0$ chooses $h_\kappa \xleftarrow{\$} \mathbb{Z}_{q_i}$. $\mathcal{B}_0$ returns h_κ that corresponds to $H_i(M_\kappa, w_\kappa, \mathbf{R}_\kappa, \mathbf{z}_\kappa, a_\kappa)$ maintaining consistency against duplicated queries.

OSign. $\mathcal{B}_0$ simulates in the following steps.

For $i \in \mathbf{U}$
  $t_i \xleftarrow{\$} \mathbb{Z}_{q_i}^*$
  $w_i \xleftarrow{\$} \{0,1\}^\lambda$ s.t. each w_i is different from others.
  $z_i \leftarrow g_i^{t_i} \cdot (\bar{H}_i(M||w_i||\mathbf{R}))^{-1} \bmod p_i$
$(\mathbf{z}, \mathbf{w}) \leftarrow ((z_1, \ldots, z_{|\mathbf{U}|}), (w_1, \ldots, w_{|\mathbf{U}|}))$
For $j = 1, \ldots, |\mathbf{P}|$
  $c_{(1,j)} \xleftarrow{\$} \mathbb{Z}_{q_1}$
  $w_j \xleftarrow{\$} \mathbf{w}$
  $\mathbf{w} \leftarrow \mathbf{w} \backslash w_j$
  For $i = 1, \ldots, |\mathbf{U}|$
    $s_{(i,j)} \xleftarrow{\$} \mathbb{Z}_{q_i}$
    $a_{(i,j)} \leftarrow z_i \cdot \bar{H}_i(M||w_j||\mathbf{R}_\kappa)\, g_i^{s_{(i,j)}} y_i^{c_{(i,j)}} \bmod p_i$ $c_{(i+1,j)} = H_{i+1}(M||w_j||\mathbf{R}_\kappa||\mathbf{z}||a_{(i,j)})$
    if $i \neq |\mathbf{U}|$
  $c_{(1,j)} \leftarrow H_1(M||w_j||\mathbf{R}_\kappa\ ||\mathbf{z}||a_{(|\mathbf{U}|,j)})$
The assignment to $c_{(1,j)}$ means to form a ring.

OFSign. $\mathcal{B}_0$ checks the inputted signature. If the equation $\mathbf{Vrfy}_{\mathbf{R}_\kappa}(M, \sigma) = 0$ holds, $\mathcal{B}_0$ outputs $\perp$. Otherwise, $\mathcal{B}_0$ generates a signature in the same manner of **OSign** except that $\mathcal{B}_0$ reuses $\mathbf{z}, \mathbf{w}$.

We prove the the rest of this proof in analogy with [1]. Let Θ, Ω be the random tapes given to the signing oracle and $\mathcal{A}_0$. The success probability of $\mathcal{A}_0$ is taken over the space defined by Θ, Ω and random oracle $H, \bar{H}$. We fix the value of random tapes and the behavior of $\bar{H}$. In the case of forming a ring by queries, there exists at least one index, say α, in $[1, n]$ such that $Q_u = (\alpha + 1, M, w, \mathbf{R}, \mathbf{z}, a_{\alpha+1})$ and $Q_v = (\alpha, M, w, \mathbf{R}, \mathbf{z}, a_\alpha)$ satisfy $u \leq v$. Split H as (H^-, h_α) where H^- corresponds to the answers to all queries except for Q_v answered with h_α.

By rewinding $\mathcal{A}_0$, $\mathcal{B}_0$ finds at least one randomly chosen $h'_\alpha (\neq h_\alpha)$ such that a relation of the query order (u, v) is unchanged over the space of (Θ, Ω). Since Q_u happens before Q_v, a_α is unchanged both run. $\mathcal{B}_0$ can compute the discrete-log since $x_\alpha = (s_\alpha - s'_\alpha)/(h'_\alpha - h_\alpha) \bmod q_\alpha$.

The analysis of the reduction cost is also most of the same with [1]. The signing simulation fails if the process of forming a ring causes inconsistency in $\bar{H}_1$. It happens with probability at most q_h/q where q is the smallest q_i in a ring. Hence, the signing simulation is successful q_s times with probability at least

$$\left(1 - \frac{q_h}{q}\right)^{q_s} \geq 1 - \frac{q_h q_s}{q}.$$

Let $\mathcal{S}$ be a set of $(\Theta, \Omega, H, \bar{H})$ with which $\mathcal{A}_0$ is successful in forgery. From the definition of ϵ_0, we have $\Pr[(\Theta, \Omega, H, \bar{H}) \in \mathcal{S}] \geq \epsilon_0$. Due to the ideal randomness of H, there exist queries $Q_\kappa = (i, M, w, \mathbf{R}, \mathbf{z}, a_i)$ for $i \in \mathbf{U}$ with probability at least $1 - 1/q$. Let $\mathcal{S}'$ be a subset of $\mathcal{S}$ where $(\Theta, \Omega, H, \bar{H}) \in \mathcal{S}'$ leads $\mathcal{A}$ to output a signature that has corresponding queries as above with successfully simulated signing oracles. Then, we have

$$\Pr[(\Theta, \Omega, H, \bar{H}) \in \mathcal{S}'] \geq \epsilon'_0,$$

where $\epsilon'_0 = (1 - q_h q_s/q)(1 - 1/q)\epsilon_0$.

We classify $\mathcal{S}'$ by the (u, v). Let $\mathcal{S}'_{(u,v)}$ denote a class where $(\Theta, \Omega, H, \bar{H}) \in \mathcal{S}'_{(u,v)}$ yields (u, v). There are at most $\binom{2}{q_h} + \binom{1}{q_h} = q_h(q_h + 1)/2$ classes. By invoking $\mathcal{A}_0$ with randomly chosen $(\Theta, \Omega, H, \bar{H})$ at most $t_1 = {\epsilon'_0}^{-1}$ times, $\mathcal{B}_0$ finds at least one $(\Theta, \Omega, H, \bar{H}) \in \mathcal{S}'_{(u,v)}$ for some (u, v) with probability

$$1 - (1 - {\epsilon'}_0)^{\frac{1}{\epsilon'_0}} > 1 - exp(-1) > \frac{3}{5}.$$

Let $GI = \{(u, v) || \mathcal{S}'_{(u,v)}|/|\mathcal{S}'| \geq 1/(q_h(q_h + 1))\}$ and $\mathcal{S}'' = \{(\Theta, \Omega, H, \bar{H}) \in \mathcal{S}'_{(u,v)} | (u, v) \in GI\}$. Then, it holds that $\Pr[\mathcal{S}''|\mathcal{S}'] \geq 1/2$. Due to the heavy-row lemma [16], $(\Theta, \Omega, H, \bar{H})$ that yields the successful run of $\mathcal{A}_0$ is in $\mathcal{S}''$ with

probability at least 1/2. Due to the heavy-row lemma, again, with probability at least 1/2, $(\Theta, \Omega, H^-, \bar{H})$ satisfies

$$\Pr[(\Theta, \Omega, H^-, h'_\alpha, \bar{H}) \in \mathcal{S}'_{(u,v)}] \geq \frac{\epsilon'_0}{2q_h(q_h+1)}.$$

Since we assume $\epsilon_0 > 8q_h^2/q$ and $q > 2q_hq_s$, it holds that $\epsilon'_0/(2q_h(q_h+1)) > 1/q$.

By the running $\mathcal{A}_0$ up to $t_2 = ((\epsilon'_0/(2q_h(q_h+1))) - 1/q)^{-1}$ times with $(\Theta, \Omega, H^-, \bar{H})$ obtained in the first 3/5, Overall success probability is

$$\mu_0 > \frac{3}{5} \cdot \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{3}{5} = \frac{9}{100},$$

and the number of invocations of $\mathcal{A}_0$ is

$$\begin{aligned} t_1 + t_2 &< \frac{1}{\epsilon'_0} + \frac{4q_h(q_h+1)}{\epsilon'_0} \\ &< \frac{4}{\epsilon_0} + \frac{4 \cdot 4 \cdot 2q_h^2}{\epsilon_0} \\ &= \frac{32q_h^2 + 4}{\epsilon_0}. \end{aligned}$$

We have proven Lemma 6. □

We now try to prove another lemma. Let $\mathcal{A}_1$ be a $(\tau_1, \epsilon_1, q_s, q_{\bar{h}})$-adversary that requests signing oracles at most q_s times, accesses random oracle $\bar{H}$ at most $q_{\bar{h}}$ times in the Game1 of Section 4.3 and outputs forged $(\sigma, M, \mathbf{R})$ with probability at least ϵ_1 and running time at most τ_1.

Lemma 7 *We assume there exists $(\tau_1, \epsilon_1, q_s, q_{\bar{h}})$-adversary $\mathcal{A}_1$ for public key set $\mathbf{R}$ of size n. Then, by using $\mathcal{A}_1$, there exists (η_1, μ_1)-adversary $\mathcal{B}_1$ that computes discrete-logarithm β_i of $(\langle g_i \rangle, p_i, q_i, g_i, Y(= g_i^{\beta_i})) \in \mathbf{R}$ for at least one i with probability at least μ_1 within running time η_1 or breaks a collision resistance property of the random oracle $\bar{H}$. Here, $\eta_0 < (32q_h^2 + 4)\epsilon_1^{-1}\tau_0$ and $\mu_1 > 9/100$ under the condition that $\epsilon_1 > 8q_h^2q^{-1}$ and $q > 2q_hq_s$ where q is the smallest q_i included in $\mathbf{R}$.*

Proof of Lemma 7. We begin by showing how to simulate the oracles in Game1.

Key generation. $\mathcal{B}_1$ generates public/secret keys by **Gen** algorithm.

Query to $\bar{H}$. $\mathcal{B}_1$ receives tuples of discrete logarithm problems. $\bar{H}$ takes $\bar{Q}_\kappa = (i, M_\kappa, w_\kappa, \mathbf{R}_\kappa)$ as κ-th query. $\mathcal{B}_1$ picks $\gamma \xleftarrow{\$} \mathbb{Z}_{q_i}$, computes $\bar{h}_\kappa \leftarrow Y^\gamma$. and returns $\bar{h}_\kappa$ that corresponds to $H_i(M_\kappa, w_\kappa, \mathbf{R}_\kappa)$ maintaining consistency against duplicated queries.

Query to H. This is the same as one in the Game0.

OSign. $\mathcal{B}_1$ simply simulates with the knowledge of secret keys.

OFSign. This is the same as one in the Game0 except that $\mathcal{B}_1$ simply simulates with the knowledge of secret keys.

Table 1. Comparison of k-out-of-n ring signature schemes.

Scheme	Signature Type	Flexibility	Independent Key Parameter †1	Signature Size
Our Scheme	Multiple Ring Structure	**Yes**	**Yes**	$\mathcal{O}(kn)$
BSS02 [7]	Partition	No	**Yes**	$\mathcal{O}(2^k n \log n)$
AOS02 [1] + BSS02 [7]	Partition	No	**Yes**	$\mathcal{O}(2^k n \log n)$
AOS04 [2]	Secret-Sharing	No	**Yes**	$\boldsymbol{\mathcal{O}(n)}$
FS05 [12]	Secret-Sharing	No	No	$\boldsymbol{\mathcal{O}(n)}$
YLASZ11 [27]	Secret-Sharing	No	No	$\boldsymbol{\mathcal{O}(n)}$
PBB12 [18]	Multivariate Vector	No	No	$\mathcal{O}(\gamma^{\dagger 2} \cdot n)$

†1: It means whether users in a ring can generate their keys with independent parameters.
†2: γ means the number of round to convert from identification to signature.
A cheating probability is less than 2^{-80} if $\gamma = 193$.

$\mathcal{A}_1$ outputs $(P_i, M, \mathbf{R})$ in the challenge phase. $\mathcal{B}_1$ computes σ' by the secret key SK_{P_i}. $\mathcal{B}_1$ corrupts the secret key and random numbers (include short-term secret keys) which are used at signing to $\mathcal{A}_1$. After that, $\mathcal{B}_1$ responses for the queries in analogous way. Finally, $\mathcal{A}_1$ outputs forged $(\sigma, M, \mathbf{R})$ with probability at least ϵ_1 and running time τ_1. By Lemma 6, $\mathcal{A}_1$ cannot forge even 1-out-of-n ring signature except for P_i because $\mathcal{A}_1$ has no knowledge of secret keys corresponding to $U \in \mathbf{U} \backslash P_i$.

Let Θ, Ω be the random tapes given to the signing oracle and $\mathcal{A}_1$. The success probability of $\mathcal{A}_1$ is taken over the space defined by Θ, Ω and random oracle $H, \bar{H}$. We fix the value of random tapes and the behavior of H. By the knowledge of SK_{P_i} and the analogous way with Lemma 6, $\mathcal{A}_1$ computes $\beta_i = (x_i(c'_i - c_i) + s'_i - s_i)/(\gamma_i - \gamma'_i)$. The equation is valid under the condition of $\gamma_i \neq \gamma'_i$. In the case of $\gamma_i = \gamma'_i$, $\mathcal{A}_1$ find w_i such that $(g_i^{\beta})^{\gamma'}(\bar{H}(M||w'_i||\mathbf{R}))^{-1} = (g_i^{\beta})^{\gamma}(\bar{H}(M||w_i||\mathbf{R}))^{-1}$. $\mathcal{B}_1$ returns w_i to break the collision resistance of $\bar{H}$. The analysis of the reduction cost is most of the same with 6. □

By the result of Lemma 6 and 7, we have proven Theorem 5. □

7 Discussions

7.1 Comparison

We show the comparison in the Table 1 among some k-out-of-n signature schemes.

The signature type is divided into three types: partition, secret-sharing and multivariate vectors.

Partition. This type is introduced by Bresson et al. [7]. Let $\pi = (\pi^1, \pi^2, \ldots, \pi^k)$ be a partition over $[1, n]$ in k subsets and $\mathbf{P}$ be a set of provers over $[1, n]$. If all proves belong to different subsets, then π is a fair partition with respect to $\mathbf{P}$. By using a perfect hash functions [4], we generate a fair partition for any set of cardinality k. It was shown in [4] that the size of (n, k) family of perfect functions is $p = 2^t \log(n)$. A super-ring consists of p partitions: fair-partition π_s and others $\{\pi_i\}_{i\in[1,p]\setminus s}$. Provers form sub-rings for $\{\pi_i\}_{i\in[1,p]\setminus s}$, then form a super-ring as if a simulation. By using the gap-value which forms a super-ring as a random numbers, provers form the sub-rings corresponding to fair-partitions. Therefore, the signature length in this scheme is exponential to k and it is non-flexible.

Secret-Sharing. There are many schemes based on secret-sharing. We give a naive construction for a type of (a, c, s)-Σ-protocol in [2] as an example. It is sufficient to demonstrate the basic idea and performance briefly. Each prover $i \in \mathbf{P}$ computes a_i and provers simulate $\{(a_i, c_i, s_i)\}_{i\in\mathbf{U}\setminus\mathbf{P}}$. Provers compute a common challenge, say c_0, by hashing all a_i's and message M. Then share c_o with a $(n-k)$-degree polynomial, say $\mathcal{P}$, that is consistent with all $n-k$ points (i, c_i) determined so far. Each prover $i \in \mathbf{P}$ computes s_i for challenge $c_i = \mathcal{P}(i)$ by using the secret-key. Therefore, the signature length in this scheme is linear to n and it is non-flexible.

Multivariate Vector. This type is introduced by Petzoldt et al. [18]. They extend the identification scheme in [20] to their threshold ring signature. Let $\mathcal{P}_i : \mathbb{F}^n \to \mathbb{F}^m$ be a *polar form* of the multivariate quadratic system which is viewed as a system parameter. Each prover $i \in \mathbf{P}$ chooses a private vector $s_i \in \mathbb{F}^n$ and creates a system $\mathcal{P}_i$ such that $\mathcal{P}_i(s_i) = 0$. Provers choose a vector $s_i = 0$ for $i \in \mathbf{U}\setminus\mathbf{P}$. Let $r_{(i,0)}, r_{(i,1)}$ be vectors such that $s_i = r_{(i,0)} + r_{(i,1)}$. We use them (and other values which are related with *polar form*) as a message in the commitment scheme of $\Sigma - protocol$. Verifier checks the number of the equations such that $Com(r_{(i,0)}) = Com(r_{(i,1)})$ is satisfied and considers it as $n-k$. Therefore, the signature length in this scheme is linear to n and depends on the number of rounds. Moreover, it is non-flexible.

7.2 Anonymity

We analyze the following two points.

Linkability. The notion of linkability was introduced by Liu, et al. [15]. Such signature scheme allow anyone to determine if two signatures (possibly generated with respect to different rings) are signed by the same member. This property is interesting in the term of functionality. However, in the term of anonymity, it is the property to avoid. Our scheme has the unlinkable property.

Insider Attacks for Anonymity. Our scheme does not have the anonymity for an exposure of short-term secret keys. However, the exposure of them does not influence on the signatures which are generated by **Sign** at different times since they are randomly picked with each function **Sign** call. On the other hand, our scheme have the anonymity for an exposure of (long-term) secret keys x_i since it does not affect the distribution of signatures which are generated by $\mathbf{\Sigma}$.

8 Conclusion

We proposed a k-out-of-n ring signature with flexible participation for provers. In our scheme, a user in a ring can sign on a same message gradually. We define its functional and security model, and prove our proposal secure in random oracle model. The user in a ring can add her signature at any time. Our scheme is convenient for petition or whistle blowing such as requiring more reliable information on the same message.

References

1. M. Abe, M. Ohkubo, and K. Suzuki. 1-out-of-n signatures from a variety of keys. In *ASIACRYPT*, volume 2501 of *LNCS*, pages 415–432. Springer-Verlag, 2002.
2. M. Abe, M. Ohkubo, and K. Suzuki. Efficient threshold signer-ambiguous signatures from variety of keys. In *IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer*, volume E87-A, No. 2, pages 471–479, 2004.
3. C. Aguilar Melchor, P. Cayrel, P. Gaborit, and F. Laguillaumie. A new efficient threshold ring signature scheme based on coding theory. *IEEE Trans. Inf. Theor.*, 57(7):4833–4842, 2011.
4. N. Alon, R. Yuster, and U. Zwick. Color-coding. *J. ACM*, 42(4):844–856, 1995.
5. M. Bellare and P. Rogaway. Random oracles are practical: a paradigm for designing efficient protocols. In *CCS*, pages 62–73. ACM, 1993.
6. A. Bender, J. Katz, and R. Morselli. Ring signatures: Stronger definitions, and constructions without random oracles. *J. Cryptol.*, 22(1):114–138, 2008.
7. E. Bresson, J. Stern, and M. Szydlo. Threshold ring signatures and applications to ad-hoc groups. In *CRYPTO*, volume 2442 of *LNCS*, pages 465–480. Springer-Verlag, 2002.
8. P.-L. Cayrel, R. Lindner, M. Rückert, and R. Silva. A lattice-based threshold ring signature scheme. In *LATINCRYPT*, volume 6212 of *LNCS*, pages 255–272. Springer-Verlag, 2010.
9. S. Chang, D. S. Wong, Y. Mu, and Z. Zhang. Certificateless threshold ring signature. *Inf. Sci.*, 179(20):3685–3696, 2009.
10. D. Chaum and E. van Heyst. Group signatures. In *EUROCRYPT*, volume 547 of *LNCS*, pages 257–265. Springer-Verlag, 1991.
11. L. Chen and T. P. Pedersen. New group signature schemes. In *EUROCRYPT*, volume 950 of *LNCS*, pages 171–181. Springer-Verlag, 1994.
12. E. Fujisaki and K. Suzuki. Exact t-out-of-n signer-ambiguous signature. *IEICE Technical Report. SITE*, 105(192):187–194, 2005.
13. E. Fujisaki and K. Suzuki. Traceable ring signature. *IEICE Transactions*, 91-A(1):83–93, 2008.
14. J. K. Liu, V. K. Wei, and D. S. Wong. A separable threshold ring signature scheme. In *ICISC*, volume 2971 of *LNCS*, pages 12–26. Springer-Verlag, 2003.
15. J. K. Liu, V. K. Wei, and D. S. Wong. Linkable spontaneous anonymous group signature for ad hoc groups. In *ACISP*, volume 3108 of *LNCS*, pages 325–335. Springer-Verlag, 2004.
16. K. Ohta and T. Okamoto. On concrete security treatment of signatures derived from identification. In *CRYPTO*, volume 1462 of *LNCSS*, pages 354–369. Springer-Verlag, 1998.

17. T. Okamoto and E. Okamoto. Proposal of "just" k-out-of-n signatures. *IPSJ. CSEC. Technical Report.*, 2003(74):151–156, 2003.
18. A. Petzoldt, S. Bulygin, and J. Buchmann. A multivariate based threshold ring signature scheme. Cryptology ePrint Archive, Report 2012/194, 2012.
19. R. L. Rivest, A. Shamir, and Y. Tauman. How to leak a secret. In *ASIACRYPT*, pages 552–565. Springer-Verlag, 2001.
20. K. Sakumoto, T. Shirai, and H. Hiwatari. Public-key identification schemes based on multivariate quadratic polynomials. In *CRYPTO*, volume 6841 of *LNCS*, page 703. Springer-Verlag, 2011.
21. C.-P. Schnorr. Efficient signature generation by smart cards. *J. Cryptology*, 4(3):161–174, 1991.
22. H. Shacham and B. Waters. Efficient ring signatures without random oracles. In *PKC*, volume 4450 of *LNCS*, pages 166–180. Springer-Verlag, 2007.
23. P. P. Tsang, M. H. Au, J. K. Liu, W. Susilo, and D. S. Wong. A suite of non-pairing id-based threshold ring signature schemes with different levels of anonymity (extended abstract). In *ProvSec*, volume 6402 of *LNCS*, pages 166–183. Springer-Verlag, 2010.
24. P. P. Tsang, V. K. Wei, T. K. Chan, M. H. Au, J. K. Liu, and D. S. Wong. Separable linkable threshold ring signatures. Cryptology ePrint Archive, Report 2004/267, 2004.
25. R. Tso, X. Yi, T. Ito, T. Okamoto, and E. Okamoto. Design and analysis of "flexible" k-out-of-n signatures. In *ATC*, volume 6407 of *LNCS*, pages 255–267. Springer-Verlag, 2010.
26. M. Yamaguchi, R. Tso, T. Okamoto, and E. Okamoto. Flexible k-out-of-n ring signature with multiple structures. *Symposium on Cryptography and Information Security (SCIS), 2A2-4*, 2013.
27. T. H. Yuen, J. K. Liu, M. H. Au, W. Susilo, and J. Zhou. Threshold ring signature without random oracles. In *ASIACCS*, pages 261–267. ACM, 2011.