2011

Threshold ring signature without random oracles

Tsz Hon Yuen
University of Wollongong, thy738@uow.edu.au

Joseph K. Liu
Institute for Infocomm Research, Singapore, ksliu@i2r.a-star.edu.sg

Man Ho Allen Au
University of Wollongong, aau@uow.edu.au

Willy Susilo
University of Wollongong, wsusilo@uow.edu.au

Jianying Zhou
Institute for Infocomm Research, Singapore, jyzhou@i2r.a-star.edu.sg

Publication Details

Yuen, T., Liu, J. K., Au, M., Susilo, W. & Zhou, J. (2011). Threshold ring signature without random oracles. Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security (pp. 261-267). NY, USA: ACM.

Threshold ring signature without random oracles

Abstract
In this paper, we present the notion and construction of threshold ring signature without random oracles. This is the first scheme in the literature that is proven secure in the standard model. Our scheme extends the Shacham-Waters signature from PKC 2007 in a non-trivial way. We note that our technique is specifically designed to achieve a threshold ring signature in the standard model. Interestingly, we can still maintain the signature size to be the same as the Shacham-Waters signature, while only a tiny computation cost is added.

Keywords
without, signature, ring, threshold, random, oracles

Disciplines
Physical Sciences and Mathematics

Publication Details
Yuen, T., Liu, J. K., Au, M., Susilo, W. & Zhou, J. (2011). Threshold ring signature without random oracles. Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security (pp. 261-267). NY, USA: ACM.

Threshold Ring Signature without Random Oracles

Tsz Hon Yuen
School of Computer Science and Software Engineering
University of Wollongong, Australia
thy738@uow.edu.au

Joseph K. Liu
Cryptography and Security Department
Institute for Infocomm Research, Singapore
ksliu@i2r.a-star.edu.sg

Man Ho Au
School of Computer Science and Software Engineering
University of Wollongong, Australia
aau@uow.edu.au

Willy Susilo
School of Computer Science and Software Engineering
University of Wollongong, Australia
wsusilo@uow.edu.au

Jianying Zhou
Cryptography and Security Department
Institute for Infocomm Research, Singapore
jyzhou@i2r.a-star.edu.sg

ABSTRACT

In this paper, we present the notion and construction of threshold ring signature without random oracles. This is the *first scheme* in the literature that is proven secure in the standard model. Our scheme extends the Shacham-Waters signature from PKC 2007 in a non-trivial way. We note that our technique is specifically designed to achieve a threshold ring signature in the standard model. Interestingly, we can still maintain the signature size to be the same as the Shacham-Waters signature, while only a tiny computation cost is added.

Categories and Subject Descriptors

E.3 [**Data Encryption**]: Public key cryptosystems

General Terms

Theory

Keywords

ring signatures, threshold ring signatures, anonymity

1. INTRODUCTION

RING SIGNATURE. A ring signature scheme (such as [24, 1, 32, 6, 30, 19, 16]) allows members of a group to sign messages on behalf of the group without the need to reveal their identities, *i.e.*, providing signer anonymity. Additionally, it is not possible to decide whether two signatures have been issued by the same group member. Different from a group signature scheme (such as [13, 9, 3]), the group formation is spontaneous and there exists no group manager to revoke the identity of the signer. That is, 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 signature schemes could be used for whistle blowing [24], anonymous membership authentication for ad hoc groups [8] 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 provided by a group member while they will not be able to 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.

Ring signature scheme can be used to derive other primitives as well. It had been utilized to construct non-interactive deniable ring authentication [27], perfect concurrent signature [28] and multi-designated verifiers signature [21].

Many reductionist security proofs used the random oracle model [4]. Several papers proved that some popular cryptosystems previously proved secure in the random oracle are actually provably insecure when the random oracle is instantiated by any real-world hashing functions [10, 2]. Thus, it is natural to design a practical ring signature scheme provably secure without requiring random oracles.

Subsequently, there are some ring signature schemes that do not rely on random oracles exist in the literature. Xu et al. [31] described a ring signature scheme in the standard model. But the proof is not rigorous and is apparently flawed [5]. Chow et al. [15] gave a ring signature scheme with proof in the standard model, though it is based on a strong new assumption. Bender et al. [5] presented a ring signature secure in the standard model assuming trapdoor permutations exists. Their scheme uses generic ZAPs for NP as a building

ASIACCS '11, March 22–24, 2011, Hong Kong, China.

block, which may not be practical. Shacham and Waters [26] proposed an efficient ring signature scheme without using random oracles, based on standard assumption. They rely on composite order pairing that results for a trusted set-up procedure. Very recently, Schäge and Schwenk [25] gave another ring signature scheme in the standard model using basic assumption. In contrast to the previous construction, they used prime order pairing instead. However, their security model does not allow the adversary to query any private key.

All the above mentioned ring signature schemes only allow one single signer, which is also known as 1-out-of-n ring signature scheme [1].

Threshold Ring Signature. A (d, n)-threshold ring signature has the similar notion to the (1-out-of-n) ring signature. First, a (d, n)-threshold ring signature scheme requires at least t signers to work jointly for generating a signature. Second, the anonymity of signers is preserved both inside and outside the signing group. Third, those t participating signers can choose any set of n entities including themselves without getting any consent from those diversion group members. The first threshold ring signature was proposed by Bresson et al. [8] in 2002 which is followed by Wong et al. [30] in 2003. Both of them extend the 1-out-of-n ring signature from [24] in a different way. However, the idea of proving "*Knowing d solutions out of n problem instance*" [17] was proposed in the early 90s. Liu et al. [22] changed the idea into threshold ring signature for separate key types. Subsequently, different types of setting or construction such as ID-based [14], certificateless-based [12], code-based [23, 18] and lattice-based [11] have also been proposed. However, all previous threshold ring signature schemes in the literature (regardless the underlying cryptosystem or construction) can be proven secure in the random oracle or ideal cipher model only[1].

1.1 Contribution

In this paper, we propose the first threshold ring signature scheme provable secure without random oracles. It is a threshold extension of the Shacham-Waters (SW) signature [26]. However, we have to note that the extension is not trivial. The typical secret sharing technique cannot be used in the ring signature case. The modified polynomial interpolation technique (e.g. [17, 22, 29]) requires random oracle to instantiate a signature scheme. Thus, we emphasize that our technique is specially designed for non-random oracle security proof. Additionally, we can still maintain the signature size to be the same as the SW signature, while only a tiny computation cost is added.

2. PRELIMINARIES

2.1 Pairings

We make use of bilinear groups of composite order. Let n be a composite number with factorization $n = pq$. We have

- $\mathbb{G}$ is a multiplicative cyclic groups of order n.
- $\mathbb{G}_p$ is its cyclic order-p subgroup, and $\mathbb{G}_q$ is its cyclic order-q subgroup
- g is a generator of $\mathbb{G}$, while h is a generator of $\mathbb{G}_q$.
- $\mathbb{G}_T$ is a multiplicative group of order n.
- $\hat{e}$ is a bilinear map such that $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ with the following properties:
  - *Bilinearity*: For all $u, v \in \mathbb{G}$, and $a, b \in \mathbb{Z}$,
    $$\hat{e}(u^a, v^b) = \hat{e}(u, v)^{ab}.$$
  - *Non-degeneracy*: $\langle \hat{e}(g, g) \rangle = \mathbb{G}_T$ whenever $\langle g \rangle = \mathbb{G}$.
  - *Computability*: It is efficient to compute $\hat{e}(u, v)$ for all $u, v \in \mathbb{G}$.
- $\mathbb{G}_{T,p}$ and $\mathbb{G}_{T,q}$ are the $\mathbb{G}_T$-subgroups of order p and q, respectively.
- The group operations on $\mathbb{G}$ and $\mathbb{G}_T$ can be performed efficiently.
- Bit strings corresponding to elements of $\mathbb{G}$ and of $\mathbb{G}_T$ can be recognized efficiently.

2.2 Mathematical Assumptions

For our scheme, we assume two problems are difficult to solve in the setting described above: computational Diffie-Hellman in $\mathbb{G}_p$ and the Subgroup Decision Problem.

Definition 1 (Computational Diffie-Hellman in $\mathbb{G}_p$). *Given the tuple (r, r^a, r^b), where $r \in_R \mathbb{G}_p$, and $a, b \in_R \mathbb{Z}_p$, compute and output r^{ab}. In the composite setting one is additionally given the description of the larger group $\mathbb{G}$, including the factorization (p, q) of its order n.*

Definition 2 (Subgroup Decision). *Given w selected at random either from $\mathbb{G}$ (with probability $1/2$) or from $\mathbb{G}_q$ (with probability $1/2$), decide whether w is in $\mathbb{G}_q$. For this problem one is given the description of $\mathbb{G}$, but not given the factorization of n.*

The assumptions are formalized by measuring an adversary's success probability for computational Diffie-Hellman and an adversary's guessing advantage for the subgroup decision problem. Note that if CDH in $\mathbb{G}_p$ as we have formulated it is hard then so is CDH in $\mathbb{G}$. The assumption that the subgroup decision problem is hard is called Subgroup Hiding (SGH) assumption, and was introduced by Boneh et al [7].

3. SECURITY MODEL

We give our security model and define relevant security notions.

3.1 Syntax of threshold ring signature

A *threshold ring signature*, (TRS) scheme, is a tuple of four algorithms (KeyGen, Sign and Verify.

- $(sk_i, pk_i) \leftarrow \mathsf{KeyGen}(\lambda)$ is a PPT algorithm which, on input a security parameter $\lambda \in \mathbb{N}$, outputs a private/public key pair (sk_i, pk_i). We denote by $\mathcal{SK}$ and $\mathcal{PK}$ the domains of possible secret keys and public keys, resp. When we say that a public key corresponds to a secret key or vice versa, we mean that the secret/public key pair is an output of KeyGen.

[1]Although Han et al. [20] claimed their threshold ring signature scheme is secure in the standard mode, Tsang et al. [29] showed that their proof is incorrect. We do not regard [20] as a provable secure scheme.

- $\mathsf{param} \leftarrow \mathsf{Setup}(\lambda)$ is a PPT algorithm which, on input a security parameter λ, outputs the set of security parameters param which includes λ.
- $\sigma'=(n,d,\mathcal{Y},\sigma) \leftarrow \mathsf{Sign}(e, n, d, \mathcal{Y}, \mathcal{X}, M)$ which, on input a 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.
- $\mathsf{accept/reject} \leftarrow \mathsf{Verify}(n, d, \mathcal{Y}, M, \sigma)$ which, on input a group size n, threshold $d \in \{1,\ldots,n\}$, a set $\mathcal{Y}$ of n public keys in $\mathcal{PK}$, a message-signature pair (M,σ) returns accept or reject. If accept, the message-signature pair is *valid*.

3.1.1 Correctness.

TRS schemes must satisfy: *Verification Correctness*. That is, all signatures signed according to specification are accepted during verification.

3.2 Notions of Security of threshold ring signature

Security of TRS schemes has two aspects: unforgeability 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.
- $sk_i \leftarrow \mathcal{CO}(pk_i)$. The *Corruption Oracle*, on input a public key $pk_i \in \mathcal{PK}$ that is a query output of $\mathcal{JO}$, returns the corresponding secret key $sk_i \in \mathcal{SK}$.
- $\sigma' \leftarrow \mathcal{SO}(n, d, \mathcal{Y}, \mathcal{V}, M)$. The *Signing Oracle*, on input a group size n, a threshold $d \in \{1,\ldots,n\}$, a set $\mathcal{Y}$ of n public keys, a signer subset $\mathcal{V}$ of $\mathcal{Y}$ with $|\mathcal{V}| = d$, and a message M, returns a valid signature σ'.

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

1. Unforgeability.

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

   (a) $\mathcal{S}$ generates and gives $\mathcal{A}$ the system parameters param.

   (b) $\mathcal{A}$ may query the oracles according to any adaptive strategy.

   (c) $\mathcal{A}$ gives $\mathcal{S}$ a group size $n \in \mathbb{N}$, a threshold $d \in \{1,\ldots,n\}$, a set $\mathcal{Y}$ of n public keys in $\mathcal{PK}$, a message $M \in \mathcal{M}$ and a signature $\sigma \in \Sigma$.

   $\mathcal{A}$ wins the game if:

   (1) $\mathsf{Verify}(\cdot)$=accept.

   (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 $\mathcal{CO}$.

   (4) $(M, \mathcal{Y})$ is not a query input to $\mathcal{SO}$.

   We denote by

   $$\mathbf{Adv}^{unf}_{\mathcal{A}}(\lambda) = \Pr[\mathcal{A} \text{ wins the game }].$$

   Definition 3 (unforgeability). *A TRS scheme is unforgeable if for all PPT adversary $\mathcal{A}$, $\mathbf{Adv}^{unf}_{\mathcal{A}}(\lambda)$ is negligible.*

2. Anonymity.

   Anonymity for TRS 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 $\mathcal{SO}$:

   (a) $\mathcal{S}$ generates and gives $\mathcal{A}$ the system parameters param.

   (b) $\mathcal{A}$ may query the oracles according to any adaptive strategy. Suppose $\mathcal{A}$ makes a total number of v queries to $\mathcal{CO}$. The restriction is that: $v < n-d$.

   (c) $\mathcal{A}$ gives $\mathcal{S}$ a group size n, threshold $d \in \{1,\ldots,n\}$, message M, and a set $\mathcal{Y}$ of n public keys all of which are query outputs of $\mathcal{JO}$. $\mathcal{S}$ picks randomly a subset $\mathcal{V}$ of $\mathcal{Y}$ with $|\mathcal{V}| = d$, such that $\mathcal{V}$ is not contained in any of the queries to $\mathcal{SO}$ and $\mathcal{CO}$. Let $\mathcal{X}$ be a set of secret keys with $|\mathcal{X}| = d$ and whose corresponding public keys are all contained in $\mathcal{V}$. $\mathcal{S}$ computes $\sigma' = \mathsf{Sign}\ (\ n,\ d,\ \mathcal{Y},\ \mathcal{V},\ \mathcal{X},\ M)$.

   (d) $\mathcal{A}$ queries the oracles adaptively. Suppose $\mathcal{A}$ makes a total number of v' queries to $\mathcal{CO}$. The restriction is that: $v' < n - d - v$. If any of the queries to $\mathcal{SO}$ or $\mathcal{CO}$ contains a public key y such that $pk \in \mathcal{Y}$, $\mathcal{S}$ halts.

   (e) $\mathcal{A}$ outputs an index $\hat{\pi}$.

   We denote by

   $$\mathbf{Adv}^{Anon}_{\mathcal{A}}(\lambda) = \Pr[\hat{\pi} \in \mathcal{Y}] - \frac{d}{n-(v+v')}.$$

   Definition 4 (Anonymity). *A TRS scheme is anonymous if for any PPT adversary $\mathcal{A}$, $\mathbf{Adv}^{Anon}_{\mathcal{A}}(\lambda)$ is negligible.*

Summarizing we have:

Definition 5 (Security of TRS Schemes). *A TRS scheme is* secure *if it is unforgeable and anonymous.*

4. OUR PROPOSED THRESHOLD RING SIGNATURE SCHEME

4.1 Construction

We extend the 1-out-of-n SW ring signature scheme [26] into a d-out-of-n threshold setting.

- Setup: The setup algorithm runs the bilinear group generator $(N = pq, \mathbb{G}, \mathbb{G}_T, \hat{e}) \leftarrow \mathcal{G}(1^\lambda)$. Suppose the group generator $\mathcal{G}$ also gives the generators $g_1, B_0, u, u_1, \ldots, u_k \in \mathbb{G}$, $h_1 \in \mathbb{G}_q$ and $\alpha \in \mathbb{Z}_N$. Set $g_2 = g_1^\alpha$ and

$h_2 = h_1^{\alpha}$. Let $H : \mathbb{N} \times \mathbb{G}^* \times \{0,1\}^* \to \{0,1\}^k$ be a collision resistant hash function. The public parameters are

$$(N, \mathbb{G}, \mathbb{G}_T, \hat{e}, g_1, g_2, B_0, h_1, h_2, u, u_1, \ldots, u_k, H).$$

Everyone can check the validity of g_1, g_2, h_1, h_2 using pairings.

- KeyGen: For user i, he picks a random $x_i \in \mathbb{Z}_N$. His public key is $g_1^{s_i}$ and his secret key is $g_2^{s_i}$.

- Sign: Suppose $\mathcal{Y} = \{pk_1, \ldots, pk_n\}$ is the user ring. $\mathcal{X}$ is the set of private keys of d participating signers, who cooperate to generate the ring signature for the message M. Without loss of generality, suppose that $\{1, 2, \ldots, d\}$ is the indices of participating signers, and $\{d+1, ..., n\}$ is the indices of non-signers.

  Define f_i such that

$$f_i = \begin{cases} 1 & \text{if } i = 1, \ldots, d, \\ 0 & \text{if } i = d+1, \ldots, n. \end{cases}$$

  1. For $i = 1, \ldots, n$, one of the signer picks $x_i \in_R \mathbb{Z}_N$ and sets

$$C_i = \left(\frac{g_1^{s_i}}{B_0}\right)^{f_i} h_1^{x_i}, \quad \pi_i = \left(\left(\frac{g_1^{s_i}}{B_0}\right)^{2f_i - 1} h_1^{x_i}\right)^{x_i}.$$

     Let $C = \prod_{i=1}^n C_i$. Then we have

$$B_0^d C = h_1^x \prod_{i=1}^d g_1^{s_i} \qquad \text{where} \qquad x = \sum_{i=1}^n x_i.$$

  2. Each signer i computes $(m_1, \ldots, m_k) = H(d, \mathcal{Y}, M)$. He picks a random $r_i \in \mathbb{Z}_N$ and computes

$$S_{1,i} = g_2^{s_i} (u \prod_{j=1}^k u_j^{m_j})^{r_i}, \quad S_{2,i} = g_1^{r_i}.$$

     Signer i sends $(S_{1,i}, S_{2,i})$ to the signer in step 1.

  3. After collecting $(S_{1,i}, S_{2,i})$ from the t signers, calculate

$$S_1 = h_2^x \prod_{i=1}^d S_{1,i}, \quad S_2 = \prod_{i=1}^d S_{2,i}.$$

     The signature is $(S_1, S_2, \{C_i, \pi_i\}_{i=1}^n)$.

- Verify: On input $(n, d, \mathcal{Y}, M, \sigma)$, first compute $(m_1, \ldots, m_k) = H(d, \mathcal{Y}, M)$. For $i = 1, \ldots, n$, check if

$$\hat{e}(C_i, C_i) = \hat{e}(h_1, \pi_i) \cdot \hat{e}(C_i, \frac{g_1^{s_i}}{B_0}).$$

  If they are true, compute $C = \prod_{i=1}^n C_i$ and check if:

$$\hat{e}(S_1, g_1) = \hat{e}(S_2, u \prod_{j=1}^k u_j^{m_j}) \cdot \hat{e}(g_2, B_0^d C).$$

Check correctness:

$$\begin{aligned}
&\hat{e}(S_2, u \prod_{j=1}^k u_j^{m_j}) \cdot \hat{e}(g_2, B_0^d C) \\
=&\hat{e}(\prod_{i=1}^d S_{2,i}, u \prod_{j=1}^k u_j^{m_j}) \cdot \hat{e}(g_2, h_1^x \prod_{i=1}^d g_1^{s_i}) \\
=&\hat{e}(\prod_{i=1}^d g_1^{r_i}, u \prod_{j=1}^k u_j^{m_j}) \cdot \prod_{i=1}^d \hat{e}(g_2, g_1^{s_i}) \cdot \hat{e}(g_2, h_1^x) \\
=&\hat{e}(g_1, (u \prod_{j=1}^k u_j^{m_j})^{\sum_{i=1}^d r_i}) \cdot \prod_{i=1}^d \hat{e}(g_2^{s_i}, g_1) \cdot \hat{e}(g_2, h_1^x) \\
=&\hat{e}(g_1, \prod_{i=1}^d g_2^{s_i} (u \prod_{j=1}^k u_j^{m_j})^{\sum_{i=1}^d r_i}) \cdot \hat{e}(g_1, h_2^x) \\
=&\hat{e}(g_1, h_2^x \prod_{i=1}^d S_{1,i}) \\
=&\hat{e}(g_1, S_1)
\end{aligned}$$

4.2 Security Proof

THEOREM 1. *The threshold ring signature scheme is unforgeable against insider corruption if the CDH assumption holds in $\mathbb{G}_p$.*

PROOF. Setup. The simulator $\mathcal{B}$ runs the bilinear group generator $(N = pq, \mathbb{G}, \mathbb{G}_T, \hat{e}) \leftarrow \mathcal{G}(1^\lambda)$. $\mathcal{B}$ is given the CDH problem instance $(g, g^a, g^b) \in \mathbb{G}_p^3$ and is asked to output g^{ab}. $\mathcal{B}$ first sets an integer, $\mu = 4q_e$, and chooses an integer, κ, uniformly at random between 0 and k. $\mathcal{B}$ picks $x', x_1, \ldots, x_k$ uniformly at random between 0 and $\mu - 1$. $\mathcal{B}$ randomly picks a $\gamma \in \mathbb{Z}_N$ and sets $z_1 = g^{\frac{p\gamma}{q}}$. Since $g \in \mathbb{G}_p$, z_1 is in $\mathbb{G}_q$. Also z_1^b can be computed from g^b.

$\mathcal{B}$ randomly picks a generator $h_1 \in \mathbb{G}_q$. $\mathcal{B}$ randomly picks $y', y_1, \ldots, y_k, \alpha, \beta \in \mathbb{Z}_N$ and sets

$$\begin{aligned}
&g_1 = g z_1, \quad g_2 = g^a z_1^{\alpha}, \quad u = g_2^{N - \kappa\mu + x'} g^{y'}, \\
&u_1 = g_2^{x_1} g^{y_1}, \quad \ldots, \quad u_k = g_2^{x_k} g^{y_k}, \quad h_2 = h_1^{\alpha}, \quad B_0 = h_1^{\beta}.
\end{aligned}$$

Note that $\hat{e}(g_1, h_2) = \hat{e}(z_1, h_1^{\alpha}) = \hat{e}(z_1^{\alpha}, h_1) = \hat{e}(g_2, h_1)$, since $\hat{e}(g, h_1) = 1$. Finally, $\mathcal{B}$ randomly chooses a collision resistant hash function $H : \mathbb{N} \times \mathbb{G}^* \times \{0,1\}^* \to \{0,1\}^k$.

Then $\mathcal{B}$ gives the public parameters

$$(N, \mathbb{G}, \mathbb{G}_T, \hat{e}, g_1, g_2, B_0, h_1, h_2, u, u_1, \ldots, u_k, H)$$

to the adversary $\mathcal{A}$. For a message $m = \{m_1, \ldots, m_k\}$, we define

$$F(m) = (N - \mu\kappa) + x' + \sum_{i=1}^k x_i m_i, \quad J(m) = y' + \sum_{i=1}^k y_i m_i.$$

Assume $\mathcal{B}$ picks τ as the challenge signer. For $i = 1, \ldots, n$, $\mathcal{B}$ picks random $s_i \in \mathbb{Z}_N$ and sets:

$$pk_i = \begin{cases} g_1^{s_i} & \text{if } i \neq \tau, \\ g^b z_1^b & \text{if } i = \tau. \end{cases}$$

$\mathcal{B}$ stores the set of public keys $\{pk_i\}_{i=1}^n$.

Oracle Simulation. $\mathcal{B}$ simulates the oracles as follows:

- $\mathcal{JO}$: on the i-th query, $\mathcal{B}$ returns pk_i.

- $\mathcal{CO}(pk_i)$: If $i = \tau$, $\mathcal{B}$ declares failure and exits. Otherwise, $\mathcal{B}$ returns $g_2^{s_i}$.
- $\mathcal{SO}(n, d, \mathcal{Y}, \mathcal{V}, M)$: On input a message M, a set of n public keys $\mathcal{Y} = \{pk_i'\}_{i=1}^n$, and a set of d signers $\mathcal{V}$, $\mathcal{B}$ calculates (C_i, π_i) according to the Sign algorithm. Note that no secret key is required to generate (C_i, π_i). Then we have

$$B_0^d C = h_1^x \prod_{i|pk_i \in \mathcal{V}} g_1^{s_i}.$$

Denote $m = H(d, \mathcal{Y}, M)$. We also write m into k bits $\{m_1, \ldots, m_k\}$. If $x' + \sum_{i=1}^k x_i m_i \equiv 0 \mod \mu$, then $\mathcal{B}$ aborts. For all $pk_i \in \mathcal{V}$ and $i \neq \tau$, $\mathcal{B}$ calculates all $(S_{1,i}, S_{2,i})$ according to the Sign algorithm. If $pk_\tau \in \mathcal{V}$,$\mathcal{B}$ chooses a random $r_\tau \in \mathbb{Z}_N$ and calculates

$$S_{1,\tau} = (g^b)^{\frac{-J(m)}{F(m)}} (u \prod_{j=1}^k u_j^{m_j})^{r_\tau},$$

$$S_{2,\tau} = (g^b z_1^b)^{\frac{-1}{F(m)}} (g z_1)^{r_\tau}.$$

Let $\bar{r} = r_\tau - \frac{b}{F(m)}$, then

$$\begin{aligned} S_{1,\tau} &= (g^b)^{\frac{-J(m)}{F(m)}} (u \prod_{j=1}^k u_j^{m_j})^{r_\tau}, \\ &= g_2^b (g_2^{F(m)} g^{J(m)})^{r_\tau - \frac{b}{F(m)}}, \\ &= g_2^b (u \prod_{j=1}^k u_j^{m_j})^{\bar{r}}, \end{aligned}$$

The simulator will be able to perform this computation if and only if $F(m) \neq 0 \mod N$. For ease of analysis the simulator will only continue in the sufficient condition where $x' + \sum_{i=1}^k x_i m_i \neq 0 \mod \mu$. (If we have $x' + \sum_{i=1}^k x_i m_i \neq 0 \mod \mu$, this implies $F(m) \neq 0 \mod N$ since we can assume $N > k\mu$ for any reasonable values of N, k, and μ).

Finally, $\mathcal{B}$ calculates the rest of the signature according to the Sign algorithm.

Output. $\mathcal{A}$ returns $(n^*, d^*, \mathcal{Y}^*, M^*, \sigma^*)$. Denote $m^* = (m_1^*, \ldots, m_k^*) = H(d^*, \mathcal{Y}^*, M^*)$. Note that this hash value is different from previous m in various $\mathcal{SO}$ queries, since $(d^*, \mathcal{Y}^*, M^*)$ cannot be the input of previous $\mathcal{SO}$ queries and H is a collision resistant hash function. If $pk_\tau \notin \mathcal{Y}^*$ and $x' + \sum_{i=1}^k x_i m_i^* \neq \mu\kappa$, then $\mathcal{B}$ aborts. Otherwise, WLOG, we assume that pk_τ is at the position τ of the signature σ^*. Since σ^* is a valid signature, then

$$\hat{e}(S_1^*, g_1) = \hat{e}(S_2^*, u \prod_{j=1}^k u_j^{m_j^*}) \cdot \hat{e}(g_2, B_0^{d^*} \prod_{i=1}^{n^*} C_i^*), \quad (1)$$

$$\hat{e}(C_i^*, C_i^*) = \hat{e}(h_1, \pi_i^*) \cdot \hat{e}(C_i^*, \frac{pk_i}{B_0}). \quad (2)$$

for $i = 1, \ldots, n^*$. Since $\hat{e}(h_1, \pi_i^*)$ has order q in $\mathbb{G}_T$, therefore either C_i^* or $\frac{B_0 C_i^*}{pk_i}$ has order q from equation 2. $\mathcal{B}$ checks if $(C_i^*)^q = 0$. If it is true, then C_i^* has prder q and then $\mathcal{B}$ sets $f_i = 0$. Otherwise, $\frac{B_0 C_i^*}{pk_i}$ has order q and then $\mathcal{B}$ sets $f_i = 1$. It follows that $C_i^* = (\frac{pk_i}{B_0})^{f_i} z_1^{r_i'}$ for some unknown r_i', no matter $f_i = 0/1$. If $f_\tau = 0$, $\mathcal{B}$ aborts.

Let $\delta \in \mathbb{Z}_N$ such that $\delta = 0 \mod q$ and $\delta = 1 \mod p$. If we raise equation 1 to the δ-th power, then we have

$$\begin{aligned} \hat{e}(S_1^*, g_1)^\delta &= \hat{e}(S_2^*, u \prod_{j=1}^k u_j^{m_j^*})^\delta \cdot \hat{e}(g_2, B_0^{d^*} \prod_{i|i \in \mathcal{Y}^*} C_i^*)^\delta, \\ \hat{e}(S_1^*, g)^\delta &= \hat{e}(S_2^*, g^{J(m^*)})^\delta \cdot \hat{e}(g^a, B_0^{d^*} \prod_{i|i \in \mathcal{Y}^*} (\frac{pk_i}{B_0})^{f_i})^\delta, \quad (3) \\ \hat{e}(S_1^*, g)^\delta &= \hat{e}(S_2^*, g^{J(m^*)})^\delta \cdot \hat{e}(g^a, \prod_{i|i \in \mathcal{Y}^*, i \neq \tau} (g^{s_i})^{f_i} \cdot g^b)^\delta. \end{aligned} \quad (4)$$

For equation 3, note that

$$u \prod_{j=1}^k u_j^{m_j^*} = g_2^{F(m^*)} g^{J(m^*)} = g^{J(m^*)},$$

since $x' + \sum_{i=1}^k x_i m_i^* = \mu\kappa$. Also

$$C_i^{*\delta} = ((\frac{pk_i}{B_0})^{f_i} z_1^{r_i'})^\delta = (\frac{pk_i}{B_0})^{f_i \delta},$$

since $z_1 \in \mathbb{G}_q$.

From equation 4, we can see that

$$S_1^{*\delta} = (S_2^{*J(m^*)} \cdot (\prod_{i|i \in \mathcal{Y}^*, i \neq \tau} g^{s_i f_i} \cdot g^b)^a)^\delta,$$

Therefore $\mathcal{B}$ can output

$$A = (S_1^* (S_2^*)^{-J(m^*)} \prod_{i|i \in \mathcal{Y}^*, i \neq \tau} (g^a)^{-s_i f_i})^\delta,$$

as the solution to the CDH problem.

Analysis. Following the probability analysis of Waters signature, the probability of $F(m) \neq 0 \mod N$ during signing oracle query and $x + \sum_{i=1}^k x_i m_i^* = \mu\kappa$ is at least $\frac{1}{8(k+1)q_s}$. The probability of not asking pk_τ in the corruption oracle is $1 - \frac{q_c}{n}$. The probability of $f_\tau = 1$ in the output phase is $\frac{d^*}{n^*}$. Therefore $\mathcal{B}$ solves the CDH problem with probability

$$\epsilon \geq \frac{d^*}{8(k+1)q_s n^*}(1 - \frac{q_c}{n}),$$

where q_s, q_c, n is the number of $\mathcal{SO}$, $\mathcal{CO}$ and $\mathcal{JO}$ respectively. □

THEOREM 2. *The threshold ring signature scheme is anonymous against full key exposure if the subgroup hiding assumption holds.*

PROOF. Setup. The simulator $\mathcal{B}$ is given the subgroup decision problem instance $(N, \mathbb{G}, \mathbb{G}_T, \hat{e}, g, h)$. $\mathcal{B}$ is asked to determine whether $h \in \mathbb{G}$ or $h \in \mathbb{G}_q$. $\mathcal{B}$ randomly picks the generators $u, u_1, \ldots u_k, B_0 \in \mathbb{G}$ and $\alpha \in \mathbb{Z}_N$. $\mathcal{B}$ sets

$$g_1 = g, \quad g_2 = g_1^\alpha, \quad h_1 = h, \quad h_2 = h^\alpha.$$

Finally, $\mathcal{B}$ randomly chooses a collision resistant hash function $H : \mathbb{N} \times \mathbb{G}^* \times \{0,1\}^* \to \{0,1\}^k$. Then $\mathcal{B}$ gives the public parameters

$$(N, \mathbb{G}, \mathbb{G}_T, \hat{e}, g_1, g_2, B_0, h_1, h_2, u, u_1, \ldots, u_k, H)$$

to the adversary $\mathcal{A}$.

For $i = 1, \ldots, n$, $\mathcal{B}$ picks random $s_i \in \mathbb{Z}_N$ and sets:

$$pk_i = g_1^{s_i}, \quad sk_i = g_2^{s_i}.$$

$\mathcal{B}$ stores the set of public keys and secret keys $\{pk_i, sk_i\}_{i=1}^{n}$.

Oracle Simulation. $\mathcal{B}$ simulates the oracles as follows:

- $\mathcal{JO}$: on the i-th query, $\mathcal{B}$ returns pk_i.
- $\mathcal{CO}(pk_i)$: $\mathcal{B}$ returns $g_2^{s_i}$.
- $\mathcal{SO}(n, d, \mathcal{Y}, \mathcal{V}, M)$: $\mathcal{B}$ answers by running the Sign algorithm honestly.

Challenge. At some point, $\mathcal{A}$ outputs a message M^*, a set of n^* public keys $\mathcal{Y}^*$ and a threshold d^*. $\mathcal{B}$ picks a random subset $\mathcal{V}^*$ of $\mathcal{Y}^*$ with $|\mathcal{V}^*| = d^*$, such that $\mathcal{V}^*$ is not contained in any query to $\mathcal{CO}$. $\mathcal{B}$ uses the secret keys of $\mathcal{V}^*$ to run the Sign algorithm to obtain the signature σ^*. $\mathcal{B}$ gives σ^* to $\mathcal{A}$.

Output. If $\mathcal{A}$ can correct guess the index $\hat{\pi}$, then $\mathcal{B}$ outputs $h \in \mathbb{G}_q$. Otherwise, $\mathcal{B}$ outputs $h \in \mathbb{G}$.

Analysis. Suppose the challenge signature is $(S_1^*, S_2^*, \{C_i^*, \pi_i^*\}_{i=1}^{n^*})$ and $\mathcal{Y}^* = \{pk_1^*, \ldots, pk_{n^*}^*\}$. If h_1 is a generator of $\mathbb{G}$, there exist $x_i, \bar{x}_i \in \mathbb{Z}_N$ such that $C_i^* = (\frac{pk_i^*}{B_0})h_1^{x_i} = h_1^{\bar{x}_i}$. Then $x_i, \bar{x}_i$ correspond to the case $f_i^* = 0$ or 1 respectively. Denote by $(\pi_i^* | f_i^* = b)$ the value of π_i^* if f_i is set to $b \in \{0, 1\}$. Then

$$(\pi_i^* | f_i^* = 0) = \big((\frac{pk_i^*}{B_0})h_1^{x_i}\big)^{x_i} = (h_1^{\bar{x}_i})^{x_i}$$
$$= (h_1^{x_i})^{\bar{x}_i} = \big((\frac{pk_i^*}{B_0})^{-1}h_1^{\bar{x}_i}\big)^{\bar{x}_i} = (\pi_i^* | f_i^* = 1).$$

Therefore $\{C_i^*, \pi_i^*\}_{i=1}^{n^*}$ has no information about the real signer if $h \in \mathbb{G}$.

On the other hand, S_2^* is computed by random numbers only and do not have information about the real signer. Finally, S_1^* is determined by the verification equation

$$\hat{e}(S_1^*, g_1) = \hat{e}(S_2^*, u\prod_{j=1}^{k} u_j^{m_j}) \cdot \hat{e}(g_2, B_0^{d^*}\prod_{i=1}^{n^*} C_i^*).$$

Hence, it leaks no useful information about the set $\mathcal{V}^*$. Therefore if $\mathcal{A}$ wins the game, $\mathcal{B}$ outputs $h \in \mathbb{G}_q$. □

4.2.1 Insider Security for Anonymity

From the above security proof of anonymity, we can see that the adversary cannot win the game even if it is given all user secret keys (which is known as the full key exposure attack [5]). According to the security model, the challenge signature σ^* is solely generated by the simulator and the adversary does not obtain any internal information during the generation of σ^*.

However, our security model does not consider the insider security during the generation of threshold ring signatures. The adversary may use the information transferred between different signers to break the anonymity. In our construction, $S_{1,i}$ and $S_{2,i}$ are sent from user i to a central signer who runs step 1 and step 3 of the Sign algorithm. If any $S_{1,i}$ and $S_{2,i}$ is eavesdropped, or the (malicious) central signer releases the $S_{1,i}$ and $S_{2,i}$, then the anonymity of user i is lost. Therefore, our current security model for anonymity assumes that the communication channel between signers are secure, and all signers are trusted during the generation of the threshold ring signatures. However, our anonymity model still captures the case that a signer loses his secret key to the adversary before or after the generation of σ^*. We just do not allow the adversary to actively participate in the generation of the threshold ring signature.

4.3 Efficiency Analysis

When comparing our scheme with the 1-out-of-n SW ring signature scheme, the size of our signature is exactly the same as the SW scheme. In terms of computation cost, the overall signing process only increases by some elliptic curve addition operations. However, if it is measured as per signer computation, each signer actually requires less, when compared to the SW scheme. The verification algorithm only requires 1 more exponentiation to the SW scheme.

5. CONCLUSION

In this paper, we presented an efficient construction of threshold ring signature without random oracles. Our scheme is a non-trivial extension of the Shacham-Waters (SW) signature [26]. Interestingly, we obtained the same signature size as the Shacham-Waters signature, while only a tiny computation cost is added. We note that our technique has been specifically customized to achieve a threshold ring signature in the standard model.

6. REFERENCES

[1] M. Abe, M. Ohkubo, and K. Suzuki. 1-out-of-n signatures from a variety of keys. In *ASIACRYPT 2002*, volume 2501 of *Lecture Notes in Computer Science*, pages 415–432. Springer, 2002.

[2] M. Bellare, A. Boldyreva, and A. Palacio. An uninstantiable random-oracle-model scheme for a hybrid-encryption problem. In *EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 171–188. Springer, 2004.

[3] M. Bellare, D. Micciancio, and B. Warinschi. Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In *EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 614–629. Springer, 2003.

[4] M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *CCS '93*, pages 62–73. ACM Press, 1993.

[5] A. Bender, J. Katz, and R. Morselli. Ring signatures: Stronger definitions, and constructions without random oracles. In *TCC*, volume 3876 of *Lecture Notes in Computer Science*, pages 60–79. Springer, 2006.

[6] D. Boneh, C. Gentry, B. Lynn, and H. Shacham. Aggregate and verifiably encrypted signatures from bilinear maps. In *EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 416–432. Springer, 2003.

[7] D. Boneh, E.-J. Goh, and K. Nissim. Evaluating 2-dnf formulas on ciphertexts. In *TCC*, volume 3378 of *Lecture Notes in Computer Science*, pages 325–341. Springer, 2005.

[8] E. Bresson, J. Stern, and M. Szydlo. Threshold ring signatures and applications to ad-hoc groups. In M. Yung, editor, *CRYPTO 2002*, volume 2442 of *Lecture Notes in Computer Science*, pages 465–480. Springer, 2002.

[9] J. Camenisch and M. Stadler. Efficient group signature schemes for large groups (extended abstract). In *CRYPTO 97*, volume 1294 of *Lecture Notes in Computer Science*, pages 410–424. Springer, 1997.

[10] R. Canetti, O. Goldreich, and S. Halevi. The Random Oracle Methodology, Revisited. In *STOC*, pages 209–218, 1998.

[11] P.-L. Cayrel, R. Lindner, M. Rückert, and R. Silva. A lattice-based threshold ring signature scheme. In *LATINCRYPT*, volume 6212 of *Lecture Notes in Computer Science*, pages 255–272. Springer, 2010.

[12] S. Chang, D. S. Wong, Y. Mu, and Z. Zhang. Certificateless threshold ring signature. *Inf. Sci.*, 179(20):3685–3696, 2009.

[13] D. Chaum and E. van Heyst. Group signatures. In *EUROCRYPT '91*, volume 547 of *Lecture Notes in Computer Science*, pages 257–265. Springer, 1991.

[14] S. S. M. Chow, L. C. Hui, and S. Yiu. Identity based threshold ring signature. In *ICISC 2004*, volume 3506 of *Lecture Notes in Computer Science*, pages 218–232. Springer, 2004. Also available at Cryptology ePrint Archive, Report 2004/179.

[15] S. S. M. Chow, J. K. Liu, V. K. Wei, and T. H. Yuen. Ring signatures without random oracles. In *ASIACCS 2006*, pages 297–302. ACM Press, 2006.

[16] S. S. M. Chow, S.-M. Yiu, and L. C. K. Hui. Efficient identity based ring signature. In *ACNS 2005*, volume 3531 of *Lecture Notes in Computer Science*, pages 499–512, 2005. Also available at Cryptology ePrint Archive, Report 2004/327.

[17] R. Cramer, I. Damgård, and B. Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In *CRYPTO 94*, volume 839 of *Lecture Notes in Computer Science*, pages 174–187. Springer, 1994.

[18] L. Dallot and D. Vergnaud. Provably secure code-based threshold ring signatures. In *IMA Int. Conf.*, volume 5921 of *Lecture Notes in Computer Science*, pages 222–235. Springer, 2009.

[19] Y. Dodis, A. Kiayias, A. Nicolosi, and V. Shoup. Anonymous identification in ad hoc groups. In *EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 609–626. Springer, 2004.

[20] J. Han, Q. Xu, and G. Chen. Efficient id-based threshold ring signature scheme. In *EUC (2)*, pages 437–442. IEEE Computer Society, 2008.

[21] F. Laguillaumie and D. Vergnaud. Multi-designated verifiers signatures. In *ICICS 2004*, volume 3269 of *Lecture Notes in Computer Science*, pages 495–507. Springer, 2004.

[22] J. K. Liu, V. K. Wei, and D. S. Wong. A separable threshold ring signature scheme. In *ICISC 2003*, volume 2971 of *Lecture Notes in Computer Science*, pages 352–369. Springer, 2003.

[23] C. A. Melchor, P.-L. Cayrel, and P. Gaborit. A new efficient threshold ring signature scheme based on coding theory. In *PQCrypto*, volume 5299 of *Lecture Notes in Computer Science*, pages 1–16. Springer, 2008.

[24] R. L. Rivest, A. Shamir, and Y. Tauman. How to leak a secret. In *ASIACRYPT 2001*, volume 2248 of *Lecture Notes in Computer Science*, pages 552–565. Springer, 2001.

[25] S. Schäge and J. Schwenk. A cdh-based ring signature scheme with short signatures and public keys. In *FC 2010*, volume 6052 of *Lecture Notes in Computer Science*, pages 129–142. Springer, 2010.

[26] H. Shacham and B. Waters. Efficient ring signatures without random oracles. In *PKC 2007*, volume 4450 of *Lecture Notes in Computer Science*, pages 166–180. Springer, 2007.

[27] W. Susilo and Y. Mu. Non-interactive deniable ring authentication. In *ICISC 2003*, volume 2971 of *Lecture Notes in Computer Science*, pages 386–401. Springer, 2004.

[28] W. Susilo, Y. Mu, and F. Zhang. Perfect concurrent signature schemes. In *ICICS 2004*, volume 3269 of *Lecture Notes in Computer Science*, pages 14–26. Springer, 2004.

[29] 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. In *ProvSec 2010*, volume 6402 of *Lecture Notes in Computer Science*, pages 166–183. Springer, 2010.

[30] D. S. Wong, K. Fung, J. K. Liu, and V. K. Wei. On the rs-code construction of ring signature schemes and a threshold setting of rst. In *ICICS 2003*, volume 2836 of *Lecture Notes in Computer Science*, pages 34–46. Springer, 2003.

[31] J. Xu, Z. Zhang, and D. Feng. A ring signature scheme using bilinear pairings. In *WISA 2004*, volume 3325 of *Lecture Notes in Computer Science*, pages 163–172. Springer, 2004.

[32] F. Zhang and K. Kim. Id-based blind signature and ring signature from pairings. In *ASIACRYPT 2002*, volume 2501 of *Lecture Notes in Computer Science*, pages 533–547. Springer, 2002.