New Convertible Ring Signatures Based on RSA

Kaibin Huang* and Raylin Tso**

[1]*Department of Computer Science, National Chengchi University, Taipei 11605, Taiwan*
*100753504@nccu.edu.tw
**raylin@cs.nccu.edu.tw

Abstract

Convertible ring signatures with gradual revelation of non-signers (GR-CRS) is first introduced by Tso in 2010. With this scheme, a genuine signer can convert a ring signature into a traditional single signer digital signature. However, a genuine signer is also able to reveal part of non-signers while still preserving the unconditional signer anonymity. The GR-CRS scheme is very useful in many applications when some members in the ring signature are not trusted by some verifiers. However, we notice that Tso's scheme is based on the discrete dogarithm assumption, and therefore, cannot work with schemes based on RSA. As we know, RSA is by far the most widely used public key cryptosystem, so it is natural to consider whether we could produce a GR-CRS scheme based on RSA. In this study, we address this problem by proposing a new GR-CRS scheme based on RSA. The security of the new scheme is rigorously proved using the random oracle model based on the hardness of the RSA assumption and the intractability of inverting cryptographic one-way hash functions.
Keywords : convertible ring signature, provable security, random oracle, RSA, reveal non-signers

1 Introduction

A digital signature [7] is a mechanism for demonstrating the authenticity of a digital message or document. A valid digital signature allows a recipient to be confident that a message is not altered in transition and it is created by the known signer. There are a lot of useful applications and implements based on digital signature like [8] and [17]. However, the identity of signers on the signature causes some problem in some specific conditions, like the signature on a guilty judgment may make the judge at risk. In this case, we need some mechanism like group-oriented signatures [5][13], which can prevent the disclosure of a signer's identity.

Group-oriented signature is introduced to provide signer anonymity in real applications. In these schemes, a signer can arbitrarily choose several members to form a group (or known as a ring) and generate a group or ring signature

without the assistance of other members. After verifying the resultant signature, any verifier will be convinced by the signature that a message was generated and signed by a member of the group, but they cannot distinguish the genuine signer. Thus, the signer's anonymity can be protected. This kind of signature scheme is very useful in applications when being applied to leak a secret.

From a formal perspective, group-oriented signatures belong to two major types: group signatures [4][6][12] and ring signatures [2][5][9] [11][13]. The major difference is that there is a trusted group manager who can convert the group signature into a conventional signature in group signatures when necessary; whereas, in ring signatures, there is no recall mechanism for disclosing the signer's identity. The current study is focused on the design of ring signatures. Ring signature schemes have been proved useful in secure communications [16] and group-oriented communication systems [15].

Related work: The concept of the ring signature was first formalized by Rivest et al. [13] in 2001. The original scheme is not convertible; thus signer's anonymity is maintained. Lee et al. [11] proposed the first convertible ring signature scheme in 2005, where a genuine signer could convert the ring signature into a conventional signature if it was necessary to verify the origin signer of a signature. In 2006, Gao et al. [9] note that Lee et al.'s scheme only provided informal security analysis rather than a rigorous security proof. In addition, the security model for the convertibility of ring signatures is also informal. Gao et al. [9] formalized the security model for the convertibility of ring signatures and also proposed a new ring signature scheme known as a controllable ring signature. Compared with the original ring signature, the controllable ring signature provides new properties such as anonymous identification, linkability and convertibility. However, this new method does not support the gradual revelation of non-signers. A ring signature with gradual revelation of non-signers would allow a genuine signer to gradually reveal the identity of non-signers before converting a ring signature into a conventional signature. In other words, if there are n members in a ring signature with only one genuine signer (i.e., the other $n-1$ members of the ring are non-signers), revealing the identity of one non-signer would mean that the ring signature will become an $(n-1)$-participants ring signature. The ring signature is converted into a conventional signature after $n-1$ identities of non-signers are all revoked; thus, anyone can know the identity of the genuine signer.

Motivation: The property of gradual non-signers revelation would be useful when some non-signers of the ring signature are not trusted by some verifiers. Ring signatures are typically used in applications of anonymously leaking secrets, so the leaked secret may also not trusted by some verifiers if someone in the signing group is not trusted. In other words, even if the signature has passed the verification process, the signature may be rejected by some verifiers if some members of the group are not trusted. Rivest et al. [13] first note this problem and provided a solution in their modified ring signature scheme. However, signer anonymity can only be guaranteed computationally difficult in their modified scheme. Tso [14] notes this problem and proposes a convertible ring signature with gradual revelation of non-signers (GR-CRS). Similar to Rivest et al.'s scheme, Tso's scheme also allowed the gradual revelation of non-signers

and conversion of the ring signature into a conventional signature when all the non-signers are revoked. But, unlike Rivest et al.'s scheme, the signer anonymity was still preserved unconditionally in Tso's scheme. Thus, attackers cannot find the genuine signer of a ring signature, even if they have unlimited computational power. However, we noted that Tso's scheme is based on the discrete logarithm assumption, which means that RSA users with RSA public keys cannot gradually reveal non-signers while preserving the unconditional signer anonymity. RSA is by far the most widely used public key cryptosystem, so it is natural to consider whether we could produce a GR-CRS scheme in the RSA setting.

Our contribution: In this paper, we consult in Tso's scheme [14] and Abe et al.'s ring signature [1], and propose a new GR-CRS scheme. An important feature is that our new scheme is based on RSA. As mentioned above, RSA is by far the most widely used public key cryptosystem; therefore it is important that the properties of GR-CRS provide additional functions for RSA users. In addition, our scheme is a modification of Abe et al.'s scheme, and therefore our scheme also inherits the advantages of Abe et al.'s scheme. Thus, our scheme allows the separability of public keys. This property means users can employee key pairs that may be generated by different Key generation centers (KGCs) within a single ring signature and the domains of keys need not be identical. The security of the new scheme was rigorously demonstrated using a random oracle model based on the hardness of the RSA assumption and the intractability of inverting cryptographic one-way hash functions.

Paper organization: The rest of this paper is organized as follows. Preliminaries will be introduced in section 2. The GR-CRS framework and security requirements are specified in sections 3. In section 4, we propose a new RSA-based convertible ring signature with gradual revelation of non-signers. The system accuracy and its security proof are described in section 5. Finally, we summarize the findings of our study and provide some comparisons with other related works in section 6.

2 Preliminaries

In the section, we review some preliminaries which are required in our scheme.

2.1 RSA

The RSA signature scheme [7] consists of three algorithms: *Key generation*, *Sign* and *Verify*.

- *Key generation*:

  1. Choose two different large primes p and q, compute $N = pq$.
  2. Compute $\varphi(N) = \gcd(p-1)(q-1)$, where φ means the Euler function.
  3. Select an integer $e < \varphi(N)$ and $\gcd(e, \varphi(N)) = 1$.
  4. Compute d, where $d \times e \equiv 1 \bmod \varphi(N)$.

  (N, e, H) is published as the public key, d is the secret key, which is kept secretly. $H : \{0,1\}^* \to \mathbb{Z}_N$ denotes a one-way hash function.

- *Sign*: to sign a message m, a signer with secret key d computes s as follows:

$$s \equiv H(m)^d \bmod N$$

- *Verify*: anyone accepts the signature s on the message m if and only if

$$H(m) \equiv s^e \bmod N$$

2.2 Strong RSA Assumption

As the definition in [3], for all probabilistic polynomial time algorithms $\mathcal{A}$, some $c > 0$, and a sufficient large number k,

$$Pr[y^e \equiv x \bmod n; e \in prime; e < n; n \in_R RSA_{\bmod k};$$

$$x \in_R \mathbb{Z}_n \mapsto (y, e) \leftarrow \mathcal{A}(n, x)] \leq k^{-c}$$

Thus the adversary $\mathcal{A}$ is given n and x as in the usual RSA assumption, but he may choose the exponent e for which he extracts the root. We are neither aware of any corroboration that it should be hard, nor can we break it. Four obvious attacks do not work, i.e., they are equivalent to breaking some other problem believed to be hard.

2.3 Discrete Logarithm Problem (DL problem)

Discrete logarithm problem is regarded as difficult in general. Let p and q be two large prime numbers, $q|(p-1)$, $g \in \mathbb{Z}_p^*$, $order(g) = q$; and $\mathbb{G}$ be a subgroup of $\mathbb{Z}_p^*$, h is random selected in $\mathbb{G}$.

DL problem: given $g, h \in \mathbb{G}$ and p, find $x \in \mathbb{Z}_q$ that $g^x \equiv h \bmod p$.

2.4 Abe et al.'s Ring Signature Scheme (in all RSA case)

Abe's scheme [1] is used as a building block to construct a new ring signature in our scheme. In this section, we briefly introduce Abe's scheme as a preliminary. For $i = 1$ to n, let $L = \{(e_1, N_1), \ldots, (e_n, N_n)\}$ be RSA public keys and $H_i : \{0,1\}^* \to \mathbb{Z}_{N_i}$ be hash functions. A signer who has the private key d_k generates a signature for message m as follows:

- *Signature generation*:
  1. *Initialization*: select $r_k \in_R \mathbb{Z}_{N_k}$ and compute $c_{k+1} = H_{k+1}(L, m, r_k)$.
  2. *Forward sequence*: for $i = k+1, \ldots, n, 1, \ldots, k-1$, select $s_i \in_R \mathbb{Z}_{N_i}$, and compute $c_{i+1} = H_{i+1}(L, m, c_i + s_i^{e_i} \bmod N_i)$.
  3. *Shaping into a ring*: compute $s_k = (r_k - c_k)^{d_k} \bmod N_k$.

  The resulting signature for m and L is $\sigma = \{c_1, s_1, \ldots, s_n\}$.

- *Signature verification*: for $i = 1, \ldots, n$,

$$r_i = c_i + s_i^{e_i} \bmod N_i; \text{ if } i \neq n,\ c_{i+1} = H_{i+1}(L, m, r_i)$$

  Accept if $c_1 = H_1(L, m, r_n)$; reject, otherwise.

3 Model and security requirements of GR-CRS

In this section, we first define the model of GR-CRS [14]. Before that, let U_i be all members of a ring signature, $1 \leq i \leq n$; U_j denotes all non-signers, $1 \leq j \leq n$, $j \neq k$; and U_k denotes the genuine signer.

3.1 Model of GR-CRS

Following the definitions in [14], the GR-CRS scheme includes *SetUp*, *RSigGen*, *RSigVerify*, *Reveal*, *RevealVer*, *Convert* and *CSVerify*, which are defined as follows:

- $(pk_i, sk_i) \leftarrow SetUp(1^{\lambda_i})$: *SetUp* is a polynomial time algorithm that takes a secure parameter λ_i as input and the output is the public/private key pair of a user. Let $L = \{pk_1, \ldots, pk_n\}$ stand for the set composing of all public keys.
- $(\sigma, z'_1, \ldots, z'_{k-1}, z'_{k+1}, \ldots, z'_n) \leftarrow RSigGen(m, L, sk_k)$: *RSigGen* is a polynomial time algorithm that takes a message m, the public key set L and the secret key sk_k of signer U_k as inputs, then outputs a signature σ on message m, and a set of secret messages $(z'_1, \ldots, z'_{k-1}, z'_{k+1}, \ldots, z'_n)$.
- $1/0 \leftarrow RSigVerify(m, L, \sigma)$: *RSigVerify* is a polynomial time algorithm that takes a message m, the public key set L and a GR-CRS signature σ as inputs, then outputs 1 for accepting the signature; or 0 for rejecting.
- $z'_j \leftarrow Reveal(U_j)$: *Reveal* is a polynomial time algorithm that takes a member U_j as input, and then outputs his or her secret message z'_j for revealing.
- $1/0 \leftarrow RevealVer(z'_j)$: *RevealVer* is a polynomial time algorithm that takes a secret message z'_j as input, then outputs 1 if U_j is **not** the genuine signer; otherwise, it returns 0, which means nothing.
- *Convert*: After the genuine signer U_k outputs a GR-CRS signature σ and reveals $n-1$ non-signers by publishing secrets $(z'_{k+1}, \ldots, z'_n, z'_1, \ldots, z'_{k-1})$, the GR-CRS signature σ is converted into a traditional single-signer digital signature.
- $1/0 \leftarrow CSVerify(m, L, \sigma')$: *CSVerify* is a polynomial time algorithm that takes message m, the public key set L, a signature σ and $n-1$ verified secret message $(z'_1, \ldots, z'_{k-1}, z'_{k+1}, \ldots, z'_n)$, then outputs 1 when the signature is accepted; otherwise, it returns 0, which is regarded as rejected.

3.2 Security requirements of GR-CRS

The security of GR-CRS comes from four aspects: *signer ambiguity*, *unforgeability of the ring signature*, *un-revealability of non-signers*, and *unforgeability of the converted signature*.

- *Signer ambiguity*: assume that there are n members $U_1, \ldots, U_n$ in the GR-CRS scheme. The probability of guessing the genuine signer U_k should be $1/n$. Furthermore, after revealing some several members $U_{revealed} \subseteq U_j$, it remains *signer ambiguity*, the probability of guessing the genuine signer U_k should be $1/(n - |U_{revealed}|)$.

- *Unforgeability of the ring signature*: let $\mathcal{SO}(L, m)$ be a signing oracle that takes public key set L and message m as inputs, then outputs a valid signature σ that can be verified by $RSigVerify(m, L, \sigma)$. Assume that an attacker $\mathcal{A}$ may use signing oracle $\sigma \leftarrow \mathcal{A}^{\mathcal{SO}}(m, L)$ to get several valid GR-CRS signatures $\bar{\sigma} = \{(\sigma_1, m_1, L), (\sigma_2, m_2, L), \ldots, (\sigma_x, m_x, L)\}$, which pass the *RSigVerify* verification. We say that a GR-CRS signature is unforgeable against adaptive chosen message attacks if $\mathcal{A}$ is not able to generate a new signature $\sigma' \notin \bar{\sigma}$ related to (m', L) even above assumption holds.

- *Un-revealability of non-signers*: to reveal non-signers, the genuine signer has to publish z'_j corresponding to user U_j which passes the *RevealVer* verification. The *un-revealability* against non-signers implies that only the genuine signer is able to reveal non-signers by publishing secrete messages z'_js in GR-CRS scheme. Only the genuine signer owns those secret information z'_js, and they should be difficult computed by those non-signers.

- *Unforgeability of the converted signature*: assume that there is a signing oracle $\mathcal{SO}(L, m)$ which takes public key set L and message m as inputs, then outputs a converted signature σ that can be verified by *CSigVerify*(m, L, σ). With the aid of $\mathcal{SO}$, any adversary $\mathcal{A}$ can request some converted signatures $\bar{\sigma} = \{(\sigma_1, m_1, L), (\sigma_2, m_2, L), \ldots, (\sigma_x, m_x, L)\}$. We say that a converted GR-CRS signature is unforgeable against adaptive chosen message attacks if $\mathcal{A}$ is not able to generate a new converted signature $\sigma' \notin \bar{\sigma}$ related to (m', L) even above assumption holds.

4 New CR-GRS Scheme Based on RSA

In this section, we propose our new RSA-based convertible ring signature scheme with the gradual revelation of non-signers. Similar to Tso's scheme [14], our scheme provides unconditional *signer anonymity*. If necessary, a genuine signer can also reveal non-signers one-by-one. The new scheme adapts RSA as its core algorithm, which is different from Tso's scheme.

- *SetUp*: each member U_i in the ring signature owns a pair of RSA-based public keys $pk_i = (N_i, e_i, H_i)$ and its corresponding secret key $sk_i = d_i$. $N_i = p_i q_i$, where p_i and q_i are two large prime numbers with the same bits. $\lambda_i = \varphi(N_i) = \gcd(p_i - 1, q_i - 1)$; $\forall n_i \in N_i : n_i^{\lambda_i} \equiv 1 \bmod N_i$. $H_i : \{0,1\}^* \rightarrow \mathbb{Z}^*_{N_i}$ denotes one-way hash functions related to user U_i. $L = \{pk_1, \ldots, pk_n\}$ stands for the public key set.

- *RSigGen*: the signer U_k generates a convertible ring signature following:

  1. Generate a set of public parameters (p, q, g, H_p, H_q), which p and q are two large primes such that $q|(p-1)$; g is a generator of group $\mathbb{Z}^*_p$, whose order is q. $H_p : \{0,1\}^* \rightarrow \mathbb{Z}^*_p$ and $H_q : \{0,1\}^* \rightarrow \mathbb{Z}^*_q$ are secure one-way hash functions.
  2. For each non-signer U_j, $1 \leq j \leq n$, $j \neq k$, U_k generates random numbers $z'_j \in_R \mathbb{Z}^*_p$, $x_j \in_R \mathbb{Z}^*_{N_j}$, $y_j \in_R \mathbb{Z}^*_q$, and computes $z_j = H_p(z'_j)$.

3. Pick $\alpha \in_R \mathbb{Z}_{N_k}^*$, $\beta \in_R \mathbb{Z}_q^*$ and $\gamma \in_R \mathbb{Z}_q^*$, then computes

$$z_k \equiv g^\beta - H_p(e_k) \bmod p, \text{ and } c_{k+1} = H_{k+1}(L, m, \alpha, g^\gamma)$$

4. Forward sequence: for $j = k+1, \ldots, n, 1, \ldots, k-1$

$$\begin{aligned} a_j &\leftarrow c_j + x_j^{e_j} \bmod N_j \\ b_j &\leftarrow g^{y_j}(z_j + H_p(e_j))^{H_q(c_j)} \bmod p \\ c_{j+1} &\leftarrow H_{j+1}(L, m, a_j, b_j) \end{aligned}$$

Closing the ring signature:

$$x_k \equiv (\alpha - c_k)^{d_k} \bmod N_k, \text{ and } y_k \equiv \gamma - \beta H_q(c_k) \bmod q$$

5. The signer U_k publishes the signature $\sigma = (c_1, (x_1, y_1, z_1), \ldots, (x_n, y_n, z_n))$ corresponding to the message m and keeps $(z'_1, \ldots, z'_{k-1}, z_{k+1}, \ldots, z'_n)$ secret for revealing purpose.

- *RSigVerify*: anyone can verify the correctness of the signature σ. For $i = 1$ to n, compute

$$\begin{aligned} a_i &\equiv c_i + x_i^{e_i} \bmod N_i \\ b_i &\equiv g^{y_i}(z_i + H_p(e_i))^{H_q(c_i)} \bmod p \\ c_{i+1} &= H_{i+1}(L, m, a_i, b_i) \end{aligned}$$

After a round of verification, the ring signature σ is accepted if $c_1 = c_{n+1}$.

- *Reveal*: to revoke U_j, the signer U_k reveals z'_j.

- *RevealVer*: anyone can verify the validation by checking $z_j \stackrel{?}{=} H_p(z'_j)$. If the equation is tenable, then, user U_j will be revoked.

- *Convert*: after gradually revealing $n-1$ non-signers in the ring signature, the ring signature σ is converted to a single-signer digital signature.

- *CSVerify*: this step first verifies *RSigVer*, and then verifies for $i = 1$ to n, $i \neq k$, $1 \leftarrow$*RevealVer*(z'_j). If they are all achieved, then the ring signature σ passes the *CSVerify*.

Remark: the main difference between the genuine signer and other non-signer is determined by z_i.

$$\begin{cases} \text{For those non-signers } U_j,\ z_j = H_p(z'_j) \\ \text{For the genuine signer } U_k,\ z_k \equiv g^\beta - H_p(e_k) \bmod p \end{cases}$$

By the one-wayness of the hash function H_p, it is convinced that U_j is a non-signer when the someone reveals a hash seed z'_j that satisfies $z_j = H_p(z'_j)$.

5 Security proof

In this section, we first demonstrate its accuracy, and then infer the security of the new scheme, which include signer ambiguity, unforgeability of ring signatures, un-revealability of non-signers, and unforgeability of converted signatures.

5.1 Accuracy

$$\begin{cases}\text{In the step 3 of } \textit{RSigGen}\text{: } c_{k+1} \leftarrow H_{k+1}(L, m, \alpha, g^{\gamma}) \\ \text{In the } \textit{RSigVer} \text{ phase, when i=k: } c_{k+1} \leftarrow H_{k+1}(L, m, a_k, b_k)\end{cases}$$

Therefore, if $a_k = \alpha$ and $b_k = g^{\gamma}$, the correctness of the scheme is sound. We give an inference below:

$$\begin{aligned} a_k &\equiv c_k + s_k^{e_k} \equiv c_k + ((\alpha - c_k)^{d_k})^{e_k} \equiv c_k + \alpha - c_k \equiv \alpha \bmod N_k \\ b_k &\equiv g^{y_k}(z_k + H_p(e_k))^{c_k} \equiv g^{y_k}(g^{\beta} - H_p(e_k) + H_p(e_k))^{c_k} \\ &\equiv g^{y_k} \cdot (g^{\beta})^{c_k} \equiv g^{\gamma - \beta c_k} \cdot g^{\beta c_k} \equiv g^{\gamma} \bmod p \end{aligned}$$

5.2 Signer ambiguity

In our scheme, there are n ring members $U_1, \ldots, U_n$. Thus, the probability of guessing a genuine signer U_k in the ring is $1/n$. After revealing several members $U_{revealed} \subseteq U_j$, it still achieves signer ambiguity and the probability of guessing a genuine signer U_k in the ring is $1/(n - |U_{revealed}|)$.

Take advantages of those proof in [1] and [14], we can easily prove that the signature $\sigma = (c_1, (x_1, y_1, z_1), \ldots, (x_n, y_n, z_n))$ is unconditionally signer ambiguous. Therefore, we simply prove that revealing phase does not reveal any information about genuine signers, even forcing unconditional powerful attackers.

Assume $\mathcal{A}$ is a powerful attacker who can overcome the DL-problem and the one-wayness of any one-way hash function. For any given z_i, g, p and e_i, $\mathcal{A}$ can find a random number β such that $z_i \equiv g^{\beta} - H_p(e_i) \bmod p$. However, $\mathcal{A}$ can also find another random number z_i' such that $z_i = H_p(z_i')$. Paradoxically, β indicates that U_i is a genuine signer, whereas z_i' indicates that U_i is a non-signer. Consequently, even an unconditional powerful attacker cannot distinguish a genuine signer from those non-signers.

5.3 Unforgeability of ring signature

Our scheme ensures the unforgeability of the ring signature against an EF-ACMA attacker [10]. Here we reduce the security of our scheme to the security of Abe et al.'s 1-out-of-n signature scheme [1]. If there is a powerful adversary $\mathcal{A}$ who can forge a valid signature σ that passes the *RSigVer* algorithm, then a simulator $\mathcal{SIM}$ can take $\mathcal{A}$ as a building block to forge a signature corresponding to Abe et al.'s ring signature scheme [1] as follows:

1. Let $\{(N_1, e_1, H_{Abe_1}), \ldots, (N_n, e_n, H_{Abe_n})\}$ be all RSA based public keys of Abe et al.'s scheme which are generated by $\mathcal{SIM}$. Each public key (N_i, e_i, H_{Abe_i}) corresponds to a user U_i, $1 \leq i \leq n$. After generation, $\mathcal{SIM}$ publishes $L = \{(N_1, e_1), \ldots, (N_n, e_n)\}$ to $\mathcal{A}$, and treats hash functions $H_i : \{0,1\}^* \rightarrow \mathbb{Z}_{N_i}$, $1 \leq i \leq n$ as random oracles that are controlled by $\mathcal{SIM}$. Let $L' \subseteq L$ be a subgroup of L, $\mathcal{SIM}$ also simulates a signing oracle that outputs a valid signature for the proposed scheme with the input (L', m) from any signing query made by the attacker $\mathcal{A}$. To make a

correct response, $\mathcal{SIM}$ will query the corresponding hash oracles and the signing oracle of Abe et al.'s scheme.

2. $\mathcal{SIM}$ generates a set of public parameters (p, q, g) following the *SetUp* algorithm, and publishes them to $\mathcal{A}$. Hash functions H_p and H_q are treated as random oracles which are controlled by $\mathcal{SIM}$.

3. When $\mathcal{A}$ requests a hash query $H_i(L, m, a_i, b_i)$ to $\mathcal{SIM}$, $\mathcal{SIM}$ first queries $H_{Abe_i}(L, m, a_i)$ to the hash oracle of Abe et al's scheme, and then returns $H_i(L, m, a_i, b_i) \leftarrow H_{Abe_i}(L, m, a_i)$ to $\mathcal{A}$. Let $L' \subseteq L$, when $\mathcal{A}$ queries the signing oracle (L', m), $\mathcal{SIM}$ queries it to Abe's signing oracle, and get a signature $\sigma_{Abe} = (c_1, s_1 s_n)$. Next, $\mathcal{SIM}$ generates random numbers $y = (y_1, \ldots, y_n)$, $z = (z_1, \ldots, z_n)$. Finally, $\mathcal{SIM}$ combines σ_{Abe}, y and z, returns $\sigma = (c_1, (s_1, y_1, z_1), \ldots, (x_n, y_n, z_n)) \leftarrow \mathcal{SO}(L', m)$ to $\mathcal{A}$ as the signature.

4. If $\mathcal{A}$ finally generates a forged signature $\sigma' = (c'_1, (x'_1, y'_1, z'_1), \ldots, (x'_n, y'_n, z'_n))$ which passes the signature verification process *RSigVer*, then $\mathcal{SIM}$ can forge a valid signature $\sigma_{Abe} = (c_{Abe_1}, x_{Abe_1}, \ldots, x_{Abe_n})$ that will pass the verification process of Abe et al.'s signature scheme.

Abe et al. have proved that their scheme was secure against EF-ACMA attackers in a random oracle model, based on the hardness of a strong RSA assumption. Based on the same hardness assumption, our scheme is also secure against EF-ACMA attackers.

5.4 Un-revealability of non-signers

To reveal non-signers, signer U_k outputs z'_j that satisfies $1 \leftarrow RevealVer(z'_j)$ to reveal U_j. In our scheme, *RevealVer* is implemented using the formula $z_j = H_p(z'_j)$ where H_p denotes a one-way hash function. Those secret information z'_js are held only by the genuine signer and it is computationally difficult to compute z'_j from z_j because of the one-wayness of the hash function H_p. Thus, the un-revealability of non-signers depends on the computational difficulty of the ony-wayness of hash functions.

5.5 Unforgeability of converted ring signature

We have already proved the unforgeability of the original ring signature. Furthermore, we have also proved that the proposed scheme provides un-revealability against non-signers. Taken together, these two proofs demonstrate the unforgeability of the converted ring signature.

6 Conclusion

In this paper, we follow the Tso's idea in [14], and take Abe's ring signature [1] as a building block to proposed a new GR-CRS scheme. One important feature is that our new scheme is based on RSA which is by far the most widely used public key cryptosystem. Our new scheme adds new functionality for RSA users. The security of the new scheme is proved in the random oracle model based on the hardness of the RSA assumption and the intractability of inverting

Table 1: Performance comparison

	Sign	*RSigVerify*	Signature length (bits)
Abe et al. [1]	$(n+1)$ Exp	n Exp	$(n+1)\|N\|$
Our scheme	$3n$ Exp	$3n$ Exp	$(n+1)\|N\| + n\|p\| + n\|q\|$

Table 2: Feature comparison

	Convertible	Reveal non-signers	Unconditional anonymous
Abe et al. [1]	No	No	No
Controllable [9]	Yes	No	Yes
Convertible [11]	Yes	No	Yes
Our scheme	Yes	Yes	Yes

cryptographic one-way hash functions. The performance comparison and feature comparison with other schemes is provided in the following tables. How to improve its performance including the computation cost and communication cost may be considered as a future work.

References

[1] Masayuki Abe, Miyako Ohkubo, and Koutarou Suzuki. 1-out-of-n signatures from a variety of keys. In *ASIACRYPT*, pages 415–432, 2002.

[2] Amit K. Awasthi and Sunder Lal. Id-based ring signature and proxy ring signature schemes from bilinear pairings. *IACR Cryptology ePrint Archive*, 2004:184, 2004.

[3] Niko Bari and Birgit Pfitzmann. Collision-free accumulators and fail-stop signature schemes without trees. In *EUROCRYPT*, pages 480–494, 1997.

[4] Mihir Bellare, Daniele Micciancio, and Bogdan Warinschi. Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In *EUROCRYPT*, pages 614–629, 2003.

[5] Emmanuel Bresson, Jacques Stern, and Michael Szydlo. Threshold ring signatures and applications to ad-hoc groups. In *CRYPTO*, pages 465–480, 2002.

[6] Elodie Briefer, Thierry Aubin, Katia Lehongre, and Fanny Rybak. How to identify dear enemies: the group signature in the complex song of the skylark alauda arvensis. *Journal of Experimental Biology*, 211(3):317–326, 2008.

[7] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.

[8] Dengguo Feng, Jing Xu, and Wei-Dong Chen. Generic constructions for strong designated verifier signature. *JIPS*, 7(1):159–172, 2011.

[9] Wei Gao, Guilin Wang, Xueli Wang, and Dongqing Xie. Controllable ring signatures. In *WISA*, pages 1–14, 2006.

[10] Shafi Goldwasser, Silvio Micali, and Ronald L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM J. Comput.*, 17(2):281–308, 1988.

[11] Kuo-Chang Lee, Hsiang-An Wen, and Tzonelih Hwang. Convertible ring signature. In *Communications, IEE Proceedings-*, volume 152, pages 411–414. IET, 2005.

[12] Joseph K. Liu, Victor K. Wei, and Duncan S. Wong. Linkable spontaneous anonymous group signature for ad hoc groups (extended abstract). In *ACISP*, pages 325–335, 2004.

[13] Ronald L. Rivest, Adi Shamir, and Yael Tauman. How to leak a secret. In *ASIACRYPT*, pages 552–565, 2001.

[14] Raylin Tso. Convertible ring signatures with gradual revelation of non-signers. *Security and Communication Networks*, 5(3):279–286, 2012.

[15] Shiuh-Jeng Wang, Yuh-Ren Tsai, Chien-Chih Shen, and Pin-You Chen. Hierarchical key derivation scheme for group-oriented communication systems. *IJITCC*, 1(1):66–76, 2010.

[16] Bin Xie, Anup Kumar, David Zhao, Ranga Reddy, and Bing He. On secure communication in integrated heterogeneous wireless networks. *IJITCC*, 1(1):4–23, 2010.

[17] Mengsong Zou, Lansheng Han, Ming Liu, and Qiwen Liu. Virus detection method based on behavior resource tree. *JIPS*, 7(1):173–186, 2011.