The Sampling Twice Technique for the RSA-based Cryptosystems with Anonymity

Ryotaro Hayashi and Keisuke Tanaka

Dept. of Mathematical and Computing Sciences, Tokyo Institute of Technology,
2-12-1 Ookayama, Meguro-ku, Tokyo 152-8552, Japan
{hayashi9, keisuke}@is.titech.ac.jp

Summary. We say that an encryption scheme or a signature scheme provides anonymity when it is infeasible to determine which user generated a ciphertext or a signature. To construct the schemes with anonymity, it is necessary that the space of ciphertexts or signatures is common to each user. In this paper, we focus on the techniques which can be used to obtain this anonymity property, and propose a new technique for obtaining the anonymity property on RSA-based cryptosystem, which we call "sampling twice." It generates the uniform distribution over 2^k by sampling the two elements from $\mathbb{Z}_N$ where $|N| = k$. Then, by applying the sampling twice technique, we construct the schemes for encryption, undeniable and confirmer signature, and ring signature, which have some advantage to the previous schemes.

Keywords: RSA, anonymity, encryption, undeniable and confirmer signature, ring signature

1 Introduction

We say that an encryption scheme or a signature scheme provides anonymity when it is infeasible to determine which user generated a ciphertext or a signature. A simple observation that seems to be folklore is that standard RSA encryption, namely, a ciphertext is $x^e \bmod N$ where x is a plaintext and (N, e) is a public key, does not provide anonymity, even when all moduli in the system have the same length. Suppose an adversary knows that the ciphertext y is created under one of two keys (N_0, e_0) or (N_1, e_1), and suppose $N_0 \leq N_1$. If $y \geq N_0$ then the adversary bets it was created under (N_1, e_1), else the adversary bets it was created under (N_0, e_0). It is not hard to see that this attack has non-negligible advantage. To construct the schemes with anonymity, it is necessary that the space of ciphertexts is common to each user. We can say the same thing about RSA-based signature schemes.

Bellare, Boldyreva, Desai, and Pointcheval [1] proposed a new security requirement of the encryption schemes called "key-privacy" or "anonymity." It asks that the encryption provide (in addition to privacy of the data being encrypted) privacy of the key under which the encryption was performed. In [1],

they provided the key-privacy encryption scheme, RSA-RAEP, which is a variant of RSA-OAEP (Bellare and Rogaway [2], Fujisaki, Okamoto, Pointcheval, and Stern [3]), and made the space of ciphertexts common to each user by repeating the evaluation of the RSA-OAEP permutation $f(x, r)$ with plaintext x and random r, each time using different r until the value is in the safe range. For deriving a value in the safe range, the number of the repetition would be very large (the value of the security parameter). In fact, their algorithm can fail to give a desired output with some (small) probability.

The anonymous encryption scheme has various applications. For example, anonymous authenticated key exchange protocol such as SKEME (Krawczyk [4]), anonymous credential system (Camenisch and Lysyanskaya [5]), and auction protocols (Sako [6]).

Chaum and Antwerpen provided undeniable signature which cannot be verified without the signer's cooperation [7, 8]. The validity or invalidity of an undeniable signature can be ascertained by conducting a protocol with the signer, assuming the signer participates. Chaum provided confirmer signature [9] which is undeniable signature where signatures may also be verified by interacting with an entity called the confirmer who has been designated by the signer. Galbraith and Mao proposed a new security notion for undeniable and confirmer signature named "anonymity" in [10]. We say that an undeniable or confirmer signature scheme provides anonymity when it is infeasible to determine which user generated the message-signature pair. In [10], Galbraith and Mao provided the undeniable and confirmer signature scheme with anonymity. They made the space of signatures common to each user by applying a standard RSA permutation to the signature and expanding it to the common domain $[0, 2^k)$ where N is a public key for each user and $|N| = k$. This technique was proposed by Desmedt [11].

Rivest, Shamir, and Tauman [12] proposed the notion of ring signature, which allows a member of an ad hoc collection of users S to prove that a message is authenticated by a member of S without revealing which member actually produced the signature. Unlike group signature, ring signature has no group managers, no setup procedures, no revocation procedures, and no coordination. The signer does not need the knowledge, consent, or assistance of the other ring members to put them in the ring. All the signer needs is knowledge of their regular public keys. They also proposed the efficient schemes based on RSA and Rabin. In their RSA-based scheme, the trap-door RSA permutations of the various ring members will have ranges of different sizes. This makes it awkward to combine the individual signatures, so one should construct some trap-door one-way permutation which has a common range for each user. Intuitively, in the ring signature scheme, Rivest, Shamir, and Tauman solved this problem by encoding the message to an N_i-ary representation and applying a standard RSA permutation f to the low-order digits where N_i is a public key for each user. This technique is considered to be essentially the same as that by Desmedt. As mentioned in [12], for deriving a secure permutation g with a common range, the range of g would be 160 bits larger than that of f.

Hayashi, Okamoto, and Tanaka [13] recently proposed the RSA family of trap-door permutations with a common domain denoted by RSACD. They showed that the θ-partial one-wayness of RSACD is equivalent to the one-wayness of RSACD for $\theta > 0.5$, and that the one-wayness of RSACD is equivalent to the one-wayness of RSA which is the standard RSA family of trap-door permutations. They also proposed the applications of RSACD to the key-privacy encryption scheme and ring signature scheme. Their schemes have some advantages to the previous schemes.

1.1 Our Contribution

In this paper, we focus on the techniques which can be used to obtain this anonymity property.

From the previous results mentioned above, we can find three techniques, repeating, expanding, and using RSACD, for anonymity of cryptosystems based on RSA.

Repeating Repeating the evaluation of the encryption (respectively the signing) with plaintext x (resp. message m), random r, and the RSA function, each time using different r until the value is smaller than any public key N of each user.
In [1], Bellare, Boldyreva, Desai, and Pointcheval used this technique for the encryption scheme.

Expanding Doing the evaluation of the encryption (respectively the signing) with plaintext x (resp. message m), random r, and the RSA function, and expanding it to the common domain.
This technique was proposed by Desmedt [11]. In [10], Galbraith and Mao used this technique for the undeniable signature scheme. In [12], Rivest, Shamir, and Tauman also used this technique for the ring signature.

RSACD Doing the evaluation of the encryption (respectively the signing) with plaintext x (resp. message m), random r, and the RSACD function. This function was proposed by Hayashi, Okamoto, and Tanaka [13].

In this paper, we propose a new technique for obtaining the anonymity property of RSA-based cryptosystems. We call this technique "sampling twice." In our technique, we employ an algorithm `ChooseAndShift`. It takes two numbers $x_1, x_2 \in \mathbb{Z}_N$ as input and returns a value $y \in [0, 2^k)$ where $|N| = k$, and if x_1 and x_2 are independently and uniformly chosen from $\mathbb{Z}_N$ then y is uniformly distributed over $[0, 2^k)$.

Sampling Twice Doing the evaluation of the encryption (respectively the signing) twice with plaintext x (resp. message m), random r_1 and r_2, and the RSA function, and applying our proposed algorithm `ChooseAndShift` for the two resulting values.

Then, by applying the sampling twice technique, we construct the schemes for encryption, undeniable and confirmer signature, and ring signature (See Figure 1.).

	Sampling Twice	Repeating	Expanding	RSACD
Encryption	**this paper**	Bellare et al.	-	Hayashi et al.
Undeniable and Confirmer Signature	**this paper**	-	Galbraith et al.	-
Ring Signature	**this paper**	-	Rivest et al.	Hayashi et al.

Fig. 1. The previous and our proposed schemes

We summarize the (dis)advantage of our proposed schemes.

Our proposed encryption scheme with sampling twice is efficient with respect to the size of ciphertexts and the decryption cost. It is also efficient with respect to the encryption cost in the worst case. On the other hand, that in the average case is larger than that of the previous schemes. More precisely, in our encryption scheme, the number of modular exponentiation to encrypt in the worst case is 2, while those in the previous schemes are 1 or 1.5.

Our proposed undeniable and confirmer signature scheme with sampling twice is efficient with respect to the size of signatures. On the other hand, the number of exponentiations for signing and that of computation of square roots is always 2, while those of the other schemes are 1 or 1.5 in the average case.

Our proposed ring signature scheme with sampling twice is efficient with respect to the size of signatures and the verification cost. On the other hand, the signing cost of our scheme is larger than those of the previous schemes.

If we use the RSACD function, the resulting value is calculated by applying the RSA function either once or twice. Fortunately, since applying the RSA function twice does not reduce security, we can prove that the RSACD function is one-way if the RSA function is one-way. Generally speaking, a one-way function does not always have this property, and we cannot construct a one-way functions with a common domain.

On the other hand, in the sampling twice, repeating, and expanding techniques, the resulting value is calculated by applying the RSA function once. Therefore, it might be possible to apply these techniques to other one-way functions and prove the security of the resulting schemes.

The organization of this paper is as follows. In Section 2, we review the definitions of families of functions and the standard RSA family. In Section 3, we construct the algorithm `ChooseAndShift` and propose the sampling twice technique. We propose the encryption schemes with anonymity in Section 4, the undeniable and confirmer signature schemes with anonymity in Section 5, and the ring signature schemes with anonymity in Section 6. We conclude in Section 7.

2 Preliminaries

We describe the definitions of families of functions, families of trap-door permutations, and θ-partial one-way.

Definition 1 (families of functions [1]). *A family of functions $F = (K, S, E)$ consists of three algorithms. The randomized key-generation algorithm K takes as input a security parameter $k \in \mathbb{N}$ and returns a pair (pk, sk) where pk is a public key and sk is an associated secret key. (In cases where the family is not trap-door, the secret key is simply the empty string.) The randomized sampling algorithm S takes input pk and returns a random point in a set that we call the domain of pk and denote by $\mathrm{Dom}_F(pk)$. The deterministic evaluation algorithm E takes input pk and a point $x \in \mathrm{Dom}_F(pk)$ and returns an output we denote by $E_{pk}(x)$. We let $\mathrm{Rng}_F(pk) = \{E_{pk}(x) \mid x \in \mathrm{Dom}_F(pk)\}$ denote the range of the function $E_{pk}(\cdot)$.*

Definition 2 (families of trap-door permutations [1]). *We say that F is a family of trap-door functions if there exists a deterministic inversion algorithm I that takes input sk and a point $y \in \mathrm{Rng}_F(pk)$ and returns a point $x \in \mathrm{Dom}_F(pk)$ such that $E_{pk}(x) = y$. We say that F is a family of trap-door permutations if F is a family of trap-door functions, $\mathrm{Dom}_F(pk) = \mathrm{Rng}_F(pk)$, and E_{pk} is a bijection on this set.*

Definition 3 (θ-partial one-way [1]). *Let $F = (K, S, E)$ be a family of functions. Let $b \in \{0, 1\}$ and $k \in \mathbb{N}$ be a security parameter. Let $0 < \theta \leq 1$ be a constant. Let A be an adversary. Now, we consider the following experiments:*

$$
\begin{array}{l}
\texttt{Experiment } \mathbf{Exp}^{\theta\text{-pow-fnc}}_{F,A}(k) \\
\quad (pk, sk) \leftarrow K(k) \\
\quad x \xleftarrow{R} \mathrm{Dom}_F(pk) \\
\quad y \leftarrow E_{pk}(x) \\
\quad x_1 \leftarrow A(pk, y) \texttt{ where } |x_1| = \lceil \theta \cdot |x| \rceil \\
\quad \texttt{if } \big(E_{pk}(x_1 || x_2) = y \texttt{ for some } x_2\big) \texttt{ return } 1 \texttt{ else return } 0
\end{array}
$$

Here " $||$ " denotes concatenation and " $x \xleftarrow{R} \mathrm{Dom}_F(pk)$ " is the operation of picking an element x uniformly from $\mathrm{Dom}_F(pk)$. We define the advantages of the adversary via

$$\mathbf{Adv}^{\theta\text{-pow-fnc}}_{F,A}(k) = \Pr[\mathbf{Exp}^{\theta\text{-pow-fnc}}_{F,A}(k) = 1]$$

where the probability is taken over K, $x \xleftarrow{R} \mathrm{Dom}_F(pk)$, E, and A. We say that the family F is θ-partial one-way if the function $\mathbf{Adv}^{\theta\text{-pow-fnc}}_{F,A}(\cdot)$ is negligible for any adversary A whose time complexity is polynomial in k.

The "time-complexity" is the worst case execution time of the experiment plus the size of the code of the adversary, in some fixed RAM model of computation.

Note that when $\theta = 1$ the notion of θ-partial one-wayness coincides with the standard notion of one-wayness. We say that the family F is one-way when F is 1-partial one-way.

We describe the standard RSA family of trap-door permutations denoted by RSA.

Definition 4 (the standard RSA family of trap-door permutations). *The specifications of the standard RSA family of trap-door permutations* $\mathsf{RSA} = (K, S, E)$ *are as follows. The key generation algorithm takes as input a security parameter* k *and picks random, distinct primes* p, q *in the range* $2^{\lceil k/2 \rceil - 1} < p, q < 2^{\lceil k/2 \rceil}$ *and* $2^{k-1} < pq < 2^k$. *It sets* $N = pq$. *It picks* $e, d \in \mathbb{Z}^*_{\phi(N)}$ *such that* $ed = 1 \pmod{\phi(N)}$ *where* $\phi(N) = (p-1)(q-1)$. *The public key is* N, e, k *and the secret key is* N, d, k. *The sets* $\mathrm{Dom}_{\mathsf{RSA}}(N, e, k)$ *and* $\mathrm{Rng}_{\mathsf{RSA}}(N, e, k)$ *are both equal to* $\mathbb{Z}^*_N$. *The evaluation algorithm* $E_{N,e,k}(x) = x^e \bmod N$ *and the inversion algorithm* $I_{N,d,k}(y) = y^d \bmod N$. *The sampling algorithm returns a random point in* $\mathbb{Z}^*_N$.

Fujisaki, Okamoto, Pointcheval, and Stern [3] showed that the θ-partial one-wayness of RSA is equivalent to the one-wayness of RSA for $\theta > 0.5$.

3 The Sampling Twice Technique

In this section, we propose a new technique for obtaining the anonymity property of RSA-based cryptosystems. We call this technique "sampling twice." In our technique, we employ the following algorithm `ChooseAndShift`. It takes two numbers $x_1, x_2 \in \mathbb{Z}_N$ as input and returns a value $y \in [0, 2^k)$ where $|N| = k$.

Algorithm $\mathtt{ChooseAndShift}_{N,k}(x_1, x_2)$
  if $(0 \le x_1, x_2 < 2^k - N)$
    return $\begin{cases} x_1 & \text{with probability } \frac{1}{2} \\ x_1 + N & \text{with probability } \frac{1}{2} \end{cases}$
  elseif $(2^k - N \le x_1, x_2 < N)$
    return x_1
  else
    $y_1 \leftarrow \min\{x_1, x_2\}$; $y_2 \leftarrow \max\{x_1, x_2\}$
    %%% *Note that* $0 \le y_1 < 2^k - N$ *and* $2^k - N \le y_2 < N$. %%%
    return $\begin{cases} y_1 & \text{with probability } (\frac{1}{2} + \frac{N}{2^{k+1}}) \times \frac{1}{2} \\ y_1 + N & \text{with probability } (\frac{1}{2} + \frac{N}{2^{k+1}}) \times \frac{1}{2} \\ y_2 & \text{with probability } \frac{1}{2} - \frac{N}{2^{k+1}} \end{cases}$

Note that $2^{k-1} < N < 2^k$ ensures $2^k - N < N$, $0 < \frac{1}{2} - \frac{N}{2^{k+1}} < 1$, and $0 < \frac{1}{2} + \frac{N}{2^{k+1}} < 1$. In order to run this algorithm, it is sufficient to prepare only $k + 3$ random bits.

We prove the following theorem on the property of `ChooseAndShift`.

Theorem 1. *If* x_1 *and* x_2 *are independently and uniformly chosen from* $\mathbb{Z}_N$ *then the output of the above algorithm is uniformly distributed over* $[0, 2^k)$.

Proof. To prove this theorem, we show that if x_1 and x_2 are independently and uniformly chosen from $\mathbb{Z}_N$ then $\Pr[\mathtt{ChooseAndShift}(x_1, x_2) = z] = 1/2^k$ for any

$z \in [0, 2^k)$. For any $z \in [0, 2^k - N)$, we have

$$\begin{aligned}
&\Pr[\texttt{ChooseAndShift}(x_1, x_2) = z] \\
&\quad = \Pr[x_1 = z \ \wedge \ 0 \le x_2 < 2^k - N] \times \tfrac{1}{2} \\
&\qquad + \Pr[(x_1 = z \ \wedge \ 2^k - N \le x_2 < N) \vee (x_2 = z \ \wedge \ 2^k - N \le x_1 < N)] \\
&\qquad\quad \times (\tfrac{1}{2} + \tfrac{N}{2^{k+1}}) \times \tfrac{1}{2} \\
&\quad = \tfrac{1}{N} \times \tfrac{2^k - N}{N} \times \tfrac{1}{2} + (\tfrac{1}{N} \times \tfrac{2N - 2^k}{N}) \times 2 \times (\tfrac{1}{2} + \tfrac{N}{2^{k+1}}) \times \tfrac{1}{2} = \tfrac{1}{2^k}.
\end{aligned}$$

It is clear that $\Pr[\texttt{ChooseAndShift}(x_1, x_2) = z'] = \Pr[\texttt{ChooseAndShift}(x_1, x_2) = z' + N]$ for any $z' \in [0, 2^k - N)$. Therefore, for any $z \in [N, 2^k)$, we have $\Pr[\texttt{ChooseAndShift}(x_1, x_2) = z] = 1/2^k$.

Furthermore, for any $z \in [2^k - N, N)$, we have

$$\begin{aligned}
&\Pr[\texttt{ChooseAndShift}(x_1, x_2) = z] \\
&\quad = \Pr[x_1 = z \ \wedge \ 2^k - N \le x_2 < N] \\
&\qquad + \Pr[(x_1 = z \ \wedge \ 0 \le x_2 < 2^k - N) \vee (x_2 = z \ \wedge \ 0 \le x_1 < 2^k - N)] \\
&\qquad\quad \times (\tfrac{1}{2} - \tfrac{N}{2^{k+1}}) \\
&\quad = \tfrac{1}{N} \times \tfrac{2N - 2^k}{N} + (\tfrac{1}{N} \times \tfrac{2^k - N}{N}) \times 2 \times (\tfrac{1}{2} - \tfrac{N}{2^{k+1}}) = \tfrac{1}{2^k}.
\end{aligned}$$

□

By using the algorithm `ChooseAndShift`, we propose a new technique for obtaining the anonymity property. We call this technique "sampling twice."

Sampling Twice Doing the evaluation of the encryption (respectively the signing) twice with plaintext x (resp. message m), random r_1 and r_2, and the RSA function, and applying our proposed algorithm `ChooseAndShift` for the two resulting values.

In the following sections, by applying the sampling twice technique, we construct the schemes for encryption, undeniable and confirmer signature, and ring signature.

4 Encryption

4.1 Definitions

In [1], Bellare, Boldyreva, Desai, and Pointcheval proposed a new security requirement of encryption schemes called "key-privacy." It asks that the encryption provide (in addition to privacy of the data being encrypted) privacy of the key under which the encryption was performed. In [1], a public-key encryption scheme with common-key generation is described as follows.

Definition 5. *A public-key encryption scheme with common-key generation* $\mathcal{PE} = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ *consists of four algorithms. The common-key generation algorithm* $\mathcal{G}$ *takes as input some security parameter* k *and returns some common key* I*. The key generation algorithm* $\mathcal{K}$ *is a randomized algorithm that takes as input the*

common key I and returns a pair (pk, sk) of keys, the public key and a matching secret key. The encryption algorithm $\mathcal{E}$ is a randomized algorithm that takes the public key pk and a plaintext x to return a ciphertext y. The decryption algorithm $\mathcal{D}$ is a deterministic algorithm that takes the secret key sk and a ciphertext y to return the corresponding plaintext x or a special symbol $\perp$ to indicate that the ciphertext was invalid.

In [1], they formalized the property of "key-privacy." This can be considered under either the chosen-plaintext attack or the chosen-ciphertext attack, yielding two notions of security, IK-CPA and IK-CCA. (IK means "indistinguishability of keys.")

Definition 6 (IK-CPA, IK-CCA [1]). *Let $\mathcal{PE} = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ be an encryption scheme. Let $b \in \{0,1\}$ and $k \in \mathbb{N}$. Let $A_{\mathrm{cpa}} = (A^1_{\mathrm{cpa}}, A^2_{\mathrm{cpa}})$, $A_{\mathrm{cca}} = (A^1_{\mathrm{cca}}, A^2_{\mathrm{cca}})$ be adversaries that run in two stages and where A_{cca} has access to the oracles $\mathcal{D}_{sk_0}(\cdot)$ and $\mathcal{D}_{sk_1}(\cdot)$. Note that* si *is the state information. It contains pk_0, pk_1, and so on. For* $\mathrm{atk} \in \{\mathrm{cpa}, \mathrm{cca}\}$*, we consider the following experiments:*

$$
\begin{array}{l}
\texttt{Experiment } \mathbf{Exp}^{\text{ik-atk-}b}_{\mathcal{PE},A_{\mathrm{atk}}}(k) \\
\quad I \leftarrow \mathcal{G}(k);\ (pk_0, sk_0) \leftarrow \mathcal{K}(I);\ (pk_1, sk_1) \leftarrow \mathcal{K}(I) \\
\quad (x, \mathsf{si}) \leftarrow A^1_{\mathrm{atk}}(pk_0, pk_1);\ y \leftarrow \mathcal{E}_{pk_b}(x);\ d \leftarrow A^2_{\mathrm{atk}}(y, \mathsf{si}) \\
\quad \texttt{return } d
\end{array}
$$

Above it is mandated that A^2_{cca} never queries $\mathcal{D}_{sk_0}(\cdot)$ and $\mathcal{D}_{sk_1}(\cdot)$ on the challenge ciphertext y. For $\mathrm{atk} \in \{\mathrm{cpa}, \mathrm{cca}\}$*, we define the advantages via*

$$\mathbf{Adv}^{\text{ik-atk}}_{\mathcal{PE},A_{\mathrm{atk}}}(k) = \left|\Pr[\mathbf{Exp}^{\text{ik-atk-1}}_{\mathcal{PE},A_{\mathrm{atk}}}(k) = 1] - \Pr[\mathbf{Exp}^{\text{ik-atk-0}}_{\mathcal{PE},A_{\mathrm{atk}}}(k) = 1]\right|.$$

The scheme $\mathcal{PE}$ is said to be IK-CPA secure (respectively IK-CCA secure) if the function $\mathbf{Adv}^{\text{ik-cpa}}_{\mathcal{PE},A_{\mathrm{cpa}}}(\cdot)$ (resp. $\mathbf{Adv}^{\text{ik-cca}}_{\mathcal{PE},A_{\mathrm{cca}}}(\cdot)$) is negligible for any adversary A whose time complexity is polynomial in k.

4.2 Encryption with Sampling Twice

In this section, we propose the encryption scheme with the sampling twice technique.

Definition 7. *The common-key generation algorithm $\mathcal{G}$ takes a security parameter k and returns parameters k, k_0 and k_1 such that $k_0(k)+k_1(k) < k$ for all $k > 1$. This defines an associated plaintext-length function $n(k) = k - k_0(k) - k_1(k)$. The key generation algorithm $\mathcal{K}$ takes k, k_0, k_1, runs the key-generation algorithm of* RSA*, and gets N, e, d. The public key pk is $(N, e), k, k_0, k_1$ and the secret key sk is $(N, d), k, k_0, k_1$. The other algorithms are depicted below. Let $G : \{0,1\}^{k_0} \to \{0,1\}^{n+k_1}$ and $H : \{0,1\}^{n+k_1} \to \{0,1\}^{k_0}$ be hash functions. Note that $[x]^n$ denotes the n most significant bits of x and $[x]_m$ denotes the m least significant bits of x. Note that the valid ciphertext y satisfies $y \in [0, 2^k)$ and $(y \bmod N) \in \mathbb{Z}^*_N$.*

$\texttt{Algorithm}\ \mathcal{E}_{pk}^{G,H}(x)$	$\texttt{Algorithm}\ \mathcal{D}_{sk}^{G,H}(y)$
$r_1, r_2 \xleftarrow{R} \{0,1\}^{k_0}$	$v \leftarrow y \bmod N$
$s_1 \leftarrow (x\|\|0^{k_1}) \oplus G(r_1);\quad t_1 \leftarrow r_1 \oplus H(s_1)$	$s \leftarrow [v^d]^{n+k_1};\quad t \leftarrow [v^d]_{k_0}$
$v_1 \leftarrow (s_1\|\|t_1)^e \bmod N$	$r \leftarrow t \oplus H(s)$
$s_2 \leftarrow (x\|\|0^{k_1}) \oplus G(r_2);\quad t_2 \leftarrow r_2 \oplus H(s_2)$	$x \leftarrow [s \oplus G(r)]^n;\ p \leftarrow [s \oplus G(r)]_{k_1}$
$v_2 \leftarrow (s_2\|\|t_2)^e \bmod N$	$\texttt{if}\ (p = 0^{k_1})\ z \leftarrow x\ \texttt{else}\ z \leftarrow \perp$
$y \leftarrow \texttt{ChooseAndShift}(v_1, v_2)$	$\texttt{return}\ z$
$\texttt{return}\ y$	

4.3 Analysis

We compare the four schemes with sampling twice, repeating, RSACD, and expanding.

Security. Bellare, Boldyreva, Desai, and Pointcheval [1] proved that the scheme with repeating (RSA-RAEP) is secure in the sense of IND-CCA2 and IK-CCA in the random oracle model assuming RSA is θ-partial one-way for $\theta > 0.5$. Hayashi, Okamoto, and Tanaka [13] proved that the encryption scheme with RSACD is also secure in the sense of IND-CCA2 and IK-CCA in the random oracle model assuming RSACD is θ-partial one-way for $\theta > 0.5$.

In order to prove that the scheme with sampling twice is secure in the sense of IK-CCA, we need the restriction as follows.

Since if c is a ciphertext of m for $pk = (N, e, k)$ and $c < 2^k - N$ then $c + N$ is also a ciphertext of m, the adversary can ask $c + N_0$ to decryption oracle $\mathcal{D}_{sk_0}$ where c is a challenge ciphertext such that $c < 2^k - N_0$ and $pk_0 = (N_0, e_0, k)$, and if the answer of $\mathcal{D}_{sk_0}$ is m, then c is encrypted by pk_0.

To prevent this attack, we add some natural restriction to the adversaries in the definitions of IK-CCA. That is, it is mandated that the adversary never queries D_{sk_0} on $(c \bmod N_0) + \beta_0 N_0$ where $\beta_0 \in \lfloor (2^k - (c \bmod N_0))/N_0 \rfloor$, and D_{sk_1} on $(c \bmod N_1) + \beta_1 N_1$ where $\beta_1 \in \lfloor (2^k - (c \bmod N_1))/N_1 \rfloor$.

Similarly, in order to prove that the scheme with sampling twice is secure in the sense of IND-CCA2, we need the same restriction. That is, in the definition of IND-CCA2, it is mandated that the adversary never queries D_{sk} on $(c \bmod N) + \gamma N$ where $\gamma \in \lfloor (2^k - (c \bmod N))/N \rfloor$.

We think these restrictions are natural and reasonable. Actually, in the case of undeniable and confirmer signature schemes, Galbraith and Mao [10] defined the anonymity on undeniable signature schemes with the above restriction.

If we add these restrictions then we can prove that the scheme with sampling twice is secure in the sense of IK-CCA in the random oracle model assuming RSA is θ-partial one-way for $\theta > 0.5$. More precisely, we can prove the following theorem.

Theorem 2. *If* RSA *is partial one-way then the encryption scheme Π with sampling twice is secure in the sense of IK-CCA in the random oracle model. More precisely, for any adversary A attacking the anonymity of our scheme under*

an adaptive chosen-ciphertext attack, and making at most q_{dec} decryption oracle queries, q_{gen} G-oracle queries, and q_{hash} H-oracle queries, there exists a θ-partial inverting adversary B for the RSA *family, such that for any $k, k_0(k), k_1(k)$, and $\theta = \frac{k-k_0(k)}{k}$,*

$$\mathbf{Adv}^{\text{ik-cca}}_{\Pi,A}(k) \leq 8q_{\mathrm{hash}} \cdot ((1-\epsilon_1)\cdot(1-\epsilon_2)\cdot(1-\epsilon_3))^{-1} \cdot \mathbf{Adv}^{\theta\text{-pow-fnc}}_{\mathsf{RSA},B}(k) + q_{\mathrm{gen}} \cdot q_{\mathrm{hash}} \cdot (1-\epsilon_3)^{-1} \cdot 2^{-k+3}$$

where

$$\epsilon_1 = \frac{1}{2}; \quad \epsilon_2 = \frac{1}{2^{k/2-3}-1}\ ; \quad \epsilon_3 = \frac{2q_{\mathrm{gen}} + q_{\mathrm{dec}} + 2q_{\mathrm{gen}}q_{\mathrm{dec}}}{2^{k_0}} + \frac{2q_{\mathrm{gen}}}{2^{k_1}} + \frac{2q_{\mathrm{hash}}}{2^{k-k_0}},$$

and the running time of B is that of A plus $q_{\mathrm{gen}} \cdot q_{\mathrm{hash}} \cdot O(k^3)$.

Noticing that the range of valid ciphertexts changes, the proof is similar to that for RSA-RAEP (See Appendix C in the full version of [1].), and will be available in the full version of this paper.

We can also prove that the scheme with sampling twice is secure in the sense of IND-CCA2 in the random oracle model assuming RSA is θ-partial one-way for $\theta > 0.5$. More precisely, we can prove that if there exists a CCA2-adversary $A = (A_1, A_2)$ attacking indistinguishability of our scheme with advantage ϵ, then there exists a CCA2-adversary $B = (B_1, B_2)$ attacking indistinguishability of RSA-OAEP with advantage $\epsilon/2$. We construct B as follows.

1. B_1 gets pk and passes it to A_1. B_1 gets (m_0, m_1, si) which is an output of A_1, and B_1 outputs it.
2. B_2 gets a challenge ciphertext y and sets $y' \leftarrow y + tN$ where $t \stackrel{R}{\leftarrow} \{0,1\}$. If $y' \geq 2^k$ then B_2 outputs Fail and halts; otherwise B_2 passes (y', si) to A_2. B_2 gets $d \in \{0,1\}$ which is an output of A_2, and B_2 outputs it.

If B does not output Fail, A outputs correctly with advantage ϵ. Since $\Pr[B$ outputs Fail$] < 1/2$, the advantage of B is greater than $\epsilon/2$.

Efficiency. We show the number of modular exponentiations to encrypt and decrypt, the size of ciphertexts, and the bit-length of randomness to encrypt in Figure 2. We assume that N is uniformly distributed in $(2^{k-1}, 2^k)$.

5 Undeniable and Confirmer Signature

5.1 Definitions

Digital signatures are easily verified as authentic by anyone using the corresponding public key. This property can be advantageous for many users, but it is unsuitable for many other users. Chaum and Antwerpen provided undeniable signature which cannot be verified without the signer's cooperation [7, 8]. The validity or invalidity of an undeniable signature can be ascertained by conducting a protocol with the signer, assuming the signer participates. Chaum

	Sampling Twice	Repeating [1]	RSACD [13]	Expanding
# of mod. exp. to encrypt (average / worst)	2 / 2	1.5 / k_1	1.5 / 2	1 / 1
# of mod. exp. to decrypt (average / worst)	1 / 1	1 / 1	1.5 / 2	1 / 1
size of ciphertexts	k	k	k	$k + 160$
# of random bits to encrypt (average / worst)	$2k_0 + k + 3$ / $2k_0 + k + 3$	$1.5k_0$ / $k_1 k_0$	$1.5k_0$ / $1.5k_0$	$k_0 + 160$ / $k_0 + 160$

Fig. 2. The comparison of the encryption schemes

provided confirmer signature [9] which is undeniable signature where signatures may also be verified by interacting with an entity called the confirmer who has been designated by the signer, and many undeniable and confirmer signature schemes were proposed. We describe the definition of undeniable and confirmer signature.

Definition 8. *An undeniable signature scheme* $\mathcal{SIG}$ = (Cgen, Kgen, Sign, Conf, Deny) *consists of three algorithms and two protocols.*

- Cgen *is a (randomized) common-key generation algorithm that takes as input some security parameter* k *and returns a common key* I*.*
- Kgen *is a (randomized) key generation algorithm that takes as input the common key* I *and returns a pair* (pk, sk) *of keys, the public key and a matching secret key.*
- Sign *is a (randomized) signing algorithm that takes as input a secret key* sk *and a message* m *and outputs a signature* s*.*
- Conf *is a confirmation protocol between a signer and a verifier which takes as input a message* m*, a signature* s*, and signer's public key* pk *and allows the signer to prove to a verifier that the signature* s *is valid for the message* m *and the key* pk*.*
- Deny *is a denial protocol between a signer and a verifier which takes as input a message* m*, a signature* s*, and signer's public key* pk *and allows the signer to prove to a verifier that the signature* s *is invalid for the message* m *and the key* pk*.*

A confirmer signature scheme is essentially the same as above, except the role of confirmation and denial can also be performed by a third party called a confirmer. The significant modification is that the key generation algorithm produces a confirmation key ck *which is needed for the confirmation or denial protocol.*

Galbraith and Mao proposed a new security notion of undeniable and confirmer signatures named "anonymity" in [10]. We say that an undeniable or confirmer signature scheme provides anonymity when it is infeasible to determine which user generated the message-signature pair.

We slightly modify the definition of anonymity in [10] in order to put a common key generation into it explicitly.

Definition 9 ([10]). *Let $\mathcal{SIG} = (\textsc{Cgen}, \textsc{Kgen}, \textsc{Sign}, \textsc{Conf}, \textsc{Deny})$ be an undeniable or confirmer signature scheme. Let $b \in \{0,1\}$ and $k \in \mathbb{N}$ (security parameter). Let $A = (A_1, A_2)$ be adversaries that run in two stages. A has access to the oracles $\textsc{Sign}_{sk_0}, \textsc{Sign}_{sk_1}$ and A can execute confirmation and denial protocols $\textsc{Conf}_{sk_0}, \textsc{Conf}_{sk_1}, \textsc{Deny}_{sk_0}, \textsc{Deny}_{sk_1}$ on any message-signature pair. However, it is mandated that A_2 never execute $\textsc{Conf}_{sk_0}, \textsc{Conf}_{sk_1}, \textsc{Deny}_{sk_0}, \textsc{Deny}_{sk_1}$ on $(m', \sigma') \in EC(m, \sigma, pk_0) \cup EC(m, \sigma, pk_1)$ (EC means "equivalence class." If we get a message-signature pair (m, σ) under the key pk, then we can easily compute all elements in $EC(m, \sigma, pk)$.). Note that* si *be a state information. It contains common keys, public keys, and so on. Now we consider the following experiments:*

$\texttt{Experiment } \mathbf{Exp}^{\text{Anonym-}b}_{\mathcal{SIG},A}(k)$
$I \leftarrow \textsc{Cgen}(1^k);\ (pk_0, sk_0) \leftarrow \textsc{Kgen}(I);\ (pk_1, sk_1) \leftarrow \textsc{Kgen}(I)$
$(m, \mathsf{si}) \leftarrow A_1(pk_0, pk_1);\ \sigma \leftarrow \textsc{Sign}_{sk_b}(m);\ d \leftarrow A_2(m, \sigma, \mathsf{si})$
$\texttt{return } d$

We define the advantages of the adversaries via:

$$\mathbf{Adv}^{\text{Anonym}}_{\mathcal{SIG},A}(k) = \left|\Pr[\mathbf{Exp}^{\text{Anonym-1}}_{\mathcal{SIG},A}(k) = 1] - \Pr[\mathbf{Exp}^{\text{Anonym-0}}_{\mathcal{SIG},A}(k) = 1]\right|.$$

The scheme $\mathcal{SIG}$ provides anonymity if the function $\mathbf{Adv}^{\text{Anonym}}_{\mathcal{SIG},A}(\cdot)$ is negligible for any adversary A whose time complexity is polynomial in k.

5.2 Undeniable and Confirmer Signature with Sampling Twice

In this section, we propose the undeniable and confirmer signature schemes with the sampling twice technique.

Definition 10. *The common-key generation algorithm* Cgen *takes a security parameter k and returns parameters k, k_0 and k_1 such that $k_0(k) + k_1(k) < k$ for all $k > 1$. The key generation algorithm* Kgen *takes k, k_0, k_1, runs the key-generation algorithm of* RSA*, and gets N, e, d, p, q where p, q are the safe primes (i.e. $(p-1)/2$ and $(q-1)/2$ are also primes) [1]. It picks g from $\mathbb{Z}_N^*$ and sets $h \leftarrow g^d \bmod N$. The public key pk is $(N, g, h), k, k_0, k_1$ and the secret key sk is $(N, e, d, p, q), k, k_0, k_1$. Let $G_0 : \{0,1\}^* \to \{0,1\}^{k_1}$, $G_1 : \{0,1\}^{k_1} \to \{0,1\}^{k_0}$, $G_2 : \{0,1\}^{k_1} \to \{0,1\}^{k-k_0-k_1-1}$, and $F : \{0,1\}^k \to \{0,1\}^k$ be hash functions. The signing algorithm is as follows.*

$\textsc{Sign}(m)$
$r_1, r_2 \xleftarrow{R} \{0,1\}^{k_0}$
$\bar{m_1} \leftarrow \textsc{Sign2}(m, r_1);\ t_1 \xleftarrow{R} \{c \in \mathbb{Z}_N \mid c^2 = \pm\bar{m_1} \pmod N\};\ s_1 \leftarrow (t_1)^d \bmod N$
$\bar{m_2} \leftarrow \textsc{Sign2}(m, r_2);\ t_2 \xleftarrow{R} \{c \in \mathbb{Z}_N \mid c^2 = \pm\bar{m_2} \pmod N\};\ s_2 \leftarrow (t_2)^d \bmod N$
$s \leftarrow \texttt{ChooseAndShift}(s_1, s_2)$
$\texttt{if } (s \bmod N = s_1)\ r \leftarrow r_1\ \texttt{else}\ r \leftarrow r_2$
$\texttt{return } (s, r)$

[1] We need this restriction for proving anonymity.

where

$\textsc{Sign2}(m, r)$
$w \leftarrow G_0(m||r);\ r^* \leftarrow G_1(w) \oplus r;\ M \leftarrow 0||w||r^*||G_2(w);\ \bar{m} \leftarrow M$
`while` $\left(\left(\frac{\bar{m}}{N}\right) \neq 1\right)$ `repeat` $\bar{m} \leftarrow F(\bar{m})$
`return` $\bar{m}$

Conf *(respectively* Deny*) is a non-interactive designated verifier proof which proves the knowledge of an integer* e *such that* $g = h^e \pmod N$ *and* $s^{2e} = \pm\textsc{Sign2}(m, r) \pmod N$ *(resp.* $g = h^e \pmod N$ *and* $s^{2e} \neq \pm\textsc{Sign2}(m, r) \pmod N$*). To construct such proofs, we first employ protocols similar to those in [14] by Galbraith, Mao, and Paterson. Then, we transform them to corresponding non-interactive designated verifier proofs by the method of Jakobsson, Sako, and Impagliazzo [15]* [2]*. The equivalence class of this scheme is* $EC(m, (s, r), pk) = \{(m, (\pm s' \pm uN, r)) \mid s' = s \bmod N \wedge u \in \lfloor (2^k - s')/N \rfloor\}$.

In our scheme (and also the scheme by Galbraith and Mao), we have to use RSA moduli which are the products of safe primes for obtaining the anonymity property. Gennaro, Krawczyk, and Rabin [16] proposed the RSA-based undeniable signature schemes where RSA moduli are restricted to the products of safe primes, and the confirmation and denial protocols in [16] is more efficient than those by Galbraith, Mao, and Paterson [14]. Therefore, it seems better to use the protocols in [16]. However, if we use the protocols in [16], the prover will have to prove that her RSA modulo has the proper form (i.e. a product of safe primes) during the protocols, and it needs a costly proof. To avoid this, Galbraith, Mao, and Paterson [14] constructed different scheme where there is no restriction for the RSA moduli.

5.3 Analysis

We compare the four schemes with sampling twice, expanding, and repeating.
Security. Galbraith and Mao [10] proved that their scheme provides anonymity in the random oracle model under the assumption that the composite decision Diffie-Hellman problem is hard (Given $(g, h, u, v) \in (\mathbb{Z}_N^*)^4$, it is infeasible to determine whether the two equations $h = g^r \pmod N$ and $v = \alpha u^r \pmod N$ hold, where $r \in \mathbb{Z}_{\phi(N)}^*$ and $\mathrm{ord}(\alpha) = 2$. See [10] for details.). They also proved that their scheme is existential unforgeable in the random oracle model under the assumption that factoring integers which are products of safe primes is hard. We can prove that the scheme with sampling twice provides anonymity in the random oracle model under the assumption that the composite decision Diffie-Hellman problem is hard, and is existential unforgeable in the random oracle model under the assumption that factoring integers which are products of safe primes is hard. Noticing that the signature space changes, the proofs are similar to those for the Galbraith–Mao scheme (See Appendices B and C in [10].).

[2] These proof transcripts must be encrypted when sent to the verifier if anonymity is to be preserved.

	Sampling Twice	Expanding [10]	Repeating
# of mod. exp. to sign (average / worst)	2 / 2	1 / 1	1.5 / k_1
# of computation of square root (average / worst)	2 / 2	1 / 1	1.5 / k_1
size of signatures	$k + k_0$	$2k + k_0$	$(k-1) + k_0$
# of random bits to sign (average / worst)	$k_0 + k + 5$ / $k_0 + k + 5$	$k_0 + k + 2$ / $k_0 + k + 2$	$1.5(k_0 + 2)$ / $k_1(k_0 + 2)$

Fig. 3. The comparison of the undeniable and confirmer signature schemes

Efficiency. We show the number of modular exponentiations to sign, the number of computation of square root, the size of signatures, and the number of random bits to sign in Figure 3. We assume that N is uniformly distributed in $(2^{k-1}, 2^k)$.

6 Ring Signature

6.1 Definitions

In [12], Rivest, Shamir, and Tauman proposed the notion of ring signature, which allows a member of an ad hoc collection of users S to prove that a message is authenticated by a member of S without revealing which member actually produced the signature. Unlike group signature, ring signature has no group managers, no setup procedures, no revocation procedures, and no coordination.

Definition 11 (Ring Signature [12]). *One assumes that each user U_i (called a ring member) has received (via a PKI or a certificate) a public key P_i, for which the corresponding secret key is denoted by S_i. A ring signature scheme consists of the following algorithms.*

- **ring-sign**$(m, P_1, P_2, \cdots, P_r, s, S_s)$ *which produces a ring signature σ for the message m, given the public keys $P_1, P_2, \cdots, P_r$ of the r ring members, together with the secret key S_s of the s-th member (who is the actual signer).*
- **ring-verify**(m, σ) *which accepts a message m and a signature σ (which includes the public key of all the possible signers), and outputs either* `valid` *or* `invalid`.

The signer does not need the knowledge, consent, or assistance of the other ring members to put them in the ring. All he needs is knowledge of their regular public keys. Verification must satisfy the usual soundness and completeness conditions, but in addition the signature scheme must satisfy "signer-ambiguity," which is the property that the verifier is unable to determine the identity of the actual signer with probability greater than $1/r + \epsilon$, where r is the size of the ring and ϵ is negligible. Furthermore, the signature scheme must satisfy "existential unforgeability under adaptive chosen message attack."

The formal concept of ring signature can be related to an abstract concept called *combining functions*. In [12], Rivest, Shamir, and Tauman proposed a combining function based on a symmetric encryption scheme E modeled by a (keyed) random permutation

$$C_{k,v}(y_1, \cdots, y_r) = E_k(y_r \oplus E_k(y_{r-1} \oplus \cdots E_k(y_2 \oplus E_k(y_1 \oplus v)) \cdots)).$$

For any k, v, z, any index s, and any fixed values of $\{y_i\}_{i \neq s}$, we can easily find y_s such that $C_{k,v}(y_1, \cdots, y_r) = z$ by using the following equation:

$$y_s = E_k^{-1}\big(y_{s+1} \oplus \cdots E_k^{-1}(y_r \oplus E_k^{-1}(z)) \cdots\big) \oplus E_k\big(y_{s-1} \oplus \cdots E_k(y_1 \oplus v) \cdots\big).$$

6.2 Ring Signature with Sampling Twice

In this section, we propose a ring signature scheme with the sampling twice technique. To verify the signatures deterministically, we add some information c_i to the signature.

Definition 12. *Let ℓ, k be security parameters. Let E be a symmetric encryption scheme over $\{0,1\}^k$ using ℓ-bit keys, and let h be a hash function which maps strings of arbitrary length to ℓ-bit strings. Each user U_i has public key $P_i = (N_i, e_i, k)$ and secret key $S_i = (N_i, d_i, k)$ by running the key generation algorithm of* RSA *with security parameter k (i.e. the size of N_i is k). Let r be the number of ring members. The signing algorithm is as follows.*

ring-sign$(m, P_1, P_2, \cdots, P_r, s, S_s)$
  `for each` $i \in \{1, \cdots, s-1, s+1, \cdots, r\}$ `do`
    $x_i^1, x_i^2 \xleftarrow{R} \mathbb{Z}_{N_i}^*$
    $y_i^1 \leftarrow (x_i^1)^{e_i} \bmod N_i$; $y_i^2 \leftarrow (x_i^2)^{e_i} \bmod N_i$
    $y_i \leftarrow$ `ChooseAndShift`(y_i^1, y_i^2)
    `if` $(y_i \bmod N_i = y_i^1)$ $x_i \leftarrow x_i^1$ `else` $x_i \leftarrow x_i^2$
    `if` $(y_i \geq N_i)$ $c_i \leftarrow 1$ `else` $c_i \leftarrow 0$

  $v \xleftarrow{R} \{0,1\}^k$
  `find` y_s `s.t.` $C_{h(m),v}(y_1, \cdots, y_r) = v$
  `if` $(y_s \geq N_s)$ $c_s \leftarrow 1$ `else` $c_s \leftarrow 0$
  $x_s \leftarrow (y_s)^{d_s} \bmod N_s$
  `return` $\sigma = (P_1, P_2, \cdots, P_r, v, (x_1, c_1), (x_2, c_2), \cdots, (x_r, c_r))$

The verification algorithm **ring-verify**(m, σ) *computes $y_i \leftarrow ((x_i)^{e_i} \bmod N_i) + c_i \cdot N_i$ for each (x_i, c_i) and $z \leftarrow C_{h(m),v}(y_1, \cdots, y_r)$. It returns* `valid` *if and only if $z = v$.*

6.3 Analysis

We compare the four schemes with sampling twice, expanding, RSACD, and repeating.

	Sampling Twice	Expanding [12]	RSACD [13]	Repeating
# of mod. exp. to sign (average / worst)	$2r$ / $2r$	r / r	$1.5r$ / $2r$	$1.5r$ / kr
# of mod. exp. to verify (average / worst)	r / r	r / r	$1.5r$ / $2r$	r / r
size of signatures	$(3r+1)k+r$	$(3r+1)k$ $+160(r+1)$	$(3r+1)k$	$(3r+1)k-1$
# of random bits to sign (average / worst)	$3(k+1)(r-1)+k$ / $3(k+1)(r-1)+k$	$(k+160)r$ / $(k+160)r$	kr / kr	$1.5k(r-1)+k-1$ / $k^2(r-1)+k-1$

Fig. 4. The comparison of the ring signature schemes ($|N_i| = k$)

Security. Rivest, Shamir, and Tauman [12] proved that their scheme is unconditionally signer-ambiguous and provably secure in the ideal cipher model assuming RSA is one-way. Hayashi, Okamoto, and Tanaka [13] proved that their scheme is unconditionally signer-ambiguous and provably secure in the ideal cipher model assuming RSACD is one-way.

We can prove that our scheme is unconditionally signer-ambiguous, since for each k and v the equation $C_{h(m),v}(y_1, \cdots, y_r) = v$ has exactly $(2^{k-1})^{r-1}$ solutions, and all of them are chosen by the signature generation procedure with equal probability, regardless of the signer's identity.

We can also prove that our scheme is existential unforgeable under adaptive chosen message attack in the ideal cipher model assuming RSA is one-way. The proof is almost the same as that for the Rivest–Shamir–Tauman scheme. The difference is as follows.

In the proof of unforgeability for the Rivest–Shamir–Tauman scheme, given $y \in \mathbb{Z}_N^*$, one slips y as a "gap" between two consecutive E functions along the ring. Then, the forger has to compute the e-th root of y, and this leads one to obtain the e-th root of y.

In the proof for our scheme, given $y \in \mathbb{Z}_N^*$, we pick a random bit $t \in \{0, 1\}$, set $y' \leftarrow y + tN$. If $y' < 2^k$ then one slips y' as a "gap" between two consecutive E functions along the ring. The rest of the proof is the same as that for the Rivest–Shamir–Tauman scheme (See Section 3.5 in [12].).

Recently, Bresson, Stern, and Szydlo [17] improved the ring signature scheme of Rivest, Shamir, and Tauman. They showed that its security can be based on the random oracle model, which is strictly weaker than the ideal cipher model. Furthermore, this greatly simplified the security proof provided in [12]. We can apply their construction to the schemes with sampling twice and RSACD.

Efficiency. We show the number of modular exponentiations to sign and to verify, the size of signatures, and the number of random bits to sign in Figure 4. We assume that each N_i is uniformly distributed in $(2^{k-1}, 2^k)$.

In the schemes with sampling twice and RSACD, it is necessary for each ring member to choose her RSA modulo with the same length, and in the scheme

with repeating, it is necessary for each ring member to choose her RSA modulo with almost the same length. In contrast to these schemes, in the scheme with expanding, there is no restriction on the lengths of users' moduli. However, if there is one ring member whose RSA modulo is much larger than the other member's moduli, then the size of the signature and the number of random bits depends on the largest modulo. For example, if there is a user whose RSA modulo has length $k + \ell$ and the other users' moduli have lengths k, then the size of signature is $(3r + 1)k + 160(r + 1) + \ell(r + 4)$ and the number of random bits to sign is $r(k + 160) + r\ell$.

7 Concluding Remarks

In this paper, we have proposed a new technique for obtaining the anonymity property of RSA-based cryptosystems, which we call "sampling twice." By applying the sampling twice technique, we have constructed the schemes for encryption, undeniable and confirmer signature, and ring signature.

In our analysis, we have observed that the scheme with sampling twice is efficient with respect to the sizes of ciphertexts and signatures, the computational costs to decrypt ciphertexts and to verify signatures in the average and worst cases, and the computational costs to encrypt messages and to sign messages in the worst case.

Acknowledgements

We thank the anonymous referees for valuable comments.

References

1. Bellare, M., Boldyreva, A., Desai, A., Pointcheval, D.: Key-Privacy in Public-Key Encryption. [18] 566–582 Full version of this paper, available via `http://www-cse.ucsd.edu/users/mihir/`.
2. Bellare, M., Rogaway, P.: Optimal Asymmetric Encryption – How to Encrypt with RSA. [19] 92–111
3. Fujisaki, E., Okamoto, T., Pointcheval, D., Stern, J.: RSA-OAEP is Secure under the RSA Assumption. In Kilian, J., ed.: Advances in Cryptology – CRYPTO 2001. Volume 2139 of Lecture Notes in Computer Science., Santa Barbara, California, USA, Springer-Verlag (2001) 260–274
4. Krawczyk, H.: SKEME: A Versatile Secure Key Exchange Mechanism for Internet. In: Proceedings of the 1996 Internet Society Symposium on Network and Distributed System Security, San Diego, CA, USA (1996) 114–127
5. Camenisch, J., Lysyanskaya, A.: Efficient Non-Transferable Anonymous Multi-Show Credential System with Optional Anonymity Revocation. In Pfitzmann, B., ed.: Advances in Cryptology – EUROCRYPT 2001. Volume 2045 of Lecture Notes in Computer Science., Innsbruck, Austria, Springer-Verlag (2001) 93–118

6. Sako, K.: An Auction Protocol Which Hides Bids of Losers. In Imai, H., Zheng, Y., eds.: Public Key Cryptography – PKC 2000. Volume 1751 of Lecture Notes in Computer Science., Melbourne, Victoria, Australia, Springer-Verlag (2000) 422–432
7. Chaum, D., Antwerpen, H.V.: Undeniable Signatures. In Brassard, G., ed.: Advances in Cryptology – CRYPTO '89. Volume 435 of Lecture Notes in Computer Science., Santa Barbara, California, USA, Springer-Verlag (1989) 212–217
8. Chaum, D.: Zero-Knowledge Undeniable Signatures. In Damgård, I., ed.: Advances in Cryptology – EUROCRYPT '90. Volume 473 of Lecture Notes in Computer Science., Aarhus, Denmark, Springer-Verlag (1990) 458–464
9. Chaum, D.: Designated Confirmer Signatures. [19] 86–91
10. Galbraith, S.D., Mao, W.: Invisibility and Anonymity of Undeniable and Confirmer Signatures. In Joye, M., ed.: Topics in Cryptology – CT-RSA 2003. Volume 2612 of Lecture Notes in Computer Science., San Francisco, CA, USA, Springer-Verlag (2003) 80–97
11. Desmedt, Y.: Securing traceability of ciphertexts: Towards a secure software escrow scheme. In Guillou, L.C., Quisquater, J.J., eds.: Advances in Cryptology – EUROCRYPT '95. Volume 921 of Lecture Notes in Computer Science., Saint-Malo, France, Springer-Verlag (1995) 147–157
12. Rivest, R.L., Shamir, A., Tauman, Y.: How to Leak a Secret. [18] 552–565
13. Hayashi, R., Okamoto, T., Tanaka, K.: An RSA Family of Trap-door Permutations with a Common Domain and its Applications. In Bao, F., Deng, R.H., Zhou, J., eds.: Public Key Cryptography – PKC 2004. Volume 2947 of Lecture Notes in Computer Science., Singapore, Springer-Verlag (2004) 291–304
14. Galbraith, S.D., Mao, W., Paterson, K.G.: RSA-based Undeniable Signatures for General Moduli. In Preneel, B., ed.: Topics in Cryptology – CT-RSA 2002. Volume 2271 of Lecture Notes in Computer Science., San Jose, CA, USA, Springer-Verlag (2002) 200–217
15. Jakobsson, M., Sako, K., Impagliazzo, R.: Designated Verifier Proofs and their Applications. In Maurer, U., ed.: Advances in Cryptology – EUROCRYPT '96. Volume 1070 of Lecture Notes in Computer Science., Saragossa, Spain, Springer-Verlag (1996) 143–154
16. Gennaro, R., Krawczyk, H., Rabin, T.: RSA-based Undeniable Signatures. In Kaliski, Jr., B.S., ed.: Advances in Cryptology – CRYPTO '97. Volume 1294 of Lecture Notes in Computer Science., Santa Barbara, California, USA, Springer-Verlag (1997) 132–149
17. Bresson, E., Stern, J., Szydlo, M.: Threshold Ring Signatures and Applications to Ad-hoc Groups. In Yung, M., ed.: Advances in Cryptology – CRYPTO 2002. Volume 2442 of Lecture Notes in Computer Science., Santa Barbara, California, USA, Springer-Verlag (2002) 465–480
18. Boyd, C., ed.: Advances in Cryptology – ASIACRYPT 2001. Volume 2248 of Lecture Notes in Computer Science., Gold Coast, Australia, Springer-Verlag (2001)
19. De Santis, A., ed.: Advances in Cryptology – EUROCRYPT '94. Volume 950 of Lecture Notes in Computer Science., Perugia, Italy, Springer-Verlag (1994)