ISSN 1342-2812

Research Reports on Mathematical and Computing Sciences

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

Ryotaro Hayashi and Keisuke Tanaka

December 2004, C–201

Department of
Mathematical and
Computing Sciences
Tokyo Institute of Technology

SERIES C: Computer Science

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
W8-55, 2-12-1 Ookayama Meguro-ku, Tokyo 152-8552, Japan
{hayashi9, keisuke}@is.titech.ac.jp

December 16, 2004

Abstract

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 $[0, 2^k)$ 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 advantages to the previous schemes.

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

1 Introduction

1.1 Background

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 [2] proposed a new security requirement of the encryption schemes called "key-privacy" or "anonymity." It asks that the encryption provide (in addition

*Supported in part by NTT Information Sharing Platform Laboratories and Grant-in-Aid for Scientific Research, Ministry of Education, Culture, Sports, Science, and Technology, 14780190, 16092206.

to privacy of the data being encrypted) privacy of the key under which the encryption was performed. In [2], they provided the key-privacy encryption scheme, RSA-RAEP, which is a variant of RSA-OAEP, (Bellare and Rogaway [3], Fujisaki, Okamoto, Pointcheval, and Stern [16]), 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 [22]), anonymous credential system (Camenisch and Lysyanskaya [7]), and auction protocols (Sako [26]).

Chaum and Antwerpen provided undeniable signature which cannot be verified without the signer's cooperation [11, 9]. 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 [10] 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 [17]. 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 [17], 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 $[0, 2^{2k})$ where N is a public key for each user and $|N| = k$. This technique was proposed by Desmedt [14].

Rivest, Shamir, and Tauman [25] 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 [25], 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 [20] 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 encryption and ring signature schemes. Their schemes have some advantages to the previous schemes.

1.2 Our Contribution

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

	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.

Figure 1: The previous and our proposed schemes

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 [2], 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 [14]. In [17], Galbraith and Mao used this technique for the undeniable signature scheme. In [25], Rivest, Shamir, and Tauman also used this technique for the ring signature scheme.

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 [20].

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.).

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 those of the previous schemes. More precisely, in our encryption scheme, the number of modular exponentiations to encrypt in the average 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 modular exponentiations for signing and that of computation of square roots are 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 in the average case.

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 function 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 concerning families of functions. We also describe the definitions of RSA and RSACD. 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, families of trap-door permutations)**.** *A family of functions $F = (K, S, E)$ is specified by three algorithms. The randomized key-generation algorithm K takes as input a security parameter k 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 pk and returns a random point in a set that we call the domain of the function and denote by $\mathrm{Dom}_F(pk)$. The deterministic evaluation algorithm E takes pk and $x \in \mathrm{Dom}_F(pk)$ and returns an output we denote by $E_{pk}(x)$. We let $\mathrm{Rng}_F(pk) = \{E_{pk}(x) \,|\, x \in \mathrm{Dom}_F(pk)\}$ denote the range of the function.*
We say that F is a family of trap-door permutations if $\mathrm{Dom}_F(pk) = \mathrm{Rng}_F(pk)$, E_{pk} is a bijection on this set, and there exists a deterministic inversion algorithm I that takes sk and $y \in \mathrm{Rng}_F(pk)$ and returns $x \in \mathrm{Dom}_F(pk)$ such that $E_{pk}(x) = y$.

Definition 2 (θ-partial one-way)**.** *Let $F = (K, S, E)$ be a family of functions. Let $b \in \{0, 1\}$ and $k \in \mathbb{N}$. Let $0 < \theta \leq 1$ be a constant. Let A be an adversary. We consider the following experiments:*

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

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.

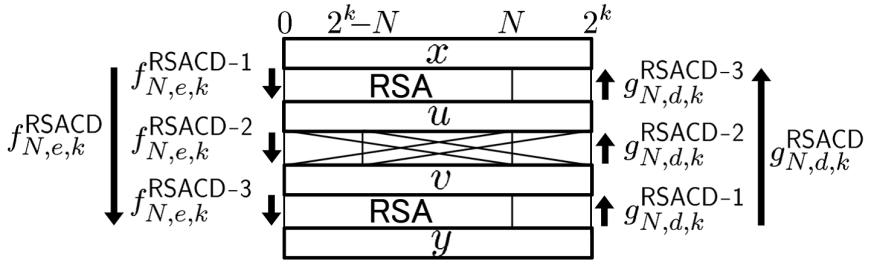

Figure 2: Functions $f^{\mathsf{RSACD}}_{N,e,k}$ and $g^{\mathsf{RSACD}}_{N,d,k}$

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 3 (the standard RSA family of trap-door permutations)**.** *The standard RSA family of trap-door permutations* $\mathsf{RSA} = (K, S, E)$ *is 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* $N = pq$ *and picks* $e, d \in \mathbb{Z}^*_{\phi(N)}$ *such that* $ed = 1 \pmod{\phi(N)}$*. 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 [16] showed that the θ-partial one-wayness of RSA is equivalent to the one-wayness of RSA for $\theta > 0.5$.

Hayashi, Okamoto, and Tanaka [20] proposed the RSA family of trap-door permutations with a common domain denoted by RSACD. We describe their family of trap-door permutations with a common domain. 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.

Definition 4 (the RSA family of trap-door permutations with a common domain [20])**.** *The specifications of the RSA family of trap-door permutations with a common domain* $\mathsf{RSACD} = (K, S, E)$ *are as follows. The key generation algorithm is the same as that for* RSA*. The sets* $\mathrm{Dom}_{\mathsf{RSACD}}(N, e, k)$ *and* $\mathrm{Rng}_{\mathsf{RSACD}}(N, e, k)$ *are both* $\{x \mid x \in [0, 2^k) \ \wedge \ x \bmod N \in \mathbb{Z}^*_N\}$*. The sampling algorithm returns a random point in* $\mathrm{Dom}_{\mathsf{RSACD}}(N, e, k)$*. The evaluation algorithm* $E_{N,e,k}(x) = f^{\mathsf{RSACD}}_{N,e,k}(x)$ *and the inversion algorithm* $I_{N,d,k}(y) = g^{\mathsf{RSACD}}_{N,d,k}(y)$ *are as follows (See Figure 2.).*

```
Function f^RSACD_{N,e,k}(x)
  u ← f^RSACD-1_{N,e,k}(x); v ← f^RSACD-2_{N,e,k}(u); y ← f^RSACD-3_{N,e,k}(v)
  return y
```

```
Function f^RSACD-1_{N,e,k}(x)
  if (x < N) u ← x^e mod N
  else u ← x
  return u
```

```
Function f^RSACD-2_{N,e,k}(u)
  if (u < 2^k − N) v ← u + N
  elseif (2^k − N ≤ u < N) v ← u
  else v ← u − N
  return v
```

```
Function f^RSACD-3_{N,e,k}(v)
  if (v < N) y ← v^e mod N
  else y ← v
  return y
```

$\texttt{Function } g^{\mathsf{RSACD}}_{N,d,k}(y)$
$\quad v \leftarrow g^{\mathsf{RSACD\text{-}1}}_{N,d,k}(y);\ u \leftarrow g^{\mathsf{RSACD\text{-}2}}_{N,d,k}(v);\ x \leftarrow g^{\mathsf{RSACD\text{-}3}}_{N,d,k}(u)$
$\quad \texttt{return } x$

$\texttt{Function } g^{\mathsf{RSACD\text{-}1}}_{N,d,k}(y)$
$\quad \texttt{if } (y < N)\ v \leftarrow y^d \bmod N$
$\quad \texttt{else } v \leftarrow y$
$\quad \texttt{return } v$

$\texttt{Function } g^{\mathsf{RSACD\text{-}2}}_{N,d,k}(v)$
$\quad \texttt{if } (v < 2^k - N)\ u \leftarrow v + N$
$\quad \texttt{elseif } (2^k - N \le v < N)\ u \leftarrow v$
$\quad \texttt{else } u \leftarrow v - N$
$\quad \texttt{return } u$

$\texttt{Function } g^{\mathsf{RSACD\text{-}3}}_{N,d,k}(u)$
$\quad \texttt{if } (u < N)\ x \leftarrow u^d \bmod N$
$\quad \texttt{else } x \leftarrow u$
$\quad \texttt{return } x$

The choice of N from $(2^{k-1}, 2^k)$ ensures that all elements in $\mathrm{Dom}_{\mathsf{RSACD}}(N, e, k)$ are permuted by the RSA function at least once. 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.

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 $|N| = k$.

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

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 $\Pr[\texttt{ChooseAndShift}_{N,k}(x_1, x_2) = z] = 1/2^k$ for any $z \in [0, 2^k)$. For any $z \in [0, 2^k - N)$, we have

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

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

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

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

□

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

The classical security requirements of public-key encryption schemes, for example indistinguishability or non-malleability under the chosen-ciphertext attack, provide privacy of the encryption data. In [2], Bellare, Boldyreva, Desai, and Pointcheval proposed a new security requirement of 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 a heterogeneous public-key environment, encryption will probably fail to be anonymous for trivial reasons. For example, different users might be using different cryptosystems, or, if the same cryptosystem, have keys of different lengths. In [2], 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 a security parameter k and returns some common key I.*
- *The key generation algorithm $\mathcal{K}$ is a randomized algorithm that takes as input a common key I and returns a pair (pk, sk) of keys, a 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 [2], they formalized the property of "key-privacy." Similar notions had been proposed Abadi and Rogaway [1], Fischlin [15], Camenisch and Lysyanskaya [7], Sako [26], and Desai [13], however, chosen-ciphertext attacks do not seem to have been considered before in the context of key-privacy. The definition by Bellare, Boldyreva, Desai, and Pointcheval [2] 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 [2]). *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 $A_{\text{cpa}} = (A^1_{\text{cpa}}, A^2_{\text{cpa}})$, $A_{\text{cca}} = (A^1_{\text{cca}}, A^2_{\text{cca}})$ 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* atk $\in$ {cpa, cca}*, we consider the following experiments:*

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

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

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

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

Bellare, Boldyreva, Desai, and Pointcheval [2] proposed the key-privacy encryption scheme with repeating called "RSA-RAEP," and Hayashi, Okamoto, and Tanaka [20] also provided that with RSACD. See Appendix A for details.

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 *with security parameter k, 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 $(y \bmod N) \in \mathbb{Z}^*_N$.*

$$\begin{array}{l}\texttt{Algorithm } \mathcal{E}^{G,H}_{pk}(x)\\ \quad r_1, r_2 \xleftarrow{R} \{0,1\}^{k_0}\\ \quad s_1 \leftarrow (x||0^{k_1}) \oplus G(r_1);\ t_1 \leftarrow r_1 \oplus H(s_1)\\ \quad v_1 \leftarrow (s_1||t_1)^e \bmod N\\ \quad s_2 \leftarrow (x||0^{k_1}) \oplus G(r_2);\ t_2 \leftarrow r_2 \oplus H(s_2)\\ \quad v_2 \leftarrow (s_2||t_2)^e \bmod N\\ \quad y \leftarrow \texttt{ChooseAndShift}(v_1, v_2)\\ \quad \texttt{return } y\end{array}$$

$$\begin{array}{l}\texttt{Algorithm } \mathcal{D}^{G,H}_{sk}(y)\\ \quad v \leftarrow y \bmod N\\ \quad s \leftarrow [v^d \bmod N]^{n+k_1};\ t \leftarrow [v^d \bmod N]_{k_0}\\ \quad r \leftarrow t \oplus H(s)\\ \quad x \leftarrow [s \oplus G(r)]^n;\ p \leftarrow [s \oplus G(r)]_{k_1}\\ \quad \texttt{if } (p = 0^{k_1})\ z \leftarrow x \texttt{ else } z \leftarrow \perp\\ \quad \texttt{return } z\end{array}$$

4.3 Analysis

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

Security. Bellare, Boldyreva, Desai, and Pointcheval [2] 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 [20] 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 the adversary can know that c was 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 either $c' \in [0, 2^k)$ such that $c' = c \pmod{N_0}$ to D_{sk_0} or $c'' \in [0, 2^k)$ such that $c'' = c \pmod{N_1}$ to D_{sk_1}.

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 $c' \in [0, 2^k)$ such that $c' = c \pmod{N}$ to D_{sk}.

We think these restrictions are natural and reasonable. Actually, in the case of undeniable and confirmer signature schemes, Galbraith and Mao [17] 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. *For any adversary A attacking the anonymity of our scheme Π 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 $\theta = \frac{k-k_0(k)}{k}$,*

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

where

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

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

The proof of the above theorem is in Appendix B.

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.

	Sampling Twice	Repeating [2]	RSACD [20]	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_1k_0	$1.5k_0$ / $1.5k_0$	$k_0 + 160$ / $k_0 + 160$

Figure 3: The comparison of the encryption schemes

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 $\Pr[B \text{ outputs Fail}] < 1/2$, the advantage of B is greater than $\epsilon/2$.

Efficiency. We show the number of modular exponentiations to encrypt, the number of modular exponentiations to decrypt, the size of ciphertexts, and the number of random bits to encrypt in Figure 3. 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 [11, 9]. 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 [10] 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 [19, 24, 8, 18]. We describe the definition of undeniable and confirmer signature.

Definition 8. *An undeniable signature scheme* $\mathcal{SIG} = (\text{CGEN}, \text{KGEN}, \text{SIGN}, \text{CONF}, \text{DENY})$ *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.

The literature on confirmer signature is inconsistent on whether the original signer has the ability to confirm and/or deny signatures. Camenisch and Michels [8] claim that it is undesirable for signers to be able to confirm or deny their signatures and the schemes in [8, 10, 24] do not allow signers to deny signatures. On the other hand, Galbraith and Mao claim that it is important for signers to be able to confirm and/or deny signatures and the schemes in [11, 9, 18, 19] do allow signers to deny signatures. In any case, these distinctions have no bearing on the discussion of the anonymity of the schemes.

Galbraith and Mao proposed a new security notion of undeniable and confirmer signatures named "anonymity" in [17]. We say that an undeniable or confirmer signature scheme provides anonymity when it is infeasible to determine which user generated the message-signature pair. Informally, this security property is as follows. Imagine a system with n users and suppose an adversary is given a valid message-signature pair and is asked to determine which user generated the signature. By running signature confirmation or denial protocols with a given user (or their designated confirmer) one can determine whether or not the user generated the signature. An undeniable or confirmer signature scheme has the anonymity property if it is infeasible to determine whether a user is or is not the signer of the message without interacting with that user or with the $n-1$ other users with given message-signature pair.

We slightly modify the definition of anonymity in [17] in order to put a common key generation into it explicitly.

Definition 9 ([17])**.** *Let $\mathcal{SIG} = (\text{CGEN}, \text{KGEN}, \text{SIGN}, \text{CONF}, \text{DENY})$ 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 $\text{SIGN}_{sk_0}, \text{SIGN}_{sk_1}$ and A can execute confirmation and denial protocols $\text{CONF}_{sk_0}, \text{CONF}_{sk_1}, \text{DENY}_{sk_0}, \text{DENY}_{sk_1}$ on any message-signature pair. However, A_2 cannot execute any one of $\text{CONF}_{sk_0}, \text{CONF}_{sk_1}, \text{DENY}_{sk_0}$, and 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:*

$$
\begin{array}{l}
\texttt{Experiment } \mathbf{Exp}^{\text{Anonym-}b}_{\mathcal{SIG},A}(k) \\
\quad I \leftarrow \text{CGEN}(1^k);\ (pk_0, sk_0) \leftarrow \text{KGEN}(I);\ (pk_1, sk_1) \leftarrow \text{KGEN}(I) \\
\quad (m, \mathsf{si}) \leftarrow A_1(pk_0, pk_1);\ \sigma \leftarrow \text{SIGN}_{sk_b}(m);\ d \leftarrow A_2(m, \sigma, \mathsf{si}) \\
\quad \texttt{return } d
\end{array}
$$

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.

In [17], Galbraith and Mao pointed out that the RSA-based scheme by Gennaro, Krawczyk and Rabin [19] does not provide anonymity, and proposed the scheme with expanding which provides anonymity. See Appendix C for details.

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 safe prime (i.e. $(p-1)/2$ and $(q-1)/2$ are also prime)* [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:*

$\textsc{Sign}(m)$
$r_1, r_2 \xleftarrow{R} \{0,1\}^{k_0}$
$\bar{m}_1 \leftarrow \textsc{Sign2}(m, r_1)$; $t_1 \xleftarrow{R} \{c \in \mathbb{Z}_N \mid c^2 = \pm\bar{m}_1 \pmod N\}$; $s_1 \leftarrow (t_1)^d \bmod N$
$\bar{m}_2 \leftarrow \textsc{Sign2}(m, r_2)$; $t_2 \xleftarrow{R} \{c \in \mathbb{Z}_N \mid c^2 = \pm\bar{m}_2 \pmod N\}$; $s_2 \leftarrow (t_2)^d \bmod N$
$s \leftarrow \texttt{ChooseAndShift}(s_1, s_2)$
`if` $(s \bmod N = s_1)$ $r \leftarrow r_1$ `else` $r \leftarrow r_2$
`return` (s, r)

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 [18] by Galbraith, Mao, and Paterson. Then, we transform them to corresponding non-interactive designated verifier proofs by the method of Jakobsson, Sako, and Impagliazzo [21]* [2]*. The equivalence class of this scheme is $EC(m, (s, r), pk) = \{(m, (\pm s' \pm uN, r)) \mid s' = s \bmod N \wedge u \in \{0, 1, 2, \ldots, \lfloor (2^k - s')/N \rfloor\}\}$.*

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 [19] 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 [19] is more efficient than those by Galbraith, Mao, and Paterson [18]. Therefore, it seems better to use the protocols in [19]. However, if we use the protocols in [19], 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 [18] constructed different scheme where there is no restriction for the RSA moduli.

To obtain the security result it is necessary that executions of the confirm and deny protocol can be simulated in the random oracle model. This is not possible with interactive proofs so we must use non-interactive proofs. To maintain the security of the system, it is necessary to use non-interactive designated verifier proofs [21].

[1] We need this restriction for proving anonymity.
[2] These proof transcripts must be encrypted when sent to the verifier if anonymity is to be preserved.

	Sampling Twice	Expanding [17]	Repeating
# of mod. exp. to sign (average / worst)	2 / 2	1 / 1	1.5 / k_1
# of computation of square roots (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)	$k_0 + k + 5$ / $k_0 + k + 5$	$k_0 + k + 2$ / $k_0 + k + 2$	$1.5(k_0 + 2)$ / $k_1(k_0 + 2)$

Figure 4: The comparison of the undeniable and confirmer signature schemes

5.3 Analysis

We compare the four schemes with sampling twice, expanding, and repeating.
Security. Galbraith and Mao [17] proved that their scheme provides anonymity in the random oracle model under the assumption that the composite decision Diffie-Hellman problem is hard.

Definition 11 (the composite decision Diffie-Hellman problem)**.** *Let N be a product of two safe primes (i.e. $N = pq$ where p, q, $p' = (p-1)/2$, $q' = (q-1)/2$ are prime). Consider the two sets*

$$\mathcal{T} = \{(g, h, u, v) \in (\mathbb{Z}_N^*)^4 | \mathrm{ord}(g) = \mathrm{ord}(h) = 2p'q',\ h \in \langle g \rangle,\ \langle g, v \rangle = \mathbb{Z}_N^*\}$$

and

$$\mathcal{T}_{\mathrm{CDDH}} = \{(g, h, u, v) \in \mathcal{T} | h = g^d \pmod N \text{ for some } d \text{ coprime to } \phi(N), \\ v = \alpha u^d \pmod N \text{ for some } \alpha \in \mathbb{Z}_N^* \text{ of order } 2\}$$

with the uniform distribution on each. We say that the composite decision Diffie-Hellman problem is hard if it is infeasible to distinguish these two distributions.

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 [17].).
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 4. We assume that N is uniformly distributed in $(2^{k-1}, 2^k)$.

6 Ring Signature

6.1 Definitions

In [25], 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 12 (Ring Signature [25])**.** *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**$(m, P_1, P_2, \ldots, P_r, s, S_s)$ *which produces a ring signature* σ *for the message* m*, given the public keys* $P_1, P_2, \ldots, P_r$ *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 [25], Rivest, Shamir, and Tauman proposed a combining function based on a symmetric encryption scheme E modeled by a (keyed) random permutation

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

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 $C_{k,v}(y_1, \ldots, y_r) = z$ by using the following equation:

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

By using this function, Rivest, Shamir, and Tauman [25] proposed the scheme with expanding, and Hayashi, Okamoto, and Tanaka [20] also provided the scheme with RSACD. See Appendix D for details.

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 13. *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$(m, P_1, P_2, \ldots, P_r, s, S_s)$
  `for each` $i \in \{1, \ldots, s-1, s+1, \ldots, r\}$ `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.` $C_{h(m),v}(y_1, \ldots, y_r) = v$
  `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` $\sigma = (P_1, P_2, \ldots, P_r, v, (x_1, c_1), (x_2, c_2), \ldots, (x_r, c_r))$

	Sampling Twice	Expanding [25]	RSACD [20]	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	$1.5k(r-1)+k-1$ / $k^2(r-1)+k-1$

Figure 5: The comparison of the ring signature schemes ($|N_i| = k$)

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* $z \leftarrow C_{h(m),v}(y_1, \ldots, y_r)$. *It returns* `valid` *if and only if* $z = v$.

6.3 Analysis

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

Security. Rivest, Shamir, and Tauman [25] 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 [20] 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 $C_{h(m),v}(y_1, \ldots, y_r) = v$ 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 [25].).

Recently, Bresson, Stern, and Szydlo [6] improved the Rivest–Shamir–Tauman scheme. 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 [25]. 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 5. 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.

References

[1] Abadi, M., and Rogaway, P. Reconciling Two Views of Cryptography (The Computational Soundness of Formal Encryption). In *Proceedings of the First IFIP International Conference on Theoretical Computer Science* (Sendai, Japan, August 2000), vol. 1872 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 3–22.

[2] Bellare, M., Boldyreva, A., Desai, A., and Pointcheval, D. Key-Privacy in Public-Key Encryption. In Boyd [5], pp. 566–582. Full version of this paper, available via `http://www-cse.ucsd.edu/users/mihir/`.

[3] Bellare, M., and Rogaway, P. Optimal Asymmetric Encryption – How to Encrypt with RSA. In De Santis [12], pp. 92–111.

[4] Bellare, M., and Rogaway, P. The Exact Security of Digital Signatures: How to Sign with RSA and Rabin. In Maurer [23], pp. 339–416.

[5] Boyd, C., Ed. *Advances in Cryptology – ASIACRYPT 2001* (Gold Coast, Australia, December 2001), vol. 2248 of *Lecture Notes in Computer Science*, Springer-Verlag.

[6] Bresson, E., Stern, J., and Szydlo, M. Threshold Ring Signatures and Applications to Ad-hoc Groups. In *Advances in Cryptology – CRYPTO 2002* (Santa Barbara, California, USA, August 2002), M. Yung, Ed., vol. 2442 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 465–480.

[7] Camenisch, J., and Lysyanskaya, A. Efficient Non-Transferable Anonymous Multi-Show Credential System with Optional Anonymity Revocation. In *Advances in Cryptology – EUROCRYPT 2001* (Innsbruck, Austria, May 2001), B. Pfitzmann, Ed., vol. 2045 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 93–118.

[8] Camenisch, J., and Michels, M. Confirmer Signature Schemes Secure against Adaptive Adversaries. In *Advances in Cryptology – EUROCRYPT 2000* (Bruges, Belgium, May 2000), B. Preneel, Ed., vol. 1807 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 243–258.

[9] Chaum, D. Zero-Knowledge Undeniable Signatures. In *Advances in Cryptology – EUROCRYPT '90* (Aarhus, Denmark, May 1990), I. Damgård, Ed., vol. 473 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 458–464.

[10] Chaum, D. Designated Confirmer Signatures. In De Santis [12], pp. 86–91.

[11] Chaum, D., and Antwerpen, H. V. Undeniable Signatures. In *Advances in Cryptology – CRYPTO '89* (Santa Barbara, California, USA, August 1989), G. Brassard, Ed., vol. 435 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 212–217.

[12] De Santis, A., Ed. *Advances in Cryptology – EUROCRYPT '94* (Perugia, Italy, May 1994), vol. 950 of *Lecture Notes in Computer Science*, Springer-Verlag.

[13] Desai, A. The Security of All-or-Nothing Encryption: Protecting against Exhaustive Key Search. In *Advances in Cryptology – CRYPTO 2000* (Santa Barbara, California, USA, August 2000), M. Bellare, Ed., vol. 1880 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 359–375.

[14] Desmedt, Y. Securing traceability of ciphertexts: Towards a secure software escrow scheme. In *Advances in Cryptology – EUROCRYPT '95* (Saint-Malo, France, May 1995), L. C. Guillou and J.-J. Quisquater, Eds., vol. 921 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 147–157.

[15] Fischlin, M. Pseudorandom Function Tribe Ensembles Based on One-Way Permutations. In *Advances in Cryptology – EUROCRYPT '99* (Prague, Czech Republic, May 1999), J. Stern, Ed., vol. 1592 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 432–445.

[16] Fujisaki, E., Okamoto, T., Pointcheval, D., and Stern, J. RSA-OAEP is Secure under the RSA Assumption. In *Advances in Cryptology – CRYPTO 2001* (Santa Barbara, California, USA, August 2001), J. Kilian, Ed., vol. 2139 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 260–274.

[17] Galbraith, S. D., and Mao, W. Invisibility and Anonymity of Undeniable and Confirmer Signatures. In *Topics in Cryptology – CT-RSA 2003* (San Francisco, CA, USA, April 2003), M. Joye, Ed., vol. 2612 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 80–97.

[18] Galbraith, S. D., Mao, W., and Paterson, K. G. RSA-based Undeniable Signatures for General Moduli. In *Topics in Cryptology – CT-RSA 2002* (San Jose, CA, USA, February 2002), B. Preneel, Ed., vol. 2271 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 200–217.

[19] Gennaro, R., Krawczyk, H., and Rabin, T. RSA-based Undeniable Signatures. In *Advances in Cryptology – CRYPTO '97* (Santa Barbara, California, USA, August 1997), B. S. Kaliski, Jr., Ed., vol. 1294 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 132–149.

[20] Hayashi, R., Okamoto, T., and Tanaka, K. An RSA Family of Trap-door Permutations with a Common Domain and its Applications. In *Public Key Cryptography – PKC 2004* (Singapore, March 2004), F. Bao, R. H. Deng, and J. Zhou, Eds., vol. 2947 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 291–304.

[21] Jakobsson, M., Sako, K., and Impagliazzo, R. Designated Verifier Proofs and their Applications. In Maurer [23], pp. 143–154.

[22] 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, February 1996), pp. 114–127.

[23] Maurer, U., Ed. *Advances in Cryptology – EUROCRYPT '96* (Saragossa, Spain, May 1996), vol. 1070 of *Lecture Notes in Computer Science*, Springer-Verlag.

[24] Michels, M., and Stadler, M. Generic Constructions for Secure and Efficient Confirmer Signature Schemes. In *Advances in Cryptology – EUROCRYPT '98* (Espoo, Finland, May 1998), K. Nyberg, Ed., vol. 1403 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 406–421.

[25] Rivest, R. L., Shamir, A., and Tauman, Y. How to Leak a Secret. In Boyd [5], pp. 552–565.

[26] Sako, K. An Auction Protocol Which Hides Bids of Losers. In *Public Key Cryptography – PKC 2000* (Melbourne, Victoria, Australia, January 2000), H. Imai and Y. Zheng, Eds., vol. 1751 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 422–432.

A The Schemes Previously Proposed for Encryption

A.1 Encryption with Repeating by Bellare, Boldyreva, Desai, and Pointcheval

In [2], Bellare, Boldyreva, Desai, and Pointcheval proposed an RSA-based encryption scheme which is secure in the sense of IK-CCA. It is RSA-RAEP which is a variant of RSA-OAEP. Since their variant chooses N from $(2^{k-1}, 2^k)$, it simply repeats the ciphertext computation, each time using new coins, until the ciphertext y satisfies $y < 2^{k-1}$.

Definition 14 (RSA-RAEP [2])**.** *RSA-RAEP* $= (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ *is as follows. 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 *with security parameter* k, *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.

```
Algorithm E^{G,H}_{pk}(x)
  ctr = −1
  repeat
    ctr ← ctr + 1
    r ←R {0,1}^{k0}
    s ← (x||0^{k1}) ⊕ G(r);  t ← r ⊕ H(s)
    v ← (s||t)^e mod N
  until ((v < 2^{k−1}) ∨ (ctr = k1))
  if (ctr = k1) y ← 1||0^{k0+k1}||x
  else y ← 0||v
  return y
```

```
Algorithm D^{G,H}_{sk}(y)
  b ← [y]^1;  v ← [y]_{k0+k1+n}
  if (b = 1)
    w ← [v]^{k0+k1};  x ← [v]_n
    if (w = 0^{k0+k1}) z ← x else z ←⊥
  else
    s ← [v^d mod N]^{n+k1};  t ← [v^d mod N]_{k0}
    r ← t ⊕ H(s)
    x ← [s ⊕ G(r)]^n;  p ← [s ⊕ G(r)]_{k1}
    if (p = 0^{k1}) z ← x else z ←⊥
  return z
```

Bellare, Boldyreva, Desai, and Pointcheval proved that RSA-RAEP is secure in the sense of IK-CCA and IND-CCA2 in the random oracle model assuming RSA is one-way.

A.2 Encryption with RSACD by Hayashi, Okamoto, and Tanaka

In [20], Hayashi, Okamoto, and Tanaka proposed an RSA-based encryption scheme. It uses RSACD instead of RSA. We describe their scheme.

Definition 15. *The common-key generation algorithm* $\mathcal{G}$, *the key generation algorithm* $\mathcal{K}$, *and the oracles* G *and* H *are the same as those for RSA-RAEP. The other algorithms are described as follows. Note that the valid ciphertext* y *satisfies* $y \in [0, 2^k)$ *and* $(y \bmod N) \in \mathbb{Z}_N^*$.

```
Algorithm E^{G,H}_{pk}(x)
  r ←R {0,1}^{k0}
  s ← (x||0^{k1}) ⊕ G(r);  t ← r ⊕ H(s)
  v ← f^{RSACD}_{N,e,k}(s||t)
  return y
```

```
Algorithm D^{G,H}_{sk}(y)
  s ← [g^{RSACD}_{N,d,k}(y)]^{n+k1};  t ← [g^{RSACD}_{N,d,k}(y)]_{k0}
  r ← t ⊕ H(s)
  x ← [s ⊕ G(r)]^n;  p ← [s ⊕ G(r)]_{k1}
  if (p = 0^{k1}) z ← x else z ←⊥
  return z
```

Hayashi, Okamoto, and Tanaka proved that their scheme is secure in the sense of IK-CCA and IND-CCA2 in the random oracle model assuming RSACD is one-way.

B Proof of Theorem 2

We first describe the RSA partial inverting algorithm M using a CCA-adversary A attacking anonymity of our encryption scheme. M is given $pk = (N, e, k)$ and a point $y \in \mathbb{Z}_N^*$ where $|y| = k = n + k_0 + k_1$. Let $sk = (N, d, k)$ be the corresponding secret key. The algorithm is trying to find the $n + k_1$ most significant bits of the e-th root of y modulo N.

1) M picks a bit $\mu \stackrel{R}{\leftarrow} \{0,1\}$ and sets $Y \leftarrow y + \mu N$. If $Y \geq 2^k$ then outputs Fail and halts; else it continues.

2) M runs the key generation algorithm of RSA with security parameter k to obtain $pk' = (N', e', k)$ and $sk' = (N', d', k)$. Then it picks a bit $b \stackrel{R}{\leftarrow} \{0,1\}$, sets $pk_b \leftarrow (N, e)$ and $pk_{1-b} \leftarrow (N', e')$. If the above y does not satisfy $y \in (\mathbb{Z}_{N_0}^* \cap \mathbb{Z}_{N_1}^*)$ then M outputs Fail and halts; else it continues.

3) M initializes for lists, called G-list, H-list, Y_0-list, and Y_1-list to empty. It then runs A as follows. Note that M simulates A's oracles G, H, $\mathcal{D}_{sk_0}$, and $\mathcal{D}_{sk_1}$ as described below.

   3-1) M runs $A_1(pk_0, pk_1)$ and gets (x, si) which is the output of A_1.

   3-2) M runs $A_2(Y, \mathsf{si})$ and gets a bit $d \in \{0,1\}$ which is the output of A_2.

4) M chooses a random element on the H-list and outputs it as its guess for the $n + k_1$ most significant bits of the e-th root of y modulo N.

M simulates the random oracles G and H, and the decryption oracle as follows:

- When A makes an oracle query g to G, then for each (h, H_h) on the H-list, M builds $z = h||(g \oplus H_h)$, and computes $y_{h,g,0} = z^{e_0} \bmod N_0$ and $y_{h,g,1} = z^{e_1} \bmod N_1$. For $i \in \{0,1\}$, M checks whether $y = y_{h,g,i}$. If for some h and i such a relation holds, then we have inverted y under pk_i, and we can still correctly simulate G by answering $G_g = h \oplus (x||0^{k_1})$. Otherwise, M outputs a random value G_g of length $n + k_1$. In both cases, M adds (g, G_g) to the G-list. Then, for all h, M checks if the k_1 least significant bits of $h \oplus G_g$ are all 0. If they are, then it adds $y_{h,g,0}$ and $y_{h,g,1}$ to the Y_0-list and the Y_1-list respectively.

- When A makes an oracle query h to H, M provides A with a random string H_h of length k_0 and adds (h, H_h) to the H-list. Then for each (g, G_g) on the G-list, M builds $z = h||(g \oplus H_h)$, and computes $y_{h,g,0} = z^{e_0} \bmod N_0$ and $y_{h,g,1} = z^{e_1} \bmod N_1$. M checks if the k_1 least significant bits of $h \oplus G_g$ are all 0. If they are, then it adds $y_{h,g,0}$ and $y_{h,g,1}$ to the Y_0-list and the Y_1-list respectively.

- When for $i \in \{0,1\}$, A makes an oracle query $y' \in \{0,1\}^k$ to $\mathcal{D}_{sk_i}$, M checks if there exists some $y_{h,g,i}$ in the Y_i-list such that $y' \bmod N_i = y_{h,g,i}$. If there is, then it returns the n most significant bits of $h \oplus G_g$ to A. Otherwise it returns $\perp$ (indicating that y' is an invalid ciphertext).

Now, we analyze the advantage of M. In the following, we consider the experiment where M does not output Fail in the first step. In this experiment, we can consider the distributions of N, e, and Y as $((N, e, k), (N, d, k)) \leftarrow K(k)$; $Y \stackrel{R}{\leftarrow} S[N]$ where $\mathcal{K}$ is the key generation algorithm of RSA and $S[N] = \{Y' \mid Y' \in [0, 2^k) \wedge (Y' \bmod N) \in \mathbb{Z}_N^*\}$.

For $i \in \{0,1\}$, let $w_i = y^{d_i} \bmod N_i$, $s_i = [w_i]^{n+k_1}$, and $t_i = [w_i]_{k_0}$. Let r_i be the random variable $t_i \oplus H(s_i)$. We consider the following events.

- FBad denotes the event that
  - A G-oracle query r_0 was made by A_1 in step 3-1, and $G_{r_0} \neq s_0 \oplus (x||0^{k_1})$, or
  - A G-oracle query r_1 was made by A_1 in step 3-1, and $G_{r_1} \neq s_1 \oplus (x||0^{k_1})$.
- GBad denotes the event that
  - A G-oracle query r_0 was made by A_2 in step 3-2, and at the point in time that it was made, the H-oracle query s_0 was not on the H-list, and $G_{r_0} \neq s_0 \oplus (x||0^{k_1})$, or
  - A G-oracle query r_1 was made by A_2 in step 3-2, and at the point in time that it was made, the H-oracle query s_1 was not on the H-list, and $G_{r_1} \neq s_1 \oplus (x||0^{k_1})$.
- DBad denotes the event that
  - A $\mathcal{D}_{sk_0}$ query is not correctly answered, or
  - A $\mathcal{D}_{sk_1}$ query is not correctly answered.
- $\mathsf{G} = \neg\mathsf{FBad} \wedge \neg\mathsf{GBad} \wedge \neg\mathsf{DBad}$.

We let $\Pr[\cdot]$ denote the probability distribution in the game defining advantage, and $\Pr_0[\cdot]$ denote the probability distribution in the simulated game where M does not output Fail in the first step. We introduce the following additional events:

- YBad denotes the event that $y \in (\mathbb{Z}^*_{N_0} \cap \mathbb{Z}^*_{N_1})$.
- FAskS denotes the event that H-oracle query s_0 or s_1 was made by A_1 in step 3-1.
- AskR denotes the event that (r_0, G_{r_0}) or (r_1, G_{r_1}) is on the G-list at the end of step 3-2.
- AskS denotes the event that (s_0, H_{s_0}) or (s_1, H_{s_1}) is on the H-list at the end of step 3-2.

Let $\Pr_1[\cdot]$ denote the probability distribution in the simulated game where M does not output Fail in the first step and $\neg\mathsf{YBad}$ occurs.

We can bound $\Pr_1[\mathsf{AskS}]$ in a similar way as in the proof of anonymity for RSA-RAEP [2], and we have

$$\Pr_1[\mathsf{AskS}] \geq \frac{1}{2} \cdot \Pr_1[\mathsf{AskR} \wedge \mathsf{AskS}|\neg\mathsf{DBad}] \cdot \Pr_1[\neg\mathsf{DBad}|\neg\mathsf{AskS}].$$

We next bound $\Pr_1[\mathsf{AskR}\wedge\mathsf{AskS}|\neg\mathsf{DBad}]$ and $\Pr_1[\neg\mathsf{DBad}|\neg\mathsf{AskS}]$. Let $\Pr_2[\cdot]$ denote the probability distribution in the simulated game where M does not output Fail in the first step and $\neg\mathsf{DBad}\wedge\neg\mathsf{YBad}$ occurs.

The proofs of the following lemmas are similar to those for RSA-RAEP.

Lemma 1.

$$\Pr_2[\mathsf{AskR} \wedge \mathsf{AskS}] \geq \frac{\epsilon}{2} \cdot \left(1 - 2q_{\mathrm{gen}} \cdot 2^{-k_0} - 2q_{\mathrm{hash}} \cdot 2^{-n-k_1}\right) - 2q_{\mathrm{gen}} \cdot 2^{-k}.$$

Lemma 2.

$$\Pr_1[\mathsf{DBad}|\neg\mathsf{AskS}] \leq q_{\mathrm{dec}} \cdot \left(2 \cdot 2^{-k_1} + (2q_{\mathrm{gen}} + 1) \cdot 2^{-k_0}\right).$$

By applying Lemmas 1 and 2, we have

$$\begin{aligned}\Pr_1[\mathsf{AskS}] &\geq \frac{1}{2}\cdot\left(\frac{\epsilon}{2}\cdot\left(1-\frac{2q_{\mathrm{gen}}}{2^{k_0}}-\frac{2q_{\mathrm{hash}}}{2^{n+k_1}}\right)-\frac{2q_{\mathrm{gen}}}{2^k}\right)\cdot\left(1-q_{\mathrm{dec}}\cdot\left(\frac{2}{2^{k_1}}+\frac{2q_{\mathrm{gen}}+1}{2^{k_0}}\right)\right)\\ &\geq \frac{\epsilon}{4}\cdot\left(\frac{2q_{\mathrm{gen}}+q_{\mathrm{dec}}+2q_{\mathrm{gen}}q_{\mathrm{dec}}}{2^{k_0}}+\frac{2q_{\mathrm{gen}}}{2^{k_1}}+\frac{2q_{\mathrm{hash}}}{2^{k-k_0}}\right)-\frac{q_{\mathrm{gen}}}{2^k}.\end{aligned}$$

Assuming that $Y < 2^k$ and $\neg\mathsf{YBad}$, we have by the random choice of b and symmetry, that the probability of M outputting s is at least $\frac{1}{2q_{\mathrm{hash}}}\cdot\Pr_1[\mathsf{AskS}]$.

We next bound the probabilities that Y is in the good range and that $\neg\mathsf{YBad}$ occurs.

Lemma 3.

$$\Pr[Y > 2^k] \leq \frac{1}{2} \quad \textit{and} \quad \Pr_0[\mathsf{YBad}] \leq \frac{2}{2^{k/2-3}-1}.$$

Proof of Lemma 3. We first bound $\Pr[Y > 2^k]$. Since $Y = y + \mu N$, $y \in \mathbb{Z}_N^*$, and $\mu \xleftarrow{R} \{0,1\}$, we have

$$\Pr[Y > 2^k] \leq \Pr[\mu = 1] = \frac{1}{2}.$$

We next bound $\Pr_0[\mathsf{YBad}]$. Let $N = pq$ and $N' = p'q'$. Note that $2^{\lceil k/2\rceil-1} < p, q, p', q' < 2^{\lceil k/2\rceil}$ and $2^{k-1} < N, N' < 2^k$. Since $\phi(N) \leq |S[N]| \leq 2^k$, we have

$$\Pr_0[\mathsf{YBad}] = \Pr[Y \xleftarrow{R} S[N] :\ Y \notin S[N']] \leq \frac{|S[N]| - |S[N']|}{|S[N]|} \leq \frac{2^k - |S[N']|}{\phi(N)}.$$

Furthermore, we have

$$\begin{aligned}2^k - |S[N']| &= \left|\{Y' | Y' \in [0, 2^k) \wedge (Y' \bmod N') \notin \mathbb{Z}_{N'}^*\}\right|\\ &\leq \left|\{Y' | Y' \in [0, 2N') \wedge (Y' \bmod N') \notin \mathbb{Z}_{N'}^*\}\right|\\ &= 2 \times \left|\{Y' | Y' \in [0, N') \wedge Y' \notin \mathbb{Z}_{N'}^*\}\right|\\ &= 2(N' - \phi(N')).\end{aligned}$$

Therefore, we can bound $\Pr_0[\mathsf{YBad}]$ as

$$\begin{aligned}\Pr_0[\mathsf{YBad}] &\leq \frac{2^k - |S[N']|}{\phi(N)} \leq \frac{2(N' - \phi(N'))}{\phi(N)} = \frac{2(p' + q' - 1)}{N - p - q + 1} \leq \frac{2(p' + q')}{N - p - q}\\ &\leq \frac{2(2^{\lceil k/2\rceil} + 2^{\lceil k/2\rceil})}{2^{k-1} - 2^{\lceil k/2\rceil} - 2^{\lceil k/2\rceil}} = \frac{2(1+1)}{2^{k-1-\lceil k/2\rceil} - 1 - 1} \leq \frac{4}{2^{k/2-2} - 2} = \frac{2}{2^{k/2-3}-1}.\end{aligned}$$

□

We have that

$$\mathbf{Adv}^{\theta\text{-pow-fnc}}_{\mathsf{RSA},B}(k) \geq (1 - \Pr[Y > 2^k]) \cdot (1 - \Pr_0[\mathsf{YBad}]) \cdot \left(\frac{\Pr_1[\mathsf{AskS}]}{2q_{\mathrm{hash}}}\right).$$

Substituting the bounds for the above probabilities and re-arranging the terms, we get the claimed result.

Finally, we estimate the time complexity of M. It is the time complexity of A plus the time for simulating the random oracles. In the random oracle simulation, for each pair $((g, G_g), (h, H_h))$, it is sufficient to compute $y_{h,g,0} = z^{e_0} \bmod N_0$ and $y_{h,g,1} = z^{e_1} \bmod N_1$. Therefore, the time complexity of M is that of A plus $q_{\mathrm{gen}} \cdot q_{\mathrm{hash}} \cdot O(k^3)$.

C The Scheme Previously Proposed for Undeniable and Confirmer Signature

C.1 Attacks on Anonymity

In [19], Gennaro, Krawczyk and Rabin described an undeniable/confirmer signature scheme based on RSA. In their case the signature for a message m is s where $s = \bar{m}^d \bmod N$ and $\bar{m}$ is a one-way encoding. The signature may be verified by proving that $s^e = \bar{m} \pmod N$ where the verification exponent e is known to the signer/confirmer. This scheme requires that the moduli be products of safe primes. Later the scheme was generalized to use arbitrary RSA moduli [18]. To handle adaptive attacks on anonymity it is clear that the one-way encoding must also be randomized. Hence, a signature becomes a pair (r, s) where r is random and $s = H(m, r)^d \pmod N$ where $H(m, r)$ is the randomized one-way encoding (such as PSS [4]).

In [17], Galbraith and Mao pointed out the Gennaro–Krawczyk–Rabin scheme does not provide anonymity. They showed the following attacks:

Jacobi Symbols Attack Since d is odd it follows that the Jacobi symbols $\left(\frac{s}{N}\right)$ and $\left(\frac{H(m,r)}{N}\right)$ are equal. Hence, given a pair $(H(m, r), s)$ and a user's public key N, if $\left(\frac{s}{N}\right) \neq \left(\frac{H(m,r)}{N}\right)$then the signature is not valid for that user. This shows that the scheme does not have anonymity.

Signature Length Attack A simple observation that seems to be folklore is that standard RSA signature does not provide anonymity, even when all moduli in the system have the same length. Suppose an adversary knows that the signature s is created under one of two keys (N_0, d_0) or (N_1, d_1) (length of N_0 and N_1 are k), and suppose $N_0 \leq N_1$. If $s \geq N_0$ then the adversary knows it was created under (N_1, d_1).

C.2 Undeniable and Confirmer Signature with Expanding by Galbraith and Mao

In [17], Galbraith and Mao proposed a new RSA-based scheme. We describe their scheme.

Definition 16 ([17])**.** *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 the safe prime (i.e. $(p-1)/2$ and $(q-1)/2$ are also prime). 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.*

Sign(m)	Sign2(m, r)
$r \xleftarrow{R} \{0,1\}^{k_0}$	$w \leftarrow G_0(m\|\|r)$
$\bar{m} \leftarrow$ Sign2(m, r)	$r^* \leftarrow G_1(w) \oplus r$
$t \xleftarrow{R} \{c \in \mathbb{Z}_N \mid c^2 = \pm\bar{m} \pmod N\}$	$M \leftarrow 0\|\|w\|\|r^*\|\|G_2(w)$
$s \leftarrow t^d \bmod N$	$\bar{m} \leftarrow M$
$u \xleftarrow{R} \{0, 1, \ldots, \lfloor(2^{2k} - s)/N\rfloor\}$	`while` $\left(\left(\frac{\bar{m}}{N}\right) \neq 1\right)$ `repeat` $\bar{m} \leftarrow F(\bar{m})$
$\hat{s} \leftarrow s + uN$	`return` $\bar{m}$
`return` $(\hat{s}, r)$	

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 $\hat{s}^{2e} = \pm$Sign2$(m, r) \pmod N$ (resp. $g = h^e \pmod N$ and $\hat{s}^{2e} \neq \pm$Sign2$(m, r) \pmod N$). Note that $\hat{s} = s + uN = s \pmod N$ and all users can*

compute SIGN2(m, r) *given* m, r, *and* N. *The equivalence class of this scheme is* $EC(m, (\hat{s}, r), pk) = \{(m, (\pm s \pm uN, r)) | s = \hat{s} \bmod N,\ u \in \{0, 1, 2, \ldots, \lfloor (2^{2k} - s)/N \rfloor\}\}$.

Since using a Blum integer N, for every $\bar{m} \in \mathbb{Z}_N^*$ with $\left(\frac{\bar{m}}{N}\right) = 1$, it follows that either $\bar{m}$ or $-\bar{m}$ is a square. One can compute square-root and randomly chooses t from four possibilities in step 3. Since $\left(\frac{t}{N}\right)$ is not fixed, their scheme prevents the Jacobi symbols attack. In step 5 and 6, it extends signatures of length k to be bit-strings of length $2k$. Since $0 \leq \hat{s} < 2^{2k}$ and $\hat{s}$ is indistinguishable from a random $2k$-bit string for any N whose length is k, their scheme prevents the signature length attack (See also [14].).

It is clear that if a message-signature pair $(m, (\hat{s}, r))$ is valid for $pk = (N, g, h)$ then $(m, (\pm s \pm uN, r))$ is also valid where $s = \hat{s} \bmod N$ and $u \in \{0, 1, \ldots, \lfloor (2^{2k} - s)/N \rfloor\}$. Thus, Galbraith and Mao defined the equivalence class for their scheme as $EC(m, (\hat{s}, r), pk) = \{(m, (\pm s \pm uN, r)) | s = \hat{s} \bmod N,\ u \in \{0, 1, \ldots, \lfloor (2^{2k} - s)/N \rfloor\}\}$.

Galbraith and Mao proved that their scheme provides anonymity in the random oracle model under the assumption that the composite decision Diffie-Hellman problem is hard. 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.

D The Schemes Previously Proposed for Ring Signature

D.1 Ring Signature with Expanding by Rivest, Shamir, and Tauman

In [25], Rivest, Shamir, and Tauman constructed ring signature schemes in which all the ring member use RSA as their individual signature schemes. Each user can uses the RSA moduli whose lengths are different from other users.

Definition 17 ([25])**.** *Let* ℓ, k, *and* b *be security parameters. Let* E *be a symmetric encryption scheme over* $\{0,1\}^b$ *using* ℓ*-bit keys and* h *be a hash function which maps arbitrary strings to* ℓ*-bit strings. They use* h *to make a key for* E. *Each user has an RSA public key* $P_i = (N_i, e_i, k_i)$ *and secret key* $S_i = (N_i, d_i, k_i)$ *where* $k_i \geq k$ *by running the key generation algorithm of* RSA. *Let* r *be a number of ring member. We define the extended trap-door permutation* g_i *over* $\{0,1\}^b$ *as follows: for any* b*-bit input* x_i *define nonnegative integers* q_i *and* r_i *so that* $x_i = q_i N_i + r_i$ *and* $0 \leq r_i < N_i$. *Then*

$$g_i(x_i) = \begin{cases} q_i N_i + (r_i^{e_i} \bmod N_i) & \texttt{if } (q_i + 1)N_i \leq 2^b \\ x_i & \texttt{otherwise.} \end{cases}$$

The signing algorithm is as follows:

ring-sign$(m, P_1, P_2, \ldots, P_r, s, S_s)$
  `for each` $i \in \{1, \ldots, s-1, s+1, \ldots, r\}$ `do`
    $x_i \xleftarrow{R} \{0,1\}^b$; $y_i \leftarrow g_i(x_i)$
  $v \xleftarrow{R} \{0,1\}^b$
  `find` y_s `s.t.` $C_{h(m),v}(y_1, \ldots, y_r) = v$
  $x_s \leftarrow g_s^{-1}(y_s)$
  `return` $\sigma = (P_1, P_2, \ldots, P_r, v, x_1, x_2, \ldots, x_r)$

ring-verify(m, σ) *computes* $y_i \leftarrow g_i(x_i)$ *for each* x_i *and* $z \leftarrow C_{h(m),v}(y_1, \ldots, y_r)$. *It returns* `valid` *if and only if* $z = v$.

If b is sufficiently large (e.g. 160 bits larger than any of the N_i), g_i is a one-way trap-door permutation, and Rivest, Shamir, and Tauman proved this scheme is unconditionally signer-ambiguous and existential unforgeable under adaptive chosen message attack in the ideal cipher model assuming RSA is one-way.

D.2 Ring Signature with RSACD by Hayashi, Okamoto, and Tanaka

We describe the scheme with RSACD by Hayashi, Okamoto, and Tanaka [20]. Their scheme is almost the same as the Rivest–Shamir–Tauman scheme. They used $f^{\mathsf{RSACD}}_{N_i,e,k}(\cdot)$ instead of $g_i(\cdot)$.

Definition 18. *The values ℓ, k, E, h, r are the same as those of the Rivest–Shamir–Tauman scheme. Each user has a 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* RSACD *with security parameter k (i.e. the size of N_i is k). The signing algorithm is as follows:*

ring-sign$(m, P_1, P_2, \ldots, P_r, s, S_s)$
  `for each` $i \in \{1, \ldots, s-1, s+1, \ldots, r\}$ `do`
    $x_i \xleftarrow{R} \{0,1\}^k$; $y_i \leftarrow f^{\mathsf{RSACD}}_{N_i,e_i,k}(x_i)$
  $v \xleftarrow{R} \{0,1\}^k$
  `find` y_s `s.t.` $C_{h(m),v}(y_1, \ldots, y_r) = v$
  $x_s \leftarrow g^{\mathsf{RSACD}}_{N_s,d_s,k}(y_s)$
  `return` $\sigma = (P_1, P_2, \ldots, P_r, v, x_1, x_2, \ldots, x_r)$

ring-verify(m, σ) *computes $y_i \leftarrow f^{\mathsf{RSACD}}_{N_i,e_i,k}(x_i)$ for each x_i and $z \leftarrow C_{h(m),v}(y_1, \ldots, y_r)$. It returns* `valid` *if and only if $z = v$.*

Hayashi, Okamoto, and Tanaka proved this scheme is unconditionally signer-ambiguous and existential unforgeable under adaptive chosen message attack in the ideal cipher model assuming RSACD is one-way.