# A New Method and Analysis of Anonymous Public-Key Cryptosystems (Extended Abstract)

Ryotaro Hayashi  Keisuke Tanaka *

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

May 25, 2004

**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. First, we 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 $|N| = k$. Second, in addition to our new technique, we consider three previously known techniques, that is, repeating, expanding, and using RSACD, for anonymity of cryptosystems based on RSA. We construct previously unproposed schemes for encryption, undeniable and confirmer signature, and ring signature by applying each technique. Then we precisely analyze the properties of these schemes and show their advantage and disadvantage.

**Keywords:** 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." 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 [12]), and made the space of ciphertexts common to each

*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.

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.

Chaum and Antwerpen provided undeniable signature which cannot be verified without the signer's cooperation [9, 7]. 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 [8] 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 [13]. 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 a given message-signature pair. In [13], Galbraith and Mao provided the undeniable and confirmer signature scheme with anonymity. They made the space of signatures common to each user by encoding the message to an $N$-ary representation and applying a standard RSA permutation to the low-order digits where $N$ is a public key for each user. This technique was proposed by Desmedt [11].

Rivest, Shamir, and Tauman [20] 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. As mentioned in [20], 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 [16] recently proposed the RSA family of trap-door permutations with a common domain denoted by RSACD. They showed that the $\theta$-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. They claimed that their schemes have some advantages to the previous schemes. However, their analysis concerning the ring signature scheme is not precise.

### 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.

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

Figure 1: The previous and our proposed schemes

**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 [13], Galbraith and Mao used this technique for the undeniable signature scheme. In [20], 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 $f_{N,e,k}^{\mathsf{RSACD}}$. This function was proposed by Hayashi, Okamoto, and Tanaka [16].

Our first main result is to 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. We also construct previously unproposed schemes for encryption, undeniable and confirmer signature, and ring signature by applying three previously known techniques, that is, repeating, expanding, and using RSACD (See Figure 1.).

Our second main result is to analyze precisely the properties of these schemes and show their advantage and disadvantage.

We briefly overview the results on the comparison in this paper.

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, and the computational costs to encrypt messages and to sign messages in the worst case. However, the number of exponentiations for encryption and signing is two, while those of the other schemes is 1 or 1.5 in the average case. Similar to the scheme with RSACD, in order to obtain the anonymity property, it is necessary for each user to choose a public key with exact the same size.

The scheme with repeating is efficient with respect to the sizes of ciphertexts and signatures, the computational costs to encrypt messages and to sign messages in the average case, and those to decrypt ciphertexts and to verify signatures. However, it is inefficient with respect to the computational costs to encrypt messages and to sign messages in the worst case. In order to obtain the anonymity property, it is necessary for each user to choose a public key with almost the same size.

The scheme with expanding provides anonymity even if each user uses the public key of different length. It is efficient with respect to the computational costs to encrypt messages, to sign messages, to decrypt ciphertexts, and to verify signatures. However, the sizes of ciphertexts and signatures are larger than those of the other schemes.

The scheme with RSACD is efficient with respect to the sizes of ciphertexts and signatures, and the computational costs to encrypt messages and to sign messages. However, it is inefficient with respect to the computational costs to decrypt a ciphertext and to verify a signature. In order to obtain the anonymity property, it is necessary for each user to choose a public key with exact the same size.

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

## 2 Preliminaries

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

**Definition 1 (families of functions [1]).** *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 $k \in \mathbb{N}$ 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 ($\theta$-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:*

$$
\begin{array}{l}
\texttt{Experiment } \mathbf{Exp}^{\theta-\mathrm{pow}-\mathrm{fnc}}_{F,A}(k) \\
\quad (pk, sk) \leftarrow K(k) \\
\quad x_1||x_2 \stackrel{R}{\leftarrow} \mathrm{Dom}_F(pk) \texttt{ where } |x_1| = \lceil \theta \cdot |(x_1||x_2)| \rceil \\
\quad y \leftarrow E_{pk}(x_1||x_2) \\
\quad x'_1 \leftarrow A(pk, y) \texttt{ where } |x'_1| = |x_1| \\
\quad \texttt{for any } x'_2 \texttt{ if } E_{pk}(x'_1||x'_2) = y \texttt{ then return } 1 \\
\quad \texttt{else return } 0
\end{array}
$$

*Here " $||$ " denotes concatenation and " $x_1||x_2 \stackrel{R}{\leftarrow} \mathrm{Dom}_F(pk)$ " is the operation of picking an element $x_1||x_2$ uniformly from $\mathrm{Dom}_F(pk)$. We define the advantages of the adversary via*

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

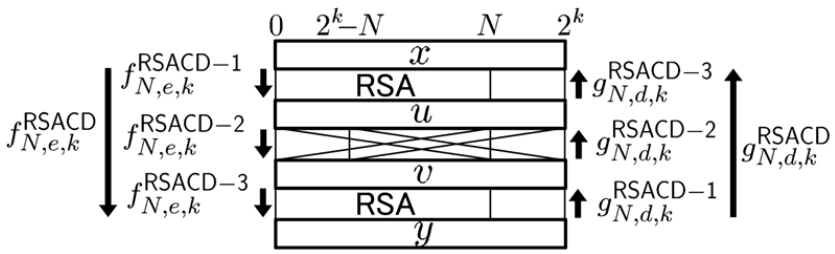


Figure 2: Function $f_{N,e,k}^{\mathsf{RSACD}}$ and $g_{N,d,k}^{\mathsf{RSACD}}$

*where the probability is taken over* $K$, $x_1||x_2 \stackrel{R}{\leftarrow} \mathrm{Dom}_F(pk)$, $E$, *and* $A$. *We say that the family* $F$ *is* $\theta$*-partial one-way if the function* $\mathbf{Adv}_{F,A}^{\theta-\mathrm{pow}-\mathrm{fnc}}(\cdot)$ *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 $\theta$-partial one-wayness coincides with the standard notion of one-wayness. In the following, 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 [1]).** *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* $2^{k-1} < N < 2^k$. *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* $E_{N,e,k}(x) = f_{N,e,k}^{\mathsf{RSA}}(x) = x^e \bmod N$ *and the inversion algorithm* $I_{N,d,k}(y) = g_{N,d,k}^{\mathsf{RSA}}(y) = y^d \bmod N$. *The sampling algorithm returns a random point in* $\mathbb{Z}^*_N$.

Fujisaki, Okamoto, Pointcheval, and Stern [12] showed that the $\theta$-partial one-wayness of RSA is equivalent to the one-wayness of RSA for $\theta > 0.5$.

Hayashi, Okamoto, and Tanaka [16] 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.

**Definition 5 (the RSA family of trap-door permutations with a common domain [16]).** *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* $\{x \,|\, x \in [0, 2^k) \ \wedge \ x \bmod N \in \mathbb{Z}^*_N\}$. *The sampling algorithm returns a random point in* $\mathrm{Dom}_{\mathsf{RSACD}}(N, e, k)$. *The evaluation algorithm* $E_{N,e,k}(x) = f_{N,e,k}^{\mathsf{RSACD}}(x)$ *and the inversion algorithm* $I_{N,d,k}(y) = g_{N,d,k}^{\mathsf{RSACD}}(y)$ *are as follows (See Figure 2.).*

**Function** $f_{N,e,k}^{\mathsf{RSACD}}(x)$
  $u \leftarrow f_{N,e,k}^{\mathsf{RSACD}-1}(x)$; $v \leftarrow f_{N,e,k}^{\mathsf{RSACD}-2}(u)$; $y \leftarrow f_{N,e,k}^{\mathsf{RSACD}-3}(v)$
  **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

Function g^{RSACD}_{N,d,k}(y)
  v ← g^{RSACD-1}_{N,d,k}(y); u ← g^{RSACD-2}_{N,d,k}(v); x ← g^{RSACD-3}_{N,d,k}(u)
  return x

Function g^{RSACD-1}_{N,d,k}(y)
  if (y < N) v ← y^d mod N
  else v ← y
  return v

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

Function g^{RSACD-3}_{N,d,k}(u)
  if (u < N) x ← u^d mod N
  else x ← u
  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 $\theta$-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 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 an algorithm `ChooseAndShift`.

We describe the 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 ChooseAndShift_{N,k}(x_1, x_2)
  if (0 ≤ x_1, x_2 < 2^k − N)
    return { x_1       with probability 1/2
           { x_1 + N   with probability 1/2
  elseif (2^k − N ≤ x_1, x_2 < N)
    return x_1
  else
    y_1 ← min{x_1, x_2}; y_2 ← max{x_1, x_2}
    %%% Note that 0 ≤ y_1 < 2^k − N and 2^k − N ≤ y_2 < N. %%%
    return { y_1       with probability (1/2 + N/2^{k+1}) × 1/2
           { y_1 + N   with probability (1/2 + N/2^{k+1}) × 1/2
           { y_2       with probability 1/2 − N/2^{k+1}
```

Note that $2^{k-1} < N < 2^k$ ensures $2^k - N < N$, $0 < \frac{1}{2} - \frac{N}{2^{k+1}} < 1$, and $0 < \frac{1}{2} + \frac{N}{2^{k+1}} < 1$.

We prove the following theorem.

**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 $\Pr[\texttt{ChooseAndShift}(x_1, x_2) = z] = 1/2^k$ for any $z \in [0, 2^k)$.

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

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

It is easy to see 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

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

□

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 siging) 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, we construct encryption, undeniable and confirmer signature, and ring signature, by applying this technique. We also construct previously unproposed schemes for encryption, undeniable and confirmer signature, and ring signature, by applying three techniques, that is, repeating, expanding, and using RSACD.

# 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 [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 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 [1], a public-key encryption scheme with common-key generation is described as follows.

**Definition 6.**
- *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 7 (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 $A_{\mathrm{cca}}$ has access to the oracles $\mathcal{D}_{sk_0}(\cdot)$ and $\mathcal{D}_{sk_1}(\cdot)$. Note that $\mathsf{si}$ is the state information. It contains $pk_0, pk_1$, and so on. For* atk $\in$ {cpa, cca}*, we consider the following experiments:*

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

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

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

*The scheme $\mathcal{PE}$ is said to be IK-CPA secure (respectively IK-CCA secure) if the function $\mathbf{Adv}^{\mathrm{ik-cpa}}_{\mathcal{PE},A_{\mathrm{cpa}}}(\cdot)$ (resp. $\mathbf{Adv}^{\mathrm{ik-cca}}_{\mathcal{PE},A_{\mathrm{cca}}}(\cdot)$) 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.

### 4.2 Encryption with Sampling Twice

For the comparison of the schemes, we describe the scheme with repeating proposed by Bellare, Boldyreva, Desai, and Pointcheval [1] and that with RSACD proposed by Hayashi, Okamoto, and Tanaka [16] in Appendix A.

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

**Definition 8.** *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 $(y \bmod N) \in \mathbb{Z}^*_N$.*

| | Sampling Twice | Repeating [1] | Expanding | RSACD [16] |
|---|---|---|---|---|
| # of mod. exp. to encrypt (average / worst) | 2 / 2 | 1.5 / $k_1$ | 1 / 1 | 1.5 / 2 |
| # of mod. exp. to decrypt (average / worst) | 1 / 1 | 1 / 1 | 1 / 1 | 1.5 / 2 |
| size of ciphertexts | $k$ | $k$ | $2k$ | $k$ |
| restriction of $N$ for anonymity (with security parameter $k$) | $\|N\| = k$ | $k \leq \|N\| \leq k + \log\log k_1$ | $\|N\| \geq k$ | $\|N\| = k$ |

Figure 3: The comparison of the encryption schemes

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

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

### 4.3 Encryption with Expanding

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

**Definition 9.** *The common-key generation algorithm $\mathcal{G}$, the key generation algorithm $\mathcal{K}$, and the oracles $G$ and $H$ are the same as that for the scheme with sampling twice. The other algorithms are described as follows. Note that the valid ciphertext $y$ satisfies $y \in [0, 2^{2k})$ and $(y \bmod N) \in \mathbb{Z}_N^*$.*

**Algorithm** $\mathcal{E}_{pk}^{G,H}(x)$
- $r \xleftarrow{R} \{0,1\}^{k_0}$
- $s \leftarrow (x \mid\mid 0^{k_1}) \oplus G(r)$
- $t \leftarrow r \oplus H(s)$
- $v \leftarrow (s||t)^e \bmod N$
- $\alpha \xleftarrow{R} \{0, 1, \cdots, \lfloor (2^{2k} - v)/N \rfloor\}$
- $y \leftarrow v + \alpha N$
- **return** $y$

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

### 4.4 Analysis

**Security.** Bellare, Boldyreva, Desai, and Pointcheval 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 one-way. Hayashi, Okamoto, and Tanaka 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 one-way.

However, we can easily see that the scheme with expanding is not secure in the sense of IK-CCA. Since if $c$ is a ciphertext of $m$ for $pk = (N, e, k)$ then $(c \bmod N) + \beta N$ is also a ciphertext of $m$ where $\beta \in \{0, 1, \cdots, \lfloor (2^{2k} - (c \bmod N))/N \rfloor\}$, the adversary can ask $(c \bmod N_0) + \beta N_0$ to decryption oracle $\mathcal{D}_{sk_0}$ where $c$ is a challenge ciphertext and $pk_0 = (N_0, e_0, k)$, and if the answer of $\mathcal{D}_{sk_0}$ is $m$, then $c$ is encrypted by $pk_0$.

We can also see that this scheme is not secure in the sense of IND-CCA2. To prevent this attack, we have to add some restriction to the adversaries in the definitions of IND-CCA2 and IK-CCA. That is, in the definition of IK-CCA, it is mandated that the adversary never queries $D_{sk_0}$ on $(c \bmod N_0) + \beta_0 N_0$ where $\beta_0 \in \{0, 1, \cdots, \lfloor (2^{2k} - (c \bmod N_0))/N_0 \rfloor\}$, and $D_{sk_1}$ on $(c \bmod N_1) + \beta_1 N_1$ where $\beta_1 \in \{0, 1, \cdots, \lfloor (2^{2k} - (c \bmod N_1))/N_1 \rfloor\}$. Similarly, in the definition of IND-CCA2, it is mandated that the adversary never queries $D_{sk}$ on $(c \bmod N) + \gamma N$ where $\gamma \in \{0, 1, \cdots, \lfloor (2^{2k} - (c \bmod N))/N \rfloor\}$.

We think these restrictions are natural and reasonable. Actually, in the case of undeniable and confirmer signature schemes, Galbraith and Mao [13] defined the anonymity on undeniable signature schemes with the above restriction.

When we consider the scheme with sampling twice, we also have to add a similar restriction to the adversaries in the definitions of IND-CCA2 and IK-CCA (The difference is $\beta_0, \beta_1, \gamma \in \{0, 1\}$.).

If we add these restrictions then we can prove that the scheme with expanding and the scheme with sampling twice are secure in the sense of IND-CCA2 in the random oracle model assuming RSA is one-way. More precisely, we can prove that if RSA-OAEP is secure in the sense of IND-CCA2 then the scheme with expanding (respectively the scheme with sampling twice) is secure in the sense of IND-CCA2. Noticing that the range of valid ciphertexts changes, we can prove that the scheme with expanding and that with sampling twice are secure in the sense of IK-CCA in the random oracle model assuming RSA is one-way.

**Efficiency.** We show the number of modular exponentiations to encrypt and decrypt, and the size of ciphertexts in Figure 3. We assume that $N$ is uniformly distributed in $(2^{k-1}, 2^k)$.

In the scheme with sampling twice, it is efficient with respect to the number of modular exponentiation to encrypt in the worst case and to decrypt. However, the number of modular exponentiation to encrypt in the average case is larger than that of the other schemes.

In the scheme with repeating, it is efficient with respect to the number of exponentiation to encrypt in the average case and to decrypt. However, the upper bound of the number of exponentiations for encryption is $k_1$ ($\gg 2$, security parameter), while that of the other schemes is at most two.

In the scheme with expanding, it is efficient with respect to the computational costs to encrypt and to decrypt. However, the size of ciphertexts is twice as that of the other schemes.

In the scheme with RSACD, the number of modular exponentiation to encrypt is small in the average and worst case. However, the number of modular exponentiation to decrypt is larger than that of the other schemes.

**Restriction of $N$.** We show the restriction of $N$ for anonymity in Figure 3. In the scheme with repeating, if user $U_0$ uses the public key $N_0$ of length $k$ and user $U_1$ uses the public key $N_1$ of length $k + \log\log k_1$, it also provides anonymity by using a ciphertext $y$ which satisfies $y \leq 2^{k-1}$, and the size of ciphertext is still $k$. In this case, the number of exponentiation of $U_1$ is about $\log k_1$. If user $U_1$ uses the public key $N_1$ of length $k + \log k_1$, the number of exponentiation of $U_1$ is about $k_1$, and $\Pr[ctr = k_1]$ is non-negligible for $U_1$. Since the above probability for $U_0$ is negligible, it does not provide anonymity. In the scheme with expanding, even if each user uses the public key of different length, it also provides anonymity. In the scheme with sampling twice and that with RSACD, it is necessary for each user to choose a public key with the same length.

# 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 [9, 7]. 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 [8] 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 [6, 8, 15, 19, 14]. We describe the definition of undeniable and confirmer signature.

**Definition 10.** *An undeniable signature scheme* $\mathcal{SIG} = (\textsc{Cgen}, \textsc{Kgen}, \textsc{Sign}, \textsc{Conf}, \textsc{Deny})$ *consists of three algorithm 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 [6] claim that it is undesirable for signers to be able to confirm or deny their signatures and the schemes in [6, 8, 19] 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 [9, 7, 14, 15] 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 [13]. 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 [13] in order to put a common key generation into it explicitly.

**Definition 11 ([13]).** *Let* $\mathcal{SIG} = (\textsc{Cgen}, \textsc{Kgen}, \textsc{Sign}, \textsc{Conf}, \textsc{Deny})$ *be a undeniable or confirmer signature scheme. Let* $b \in \{0,1\}$ *and* $k \in \mathbb{N}$*(security parameter). Let* $\mathsf{si}$ *be a state information. It contains common keys, public keys, and so on. Let* $A = (A_1, A_2)$ *be adversaries that run in two stages and* $A$ *is permitted to interact with* Sign, Conf, *and* Deny*. But* $A_2$ *is not permitted to interact with* Conf, *and* Deny *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 $(m, \sigma)$ under the key $pk$, then we can easily compute all elements in $EC(m, \sigma, pk)$.). Now we consider the following experiments:*

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

*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 $k$.*

**Attacks on Anonymity**

In [15], 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 [14]. 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 [3]).

In [13], 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 $(N_1, d_1)$.

## 5.2 Undeniable and Confirmer Signature with Sampling Twice

For the comparison of the schemes, we describe the scheme with expanding proposed by Galbraith and Mao in Appendix C.

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

**Definition 12.** *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). 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.*

SIGN($m$)
  $r_1, r_2 \xleftarrow{R} \{0,1\}^{k_0}$
  $\bar{m}_1 \leftarrow$ SIGN2$(m, r_1)$; $t_1 \xleftarrow{R} \{c \in \mathbb{Z}_N \mid c^2 = \pm\bar{m}_1 \pmod N\}$; $s_1 \leftarrow (t_1)^d \bmod N$
  $\bar{m}_2 \leftarrow$ SIGN2$(m, r_2)$; $t_2 \xleftarrow{R} \{c' \in \mathbb{Z}_N \mid (c')^2 = \pm\bar{m}_2 \pmod N\}$; $s_2 \leftarrow (t_2)^d \bmod N$
  $s \leftarrow \texttt{ChooseAndShift2}^k(s_1, s_2)$
  **if** $(s \bmod N = s_1)$ $r \leftarrow r_1$ **else** $r \leftarrow r_2$
  **return** $(s, r)$

CONF *or* DENY *executes non-interactive designated verifier proofs like those in [14] which prove knowledge of an integer $e$ such that $g = h^e \pmod N$ and $s^{2e} \stackrel{?}{=} \pm$SIGN2$(m, r) \pmod N$.*

The equivalence class of this scheme is $EC(m, (s, r), pk) = \{(m, (\pm s' \pm uN, r)) | s' = s \bmod N,\ u \in \{0,1\}\}$.

### 5.3 Undeniable and Confirmer Signature with Repeating

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

**Definition 13.** *The algorithms* CGEN, KGEN, *and* SIGN2 *are the same as those of the scheme with sampling twice. The signing algorithm is as follows.*

SIGN($m$)
  $ctr \leftarrow -1$
  **repeat**
    $ctr \leftarrow ctr + 1$
    $r \xleftarrow{R} \{0,1\}^{k_0}$
    $\bar{m} \leftarrow$ SIGN2$(m, r)$
    $t \xleftarrow{R} \{c \in \mathbb{Z}_N \mid c^2 = \pm\bar{m} \pmod N\}$
    $s \leftarrow t^d \bmod N$
  **until** $(s < 2^{k-1} \vee ctr = k_1)$
**return** $(s, r)$

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 *or* DENY *executes non-interactive designated verifier proofs like those in [14] which prove knowledge of an integer $e$ such that $g = h^e \pmod N$ and $s^{2e} \stackrel{?}{=} \pm$SIGN2$(m, r) \pmod N$. Note that all users can compute* SIGN2$(m, r)$ *given $m, r$, and $N$.*

The equivalence class of this scheme is $EC(m, (s, r), pk) = \{(m, (s, r))\}$.

### 5.4 Analysis

Galbraith and Mao [13] proved that their scheme provides anonymity in the random oracle model under the assumption that the composite decision Diffie-Hellman problem is hard (See Appendix C for the definition of the composite decision Diffie-Hellman problem.). 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 both schemes with sampling twice and with repeating provide anonymity in the random oracle model under the assumption that the composite

| | Sampling Twice | Repeating | Expanding [13] |
|---|---|---|---|
| # of mod. exp. to sign (average / worst) | 2 / 2 | 1.5 / $k_1$ | 1 / 1 |
| # of computation of square root (average / worst) | 2 / 2 | 1.5 / $k_1$ | 1 / 1 |
| size of signatures | $k + k_0$ | $(k-1) + k_0$ | $2k + k_0$ |
| restriction of $N$ for anonymity (with security parameter $k$) | $\|N\| = k$ | $k \leq \|N\| \leq k + \log\log k_1$ | $\|N\| \geq k$ |

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

decision Diffie-Hellman problem is hard, and are existential unforgeable in the random oracle model under the assumption that factoring integers which are products of safe primes is hard.

We show the number of modular exponentiation to sign, the number of computation of square root, and the size of signatures in Figure 4.

In the scheme with sampling twice, the number of exponentiations for signing and that of computation of square roots is always two, while those of the other schemes are 1 or 1.5 in the average case.

In the scheme with repeating, the upper bound of the number of exponentiations for signing and that of computation of square root are $k_1$ ($\gg 2$, security parameter), while those of the other schemes are at most two.

In the scheme with expanding, the size of signatures is twice as that of the other schemes.

We have not presented a scheme with RSACD, since we have not succeed to construct such a scheme.

# 6 Ring Signature

## 6.1 Definitions

In [20], 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 14 (Ring Signature [20]).** *One assumes that each user (called a ring member) has received (via a PKI or a certificate) a public key $P_k$, for which the corresponding secret key is denoted by $S_k$. 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 $\sigma$ 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**$(m, \sigma)$ *which accepts a message $m$ and a signature $\sigma$ (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-ambiguous", which is the property that the verifier should be 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 $\epsilon$ is negligible.

Furthermore, the signature scheme must satisfy "existential unforgeability under adaptive chosen message attack", which is the property that any polynomial time adversary, where she can have access

to the signing oracle and get signatures for any message, cannot forge a message-signature pair with non-negligible probability, other than the pairs the signing oracle has previously produced.

The formal concept of ring signature can be related to an abstract concept called *combining functions*. A combining function $C_{k,v}(y_1, y_2, \cdots, y_r)$ takes as input a key $k$, an initialization value $v$, and a list of arbitrary values of the same length $\ell$. It outputs a single value $z \in \{0,1\}^\ell$ such that for any $k, v$, any index $s$, and any fixed values of $\{y_i\}_{i \neq s}$, $C_{k,v}$ is a permutation over $\{0,1\}^\ell$, when seen as a function of $y_s$. Moreover, this permutation is efficiently computable as well as its inverse.

In [20], Rivest, Shamir, and Tauman proposed a combining function based on a symmetric encryption scheme $E$ modeled by a (keyed) random permutation

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

For any index $s$, we can easily verify that $C_{k,v}$ is a combining function by rewriting equation 1 as follows:

$$y_s = E_k^{-1}\big(y_{s+1} \oplus \cdots E_k^{-1}(y_r \oplus E_k^{-1}(v)) \cdots\big) \oplus E_k\big(y_{s-1} \oplus \cdots E_k(y_1 \oplus v) \cdots\big). \tag{2}$$

### 6.2 Ring Signature with Sampling Twice

For the comparison of the schemes, we describe the scheme with expanding proposed by Rivest, Shamir, and Tauman [20] and that with RSACD proposed by Hayashi, Okamoto, and Tanaka [16] in Appendix B.

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

**Definition 15.** *Let $\ell, k$, and $b = k$ be security parameters. Let $E$ be a symmetric encryption scheme over $\{0,1\}^b$ using $\ell$-bit keys and $h$ be a hash function which maps arbitrary strings to $\ell$-bit strings. They use $h$ to make a key for $E$. Each user 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 a number of ring member. 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`
    $x_i^1, x_i^2 \xleftarrow{R} \{0,1\}^k$
    $y_i^1 \leftarrow (x_i^1)^{e_i} \bmod N_i$; $\quad y_i^2 \leftarrow (x_i^2)^{e_i} \bmod N_i$
    $y_i \leftarrow$ `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` $(y_s \geq N)$ $c_s \leftarrow 1$ `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**$(m, \sigma)$ *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 $z = v$.*

### 6.3 Ring Signature with Repeating

In this section, we propose the ring signature scheme with the repeating technique.

**Definition 16.** *The values $\ell, k, h, r$ and the function $E$ are the same as those of the scheme with sampling twice. Let $b = k - 1$. 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*. Note that $b$ is smaller than any of the size of $N_i$. The signing algorithm is as follows.*

| | Sampling Twice | Repeating | Expanding [20] | RSACD [16] |
|---|---|---|---|---|
| # of mod. exp. to sign (average / worst) | $2r$ / $2r$ | $1.5r$ / $kr$ | $r$ / $r$ | $1.5r$ / $2r$ |
| # of mod. exp. to verify (average / worst) | $r$ / $r$ | $r$ / $r$ | $r$ / $r$ | $1.5r$ / $2r$ |
| size of signature (average) | $(3r+1)k+r$ | $(3r+1)k-1$ $+\log\log k/2$ | $(r+1)(k_{\max}+160)$ $+r(k+k_{\max})$ | $(3r+1)k$ |
| size of signatures ($\lvert N_i\rvert = k$) | $(3r+1)k+r$ | $(3r+1)k-1$ | $(3r+1)k$ $+160(r+1)$ | $(3r+1)k$ |
| restriction of $N_i$ (with security parameter $k$) | $\lvert N_i\rvert = k$ | $k \le \lvert N_i\rvert \le$ $k+\log\log k$ | $\lvert N_i\rvert \ge k$ | $\lvert N_i\rvert = k$ |

Figure 5: The comparison of the ring signature schemes

**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`
    $ctr \leftarrow -1$
    `repeat`
      $ctr \leftarrow ctr + 1$
      $x_i \stackrel{R}{\leftarrow} \mathbb{Z}_{N_i}$; $y_i \leftarrow (x_i)^{e_i} \bmod N_i$
    `until` $(y_i < 2^{k-1} \vee ctr = k)$
    `if` $(ctr = k)$ $x_i \leftarrow 1$; $y_i \leftarrow 1$
  $v \stackrel{R}{\leftarrow} \{0,1\}^b$
  `find` $y_s$ `s.t.` $C_{h(m),v}(y_1, \cdots, y_r) = v$
  $x_s \leftarrow y_s^{d_s} \bmod N_s$
  `return` $\sigma = (P_1, P_2, \cdots, P_r, v, x_1, x_2, \cdots, x_r)$

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

### 6.4 Analysis

Rivest, Shamir, and Tauman 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 proved that their scheme is unconditionally signer-ambiguous and provably secure in the ideal cipher model assuming RSACD is one-way. We can prove the ring signature scheme with repeating and that with sampling twice are also unconditionally signer-ambiguous and provably secure in the ideal cipher model assuming RSA is one-way. Recently, Bresson, Stern, and Szydlo [5] improved the Rivest–Shamir–Tauman scheme. They showed that security can be based on the random oracle model which is strictly weaker complexity assumption than the ideal cipher model. Furthermore, this greatly simplified the security proof provided in [20]. We can apply their construction to the scheme with repeating, that with sampling twice, and that of Hayashi–Okamoto–Tanaka.

We show the number of modular exponentiations to sign and to verify, the size of signatures, and the restriction of the size of $N$ in Figure 5. We assume that the length of $N$ is uniformly distributed over $[k, k_{\max}]$ in the scheme with expanding, and over $[k, k+\log\log k]$ in the scheme with repeating.

In the scheme with expanding, there is no restriction of length of each user's modulo. Therefore,

the signer does not need the setting for making ring signatures. On the other hand, in the scheme with RSACD and that with sampling twice, it is necessary for each ring member to choose a public key with the same length, and in the scheme with repeating, it is necessary for each ring member to choose a public key with almost the same length.

In the situation that each user uses a public key of length $k$, the scheme with expanding is efficient than the other schemes with respect to the number of modular exponentiations to sign and to verify. On the other hand, if $k_{\max}$ is large, then the computational costs of modular exponentiations in the scheme with expanding is also large.

If $k_{\max}$ is large, it is clear that the size of signatures in the scheme with expanding is larger than that of the other schemes. In addition to it, the scheme with sampling twice, that with repeating, and that with RSACD are efficient with respect to the size of signatures than the scheme with expanding where each user uses a public key of length $k$. In this case, the size of signatures of the scheme with sampling twice, that with repeating, and that with RSACD is about $160r$ bits smaller than that of the scheme with expanding where $r$ is the number of ring members.

# 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. We have also constructed previously unproposed schemes for encryption, undeniable and confirmer signature, and ring signature by applying three previously known techniques, that is, repeating, expanding, and using RSACD. We have precisely analyzed the properties of these schemes and show their advantage and disadvantage.

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, and the computational costs to encrypt messages and to sign messages in the worst case. However, the number of exponentiations for encryption or signing is two, while that of the other schemes is one or 1.5 in the average case. Similar to the scheme with RSACD, in order to obtain the anonymity property, it is necessary for each user to choose a public key with exact the same size.

We have not succeed to construct the undeniable and confirmer signature scheme with RSACD, so it is interesting to make such a scheme. It is also interesting to consider other techniques to obtain the anonymity property.

# References


[1] Bellare, M., Boldyreva, A., Desai, A., and Pointcheval, D. Key-Privacy in Public-Key Encryption. In Boyd [4], pp. 566–582.

[2] Bellare, M., and Rogaway, P. Optimal Asymmetric Encryption – How to Encrypt with RSA. In De Santis [10], pp. 92–111.

[3] Bellare, M., and Rogaway, P. The Exact Security of Digital Signatures: How to Sign with RSA and Rabin. In Maurer [18], pp. 339–416.

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

[5] 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 Conputer Science*, Springer-Verlag, pp. 465–480.

[6] 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 Conputer Science*, Springer-Verlag, pp. 243–258.

[7] 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 Conputer Science*, Springer-Verlag, pp. 458–464.

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

[9] 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 Conputer Science*, Springer-Verlag, pp. 212–217.

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

[11] 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 Conputer Science*, Springer-Verlag, pp. 147–157.

[12] 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 Conputer Science*, Springer-Verlag, pp. 260–274.

[13] 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 Conputer Science*, Springer-Verlag, pp. 80–97.

[14] 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 Conputer Science*, Springer-Verlag, pp. 200–217.

[15] 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 Conputer Science*, Springer-Verlag, pp. 132–149.

[16] Hayashi, R., Okamoto, T., and Tanaka, K. An RSA Family of Trap-door Permutations with a Common Domain and its Applications. To appear in *Public Key Cryptography – PKC 2004* (Singapore, March 2004).

[17] Jakobsson, M., Sako, K., and Impagliazzo, R. Designated Verifier Proofs and their Applications. In Maurer [18], pp. 143–154.

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

[19] 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 Conputer Science*, Springer-Verlag, pp. 406–421.

[20] Rivest, R. L., Shamir, A., and Tauman, Y. How to Leak a Secret. In Boyd [4], pp. 552–565.

# A The Schemes Previously Proposed for Encryption

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

In [1], 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 17 (RSA-RAEP [1]).** *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, *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** $\mathcal{E}_{pk}^{G,H}(x)$

$ctr = -1$
**repeat**
  $ctr \leftarrow ctr + 1$
  $r \xleftarrow{R} \{0,1\}^{k_0}$
  $s \leftarrow (x \;||\; 0^{k_1}) \oplus G(r)$; $t \leftarrow r \oplus H(s)$
  $v \leftarrow (s||t)^e \bmod N$
**until** $((v < 2^{k-1}) \vee (ctr = k_1))$
**if** $(ctr = k_1)$ $y \leftarrow 1||0^{k_0+k_1}||x$
**else** $y \leftarrow 0||v$
**return** $y$

**Algorithm** $\mathcal{D}_{sk}^{G,H}(y)$

$b \leftarrow [y]^1$; $v \leftarrow [y]_{k_0+k_1+n}$
**if** $(b = 1)$
  $w \leftarrow [v]^{k_0+k_1}$; $x \leftarrow [v]_n$
  **if** $(w = 0^{k_0+k_1})$ $z \leftarrow x$ **else** $z \leftarrow \perp$
**else**
  $s \leftarrow [v^d]^{n+k_1}$; $t \leftarrow [v^d]_{k_0}$
  $r \leftarrow t \oplus H(s)$
  $x \leftarrow [s \oplus G(r)]^n$; $p \leftarrow [s \oplus G(r)]_{k_1}$
  **if** $(p = 0^{k_1})$ $z \leftarrow x$ **else** $z \leftarrow \perp$
**return** $z$

They proved that RSA-RAEP is secure in the sense of IK-CCA in the random oracle model assuming RSA is one-way.

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

In [16], Hayashi, Okamoto, and Tanaka proposed an RSA-based encryption scheme. It uses RSACD instead of RSA. We describe their scheme.

**Definition 18.** *The common-key generation algorithm* $\mathcal{G}$, *the key generation algorithm* $\mathcal{K}$, *and the oracles* $G$ *and* $H$ *are the same as that 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** $\mathcal{E}_{pk}^{G,H}(x)$

$r \xleftarrow{R} \{0,1\}^{k_0}$
$s \leftarrow (x \;||\; 0^{k_1}) \oplus G(r)$
$t \leftarrow r \oplus H(s)$
$v \leftarrow f_{N,e,k}^{\mathsf{RSACD}}(s||t)$
**return** $y$

**Algorithm** $\mathcal{D}_{sk}^{G,H}(y)$

$s \leftarrow [g_{N,d,k}^{\mathsf{RSACD}}(y)]^{n+k_1}$; $t \leftarrow [g_{N,d,k}^{\mathsf{RSACD}}(y)]_{k_0}$
$r \leftarrow t \oplus H(s)$
$x \leftarrow [s \oplus G(r)]^n$; $p \leftarrow [s \oplus G(r)]_{k_1}$
**if** $(p = 0^{k_1})$ $z \leftarrow x$ **else** $z \leftarrow \perp$
**return** $z$

They proved that their scheme is secure in the sense of IK-CCA in the random oracle model assuming RSACD is one-way.

# B The Schemes Previously Proposed for Ring Signature

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

In [20], 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 19 ([20]).** *Let $\ell, k$, and $b$ be security parameters. Let $E$ be a symmetric encryption scheme over $\{0,1\}^b$ using $\ell$-bit keys and $h$ be a hash function which maps arbitrary strings to $\ell$-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**$(m, P_1, P_2, \cdots, P_r, s, S_s)$
  `for each` $i \in \{1, \cdots, s-1, s+1, \cdots, r\}$ `do`
    $x_i \stackrel{R}{\leftarrow} \{0,1\}^b$; $y_i \leftarrow g_i(x_i)$
  $k \leftarrow h(m)$; $v \stackrel{R}{\leftarrow} \{0,1\}^b$
  `find` $y_s$ `s.t.` $C_{h(m),v}(y_1, \cdots, y_r) = v$
  $x_s \leftarrow g_s^{-1}(y_s)$
  `return` $\sigma = (P_1, P_2, \cdots, P_r, v, x_1, x_2, \cdots, x_r)$

*Note that we can find $y_s$ such that $C_{h(m),v}(y_1, \cdots, y_r) = v$ in the signing algorithm by using equation 2. The verification algorithm* **ring-verify**$(m, \sigma)$ *computes $y_i \leftarrow g_i(x_i)$ for each $x_i$ and $z \leftarrow C_{h(m),v}(y_1, \cdots, y_r)$. It returns* `valid` *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.

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

In [20], Rivest, Shamir, and Tauman mentioned the case that a member of the cabinet of some country wished to leak her secret to a journalist. In this kind of situation, it is reasonable to consider that the members of the same group use the RSA moduli of the same length. In [16], Hayashi, Okamoto, and Tanaka considered the situation that each user chooses her public key with the same size. We describe their ring signature scheme. Their scheme is almost the same as the Rivest–Shamir–Tauman scheme. They used $f_{N_i,e,k}^{\mathsf{RSACD}}(\cdot)$ instead of $g_i(\cdot)$.

**Definition 20.** *The values $\ell, k, E, h, r$ are the same as those of 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**$(m, P_1, P_2, \cdots, P_r, s, S_s)$
  **for each** $i \in \{1, \cdots, s-1, s+1, \cdots, r\}$ **do**
    $x_i \stackrel{R}{\leftarrow} \{0,1\}^k$; $y_i \leftarrow f^{\mathsf{RSACD}}_{N_i,e_i,k}(x_i)$
  $v \stackrel{R}{\leftarrow} \{0,1\}^k$
  **find** $y_s$ **s.t.** $C_{h(m),v}(y_1, \cdots, y_r) = v$
  $x_s \leftarrow g^{\mathsf{RSACD}}_{N_s,d_s,k}(y_s)$
  **return** $\sigma = (P_1, P_2, \cdots, P_r, v, x_1, x_2, \cdots, x_r)$

*The verification algorithm* **ring-verify**$(m, \sigma)$ *computes* $y_i \leftarrow f^{\mathsf{RSACD}}_{N_i,e_i,k}(x_i)$ *for each* $x_i$ *and* $z \leftarrow C_{h(m),v}(y_1, \cdots, y_r)$. *It returns* `valid` *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.

# C The Schemes Previously Proposed for Undeniable and Confirmer Signature

## C.1 Undeniable and Confirmer Signature with Expanding by Galbraith and Mao

In [13], Galbraith and Mao proposed a new RSA-based scheme. We describe their scheme.

**Definition 21 ([13]).** *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). 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$(m)$ |
|---|---|
| 1 | $r \stackrel{R}{\leftarrow} \{0,1\}^{k_0}$ |
| 2 | $\bar{m} \leftarrow$ SIGN2$(m, r)$ |
| 3 | $t \stackrel{R}{\leftarrow} \{c \in \mathbb{Z}_N \mid c^2 = \pm\bar{m} \pmod N\}$ |
| 4 | $s \leftarrow t^d \bmod N$ |
| 5 | $u \stackrel{R}{\leftarrow} \{0, 1, \cdots, \lfloor (2^{2k} - s)/N \rfloor\}$ |
| 6 | $\hat{s} \leftarrow s + uN$ |
| 7 | **return** $(\hat{s}, r)$ |

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 *or* DENY *executes non-interactive designated verifier proofs like those in [14] which prove knowledge of an integer* $e$ *such that* $g = h^e \pmod N$ *and* $\hat{s}^{2e} \stackrel{?}{=} \pm$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* $N$.

Galbraith and Mao defined $EC(m, (\hat{s}, r), pk) = \{(m, (\pm s \pm uN, r)) | s = \hat{s} \bmod N,\ u \in \{0, 1, \cdots, \lfloor (2^{2k} - s)/N \rfloor\}\}$.
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 bitstrings 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 [11].).

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 [17].

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.

**Remark (Composite Decision Diffie-Hellman Problem)** Let $N$ be a product of two safe primes (i.e. $N = pq$ where $p$, $q$, $(p-1)/2$, $(q-1)/2$ are primes). Consider the two sets

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

and

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

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.