RSA Family of Trap-door Permutations with a Common Domain and its Applications (Extended Abstract)

Ryotaro Hayashi[1] Tatsuaki Okamoto[2] Keisuke Tanaka[1] *

[1] 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

[2] NTT Labs
1-1 Hikarino-oka, Yokosuka-shi, Kanagawa 239-0847, Japan
okamoto@isl.ntt.co.jp

May 23, 2003

Summary

Bellare , Boldyreva , Desai, and Pointcheval [1] recently 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. Incidentally, Rivest, Shamir, and Tauman [3] recently 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.

We are concerned with an underlying primitive element common to the key-privacy encryption and the ring signature schemes, that is, a family of trap-door RSA permutations with a common domain. For a standard family of trap-door RSA permutations, even if all of the functions in a family use RSA moduli of the same size (the same number of bits), it will have domains with different sizes. In this paper, we construct a new family of trap-door RSA permutations with a common domain, and propose the applications of our construction to the key-privacy encryption and ring signature schemes, which have some advantage to the previous schemes.

Keywords: RSA, family of trap-door permutations, key-privacy, anonymity, encryption, ring signature

1 Introduction

We propose a new family of trap-door permutations called "RSA family of trap-door permutations with a common domain", which can be applied to public-key encryption with the key-privacy property and ring signature.

*Supported in part by Grant-in-Aid for Scientific Research, Ministry of Education, Culture, Sports, Science and Technology, 14780190.

Bellare , Boldyreva , Desai, and Pointcheval [1] recently 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. The standard RSA encryption does not provide key-privacy. Since even if two public-key N_0 and N_1 ($N_0 < N_1$) are the same bits, $N_1 - N_0$ may be large. In [1], they provided the key-privacy encryption scheme, RSA-RAEP, which is variant of RSA-OAEP (Fujisaki, Okamoto, Pointcheval, and Stern [2]), and solved this problem by repeating the evaluation of an RSA-OAEP $f(x, r)$ permutation with plaintext x and random r, each time using a different r until the value is in the safe range (see section 3.2). For deriving a value in the safe range, the number of the repetition would be very large (the value of the security parameter).

Incidentally, Rivest, Shamir, and Tauman [3] recently 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 member will have domains of different sizes. This makes it awkward to combine the individual signatures, so one should construct some trap-door one-way permutation which has common domain for each user. Intuitively, in ring signature scheme, Rivest, Shamir, and Tauman solved this by encoding the message to an N_i-ary representation and applying a standard permutation f to the low-order digit (see section 4.2). For deriving a secure permutation g with a common domain, the domain of g would be 160 bits larger than that of f.

In this paper, we will take a different approach. We use neither an n_i-ary representation nor the repetition of the evaluation of a permutation. We are concerned with an underlying primitive element common to the key-privacy encryption and the ring signature schemes, that is, a family of trap-door RSA permutations with a common domain. For a standard family of trap-door RSA permutations, even if all of the functions in a family use RSA moduli of the same size (the same number of bits), it will have domains with different sizes. We construct a new family of trap-door RSA permutations with a common domain

We then propose the application to the key-privacy encryption scheme. Our proposed scheme is more efficient with respect to the number of exponentiations for encryption. When we use the RSA moduli which is uniformly distributed in $(2^{k-1}, 2^k)$, the expected number of our scheme is the same as that of RSA-RAEP. But in our scheme, the number of exponentiations for encryption is at most two, while in RSA-RAEP, the upper bound of this number is k_1 ($\gg 2$, security parameter). Notice that we use the randomness only for an RSA-OAEP.

We also propose the application to the ring signature scheme. We consider the case that the members of the same group use the RSA moduli of the same length. In our scheme, the domain of $f_{N,e}(\cdot)$ is $\{0,1\}^k$, while that of the previous scheme is $\{0,1\}^{k+160}$, where k is the length of the RSA moduli. Thus, we can reduce the size of signature in this situation. We can also apply this scheme to the Rabin-based ring signature scheme in [3].

The organization of this paper is as follows. In Section 2, we briefly review the definitions of functions and the standard RSA family. We also propose the RSA family of trap-door permutations with a common domain and prove that if the standard RSA family is partial one-way, then the RSA family of trap-door permutations with a common domain is partial one-way. In Section 3, we review the key-privacy encryption scheme and propose the application of our new family to the key-privacy encryption scheme. In Section 4, we review ring signature and propose the application of our new family to the ring signature. We conclude in Section 5.

2 RSA Family of Trap-door Permutations with a Common Domain

2.1 Definitions

In this section, we first briefly review the definitions of family of functions, and the standard RSA family of trap-door permutations.

Definition 1 (family of functions [1]). A family of functions $F = (K, S, E)$ is specified by three algorithms.

- The randomized key-generation algorithm K takes input the 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 $\mathrm{Dom}_F(pk)$.
- 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 $\mathrm{Rng}_F(pk) = \{E_{pk}(x) \ : \ x \in \mathrm{Dom}_F(pk)$ denote the range of the function $E_{pk}(\cdot)$.

□

Definition 2 (family of trap-door permutations [1]). We say that F is 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, and $\mathrm{Dom}_F(pk) = \mathrm{Rng}_F(pk)$, and E_{pk} is a permutation on this set. □

Now, we describe the definition of partial one-way.

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

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

We define the advantages of the adversary via

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

The family F is said to be 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. □

Note that when $\theta = 1$ the notion of θ-partial one-wayness coincides with the standard notion of one-wayness.

We now describe the standard RSA family of trap-door permutations denoted by RSA. The θ-partial one-wayness of RSA has been shown to be equivalent to the one-wayness of RSA, for $\theta = 0.5$ (Fujisaki, Okamoto, Pointcheval, and Stern [2]).

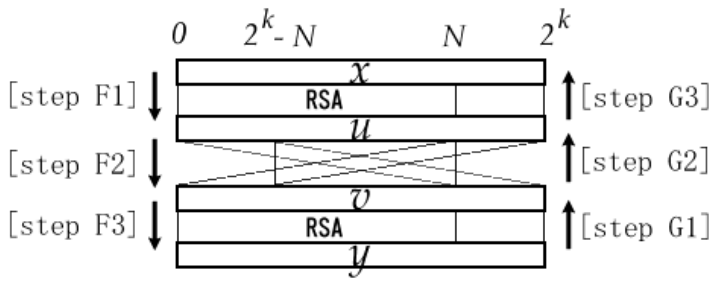

Figure 1: function $f_{N,e}$ and $g_{N,d}$

Definition 4 (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 $2^{k/2-1} < p, q < 2^{k/2}$. (If k is odd, increment it by 1 before picking the primes.) It sets $N = pq$. (i.e. $2^{k-2} < N < 2^k$.) It picks $e, d \in \mathbb{Z}^*_{\phi(N)}$ such that $ed \equiv 1 \pmod{\phi(N)}$ where $\phi(N) = (p-1)(q-1)$. The public key is N, e and the secret key is N, d. The sets $\mathrm{Dom}_{\mathsf{RSA}}(N, e)$ and $\mathrm{Rng}_{\mathsf{RSA}}(N, e)$ are both equal to $\mathbb{Z}^*_N$. The evaluation algorithm $E_{N,e}(x) = x^e \bmod N$ and the inversion algorithm $I_{N,d}(y) = y^d \bmod N$. The sampling algorithm returns a random point in $\mathbb{Z}^*_N$. □

2.2 RSA Family of Trap-door Permutations with a Common Domain

In this section, we propose the RSA family of trap-door permutations with a common domain denoted by RSACD and prove that if the standard RSA family is partial one-way, then the modified RSA family is partial one-way.

Definition 5 (RSA family of trap-door permutations with a common domain). 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 almost the same as that for RSA family. The difference is picking two distinct primes p, q such that $2^{k/2-1} < p, q < 2^{k/2}$ and $2^{k-1} < N < 2^k$. The sets $\mathrm{Dom}_{\mathsf{RSACD}}(N, e)$ and $\mathrm{Rng}_{\mathsf{RSACD}}(N, e)$ are both $\{x | x \in [0, 2^k) \ \wedge \ x \bmod N \in \mathbb{Z}^*_N\}$. The sampling algorithm returns a random point in $\mathrm{Dom}_{\mathsf{RSACD}}(N, e)$. The evaluation algorithm $E_{N,e}(x) = f_{N,e}(x)$, and the inversion algorithm $I_{N,d}(y) = g_{N,e}(y)$ are as follows. (See Figure 1.)

```
Function f_{N,e}(x)
  [step F1] if (x < N) u ← x^e mod N else u ← x
  [step F2] if (u < 2^k − N) v ← u + N elseif (2^k − N ≤ u < N) v ← u else v ← u − N
  [step F3] if (v < N) y ← v^e mod N else y ← v
  return y

Function g_{N,d}(y)
  [step G1] if (y < N) v ← y^d mod N else v ← y
  [step G2] if (v < 2^k − N) u ← v + N elseif (2^k − N ≤ v < N) u ← v else u ← v − N
  [step G3] 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 $\mathrm{Dom}_{\mathsf{RSACD}}(N, e)$ are permuted by the RSA function at least once. Then, we have the following theorem.

Theorem 1. If the standard RSA family of trap-door permutations is (θ-partial) one-way, then the RSA family of trap-door permutations with a common domain is (θ-partial) one-way. □

Proof. Let $\mathrm{RSA}_{N,e}(\cdot)$ be the standard RSA function (i.e. $\mathrm{RSA}_{N,e}(x) = x^e \bmod N$). If there exists a polynomial-time algorithm which can (θ-partial) invert $f_{N,e}$ with non-negligible probability, then there exists a polynomial-time algorithm which can (θ-partial) invert $\mathrm{RSA}_{N,e}$ with non-negligible probability.

Let $P = [0, 2^k - N), Q = [2^k - N, N), R = [N, 2^k)$. We specify a ($\theta$-partial) inverting algorithm D for the standard RSA function using the (θ-partial) inverting algorithm A for $f_{N,e}$ whose advantage is non-negligible.

```
                Algorithm D(N, e, Y)
                  x ← A(N, e, Y)
                  if (x ∈ R)
[step R]            t ← x − N
                  else
                    if (Y ∈ P)
[step P]              y ← Y + N; t ← A(N, e, y)
                    else
[step Q]              y ← Y^e mod N; t ← A(N, e, y)
                  return t
```

Now, we analyze the advantage of D. If A can invert y to get x with non-negligible provability, then

$$\Pr[u \in P]\Pr[A(y) = x | u \in P] + \Pr[u \in Q]\Pr[A(y) = x | u \in Q] + \Pr[u \in R]\Pr[A(y) = x | u \in R]$$

is non-negligible. So, for some $T \in \{P, Q, R\}$, $\Pr[u \in T]\Pr[A(y) = x | u \in T]$ is non-negligible, and for such T, $\Pr[A_T]\Pr[t = Y^d \bmod n | A_T]$ is non-negligible. (A_T denotes "[step T] is executed".) Therefore, the advantage of the above algorithm is non-negligible.

The input N of A is chosen from $(2^{k-1}, 2^k)$, and the input N of D is chosen from $(2^{k_2}, 2^k)$. Note that even if D outputs `fail` when $2^{k-2} < N < 2^{k-1}$, D's success probability is non-negligible because of D can success when $2^{k-1} < N < 2^k$. □

3 Application to Key-Privacy Encryption

3.1 Definitions of Key-Privacy

The classical security requirements of an encryption scheme, for example indistinguishability or non-malleability under 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 is defined as follows.

Definition 6 (public-key encryption scheme [1]). A public-key encryption scheme $\mathcal{PE} = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ consists of four algorithms.

- The common-key generation algorithm $\mathcal{G}$ takes as input some security parameter k and returns some common key I.

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

□

In [1], they formalize the property of "key-privacy". This can be considered under either chosen-plaintext attack or 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 $k \in \mathbf{N}$. Let A_{cpa}, A_{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. Now, we consider the following experiments:

$\texttt{Experiment}\ \mathbf{Exp}^{\text{ik-cpa-}b}_{\mathcal{PE},A_{\text{cpa}}}(k)$	$\texttt{Experiment}\ \mathbf{Exp}^{\text{ik-cca-}b}_{\mathcal{PE},A_{\text{cca}}}(k)$
$I \xleftarrow{R} \mathcal{G}(k)$	$I \xleftarrow{R} \mathcal{G}(k)$
$(pk_0, sk_0) \xleftarrow{R} \mathcal{K}(I);\ (pk_1, sk_1) \xleftarrow{R} \mathcal{K}(I)$	$(pk_0, sk_0) \xleftarrow{R} \mathcal{K}(I);\ (pk_1, sk_1) \xleftarrow{R} \mathcal{K}(I)$
$(x, \mathsf{si}) \leftarrow A_{\text{cpa}}(\mathsf{find}, pk_0, pk_1)$	$(x, \mathsf{si}) \leftarrow A_{\text{cca}}^{\mathcal{D}_{sk_0}(\cdot),\mathcal{D}_{sk_1}(\cdot)}(\mathsf{find}, pk_0, pk_1)$
$y \leftarrow \mathcal{E}_{pk_b}(x)$	$y \leftarrow \mathcal{E}_{pk_b}(x)$
$d \leftarrow A_{\text{cpa}}(\mathsf{guess}, y, \mathsf{si})$	$d \leftarrow A_{\text{cca}}^{\mathcal{D}_{sk_0}(\cdot),\mathcal{D}_{sk_1}(\cdot)}(\mathsf{guess}, y, \mathsf{si})$
$\texttt{return}\ d$	$\texttt{return}\ d$

Above it is mandated that A_{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

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

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. □

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.

3.2 RSA-RAEP by Bellare, Boldyreva, Desai, and Pointcheval

A simple observation that seems to be folklore is that standard RSA encryption does not provide key-privacy, 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 it bets it was created under (N_0, e_0). It is not hard to see that this attack has non-negligible advantage.

In [1], they propose 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 $(2^{k-2}, 2^k)$, it simply repeats the ciphertext computation, each time using new coins, until the ciphertext y satisfies $y < 2^{k-2}$.

Definition 8 (RSA-RAEP [1]). RSA-RAEP $= (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ is as follows. The common key generator 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 the RSA family, and get (N, e) and (N, d). The public key for the scheme 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. The oracles G and H which $\mathcal{E}_{pk}$ and $\mathcal{D}_{sk}$ reference below map bit strings as follows: $G : \{0,1\}^{k_0} \to \{0,1\}^{n+k_1}$ and $H : \{0,1\}^{n+k_1} \to \{0,1\}^{k_0}$. Note that $[x]^n$ denotes the n most significant bits of x, while $[x]_m$ 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** $((v < 2^{k-2}) \vee (ctr = k_1))$
- **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 RSA-RAEP is secure in the sense of IND-CCA2 and IK-CCA in the random oracle model assuming RSA family of trap-door permutations is partial one-way. In RSA-RAEP, the expected number of exponentiations for encryption is

$$\sum_{i=1}^{k_1} i \left(1 - \frac{2^{k-2}}{N}\right)^{i-1} \frac{2^{k-2}}{N} = \frac{1-(1-p)^{k_1}}{p} - k_1(1-p)^{k_1}$$

where $p = 2^{k-2}/N$. Suppose that N is uniformly distributed in $(2^{k-2}, 2^k)$, the expected number of this scheme is two. However, the upper bound of the number of exponentiations for encryption is $k_1 (\gg 2)$ (security parameter).

3.3 Our Proposed Encryption Scheme

We now propose our encryption scheme. In our scheme, we use the RSA family of trap-door permutations with a common key instead of the standard RSA family of trap-door permutations. Now, we describe our proposed scheme.

Definition 9. The common key generator algorithm $\mathcal{G}$, and the oracles G and H are the same as RSA-RAEP. The key generation algorithm $\mathcal{K}$ is almost the same as RSA-RAEP. The difference is running the key-generation algorithm of the RSACD family instead of the RSA family. 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 $\mathcal{E}_{pk}^{G,H}(x)$
- $r \xleftarrow{R} \{0,1\}^{k_0}$
- $s \leftarrow (x \;||\; 0^{k_1}) \oplus G(r)$
- $t \leftarrow r \oplus H(s)$
- $v \leftarrow f_{N,e}(s||t)$
- **return** y

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

□

Using Theorem 1, we can prove the following theorem.

Theorem 2. Our scheme is secure in the sense of IND-CCA2 and IK-CCA under the assumption that RSA is partial one-way. □

Proof Idea. By Theorem 1, if RSA is partial one-way, our proposed function $f_{N,e}$ is also partial one-way. Fujisaki, Okamoto, Poincheval, and Stern [2] proved OAEP with partial one-way function is secure in the sense of IND-CCA2. Thus, OAEP with $f_{N,e}$ is secure in the sense of IND-CCA2 under the assumption that RSA is one-way. Noticing that the function and the domain of valid signature change, the IK-CCA proof of our scheme is similar to that of RSA-RAEP. □

In this scheme, the expected number of exponentiations in encryption is

$$1 \times \frac{2(2^k - N)}{2^k} + 2 \times \left(1 - \frac{2(2^k - N)}{2^k}\right) = \frac{N}{2^{k-1}}.$$

Suppose that N is uniformly distributed in $(2^{k-1}, 2^k)$, the expected number of our scheme is two, the same as RSA-RAEP. In our scheme, the number of exponentiations for encryption is at most two, while in RSA-RAEP, the upper bound of this number is k_1 ($\gg 2$, security parameter). Notice that we use the randomness only for an RSA-OAEP.

4 Application to Ring Signature

4.1 Definitions of Ring Signature

In [3], 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. (This property is called "signer-ambiguous.") Unlike group signature, ring signature has no group managers, no setup procedures, no revocation procedures, and no coordination.

Definition 10 (Ring Signature [3]). 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 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 σ for the message m, given the public keys $P_1, P_2, \cdots, P_r$ of the r ring members, together with the secret key S_s of the s-th member (who is the actual signer).
- **ring-verify**(m, σ) which accepts a message m and a signature σ (which includes the public key of all the possible signers), and outputs either "true" or "false".

□

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 ϵ is negligible.

In [3], they defined combining functions to sign and verify a ring signature efficiently as follows. We follow the definition given in [4] by Bresson, Stern, and Szydlo, which is slightly modified version of the original one in [3].

Definition 11 (Combining functions [3, 4]). A combining function $C_{k,v}(y_1, \cdots, y_r)$ takes as input a key k , an initialization value v , and arbitrary values $y_1, \cdots, y_r \in \{0,1\}^b$, and outputs a single value $z \in \{0,1\}^b$, 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 $\{0,1\}^b$, when seen as a function of y_s. Moreover, this permutation is efficiently computable as well as its inverse. □

4.2 RSA-based Ring Signature Scheme by Rivest, Shamir, and Tauman

In [3], they constructed ring signature schemes in which all the ring member use RSA as their individual signature schemes and these schemes are unconditionally signer-ambiguous and provably secure in the random oracle model. We also use the construction proposed in [3].

We denote by ℓ, k, b are 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. We use h to make key for E. We assume that each user has an RSA public key $P_i = (N_i, e_i)$, and which specifies the trap-door one-way permutation f_i of $\mathbb{Z}_{N_i}$:

$$f_i(x) = x^e \bmod N_i.$$

In [3], they 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:

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

In their scheme, the inputs y_i to the combining function are computed as $f_i(x_i)$ for some $x_i \in \{0,1\}^k$. A ring signature on a message m consists in a tuple $(v, x_1, \cdots, x_r)$. Setting $z = C_{h(m),v}(f_1(x_1), \cdots, f_r(x_r))$, the signature is valid iff $z = v$.

For any message m, any fixed values v and $\{x_i\}_{i \neq s}$, one can efficiently compute the value y_s such that the combining function outputs v. Now using his knowledge of the trapdoor for function f_s, s-th member (the actual signer) is able to compute x_s such that $f_s(x_s) = y_s$.

However, the trap-door RSA permutations of the various ring member will have domains of different sizes. This makes it awkward to combine the individual signatures, so we should construct some trap-door one-way permutation which has common domain for each user. They defined the extended trap-door permutation g_i over $\{0,1\}^b$ as follows.

For any b-bit input m define nonnegative integers q_i and r_i so that $m = q_i N_i + r_i$ and $0 \leq r_i < N_i$. Then

$$g_i(m) = \begin{cases} q_i N_i + f_i(r_i) & \text{if } (q_i+1)N_i \leq 2^b \\ m & \text{otherwise} \end{cases}$$

If b is sufficient large (e.g. 160 bits larger than any of the N_i), the chance that a randomly chosen m is unchanged by the extended g_i becomes negligible, so g_i is one-way trap-door permutation. (See also [5].)

4.3 Our Proposed Ring Signature Scheme

Since ring signature is set-up free, each user can use a public key whose length is different from other users. In [3], Rivest, Shamir, and Tauman mentioned the case that a member of the cabinet of some country wished to leak his secret to a journalist. In this kind of situation, it was reasonable to consider that the members of the same group use the RSA moduli of the same length.

In our scheme, we assume the situation that each user chooses her public key with the same size. We emply a one-way trap-door permutation $f_{N_i,e}(\cdot)$ in Section 2.2 instead of $g_i(\cdot)$ in Section 4.2. Then, the domain of $f_{N,e}(\cdot)$ is $\{0,1\}^k$, while that of the previous scheme is $\{0,1\}^{k+160}$, where k is the length of the RSA moduli. Thus, we can reduce the size of signature in this situation. However, in our scheme, the number of exponentiations is one or two, while that of the original scheme in [3] is one. We can also apply this scheme to the Rabin-based ring signature scheme in [3] in a similar way.

5 Conclusion

Bellare , Boldyreva , Desai, and Pointcheval [1] recently 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. Incidentally, Rivest, Shamir, and Tauman [3] recently 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.

We are concerned with an underlying primitive element common to the key-privacy encryption and the ring signature schemes, that is, a family of trap-door RSA permutations with a common domain. For a standard family of trap-door RSA permutations, even if all of the functions in a family use RSA moduli of the same size (the same number of bits), it will have domains with different sizes. In this paper, we construct a new family of trap-door RSA permutations with a common domain and propose the applications of our construction to the key-privacy encryption and ring signature schemes, which have some advantage to the previous schemes.

It might be interesting to consider other applications of our family, and different constructions of a family of trap-door permutations with a common domain.

References

[1] M. Bellare, A. Boldyreva, A. Desai and D. Pointcheval. Key-Privacy in Public-Key Encryption. Proc. Asiacrypt 2001, pp. 566-582.

[2] E. Fujisaki, T. Okamoto, D. Pointcheval, and J. Stern, RSA-OAEP is Secure under the RSA Assumption, Advances in Cryptology - Proceedings of CRYPTO '01, LNCS 2139, Springer-Verlag, 2001, pp. 260-274.

[3] R. L. Rivest, A. Shamir, and Y. Tauman, How to Leak A Secret, Advances in Cryptology - ASIACRYPT 2001, LNCS 2248, Springer-Verlag, 2001, pp. 552-565.

[4] E. Bresson, J. Stern, and M, Szydlo, Threshold Ring Signatures and Applications to Ad-hoc Groups, Proceedings of CRYPTO 2002, LNCS 2442, Springer-Verlag, 2002, pp. 465-480.

[5] Y. Desmedt, Securing traceability of ciphertexts: Towards a secure software escrow scheme, in Guillou et al. (eds.), EUROCRYPT '95, LNCS 921, Springer-Verlag, 1995, pp. 147-157.