New Semantically Secure Public-Key Cryptosystems from the RSA-Primitive

Kouichi Sakurai[1] and Tsuyoshi Takagi[2]

[1] Kyushu University
Department of Computer Science and Communication Engineering
Hakozaki, Fukuoka 812-81, Japan
sakurai@csce.kyushu-u.ac.jp
[2] Technische Universität Darmstadt,
Fachbereich Informatik,
Alexanderstr.10, D-64283 Darmstadt, Germany
ttakagi@cdc.informatik.tu-darmstadt.de

Abstract. We analyze the security of the simplified Paillier (S-Paillier) cryptosystem, which was proposed by Catalano et al. We prove that the one-wayness of the S-Paillier scheme is as intractable as the standard RSA problem. We also prove that an adversary, which breaks the semantic security, can compute the least significant bits of the nonce. This observation is interesting, because the least significant bit of the nonce is the hard core bit of the encryption function. Moreover, we proposed a novel semantically secure cryptosystem, based on the one-way function $f^{e,n}_{MSBZ(l)}(r) = (r - MSB_l(r))^e \bmod n$, where (e, n) is the RSA public-key and $r - MSB_l(r)$ means that the l most significant bits of r are zeroed. We proved that the one-wayness of the proposed scheme is as intractable as the standard RSA problem. An adversary, which breaks the semantic security of the proposed scheme, can break the least significant bits of the nonce. These security results of the proposed scheme are similar to those of the S-Paillier cryptosystem. However, the proposed scheme is more efficient than the S-Paillier cryptosystem.

1 Introduction

One of the requirements for a secure public-key cryptosystem is the semantic security, which is assorted the indistinguishability against the chosen plaintext attack (IND-CPA) and the indistinguishability against the chosen ciphertext attack (IND-CCA) [BDPR98]. Although there is an IND-CCA public-key cryptosystem from the discrete logarithm primitive in a standard model, namely the Cramer-Shoup cryptosystem [CS98], there is no IND-CCA public-key cryptosystem from the RSA primitive in a standard model. It is an interesting problem to find such a public-key cryptosystem. The Cramer-Shoup cryptosystem is converted a standard IND-CPA ElGamal cryptosystem to be an IND-CCA scheme using hash functions. The security of the IND-CPA ElGamal cryptosystem relies on the decisional Diffie-Hellman (DDH) assumption. On the contrary, the RSA

D. Naccache and P. Paillier (Eds.): PKC 2002, LNCS 2274, pp. 1–16, 2002.

primitive has no standard IND-CPA cryptosystem corresponding to the standard ElGamal. We first have to consider an IND-CPA public-key cryptosystem from the RSA primitive in order to construct an IND-CCA public-key cryptosystem from the RSA primitive in a standard model. Because there is no decisional RSA problem, we need a contrivance for exploring a suitable decisional problem from the RSA primitive. In this paper, we investigate the security of an IND-CPA cryptosystem from the RSA primitive and we call IND-CPA as semantically secure in the following. The Pointcheval cryptosystem [Poi99] and the simplified version of Paillier cryptosystem [CGHN01] are known as semantically secure public-key cryptosystems from the RSA primitive. However, the security of these cryptosystems is not well studied comparing with the standard ElGamal cryptosystem. It is unknown that the one-wayness of these cryptosystem is as hard as solving the standard problem, e.g. the RSA problem or factoring problem. Although the semantic security of these scheme is proved equivalent to a decisional number-theoretic problem, the decisional problem has not been well studied, and no non-trivial relationship between the computational problem and its corresponding decisional problem is known.

The Paillier cryptosystem is a probabilistic encryption scheme over the ring $\mathbb{Z}/n^2\mathbb{Z}$, where n is the RSA modulus [Pai99]. It encrypts a message $m \in \mathbb{Z}/n\mathbb{Z}$ by computing $E(m, r) = g^m r^n \bmod n^2$, where r is a random integer in $\mathbb{Z}/n\mathbb{Z}$, and g is an element whose order in $\mathbb{Z}/n^2\mathbb{Z}$ is divisible by n. The encryption function $E(m, r)$ has a homomorphic property: $E(m_1, r_1)E(m_2, r_2) = E(m_1 + m_2, r_1 r_2)$. Therefore, the Paillier cryptosystem can be used as the primitives for voting systems, commitment schemes, threshold schemes, etc [DJ01] [CGHN01]. The security of the Paillier cryptosystem has been investigated in [Pai99]. The one-wayness of the Paillier cryptosystem is related to the computational composite residuosity (C-CR) problem, which finds m from its encryption $g^m r^n \bmod n^2$. It is known that an algorithm, which solves the RSA problem with the encryption exponent $e = n$, can solve the C-CR problem. The semantic security of the Paillier cryptosystem is based on the decisional composite residuosity (D-CR) problem, which determines whether an integer x of $\mathbb{Z}/n^2\mathbb{Z}$ is represented as $x = a^n \bmod n^2$ for an integer a of $\mathbb{Z}/n^2\mathbb{Z}$. Then, Catalano et al. proved that $n - b$ least significant bits of the message are simultaneously secure under the 2^b-hard C-CR assumption, where the 2^b-hard C-CR assumption uses the short message space such that $m \in \{0, 1, .., 2^b\}$ [CGH01]. The Paillier cryptosystem is a generalization of the Goldwasser-Micali cryptosystem based on the quadratic residuosity problem [GM84]. [1] Okamoto and Uchiyama proposed a similar construction over the integer ring $\mathbb{Z}/(p^2q)\mathbb{Z}$, where p, q are primes [OU98].

The simplified version of the Paillier cryptosystem is proposed by Catalano et al. [CGHN01]. We call it the S-Paillier cryptosystem in this paper. The S-Paillier cryptosystem is strongly related to the RSA cryptosystem modulo n^2, where n is the RSA modulus. They choose the public key g as $g = (1+n)$, whose order in $\mathbb{Z}/n^2\mathbb{Z}$ is n. Then $g^m \bmod n^2$ is represented by $g^m = (1 + n)^m =$

[1] Recently, Cramer and Shoup proposed IND-CCA cryptosystems based on the Paillier cryptosystem or the Goldwasser-Micali cryptosystem [CS01].

$(1 + mn) \bmod n^2$. The encryption of the S-Paillier scheme is carried out by $E(m, r) = r^e(1 + mn) \bmod n^2$ for a random integer $r \in (\mathbb{Z}/n\mathbb{Z})^\times$, where e is an integer. They proved that the one-way security of the S-Paillier scheme is at least as hard as the computational small e-root problem (C-SR) problem, which computes $x \in \mathbb{Z}/n\mathbb{Z}$ from given $x^e \bmod n^2$. They also proved that the semantic security of the S-Paillier scheme is as hard as to solve the decisional small e-root problem (D-SR) problem, which decides whether $y \in \mathbb{Z}/n^2\mathbb{Z}$ is represented as $y = x^e \bmod n^2$ for $x \in \mathbb{Z}/n\mathbb{Z}$.

Contributions of This Paper

In this paper we investigate the security of the S-Paillier cryptosystem. At first we prove that the one-way security of the S-Paillier cryptosystem is as intractable as the standard RSA problem. Let an adversary A be an algorithm that breaks the one-wayness of the S-Paillier cryptosystem. We construct an adversary, which can compute the least significant bit of x for given $x^e \bmod n$, where $x \in_R (\mathbb{Z}/n\mathbb{Z})^\times$. An integer c of $\mathbb{Z}/n^2\mathbb{Z}$ is uniquely represented as $c = [c]_0+n[c]_1$, where $0 \le [c]_0, [c]_1 < n$. The adversary A can compute $[x^e \bmod n^2]_1$ for a given $x^e \bmod n$. Then the difference between $[2^{-e}x^e \bmod n^2]_1$ and $A(2^{-e}x^e \bmod n)$ gives us the information about the least significant bit of x. Moreover, we prove that an adversary, which breaks the semantic security of the S-Paillier cryptosystem, can compute the least significant bits of the nonce r. This observation is interesting, because the least significant bit of the nonce r is the hard core bit of $E(m, r) \bmod n$. The adversary is equivalent to solving the D-SR problem and can learn the least significant bit of r by multiplying $2^{-e} \bmod n^2$ with y.

We also propose a general conversion technique, which enhances the RSA cryptosystem to be semantically secure using a one-way function f, where f is a function $\mathbb{Z}/n\mathbb{Z} \to \mathbb{Z}/n\mathbb{Z}$. A message m is encrypted by $(c_0 = r^e \bmod n, c_1 = f(r) + mc_0 \bmod n)$. We analyze the requirements for the one-way function. The computational RSA + one-way function (C-RSA+OW) problem is to find $f(r)$ for a given $r^e \bmod n$. The decisional RSA+OW (D-RSA+OW) problem is to distinguish the distribution $(r^e \bmod n, f(r))$ from the uniform distribution. The converted scheme is one-way if and only if the C-RSA+OW problem of f is hard, and it is semantically secure if and only if the D-RSA+OW of f is hard. The S-Paillier cryptosystem uses the one-way function $f(r) : r \bmod n \to [r^e]_1$. We also discuss the relationship between the converted scheme and the Pointcheval cryptosystem using the dependent RSA problem [Poi99]. The Pointcheval cryptosystem encrypts a message m by $(c_0 = r^e \bmod n, c_1 = m(r + 1)^e \bmod n)$. We generalized this encryption to $(c_0 = r^e \bmod n, c_1 = mf(r) \bmod n)$ and its security has the same properties as the above conversion.

Moreover, we propose a novel one-way function $f^{e,n}_{MSBZ(l)}(r) = (r - MSB_l(r))^e \bmod n$, where $r - MSB_l(r)$ makes the l most significant bits of r zero for a large enough l. The computational RSA+MSBZ problem is to find the $f^{e,n}_{MSBZ(l)}(r)$ from a given $r^e \bmod n$. The RSA+MSBZ problem is different from the dependent RSA problem, because we do not know the $MSB_l(r)$ of $(r - MSB(r))^e$ mod

n and there is no dependence between $r^e \bmod n$ and $f^{e,n}_{MSBZ(l)}(r)$. The decisional RSA+MSBZ problem is to distinguish $(r^e \bmod n, f^{e,n}_{MSBZ(l)}(r))$ from the uniform distribution. We prove that the computational RSA+MSBZ is as intractable as the standard RSA problem. An adversary, which breaks the decisional RSA+MSBZ problem, can break the least significant bits of the computational RSA+MSBZ problem. These security results are similar to those of the S-Paillier, but the encryption/decryption of our proposed cryptosystem are more efficient than those of the S-Paillier.

Notation. In this paper we choose $\{0, 1, 2, .., m-1\}$ as the reduced residue class of modulo m, namely the elements of $\mathbb{Z}/m\mathbb{Z}$ are $\{0, 1, 2, .., m-1\}$.

2 Simplified Paillier Cryptosystem

In this section we review the simplified Paillier (S-Paillier) cryptosystem proposed by Catalano et al. [CGHN01]. The S-Paillier cryptosystem is related to the RSA cryptosystem over $\mathbb{Z}/n^2\mathbb{Z}$. The description of the S-Paillier cryptosystem in this paper is a little different from the paper [CGHN01]. Indeed we use the standard RSA key. Let RSA_{public} be the set of the RSA modulus n and the RSA encryption exponent e of n, respectively.

$$RSA_{public} = \{(n, e) | n \leftarrow \text{RSA modulus}, e \leftarrow \mathbb{Z}_{>2}, s.t. \gcd(e, \varphi(n)) = 1\} \qquad (1)$$

We explain the S-Paillier cryptosystem in the following.

Key generation: Let $(n, e) \leftarrow_R RSA_{public}$. The integer d is computed by $ed = 1 \bmod \varphi(n)$. Then (n, e) is the public key and d is the secret key.
Encryption: Let $m \in \mathbb{Z}/n\mathbb{Z}$ be a message. We generate a random integer $r \in (\mathbb{Z}/n\mathbb{Z})^\times$ and encrypt the message m by $c = r^e(1 + mn) \bmod n^2$.
Decryption: At first r is recovered with computing $r = c^d \bmod n$. Then the message m is decrypted by $m = L(cr^{-e} \bmod n^2)$, where $L(k) = (k-1)/n$.

Remark 1. In the key generation we do not assume $\gcd(e, \varphi(n^2)) = 1$ as described in the paper [CGHN01], which guarantees that the function $r \to r^e \bmod n^2$ is a permutation function over $\mathbb{Z}/n^2\mathbb{Z}$. The difference to the RSA exponent e is the condition $\gcd(e, n) = 1$. This condition is not necessary for the decryption of the S-Paillier cryptosystem and does not affect its security, as we will prove in the next section. Moreover, the probability that an integer is relatively prime to the primes p or q is upper-bounded $1/p + 1/q$, where $n = pq$. When we randomly choose the exponent e from RSA_{public}, the probability is negligible in $\log n$.

The problem of breaking the one-wayness of the S-Paillier cryptosystem is to find the integer m for given $(n, e) \leftarrow RSA_{public}$, $r \leftarrow (\mathbb{Z}/n\mathbb{Z})^\times$, and $r^e(1 + mn) \bmod n^2$. The one-wayness assumption of the S-Paillier cryptosystem is that for any probabilistic polynomial time algorithm $A^{OW}_{S-Paillier}$ the probability

$$Pr_{m \in_R \mathbb{Z}/n\mathbb{Z}}[(n, e) \leftarrow RSA_{public}, r \leftarrow_R (\mathbb{Z}/n\mathbb{Z})^\times, \\ c = r^e(1 + mn) \bmod n^2 : A^{OW}_{S\text{-}Paillier}(c) = m]$$

is negligible in $\log n$. Catalano et al. proposed a number theoretic problem in order to investigate the one-wayness of the S-Paillier cryptosystem. They defined the computational small e-roots (C-SR) problem, which is to find the integer $r \in (\mathbb{Z}/n\mathbb{Z})^\times$ for given $(n,e) \leftarrow RSA_{public}$ and $r^e \bmod n^2$. The computational small e-root (C-SR) assumption is as follows: for any probabilistic polynomial time algorithm $A_{C\text{-}SR}$ the probability

$$Pr_{r \in_R (\mathbb{Z}/n\mathbb{Z})^\times}\left[(n,e) \leftarrow RSA_{public}, c = r^e \bmod n^2 : A_{C\text{-}SR}(c) = r\right]$$

is negligible in $\log n$. It is clear that the one-wayness of the S-Paillier cryptosystem can be solved by the oracle that solves the C-SR problem. However the opposite direction is unknown and there is possibility of breaking the one-wayness of the S-Paillier scheme without solving the C-SR problem.

We explain the semantic security of the S-Paillier cryptosystem. A semantic security adversary $A^{SS}_{S\text{-}Paillier}$ consists of the find stage $A^{SS1}_{S\text{-}Paillier}$ and the guess stage $A^{SS2}_{S\text{-}Paillier}$. The $A^{SS1}_{S\text{-}Paillier}$ outputs two messages m_0, m_1 and a state information st for a public-key n. Let c be a ciphertext of either m_0 or m_1. The $A^{SS1}_{S\text{-}Paillier}$ guesses whether the ciphertext c is the encryption of $m_b (b \in \{0,1\})$ for given (c, m_0, m_1, st) and outputs b. The semantic security of the S-Paillier cryptosystem is that for any probabilistic polynomial time algorithm $A^{SS}_{S\text{-}Paillier}$ the probability

$$\begin{aligned} 2Pr\ \ &[(n,e) \leftarrow RSA_{public}, (m_0, m_1, st) \leftarrow A^{SS1}_{S\text{-}Paillier}(e,n), b \leftarrow \{0,1\}, \\ &r \leftarrow_R (\mathbb{Z}/n\mathbb{Z})^\times, c = r^e(1+m_b n) \bmod n^2 : A^{SS2}_{S\text{-}Paillier}(c, m_0, m_1, st) = b] - 1 \end{aligned}$$

is negligible in $\log n$. The semantic security of the S-Paillier cryptosystem is related to the decisional version of the C-SR problem, which distinguishes whether an element of $\mathbb{Z}/n^2\mathbb{Z}$ comes from the distribution $\{r^e \bmod n^2 | r \in (\mathbb{Z}/n\mathbb{Z})^\times\}$. The decisional small e-residue (D-SR) assumption is defined as follows: for any probabilistic polynomial time algorithm $A_{D\text{-}SR}$ the probability of distinguishing the two distributions

$$\begin{aligned} &|Pr[x \leftarrow (\mathbb{Z}/n^2\mathbb{Z})^\times : A_{D\text{-}SR}(x) = 1] \\ &\quad -Pr[x \leftarrow (\mathbb{Z}/n\mathbb{Z})^\times, y = x^e \bmod n : A_{D\text{-}SR}(y) = 1]| \end{aligned}$$

is negligible in $\log n$. Catalano et al. proved that the S-Paillier cryptosystem is semantically secure if and only if the D-SR assumption holds.

2.1 One-Wayness of the S-Paillier Scheme

The one-wayness is the simplest requirement for public-key cryptosystems. We prove that the one-way security of the S-Paillier scheme is as intractable as the RSA problem. The RSA problem is to find the integer $r \in (\mathbb{Z}/n\mathbb{Z})^\times$ for given $(n,e) \leftarrow RSA_{public}$ and $r^e \bmod n$. The RSA assumption is as follows: for any probabilistic polynomial time algorithm A_{RSA}, the probability

$$Pr_{r \in_R (\mathbb{Z}/n\mathbb{Z})^\times}[(n,e) \leftarrow RSA_{public}, c = r^e \bmod n : A_{RSA}(c) = r] \qquad (2)$$

is negligible in $\log n$.

We define a novel problem in order to investigate the security of the S-Paillier cryptosystem. We denote an element c of $\mathbb{Z}/n^2\mathbb{Z}$ by the unique n-adic representation so that $c = [c]_0 + n[c]_1$, where $0 \leq [c]_0, [c]_1 < n$. The RSA approximation (RSAaprx) problem is to find the integer $[r^e]_1$ for given $(n, e) \leftarrow RSA_{public}$ and $[r^e]_0 = r^e \bmod n$. The value $[r^e]_1$ is the first approximation of the n-adic representation of $r^e \bmod n^2 = [r^e]_0 + n[r^e]_1$. A similar problem is discussed for the ESIGN [Oka90].

The RSA approximation (RSAaprx) assumption is as follows: for any probabilistic polynomial time algorithm $A_{RSAaprx}$ the probability

$$Pr_{r \in_R (\mathbb{Z}/n\mathbb{Z})^\times} [(n, e) \leftarrow RSA_{public}, c = r^e \bmod n : A_{RSAaprx}(c) = [r^e]_1] \quad (3)$$

is negligible in $\log n$.

The RSAaprx problem and the S-Paillier cryptosystem are nicely related. Indeed, we prove the following theorem.

Theorem 1. *The encryption function of the S-Paillier cryptosystem is one-way if and only if the RSAaprx assumption holds.*

Proof. Note that we can compute the value $[r^e]_0 = r^e \bmod n$ from the ciphertext $c = r^e(1 + mn) \bmod n^2$. If the RSAaprx assumption is not true, we can find the first approximation $[r^e]_1$ from $[r^e]_0$ and we obtain $r^e \bmod n^2 = [r^e]_0 + n[r^e]_1$. Thus we can break the one-wayness of the S-Paillier cryptosystem by computing $m = k/n, k = c(r^e)^{-1} - 1 \bmod n^2$. On the contrary, assume that there is an algorithm $A^{OW}_{S\text{-}Paillier}$, which breaks the one-wayness of the S-Paillier cryptosystem. We will construct an algorithm $A_{RSAaprx}$, which breaks the RSAaprx problem using the algorithm $A^{OW}_{S\text{-}Paillier}$. Let $b = a^e \bmod n$ be a random ciphertext of the RSA cryptosystem for (n, e) as the input of the algorithm $A_{RSAaprx}$. The algorithm $A_{RSAaprx}$ works as follows:

1. $A_{RSAaprx}$ generates a random t in $\mathbb{Z}/n\mathbb{Z}$ and computes $c = b + nt$.
2. $A_{RSAaprx}$ runs $A^{OW}_{S\text{-}Paillier}(c)$ and obtains the message m of c.
3. $A_{RSAaprx}$ outputs $t - bm \bmod n$.

In step 1, the algorithm $A_{RSAaprx}$ generates a random number $t \in \mathbb{Z}/n\mathbb{Z}$ and computes $c = b+nt$. The distribution of c is equivalent to that of the ciphertext of the S-Paillier cryptosystem. Indeed, the ciphertext of the S-Paillier cryptosystem is represented by $[r^e]_0 + ([r^e]_1 + [r^e]_0 m)n$, and the value $([r^e]_1 + [r^e]_0 m)$ is uniformly distributed over $\mathbb{Z}/n\mathbb{Z}$, because the message m is uniformly distributed over $\mathbb{Z}/n\mathbb{Z}$ and $\gcd([r^e]_0, n) = 1$. Thus, in step 2, the algorithm $A_{RSAaprx}$ finds the message m for input $c = b + nt$. Finally, in step 3, the algorithm $A_{RSAaprx}$ outputs $[a^e]_1$ by computing $[a^e]_1 = t - [a^e]_0 m = t - bm \bmod n$, which is the first approximation of $a^e \bmod n^2$.

It is obvious that the RSAaprx problem can be solved by the oracle that solves the RSA problem. Indeed, for inputs $r^e \bmod n$ and $(e, n) \leftarrow RSA_{public}$, the oracle can find the integer r. Then $[r^e]_1$ can be easily computed by $[r^e]_1 = k/n$ for $k = (r^e \bmod n^2) - r^e \bmod n$. The opposite direction is not trivial. However, we prove the following theorem.

Theorem 2. *The RSAaprx assumption holds if and only if the RSA assumption holds.*

Proof. Let $A_{RSAaprx}$ be an algorithm, which solves the RSAaprx problem with advantage ε in time t. We will construct an algorithm $A_{RSA\text{-}LSB}$, which finds the least significant bit of the RSA problem with advantage ε^2 in time $2t+\mathcal{O}((\log n)^3)$ using the algorithm $A_{RSAaprx}$. Let $b_0 = r^e \bmod n$ be a random ciphertext of the RSA cryptosystem for (n, e) as the input of the algorithm $A_{RSA\text{-}LSB}$. The algorithm $A_{RSA-LSB}$ works as follows:

1. $A_{RSA\text{-}LSB}$ runs $A_{RSAaprx}(b_0)$ and obtains the first approximation b_1 of b_0.
2. $A_{RSA\text{-}LSB}$ computes $a_0 = b_0 2^{-e} \bmod n$, runs $A_{RSAaprx}(a_0)$, and obtains the first approximation a_1 of a_0.
3. $A_{RSA\text{-}LSB}$ returns 1 as the least significant bit of r, if $a_0 + na_1 = 2^{-e}(b_0 + nb_1) \bmod n^2$ holds, otherwise it returns 0.

In step 1, the algorithm $A_{RSA\text{-}LSB}$ obtains the first approximation b_1 of b_0, so that it knows $r^e \bmod n^2$. In step 2, the algorithm $A_{RSA\text{-}LSB}$ computes $a_0 = b_0 2^{-e} \bmod n$ and obtains the first approximation a_1 of a_0, so that it knows $a_0 + na_1$. In step 3, the algorithm $A_{RSA\text{-}LSB}$ compares the two values $(2^{-1}r)^e \bmod n^2$ and $a_0 + na_1$. Note that $2^{-1} \equiv \frac{n^2+1}{2} \bmod n^2$ and $\frac{n^2+1}{2} = \frac{n+1}{2} + n\frac{n-1}{2}$. Thus if $\gcd(e, n) = 1$ we have the following relations:

$$\begin{aligned}
LSB(r) = 0 &\Leftrightarrow (2^{-1}r)^e \bmod n^2 = (r/2)^e \bmod n^2 \\
&\Leftrightarrow a_0 + na_1 = (r/2)^e \bmod n^2, \\
LSB(r) = 1 &\Leftrightarrow (2^{-1}r)^e \bmod n^2 = \left(\frac{r+n}{2} + n\frac{n-1}{2}\right)^e \bmod n^2 \\
&\Leftrightarrow a_0 + na_1 = \left(\frac{r+n}{2}\right)^e \bmod n^2,
\end{aligned}$$

where $LSB(r)$ is the least significant bit of r. The probability $\gcd(n, e) = 1$ is upper-bounded by the negligible probability $(1/p + 1/q)$. Thus, in the step 3, we have $2^{-e}(b_0 + nb_1) \bmod n^2 = a_0 + na_1$ if and only if the least significant bit of r is equal to 0.

We estimate the advantage and the time of the algorithm $A_{RSA\text{-}LSB}$ in the following. In step 1 and step 2 the algorithm $A_{RSAaprx}$ is used as an oracle, and in step 2 and step 3 two modular exponentiations are computed. The advantage and the time of the algorithm $A_{RSA\text{-}LSB}$ are ε^2 and $2t + \mathcal{O}((\log n)^3)$, respectively. Next, Fishlin and Schnorr proved that the RSA problem is solved in time $\mathcal{O}((\log n)^2\varepsilon^{-2}t + (\log n)^2\varepsilon^{-6})$ using an oracle that predicts the least significant bit with advantage ε and in time t [FS00]. Thus the algorithm $A_{RSA\text{-}LSB}$ solves the RSA problem in time $\mathcal{O}((\log n)^2\varepsilon^{-4}t + (\log n)^5\varepsilon^{-4} + (\log n)^2\varepsilon^{-12})$. When we choose ε^{-1} as the polynomial of $\log n$, the time becomes the polynomial time in $\log n$. Thus we have proven the theorem.

From theorem 1 and theorem 2 we have proven that the encryption function of the S-Paillier cryptosystem is one-way if and only if the standard RSA assumption holds.

2.2 Semantic Security of the S-Paillier Cryptosystem

Let c be the ciphertext of either the message m_0 or m_1. Loosely speaking, if the cryptosystem is semantically secure, any adversary can not distinguish whether the ciphertext c is the encryption of m_0 or m_1 with more than negligible probability. Several public-key cryptosystems have been proven semantically secure under a standard model [OU98], [CS98], [Pai99], [Poi99]. The reduced number-theoretic problems are not computational problems but decisional problems, e.g. the decisional p-subgroup problem, the decisional Diffie-Hellman problem, the decisional n-residue problem, and the decisional dependent-RSA problem. The difficulties of these decisional problems have not been studied well. Then a new number-theoretic problem, the so called Gap problem, has been proposed [OP01]. The Gap problem is a problem to solve the computational problem with the help of its decisional problem. Several fundamental security problems can be reduced to the Gap problem. To investigate the relation between the computational problem and its decisional problem is an important problem.

Catalano et al. proved that the semantic security of the S-Paillier cryptosystem is as hard as the decisional small e-root problem (D-SR) problem [CGHN01]. In this section we study how to relate the D-SR problem with the C-SR problem. We can prove the following theorem:

Theorem 3. *Let $(n, e) \leftarrow RSA_{public}$ and $c = r^e \bmod n^2 (0 \leq r < n)$ be the inputs of the computational small e-root problem. An adversary, which breaks the decisional small e-root problem, can compute the least significant bit of r. If the least significant bits of r are zero, the next bit after the zeros can be compute by the adversary.*

Proof. Let $A_{D\text{-}SR}$ be an adversary, which solves the D-SR problem. We can assume that with non-negligible probability the adversary $A_{D\text{-}SR}$ answers $A_{D\text{-}SR}(y) = 1$ if y is the small e-root residue, and it answers $A_{D\text{-}SR}(y) = 0$ otherwise. We will construct an algorithm A_{LSB}, which computes the least significant bit of r using algorithm $A_{D\text{-}SR}$. The algorithm A_{LSB} works as follows:

1. A_{LSB} computes $y = 2^{-e} c^e \bmod n^2$.
2. A_{LSB} runs algorithm $A_{D\text{-}SR}(y)$, and obtains $b = A_{D\text{-}SR}(y)$
3. A_{LSB} returns $\overline{b}$.

In step 1 the integer y is computed as $y = 2^{-e} c \bmod n^2$. As we showed in the proof of theorem 2, the least significant bit of r is 0 if and only if $2^{-1} r \bmod n^2 = r/2$, and it is 1 if and only if $2^{-1} r \bmod n^2 = \frac{r+n}{2} + \frac{n-1}{2} n$. Therefore, the least significant bit of r is 0 if and only if y is the small e-root residue, and it is 1 if and only if y is not the small e-root residue. Thus, the output $\overline{b}$ of the A_{LSB} is the least significant bit of r. If the k least significant bits r are zero, $(r/2^k)^e = 2^{-ke} r^e \bmod n^2$ is the small e-root residue. We can detect the $(k+1)$-th bit of r using the above algorithm.

The S-Paillier cryptosystem encrypts a message m by $c = r^e(1+mn) \bmod n^2$ where $r \in (\mathbb{Z}/n\mathbb{Z})^\times$. By the result of Catalano et al., the adversary $A_{D\text{-}SR}$, which breaks the D-SR problem, can break the semantic security of the S-Paillier cryptosystem [CGHN01]. Then, we can obtain the $r^e \bmod n^2$. With the theorem the least significant bit of the nonce r can be computed by invoking $A_{D\text{-}SR}$. Thus the least significant bit of r can be computed. If the k-th least significant bits of r is zero, then we learn the $(k+1)$-th bit of r. Thus, we have proven the following corollary.

Corollary 1. *An adversary, which breaks the decisional small e-root problem, can compute the least significant bit of the nonce r of the S-Paillier cryptosystem. If the least significant bits of r are zero, then the adversary can compute the next bit after the zeros.*

This observation is interesting because the least significant bits of the nonce r is the hard core bit of the ciphertext $c \bmod n$.

3 General Conversion of the RSA Cryptosystem

In this section we generalize the encryption mechanism of the S-Paillier cryptosystem to a general RSA-type encryption scheme. We discuss the one-way security and the semantic security of the general RSA-type encryption scheme. The scheme is also related to the dependent RSA cryptosystem proposed by Pointcheval [Poi99]. Moreover, we propose a novel cryptosystem based on the most significant bits zero problem.

The S-Paillier cryptosystem encrypts a message m by $c = r^e(1+mn) \bmod n^2$ where $(n,e) \leftarrow RSA_{public}$ and r is a random integer in $(\mathbb{Z}/n\mathbb{Z})^\times$. If we represent the ciphertext c as the n-adic expansion $c = [c]_0 + [c]_1 n$, where $0 \leq [c]_0, [c]_1 < n$, we have the following relationship:

$$[c]_0 = r^e \bmod n, \quad [c]_1 = [r^e]_1 + mr^e \bmod n. \tag{4}$$

The message is randomized by the value $[r^e]_1$. Let f be a function $f : r \rightarrow [r^e]_1$ for $r \in (\mathbb{Z}/n\mathbb{Z})^\times$. We proved that computing the value $f(r)$ from $r^e \bmod n$ is as hard as breaking the RSA problem.

Our proposed scheme uses a general one-way function $f : \mathbb{Z}/n\mathbb{Z} \rightarrow \mathbb{Z}/n\mathbb{Z}$ instead of the function $r \rightarrow [r^e]_1$. The proposed scheme is as follow:

Key generation: Let $(n,e) \leftarrow_R RSA_{public}$. The integer d is computed by $ed = 1 \bmod \varphi(n)$. Then (n,e) is the public key and d is the secret key. Moreover, we use a one-way function $f : \mathbb{Z}/n\mathbb{Z} \rightarrow \mathbb{Z}/n\mathbb{Z}$ as a system parameter.
Encryption: Let $m \in \mathbb{Z}/n\mathbb{Z}$ be a message. We generate a random integer $r \in (\mathbb{Z}/n\mathbb{Z})^\times$ and encrypt the message m by $c_0 = r^e \bmod n$ and $c_1 = f(r) + mc_0 \bmod n$. The ciphertext is (c_0, c_1).
Decryption: At first r is recovered by $r = c_0^d \bmod n$. Then the message m is decrypted by $m = (c_1 - f(r))c_0^{-1} \bmod n$.

We call this scheme the G-RSA cryptosystem in this paper and define several assumptions of this G-RSA cryptosystem. Let OW be a class of the one-way function $\mathbb{Z}/n\mathbb{Z} \to \mathbb{Z}/n\mathbb{Z}$. The one-wayness assumption of the G-RSA cryptosystem is that, for any probabilistic polynomial time algorithm $A^{OW}_{G\text{-}RSA}$, the probability

$$\begin{aligned} &Pr_{m \in_R (\mathbb{Z}/n\mathbb{Z})}[(n,e) \leftarrow RSA_{public}, f \leftarrow \mathsf{OW}, r \leftarrow_R (\mathbb{Z}/n\mathbb{Z})^{\times}, \\ &\qquad c_0 = r^e \bmod n, c_1 = f(r) + mc_0 \bmod n : A^{OW}_{G\text{-}RSA}((c_0, c_1)) = m] \end{aligned}$$

is negligible in $\log n$. A semantic security adversary $A^{SS}_{G\text{-}RSA}$ against the G-RSA cryptosystem consists of the find stage $A^{SS1}_{G\text{-}RSA}$ and the guess stage $A^{SS2}_{G\text{-}RSA}$. The semantic security of the G-RSA cryptosystem is that, for any probabilistic polynomial time algorithm $A^{SS}_{G\text{-}RSA}$, the probability

$$\begin{aligned} 2Pr\ &[(n,e) \leftarrow RSA_{public}, f \leftarrow \mathsf{OW}, (m_0, m_1, st) \leftarrow A^{SS1}_{G\text{-}RSA}(e,n), \\ &b \leftarrow \{0,1\}, r \leftarrow_R (\mathbb{Z}/n\mathbb{Z})^{\times}, c_0 = r^e \bmod n, \\ &\qquad c_1 = f(r) + m_b c_0 \bmod n^2 : A^{SS2}_{G\text{-}RSA}((c_0,c_1), m_0, m_1, st) = b] - 1 \end{aligned}$$

is negligible in $\log n$.

3.1 Security of the G-RSA Cryptosystem

We define the following two problems in order to investigate the security of the G-RSA cryptosystem based on a one-way function $f : \mathbb{Z}/n\mathbb{Z} \to \mathbb{Z}/n\mathbb{Z}$. The computational RSA + one-way function (C-RSA+OW) problem is to compute the value $f(r)$ for a given RSA public-key (e,n) and a ciphertext $r^e \bmod n$. The C-RSA+OW assumption is as follows: for any probabilistic polynomial time algorithm $A_{C\text{-}RSA+OW}$, the probability

$$\begin{aligned} Pr_{r \in_R (\mathbb{Z}/n\mathbb{Z})^{\times}}\ &[(n,e) \leftarrow RSA_{public}, f \leftarrow \mathsf{OW}, \\ &\qquad c = r^e \bmod n^2 : A_{C\text{-}RSA+OW}(c) = f(r)] \end{aligned}$$

is negligible in $\log n$. The decisional version of the C-RSA+OW problem is to distinguish whether an element $(x,y) \in \mathbb{Z}/n\mathbb{Z} \times \mathbb{Z}/n\mathbb{Z}$ comes from the distribution $(r^e \bmod n, f(r))$ for $r \in (\mathbb{Z}/n\mathbb{Z})^{\times}$. The decisional RSA + one-way function (D-RSA+OW) assumption is defined as follows: for any probabilistic polynomial time algorithm $A_{D\text{-}RSA+OW}$, the probability of distinguishing the two distributions

$$\begin{aligned} &|Pr[(x,y) \leftarrow \mathbb{Z}/n\mathbb{Z} \times \mathbb{Z}/n\mathbb{Z} : A_{D\text{-}RSA+OW}(x,y) = 1] - Pr[r \leftarrow (\mathbb{Z}/n\mathbb{Z})^{\times}, \\ &\quad x = r^e \bmod n, f \leftarrow \mathsf{OW}, y = f(r) : A_{D\text{-}RSA+OW}(x,y) = 1]| \end{aligned}$$

is negligible in $\log n$.

The one-way security and the semantic security are as intractable as the C-RSA+OW problem and the D-RSA+OW problem, respectively. These properties can be proved by applying the same techniques used in theorem 1 and for the S-Paillier cryptosystem [CGHN01], respectively. The statements are as follows:

Theorem 4. *The encryption function of the G-RSA cryptosystem is one-way if and only if the C-RSA+OW assumption holds.*

Theorem 5. *The G-RSA cryptosystem is semantically secure if and only if the D-RSA+OW assumption holds.*

We can recognize that the G-RSA cryptosystem is a conversion technique, which enhances the security of the RSA cryptosystem to be semantically secure. If we find a one-way function, whose C-RSA+OW problem and D-RSA+OW problem are intractable, then we can construct a semantically secure encryption scheme. The S-Paillier cryptosystem is an example of the G-RSA cryptosystem, whose security is based on the RSA+RSAaprx problem.

To find another one-way function for the G-RSA cryptosystem is a quite difficult problem. Consider the function $f : r \to r^2 \bmod n$. Computing $f(r)$ for a given $r^e \bmod n$ is as hard as the RSA problem, because $r = (r^e)^x(r^2)^y \bmod n$ holds for integers x, y such that $ex+2y = 1$. Therefore this C-RSA+OW problem of the function f is as intractable as the RSA problem. However, the distribution (r^e, r^2) can be distinguished from the random distribution using the same gcd computation, and this D-RSA+OW problem is easily broken.

3.2 Relation to the Pointcheval Cryptosystem

The Pointcheval public-key cryptosystem encrypts a message m by $c_0 = r^e \bmod n$ and $c_1 = m(r + 1)^e \bmod n$ [Poi99]. The one-way security of the Pointcheval scheme is based on the difficulty of computing $(r + 1)^e \bmod n$ for given $(n, e) \leftarrow RSA_{public}$ and $r^e \bmod n$, which is called the computational dependent RSA (C-DpdRSA) problem. The semantic security of the Pointcheval scheme is as hard as to distinguishes the distribution $(r^e \bmod n, (r+1)^e \bmod n)$ from the uniform distribution, which is called the decisional dependent RSA (D-DpdRSA) problem. The one-way function that the Pointcheval scheme uses is $r \to (r + 1)^e \bmod n$ for $r \in (\mathbb{Z}/n\mathbb{Z})^\times$.

We can generalize the Pointcheval scheme using a general one-way function $f : \mathbb{Z}/n\mathbb{Z} \to \mathbb{Z}/n\mathbb{Z}$ instead of the function $r \to (r+1)^e \bmod n$. The encryption is carried out as follows:

$$c_0 = r^e \bmod n, \quad c_1 = mf(r) \bmod n. \tag{5}$$

The difference to the G-RSA cryptosystem is to mask the message m using $c_1 = mf(r) \bmod n$ instead of $c_1 = f(r) + mc_0 \bmod n$. In the same manner as in the previous section, we can prove that the one-way security and semantic security of the generalized Pointcheval scheme is as intractable as solving the C-RSA+OW problem and the D-RSA+OW problem, respectively. Our generalized Pointcheval cryptosystem is another conversion technique, which depends on the same security requirements as the G-RSA cryptosystem.

We can also choose different conversion forms like $c_1 = m(f(r)+c_0) \bmod n$ or $c_1 = f(r)+m \bmod n$, whose one-way and semantic security are equivalent to the

C-RSA+OW and the D-RSA+OW, respectively. Let g be a function, which can efficiently compute both $c_1 = g(m, f(r), c_0)$ and $m = g^{-1}(c_1, f(r), c_0)$. Then we can generate the secure conversion analogue to the G-RSA cryptosystem using the function g.

If we apply the one-way function of the S-Paillier cryptosystem to the Pointcheval conversion, the converted encryption scheme encrypts a message m by $c_0 = r^e \bmod n$ and $c_1 = m[r^e]_1 \bmod n$. This encryption method is also observed by Catalano et al. [CGHN01].

3.3 A New One-Way Function

We propose a new one-way function, which is provably secure in the sense of the RSA+OW problem. We prove that the C-RSA+OW problem of the proposed one-way function is as intractable as the standard RSA problem. We prove that an adversary, which breaks the D-CRSA+OW problem of the proposed one-way function, can break the least significant bits of the C-RSA+OW problem.

We explain our new one-way function in the following. Let r be a k-bit random integer in $(\mathbb{Z}/n\mathbb{Z})^\times$. The binary presentation of r is $r = r_0 2^0 + r_1 2^1 + ... + r_{k-1}2^{k-1}$, where $r_{k-1} = 1$. Denote by $MSB_l(r)$ the l-bit upper part of r such that $r_{k-l}2^{k-l} + r_{k-l+1}2^{k-l+1} + ... + r_{k-1}2^{k-1}$. The proposed one-way function is defined by

$$f^{e,n}_{MSBZ(l)}(r) = (r - MSB_l(r))^e \bmod n, \tag{6}$$

where l is large enough. The l most significant bits of r are chosen as zeros by $r - MSB_l(r)$. We call the one-way function the RSA most significant bits zero (MSBZ) function. Micali and Schnorr proposed a similar one-way function, which is used for a parallel generation of a pseudo random number generator [MS88].

The computational RSA+MSBZ (C-RSA+MSBZ) problem is to compute $f^{e,n}_{MSBZ(l)}(r)$ for a given $r^e \bmod n$, where r is a random integer r in $(\mathbb{Z}/n\mathbb{Z})^\times$. The C-RSA+MSBZ assumption is that, for any probabilistic polynomial time algorithm $A_{C\text{-}RSA+MSBZ}$, the probability

$$Pr_{r \in_R (\mathbb{Z}/n\mathbb{Z})^\times} \ [(n,e) \leftarrow RSA_{public}, c = r^e \bmod n : \\ A_{C\text{-}RSA+MSBZ}(c) = f^{e,n}_{MSBZ(l)}(r)]$$

is negligible in $\log n$.

The RSA+MSBZ problem is different from the DpdRSA problem by the Pointcheval cryptosystem [Poi99], because $MSB_l(r)$ of the $f^{e,n}_{MSBZ(l)}(r) = (r - MSB_l(r))^e \bmod n$ is unknown. We have no known dependences between $r^e \bmod n$ and $(r - MSB_l(r))^e \bmod n$. A possible attack to break the C-RSA+MSBZ problem is to use the Coppersmith algorithm [Cop96]. The Coppersmith algorithm can find the integer r from two values $r^e \bmod n$ and $(r+t)^e \bmod n$, where t is a unknown random integer with $|t| < n^{1/e^2}$. Therefore, if l is small for small exponent e, the Coppersmith attack finds the r. When n is 1024 bits, we have to choose $l > 114$ for $e = 3$ and $l > 21$ for $e = 7$. The other attack is the low exponent attack with known related messages from Coppersmith et al. [CFPR96]. In this case an attacker computes $\gcd(x^e, (x+t)^e)$ over the polynomial ring $\mathbb{Z}/n\mathbb{Z}[x]$ for all possible t. If l is small, the attacker can find the r. Therefore, we have to make l enough large. For example, we recommend $l = 160$ for a 1024-bit RSA modulus n.

The decisional RSA+MSBZ (D-RSA+MSBZ) problem is to distinguish

$$(r^e \bmod n, f_{MSBZ(l)}(r))$$

from the uniform distribution. The D-RSA+MSBZ assumption is defined as follows: for any probabilistic polynomial time algorithm $A_{D\text{-}RSA+MSBZ}$, the probability to distinguish the two distributions

$$\begin{aligned}&|Pr[x, y \leftarrow \mathbb{Z}/n\mathbb{Z} \times \mathbb{Z}/n\mathbb{Z} : A_{D\text{-}RSA+MSBZ}(x, y) = 1] - Pr[r \leftarrow (\mathbb{Z}/n\mathbb{Z})^{\times},\\ &\quad c = r^e \bmod n, z = f^{e,n}_{MSBZ(l)}(r) \bmod n : A_{D\text{-}RSA+MSBZ}(c, z) = 1]|\end{aligned}$$

is negligible in $\log n$. We prove the following theorem:

Theorem 6. *The C-RSA+MSBZ assumption holds if and only if the RSA assumption holds*

Proof. The proof of the theorem is similar to that of theorem 2. We compare $2^{-e} f^{e,n}_{MSBZ(l)}(r) \bmod n$ with $f^{e,n}_{MSBZ(l)}(2^{-1}r \bmod n)$. Let $A_{C\text{-}RSA+MSBZ}$ be an adversary, which breaks the C-RSA+MSBZ assumption. Then we construct an algorithm $A_{RSA\text{-}LSB}$, which breaks the least significant bit of the RSA problem using $A_{C\text{-}RSA+MSBZ}$. Let $y = r^e \bmod n$ be a random input for the RSA problem. At first, the algorithm $A_{RSA\text{-}LSB}$ runs the adversary $A_{C\text{-}RSA+MSBZ}$ and obtains $f^{e,n}_{MSBZ(l)}(r)$. Second, it computes $w = y2^{-e} \bmod n$, runs the adversary $A_{C\text{-}RSA+MSBZ}$, and obtains $f^{e,n}_{MSBZ(l)}(2^{-1}r \bmod n)$. Finally, the algorithm $A_{RSA\text{-}LSB}$ outputs 0 if $2^{-e}\ f^{e,n}_{MSBZ(l)}(r) \bmod n = f^{e,n}_{MSBZ(l)}(2^{-1}r \bmod n)$ holds, and it outputs 1 otherwise.

The least significant bit of r (we denote it by $LSB(r)$) is zero if and only if $r/2 = 2^{-1}r \bmod n$. The $LSB(r)$ is one if and only if $(r+n)/2 = 2^{-1}r \bmod n$. Let $r' = r - MSB_l(r)$, so that $r'^e \bmod n = f^{e,n}_{MSBZ(l)}(r)$ holds. Note that $LSB(r') = LSB(r)$ and $MSB_l(a/2) = MSB_l(a)/2$ for all at least $(l+2)$-bit integers a. Here, for $LSB(r) = 0$, we have

$$2^{-1}r' \bmod n = r'/2 = (r - MSB_l(r))/2 = r/2 - MSB_l(r/2). \tag{7}$$

On the contrary, for $LSB(r) = 1$, we have $2^{-1}r' \bmod n = (r - MSB_l(r) + n)/2$, which is not equivalent to $2^{-1}r \bmod n - MSB_l(2^{-1}r \bmod n) = (r+n)/2 - MSB_l((r+n)/2)$, because of $MSB_l(r) \neq MSB_l(r+n)$. Therefore, the least significant bit of r is zero if and only if $2^{-e} f^{e,n}_{MSBZ(l)}(r) \bmod n = f^{e,n}_{MSBZ(l)}(w)$ holds.

Let ε and t be the advantage and the time of the adversary $A_{C\text{-}RSA+MSBZ}$, respectively. Then the advantage and time of algorithm $A_{RSA\text{-}LSB}$ are ε^2 and $2t + \mathcal{O}((\log n)^3)$, respectively. By the result of [FS00], the algorithm $A_{RSA\text{-}LSB}$ can solve the RSA problem. Thus we have proven the theorem.

There is a relation between the D-RSA+MSBZ problem and C-RSA+MSBZ problem. It is similar with that of the D-RSAaprx problem and the C-RSAaprx problem. We prove the following theorem.

Theorem 7. *Let $(n, e) \leftarrow RSA_{public}$ and $c = r^e \bmod n$ be the input of the computational RSA+MSBZ problem. An adversary, which breaks the decisional RSA+MSBZ problem, can compute the least significant bit of r. If the least significant bits of r are zero, the next bit after the zeros can be computed by the adversary.*

Proof. The proof of the theorem is similar to that of theorem 3. Let $A_{D\text{-}RSA+MSBZ}$ be an adversary, which solves the D-RSA+MSBZ problem. We can assume that, with non-negligible probability, the adversary $A_{D\text{-}RSA+MSBZ}$ answers $A_{D\text{-}RSA+MSBZ}(y) = 1$ if y comes from the distribution $(x^e \bmod n, f^{e,n}_{MSBZ(l)}(x))$ for an integer $x \in (\mathbb{Z}/n\mathbb{Z})^\times$, and it answers $A_{D\text{-}RSA+MSBZ}(y) = 0$ otherwise. We will construct an algorithm A_{LSB}, which computes the least significant bit of r using algorithm $A_{D\text{-}RSA+MSBZ}$. At first the algorithm A_{LSB} computes $y = (2^{-e}c \bmod n,\ 2^{-e}f^{e,n}_{MSBZ(l)}\ (r) \bmod n)$. Second, A_{LSB} runs algorithm $A_{D\text{-}RSA+MSBZ}(y)$ and obtains $b = A_{D\text{-}RSA\ +MSBZ}(y)$. Finally, A_{LSB} returns $\overline{b}$. As we showed in the proof of theorem 6, the least significant bit of r is 0 if and only if y comes from the distribution $(x^e \bmod n, f^{e,n}_{MSBZ(l)}(x))$ for an integer $x \in (\mathbb{Z}/n\mathbb{Z})^\times$. If the k least significant bits of r are zero, $(r/2^k)^e = 2^{-ke}r^e \bmod n$ is the image of $f^{e,n}_{MSBZ(l)}(r/2^k)$. We can detect the $(k+1)$-th bit of r using the above algorithm.

By these theorems, the G-RSA cryptosystem using the one-way function $f^{e,n}_{MSBZ(l)}$ has similar security conditions as the S-Paillier cryptosystem. An adversary, which breaks the D-RSA+MSBZ problem, can compute the least significant bits of the nonce of the G-RSA cryptosystem with $f^{e,n}_{MSBZ(l)}$.

3.4 Comparison

We compare the public-key cryptosystems discussed in this paper, i.e., the Paillier cryptosystem, the S-Paillier cryptosystem, the Pointcheval cryptosystem, and the proposed cryptosystem. The security and the efficiency of these cryptosystems are compared. For the efficiency we count the number of modular exponentiations in the encryption and decryption process, because the computation of the modular exponents is dominant for the efficiency. Denote by $ME(k)$ a modular exponentiation modulo k. We assume that these cryptosystems use the same length RSA keys n. In table 1 we indicate the comparison.

The Paillier cryptosystem encrypts a message m by $c = g^m r^n \bmod n^2$ and decrypts it by $m = L(c^{\varphi(n)} \bmod n^2)/L(g^{\varphi(n)} \bmod n^2)$, where $L(u) = (u-1)/n$. The encryption and the decryption of the Paillier cryptosystem require two $ME(n^2)$ and two $ME(n^2)$, respectively. If we use the key $g = 1 + n$, the encryption requires only one $ME(n^2)$. If we precompute the g, the decryption needs only one $ME(n^2)$. The one-way security and the semantic security are as hard as the C-CR problem and the D-CR problem, respectively.

Table 1. Comparison of security and efficiency among several schemes

	Paillier	S-Paillier	Pointcheval	Proposed scheme
One-wayness	C-CR	**RSA**	C-DpdRSA	**RSA**
Semantic security	D-CR	**D-RSA+RSAaprx**	D-DpdRSA	**D-RSA+MSBZ**
Encryption	1 $ME(n^2)$	1 $ME(n^2)$	2 $ME(n)$	**2** $\boldsymbol{ME(n)}$
Decryption	1 $ME(n^2)$	1 $ME(n^2)$ + 1 $ME(n)$	2 $ME(n)$	**2** $\boldsymbol{ME(n)}$

The S-Paillier cryptosystem requires one $ME(n^2)$ for the encryption $c = r^e(1 + nm) \bmod n^2$, and one $ME(n)$ and one $ME(n^2)$ for the decryption $m = L(cr^{-e} \bmod n^2)$ and $r = c^d \bmod n$. Note that a small encryption exponent e can be used. We proved that the one-way security and the semantic security are as hard as the RSA problem and the D-RSA+RSAaprx problem, respectively. We also proved that an adversary, which breaks the semantic security, can compute the least significant bits of the nonce r.

The Pointcheval cryptosystem requires two $ME(n)$ for the encryption $c_0 = r^e \bmod n$ and $c_1 = m(r + 1)^e \bmod n$, and two $ME(n)$ for the decryption $m = c_1(r+1)^{-e} \bmod n$ and $r = c_0^d \bmod n$. Note that a small exponent e is not secure for this scheme because of the message related attack and e must be at least 32 bits [CFPR96]. The one-way security and the semantic security are as hard as the C-DpdRSA problem and the D-DpdRSA problem, respectively.

The proposed cryptosystem encrypts a message m by $c_0 = r^e \bmod n, c_1 = f^{e,n}_{MSBZ(l)}(r) + mc_0 \bmod n$ and decrypts it by $m = (c_1 - f^{e,n}_{MSBZ(l)}(r))c_0^{-1} \bmod n, r = c_0^d \bmod n$, where $f^{e,n}_{MSBZ(l)}(r) = (r - MSBZ_l(r))^e \bmod n$. Therefore, the encryption and the decryption of the proposed cryptosystem require two $ME(n)$ and two $ME(n)$, respectively. The computation time of the function $f^{e,n}_{MSBZ(l)}(r)$ is about 4 times faster than that of the S-Paillier cryptosystem $r^e \bmod n^2$. Thus the encryption/decryption of the proposed cryptosystem are more efficient than those of the S-Paillier cryptosystem. Because a small exponent key e can be used for a enough large l, the proposed cryptosystem is more efficient than the Pointcheval cryptosystem. We proved that the one-way security and the semantic security are as hard as the RSA problem and the D-RSA+MSBZ problem, respectively. An adversary, which breaks the semantic security, can break the least significant bits of the nonce r. The proposed cryptosystem has the similar security properties as the S-Paillier cryptosystem.

Acknowledgments

We would like to thank Holger Vogt for his valuable comments, and the anonymous referees of PKC 2002 for their helpful comments.

References

BDPR98. M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway, Relations among notions of security for public-key encryption schemes," CRYPTO'98, LNCS 1462, (1998), pp.26-45.

CGH01. D. Catalano, R. Gennaro, and N. Howgraw-Graham; "The bit security of Paillier's encryption scheme and its applications," Eurocrypt 2001, LNCS 2045, pp.229-243, 2001.

CGHN01. D. Catalano, R. Gennaro, N. Howgrave-Graham, and P. Nguyen; "Paillier's cryptosystem revisited," to appear in the ACM conference on Computer and Communication Security, 2001.

Cop96. D. Coppersmith, "Finding a small root of a univariate modular equation," EUROCRYPT '96, LNCS 1070, pp.155–165, 1996.

CFPR96. D. Coppersmith, M. Franklin, J. Patarin, M. Reiter, "Low-exponent RSA with related messages," EUROCRYPT '96, LNCS 1070, (1996), pp.1-9.

CS98. R. Cramer and V. Shoup, "A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack," CRYPTO'98, LNCS 1462, pp.13-25, 1998.

CS01. R. Cramer and V. Shoup, "Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-Key encryption ," Cryptology ePrint Archive, IACR, `http://eprint.iacr.org/`, 2001.

DJ01. I. Damgård and M. Jurik; "A generalization, a simplification and some applications of Paillier's probabilistic public-Key system," PKC 2001, LNCS 1992, pp.119-136, 2001.

FS00. R. Fischlin and C.P. Schnorr; "Stronger security proofs for RSA and Rabin bits," Journal of Cryptology, 13 (2), pp.221-244, 2000.

GM84. S. Goldwasser and S. Micali; "Probabilistic encryption," Journal of Computer and System Science, Vol.28, No.2, pp.270-299, 1984.

MS88. S. Micali and C. Schnorr, "Efficient, perfect random number generators," Crypto'88, LNCS 403, pp.173-199, 1988.

Oka90. T. Okamoto; "A fast signature scheme based on congruential polynomial operations," IEEE Transactions on Information Theory, IT-36, pp.47-53, 1990.

OP01. T. Okamoto and D. Pointcheval, "The Gap-Problems: a new class of problems for the security of cryptographic schemes," PKC 2001, LNCS 1992, pp.104-118, 2001.

OU98. T. Okamoto and S. Uchiyama; "A new public-key cryptosystem as secure as factoring," Eurocrypt'98, LNCS 1403, pp.308-318, 1998.

Pai99. P. Paillier; "Public-key cryptosystems based on composite degree residuosity classes," Eurocrypt'99, LNCS 1592, pp.223-238, 1999.

Poi99. D. Pointcheval, "New public key cryptosystems based on the dependent-RSA problems," Eurocryt'99, LNCS 1592, pp. 239-254, 1999.

Tak97. T. Takagi, "Fast RSA-type cryptosystems using n-adic expansion," CRYPTO '97, LNCS 1294, pp.372-384, 1997.