暗号化関数とその性質について — RSA 関数と Paillier 関数

Takato Hirano [*] Keisuke Tanaka [*]

Abstract— Paillier proposed an additively homomorphic encryption scheme which is known as a variant of the RSA encryption scheme. There are few variants on discussion for decreasing computational costs of the Paillier encryption function. In this paper, we study fast inversion of the Paillier encryption function. Especially, we do not modify the encryption function and focus on key generation. We propose two schemes: Scheme 1 is based on the Euclidean Extended Algorithm, and can make small inversion keys. Scheme 2 is based on the factorization, and can make small sparse inversion keys. We also analyze the security of the variants against known attacks.

Keywords: Paillier's encryption scheme, fast inversion, key generation, factoring, security.

1 Introduction

The RSA encryption scheme [20], proposed by Rivest, Shamir, and Adleman in 1978, is the most widely used public-key encryption scheme. Paillier [18] proposed an additively homomorphic encryption scheme which is known as a variant of the RSA encryption scheme. Additive homomorphism is a good property which can be applied to many cryptographic applications such as electronic voting, electronic cash, and so on. However, for almost all of the proposed public-key encryption schemes, including the RSA and Paillier schemes, the computational costs of encryption and decryption are relatively large compared to the symmetric-key encryption. Therefore, it is important to decrease these costs of the public-key schemes.

As a well-known way in the RSA encryption scheme, the Chinese Remainder Theorem (CRT) is useful for decreasing the decryption costs. By applying CRT to the RSA encryption scheme, the computational cost over the public modulus N can be reduced to that over the private prime factors p and q, and the private exponent d can also be replaced by $d_p = d \bmod (p-1)$ and $d_q = d \bmod (q-1)$. We call such RSA schemes "RSA-CRT". As a result, the decryption costs of RSA-CRT are about 4 times as much as those of Standard-RSA.

As another practical way, one arbitrarily chooses the public exponent e in the key generation phase, in order to decrease the encryption costs (for example, $e = 3$ or $2^{16}+1$). Then, the private exponent d is determined uniquely from the private (randomly chosen) prime factors p and q and the public exponent e, and has about the same size as $\varphi(N)$ or $\lambda(N)$, where φ is the Euler phi-function and λ is the Carmichael function.

As an alternative approach, one arbitrarily chooses the private exponent d or the private CRT exponents (d_p, d_q) in the key generation phase, and computes the public exponent e from (p, q, d) or (p, q, d_p, d_q), in order to decrease the decryption costs. Then, the public exponent e has about the same size as $\varphi(N)$ or $\lambda(N)$. As a well-known fact, the system with d or (d_p, d_q) which are much smaller than N becomes insecure as follows.

In 1990, Wiener [27] showed that by using continued fractions, one could easily compute the secret key d in polynomial-time from the public key e and N such that $d \leq N^{0.25}$. In 1999, Boneh and Durfee [2] improved Wiener's bound to $d \leq N^{0.292}$ with Coppersmith's lattice-based techniques [5] on finding small modular and integer roots of (bivariate) polynomials via Lenstra-Lenstra-Lovász's lattice reduction algorithm [17]. Although their attack requires a few heuristic assumptions, the attack works very well in practice. In 2007, Jochemsz and May [14] proposed an attack to the private CRT exponents such that either d_p and d_q are smaller than $N^{0.0734}$. The attack is based on Boneh-Durfee's lattice-based attack and also requires a few appropriate assumptions.

As mentioned above, in the RSA encryption scheme, decreasing the encryption costs is easy but the decryption costs heavily, and vice versa. To reduce computational costs of encryption and decryption simultaneously is not easy. Therefore, it is an interesting problem to solve this situation.

On the other hand, there are few variants [4, 3] on discussion for decreasing computational costs of the Paillier encryption scheme, although the Paillier encryption scheme is known as a variant of the RSA encryption scheme and has more computational costs in encryption and decryption than those of the RSA encryption scheme. However, these variants employ modified encryption functions which lose a few math-

[*] Department of Mathematical and Computing Sciences, Tokyo Institute of Technology. W8-55, 2-12-1 Ookayama, Meguro-ku, Tokyo, 152-8552, Japan. {hirano6, keisuke}@is.titech.ac.jp. This research was supported in part by NTT Information Sharing Platform Laboratories and JSPS Global COE program "Computationism as a Foundation for the Sciences".

ematical structures and advantages.

1.1 Related Works

Up to now, many fast variants with decreasing both the encryption and decryption costs of the RSA encryption scheme have been proposed [25, 24, 23, 22, 8, 11]. These variants are on methods for the key generation, and the encryption and decryption functions are not improved. On the other hand, lattice-based attacks by Boneh-Durfee's idea to the variants have also been studied [7, 1, 13].

In 1999, Sun, Yang, and Laih [25] proposed three variants with small public and private exponents (e, d) and unbalanced private prime factors (p, q) such that $p \lll q$, which are resisted to Wiener's or Boneh-Durfee's attacks. Basic idea of their variants is as follows: First, choose (small) integers e and d as the public and private exponents. Second, compute some integers p and q from (e, d), by using the Extended Euclidean Algorithm. If p and q are prime, then set the public key (N, e) and the private key (p, q, d). Moreover, the above method via the Extended Euclidean Algorithm generates only unbalanced primes p and q. Unfortunately, Durfee and Nguyen [7] proposed an efficient attack based on Boneh-Durfee's lattice-based attack to their recommended parameters by using trivariate modular polynomials and these spanned lattices. They stated that unbalanced private factors should not be used in the RSA encryption scheme. In fact, Bleichenbacher and May [1] showed that given an RSA modulus N, whose private prime factor p is smaller than $N^{0.468}$, it can be factorized in polynomial-time.

By factoring much smaller numbers than N, Sun and Yang [24] improved Sun et al.'s variant [25] to generating balanced primes p and q. Hence, the attacks [7, 1] on unbalanced primes cannot be applied to the variant. Furthermore, Sun, Hinek, and Wu [22] extended this improved variant in such a way that the private CRT exponents (d_p, d_q) are also small (this variant is a revised version of [23]). On the other hand, Bleichenbacher and May proposed, in addition to unbalanced private prime factors, an attack to the variants [24, 23, 22]. However, this attack is critical to their recommended parameters, but not to the basis of their systems.

Independently, Galbraith, Heneghan, and McKee [8] proposed another fast variant which uses no technical method such as factoring small numbers. They just reconsidered some relations among the public and private keys. This variant finds not only small public and private CRT exponents (e, d_p, d_q), but also with low Hamming weight. Repeated squaring method, which is widely used as a computational method for exponentiation, strongly depends on the binary representation of the exponent. If the binary representation is sparse (that is, low Hamming weight), computational costs of the method are reduced in practice. In a similar to the Sun et al.'s variants [24, 23, 22], the attack by Bleichenbacher and May [1] can be applied to the variant, and is not critical to the base of their variant.

Hinek [11] reconsidered the common prime RSA, proposed by Wiener [27], whose prime factors p and q of the RSA modulus N have a relation that $g = \gcd(p-1, q-1)$ is not small. If g is large, $\lambda(N) = \mathrm{lcm}(p-1, q-1)$ is small, and then, in the common prime RSA, the public and private exponents e and d which are elements of $(\mathbb{Z}/\lambda(N))^{\times}$ are also small. He pointed out that the common prime RSA with private exponents smaller than $N^{0.25}$ has resistant to Wiener and the other known lattice-based attacks, when g is large. On the other hand, Jochemsz and May [13] studied an attack to his proposed key space in the common prime RSA.

1.2 Our Contribution

In this paper, we study fast inversion of the Paillier encryption function. We do not modify the encryption function and focus on key generation, since the original encryption function has rich mathematical structures and advantages. We propose two schemes: Scheme 1 is based on the Euclidean Extended Algorithm, and can make small inversion keys. Scheme 2 is based on the factorization of much smaller numbers than the public RSA modulus N, and can make small sparse inversion keys. We note that Scheme 2 is much slower than Scheme 1 since Scheme 2 is constructed by using factoring algorithms. Furthermore, we analyze the security of the schemes against known attacks and propose security parameters choices.

1.3 Organization

The organization of this paper is as follows. In Section 2, we give some notations and attacks to two variants of the RSA encryption scheme. In Section 3, we review the Paillier encryption scheme and its variant "Paillier-CRT". In Section 4, we focus on some relations between the public and private keys in the Paillier encryption function, and propose fast inversion variants on key generation. In Section 5, we discuss on fast encryption, the security of the variants, and security parameters choice.

2 Preliminaries

2.1 Notations

Let N be a positive integer. We denote $\{0, 1, \ldots, N-1\}$ by $\mathbb{Z}/N$, and its reduced residue class group by $(\mathbb{Z}/N)^{\times}$, namely, $(\mathbb{Z}/N)^{\times} = \{x \in \mathbb{Z}/N \mid \gcd(x, N) = 1\}$. For $g \in (\mathbb{Z}/N)^{\times}$, $\mathrm{ord}_N\, g$ is defined as the smallest positive integer e such that $g^e \equiv 1 \pmod{N}$.

Let k be a positive integer. We denote the set of k-bit integers by $\mathcal{N}_k$, and the set of k-bit prime numbers by $\mathcal{P}_k$.

2.2 Attacks for Variants of the RSA Encryption Scheme

We briefly describe attacks to two variants "RSA-CRT" and "RSA-CRT with Known Difference $d_p - d_q$" of the RSA encryption scheme. We refer to the paper [12] for more details of the attacks.

Lattice-Based Attack on RSA-CRT: Jochemsz and May [14] proposed a polynomial-time attack to RSA-CRT if d_p and d_q are smaller than $N^{0.073}$. So far, the best attack on RSA-CRT is a square-root attack that enables an adversary to factor N in time and space $\tilde{O}(\min\{\sqrt{d_p}, \sqrt{d_q})\}$, which is exponential in the bit length of d_p or d_q. Jochemsz-May's attack is as follows:

theorem 1 *For every $\epsilon > 0$ and sufficiently large n, the following holds: Let N be an n-bit modulus, and p, q be primes of the bit length $\frac{n}{2}$. Let $e < \varphi(N)$, $d_p < p - 1$, and $d_q < q - 1$ be the public and private CRT exponents satisfying $ed_p \equiv 1 \pmod{p-1}$ and $ed_q \equiv \pmod{q-1}$. Let the bit length of d_p and d_q be smaller than δn. Then N can be factored in polynomial-time provided that $\delta < 0.0734 - \epsilon$.*

Lattice-Based Attack on RSA-CRT with Known Difference: From a storage point of view, Qiao and Lam [19] proposed a variant of the RSA encryption scheme, whose CRT-exponents d_p and d_q has given small difference $c \in \mathbb{Z}$. Jochemsz and May [13] proposed a polynomial-time attack to the variant if d_p or d_q are smaller than $N^{0.099}$. The attack is as follows:

theorem 2 *For every $\epsilon > 0$ and sufficiently large n, the following holds: Let N be an n-bit modulus, and p, q be primes of the bit length $\frac{n}{2}$. Let $e < \varphi(N)$, $d_p < p - 1$, and $d_q < q - 1$ be the public and private CRT exponents satisfying $ed_p \equiv 1 \pmod{p-1}$ and $ed_q \equiv \pmod{q-1}$. Assume that d_p and d_q are chosen such that $d_p = d_q + c$ for given c, and that the bit length of d_p and d_q be smaller than δn. Then N can be factored in polynomial-time provided that $\delta < \frac{1}{4}(4 - \sqrt{13}) - \epsilon$. Notice that $\frac{1}{4}(4 - \sqrt{13}) \sim 0.099$.*

3 The Paillier Encryption Scheme

In 1999, Paillier [18] proposed the public-key encryption scheme with the additively homomorphic property which can be applied to many cryptographic applications. Several variants and applications of the Paillier encryption scheme have been well-studied. The based encryption function is as follows.

Definition 3 *The Paillier encryption function $\mathcal{E}$ is as follows:*

$$\begin{array}{ccc}(\mathbb{Z}/N)^\times \times \mathbb{Z}/N & \longrightarrow & (\mathbb{Z}/N^2)^\times \\ (r, m) & \longmapsto & r^N g^m \bmod N^2,\end{array}$$

where g is an element of $(\mathbb{Z}/N^2)^\times$ and $\mathrm{ord}_{N^2}\, g = aN$ $(1 \le a \le \lambda(N)$ and $a \mid \lambda(N))$.

For the sake of simplicity, we usually use $g = 1 + N$. Then, the Paillier encryption scheme is as follows.

Cryptosystem 4 *(The Original Paillier Encryption Scheme)*

Key Generation: *Given a security parameter n, choose $n/2$-bit primes p and q at random, and set $N = pq$. Compute the Carmichael function λ of N (that is, $\lambda(N) = \mathrm{lcm}(p-1, q-1)$). Then, the public key is $pk = N$ and the secret key is $sk = \lambda(N)$.*

Encryption: *To encrypt a message $m \in \mathbb{Z}/N$, choose $r \in (\mathbb{Z}/N)^\times$ at random, and compute the ciphertext $c \in (\mathbb{Z}/N^2)^\times$ such that*

$$c = \mathcal{E}(r, m) = r^N(1 + N)^m \bmod N^2.$$

Decryption: *To obtain the message $m \in \mathbb{Z}/N$, compute $y = c^{\lambda(N)} \bmod N^2$ and*

$$m = L_N(y)\lambda^{-1}(N) \bmod N,$$

where $L_N(x) = \frac{x-1}{N}$ for $x \in \mathbb{Z}$.

This scheme is secure in the sense of IND-CPA under the decisional composite residuosity assumption that there is no polynomial-time algorithm which solves the following "the decisional composite residuosity problem" with non-negligible advantage.

Definition 5 *(The Decisional Composite Residuosity Problem)*
Let N be a randomly chosen n-bit pq modulus. For a probabilistic polynomial-time algorithm $\mathcal{A}$, we define the following probabilities:

$$P_{Random} = \Pr_x[x \leftarrow (\mathbb{Z}/N^2)^\times : \mathcal{A}(x) = 1]$$

and

$$P_{Residue} = \Pr_x[x \leftarrow (\mathbb{Z}/N)^\times : \mathcal{A}(x^N \bmod N^2) = 1].$$

Then, we denote an advantage of $\mathcal{A}$ by

$$\mathbf{Adv}_{\mathcal{A}}^{DCR}(n) = |P_{Random} - P_{Residue}|.$$

In application settings of the Paillier encryption scheme (such as Trapdoor Commitment Scheme [3] and Paillier-OAEP [10], and so on), we often need to extract random numbers used in the scheme. Computing random numbers in the Paillier encryption scheme is equivalent to solving the RSA(N, N) problem which is given an RSA modulus N and $c \in (\mathbb{Z}/N)^\times$, to compute an N-th root of c modulo N, that is, $c^{\frac{1}{N}}$ (mod N). Moreover, computing inversion of the Paillier encryption function is equivalent to solving the RSA(N, N) problem.

As a well-known fact, $(1+N)^m \equiv 1+mN \pmod{N^2}$ for any $m \in \mathbb{Z}/N$. In other words, the computational cost for computing $(1+N)^m$ over $(\mathbb{Z}/N^2)^\times$, which is $O(\lg^3 N)$ in general, can be reduced to $O(\lg^2 N)$.

In addition to decreasing the encryption costs, CRT can be applied to the Paillier encryption scheme in order to decrease the decryption costs. We call such Paillier's schemes "Paillier-CRT". Now, we describe Paillier-CRT with extracting random numbers as follows.

Cryptosystem 6 *(Paillier-CRT with Extracting Random Numbers)*

Key Generation: *Given a security parameter n, choose $n/2$-bit primes p and q at random, and set $N = pq$. Compute $\lambda(N) = \mathrm{lcm}(p-1, q-1)$, $(u, v) \in \mathbb{Z}^2$ such that $up + vq = 1$, and $d \in \mathbb{Z}/\lambda(N)$ such that $d \equiv N^{-1} \pmod{\lambda(N)}$. Let $d_p = d \bmod (p-1)$ and $d_q = d \bmod (q-1)$. Then, the public key is $pk = N$ and the secret key is $sk = (p, q, u, v, d_p, d_q, \lambda(N))$.*

Encryption: *To encrypt a message $m \in \mathbb{Z}/N$, choose $r \in (\mathbb{Z}/N)^\times$ at random, and compute the ciphertext $c \in (\mathbb{Z}/N^2)^\times$ such that*

$$c = \mathcal{E}(r, m) = r^N(1+mN) \bmod N^2.$$

Decryption: *To obtain the message $m \in \mathbb{Z}/N$, compute $(y_p, y_q) = (c^{p-1} \bmod p^2, c^{q-1} \bmod q^2)$ and $(M_p, M_q) = (L_p(y_p), L_q(y_q))$. Then,*

$$m = -(vM_p + uM_q) \bmod N.$$

To extract the random number $r \in (\mathbb{Z}/N)^\times$, compute $(r_p, r_q) = (c^{d_p} \bmod p, c^{d_q} \bmod q)$. Then,

$$r = vqr_p + upr_q \bmod N.$$

We call the private CRT exponents (d_p, d_q) "the inversion keys" in the Paillier encryption function.

4 The Proposed Variants on Key Generation

In this section, we focus on fast inversion of the Paillier encryption function. Especially, we point out fast extracting random numbers used in the scheme. As a similar fashion to the variants [22, 8] of the RSA encryption scheme, CRT can be also applied to the Paillier encryption scheme, in order to decrease the decryption costs.

In the Paillier encryption scheme, the secret key $d \in (\mathbb{Z}/\lambda(N))^\times$, which is used for computing random numbers, is defined as $dN \equiv 1 \pmod{\lambda(N)}$, where $\lambda(N) = \mathrm{lcm}(p-1, q-1)$. From a CRT point of view, we obtain the following equations:

$$\begin{aligned} Nd_p &\equiv 1 \pmod{(p-1)}, \\ Nd_q &\equiv 1 \pmod{(q-1)}. \end{aligned}$$

Notice that the converse is not true by $\gcd(p-1, q-1) \geq 2$. Over the rational integer ring $\mathbb{Z}$, the equations can be represented as

$$Nd_p = 1 + k_p(p-1), \qquad (1)$$
$$Nd_q = 1 + k_q(q-1), \qquad (2)$$

for some integers k_p and k_q.

As a simple (and faulty) way, since the equation (1) is equivalent to

$$q = \frac{1 + k_p(p-1)}{pd_p}$$

over the rational number field $\mathbb{Q}$, we substitute it for the equation (2). We have the following equation:

$$pd_q\frac{1+k_p(p-1)}{pd_p} = 1 + k_q(\frac{1+k_p(p-1)}{pd_p} - 1).$$

Then, the equation is quadratic on p:

$$d_qk_pp^2 + (d_pk_q - d_qk_p - k_pk_q - d_p + d_q)p + k_pk_q - k_q = 0.$$

With well-known formula for quadratic equations over $\mathbb{R}$, all solutions of the equation (4) can be formed by

$$p = \frac{-A \pm \sqrt{B}}{2d_qk_p}$$

over the real field $\mathbb{R}$, where $A = d_pk_q - d_qk_p - k_pk_q - d_p + d_q$ and $B = (d_pk_q - d_qk_p - k_pk_q - d_p + d_q)^2 - 4d_qk_p(k_pk_q - k_q)$. Since p is prime over $\mathbb{Z}$, it must be satisfied the following three conditions:

1. B is square of some integers, that is, $\sqrt{B} \in \mathbb{Z}$.
2. $-A - \sqrt{B}$ or $-A + \sqrt{B}$ is a positive integer.
3. $2d_qk_p$ divides either $-A - \sqrt{B}$ or $-A + \sqrt{B}$.

However, to construct algorithms satisfying such conditions could be harder. Moreover, there might be no guarantee of polynomial-time, as far as designing them via the integers (d_p, d_q, k_p, k_q). We note that if one follows the original key generation of the Paillier encryption scheme, then p and q are always satisfy the three conditions.

To avoid the argument above, we use the following technique. Since $N = pq = (p-1+1)q \equiv q \pmod{(p-1)}$ and $N \equiv p \pmod{(q-1)}$, we can reduce the equation (1) and (2) to as follows:

$$\begin{aligned} qd_p &= 1 + k_p(p-1), & (3) \\ pd_q &= 1 + k_q(q-1). & (4) \end{aligned}$$

The equation (3) over $\mathbb{Z}$ is equivalent to

$$q = \frac{1 + k_p(p-1)}{d_p}$$

over $\mathbb{Q}$, and we substitute it for the equation (4). We have the following equation:

$$pd_q = 1 + k_q(\frac{1 + k_p(p-1)}{d_p} - 1).$$

Then, the equation is linear on p, described as follows:

$$p = \frac{k_pk_q + d_pk_q - k_q - d_p}{k_pk_q - d_pd_q}. \qquad (5)$$

Therefore, the three conditions, as mentioned before, are reduced to the following condition:

$$k_pk_q + d_pk_q - k_q - d_p \equiv 0 \pmod{(k_pk_q - d_pd_q)}. \qquad (6)$$

In a similar fashion, the condition on q is as follows:

$$q = \frac{k_pk_q + d_qk_p - k_p - d_q}{k_pk_q - d_pd_q}. \qquad (7)$$

However, to construct algorithms satisfying the condition (6) could be harder yet since it might be large both integers $k_pk_q + d_pk_q - k_q - d_p$ and $k_pk_q - d_pd_q$. In order to design efficient algorithms for the condition (6), we restrict $k_pk_q - d_pd_q$ to as follows:

$$k_pk_q - d_pd_q = 1. \qquad (8)$$

Then, $k_pk_q + d_pk_q - k_q - d_p$ is always divided by $k_pk_q - d_pd_q = 1$. Security of this restriction will be discussed in Section 5.2 and 5.3.

Let ℓ_{d_p}, ℓ_{d_q}, ℓ_{k_p}, and ℓ_{k_p} be the bit lengths of d_p, d_q, k_p, and k_q, respectively. Since $k_pk_q - d_pd_q = 1$, we obtain that

$$\ell_{d_p} + \ell_{d_q} \sim \ell_{k_p} + \ell_{k_p}. \qquad (9)$$

The bit lengths of primes p and q, denoted by ℓ_p and ℓ_q respectively, satisfy the following relations.

$$\begin{aligned} \ell_p &\sim \ell_{k_q} + \max\{\ell_{k_p}, \ell_{d_p}\}, & (10) \\ \ell_q &\sim \ell_{k_p} + \max\{\ell_{k_q}, \ell_{d_q}\}. & (11) \end{aligned}$$

Then, with the relations (9), (10), and (11), the bit length of the RSA modulus $N = pq$, which is the security parameter n, is described as

$$\begin{aligned} n &= \ell_p + \ell_q \\ &= \ell_{d_p} + \ell_{d_q} + \max\{\ell_{d_p} + \ell_{d_q}, \ell_{d_q} + \ell_{k_p}, \ell_{d_p} + \ell_{k_q}\}. \end{aligned}$$

We note that ℓ_p and ℓ_q must have almost the same size, since there is the polynomial-time attack [1] on factoring N whereas $p < N^{0.468}$ and $p < q$.

4.1 Scheme 1

Scheme 1 is based on the Extended Euclidean Algorithm. The following is fundamental theorem in number theory, which is the key idea of Scheme 1.

Lemma 7 *Let a and b be integers such that* $\gcd(a, b) = 1$. *For any integer h, there exists a unique and efficient computable integers (u_h, v_h) such that $au_h + bv_h = 1$, where $(h-1)b < u_h < hb$ and $(h-1)a < v_h < ha$.*

We are interesting in $h = 2$. Scheme 1 is as follows:

Cryptosystem 8 *The following key generation algorithm takes integers (ℓ_{d_p}, ℓ_{k_p}) and outputs (p, q, d_p, d_q).*

1. *Choose an integer d_p of ℓ_{d_p} bits.*
2. *Choose an integer k_p of ℓ_{k_p} bits.*
3. *By using Lemma 7 with $h = 2$, compute integers d_q and k_q (which are ℓ_{k_p} and ℓ_{d_p} bits, respectively) such that $k_pk_q + d_pd_q = 1$.*
4. *Compute $p = k_pk_q + d_pk_q - k_q - d_p$.*
5. *If p is composite, go to 1.*
6. *Compute $q = k_pk_q + d_qk_p - k_p - d_q$.*
7. *If q is composite, go to 1.*
8. *Return (p, q, d_p, d_q).*

4.2 Scheme 2

Scheme 2 is based on factoring of much smaller than N. We refer to the book [6] for more details of factoring algorithms. Furthermore, in Scheme 2 we can decide the difference $d_p - d_q$ for saving storage. Scheme 2 is as follows:

Cryptosystem 9 *The following key generation algorithm takes integers (c, ℓ_{d_p}) and outputs (p, q, d_p).*

1. *Choose an integer d_p of ℓ_{d_p} bits (with sparse).*
2. *Compute an integer $d_q = d_p + c$.*
3. *By using factoring algorithms such as the general number field sieve, factorize $d_pd_q + 1$ and assign its factors to integers k_p and k_q.*

4. *Compute* $p = k_p k_q + d_p k_q - k_q - d_p$.
5. *If* p *is composite, go to 1.*
6. *Compute* $q = k_p k_q + d_q k_p - k_p - d_q$.
7. *If* q *is composite, go to 1.*
8. *Return* (p, q, d_p, d_q).

Although Scheme 2 uses factoring algorithms which are sub-exponential time, it is feasible in practice. If N is of 1024 bits, then the target factoring number $d_p d_q+1$ is of about 512 bits, which can be factorized in practical time. The assign to k_p and k_q in Step 3 is fixed in Section 5.3.

5 Discussion

In this section, we discuss on fast encryption, the security of the variants, and security parameters choice.

5.1 Fast Encryption

We have seen fast inversion variants of the Paillier encryption function. On the other hand, from a storage point of view, compression and short expression of RSA modulus N have been studied [26, 16, 21, 15, 9]. Main idea of these papers is, given an integer s, to find an RSA modulus N such that $N = pq$ and $N = s \parallel t$ for some integer t and primes p and q. Moreover, it is interesting in the setting $s = 0$. This means that the RSA modulus N would be sparse. In fact, this approach can be applied to fast encryption of the Paillier encryption scheme, since we usually use Repeated squaring method for computing modular exponentiation. However, it might be hard to combine our proposed schemes with taking sparse RSA moduli approach.

5.2 The Security of Our Variants

In order to avoid the attack to RSA-CRT, which is as mentioned in Section 2.2, d_p and d_q in Scheme 1 must be larger than $N^{0.0734}$. Similarly, in order to avoid the attack to RSA-CRT with known difference which is as mentioned in Section 2.2, d_p or d_q in our Scheme 2 must be larger than $N^{0.099}$.

5.3 Security Parameters Choice

At first, we give a property on factoring integers, which can factorize two arbitrary numbers for given an integer, with high probability.

Lemma 10 *For non-negative integers* k, s, t *such that* $s + t = k$, *let* $\mathcal{N}_k(s,t)$ *be a set of k-bit integers whose elements have s-bit and k-bit integer factors, that is, for an element* $x \in \mathcal{N}_k(s,t)$, *there exist* y *and* z *such that* $x = yz$, *where* y *and* z *are s-bit and t-bit (composite or prime) numbers, respectively. Then the number of* $\mathcal{N}_k(s,t)$, *denoted by* $\#\mathcal{N}_k(s,t)$, *is asymptotically*

$$\#\mathcal{N}_k(s,t) \geq \frac{2^{k-4}}{st \ln^2 2}.$$

In particular, the ratio of $\mathcal{N}_k(s,t)$ *to* $\mathcal{N}_k$ *is at least* $\frac{1}{8st \ln^2 2}$.

We skip the details of the proof due to the space limitation. We refer to the full version of this paper. The proof is based on the Prime Number Theory.

Remark 11 *In fact, we can easily obtain an improved bound*

$$\begin{aligned} \#\mathcal{N}_k(s,t) &\geq \#\tilde{\mathcal{N}}_s \#\tilde{\mathcal{N}}_t \\ &= (2^s - 2^{s-\frac{1}{2}})(2^t - 2^{t-\frac{1}{2}}) \\ &> 2^{s-2} 2^{t-2} = 2^{k-4}, \end{aligned}$$

which is exact but not asymptotic, hence, also obtain an improved ratio $> \frac{1}{8}$, *where* $\tilde{\mathcal{N}}_i = \{x \in \mathbb{Z} \mid 2^{i-\frac{1}{2}} \leq x < 2^i\}$ *for* $i \in \mathbb{Z}$. *However, these bound and ratio include the possibility that there are some composite numbers with special forms such as smooth numbers.*

Then, by applying Lemma 10, we propose security parameters choice as follows. For the security parameter n, we recommend to set

$$\ell_{d_p} = \ell_{d_q} = \ell_{k_p} = \ell_{k_q} = \frac{n}{4}.$$

Then, when N is of 1024 bits, d_p and d_q are of 256 bits. Therefore we can avoid the attacks in Section 2.2.

References

[1] Daniel Bleichenbacher and Alexander May. New Attacks on RSA with Small Secret CRT-Exponents. *PKC 06*, pages 1–13, 2006.

[2] Dan Boneh and Glenn Durfee. Cryptanalysis of RSA with Private Key d Less than $N^{0.292}$. *IEEE Transactions on Information Theory*, pages 1339–1349, 2000.

[3] Dario Catalano, Rosario Gennaro, Nick Howgrave-Graham, and Phong Q. Nguyen. Paillier's cryptosystem revisited. *CCS 01*, pages 206–214, 2001.

[4] Dug-Hwan Choi, Seungbok Choi, and Dongho Won. Improvement of Probabilistic Public Key Cryptosystem using Discrete Logarithm. *ICISC 01*, pages 72–80, 2001.

[5] Don Coppersmith. Small Solutions to Polynomial Equations, and Low Exponent RSA Vulnerabilities. *Journal of Cryptology*, pages 233–260, 1997.

[6] Richard Crandall and Carl Pomerance. *Prime Numbers: A Computational Perspective.* Springer, 2005. Second Edition.

[7] Glenn Durfee and Phong Q. Nguyen. Cryptanalysis of the RSA Schemes with Short Secret Exponent from Asiacrypt'99. *ASIACRYPT 00*, pages 14–29, 2000.

[8] Steven D. Galbraith, Chris Heneghan, and James F. McKee. Tunable Balancing of RSA. *ACISP 05*, pages 280–292, 2005.

[9] Sidney W. Graham and Igor E. Shparlinski. On RSA Moduli with Almost Half of the Bits Prescribed. *Discrete Applied Mathematics*, pages 3150–3154, 2008.

[10] Ryotaro Hayashi and Keisuke Tanaka. Anonymity on Paillier's Trap-Door Permutation. *ACISP 07*, pages 200–214, 2007.

[11] M. Jason Hinek. Another Look at Small RSA Exponents. *CTRSA 06*, pages 82–98, 2006.

[12] Ellen Jochemsz. *Cryptanalysis of RSA Variants Using Small Roots of Polynomials.* PhD thesis, Technische Universiteit Eindhoven, 2007.

[13] Ellen Jochemsz and Alexander May. A Strategy for Finding Roots of Multivariate Polynomials with New Applications in Attacking RSA Variants. *ASIACRYPT 06*, pages 267–282, 2006.

[14] Ellen Jochemsz and Alexander May. A Polynomial Time Attack on RSA with Private CRT-Exponents Smaller Than $N^{0.0734}$. *CRYPTO 07*, pages 395–411, 2007.

[15] Marc Joye. RSA Moduli with a Predetermined Portion: Techniques and Applications. *ISPEC 08*, pages 116–130, 2008.

[16] Arjen K. Lenstra. Generating RSA Moduli with a Predetermined Portion. *ASIACRYPT 99*, pages 1–10, 1999.

[17] Arjen K. Lenstra, Hendrik W. Lenstra, and Lászl6 Lovász. Factoring Polynomials with Rational Coefficients. *Mathematische Annalen*, pages 515–534, 1982.

[18] Pascal Paillier. Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. *EUROCRYPT 99*, pages 223–238, 1999.

[19] Guopei Qiao and Kwok-Yan Lam. Rsa signature algorithm for microcontroller implementation. *CARDIS 98*, pages 353–356, 1998.

[20] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A Method for Obtaining Digital Signatures and Public-key Cryptosystems. *Communications of the ACM*, pages 120–126, 1978.

[21] Igor E. Shparlinski. On RSA Moduli with Prescribed Bit Patterns. *Designs, Codes and Cryptography*, pages 113–122, 2006.

[22] Hung-Min Sun, M. Jason Hinek, and Mu-En Wu. On the Design of Rebalanced RSA-CRT. Technical report, CACR 2005-35, 2005. Revised Version of [23].

[23] Hung-Min Sun and Mu-En Wu. An Approach Towards Rebalanced RSA-CRT with Short Public Exponents, 2005. Cryptology ePrint Archive: Report 2005/053.

[24] Hung-Min Sun and Cheng-Ta Yang. RSA with Balanced Short Exponents and Its Application to Entity Authentication. *PKC 05*, pages 199–215, 2005.

[25] Hung-Min Sun, Wu-Chuan Yang, and Chi-Sung Laih. On the Design of RSA with Short Secret Exponent. *ASIACRYPT 99*, pages 150–164, 1999.

[26] Scott A. Vanstone and Robert J. Zuccherato. Short RSA Keys and Their Generation. *Journal of Cryptology*, pages 101–114, 1995.

[27] Michael J. Wiener. Cryptanalysis of Short RSA Secret Exponents. *IEEE Transactions on Information Theory*, pages 553–558, 1990.