ISSN 1342-2812

Research Reports on Mathematical and Computing Sciences

Simple Decomposition of Ciphertexts

Takato Hirano, Koichiro Wada, and Keisuke Tanaka

February 2008, C–253

Department of
Mathematical and
Computing Sciences
Tokyo Institute of Technology

SERIES C: Computer Science

Simple Decomposition of Ciphertexts

Takato Hirano, Koichiro Wada, and Keisuke Tanaka

Dept. of Mathematical and Computing Sciences
Tokyo Institute of Technology
W8-55, 2-12-1 Ookayama Meguro-ku, Tokyo 152-8552, Japan
{hirano6, wada4, keisuke}@is.titech.ac.jp

February 28, 2008

Abstract

In this paper, we introduce a new cryptographic model, *the encryption scheme with decomposition of ciphertexts* which is related to the notion of the secret sharing scheme and additively homomorphism. The model is described as follows: Let $\mathcal{R}$ be a receiver, $\mathcal{S}_1, \ldots, \mathcal{S}_u$ be senders, and $\mathcal{V}_1, \ldots, \mathcal{V}_y$ be servers. By using servers $\mathcal{V}_1, \ldots, \mathcal{V}_y$ as mediators between servers and the receiver, senders $\mathcal{S}_1, \ldots, \mathcal{S}_u$ want to send information on $R(m_1, m_2, \ldots, m_u)$ without providing each messages m_i, where R is a operation (for example, $R(m_1, m_2, \ldots, m_u) = m_1 + m_2 + \cdots + m_u$). In addition, by using receiver's public key pk, each sender $\mathcal{S}_i$ wants to divide a message m_i to y shares, and then to distribute a part of y shares to each servers. Furthermore we develop with a variant of the Paillier encryption function which has several properties related to homomorphism [3]. In fact, we construct our scheme by using homomorphic properties of the encryption function and decomposition of (cyclotomic) polynomials.

Keywords: Paillier encryption function, decomposition, additively homomorphism.

1 Introduction

In cryptographic applications, it is important to manage much information efficiently, to divide secret information, and to give anonymity to players or information.

The secret sharing scheme which was proposed by Shamir is to divide a secret into shares [6]. These shares are distributed among users in a secure way. A cooperation of some of the users is able to reconstruct the secret. Classical secret sharing schemes tends to follow the traditions and sensitivities of information theory or coding theory, but not those of computational complexity. Recently, Bellare and Rogaway formalize the secret sharing scheme on the point of view of computational complexity, and they re-analyse a previous schemes which has no formal proofs [5].

The shuffle scheme takes as input an array of ciphertexts, and outputs a permuted and re-encrypted array of inputs. Re-encryption means that to generate c' from a given ciphertext c such that $\mathcal{D}(c') = \mathcal{D}(c)$, where $\mathcal{D}$ is a decryption algorithm. The mix-net scheme applies the shuffle scheme. The idea was proposed by Charm [1]. The mix-net scheme provides communication unlinkability and anonymity.

In 1999, Paillier proposed a public-key encryption scheme, which is the indistinguishability against the chosen plaintext attack (IND-CPA) under the decisional composite residuosity assumption [4]. This scheme has an additively homomorphic property, which can be applied to many cryptographic applications. There are some variants of this scheme [2, 3].

In this paper, we introduce a new cryptographic model, *the encryption scheme with decomposition of ciphertexts* which is related to the notion of the secret sharing scheme and additively

homomorphism. The model is described as follows: Let $\mathcal{R}$ be a receiver, $\mathcal{S}_1, \ldots, \mathcal{S}_u$ be senders, and $\mathcal{V}_1, \ldots, \mathcal{V}_y$ be servers. By using servers $\mathcal{V}_1, \ldots, \mathcal{V}_y$ as mediators between servers and the receiver, senders $\mathcal{S}_1, \ldots, \mathcal{S}_u$ want to send information on $R(m_1, m_2, \ldots, m_u)$ without providing each messages m_i, where R is a operation (for example, $R(m_1, \ldots, m_u)$
$= m_1 + \cdots + m_u$). In addition, by using receiver's public key pk, each sender $\mathcal{S}_i$ wants to divide a message m_i to y shares, and then to distribute a part of y shares to each servers. Intuitively, a scheme decomposition of ciphertexts efficiently transforms $\mathcal{S}_i$'s message m_i $(1 \leq i \leq u)$ into a tuple of ciphertexts $(c_{i,1}, c_{i,2}, \ldots, c_{i,y})$, concerned with $\mathcal{R}$'s public key pk, such that:

(1) With $\mathcal{R}$'s secret key sk, $(C_1, C_2, \ldots, C_y)$ reveal $R(m_1, m_2, \ldots m_u)$ but not each m_i, where C_j is $\mathcal{V}_j$'s composition of $(c_{1,j}, c_{2,j}, \ldots, c_{u,j})$.

(2) A lack of at least one of C_j reveals no information of $R(m_1, m_2, \ldots, m_u)$.

In practice, $\mathcal{S}_i$ makes ciphertexts $c_{i,j}$ $(1 \leq j \leq y)$ from own message m_i and sends $c_{i,j}$ to $\mathcal{V}_j$. $\mathcal{V}_j$ composes C_j from $c_{i,j}$s. $\mathcal{R}$ receives C_j from $\mathcal{V}_j$ for all j, and composes them. And then, $\mathcal{R}$ obtains $R(m_1, m_2, \ldots, m_u)$.

Furthermore, we develop with a variant of the Paillier encryption function which has several properties related to homomorphism [3], described as follow: The encryption function is $\mathcal{E}_t(r, m) = r^{n^s}(1 + mn^t) \bmod n^{s+1}$, where n, s, t are public key, m is a message, r is a random value. The parameter t is important to our encryption scheme. It is known that $\mathcal{E}_t$ is additively homomorphic, if t is at least $\frac{s+1}{2}$. In fact, we construct our scheme by using this fact, other homomorphic properties of the encryption function, and the decomposition of (cyclotomic) polynomials.

This paper is organized as follows: In section 2, we briefly recall a variant of Paillier encryption scheme. In section 3, we introduce a new model and construct two servers model, practically. Furthermore we discuss its security. In section 4, we discuss to construct y servers model when $y \geq 3$. In section 5, we conclude and provide some open problems.

2 Preliminaries

2.1 A Variant of the Paillier Encryption Scheme

In this section, we briefly recall a variant of Paillier encryption scheme [3]. Let $n = p^2q$, where p, q are large primes with the same length, and $s, t \in \mathbb{N}$. The encryption function $\mathcal{E}_t$ is the following function:

$$\begin{array}{ccc} (\mathbb{Z}/n)^\times \times \mathbb{Z}/(n^{s-t+1}/p) & \longrightarrow & (\mathbb{Z}/n^{s+1})^\times \\ (r, m) & \longmapsto & r^{n^s}(1 + mn^t) \bmod n^{s+1}, \end{array}$$

where r is a random value, and m is a message. The encryption function $\mathcal{E}_t$ has properties as follows:

- $\mathcal{E}_t$ is an injective function.
- $\mathcal{E}_t$ has homomorphism in m if and only if $t \geq \frac{s+1}{2}$.

We describe this scheme as follows:

Key Generation: Given a security parameter k, choose at random a modulus $n = p^2q$ of length k bits, where p and q are the same length, and $p \nmid q - 1$, $q \nmid p - 1$. Compute $d \equiv n^{-s} \pmod{(p-1)(q-1)}$ and $l \in \mathbb{Z}$ such that $2^l < pq < 2^{l+1}$. Then the public key is $\mathrm{pk} = (n, l, s)$ and the secret key is $\mathrm{sk} = (p, q, d)$.

Encryption: To encrypt a message $m \in \mathbb{Z}/(n^s/p)$, choose $r \in (\mathbb{Z}/n)^\times$ at random, and compute $\mathcal{E}_t(r, m)$.

Decryption: To decrypt a ciphertext c, compute $r = c^d \bmod pq$. Then

$$\mathcal{D}(c) = L_{n^t}(c(r^{n^s})^{-1} \bmod n^{s+1}) \bmod n^s,$$

where $L_{n^t}(x) = \frac{x-1}{n^t}$.

This scheme has the indistinguishability against the chosen plaintext attack (IND-CPA) under the decisional composite residuosity assumption.

2.2 Quadratic Residues

To decide whether an element x is a quadratic residue in $(\mathbb{Z}/p^a)^\times$ or not, we apply the following Lemma:

Lemma 1. *Let p be a prime. Then for $s \in \mathbb{Z}$*

$$x \in QR_p \Leftrightarrow x \in QR_{p^s},$$

where $QR_p = \{x \in (\mathbb{Z}/p)^\times | x = y^2 \bmod p, y \in (\mathbb{Z}/p)^\times\}$, that is, a set of a quadratic residues over $\mathbb{Z}/p$.

Next, for $a, b \in \mathbb{N}$, let $n = p^a q^b$. To decide whether an element x is a quadratic residue in $(\mathbb{Z}/n)^\times$ or not, we apply the Lemma 1 and the following Lemma:

Lemma 2. *Let p, q be distinct primes. Then*

$$x \in QR_{p^a} \text{ and } x \in QR_{q^b} \Leftrightarrow x \in QR_{p^a q^b}.$$

3 Our Model

3.1 Definitions

We consider the following situation. Let $\mathcal{R}$ be a receiver, $\mathcal{S}_1, \ldots, \mathcal{S}_u$ be senders, and $\mathcal{V}_1, \ldots, \mathcal{V}_y$ be servers. By using servers $\mathcal{V}_1, \ldots, \mathcal{V}_y$ as mediators between servers and the receiver, senders $\mathcal{S}_1, \ldots, \mathcal{S}_u$ want to send information on $R(m_1, m_2, \ldots, m_u)$ without providing each messages m_i, where R is a operation. In addition, by using receiver's public key pk, each sender $\mathcal{S}_i$ wants to divide a message m_i to y shares, and then to distribute a part of y shares to each servers.

In practice, $\mathcal{S}_i$ computes ciphertexts $(c_{i,1}, c_{i,2}, \ldots, c_{i,y})$ from a message m_i, concerned with $\mathcal{R}$'s public key. Then $\mathcal{S}_i$ sends $c_{i,j}$ to $\mathcal{V}_j$. $\mathcal{V}_j$ composes C_j from ciphertexts $c_{1,j}, c_{2,j}, \ldots, c_{u,j}$ received from respective $\mathcal{S}_i$s, and sends them to $\mathcal{R}$. $\mathcal{R}$ computes C from $C_1, C_2, \ldots, C_y$. In this model, we require the following conditions:

(1) With $\mathcal{R}$'s secret key sk, $(C_1, C_2, \ldots, C_y)$ reveal $R(m_1, m_2, \ldots m_u)$, where C_j is $\mathcal{V}_j$'s composition of $(c_{1,j}, c_{2,j}, \ldots, c_{u,j})$. Furthermore, reveal no information of each m_i from $(C_1, C_2, \ldots C_y)$.

(2) A lack of at least one of C_j reveals no information of $R(m_1, m_2, \ldots, m_u)$.

That is, $\mathcal{R}$ obtains information on messages $R(m_1, m_2, \ldots, m_u)$ if and only if $\mathcal{R}$ receives the divided information C_j from all $\mathcal{V}_1, \ldots, \mathcal{V}_y$.

Formally, this model consists of the following algorithms:

Key Generation(1^k)**:** A *probabilistic key generation algorithm* that takes as input a security parameter k, it outputs $(\mathrm{pk}, \mathrm{sk})$, where pk is the public key and sk is the secret key.

Encryption$(m, \mathbf{pk})$**:** A *probabilistic encryption algorithm* that takes as input a message m and the public key pk, it outputs ciphertexts $(c_{i,1}, c_{i,2}, \ldots, c_{i,y})$.

Composition$(c_{1,j}, c_{2,j}, \ldots, c_{u,j}, \mathbf{pk})$**:** A *probabilistic composition algorithm* that takes as input ciphertexts $(c_{1,j}, c_{2,j}, \ldots, c_{u,j})$ and pk, it outputs composition C_i.

Decryption$(C_1, C_2, \ldots, C_y, \mathbf{sk})$**:** A *deterministic decryption algorithm* that takes as input compositions $(C_1, C_2, \ldots, C_y)$ and the secret key sk, it outputs information of messages m_1, $m_2, \ldots, m_u$.

All of these algorithms should run in polynomial time in the length of their inputs. Before going further, we introduce some further conventions.

- We need to run two protocols. One is between $\mathcal{S}_i$ and $\mathcal{V}_j$, and the other is between $\mathcal{V}_j$ and $\mathcal{R}$.
- We assume that $\mathcal{S}_i$ sends $c_{i,j}$ to $\mathcal{V}_j$.

3.2 Our idea

To develop this model, we use the following ideas: We recall that since $\mathcal{E}_t(r, m) = r^{n^s}(1+mn^t) \bmod n^{s+1}$, $\mathcal{E}_t$ has homomorphism if and only if $t \geq \frac{s+1}{2}$. Hence, $\mathcal{E}_{\frac{s+1}{2}}$ has homomorphism, but $\mathcal{E}_{\frac{s+1}{4}}$ not. For simplicity, we fix $s = 3$. In this case, $\mathcal{E}_2$ has homomorphism, but $\mathcal{E}_1$ not. In other words, if we encrypt messages m_i $(1 \leq i \leq u)$ using $\mathcal{E}_2$, then the product of the ciphertexts is $\prod_i \mathcal{E}_2(r_i, m_i) = \mathcal{E}_2(\prod_i r_i, \sum_i m_i)$. However, if we encrypt using $\mathcal{E}_1$, then the product is not $\mathcal{E}_1(r, \sum_i m_i)$. By the way, we know the factorization of the following equation $1 - x^2$, that is, $(1-x)(1+x)$. We can easily see

$$\begin{aligned}\mathcal{E}_1(r_1, m)\mathcal{E}_1(r_2, -m) &= (r_1 r_2)^{n^3}(1+mn)(1-mn) \bmod n^4 \\ &= (r_1 r_2)^{n^3}(1-m^2n^2) \bmod n^4 \\ &= \mathcal{E}_2(r_1 r_2, -m^2).\end{aligned}$$

Therefore,

$$\begin{aligned}&\mathcal{E}_1(r_{1,1}, m_1)\mathcal{E}_1(r_{1,2}, m_1)\mathcal{E}_1(r_{2,1}, m_2)\mathcal{E}_1(r_{2,2}, m_2) \\ =\ &\mathcal{E}_2\left(r_{1,1}r_{1,2}r_{2,1}r_{2,2}, -(m_1^2 + m_2^2)\right) \\ =\ &\mathcal{E}_2\left(\prod r_{i,j}, -(m_1^2 + m_2^2)\right).\end{aligned}$$

If we choose messages $m_1, \ldots, m_u$ and use this technique, then we obtain the ciphertext on sum of message square, that is, $\sum_i m_i^2$.

In order for transformation $\sum_i m_i^2$ to $\sum_i m_i$, we use the following idea. If $\forall m_i = 1$, then $\sum_i m_i^2 = \sum_i m_i$. We regard m as $\sum_{i=1}^{m} 1$, and use $(1+n)^m$ instead of $(1+mn)$. More precisely, we identify a ciphertext of m as the product of m ciphertexts of 1, that is,

$$(1+n)^m = \underbrace{(1+n)\cdots(1+n)}_{m \text{ times}}. \tag{1}$$

In the same way, we identify a ciphertext of $-m$ as the products of m ciphertexts of -1.

Now, we denote $\mathcal{E}'_t(r, m, w)$ by $r^{n^s}(1+wn^t)^m \bmod n^{s+1}$. As described later, w depends on the number of servers. $\mathcal{E}'_t$ inherits some properties of $\mathcal{E}_t$.

3.3 Two Servers Model

We describe two servers model as follows:

Key Generation: Given a security parameter k, choose at random a modulus $n = p^2q$ of length k bits, where p and q are the same length, and $p \nmid q-1$, $q \nmid p-1$. Compute $d \equiv n^{-3} \pmod{(p-1)(q-1)}$ and $l \in \mathbb{Z}$ such that $2^l < pq < 2^{l+1}$. Then the public key is pk$= (n, l, 3)$ and the secret key is sk$= (p, q, d)$.

Encryption: To encrypt a message $m \in \mathbb{Z}/(n^3/p)$, choose random numbers $r_1, r_2 \in (\mathbb{Z}/n)^\times$ and a bit $b \in \{0, 1\}$ at random, and compute $(c, c') = (\mathcal{E}'_1(r_1, m, 1), \mathcal{E}'_1(r_2, m, -1))$. Then set (c_1, c_2) to (c, c') if $b = 0$, or to (c', c) otherwise, and send c_1 to $\mathcal{V}_1$ and c_2 to $\mathcal{V}_2$.

Composition: Let $(c_{i,1}, c_{i,2})$ be $\mathcal{S}_i$'s ciphertexts. $\mathcal{V}_j$ receives $(c_{1,j}, c_{2,j}, \ldots, c_{u,j})$ and compute $C_j = \Pi_i c_{i,j}$.

Decryption: To decrypt a composed ciphertext $C = C_1C_2$, compute $r = C^d \bmod pq$. Then

$$\mathcal{D}(C) = L_{n^2}(C(r^{n^3})^{-1} \bmod n^4) \bmod n^3,$$

Since $\mathcal{E}'_1$ does not have homomorphism, $\mathcal{D}(C_1) \neq \sum_i m_i$ and also C_2. Fortunately, from the following theorem, we see that $\mathcal{D}(C_1C_2) = -\sum_{i=1}^{u} m_i$.

Theorem 3. *For random numbers $r_{i,j} \in (\mathbb{Z}/n)^\times$ $(1 \le i \le u, j = 1, 2)$, messages $m_i \in \mathbb{Z}/(n^3/p)$,*

$$\mathcal{D}\big(\prod_i \mathcal{E}'_1(r_{i,1}, m_i, 1)\mathcal{E}'_1(r_{i,2}, m_i, -1)\big) = -\sum_i m_i.$$

Proof. We have

$$\begin{aligned}
& \mathcal{E}'_1(r_{i,1}, m_i, 1)\mathcal{E}'_1(r_{i,2}, m_i, -1) \\
= \; & r_{i,1}^{n^3}(1+n)^{m_i} r_{i,2}^{n^3}(1-n)^{m_i} \bmod n^4 \\
= \; & (r_{i,1}r_{i,2})^{n^3}\big((1+n)(1-n)\big)^{m_i} \bmod n^4 \\
= \; & (r_{i,1}r_{i,2})^{n^3}(1-n^2)^{m_i} \bmod n^4 \\
= \; & \mathcal{E}'_2(r_{i,1}r_{i,2}, m_i, -1).
\end{aligned}$$

Hence, we have

$$\begin{aligned}
& \prod_i \mathcal{E}'_1(r_{i,1}, m_i, 1)\mathcal{E}'_1(r_{i,2}, m_i, -1) \\
= \; & \prod_i \mathcal{E}'_2(r_{i,1}r_{i,2}, m_i, -1) \\
= \; & \mathcal{E}'_2(\prod_i r_{i,1}r_{i,2}, \sum_i m_i, -1) \\
= \; & (\prod_i r_{i,1}r_{i,2})^{n^3}(1-n^2)^{\sum_i m_i}.
\end{aligned}$$

We denote R by $\prod_i r_{i,1}r_{i,2}$ and M by $\sum_i m_i$. We can recover R as $(R^{n^3}(1-n^2)^M)^d \bmod n$ because

$(1-n^2)^{Md} \equiv 1 \pmod{n}$. Therefore,

$$\begin{aligned}
& \mathcal{D}\big(\prod_i \mathcal{E}'_1(r_{i,1}, m_i, 1)\mathcal{E}'_1(r_{i,2}, m_i, -1)\big) \\
= \ & \mathcal{D}\big(R^{n^3}(1-n^2)^M \bmod n^4\big) \\
= \ & L_{n^2}\big(R^{n^3}(1-n^2)^M(R^{-1})^{n^3} \bmod n^4\big) \bmod n^3 \\
= \ & L_{n^2}\big((1-n^2)^M \bmod n^4\big) \bmod n^3 \\
= \ & L_{n^2}(1 - Mn^2 \bmod n^4) \bmod n^3 \\
= \ & -M \bmod n^3.
\end{aligned}$$

□

3.4 Security

$\mathcal{E}'_t$ inherits the security of $\mathcal{E}_t$, since $\mathcal{E}'_t(r, w, m) = \prod_{i=1}^m \mathcal{E}_t(r_i, w)$ and $\mathcal{E}'_t(r, w, m) = \mathcal{E}_t(r, wm)$ when $t \geq \frac{s+1}{2}$. Hence, we obtain $\mathcal{E}'_t$ is IND-CPA under the same assumption.

$\mathcal{R}$ with the secret key sk can decrypt each C_j. Now, we consider whether $\mathcal{R}$ obtains information on a message m_i or $\sum_i m_i$ from $D(C_j)$. In encryption algorithm, we shuffle ciphertexts before sending a part of them to $\mathcal{V}_1, \mathcal{V}_2$. This action is necessary, otherwise $\mathcal{R}$ can recover $\sum_i m_i$ from $r^{n^3}(1+n)^{\sum_i m_i} \bmod n^4$ or $r^{n^3}(1-n)^{\sum_i m_i} \bmod n^4$ by techniques for decryption in [2]. In particular, the probability that $C_j = r^{n^3}(1+n)^{\sum_i m_i} \bmod n^4$ or $r^{n^3}(1-n)^{\sum_i m_i} \bmod n^4$ is $\frac{1}{2^u}$, where u is the number of senders, if we shuffle ciphertexts.

4 Variants with More Servers

In order to construct two servers model, we used $\omega_2 = \pm 1$, which are square roots of 1. This two servers model depends on square roots. In particular, there exists -1 in $(\mathbb{Z}/n)^\times$ anytime, hence we can easily construct two servers model. Note that it is hard for anyone without knowing the factor p or q of n to find $v \in (\mathbb{Z}/n)^\times$ such that $v^2 \equiv 1 \pmod{n}$ and $v \not\equiv \pm 1 \pmod{n}$, that is, to find v is equivalent to factoring $n = p^2 q$. In fact, by using the Chinese Remainder Theorem (CRT):

$$\begin{array}{rccc}
\psi & : & \mathbb{Z}/n & \longrightarrow & \mathbb{Z}/p^2 \times \mathbb{Z}/q \\
 & & x & \longmapsto & (x \bmod p^2, x \bmod q)
\end{array}$$

We obtain $\psi(1) = (1, 1)$ and $\psi(v) = (1, -1)$ or $(-1, 1)$, hence $\psi(1 \pm v) = (2, 0)$ or $(0, 2)$. We obtain the factor of n from $1 < \gcd(n, 1 \pm v) < n$. In addition, the reason for $s = 3$ is optimal from the point of view of computational costs.

Next, we consider how to increase the number of servers by applying the technique above. If there exists a non-trivial y-th root of 1 in $(\mathbb{Z}/n)^\times$, we can construct y servers as follows: Let $y = 3$ and $t = 1$. Then, our scheme uses the encryption function $\mathcal{E}'_1(r, m, w)$. Now, we compute a cube root ω_3 of 1. It is non-trivial to compute ω_3 in $(\mathbb{Z}/n)^\times$, although we easily see that one of square roots is -1. In addition, in three servers model, we also require that the following equations hold:

$$\begin{aligned}
& \mathcal{E}'_3(r_1, m_1, -1)\mathcal{E}'_3(r_2, m_2, -1) \\
= \ & \mathcal{E}'_3(r_1 r_2, m_1 + m_2, -1) \bmod n^{s+1}
\end{aligned} \tag{2}$$

$$\begin{aligned}
& \mathcal{E}'_1(r_1, m, -\omega_3^1)\mathcal{E}'_1(r_2, m, -\omega_3^2)\mathcal{E}'_1(r_3, m, -1) \\
= \ & \mathcal{E}'_3(r_1 r_2 r_3, m, -1) \bmod n^{s+1}.
\end{aligned} \tag{3}$$

First, we refer to equation (2) in order to fix the parameter s. Next, we discuss the existence of non-trivial cube roots ω_3.

For the first equation (2), this means that $(1-n^3)^x \equiv 1 - xn^3 \pmod{n^{s+1}}$. We know that,

$$\begin{aligned}(1-n^3)^x &= \sum_{i=0}^{x} \binom{x}{i}(-n^3)^x \\ &= 1 - xn^3 + \frac{x(x-1)}{2}n^6 - \cdots + (-n^3)^x,\end{aligned}$$

over $\mathbb{Z}$. Hence, for $3 \leq s \leq 5$, we obtain $\sum_{i=0}^{x}\binom{x}{i}(-n^3)^x$
$\equiv 1 - xn^3 \pmod{n^{s+1}}$. Now, fix $s = 5$. Then,

$$\begin{aligned}\mathcal{E}'_1(r, m, w) &= r^{n^5}(1+wn)^m \bmod n^6 \\ \mathcal{E}'_3(r, m, -1) &= r^{n^5}(1-n^3)^m \bmod n^6.\end{aligned}$$

For the second equation (3), we must find $x \neq 1$ such that $x^3 - 1 \equiv 0 \pmod{n^5}$. Such x also holds an equation $x^2 + x + 1 \equiv 0 \pmod{n^5}$. We know a fact that the solution of this equation over $\mathbb{C}$ is $x = \frac{(-1\pm\sqrt{-3})}{2}$. Then, we consider to apply this fact to $\mathbb{Z}/n$. Since $n = p^2q$ is odd, there always exists 2^{-1} over $(\mathbb{Z}/n)^\times$. An important point is whether $\sqrt{-3}$ is an element in $(\mathbb{Z}/n)^\times$ or not. If $\sqrt{-3} \in (\mathbb{Z}/n)^\times$, then $\omega_3 = (-1+\sqrt{-3})2^{-1} \in (\mathbb{Z}/n)^\times$. In fact, with a knowledge of p or q, it is easy to check whether an element $x \in (\mathbb{Z}/n)^\times$ is a quadratic residue over $(\mathbb{Z}/p)^\times$ and $(\mathbb{Z}/q)^\times$. Therefore, it suffices to compute Legendre symbols $\left(\frac{-3}{p}\right)$ and $\left(\frac{-3}{q}\right)$.

We now demonstrate to concrete three servers, let $n = 637 = 7^2 \times 13$. By applying Lemma 1 and 2, we can check easily whether -3 is a quadratic residue over $(\mathbb{Z}/7)^\times$ and $(\mathbb{Z}/13)^\times$ using Legendre symbols. We then see that $\left(\frac{-3}{7}\right) = \left(\frac{-3}{13}\right) = 1$. Hence, there exists $\sqrt{-3}$ in $(\mathbb{Z}/637)^\times$. Next, we determine $\sqrt{-3}$ concretely. First, it holds $-3 \equiv 46 \pmod{7^2}, -3 \equiv 10 \pmod{13}$. Running an algorithm that compute a square root over prime fields, we obtain $\sqrt{-3} \equiv \pm 2 \pmod 7, \sqrt{-3} \equiv \pm 6 \pmod{13}$. Furthermore, we obtain $\sqrt{-3} \equiv \pm 12 \pmod{7^2}$ with the Hensel lifting. Hence, $\psi(\sqrt{-3}) = (12, 6), (12, 7), (37, 6), (37, 7)$, where ψ is the function above. We find that $\sqrt{-3} \equiv 110, 306, 331, 527 \pmod{637}$, using ψ^{-1}. Then one of cube roots of 1 is 165 as $\sqrt{-3} \equiv 331 \pmod{637}$. In fact, $165^3 = 4492125 = 637 \times 7052 + 1 \equiv 1 \pmod{637}$. Hence, we can construct three servers model as follows:

$$\begin{aligned}\mathcal{E}'_1(r_1, m, -165) &= r_1^{n^5}(1-165n)^m \bmod n^6 \\ \mathcal{E}'_1(r_2, m, -165^2) &= r_2^{n^5}(1-27225n)^m \bmod n^6 \\ \mathcal{E}'_1(r_3, m, -1) &= r_3^{n^5}(1-n)^m \bmod n^6\end{aligned}$$

and the composite is $(r_1r_2r_3)^{n^5}(1-n^3)^m \bmod n^6$. We note that 268 is also a cube root of 1. Then, we must not reveal 165 and 268 to senders or servers. Anyone who knows them can recover $\sqrt{-3} \equiv 331, 527 \pmod{637}$ and obtain a factor of n from $\gcd(331 + 527, 637) = 13$.

We have seen a way to construct two or three servers model. Actually, our technique is based on factorization of cyclotomic polynomials and quadratic residues, since there are many useful results for quadratic residues. We remark that s increases with y. We describe $y = 3, 4, 5, 6$, and 8 as follows:

3 servers: Let $s = 5$. A primitive cube root ω_3 holds $\omega_3^2 + \omega_3 + 1 \equiv 0 \pmod{n^5}$. We solve this equation over $\mathbb{C}$, and obtain $\omega_3 = \frac{(-1+\sqrt{-3})}{2}$. Hence we must take p, q such that $\left(\frac{-3}{p}\right) = \left(\frac{-3}{q}\right) = 1$. In other words, $p, q \equiv 1 \pmod 3$. Then, public key is $\mathrm{pk} = (n, l, 5, \omega_3)$. $\mathcal{S}_i$ computes $\mathcal{E}'_1(r_{i,j}, m_i, -\omega_3^j)$, for $j = 1, 2, 3$.

y	$* \in (\mathbb{Z}/n)^\times$	condition of p, q
2	-1	(anytime)
3	$\sqrt{-3}$	$p, q \equiv 1 \pmod 3$
4	$\sqrt{-1}$	$p, q \equiv 3 \pmod 4$
5	$\sqrt{5}, \sqrt{2\sqrt{5}-10}$	(at least) $p, q \equiv \pm 1 \pmod 5$
6	$\sqrt{-3}$	$p, q \equiv 1 \pmod 3$
8	$\sqrt{-1}, \sqrt{2}$	$p, q \equiv 1 \pmod 8$

Table 1: Construction of y servers model.

4 servers: Let $s = 7$. A primitive fourth root ω_4 holds $\omega_4^2 + 1 \equiv 0 \pmod{n^7}$ that is $\omega_4 = \sqrt{-1}$. Hence we must take p, q such that $\left(\frac{-1}{p}\right) = \left(\frac{-1}{q}\right) = 1$. In other words, $p, q \equiv 3 \pmod 4$. Then, public key is $\mathrm{pk} = (n, l, 7, \omega_4)$. $\mathcal{S}_i$ computes $\mathcal{E}_1'(r_{i,j}, m_i, -\omega_4^j)$, for $j = 1, \ldots, 4$.

5 servers: Let $s = 9$. The ω_5 holds $\omega_5^4 + \omega_5^3 + \omega_5^2 + \omega_5 + 1 = \left(\omega_5^2 - (1-\sqrt{5})2^{-1}\omega_5 + 1\right)\left(\omega_5^2 + (1-\sqrt{5})2^{-1}\omega_5 + 1\right) \equiv 0 \pmod{n^9}$. Consequently, we require that there exists $\sqrt{5}$ in $(\mathbb{Z}/n)^\times$. ω_5 holds $\left(1 + \sqrt{5} + \sqrt{2\sqrt{5}-10}\right)4^{-1}$, if $\sqrt{5} \in (\mathbb{Z}/n)^\times$. In addition, to construct five servers model, we require that it holds $\sqrt{2\sqrt{5}-10} \in (\mathbb{Z}/n)^\times$.

6 servers: Let $s = 11$. The ω_6 holds $\omega_6^2 - \omega_6 + 1 = 0$. We obtain $\omega_6 = (1 + \sqrt{-3})2^{-1} = \omega_3 + 1$. Hence we set p, q as a case $r = 3$. Then, public key is $\mathrm{pk} = (n, l, 11, \omega_6)$. $\mathcal{S}_i$ computes $\mathcal{E}_1'(r_{i,j}, m_i, -\omega_6^j)$, for $j = 1, \ldots, 6$ and send to each servers.

8 servers: Let $s = 15$. The ω_8 holds $\omega_8^4 + 1 = 0$. We obtain $\omega_8 = \sqrt{2}(1 + \sqrt{-1})2^{-1}$ over $\mathbb{C}$. We want p, q to hold $\sqrt{-1}, \sqrt{2} \in (\mathbb{Z}/p)^\times, (\mathbb{Z}/q)^\times$. So we set $p, q \equiv 1 \pmod 8$. Then, public key is $\mathrm{pk} = (n, l, 15, \omega_8)$. Senders compute ciphertexts and send to each servers.

Remark 4. *We have shown a method to decompose ciphertexts by applying the factoring of cyclotomic polynomial over* $\mathbb{C}$ *when there exist primitive roots of* 1 *in* $(\mathbb{Z}/n)^\times$. *In fact, we can construct without computing a* y*-th primitive root. For example, in four servers model, we use encryption functions* $r_1^{n^s}(1-n)^m, r_2^{n^s}(1+n)^m, r_3^{n^s}(1-n+n^2)^m, r_4^{n^s}(1+n+n^2)^m$ *as* $1-x, 1+x, 1-x+x^2, 1+x+x^2$ *respectively* $1 - x^6 = (1-x)(1+x)(1-x+x^2)(1+x+x^2)$. *However, in this example, we need* $s \geq 11$.

Remark 5. *It is well-known that there are no formulas to solve equations of degree at least 5. Hence, it is not easy to factor* $1 - x^y$ *when* y *is large. In other words, for a function* $f_y(x) = x^y \bmod n^s$ *it is not easy to find* $x \in (\mathbb{Z}/n^s)^\times$ *such that* $f_y(x) = 1$. *On the other hand, we observe* f_y *is closely related to the RSA function. If* $gcd(y, \varphi(n^s)) = 1$, f_y *is a permutation polynomial over* $(\mathbb{Z}/n^s)^\times$, *where* φ *is Euler's totient function. Therefore, there are no* y*-th primitive roots. So, we require that* $gcd(y, \varphi(n^s)) > 1$.

5 Conclusions

We have introduced a new model "decomposition of ciphertext" such as a primitive combined the secret sharing with additively homomorphism. And we have developed y servers model by applying a variant of Paillier encryption scheme which has some homomorphic properties and decomposition of cyclotomic polynomials. We have remarked the number of servers. It will be a further work to construct to this model without decomposition of cyclotomic polynomials.

References

[1] Chaum, D. L. Untraceable electronic mail, return addresses, and digital pseudonyms. *Commun. ACM 24*, 2 (1981), 84–90.

[2] Damgård, I., and Jurik, M. A Generalisation, a Simplification and Some Applications of Paillier's Probabilistic Public-Key System. *PKC 2001, Lecture Notes in Computer Science 1992* (2001), 119–136.

[3] Hirano, T., Wada, K., and Tanaka, K. A Variant of the Schmidt-Takagi Encryption Scheme. *SCIS 2008* (2008). to appear.

[4] Paillier, P. Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. *Advances in Cryptology-EUROCRYPT'99, Lecture Notes in Computer Science 1592* (1999), 223–238.

[5] Rogaway, P., and Bellare, M. Robust computational secret sharing and a unified account of classical secret-sharing goals. In *CCS '07: Proceedings of the 14th ACM conference on Computer and communications security* (New York, NY, USA, 2007), ACM, pp. 172–184.

[6] Shamir, A. How to Share a Secret. *Commun. ACM 22*, 11 (1979), 612–613.