A New Class of Product-sum Type Public Key Cryptosystem, K(V)ΣΠPKC, Constructed Based on Maximum Length Code

Masao KASAHARA *

Abstract

The author recently proposed a new class of knapsack type PKC referred to as K(Ⅱ)ΣΠPKC [1]. In K(Ⅱ)ΣΠPKC with old algorithm DA[I], Bob randomly constructs a very small subset of Alice's set of public key whose order is very large, under the condition that the coding rate ρ satisfies $0.01 < \rho < 0.2$. In K(Ⅱ)ΣΠPKC, no secret sequence such as super-increasing sequence or shifted-odd sequence but the sequence whose components are constructed by a product of the same number of many prime numbers of the same size, is used. In this paper we present a new algorithm, DA(Ⅱ) for decoding K(Ⅱ)ΣΠPKC. We show that with new decoding algorithm, DA(Ⅱ), K(Ⅱ)ΣΠPKC yields a higher coding rate and a smaller size of public key compared with K(Ⅱ)ΣΠPKC using old decoding algorithm, DA(I). We further present a generalized version of K(Ⅱ)ΣΠPKC, referred to as K(V)ΣΠPKC. We finally present a new decoding algorithm DA(Ⅲ) and show that, in K(V)ΣΠPKC with DA(Ⅲ), the relation, $r_F \cong 0, \rho \cong \frac{2}{3}$ holds, where r_F is the factor ratio that will be defined in this paper. We show that K(V)ΣΠPKC yields a higher security compared with K(Ⅱ)ΣΠPKC.

keyword

Public-key cryptosystem(PKC), Product-sum type PKC, Knapsack-type PKC, LLL algorithm, PQC.

1 Introduction

Various studies have been made of the Public-Key Cryptosystem (PKC). The security of the PKC's proposed so far, in most cases, depends on the difficulty of discrete logarithm problem or factoring problem. For this reason, it is desired to investigate another class of PKC, so called PQC, that does not rely on the difficulty of these two problems.

Two of the promising candidates among the members of class of PKC are the code-based PKC and the product-sum type PKC [1]∼ [21].

The author recently proposed a new class of product-sum type PKC referred to as K(Ⅱ)ΣΠPKC. In K(Ⅱ)ΣΠPKC with old decoding algorithm DA[I], Bob randomly constructs a very small subset of Alice's set of public key whose order is very large, under the condition that the coding rate ρ satisfies $0.01 < \rho < 0.2$. In K(Ⅱ)ΣΠPKC, no secret sequence such as super-increasing sequence [6] or shifted-odd sequence [13]∼ [15] but the sequence whose components are constructed by the products of the same number of many prime numbers of the same size, is used. Namely each of the components of the secret sequence such as super-increasing sequence or shifted-sequence has a different entropy. On the other hand the components of the secret sequence used in K(Ⅱ)ΣΠPKC take on the same entropy.

*Research Institute for Science and Engineerging, Waseda University. http://www.informatics-culture.net, kasahara@ogu.ac.jp

In this paper we present a new algorithm, DA(Ⅱ) for decoding K(Ⅱ)ΣΠPKC. We show that with new decoding algorithm, DA(Ⅱ), K(Ⅱ)ΣΠPKC yields a higher coding rate and a smaller size of public key compared with K(Ⅱ)ΣΠPKC using old decoding algorithm, DA(I). We further present a generalized version of K(Ⅱ)ΣΠPKC, referred to as K(V)ΣΠPKC. We finally present a new decoding algorithm DA(Ⅲ) and show that, in K(V)ΣΠPKC with DA(Ⅲ), the relation, $r_F \cong 0, \rho \cong \frac{2}{3}$ holds, where r_F is the factor ratio that will be defined in this paper. We show that K(V)ΣΠPKC yields a higher security compared with K(Ⅱ)ΣΠPKC.

In the following sections, in order to let this paper be self-contained we shall briefly describe K(Ⅱ)ΣΠPKC.

2 K(Ⅱ)ΣΠPKC for two messages

2.1 Preliminaries

Let us define several symbols :

$$
\begin{aligned}
G_F(x) &: \text{primitive polynomial over } \mathbb{F}_2 \text{ of degree } g \\
m_i &: \text{message symbol over } \mathbb{Z} \;\; ; \;\; i = 1, 2, \cdots, \lambda. \\
w, W &: \text{secret key for generating a set of public key.} \\
p_i &: \text{prime number ; } i = 1, 2, \cdots, n. \\
\boldsymbol{p} &: \text{prime number vector ; } (p_1, p_2, \cdots, p_n). \\
q_i &: \text{product of prime numbers, } p_{i1}, p_{i2}, \cdots, p_{i\sigma} \;\; ; \;\; \sigma < n, p_{ij} \in \{p_i\}. \\
k_i &: \text{public key, } wq_i \equiv k_i \mod W \;\; ; \;\; i = 1, 2, \cdots, n. \\
C &: \text{ciphertext, } C = m_1k_1 + m_2k_2 + \cdots + m_nk_n. \\
\Gamma &: \text{Intermediate message } w^{-1}C \equiv \Gamma \mod W. \\
|p_i| &: \text{size of } p_i, p \text{ (in bit).} \\
|m_i| &: \text{size of } m_i, m \text{ (in bit).}
\end{aligned}
$$

The conventional knapsack type PKC are constructed using the following sequences:

(i) : super-increasing sequence [6]

(ii) : shifted-odd sequence [12] ∼ [14]

In these sequences, entropies of the components are not necessarily same. On the other hands, the entropies of the components of the secret sequence used in K(Ⅱ)ΣΠPKC are exactly same. We shall refer to such secret sequencce as uniform sequence [1], [19]∼ [21].

In the following sections, when the variable x_i takes on an actual value $\tilde{x}_i$, we shall denote the corresponding vector, $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)$, as

$$\widetilde{\boldsymbol{x}} = (\widetilde{x}_1, \widetilde{x}_2, \cdots, \widetilde{x}_n). \tag{1}$$

The $\widetilde{C}$ and $\widetilde{M}$ et al. will be defined in a similar manner.

2.2 Summary of idea of K(Ⅱ)ΣΠPKC

In this sub-section let us summarize the idea of a secret system using an example of K(Ⅱ)ΣΠPKC for two messages.

Let the Alice's set of public key, be denoted $\{k_i\}_A$.

For example, for the message $\boldsymbol{m} = (m_A, m_B)$, Bob randomly chooses two keys, k_A and k_B, from the set of Alice's public key $\{k_i\}_A$.

Bob encrypts the message $\boldsymbol{m}$ into

$$\boldsymbol{m} \mapsto \boldsymbol{C} = m_A k_A + m_B k_B. \tag{2}$$

Alice decrypts the ciphertext $\boldsymbol{C}$ into

$$\boldsymbol{C} \mapsto \boldsymbol{m} = (m_A, m_B). \tag{3}$$

2.3 Maximum length code

In this sub-section, we assume that n is

$$n = 2^g - 1. \tag{4}$$

The maximum length code $\{F_M(x)\}$ is a cyclic code that satisfies

$$F_M(x) \equiv 0 \mod \frac{x^n - 1}{G_F(x)}, \tag{5}$$

where $G_F(x)$ over $\mathbb{F}_2$ is a primitive polynomial of degree g [22].

In the followings $\{F_M(x)\}$ will also be denoted simply by $\{F_M\}$.

Let the two code words (m-sequences) of $\{F_M\}$, M_α and M_β over $\mathbb{F}_2$, be denoted

$$\begin{aligned} M_\alpha &= (\alpha_1, \alpha_2, \cdots, \alpha_n), \\ M_\beta &= (\beta_1, \beta_2, \cdots, \beta_n). \end{aligned} \tag{6}$$

Let the sets S_1, S_2, S_3 be defined as follows :

$$\begin{aligned}
&S_1 : \text{Set of pairs } (\alpha_i, \beta_i)\text{'s such that} && \alpha_i = 1, \ \ \beta_i = 1 \ \ ; \ \ i = 1, 2, \cdots, n. \\
&S_2 : \text{Set of pairs } (\alpha_i, \beta_i)\text{'s such that} && \alpha_i = 0, \ \ \beta_i = 0 \ \ ; \ \ i = 1, 2, \cdots, n. \\
&S_3 : \text{Set of pairs } (\alpha_i, \beta_i)\text{'s such that} && \alpha_i = 0, \ \ \beta_i = 1 \ \text{ or } \ \alpha_i = 1, \beta_i = 0 \ \ ; \ \ i = 1, 2, \cdots, n.
\end{aligned}$$

Theorem 1 : The orders $\#S_1$, $\#S_2$ and $\#S_3$ are given by

$$\begin{aligned} \#S_1 &= \frac{n+1}{4}, \\ \#S_2 &= \frac{n-3}{4}, \\ \#S_3 &= \frac{n+1}{2}. \end{aligned} \tag{7}$$

Proof : See Ref.[1].

2.4 Construction of composite numbers $\{q_i\}$

Let $\boldsymbol{A}$ be a code word of $\{F_M\}$ and $\boldsymbol{p}$, a prime number vector whose components are randomly chosen prime numbers. Let $\boldsymbol{A}$ and $\boldsymbol{p}$ be denoted

$$\begin{aligned} \boldsymbol{A} =& (a_1, a_2, \cdots, a_n). \\ \boldsymbol{p} =& (p_1, p_2, \cdots, p_n) \ \ ; \ \ p_i \neq p_j \ \text{ for } \ i \neq j \ \ ; \ \ |p_i| = p. \end{aligned} \tag{8}$$

$\mathbf{p}$:	p_1	p_2	p_3	p_4	p_5	p_6	p_7
M_1 :	0	0	1	0	1	1	1
M_2 :	1	0	0	1	0	1	1
M_3 :	1	1	0	0	1	0	1
M_4 :	1	1	1	0	0	1	0
M_5 :	0	1	1	1	0	0	1
M_6 :	1	0	1	1	1	0	0
M_7 :	0	1	0	1	1	1	0

Figure 1: Array $((x+1)(x^3+x+1))$

Let $\boldsymbol{w}$ be defined

$$\boldsymbol{w}_A = (a_1 p_1, a_2 p_2, \cdots, a_n p_n). \tag{9}$$

Let the composite number $q^{(A)}$ be defined by the products of the non-zero components of $\boldsymbol{w}_A$. Namely $q^{(A)}$ can be represented by

$$q^{(A)} = \prod_{i=1}^{n} a'_i p_i, \tag{10}$$

where $a'_i \neq 0$ is an i-th component of $\boldsymbol{A}$.

Let another code word $\boldsymbol{B}$ be denoted

$$\boldsymbol{B} = (b_1, b_2, \cdots, b_n). \tag{11}$$

The following composite number $q^{(B)}$ can be obtained from $\boldsymbol{w}_B = (b_1 p_1, b_2 p_2, \cdots, b_n p_n)$ in a similar manner as $q^{(A)}$:

$$q^{(B)} = \prod_{i=1}^{n} b'_i p_i. \tag{12}$$

We have the following straightforward theorem.

Theorem 2 : Letting the largest common divisor $(q^{(A)}, q^{(B)})$ be denoted $d_{A,B}$, it is

$$d_{A,B} = \prod_{i=1}^{\#S_1} p_i^{(A,B)}, \tag{13}$$

where $p_i^{(A,B)}$ denotes a prime number for which $(a_i, b_i) \in S_1$.

Let w and W be relatively prime positive integers such that

$$w < W, \ \ (w, W) = 1. \tag{14}$$

The set of public keys, $\{k_i\}$, is given by

$$w q_i = k_i \bmod W \ \ ; \ \ i = 1, \cdots, n. \tag{15}$$

Public key	:	$\{k_i\}$
Secret key	:	w, W, $\{p_i\}$, $\{q_i\}$, $\{M_i\}$

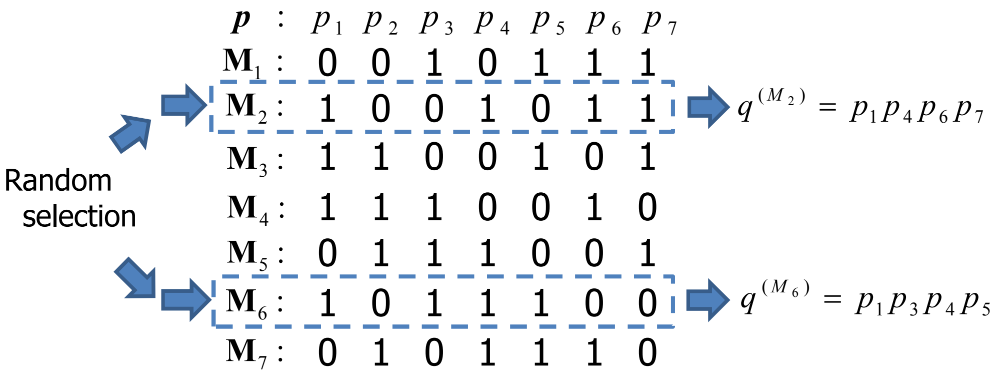

Figure 2: Random selection of q^{M2} and q^{M6}

Example 1 : Maximum length code of length $n = 2^3 - 1$.
Let $G_F(x)$ be

$$G_F(x) = x^3 + x + 1. \tag{16}$$

All the code words (m-sequences) generated by $(x^7+1)/G_F(x) = (x+1)(x^3+x^2+1)$ are listed in Fig.1. List of the code words will be referred to as Array $((x+1)(x^3+x+1))$.

Let us assume that the two keys k_2 and k_6 are randomly chosen from the set $\{k_i\}$ by Bob (correspondingly two code words M_2 and M_6 in Fig.1 are chosen from $\{M_i\}$), as shown in Fig.2.

Let the prime number vector be

$$\boldsymbol{p} = (p_1, p_2, \cdots, p_7). \tag{17}$$

From Fig.1, w_2 and w_6 are

$$\begin{aligned} \boldsymbol{w}_2 &= (p_1, 0, 0, p_4, 0, p_6, p_7), \\ \boldsymbol{w}_6 &= (p_1, 0, p_3, p_4, p_5, 0, 0). \end{aligned} \tag{18}$$

As show in Fig.2, $q^{(M_2)}$ and $q^{(M_6)}$ are

$$\begin{aligned} q^{(M_2)} &= p_1 p_4 p_6 p_7, \\ q^{(M_6)} &= p_1 p_3 p_4 p_5. \end{aligned} \tag{19}$$

Alice calculates

$$\begin{aligned} (q^{(M_2)}, q^{(M_6)}) &= (p_1 p_4 p_6 p_7, p_1 p_3 p_4 p_5) \\ &= p_1 p_4 = d_{1,4}, \end{aligned} \tag{20}$$

and knows for certain that Bob has randomly selected M_2 and M_6 (correspondingly k_2 and k_6 from the set $\{k_i\}_A$, where $\{k_i\}_A$ implies the Alice's set of public key).

From this example, it is easy to see that any selection of $(M_i, M_j); i \neq j$ by Bob can be successfully known to Alice, who knows the secret key.

2.5 Construction of composite numbers $\{q_i\}$ for general λ

Bob randomly chooses λ code words of $\{F_M\}$. Without loss of generality let us assume that the list of the randomly chosen code words by Bob are the followings:

$$\begin{aligned} M_1 = &\quad (t_{11}, t_{12}, \cdots, t_{1n}), \\ M_2 = &\quad (t_{21}, t_{22}, \cdots, t_{2n}), \\ \vdots & \\ M_\lambda = &\quad (t_{\lambda 1}, t_{\lambda 2}, \cdots, t_{\lambda n}). \end{aligned} \tag{21}$$

Let the column vector $\boldsymbol{t}_i$ be denoted by

$$\boldsymbol{t}_i = \begin{bmatrix} t_{1i} \\ t_{2i} \\ \vdots \\ t_{\lambda i} \end{bmatrix}. \tag{22}$$

Let the total number of $\boldsymbol{t}_i$'s such that $\boldsymbol{t}_i$'s take on the same value $\boldsymbol{a}^{(i)}$ over $\mathbb{F}_2$ be denoted by $N(\boldsymbol{a}^{(i)})$.

Theorem 3 : The $N(\boldsymbol{a}^{(i)})$ is given by

$$\begin{aligned} N(\boldsymbol{a}^{(i)}) &= 2^{g-\lambda} \ \text{ for } \ \boldsymbol{t}_i \neq \mathbf{0}, \\ &= 2^{g-\lambda} - 1 \ \text{ for } \ \boldsymbol{t}_i = \mathbf{0}. \end{aligned} \tag{23}$$

Proof : See, for example, Ref.[1]. □

From Theorem 3 we see that when Bob, in accordance with a random choice of λ public keys, $k_{(1)}, k_{(2)}, \cdots, k_{(\lambda)}$ selects λ code words $M_1, M_2, \ldots, M_\lambda$ among the code words of $\{F_M\}$ for given messages $m_1, m_2, \cdots, m_\lambda$, the largest common divisor of $q^{(M_1)}, q^{(M_2)}, \cdots, q^{(M_\lambda)}$ consists of a product of $2^{g-\lambda}$ prime numbers.

The intermediate message Γ is given as

$$\begin{aligned} \omega^{-1} C \equiv \Gamma \mod W & \\ = m_1 \boldsymbol{q}^{(M_1)} + m_2 \boldsymbol{q}^{(M_2)} + \cdots + m_\lambda \boldsymbol{q}^{(M_\lambda)}. \end{aligned} \tag{24}$$

Let the largest common divisor between $q^{(M_i)}$ and $q^{(M_j)}$ be denoted by d_{ij}. It is easy to see that the size of d_{ij} takes on the same value, namely

$$|d_{ij}| = |d_2|. \tag{25}$$

□

From d_λ, Alice is able to know for certain that Bob has random chosen a set of keys, $k_{(1)}, k_{(2)}, \cdots, k_{(\lambda)}$ from the Alice's set of public key $\{k_i\}_A$.

Let the factor ratio r_F be defined by

$$r_F = \frac{\text{Size of the largest common divisor of } q^{(i)} \text{and } q^{(j)}}{\text{Size of } q^{(i)}}. \tag{26}$$

From Eq.(25), we see that the factor ratio is

$$r_F = \frac{|d_2|}{|q^{(M_i)}|}. \tag{27}$$

The common divisor $(q^{(M_i)}, q^{(M_j)})$ can be disclosed, when $r_F > \frac{1}{2}$, by an algorithm similar to the Euclidean division algorithm, yielding all the secret prime numbers $\{p_i\}$. We shall refer to this attack as Factor Attack.

2.6 Brief sketch of a communication scheme using K(Ⅱ)ΣΠPKC

Encryption process can be performed as follows :

Step 1 : For a given message sequence $\widetilde{m}_1$, $\widetilde{m}_2$, $\cdots$, $\widetilde{m}_\lambda$, Bob randomly chooses λ keys k_{B1}, k_{B2}, $\cdots$, $k_{B\lambda}$ by just taking a look at Alice's public key set $\{k_i\}_A$.

Step 2 : Bob encrypts messages $\widetilde{m}_1$, $\widetilde{m}_2$, $\cdots$, $\widetilde{m}_\lambda$ into

$$\widetilde{C}_B = \widetilde{m}_1 k_{B1} + \widetilde{m}_2 k_{B2} + \cdots + \widetilde{m}_\lambda k_{B\lambda}. \tag{28}$$

Step 3 : Bob sends the ciphertext $\widetilde{C}_B$ to Alice.

Decryption process by Alice is given as shown below :

Step 1 : Alice calculates the intermediate message $\widetilde{\Gamma}_B$ by

$$w^{-1}\widetilde{C}_B \equiv \widetilde{\Gamma}_B = \widetilde{m}_1 q_{B1} + \widetilde{m}_2 q_{B2} + \cdots + \widetilde{m}_\lambda q_{B\lambda} \bmod W. \tag{29}$$

Step 2 : By simply calculating the largest common divisor of $\widetilde{q}_{B1}, \widetilde{q}_{B2}, \cdots, \widetilde{q}_{B\lambda}, \widetilde{d}_\lambda$, Alice decodes $\widetilde{M}_{B1}, \widetilde{M}_{B2}, \cdots, \widetilde{M}_{B\lambda}$ randomly chosen by Bob.

In the next section we shall present a new decoding algorithm for improving the coding rate of K(Ⅱ)ΣΠPKC with old algorithm DA(I) [1] and show how to decode the messages $\widetilde{m}_1$, $\widetilde{m}_2$, $\cdots$, $\widetilde{m}_\lambda$ after knowing $\widetilde{M}_{B1}, \widetilde{M}_{B2}, \cdots, \widetilde{M}_{B\lambda}$.

Theorem 4 : For the given messages $\widetilde{m}_1$, $\widetilde{m}_2$, $\cdots$, $\widetilde{m}_\lambda$, the ciphertext can be uniquely decoded, as far as

$$\log_2 \lambda + \lambda \leqq g \tag{30}$$

is satisfied.

Proof : We see that when all the code words whose generator polynomial is given by $(x^n - 1)/G_F(x)$ are listed, for example, as shown in Fig.1, any column vector is a code word generated by $(x^n - 1)/x^g G_F(x^{-1})$. We then see that the following relation:

$$\lambda 2^\lambda \leqq n + 1 = 2^g \tag{31}$$

should be satisfied, for uniquely decoding $\widetilde{m}_1, \widetilde{m}_2, \cdots, \widetilde{m}_\lambda$, yielding the proof. □

It is easy to see that when λ is $2^a, a = 1, 2, 3, \cdots$, the equality holds in Eq.(31). we shall refer to such λ as optimum λ and denote it by λ_o. We shall also refer to the largest λ such that it satisfies the inequality of Eq(30) as quasi-optimum λ and denote it by λ_{qo}.

The maximum λ's that satisfy Eq.(31), for $g = 2, 3, 4, 5$ and 6 are

$$\begin{aligned} g &= 2, \ \lambda_o = 1, \\ g &= 3, \ \lambda_o = 2, \\ g &= 4, \ \lambda_{qo} = 2, \\ g &= 5, \ \lambda_{qo} = 3, \\ g &= 6, \ \lambda_o = 4. \end{aligned}$$

2.7 An example of decoding process, DA(Ⅱ)

Throughout this section let us discuss on a new decoding algorithm, referred to as DA(Ⅱ), for decoding messages $m_1, m_2, \cdots, m_\lambda$, when Bob randomly has chosen public keys $k_{B1}, k_{B2}, \cdots, k_{B\lambda}$, from Alice's public key, $\{k_i\}_A$, using Example 2 given below.

According to the random choice of public keys $k_{B1}, k_{B2}, \cdots, k_{B\lambda}$, the code words $\boldsymbol{M}_{B1}, \boldsymbol{M}_{B2}, \cdots, \boldsymbol{M}_{B\lambda}$ are chosen.

Before presenting a toy example for illustrating DA(Ⅱ), let us define the symbols:

$$\begin{aligned} \mathbf{0} &= (0000), \\ \mathbf{0}' &= (000), \\ \mathbf{1} &= (1111). \end{aligned} \tag{32}$$

Example 2 : Maximum length code of length $n = 2^6 - 1$, generated by $G_F(x) = x^6 + x + 1$. $\lambda_o = 4$.

Code words $\boldsymbol{M}_i, \boldsymbol{M}_j, \boldsymbol{M}_k, \boldsymbol{M}_l$ can be rearranged as shown in Table 1 by a pertinent column permutation of Array $((x^{63}+1)/(x^6+x+1))$, where q_i implies the composite numbers of four different prime numbers $\in \{p_i\}$.

Table. 1: Rearranged M-sequences

	q_1	q_2	q_3	q_4	q_5	q_6	q_7	q_8	q_9	q_{10}	q_{11}	q_{12}	q_{13}	q_{14}	q_{15}	q_{16}
$\boldsymbol{M}_i$	$\mathbf{0}'$	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
$\boldsymbol{M}_j$	$\mathbf{0}'$	**0**	**0**	**1**	**0**	**1**	**1**	**1**	**0**	**1**	**1**	**0**	**0**	**1**	**0**	**1**
$\boldsymbol{M}_k$	$\mathbf{0}'$	**0**	**1**	**0**	**1**	**0**	**1**	**1**	**1**	**0**	**1**	**0**	**1**	**0**	**0**	**1**
$\boldsymbol{M}_l$	$\mathbf{0}'$	**1**	**0**	**0**	**1**	**1**	**0**	**1**	**1**	**1**	**0**	**1**	**0**	**0**	**0**	**1**

The intermediate message Γ is

$$\Gamma = m_i q(i) + m_j q(j) + m_k q(k) + m_l q(l), \tag{33}$$

where $q(i), q(j), q(k)$ and $q(l)$ are

$$\begin{aligned} q(i) &= q_9 q_{10} q_{11} q_{12} q_{13} q_{14} q_{15} q_{16}, \\ q(j) &= q_4 q_6 q_7 q_8 q_{10} q_{11} q_{14} q_{16}, \\ q(k) &= q_3 q_5 q_7 q_8 q_9 q_{11} q_{13} q_{16}, \\ q(l) &= q_2 q_5 q_6 q_8 q_9 q_{10} q_{12} q_{16}. \end{aligned} \tag{34}$$

Alice is able to know that Bob randomly has chosen M_i, M_j, M_k and M_l from the relation :

$$\Gamma \equiv 0 \mod q_{16}. \tag{35}$$

The messages m_i, m_j, m_k and m_l can be decoded according to the following steps with DA(Ⅱ).

Step 1 : $q(i)^{-1}\Gamma \equiv m_i \bmod q_8$.
Step 2 : $q(j)^{-1}(\Gamma - m_i q(i)) \equiv m_j \bmod q_5 q_9$.
Step 3 : $q(k)^{-1}(\Gamma - m_i q(i) - m_j q(j)) \equiv m_k \bmod q_2 q_6 q_{10} q_{12}$.
Step 4 : $\Gamma - m_i q(i) - m_j q(j) - m_k q(k) = \Gamma_2 - m_k q(k) = m_l q(l)$, yielding m_l.

We see that the factor ratio r_F is

$$r_F = \frac{2^{g-1}|p|/2}{2^{g-1}|p|} = \frac{1}{2}. \tag{36}$$

Let the sizes of messages be

$$\begin{aligned}|m_i| &= \lambda|p| - 1,\\ |m_j| &= 2\lambda|p| - 1,\\ |m_k| &= 4\lambda|p| - 1,\\ |m_l| &= 4\lambda|p| - 1.\end{aligned} \tag{37}$$

The sizes of the intermediate message, public key and ciphertext are

$$\begin{aligned}|\Gamma| &= |m_l| + 2^{\lambda-1}\lambda|p|,\\ |k_i| &= |W| = |\Gamma| + 1,\\ |C| &= |k_i| + 2^{\lambda-2}\lambda|p| + \log_2 2.\end{aligned} \tag{38}$$

The coding rate ρ is given by

$$\rho = \frac{|m_i| + |m_j| + |m_k| + |m_l|}{|m_l| + 32|p| + 1 + 16|p|} \cong \frac{44|p|}{64|p|} = 0.688,\ \text{for } |p| \gtrsim 32. \tag{39}$$

Let the probability that all the elements of $S_B = \{k_i\}_B$ is correctly estimated by an attacker be denoted $P_C[\hat{S}_B]$. The $P_C[\hat{S}_B]$ is

$$P_C[\hat{S}_B] = \begin{pmatrix} n \\ \lambda \end{pmatrix}^{-1}. \tag{40}$$

In Table 2 we present several examples of K(Ⅱ)ΣΠPKC under the condition that

$$P_C[\hat{S}_B] < 2^{-80} = 8.27 \times 10^{-25}, \tag{41}$$

where $|p_i| = 80$(bit).

We see that the coding rate ρ is much improved with DA(Ⅱ).

Table. 2: Examples of K(Ⅱ)ΣΠPKC with DA(I) and DA(Ⅱ)

Decoding Algorithm	Example	n	p	λ	$P_C[\widehat{S}_B]$	$\lvert\{k_i\}_A\rvert$ (MB)	$\lvert\{k_i\}_B\rvert$ (KB)	ρ	r_F
DA(I)	I	4095	80	8	5.13×10^{-25}	83.9	164	0.059	0.5
	Ⅱ	32767	80	6	5.18×10^{-25}	335.6	1014	0.176	0.5
DA(Ⅱ)	Ⅲ	4095	80	8	5.13×10^{-25}	83.9	164	0.746	0.5

2.8 Security considerations on K(V)ΣΠPKC

Remark 1 :

For any given message $\widetilde{m}_1, \widetilde{m}_2, \cdots, \widetilde{m}_\lambda$, we assume that Bob chooses encryption key $\widetilde{k}_1^{(1)}, \widetilde{k}_2^{(2)}, \cdots, \widetilde{k}_\lambda^{(\lambda)}$ all over again from his key set $\{k_i\}_B$. The order $\sharp\{k_i\}_B$ is made so that $\begin{pmatrix} \sharp\{k_i\}_B \\ \lambda \end{pmatrix}$ may take on a sufficiently large value of 2^{15} for $\lambda = 6$. When Bob wants to his set of chosen public key, $\{k_i\}_B$, for a relatively long period of time, the order of $\{k_i\}_B, \sharp\{k_i\}_B$ is recommended to large so that the relation may hold :

$$\begin{pmatrix} \sharp\{k_i\}_B \\ \lambda \end{pmatrix}^{\lambda} \cong 2^{80} \tag{42}$$

Attack 1 : Exhaustive attack on $\{k_i\}_B$

By letting n be sufficiently large and appropriately determing the size of λ, the probability of successfully estimating the subset of $\{k_i\}_B$, $P_C[\hat{S}_B]$, can be made sufficiently small.

Attack 2 : Attack on secret keys

In a sharp contrast with the conventional knapsack type PKC where super-increasing sequence or shifted-odd sequence is used, K(Ⅱ)ΣΠPKC uses a uniform sequence whose components have exactly same entropy. Namely a random product of the same number of prime numbers of the same size. Thus it seems very hard to attack on the secret keys $k_1,\ k_2,\ \cdots,\ k_n$. However the factor rate of secret keys $\{q_i\}$ takes on 0.5. As a result K(Ⅱ)ΣΠPKC would be threatened by Factor Attack.

Attack 3 : LLL attack on the ciphertext

In K(V)ΣΠPKC, n takes on a sufficiently large value although λ is a small value, realizing a sufficiently high security, for the LLL attack.

Attack 4 : Shamir's attack on secret key

As the components of the secret sequence has the same entropy, K(V)ΣΠPKC can be secure against the Shamir's attack.

3 K(V)ΣΠPKC with decoding algorithm, DA(Ⅲ)

In order to improve the factor ratio of K(Ⅱ)ΣΠPKC given in Section 2, we let the composite number q_i be transformed into

$$q_i \mapsto q_i R_i \quad ; \quad i = 1, 2, \cdots, n \tag{43}$$

where R_i is a large prime number of size $= 2^{g-1}\mu p$.

It is easy to see that the factor ratio r_F is given by

$$r_F = \frac{2^{g-2}}{2^{g-1} + \mu \cdot 2^{g-1}} = \frac{1}{2(1+\mu)}. \tag{44}$$

Let us refer to a revised version of K(Ⅱ)ΣΠPKC with a new set of composite numbers $\{q_i R_i\}$ will be referred to as K(V)ΣΠPKC.

An example of Problem A

Construc $\{q_1\}$ so that ciphertext : $\widetilde{C} = \widetilde{m}_1\widetilde{k}_1 + \widetilde{m}_2\widetilde{k}_2 + \cdots + \widetilde{m}_8\widetilde{k}_8$ may be decoded as

$$\widetilde{C} \mapsto (\widetilde{m}_1, \widetilde{m}_2, \cdots, \widetilde{m}_8)$$

under the condition that factor ratio $r_F \cong 0.05$ and coding rate $\rho \cong 2/3$.

Let us show an example of composite numbers for $n = 4095$, in Fig.3.

The set of $k_1, \cdots, k_{4095}$ are constructed from almost random sequence $Q_1, Q_2, \cdots, Q_{4095}$ such that $r_F \lesssim 0.05$.

Several examples of r_F's and ρ's for $g \gtrsim 6$ are shown in Table 3.

We conclude that K(V)ΣΠPKC would be secure against Factor Attack when $\mu \geq 2$, although coding rate takes on a little smaller value, 1/2.

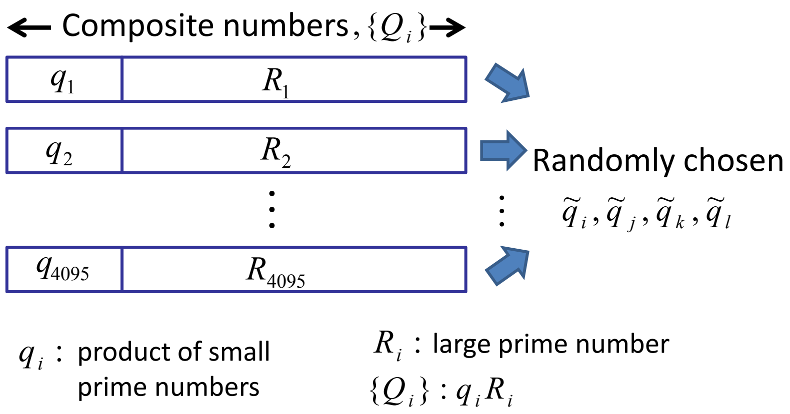

Figure 3: Example of composite numbers of Problem A

Table. 3: r_F and ρ

μ	r_F	ρ
2	1/6	1/2
4	1/10	1/3
8	1/18	1/4

4 New decoding algorithm DA(Ⅲ) for K(V)ΣΠPKC

For an easy understanding, in the followings, let us explain a new decoding algorithm, referred to as DA(Ⅲ), using Example 2 given in 2.7.

In K(V)ΣΠPKC the composite numbers $q(i), q(j), q(k)$ and $q(l)$ in Example 2 are transformed into

$$\begin{aligned} q(i) &= q_9 q_{10} q_{11} q_{12} q_{13} q_{14} q_{15} q_{16} R_i \\ q(j) &= q_4 q_6 q_7 q_8 q_{10} q_{11} q_{14} q_{16} R_j \\ q(k) &= q_3 q_5 q_7 q_8 q_9 q_{11} q_{13} q_{16} R_k \\ q(l) &= q_2 q_5 q_6 q_8 q_9 q_{10} q_{12} q_{16} R_l \end{aligned} \tag{45}$$

The decoding of messages m_i, m_j, m_k and m_l can be performed exactly similar manner as in Example 2. Namely the messages m_i, m_j, m_k and m_l can be decoded according to the following steps:

The m_i and m_j can be decoded with Steps 1 and 2 given in Section 2.7, yielding, $m_k q(k) R_k + m_l q(l) R_l$.

Step 3 : $(R_k q(k))^{-1}(\Gamma - m_i q(i) R_i - m_j q(j) R_j) = m_k \bmod \bar{q}^{(M_l)} R_l$.

Step 4 : $(R_l q(l))^{-1}(\Gamma - m_i q(i) R_i - m_j q(j) R_j) = m_l \bmod \bar{q}^{(M_k)} R_k$.

In Example 2, the size of the messages m_i, m_j, m_k and m_l can be made

$$\begin{aligned} |m_i| &= 4p \text{ (bit)}, \\ |m_j| &= 8p \text{ (bit)}, \\ |m_k| &= 48p \text{ (bit)}, \\ |m_l| &= 48p \text{ (bit)}. \end{aligned} \tag{46}$$

Let the sizes $|R|$ of R_i, R_j, R_k and R_l be $|R| = |R_i| = |R_j| = |R_k| = |R_l| = 32p$ (bit).

The factor ratio r_F and the coding rate ρ are

$$\rho = \frac{4+8+48+48}{48+48+32+32} = \frac{108}{160} = 0.675, \tag{47}$$

$$r_F = \frac{16p}{32p+32p} = \frac{1}{4}. \tag{48}$$

We see that the factor ratio can be improved by using DA(Ⅲ).

It is easy to see that the following relation asymptotically holds:

$$\rho = 2/3, r_F = 0 \text{ as } |R| \to \infty. \tag{49}$$

We thus conclude that the secret sequence used for K(V)ΣΠPKC can be made almost perfectly random. In other words K(V)ΣΠPKC, a product-sum type (knapsack-type) PKC, would be secure against the various conventional attacks.

For example, when $n = 4095, |p_i| = 32, \lambda_{qo} = 8$, the factor ratio r_F and coding rate ρ are

$$\begin{aligned} r_F &\cong 0.045, \\ \rho &\cong \frac{2}{3}. \\ |\{k_i\}_A| &= 302\text{MB} \\ |\{k_i\}_B| &= 58.9\text{KB} \end{aligned} \tag{50}$$

5 Conclusion

We have presented a new class of PKC, K(V)ΣΠPKC.

We have clarified the following results on K(V)ΣΠPKC:

- In a sharp ccontrast with the convetional knapsack PKC where the super-increasing sequence or shifted-odd sequence is used, in K(V)ΣΠPKC, a uniform sequence is used.
- We have presented a generalized version of K(Ⅱ)ΣΠPKC referred to as K(V)ΣΠPKC, by appending a large prime numbers to the secret composite numbers. We have presented a new decoding algorithm DA(Ⅲ) and have shown that the following relation holds: $r_F \cong 0, \rho \cong \frac{2}{3}$, as size of $R_i, |R|$ increases. Thus secret key can be made almost random in a sense that factor ration can be made $r_F \cong 0$. We conclude that K(V)ΣΠPKC can be secure against Factor Attack.
- K(V)ΣΠPKC can be secure against the various attacks such as LLL attack.

References

[1] M.Kasahara, “Construction of New Classes of Knapsack Type Public Key Cryptosystem Using Uniform Secret Sequence, K(Ⅱ)ΣΠPKC, Constructed Based on Maximum Length Code”, Cryptology ePrint Archive, 2012/344 (2012).

[2] R.McEliece, “A Public-Key Cryptosystem Based on Algebraic Coding Tehory”, DSN Progress Report, pp.42-44, (1978).

[3] M.Kasahara, “A New Class of Public Key Cryptosystems Constructed Based on Perfect Error-Correcting Codes Realizing Coding Rate of Exactly 1.0”, Cryptdogy ePrint Archive, 2010/139 (2010).

[4] M.Kasahara, “A New Class of Public Key Cryptosystems Constructed Based on Error-Correcting Codes Using K(Ⅲ) Scheme”, Cryptdogy ePrint Archive, 2010/341 (2010).

[5] M.Kasahara, “Public Key Cryptosystems Constructed Based on Random Pseudo Cyclic Codes, K(Ⅸ)SE(1)PKC, Realizing Coding Rate of Exactly 1.0”, Cryptdogy ePrint Archive, 2011/545 (2011).

[6] R.C. Merkle and M.E. Hellman, “Hiding information and signatures in trapdoor knapsacks”, IEEE Trans. Inf. Theory, IT-24(5), pp.525-530, (1978).

[7] A. Shamir, “A polynomial-time algorithm for breaking the basic Merkle-Hellman cryptosystem”, Proc. Crypto’82, LNCS, pp.279-288, Springer-Verlag, Berlin, (1982).

[8] E.F. Brickell, “Solving low density knapsacks”, Proc. Crypto’83, LNCS, pp.25-37, Springer-Verlag, Berlin, (1984).

[9] J.C. Lagarias and A.M. Odlyzko, “Solving Low Density Subset Sum Problems”, J. Assoc. Comp. Math., vol.32, pp.229-246, Preliminary version in Proc. 24th IEEE, (1985).

[10] M.J. Coster, B.A. LaMacchia, A.M. Odlyzko and C.P. Schnorr, “An Improved Low-Density Subset Sum Algorithm”, Advances in Cryptology Proc. EUROCRYPT’91, LNCS, pp.54-67. Springer-Verlag, Berlin, (1991).

[11] Leonard M.Adleman, “On Breaking Generalized Knapsack Public Key Cryptosystms”, In Proceedings of the Fifteenth Annual ACM Symposium on Theory of Computing. AXM, pp.402-412, (1983).

[12] B. Chor and R.L. Rivest, “A knapsack-type public-key cryptosystem based on arithmetic in finite fields”, IEEE Trans. on Inf. Theory, IT-34, pp.901-909, (1988).

[13] M.Kasahara and Y.Murakami, “New Public-Key Cryptosystems”, Tecnical Report of IEICE, ISEC 98-32 (1998-09).

[14] M.Kasahara and Y.Murakami, “Several Methods for Realizing New Public Key Cryptosystems”, Technical Report of IEICE, ISEC 99-45 (1999-09).

[15] R.Sakai and Y.Murakami and M.Kasahara, ‘Notes on Product-Sum Type Public Key Cryptosystem”, Technical Report of IEICE, ISEC 99-46 (1999-09).

[16] M.Kasahara, “A Construction of New Class of Knapsack-Type Public Key Cryptosystem, K(I)ΣPKC, Constructed Based on K(I)Scheme”, IEICE Technical Report, ISEC, Sept, (2010-09).

[17] M.Kasahara, “A Construction of New Class of Knapsack-Type Public Key Cryptosystem, K(Ⅱ)ΣPKC”, IEICE Technical Report, ISEC, Sept, (2010-09).

[18] M. Kasahara: “Construction of A New Class of Product-Sum Type Public Key Cryptosystem, K(Ⅳ)ΣPKC and K(I)ΣPKC”, IEICE Tech. Report, ISEC 2011-24 (2011-07).

[19] Y. Murakami and M. Kasahara: “A Probabilistic Knapsack Public-Key Cryptosystem”, SITA2010, 30-2.pdf, pp.615-618 (2010-11).

[20] Y. Murakami, S. Hamasho and M. Kasahara: "A probabilistic encryption scheme based on subset sum problem", Proc. 2012 Symposium on Cryptograpy and Information Security, SCIS2012, 3A1-2, 3A1-2.pdf (2012-01).

[21] Y. Murakami, S. Hamasho and M. Kasahara: "A Knapsack Public-Key Cryptosystem using Random Secret sequence", Proc. 2012 Symposium on Cryptograpy and Information Security, SCIS2012, 3A2-1, 3A2-1.pdf (2012-01).

[22] W. W. Peterson: "Error correcting codes", M.I.T. Press (1961).