Product Sum Cryptosystem with Powered Messages using Chinese Remainder Theorem

Yasuyuki Murakami *　　Masao Kasahara †

Abstract

In 2006, Kasahara and Murakami proposed two types of product-sum PKCs, KM-Fundamental PKC and KM-CR, and presented challenge problems of them. However, Nguyen solve most of all of the challenge problems except KM-CR. From the fact that the challenge problem of KM-CR has not been solved yet, KM-CR is considered to be an invulnerable scheme. In this paper, we describe KM-Fundamental and Nguyen's attack against KM-Fundamental. We then revisit KM-CR, discuss about the security and resubmit the challenge problem. Finally we would like to call for attack again.

1 Introduction

The security of most of the public-key cryptosystem depends on the difficulty of the factoring problem, the discrete logarithm problem or the elliptic curve discrete logarithm problem. However, it is shown that the quantum computer can solve these problems in polynomial time[1]. Thus, it is desired to investigate other classes of PKCs(Public Key Cryptosystems) that do not rely on the difficulty of these problems.

It is believed that even the quantum computer can not solve NP-hard problems. The subset-sum problem and the shortest vector problem are known to be NP-hard. The former is used in the knapsack PKC, and the latter, the product-sum PKC.

In 1978, the first knapsack PKC was proposed by Merkle and Hellman[2]. However, Shamir proposed the attack that can compute the secret key of Merkle-Hellman scheme from the public key[3, 4]. Merkle-Hellman scheme can be also broken with the low-density attack[5, 6]. Concerning knapsack-type PKC, the various interesting schemes have been also proposed. In 1979, another interesting knapsack-type PKC was proposed by Lu and Lee[7]. However, soon after the proposal, Lu-Lee scheme was broken by Adleman and Rivest[8].

The product-sum PKC is considered to be the generalized version of the knapsack PKC. The Lu-Lee scheme can be regarded as a kind of product-sum PKC. The authors proposed some product-sum type PKCs[9, 10, 11].

Let us define here the rate R and the density D as follows:

$$R = \frac{\text{size of message (in bits)}}{\text{size of ciphertext (in bits)}},$$

$$D = \frac{\text{size of pertinently enlarged message (in bits)}}{\text{size of ciphertext (in bits)}}.$$

*Department of Telecommunications and Computer Networks, Faculty of Information and Communication Engineering, Osaka Electro-Communication University

†Research Institute for Science and Engineering, Waseda University

2017 年 5 月 1 日受理

It is seen that the density D and the rate R are equal when the messages are not enlarged.

In the product-sum PKCs as well as in the knapsack schemes, the message can be computed from the ciphetext with the low-density attack when the density is low[12]. Thus, it is important that the message is enlarged before encryption in order to realize a high density. The authors proposed several high-density product-sum type schemes[13, 14, 15].

In SCIS2006, Kasahara and Murakami proposed product-sum type PKCs, KM-Fundamental PKC(Kasahara Murakami-Fundamental PKC)[1] and KM-CR PKC(Kasahara Murakami-Chinese Remainder PKC) which enlarge the message by powering by a small number like RSA scheme[15]. Indeed, these schemes are secure against the low-density attack because the density is sufficiently high. The authors also presented challenge problems of these schemes and called for attack[**?**]. Nguyen solved the challenge problems except KM-CR by computing secret key as like Shamir's attack[16]. From the fact that the challenge problem of KM-CR has not been solved yet, KM-CR can be considered invulnerable. Thus, it would be reasonable to resubmit the unsolved challenge problem of KM-CR.

In this paper, we describe KM-Fundamental and Nguyen's attack against the simple challenge problem of KM-Fundamental. We then revisit KM-CR, discuss about the security and resubmit the simple challenge problem of KM-CR.

2 KM-Fundamental PKC

In this section, we shall describe KM-Fundamental PKC where the system parameter e satisfies $e \geq 3$.

2.1 Generation of Secret Keys and Public Keys

Let us first define the symbols, where we let $i = 1, 2, 3.$:

$|I|$: size of integer I (in bits);

b_i: random positive integer bases;

N: modulus of a random positive integer;

e: public system parameter;

d_i: inverse element of e modulo $\lambda(b_i)$;

$\lambda(\cdot)$: Carmichael function;

S_{pk}: size of public key.

We shall now present the outline of the algorithm for the key generation.

[Key Generation Algorithm]

Step 1: Generate $(h+1)$-bit random integers[2] b_1, b_2 and b_3 such that $\gcd(b_i, b_j) = 1$ for $i \neq j$.

Step 2: Generate a secret key u for which the relation $\gcd(u, N) = 1$ holds.

[1] In [15], KM-Fundamental PKC was proposed as KM(3)-II PKC.

[2] In order to realize a high density, we strongly recommend to use the following b_i: $b_i = 2^h + \varepsilon_i$, where $1 \ll \varepsilon_i \ll 2^h$.

Step 3: Let $b'_i = (b_1 b_2 b_3)/b_i$. Given b_1, b_2, b_3, u and N, the public keys a_1, a_2 and a_3 are obtained as follows:

$$a_i = u b'_i \bmod N. \tag{1}$$

□

The secret keys and public keys are given as follows:

Secret key : b_1, b_2, b_3, u, N
Public key : a_1, a_2, a_3

2.2 Encryption

Letting the messages, m_1, m_2 and m_3 be h-bit positive integers, the encryption can be performed in the following manner:

$$C = a_1 m_1^e + a_2 m_2^e + a_3 m_3^e. \tag{2}$$

We see that the encryption can be performed very fast.

2.3 Decryption

Letting the intermediate message M be

$$M = b'_1 m_1^e + b'_2 m_2^e + b'_3 m_3^e, \tag{3}$$

the decryption can be performed in the following manner:

[Decryption Algorithm]

Step 1: The intermediate message M can be obtained as follows:

$$M = u^{-1} C \bmod N. \tag{4}$$

Step 2: The messages m_1, m_2 and m_3 can be obtained as follows:

$$m_i = (b_i'^{-1} M)^{d_i} \bmod b_i, \tag{5}$$

where $d_i \equiv e^{-1} \pmod{\lambda(b_i)}$.

□

We see that the decryption can be performed also very fast.

2.4 Design Conditions

Condition 1 (Decryption)

$$M < N. \tag{6}$$

Condition 2 (Density over 1)

$$|C| < |m_1^e| + |m_2^e| + |m_3^e|. \tag{7}$$

The size of each term of M can be estimated as $|b_i' m_i^e| \simeq (e+2)h$ bit. By adding the number of bits of carrying-up, the size of the intermediate message M can be estimated as $(e+2)h+2$ bit. From Condition 1, the size of the modulus N can be also estimated as $(e+2)h+2$ bit. Consequently, the size of each term of C can be estimated as $|a_i m_i^e| \simeq (2e+2)h+2$ bit. By adding the number of bits of carrying-up, the size of ciphertext C can be estimated as $(2e+2)h+4$ bit. The total size of message and that of the pertinently enlarged message are $3h$ bit and $3eh$ bit, respectively. Thus the rate R and the density D are given by

$$R = \frac{3h}{(2e+2)h+4}, \tag{8}$$

$$D = \frac{3eh}{(2e+2)h+4}. \tag{9}$$

For satisfying Condition 2, we have

$$(e-2)h > 4. \tag{10}$$

We see that $e \geq 3$ is required in order to obtain a high density over 1.

3 Nguyen's Attack for KM-Fundamental

3.1 Simple Challenge Problem of KM-Fundamental

In [15], we proposed a simple challenge problem of KM-Fundamental with only 906-bit public key and 480-bit ciphertext and called for attack. However, the problem has been broken by Nguyen[16].

In this section, we represent the simple challenge problem of KM-Fundamental.

Simple Challenge Problem of KM-Fundamental

$e = 3$, $|m_i| = 60$bit, $|C| = 480$bit, $S_{pk} = 906$bit,
Public Key:
$a_1 = 1310094714668124925591873601933757628299390167237637612376300404621928503560543743327380847$,
$a_2 = 4911492739270653495296997799033661439206560171382214769805578752497330214446509237880958602$,
$a_3 = 1805283598097200756346687715307430041381092846290659965715777427018115107951883376245628902$,
Ciphertext:
$C = 30608660574379179704289879614955693467362033170269504878753573283024880018416437372540711054637 7$
$7766802940192216671167024933872475855670912918886$.
The density D and the rate R are $D \simeq 1.125$ and $R \simeq 0.375$, respectively.

3.2 Nguyen's Attack for Simple Challenge Problem of KM-Fundamental

Nguyen proposed an attack against the above simple challenge problem of KM-Fundamental[16]. In this section, we shall describe the Nguyen's attack.

In KM-Fundamental, it follows that

$$a_i b_i \equiv u b_1 b_2 b_3 \pmod{N} \tag{11}$$

for $i = 1, 2, 3$ from Eq. (1). Thus, it follows that

$$a_i b_i \equiv a_j b_j \pmod{N} \tag{12}$$

for all i, j. Thus,

$$a_1 b_1 - a_3 b_3 = c_1 N; \tag{13}$$

$$a_2 b_2 - a_3 b_3 = c_2 N, \tag{14}$$

where c_1 and c_2 are approximately h-bit integers. Consequently, the following equation holds:

$$a_1 b_1 c_2 - a_2 b_2 c_1 + a_3 b_3 (c_1 - c_2) = 0. \tag{15}$$

In other words, the secret vector $\boldsymbol{s} = (b_1 c_2, -b_2 c_1, b_3(c_1 - c_2))$ belongs to the lattice L spanned by the following row matrix:

$$\begin{pmatrix} \lambda a_1 & 1 & 0 & 0 \\ \lambda a_2 & 0 & 1 & 0 \\ \lambda a_3 & 0 & 0 & 1 \end{pmatrix}, \tag{16}$$

where λ is an arbitrary integer.

Using a lattice reduction algorithm to L for an appropriately-chosen large integer λ on this challenge problem[3], the secret vector $\boldsymbol{s}$ can be obtained as the shortest vector of the reduced basis as follows:

$$\begin{aligned} \boldsymbol{s} = \quad & \pm(675631984421773376801865248254577696, \\ & 107708622938802140442523580109680195, \\ & -783340641250499071658778379692898501) \\ = \quad & \pm(25 \times 14947 \times 1225198472471 \times 1152921508682981069, \\ & 5 \times 37 \times 504985603095401 \times 1152921527016650147, \\ & 679439666749418369 \times 1152921561083068229) \end{aligned}$$

It is seen that the largest prime factor of each component is very close to $2^{60} = 115292150 4606846976$. He thus guess that

$$\begin{aligned} b_1 &= 1152921508682981069, \\ b_2 &= 1152921527016650147, \\ b_3 &= 1152921561083068229. \end{aligned}$$

He can deduce

$$\begin{aligned} N = \; & 611110822216635320211719338072697926849414833129845014949693857702136560 6 \\ & 023520906499789179 \end{aligned}$$

from $N \gcd(c_1, c_2) = \gcd(a_1 b_1 - a_3 b_3, a_2 b_2 - a_3 b_3)$. The b_i''s are easily deduced from the b_i's:

$$\begin{aligned} b_1' &= b_2 b_3 = 1329228086734311109895368248223879663, \\ b_2' &= b_1 b_3 = 1329228065597028736107232068842356801, \\ b_3' &= b_1 b_2 = 1329228026321122605582605774197067143. \end{aligned}$$

He thus obtain:

$$\begin{aligned} u &\equiv a_i b_i'^{-1} \pmod{N} \\ &\equiv 147526558477179097567884256739215668900343605034318411309221850277888503 9 \\ &\quad 528569736575007537. \end{aligned}$$

[3]Nguyen could solve the problem by setting $\lambda = 10^{100}$.

He then recovered the plaintext as

$$\begin{aligned}
m_1 &= 1019888794176532808,\\
m_2 &= 623429283177510895,\\
m_3 &= 643459722121049246.
\end{aligned}$$

4 KM-CR PKC

In this section, we shall revisit KM-CR PKC which uses the Chinese remainder theorem.

4.1 Generation of Secret Keys and Public Keys

Let us first define the symbols: In the following, we let $i = 1, 2, \ldots, n$.

$$\begin{aligned}
v_i^{(P)}, v_i^{(Q)} &: \text{random positive integers;}\\
b_i^{(P)}, b_i^{(Q)} &: \text{random positive integer bases;}\\
P, Q &: \text{prime numbers;}\\
N &: \text{composite modulus where } N = PQ;\\
e &: \text{public system parameter;}\\
d_i^{(P)} &: \text{inverse element of } e \text{ modulo } \lambda(b_i^{(P)});\\
d_i^{(Q)} &: \text{inverse element of } e \text{ modulo } \lambda(b_i^{(Q)}).
\end{aligned}$$

We shall now present the algorithm for key generation.

[Key Generation Algorithm]

Step 1: Generate random positive integers $b_i^{(P)}$ and $b_i^{(Q)}$ for $i = 1, 2, \ldots, n$ such that

$$b_i^{(P)} b_i^{(Q)} = 2^h + \varepsilon_i, \tag{17}$$

$$\gcd(b_i^{(P)}, b_i^{(Q)}) = 1, \tag{18}$$

$$\gcd(b_i^{(P)}, b_j^{(P)}) = 1, \quad \text{for } i \neq j, \tag{19}$$

where $1 \ll \varepsilon_i \ll 2^h$.

Step 2: Generate l-bit random positive integers[4] $v_i^{(P)}$ and $v_i^{(Q)}$ for $i = 1, 2, \ldots, n$ such that

$$\gcd(v_i^{(P)}, b_i^{(P)}) = 1, \tag{20}$$

$$\gcd(v_i^{(Q)}, b_i^{(Q)}) = 1. \tag{21}$$

Step 3: Generate a secret key u for which the relation $\gcd(u, N) = 1$ holds.

[4] $v_i^{(P)} = 1$ and $v_i^{(Q)} = 1$ for $i = 1, 2, \ldots, n$ can be possible. In this case the highest density can be obtained.

Step 4: Let $b_i'^{(P)} = (v_i^{(P)} \prod_{k=1}^n b_k^{(P)})/b_i^{(P)}$ and $b_i'^{(Q)} = (v_i^{(Q)} \prod_{k=1}^n b_k^{(Q)})/b_i^{(Q)}$. We can obtain $b_i < N$ with the Chinese remainder theorem for $i = 1, 2, \ldots, n$ as follows:

$$b_i' \equiv \begin{cases} b_i'^{(P)} & \pmod P, \\ b_i'^{(Q)} & \pmod Q. \end{cases} \tag{22}$$

Step 5: Given b_i', u and N, the public keys a_i for $i = 1, 2, \ldots, n$ are obtained as follows:

$$a_i = ub_i' \bmod N. \tag{23}$$

□

The secret keys and public keys are given as follows:

Secret key : $b_i^{(P)}, b_i^{(Q)}, v_i^{(P)}, v_i^{(Q)}, u, N, P, Q$
Public key : a_i

4.2 Encryption

Letting the messages, m_i be h-bit positive integers for $i = 1, 2, \ldots, n$. The ciphertext, $C \in \mathbb{Z}$ is obtained as follows:

Encryption can be performed in the following manner:

$$C = \sum_{k=1}^{n} a_k m_k^e. \tag{24}$$

We see that the encryption can be performed very fast.

4.3 Decryption

Decryption can be performed in the following manner:

Let the intermediate messages $M^{(P)}$ and $M^{(Q)}$ be

$$M^{(P)} = \sum_{k=1}^{n} b_k'^{(P)} m_k^e, \tag{25}$$

$$M^{(Q)} = \sum_{k=1}^{n} b_k'^{(Q)} m_k^e. \tag{26}$$

[Decryption Algorithm]

Step 1: The intermediate messages $M^{(P)}$ and $M^{(Q)}$ can be obtained as follows:

$$M^{(P)} = u^{-1} C \bmod P, \tag{27}$$

$$M^{(Q)} = u^{-1} C \bmod Q. \tag{28}$$

Step 2: Letting

$$m_i^{(P)} = m_i \bmod b_i^{(P)}, \tag{29}$$
$$m_i^{(Q)} = m_i \bmod b_i^{(Q)}, \tag{30}$$

$m_i^{(P)}$ and $m_i^{(P)}$ can be obtained as follows:

$$m_i^{(P)} = ({b_i'^{(P)}}^{-1} M)^{d_i^{(P)}} \bmod b_i^{(P)}, \tag{31}$$
$$m_i^{(Q)} = ({b_i'^{(Q)}}^{-1} M)^{d_i^{(Q)}} \bmod b_i^{(Q)}, \tag{32}$$

where $ed_i^{(P)} \equiv 1 \pmod{\lambda(b_i^{(P)})}$ and $ed_i^{(Q)} \equiv 1 \pmod{\lambda(b_i^{(Q)})}$. Thus the messages $m_i < b_i^{(P)} b_i^{(Q)}$ for $i = 1, 2, \ldots, n$ can be obtained from $m_i^{(P)}$ and $m_i^{(Q)}$ with the Chinese remainder theorem:

$$m_i \equiv \begin{cases} m_i^{(P)} & \pmod{b_i^{(P)}}, \\ m_i^{(Q)} & \pmod{b_i^{(Q)}}. \end{cases} \tag{33}$$

□

We also see that the decryption can be performed very fast.

4.4 Design Conditions

Condition 3 (Decryption)

$$M^{(P)} < P, \tag{34}$$
$$M^{(Q)} < Q. \tag{35}$$

Condition 4 (Density over 1)

$$|C| < \sum_{k=1}^{n} |m_k^e|. \tag{36}$$

Assuming that $|b_i^{(P)}| = |b_i^{(Q)}| = h/2$, the size of each term of $M^{(P)}$ can be estimated by $|b_i'^{(P)} m_i^e| \simeq (e + n/2 - 1/2)h + l$ bit. By adding the number of bits of carrying-up, the size of the intermediate message $M^{(P)}$ can be estimated by $(e + n/2 - 1/2)h + l + 2$ bit. The size of $M^{(Q)}$ can be similarly estimated by $(e + n/2 - 1/2)h + l + 2$ bit. The size of the modulus N can be estimated by $(2e+n-1)h+2l+4$ bit. Consequently, the size of each term of C can be estimated by $|a_i m_i^e| \simeq (3e + n - 1)h + 2l + 4$ bit. By considering the number of bits of carrying-up, the size of the ciphertext C can be estimated by $(3e+n-1)h+2l+6$ bit. The total size of message and that of pertinently enlarged message are nh bit and neh bit, respectively. Thus the density R and the rate D are represented by

$$R = \frac{nh}{(3e + n - 1)h + 2l + 6}, \tag{37}$$
$$D = \frac{neh}{(3e + n - 1)h + 2l + 6}. \tag{38}$$

From Condition 4, $D > 1$ must be required to be secure against the low-density attack. It is required that

$$neh > (3e + n - 1)h. \tag{39}$$

Consequently, we have

$$n > \frac{3e-1}{e-1}. \tag{40}$$

We see that $n \geq 5$ is required when $e = 3$ and that $n \geq 4$ is required when $e = 5$, in order to obtain a high density over 1. It is recommended that $e = 3$ and $n = 5$ in order to obtain a relatively high rate.

5 Discussions

In this subsection, we shall discuss about the security on KM-CR PKC.

5.1 Security against Low-Density Attack

Letting (m_1^e, m_2^e, m_3^e) be (x_1, x_2, x_3), we see that the deciphering KM-CR PKC is equivalent to the solving of the following linear Diophantine equation:

$$C = a_1x_1 + a_2x_2 + a_3x_3. \tag{41}$$

Evidently, there exists many solutions of Eq.(41) when $D > 1$ holds. In KM-CR PKC, the ratio of the spaces required for the enlarged message and ciphertext is given approximately by 2^{3eh} : $2^{(3e+n-1)h+2l+6} \simeq 2^{eh} : 1$. Thus the total number of different solutions can be approximately given by $E = 2^{eh}$. For the purpose of only solving Eq.(41), it is required only to obtain an arbitrary solution among many solutions. On the other hand, when deciphering KM-CR PKC, it is required to obtain one and only one correct solution that coincides with the original plaintext. Thus obtaining correct solution in KM-CR PKC can be considered more difficult compared with the solving of linear Diophantine equation of Eq.(41) and would be made difficult by letting E sufficiently large. We can conclude that KM-CR PKC is invulnerable to LDA.

5.2 Security against Exhaustive Search

In KM-CR PKC, the ciphertext C is given by

$$C = \sum_{k=1}^{n} a_i m_i^e. \tag{42}$$

When $\widehat{m_k} = m_k$ holds for $k = 3, \ldots, n$, we obtain the following ciphertext $\widetilde{C}$ which is equivalent to that of the two terms public key cryptosystem:

$$\widetilde{C} = a_1 m_1^e + a_2 m_2^e. \tag{43}$$

It is easy to see that the ciphertext $\widetilde{C}$ can be easily deciphered. Thus in order to make the proposed KM-CR PKC invulnerable to the exhaustive search on m_i, it is recommended that m_i satisfy the following:

$$\sum_{k=3}^{n} |m_k| \geq 128 \quad \text{(in bits)}. \tag{44}$$

Thus in order to let KM-CR PKC be invulnerable to the exhaustive search on m_i, it is recommended that m_i satisfy the following:

$$|m_i| \geq 43 \quad \text{(in bits)}. \tag{45}$$

5.3 Security on Secret Keys

Nguyen's attack for computing secret keys can be generalized as follows.

From Eq.(1), the following relation holds for $i \neq j$:

$$a_i b'_j \equiv a_j b'_i \pmod{N}. \tag{46}$$

Thus

$$a_1 b'_2 - a_2 b'_1 = c_1 N, \tag{47}$$
$$a_1 b'_3 - a_3 b'_1 = c_2 N, \tag{48}$$

where c_1 and c_2 are approximately h-bit integers. By extinguishing N, one can obtain the following equation:

$$a_1(c_2 b'_2 - c_1 b'_3) - a_2 c_2 b'_1 + a_3 c_1 b'_1 = 0. \tag{49}$$

In KM-Fundamental, the secret keys b_i can be easily obtained because the unknown values $c_2 b'_2 - c_1 b'_3$, $c_2 b'_1$ and $c_1 b'_1$ are relatively small compared with a_1, a_2 and a_3.

On the other hand, in KM-CR PKC, the unknown values $c_2 b'_2 - c_1 b'_3$, $c_2 b'_1$ and $c_1 b'_1$ are larger than a_1, a_2 and a_3. Thus, we can conclude that KM-CR can not be broken with Nguyen's secret key attack.

6 Resubmission of Simple Challenge Problem of KM-CR

In [15], we proposed a simple challenge problem of KM-CR with only 2060-bit public key and 530-bit ciphertext and called for attack. However, the problem has not been broken yet, even the parameters are relatively small.

In this section, we resubmit the simple challenge problem of KM-CR and call for attack again. In this example, the size of the public key in KM-CR PKC is equal to that in RSA PKC. However, the sizes of message and ciphertext in KM-CR PKC are respectively shortened by a factor of about 10 and about 2 compared with those in the conventional RSA cryptosystem.

Simple Challenge Problem of KM-CR

$e = 3$, $n = 5$, $|m_i| = 40$bit, $|C| = 530$bit, $S_{pk} = 2060$bit,
Public Key:
$a_1 =$ 3928961564655522098362462415264116432705089160021656976681307849060642042916850359798997162059016915283422816873031893581926,
$a_2 =$ 6588038704281632936925454669632597111304620876795584786306771947655784629568566746479418527624351975634394641852780618742064,
$a_3 =$ 1202523617738508194740602299695228349289499586123679577162153793967960940735188124709150532694730372849261784600059503515 2769,
$a_4 =$ 8689221714349377745293892682451676381556143866527533140024622406588470514154146287844882214497445536903069208756517344413837,
$a_5 =$ 898358134062019842173837528935232617182921712056678130214784970506886502499427673821786134092871742386403807849501786 2145840
Ciphertext:
$C =$ 308785037719256782056777621437323050096998700176880936853932977548742955714015745998791040673637235642020963893560427158333836146029434162811538205765619443 1948.

The density D and the rate R are $D \simeq 1.124$ and $R \simeq 0.375$, respectively.

7 Conclusion

In this paper, we have described KM-Fundamental PKC and Nguyen's attack for the simple challenge problem of KM-Fundamental. We have then revisited KM-CR PKC which has not been solved yet and discussed the security of KM-CR. Moreover, we have resubmitted the simple challenge problem of KM-CR and call for attack again.

One of the important advantages of KM-CR is that the encryption and decryption can be performed very fast. We sincerely wish the resubmitted simple challenge problem be solved by elegant methods.

Acknowledgment

The authors would like to thank Dr. Nguyen for his attack and helpful advise.

References

[1] W. P. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," Proc. the 35th Annual Symposium on Foundations of Computer Science, IEEE Computer Society Press, pp.124–134 (quant-ph/9508027), Santa Fe, NM, Nov. 20–22, 1994.

[2] 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.

[3] A. Shamir, "A polynomial-time algorithm for breaking the basic Merkle-Hellman cryptosystem," Proc. Crypto'82, LNCS, pp.279–288, Springer-Verlag, Berlin, 1982.

[4] A. Shamir, "A polynomial-time algorithm for breaking the basic Merkle-Hellman cryptosystem," IEEE Trans. Inf. Theory, IT-30, pp.699–704, 1984.

[5] 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.

[6] 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.

[7] S. C. Lu and L. N. Lee, "A simple and effective public-key cryptosystem," COMSAT Tech, Rev., 9(1), pp.15–24, 1979.

[8] L. M. Adleman and R. L. Rivest, "How to break the Lu-Lee(comsat) public-key cryptosystem," MIT Laboratory for Computer Science, 1979.

[9] M. Kasahara and Y. Murakami, "New public-key cryptosystems," Technical Report of IEICE, ISEC98-32, pp.33–40, 1998.

[10] M. Kasahara and Y. Murakami, "New product-sum type public-key cryptosystems," Proc. of SCIS'99, pp.15–20, 1999.

[11] M. Kasahara and Y. Murakami, "New public key cryptosystems and the application," Technical Report of IEICE, ISEC99-55, pp.21–28, 1999.

[12] H. Shimizu, "On the security of Kasahara-Murakami cryptosystems," Technical Report of IEICE, ISEC99-56, pp.29–36, 1999.

[13] K. Katayanagi, Y. Murakami and M. Kasahara, "A new product-sum type public key cryptosystem using message extension," IEICE Trans. on Fundamentals, vol.E84-A, no.10, pp.2482–2487, 2001.

[14] M. Kasahara and Y. Murakami, "A proposal of three terms public-key cryptosystem, KM(3)-PKC — along with challenge problems on KM(3)-PKC of very small key-size — ," Technical Report of IEICE, ISEC2005-94, pp.33–37, 2005.

[15] M. Kasahara and Y. Murakami, "A new class of public-key cryptosystem with several terms of product-sum operation," SCIS2006, no.2A3-3, 2A3-3.pdf, Jan. 2006.

[16] P. Q. Nguyen, Private communications from April to June in 2006.