JSIAM Letters Vol.2 (2010) pp.29–32 ©2010 Japan Society for Industrial and Applied Mathematics

# A knapsack public-key cryptosystem with cyclic code over $GF(2)$

Yasuyuki Murakami[1] and Takeshi Nasako[2]

[1] Department of Telecommunications and Computer Networks, Faculty of Information and Communication Engineering, Osaka Electro-Communication University, 18–8, Hatsu-Cho, Neyagawa-shi, Osaka 572–8530, Japan

[2] Graduate School of Engineering, Osaka Electro-Communication University, 18–8, Hatsu-Cho, Neyagawa-shi, Osaka 572–8530, Japan

E-mail *yasuyuki@isc.osakac.ac.jp*

Received November 8, 2009, Accepted February 8, 2010

**Abstract**

It is required to invent the public-key cryptosystem (PKC) that is based on an *NP*-hard problem so that the quantum computer might be realized. The knapsack PKC is based on the subset sum problem which is *NP*-hard. In this paper, we propose a knapsack PKC with a cyclic code over $GF(2)$ using the Chinese remainder theorem. The proposed scheme is secure against Shamir's attack and Adleman's attack and invulnerable to the low-density attack. Furthermore, the proposed scheme can reduce the size of public key by almost 25% $\sim$ 50% of the conventional scheme using a linear code.

**Keywords** knapsack public-key cryptosystem, subset sum problem, cyclic code, Chinese remainder theorem, low-density attack

**Research Activity Group** Algorithmic Number Theory and Its Applications

## 1. Introduction

It was shown that the quantum computer can solve the factoring problem, the discrete logarithm problem and the elliptic curve discrete logarithm problem in a polynomial time [1]. However, it is considered that even the quantum computer can not solve *NP*-hard problems in a polynomial time. Thus, it is required to invent the public-key cryptosystem (PKC) that is based on an *NP*-hard problem so that the quantum computer might be realized. The subset sum problem is one of the *NP*-hard problems.

The subset sum problem is to find the solution $(x_1, x_2, \ldots, x_n) \in \{0,1\}^n$ such that

$$C = a_1x_1 + a_2x_2 + \cdots + a_nx_n$$

for the given positive integers $a_1, a_2, \ldots, a_n$ and the given sum $C$. The public-key cryptosystem using the subset sum problem has been conventionally called the knapsack cryptosystem. The knapsack cryptosystem has a remarkable feature that the encryption can be performed very fast.

The first knapsack PKC was proposed by Merkle and Hellman [2]. However, the secret key can be disclosed by Shamir's attack [3] or Adleman's attack [4] because the public key is generated with a linear transformation of a super-increasing sequence. The plaintext message can be also disclosed with the low-destiny attack (LDA) [5, 6] because the density is not sufficiently high. These attacks have given the impression that knapsack PKCs are insecure. It is, however, difficult to condemn that all the knapsack PKCs are not secure.

In LDA, the subset sum problem is converted into the problem of finding the shortest vector in a lattice. LDA was proposed by Lagarias and Odlyzko for solving the subset sum problem of low density [5]. The density, an important parameter in knapsack schemes, is defined by

$$d = \frac{n}{\log_2[\max(a_1, a_2, \ldots, a_n)]}.$$

Coster et al. improved LDA so that it can solve almost all subset sum problems of the density less than 0.9408 [6]. Nguyen and Stern proposed an adapting density attack for low-weight knapsack PKCs [7], which will be referred to as the low-weight attack (LWA). They showed that LWA could solve the subset sum problem with high probability when Hamming weight is low.

Murakami and Nasako proposed the knapsack PKC with the Chinese remainder theorem (CRT) [8]. The knapsack PKC with CRT can avoid Shamir's attack and Adleman's attack. However, the knapsack PKC with CRT needs a large dimension $n$ for realizing the density invulnerable to LDA. They also proposed the method of encoding the plaintext before encryption with a linear code in order to realize a high density above 1 [9]. However, the size of public key is significantly large when using a linear code as the encoding.

In this paper, we shall propose a knapsack PKC using CRT which uses a cyclic code over $GF(2)$ as the encoding. The proposed scheme is secure against Shamir's attack and Adleman's attack and invulnerable to LDA. The proposed scheme has an advantage that the size of the public key can be made much smaller than the conventional scheme using a linear code.

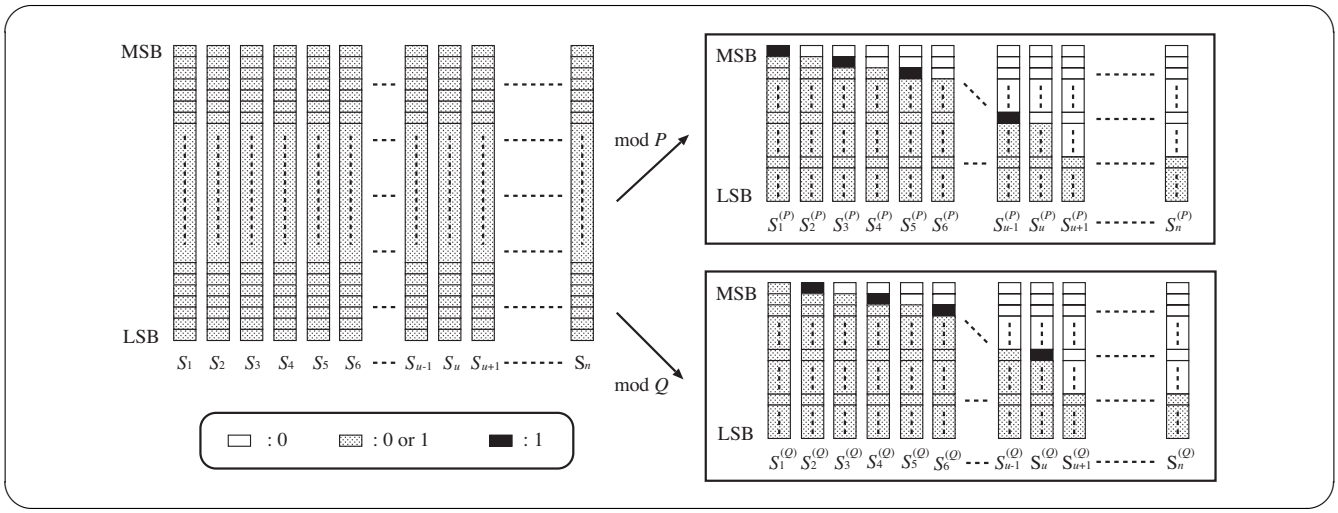


Fig. 1. Trapdoor of the proposed scheme.

## 2. Proposed scheme

In this section, we shall propose a knapsack PKC using a cyclic code over $GF(2)$ with CRT. The proposed scheme adopts the trapdoor sequence proposed in [10], but not limited.

The keys of the proposed knapsack PKC are the followings:

**Public key** $\mathcal{PK}$ : $\mathcal{PK} = \{\boldsymbol{a}, G(x)\}$.

**Secret key** $\mathcal{SK}$ :
$\mathcal{SK} = \{\boldsymbol{s}^{(P)}, \boldsymbol{s}^{(Q)}, \boldsymbol{s}, P, Q, \sigma, T_P, T_Q\}$.

### *2.1 Key generation*

Bob creates a public key $\mathcal{PK}$ and a corresponding secret key $\mathcal{SK}$ by doing the following:

**Algorithm $\mathcal{K}$**

(1) Decide the dimensions $n$ and $u$ such that $n > u$.

(2) Define the sets $T_P$ and $T_Q$ such that $T_P \cup T_Q = \{1, 2, \ldots, u\}$ and $T_P \cap T_Q = \phi$.

(3) For $i = n$ downto $u + 1$ do:
Generate $r$-bit random positive integers $s_i^{(P)}$ and $s_i^{(Q)}$.

(4) For $i = u$ downto 1 do:
Generate random positive integers $s_i^{(P)}$ and $s_i^{(Q)}$ such that

$$\begin{cases} s_i^{(P)} > \displaystyle\sum_{k=i+1}^{n} s_k^{(P)}, & \text{if } i \in T_P, \\ s_i^{(P)} < s_{i+1}^{(P)}, & \text{otherwise,} \end{cases}$$

$$\begin{cases} s_i^{(Q)} > \displaystyle\sum_{k=i+1}^{n} s_k^{(Q)}, & \text{if } i \in T_Q, \\ s_i^{(Q)} < s_{i+1}^{(Q)}, & \text{otherwise.} \end{cases}$$

(5) Choose integers $P$ and $Q$ such that

$$\begin{cases} P > \displaystyle\sum_{k=1}^{n} s_k^{(P)}, \\ Q > \displaystyle\sum_{k=1}^{n} s_k^{(Q)}, \end{cases}$$

and $\gcd(P, Q) = 1$.

(6) Compute $\boldsymbol{s} = (s_1, s_2, \ldots, s_n) \in \mathbb{Z}_{PQ}^n$ such that

$$s_i \equiv \begin{cases} s_i^{(P)} & (\text{mod } P), \\ s_i^{(Q)} & (\text{mod } Q), \end{cases}$$

for $i = 1$ to $n$ with CRT.

(7) Generate a polynomial $F(x)$ of period $n$ over $GF(2)$.

(8) Generate a polynomial $G(x) = g_0 + g_1 x + g_2 x^2 + \cdots + g_{n-1} x^{n-1}$ of degree $n - 1$ such that $F(x)|G(x)$ over $GF(2)$.

(9) Let $S_n$ denote the set of permutations of integers $\{1, 2, \ldots, n\}$. Let the generator matrix $G$ be

$$G = [\boldsymbol{\xi}_1 \mid \boldsymbol{\xi}_2 \mid \cdots \mid \boldsymbol{\xi}_n]$$
$$= \begin{pmatrix} g_0 & g_1 & g_2 & \cdots & g_{n-1} \\ g_{n-1} & g_0 & g_1 & \cdots & g_{n-2} \\ g_{n-2} & g_{n-1} & g_0 & \cdots & g_{n-3} \\ \vdots & \vdots & \vdots & & \vdots \\ g_{n-u+1} & g_{n-u+2} & g_{n-u+3} & \cdots & g_{n-u} \end{pmatrix}.$$

Select a random permutation $\sigma \in S_n$ such that $\det(\widehat{G}) \neq 0$ over $GF(2)$ where $\widehat{G} = [\boldsymbol{\xi}_{\sigma(1)} \mid \boldsymbol{\xi}_{\sigma(2)} \mid \cdots \mid \boldsymbol{\xi}_{\sigma(u)}]$.

(10) Obtain $\boldsymbol{a} = (a_1, a_2, \ldots, a_n) \in \mathbb{Z}_{PQ}^n$ such that

$$a_i = s_{\sigma(i)}.$$

for $i = 1$ to $n$ with the permutation $\sigma$.

(11) Publicize the public key $\{\boldsymbol{a}, G(x)\}$ and the public information on the dimension $\{u, n\}$.

Fig. 1 illustrates the trapdoor of the proposed scheme.

*2.2 Encryption*

Alice encrypts a message $\boldsymbol{m} = (m_1, m_2, \ldots, m_u) \in \{0,1\}^u$ into the ciphertext $C \in \mathbb{Z}$ by doing the following:

**Algorithm $\mathcal{E}$**

(1) Encode the message $\boldsymbol{m}$ into $\boldsymbol{m}' = (m'_1, m'_2, \ldots, m'_n) \in \{0,1\}^n$ as follows:

$$M'(x) = M(x)G(x) \bmod (x^n - 1)$$

where $M(x) = m_1 + m_2 x + \cdots + m_u x^{u-1}$ and $M'(x) = m'_1 + m'_2 x + \cdots + m'_n x^{n-1}$ are polynomial representations of $\boldsymbol{m}$ and $\boldsymbol{m}'$ over $GF(2)$, respectively. It should be noted that $\boldsymbol{m}'$ is a codeword of the cyclic code generated by $F(x)$.

(2) Compute the ciphertext $C \in \mathbb{Z}$ as follows:

$$C = \sum_{i=1}^{n} a_i m'_i.$$

(3) Send the ciphertext $C$ to Bob.

*2.3 Decryption*

Bob decrypts the message $\boldsymbol{m} = (m_1, m_2, \ldots, m_u) \in \{0,1\}^u$ from the ciphertext $C \in \mathbb{Z}$ by doing the following:

**Algorithm $\mathcal{D}$**

(1) Compute $C_P \in \mathbb{Z}_P$ and $C_Q \in \mathbb{Z}_Q$ as follows:

$$\begin{cases} C_P = C \bmod P, \\ C_Q = C \bmod Q. \end{cases}$$

(2) For $i = 1$ to $u$ do:
If $i \in T_P$ {

$$\widehat{m}_i = \begin{cases} 0 & \text{if } C_P < s_i^{(P)}, \\ 1 & \text{if } C_P \geq s_i^{(P)}, \end{cases}$$

} Else {

$$\widehat{m}_i = \begin{cases} 0 & \text{if } C_Q < s_i^{(Q)}, \\ 1 & \text{if } C_Q \geq s_i^{(Q)}, \end{cases}$$

}
$C_P \leftarrow C_P - \widehat{m}_i s_i^{(P)}$,
$C_Q \leftarrow C_Q - \widehat{m}_i s_i^{(Q)}$,
where $\widehat{\boldsymbol{m}} = (\widehat{m}_1, \widehat{m}_2, \ldots, \widehat{m}_u) \in \{0,1\}^u$.

(3) Obtain the message $\boldsymbol{m}$ as follows:

$$\boldsymbol{m} = \widehat{\boldsymbol{m}} \widehat{G}^{-1} \bmod 2.$$

## 3. Discussions

*3.1 Security of secret key*

Several attacks of computing the secret key from the public key are proposed on the knapsack PKC such as Shamir's attack [3] and Adleman's attack [4]. These attacks are effective only when the public key is generated with a modular multiplication of a super-increasing sequence. However, the proposed scheme uses CRT instead of the modular multiplication in order to generate the public key from the secret key. Thus, these attacks are not applicable to the proposed scheme.

*3.2 Security against exhaustive search*

The exhaustive search is an attack by searching plaintext at all possibilities. It requires a great investment of time to search for 80 bits even by the latest computers. Thus, it is required that $u \geq 80$ in order to be secure against the exhaustive search.

*3.3 Security against space-time tradeoff attack*

In general, the computation time can be reduced by increasing the memory use. This type of attacks is called the space-time tradeoff attack. We can reasonably assume that the time complexity of $O(N)$ can be divided into the time complexity of $O(\sqrt{N})$ and the space complexity of $O(\sqrt{N})$. Thus, it is required that $n \geq 160$ in order to be secure against the space-time tradeoff attack.

*3.4 Security against low-density attack*

*3.4.1 Density of proposed scheme*

LDA works effectively for a low-density knapsack PKC, irrespective of the trapdoors. This attack converts the subset sum problem into the problem of finding a short vector in a lattice. LDA proposed by Coster et al. can solve the almost all subset sum problems when the density is less than 0.9408 [6].

The density $d$ of the proposed knapsack scheme is given by

$$d \simeq \frac{n}{\log_2 PQ}.$$

For simplicity, we assume that $u = 2u'$, $T_P = \{1, 3, \ldots, 2u' - 1\}$ and $T_Q = \{2, 4, \ldots, 2u'\}$. Then, we can estimate that $\log_2 P \simeq \log_2 Q \simeq r + u/2 + \log_2(n-u)$ and we have

$$\log_2 PQ \simeq 2r + u + 2\log_2(n-u). \tag{1}$$

Therefore the density $d$ can be estimated by

$$d \simeq \frac{n}{2r + u + 2\log_2(n-u)}.$$

For example, $d > 1$ can be achieved when $r = 40$, $u = 80$, $n = 173$. Therefore, the proposed scheme is invulnerable to LDA because a high density above 1 can be realized by encoding the plaintext before encryption. Thus, we can conclude that the proposed scheme is invulnerable to LDA. We recommend $r \geq 40, u \geq 80$ and $n \geq 160$ in order to realize a high security.

*3.4.2 Effect of encoding*

In the proposed scheme, the $\boldsymbol{m}'$ can be represented as

$$\boldsymbol{m}' = \boldsymbol{m}G \quad \text{over } GF(2). \tag{2}$$

If $\boldsymbol{m}'$ can be represented as

$$\boldsymbol{m}' = \boldsymbol{m}G \quad \text{over } \mathbb{Z}, \tag{3}$$

then $C$ can be represented by

$$C = \sum_{i=1}^{u} a'_i m_i, \tag{4}$$

where we let the $u$-dimensional integer vector $\boldsymbol{a}'$ be $\boldsymbol{a}' = \boldsymbol{a}G^T$ over $\mathbb{Z}$. Indeed, there are several cases that (3)

holds. For example, (3) holds when the generator matrix $G$ is sparse such as $G(x) = 1$.

The bit-length of each $a'_i$ can be estimated by $\lceil \log_2 PQ + \log_2 n \rceil$[bit] at maximum. Thus, it is seen that the density $d'$ of the knapsack $\boldsymbol{a}'$ can be estimated by

$$d' < \frac{u}{\log_2 PQ + \log_2 n}.$$

This means that the proposed scheme would not be secure against LDA if (3) holds. However, there are little cases that (3) holds when $G$ is not sparse. In order to let the generator matrix $G$ be non-sparse, we have only to let the number of terms of $G(x)$ be sufficiently large. We strongly recommend to let the number of terms of $G(x)$ be approximately $n/2$. In this case, the number of non-zero elements of $G$ can be estimated as $un/2$ which is sufficiently large. Thus, we can conclude that it is difficult to convert the proposed scheme into a subset sum problem of low density.

### *3.5 Security against low-weight attack*

The pseudo-density $\kappa$ is defined by

$$\kappa = \frac{k \log_2 n}{\log_2[\max(a_1, a_2, \ldots, a_n)]}$$

where $k$ is Hamming weight of the encoded message $\boldsymbol{m}'$. LWA can solve a subset sum problem with high probability when the pseudo-density $\kappa$ is lower than 1 even if the density $d$ is higher than 1 [7].

Let the number of terms of $G(x)$ be $n/2$ which is recommended value. Assuming that the plaintext message $\boldsymbol{m}$ be randomly generated, Hamming weight of $\boldsymbol{m}'$ can be estimated as $k = n/2$. In this case, it is seen that $\kappa = d \log_2 n/2 > d > 1$ can be realized. Consequently, we can conclude that the proposed scheme is invulnerable to LWA.

### *3.6 Size of public key*

Let the function $I()$ return the total amount of data in parenthesis. From (1), the amount of the public key $\boldsymbol{a}$, $I(\boldsymbol{a})$, can be estimated by

$$I(\boldsymbol{a}) = n[2r + u + 2\log_2(n - u)].$$

In the proposed scheme, the size required for representing $G(x)$ is only $I(G(x)) = n$. Thus, the total amount of the public key in the proposed scheme can be estimated as

$$I(\boldsymbol{a}, G(x)) = n[2r + u + 1 + 2\log_2(n - u)].$$

On the other hand, in the conventional scheme which uses a linear code as the encoding, $I(G) = un$ is required for representing $G$. Thus, the total amount of the public key in the conventional scheme can be estimated as

$$I(\boldsymbol{a}, G) = n[2r + 2u + 2\log_2(n - u)].$$

The ratio of $I(\boldsymbol{a}, G(x))$ to $I(\boldsymbol{a}, G)$ can be estimated as

$$\frac{I(\boldsymbol{a}, G(x))}{I(\boldsymbol{a}, G)} \simeq \frac{2r + u}{2r + 2u}$$

when $n$ and $u$ are sufficiently large. Since we usually let the parameter $u$ be $0 < r < u$, it is seen that the proposed scheme can reduce the size of public key by almost 25% $\sim$ 50% of the conventional scheme. For example, the proposed scheme can reduce the size of public key by almost 30% when $r = 40$, $u = 80$ and $n = 255$. We can conclude that the size of public key in the proposed scheme is sufficiently practical.

## 4. Conclusion

In this paper, we have proposed a knapsack PKC with a cyclic code over $GF(2)$ which uses CRT as the trapdoor. The proposed scheme is secure against Shamir's attack and Adleman's attack which are the attack of computing the secret key. Moreover, the proposed scheme is invulnerable to both LDA and LWA because it can realize a high density above 1 and a high pseudo-density above 1. Furthermore, the proposed scheme can reduce the size of public key by almost 25% $\sim$ 50% of the conventional scheme using a linear code.

## Acknowledgments

This work was supported by SCOPE (Strategic Information and Communications R&D Promotion Programme) from the Ministry of Internal Affairs and Communications of Japan.

## References

[1] W. P. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, in: Proc. of the 35th Annual Symposium on Foundations of Computer Science, pp. 124–134, IEEE Computer Society Press, 1994.

[2] R. C. Merkle and M. E. Hellman, Hiding information and signatures in trapdoor knapsacks, IEEE Trans. Inform. Theory, **IT-24** (1978), 525–530.

[3] A. Shamir, A polynomial time algorithm for breaking the basic Merkle-Hellman cryptosystems, in: Proc. of Crypto'82, pp. 279–288, Plenum Press, 1982.

[4] L. M. Adleman, On breaking the iterated Merkle-Hellman public-key cryptosystem, in: Proc. of Crypto'82, pp. 303–308, Plenum Press, 1982.

[5] J. C. Lagarias and A. M. Odlyzko, Solving low density subset sum problems, J. Assoc. Comp. Mach., **32** (1985), 229–246.

[6] M. J. Coster, B. A. LaMacchia, A. M. Odlyzko and C. P. Schnorr, An improved low-density subset sum algorithm, in: Proc. of Eurocrypt'91, D. W. Davies ed., Lect. Notes Comput. Sci., Vol. 547, pp. 54–67, Springer-Verlag, Berlin, 1991.

[7] P. Q. Nguyen and J. Stern, Adapting density attacks to low-weight knapsacks, in: Proc. of Asiacrypt 2005, B. Roy ed., Lect. Notes Comput. Sci., Vol. 3788, pp. 41–58, Springer-Verlag, Berlin, 2005.

[8] Y. Murakami and T. Nasako, Knapsack public-key cryptosystem using Chinese remainder theorem, in: Proc. of the 29th Symposium on Information Theory and Its Applications, pp. 207–210, 2006.

[9] T. Nasako and Y. Murakami, A high-density knapsack cryptosystem using combined trapdoor (in Japanese), Trans. JSIAM, **16** (2006), 591–605.

[10] Y. Murakami and T. Nasako, A new class of knapsack public-key cryptosystems using modular multiplication, in: Proc. of the 1st Joint Workshop on Information Security, pp. 351–354, 2006.