A Knapsack Cryptosystem Based on Multiple Knapsacks

Kunikatsu Kobayashi†, Kohtaro Tadaki†*, Masao Kasahara‡ and Shigeo Tsujii†
†Research and Development Initiative, Chuo University
1–13–27 Kasuga, Bunkyo-ku, Tokyo, 112–8551 Japan
Email: kobayash@yz.yamagata-u.ac.jp, tadaki@kc.chuo-u.ac.jp, tsujii@tamacc.chuo-u.ac.jp
‡Faculty of Informatics, Osaka Gakuin University
2–36–1 Kishibe-Minami, Suita-shi, Osaka, 564–8511 Japan
Email: kasahara@ogu.ac.jp

***Abstract*—In this paper, we propose a knapsack cryptosystem based on three knapsacks. Although one of the three secret knapsacks is superincreasing, the other two are non-superincreasing. On the encryption, a ciphertext is formed by multiplying the two non-superincreasing knapsacks together and then adding it to the superincreasing knapsack. Due to this structure of the cipher text, our knapsack cryptosystem is thought to be secure against all existing attacks, i.e., the low density attack and Shamir attack.**

I. Introduction

In 1978 Merkle and Hellman [4] first proposed knapsack cryptosystem, which is known as *MH knapsack cryptosystem* nowadays. In particular, the encryption and decryption of MH knapsack cryptosystem can be performed very fast. However, it is known that the secret key of MH knapsack cryptosystem can be disclosed by Shamir attack [5]. In addition, the plain text of MH knapsack cryptosystem can be disclosed by the low density attack [3] using LLL algorithm [2].

In this paper, we propose a knapsack cryptosystem based on three knapsacks. The secret sequence corresponding two knapsacks are non-superincreasing, while the other one is superincreasing. This knapsack cryptosystem uses nonlinear multiplicative operation in the encryption, and therefore is thought to be secure against the low density attack using LLL algorithm. On the other hand, the two secret sequence are non-superincreasing in this knapsack cryptosystem, and therefore it is thought to be secure against the attack using Shamir algorithm.

Note that the first, third, and fourth authors proposed a knapsack cryptosystem based on three knapsacks in the paper [1] in January, 2010. However, the second author pointed out that the decryption in the knapsack cryptosystem does not work properly; that is to say some plaintexts cannot be recovered in the decryption of the knapsack cryptosystem. He analyzed this phenomenon theoretically and then proposed a new version of knapsack cryptosystem based on three knapsacks, which will be reported in the present paper.

The paper is organized as follows. In Section II we describe our knapsack cryptosystem based on three knapsacks. We then illustrate this cryptosystem using a toy example in Section III. In Section IV, we consider the security of our cryptosystem. We conclude this paper in Section V.

* Corresponding author

II. Description of Our Cryptosystem

Our knapsack cryptosystem is described as follows.

A. Key-Generation

A sextuple (A, B, E, p, u, v) forms the secret key, and a triplet (F, G, H) forms public key. In the key-generation stage, these keys are generated as follows.

Step 1: Choose the secret keys $A = (a_1, a_2, \ldots, a_n)$, $B = (b_1, b_2, \ldots, b_n)$, and $E = (e_1, e_2, \ldots, e_n)$ corresponding to the three knapsacks, where a_i's and b_i's are positive integers and e_i's are non-negative integers. The secret sequences A, B, and E are chosen so as to satisfy Conditions 1 and 2 below.

Condition 1. *The secret sequences $A = (a_1, a_2, \ldots, a_n)$ and $B = (b_1, b_2, \ldots, b_n)$ are non-superincreasing, i.e.,*

$$a_k \leq \sum_{i=1}^{k-1} a_i \quad \& \quad b_k \leq \sum_{i=1}^{k-1} b_i \tag{1}$$

for every $k = 2, \ldots, n$. ■

Condition 2.

$$\begin{aligned} & a_k b_k - \left(\sum_{i=1}^{k-1} a_i \right) \left(\sum_{i=1}^{k-1} b_i \right) + e_k - \sum_{i=1}^{k-1} e_i \\ & + a_1 \left(2^{n-1} - 2^{k-1} \right) \left(a_k - \sum_{i=1}^{k-1} a_i \right) \\ & + b_1 \left(2^{n-1} - 2^{k-1} \right) \left(b_k - \sum_{i=1}^{k-1} b_i \right) > 0 \end{aligned} \tag{2}$$

for every $k = 2, \ldots, n$. ■

For each $k = 2, \ldots, n$, assume that the components $a_1, a_2, \ldots, a_{k-1}; b_1, b_2, \ldots, b_{k-1}; e_1, e_2, \ldots, e_{k-1}$ chosen so far. Then the next components a_k, b_k, and e_k are chosen easily so as to satisfy (1) and (2) simultaneously. This is because the

978–1–4244–6017–5/10/$26.00 ©

inequality (2) is equivalent to the inequality

$$\begin{aligned} e_k > & - a_k b_k + \left(\sum_{i=1}^{k-1} a_i\right)\left(\sum_{i=1}^{k-1} b_i\right) + \sum_{i=1}^{k-1} e_i \\ & - a_1\left(2^{n-1} - 2^{k-1}\right)\left(a_k - \sum_{i=1}^{k-1} a_i\right) \\ & - b_1\left(2^{n-1} - 2^{k-1}\right)\left(b_k - \sum_{i=1}^{k-1} b_i\right), \end{aligned} \tag{3}$$

and therefore one can choose e_k to satisfy (2) after choosing a_k and b_k to satisfy (1). More formally, the sequences $A = (a_1, a_2, \ldots, a_n)$, $B = (b_1, b_2, \ldots, b_n)$, and $E = (e_1, e_2, \ldots, e_n)$ satisfying Conditions 1 and 2 can be chosen using Algorithm 1 below.

Algorithm 1

Input: A positive integer n
Output: A triplet (A, B, E) of sequences of integers satisfying Conditions 1 and 2

1. Choose positive integers a_1 and b_1 and a non-negative integer e_1.
2. **for** $k := 2$ **to** n **do**
3. Choose positive integers a_k and b_k satisfying (1).
4. Choose a non-negative integer e_k satisfying (3).
5. Set $A := (a_1, a_2, \ldots, a_n)$, $B := (b_1, b_2, \ldots, b_n)$, and $E := (e_1, e_2, \ldots, e_n)$.

Note that the secret sequences A and B are not chosen to be superincreasing by Condition 1. On the other hand, since

$$a_k b_k \le \left(\sum_{i=1}^{k-1} a_i\right)\left(\sum_{i=1}^{k-1} b_i\right)$$

by Condition 1, it follows from Conditions 1 and 2 that the secret sequences E is superincreasing, i.e.,

$$e_k > \sum_{i=1}^{k-1} e_i$$

holds for every $k = 2, \ldots, n$.

Step 2: Choose positive integers p, u, and v so as to satisfy that

$$p > \left(\sum_{i=1}^{n} a_i\right)\left(\sum_{i=1}^{n} b_i\right) + \sum_{i=1}^{n} e_i \tag{4}$$

and $\gcd(u, p) = \gcd(v, p) = 1$.

Step 3: Based on the secret key (A, B, E, p, u, v), calculate the public keys $F = (f_1, f_2, \ldots, f_n)$, $G = (g_1, g_2, \ldots, g_n)$, and $H = (h_1, h_2, \ldots, h_n)$ by

$$\begin{aligned} f_i &\equiv u a_i \pmod{p}, \\ g_i &\equiv v b_i \pmod{p}, \\ h_i &\equiv uv e_i \pmod{p} \end{aligned}$$

for each $i = 1, \ldots, n$.

B. Encryption

A plaintext is represented as a vector $M = (m_1, m_2, \ldots, m_n)$ with $m_i \in \{0, 1\}$ $(1 \le i \le n)$, and the corresponding ciphertext is represented by an positive integer C. On the encryption, the ciphertext C is calculated by

$$C := C_1 C_2 + C_3,$$

where

$$\begin{aligned} C_1 &:= f_1 m_1 + f_2 m_2 + \cdots + f_n m_n, \\ C_2 &:= g_1 m_1 + g_2 m_2 + \cdots + g_n m_n, \\ C_3 &:= h_1 m_1 + h_2 m_2 + \cdots + h_n m_n. \end{aligned}$$

C. Decryption

The decryption of our cryptosystem is performed as follows.

Let u^{-1} and v^{-1} be the modular inverses of $u \pmod{p}$ and $u \pmod{p}$, respectively. Namely, let u^{-1} and v^{-1} be integers which satisfy that $uu^{-1} \equiv 1 \pmod{p}$ and $vv^{-1} \equiv 1 \pmod{p}$, respectively. By using the secret keys A, B, and E, the plaintext M is recovered as follows.

First, calculate $D(C) := u^{-1}v^{-1}C \pmod{p}$. Then

$$D(C) = \left(\sum_{i=1}^{n} a_i m_i\right)\left(\sum_{i=1}^{n} b_i m_i\right) + \sum_{i=1}^{n} e_i m_i. \tag{5}$$

The plaintext $M = (m_1, \ldots, m_n)$ is calculated based on Algorithm 2 below.

Algorithm 2

Input: a positive integer n, a non-negative integer $D(C)$, and a triplet (A, B, E) of sequences of integers satisfying Conditions 1 and 2
Output: A sequence M of 0 and 1 of length n

1. **for** $k := n$ **down to** 1 **do**
2. Set

$$\begin{aligned} d := & \left(\sum_{i=k+1}^{n} a_i m_i + a_k\right)\left(\sum_{i=k+1}^{n} b_i m_i + b_k\right) \\ & + \sum_{i=k+1}^{n} e_i m_i + e_k. \end{aligned}$$

3. **if** $D(C) \ge d$ **then** $m_k := 1$ **else** $m_k := 0$.
4. Set $M := (m_1, m_2, \ldots, m_n)$.

Note here that each of the sums $\sum_{i=k+1}^{n} a_i m_i$, $\sum_{i=k+1}^{n} b_i m_i$, and $\sum_{i=k+1}^{n} e_i m_i$ in Algorithm 2 is interpreted as 0 in the case of $k = n$. The correctness of Algorithm 2 is guaranteed by Theorem 1 (ii) below.

Theorem 1.

(i) *Algorithm 2 works properly in calculating every plaintext if and only if the following condition holds for the secret keys* $A = (a_1, a_2, \ldots, a_n)$, $B = (b_1, b_2, \ldots, b_n)$,

and $E = (e_1, e_2, \ldots, e_n)$: For every $k = 1, \ldots, n$ and every $(m_1, \ldots, m_{k-1}, m_{k+1}, \ldots, m_n) \in \{0,1\}^{n-1}$,

$$\begin{aligned} & a_k b_k - \left(\sum_{i=1}^{k-1} a_i m_i\right)\left(\sum_{i=1}^{k-1} b_i m_i\right) \\ & + e_k - \sum_{i=1}^{k-1} e_i m_i \\ & + \left(\sum_{i=k+1}^{n} b_i m_i\right)\left(a_k - \sum_{i=1}^{k-1} a_i m_i\right) \\ & + \left(\sum_{i=k+1}^{n} a_i m_i\right)\left(b_k - \sum_{i=1}^{k-1} b_i m_i\right) > 0. \end{aligned} \tag{6}$$

(ii) *Suppose that Condition 1 holds for the secret keys $A = (a_1, a_2, \ldots, a_n)$ and $B = (b_1, b_2, \ldots, b_n)$. Then Algorithm 2 works properly in calculating every plaintext if and only if the following condition holds for the secret keys A, B, and $E = (e_1, e_2, \ldots, e_n)$: For every $k = 2, \ldots, n$,*

$$\begin{aligned} & a_k b_k - \left(\sum_{i=1}^{k-1} a_i\right)\left(\sum_{i=1}^{k-1} b_i\right) \\ & + e_k - \sum_{i=1}^{k-1} e_i \\ & + \left(\sum_{i=k+1}^{n} b_i\right)\left(a_k - \sum_{i=1}^{k-1} a_i\right) \\ & + \left(\sum_{i=k+1}^{n} a_i\right)\left(b_k - \sum_{i=1}^{k-1} b_i\right) > 0. \end{aligned} \tag{7}$$

(iii) *Algorithm 2 works properly in calculating every plaintext if Conditions 1 and 2 hold for the secret keys $A = (a_1, a_2, \ldots, a_n)$, $B = (b_1, b_2, \ldots, b_n)$, and $E = (e_1, e_2, \ldots, e_n)$.*

Proof: (i) First, it is easy to see that Algorithm 2 works properly in calculating every plaintext if and only if the following two conditions [A] and [B] hold:

[A] For every $(m_1, m_2, \ldots, m_n) \in \{0,1\}^n$ and every $k = 1, \ldots, n$, if $m_k = 1$ then

$$D(C) \geq \left(\sum_{i=k+1}^{n} a_i m_i + a_k\right)\left(\sum_{i=k+1}^{n} b_i m_i + b_k\right) + \sum_{i=k+1}^{n} e_i m_i + e_k.$$

[B] For every $(m_1, m_2, \ldots, m_n) \in \{0,1\}^n$ and every $k = 1, \ldots, n$, if $m_k = 0$ then

$$D(C) < \left(\sum_{i=k+1}^{n} a_i m_i + a_k\right)\left(\sum_{i=k+1}^{n} b_i m_i + b_k\right) + \sum_{i=k+1}^{n} e_i m_i + e_k.$$

Here $D(C)$ is defined by (5). Note that the condition [A] always holds, no matter how the secret keys $A = (a_1, a_2, \ldots, a_n)$, $B = (b_1, b_2, \ldots, b_n)$, and $E = (e_1, e_2, \ldots, e_n)$ are chosen. Thus, we see that Algorithm 2 works properly in calculating every plaintext if and only if for every $k = 1, \ldots, n$ and every $(m_1, \ldots, m_{k-1}, m_{k+1}, \ldots, m_n) \in \{0,1\}^{n-1}$,

$$\begin{aligned} & \left(\sum_{i=k+1}^{n} a_i m_i + \sum_{i=1}^{k-1} a_i m_i\right)\left(\sum_{i=k+1}^{n} b_i m_i + \sum_{i=1}^{k-1} b_i m_i\right) \\ & + \left(\sum_{i=k+1}^{n} e_i m_i + \sum_{i=1}^{k-1} e_i m_i\right) \\ & < \left(\sum_{i=k+1}^{n} a_i m_i + a_k\right)\left(\sum_{i=k+1}^{n} b_i m_i + b_k\right) \\ & + \left(\sum_{i=k+1}^{n} e_i m_i + e_k\right). \end{aligned}$$

Since the above condition is equivalent to the condition (6), Theorem 1 (i) follows.

(ii) Assume that Condition 1 holds for the secret keys A and B. Then it is easy to see that the condition (7) is equivalent to the condition (6). Thus, Theorem 1 (ii) follows from Theorem 1 (i).

(iii) Assume that Conditions 1 and 2 hold for the secret keys A, B, and E. Then it follows from Condition 1 that

$$\begin{aligned} a_k &\leq 2^{k-1} a_1, \\ b_k &\leq 2^{k-1} b_1 \end{aligned}$$

for every $k = 2, \ldots, n$. Therefore, we have

$$\begin{aligned} \sum_{i=k+1}^{n} a_i &\leq a_1 \left(2^{n-1} - 2^{k-1}\right), \\ \sum_{i=k+1}^{n} b_i &\leq b_1 \left(2^{n-1} - 2^{k-1}\right) \end{aligned}$$

for every $k = 1, \ldots, n-1$. Thus, by Conditions 1 and 2, we see that the inequality (7) holds. It follows from Theorem 1 (ii) that Algorithm 2 works properly in calculating every plaintext. ■

Remark 1. In our knapsack cryptosystem, the three secret sequences A, B, and E are used, where A and B are non-superincreasing and E is superincreasing. Theorem 1 (ii) implies that we cannot construct a knapsack cryptosystem only using two non-superincreasing secret sequences A and B. Namely, it follows from Theorem 1 (ii) that if $E = (0, 0, \ldots, 0)$ and Condition 1 holds for the secret sequence A and B, then Algorithm 2 does not work properly in calculating all plaintexts. ■

III. Toy Example

In this section, using a toy example, we illustrate our knapsack cryptosystem proposed in the previous section.

We consider a toy example with $n = 4$. First, the secret keys are chosen as follows.

$$\begin{aligned} A &= (3, 3, 6, 12), \\ B &= (4, 3, 7, 14), \\ E &= (2, 6, 9, 19), \\ p &= 709, \\ u &= 642, \\ v &= 579. \end{aligned}$$

We can easily check that Conditions 1 and 2 hold for the secret keys A, B, and E, and moreover (4) and $\gcd(u, p) = \gcd(v, p) = 1$ hold. Thus, the corresponding public keys are calculated as follows.

$$\begin{aligned} F &= (508, 508, 307, 614), \\ G &= (189, 319, 508, 307), \\ H &= (404, 503, 400, 293). \end{aligned}$$

We consider the encryption and decryption of the plaintext $M = (0, 1, 0, 1)$.

On the encryption, the corresponding ciphertext C is calculated as

$$C = 703168,$$

since

$$\begin{aligned} C_1 &= 508 + 614, \\ C_2 &= 319 + 307, \\ C_3 &= 503 + 293. \end{aligned}$$

The decryption of $C = 703168$ is performed as follows. First, the modular inverses of $u \pmod{p}$ and $u \pmod{p}$ are calculated as

$$\begin{aligned} u^{-1} &= 582, \\ v^{-1} &= 649. \end{aligned}$$

Therefore

$$\begin{aligned} D(C) &\equiv 582 \times 649 \times 703168 \pmod{709} \\ &= 280. \end{aligned}$$

Then, the components m_4, m_3, m_2, m_1 of M are calculated in sequence as follows:

(i) Since $D(C) > a_4 b_4 + e_4 = 12 \times 14 + 19 = 187$, we have $m_4 = 1$.

(ii) Since $D(C) < (a_4 + a_3)(b_4 + b_3) + (e_4 + e_3) = 18 \times 21 + 28 = 406$, we have $m_3 = 0$.

(iii) Since $D(C) = (a_4 + a_2)(b_4 + b_2) + (e_4 + e_2) = 15 \times 17 + 25 = 280$, we have $m_2 = 1$.

(iv) Obviously, $D(C) < (a_4 + a_2 + a_1)(b_4 + b_2 + b_1) + (e_4 + e_2 + e_1)$ by the above. Thus, we have $m_1 = 0$.

Hence, the plaintext M is recovered properly as $M = (0, 1, 0, 1)$.

IV. Security Consideration

In this section, we consider the proposed knapsack cryptosystem. First, we consider the relation between the sequence used in knapsack cryptosystem and the cryptanalysis algorithm.

MH knapsack cryptosystem has one secret sequence $A = (a_1, a_2, \ldots, a_n)$, which is chosen to be superincresing, i.e., to satisfy that

$$a_k > a_1 + a_2 + \cdots + a_{k-1}$$

for every $k = 2, \ldots, n$. This type of knapsack cryptosystem is known to be insecure both against the low density attack using LLL algorithm [3] and against the attack using Shamir algorithm [5].

In comparison, the proposed knapsack cryptosystem has the three secret sequences $A = (a_1, a_2, \ldots, a_n)$, $B = (b_1, b_2, \ldots, b_n)$, and $E = (e_1, e_2, \ldots, e_n)$, where A and B are chosen to be non-superincreasing, while E are chosen to be superincreasing. Since A and B are non-superincreasing, the proposed knapsack cryptosystem is thought to be secure both against the low density attack using LLL algorithm and against the attack using Shamir algorithm, due to the reason explained below.

First, we consider the security of the proposed knapsack cryptosystem against the low density attack using LLL algorithm. For the purpose of being secure against the exhaustive attack, the size n of a plaintext has to be at least 100 bits. In general, the two knapsacks F and G of size n is thought to be equivalent to one knapsack of size n^2, with respect to the security against the low density attack. Therefore, in the case of $n = 100$, the security of the proposed knapsack cryptosystem is thought to be equivalent to one of MH knapsack cryptosystem of size 10000. Thus, in this case, the size of the matrix used by LLL algorithm in the low density attack is 10000×10000. However, in the current level of technology, LLL algorithm can work effectively only for the matrix of at most size 500×500 [6].[1] Hence, the proposed knapsack cryptosystem with plaintext of size at least 100 bits is thought to be secure against the low density attack using LLL algorithm.

Next, we consider the security of the proposed knapsack cryptosystem against the attack using Shamir algorithm. Shamir algorithm against MH knapsack cryptosystem can recover the secret sequence $A = (a_1, a_2, \ldots, a_n)$, and u' and p' which are equivalent to the original u and p, respectively, based on the fact that the superincreasing property holds for arbitrary four components in the public key $F = (f_1, f_2, \ldots, f_n)$. In comparison, each of the two secret sequences A and B in the proposed knapsack cryptosystem does not have the superincreasing property. Thus, Shamir algorithm can not apply to the public key $F = (f_1, f_2, \ldots, f_n)$

[1] When the paper [6] was published, LLL algorithm can work effectively only for the matrix of at most size 200×200. At present, however, LLL algorithm can work somehow for the matrix of size 500×500, due to the development of computer technology.

and $G = (g_1, g_2, \ldots, g_n)$ of the proposed knapsack cryptosystem to recover the secret sequences A and B, and u', v', and p' which are equivalent to the original u, v, and p, respectively. On the other hand, the secret sequence E has the superincreasing property in the proposed knapsack cryptosystem. Thus, the application of Shamir algorithm to the public key $H = (h_1, h_2, \ldots, h_n)$ may reveal w and p' which are equivalent to the original uv and p, respectively. However, the calculation of u and v from $uv \pmod{p}$ is thought to be more difficult than the prime factorization problem. Thus, even if the secret sequence E can be calculated by Shamir algorithm, it is still difficult to calculate the secret sequences A and B by applying Shamir algorithm to the public keys F and G in the proposed knapsack cryptosystem. Hence, the proposed knapsack cryptosystem is thought to be secure against the attack using Shamir algorithm.

As a result, the proposed knapsack cryptosystem is thought to be secure against all existing attacks at present.

V. Conclusion

We have proposed a knapsack cryptosystem based on three knapsacks. This knapsack cryptosystem is thought to be secure both against the low density attack using LLL algorithm and against the attack for computing secret keys from public keys using Shamir algorithm. The key size of the proposed knapsack cryptosystem is large in comparison with a usual knapsack cryptosystem based on one knapsack. However, the enhancement of the security compensates for this flaw in the proposed knapsack cryptosystem.

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. The second author was also supported by CREST (Core Research for Evolutional Science and Technology) from Japan Science and Technology Agency. We are grateful to the members of our SCOPE team, especially, to Professor Yasuyuki Murakami and Professor Ryuichi Sakai for their security consideration of our knapsack cryptosystem.

References

[1] K. Kobayashi, M. Kasahara, and S. Tsujii, "A knapsack cryptosystem using plural knapsacks," Proceedings of the 2010 Symposium on Cryptography and Information Security (SCIS2010), 2A1-1, January 19-22, 2010, Takamatsu, Japan. In Japanese.

[2] A. K. Lenstra, H. W. Lenstra, and L. Lovasz, "Factoring polynomials with rational coefficients," *Mathematische Annalen*, Vol.261, No.4 (1982), 515–534.

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

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

[5] A. Shamir, "A polynomial-time algorithm for breaking the basic Merkle-Hellman cryptosystem," Proc. CRYPTO '82, Lecture Notes in Computer Science, pp.279–288, Springer, 1982.

[6] H. Shimizu, "Lattice basis reduction and its application," Presented at the 6th JANT meeting, The Japan Society for Industrial and Applied Mathematics, January 26, 2002, Ochanomizu University, Tokyo, Japan. In Japanese. Available at URL: http://tnt.math.se.tmu.ac.jp/jant/006-shimizu.pdf