A New Knapsack Public-Key Cryptosystem Based on Permutation Combination Algorithm *

Min-Shiang Hwang[†] Cheng-Chi Lee[‡] Shiang-Feng Tzeng[§]

Department of Management Information System[†]
National Chung Hsing University
250 Kuo Kuang Road,
402 Taichung, Taiwan, R.O.C.
Email: mshwang@nchu.edu.tw

Department of Information & Communication Engineering[‡]
Asia University
No. 500, Lioufeng Raod,
Wufeng Shiang, Taichung, Taiwan, R.O.C.
Email: cclee@asia.edu.tw

Department of Information Management[§]
Chaoyang University of Technology
168 Gifeng E. Rd., Wufeng,
Taichung County, Taiwan 413, R.O.C.

July 1, 2008

A New Knapsack Public-Key Cryptosystem Based on Permutation Combination Algorithm

Abstract

A new secure knapsack cryptosystem based on the Merkle-Hellman public key cryptosystem will be proposed in this paper. Although it is common sense that when the density is low, the knapsack cryptosystem turns vulnerable to the low-density attack. The density d of a secure knapsack cryptosystem must be larger than 0.9408 to avoid low-density attack. In this paper, we investigate a new Permutation Combination Algorithm. By exploiting this algorithm, we shall propose a novel knapsack public-key cryptosystem. Our proposed scheme can enjoy a high density to avoid the low-density attack. The density d can also exceed 0.9408 to avoid the low-density attack.

Keywords: Public key, Knapsack problem, Knapsack cryptosystem, low-density attack.

1 Introduction

In 1976, Diffie and Hellman [6] introduced the public key cryptosystem. In their cryptosystem, two different keys are used: one for encryption and the other for decryption. Each user makes the encryption key public and keeps her/his decryption key secret. Everyone can use the user's encryption key to send encrypted message

*This work was supported in part by Taiwan Information Security Center (TWISC), National Science Council under the grant NSC 96-2219-E-009-013.

‡Responsible for correspondence: Dr. Cheng-Chi Lee.

to her/him. Then, the user can use his/her decryption key to recover the message. Most public key cryptosystems fall into one of the two categories below: [3]:

- Public key cryptosystems based on hard number-theoretic problems: e.g., RSA [2, 9, 10, 16], Rabin [15], and ElGamal [7, 8, 19] cryptosystems.
- Public key cryptosystems related to the knapsack problem: e.g., Merkle-Hellman [13] and Shamir [17] cryptosystems.

Unlike hard number-theoretic problems, the knapsack problem has been proven to be NP-complete [14]. That is to say, there is no polynomial algorithm will be invented to solve the knapsack problem. Breaking hard number-theoretic problems within a reasonable amount of time can be found. Hence, a knapsack cryptosystem is better than those based on hard number-theoretic problems. The knapsack problem is a problem of solving the liner diophantine equation. Let both C and a set $A = \{a_1, a_2, \cdots, a_n\}$ be given integers and $\{x_1, x_2, \cdots, x_n\}$ be unknown variables. The liner diophantine equation is as follows $C = a_1x_1 + a_2x_2 + \cdots + a_nx_n$. To solve the diophantine equation is to find integer-valued solutions of $\{x_1, x_2, \cdots, x_n\}$.

A lot of knapsack-type public key cryptosystems have been suggested to be insecure since when the density is low. It becomes weak against the low-density attack [1, 4, 5, 12]. To make sure of the security against the low-density attack, the density must be kept at a high level. In [4], they suggested that the density d of the knapsack vector must be larger than 0.9408 to avoid low-density attack. Recently, several knapsack cryptosystems with modified low-density attacks [11, 18, 20, 21, 22] have been proposed. They tried to take advantage of the exceedingly high speed encryption as well as deciphering operations of the Merkle-Hellman public key cryptosystem. In [20], Su et al. proposed a new knapsack public-key cryptosystem based on elliptic curve discrete logarithm. By appropriately choosing the parameters, one can control the ratio between the number of elements in the elliptic curve cryptography and their size in bits. It has a great effect on completely disguising the vulnerable knapsacks. In [22], Wang et al. proposed a novel probabilistic knapsack-based cryptosystem based on a new easy compact knapsack problem. The proposed scheme enjoyed a high knapsack density and it is secure against low-density attacks. In this paper, we investigate a new Permutation Combination Algorithm. By exploiting this algorithm, we shall propose a novel knapsack public-key cryptosystem. Our proposed scheme can enjoy a high density to avoid the low-density attack. The density d can also exceed 0.9408 to avoid the low-density attack.

The rest of this paper is organized as follows. We shall propose an algorithm built upon the knapsack problem in the following section. Then, in Section 3, we shall propose our new secure knapsack cryptosystem. The security analysis of the proposed cryptosystem will be discussed in Section 4. Finally, the concluding remarks will be in the last section.

2 Permutation Combination Algorithm

In this section, a new algorithm called the Permutation Combination Algorithm is to be proposed. Given a defined original sequence, the Permutation Combination Algorithm can generate a series of permutations developed from the original sequence. In our new cryptosystem to be proposed in Section 3 later, we will use this algorithm. The algorithm is as follows:

1. Define an original sequence $D_0 = \{E_n, E_{n-1}, \cdots, E_2, E_1\}$.

2. Re-combine all the elements of the original sequence D_0 which obtain $(n! - 1)$

sequences $D_1, D_2, \cdots, D_{n!-1}$.

$$n! = n \times (n-1) \times (n-2) \times \cdots \times 2 \times 1.$$

The sequences D_i $(i = 1, 2, \cdots, n!-1)$ are then defined as follows:

$$\begin{aligned}
D_0 &= \{E_n, E_{n-1}, E_{n-2}, \cdots, E_5, E_4, E_3, E_2, E_1\} \\
D_1 &= \{E_n, E_{n-1}, E_{n-2}, \cdots, E_5, E_4, E_3, E_1, E_2\} \\
D_2 &= \{E_n, E_{n-1}, E_{n-2}, \cdots, E_5, E_4, E_2, E_3, E_1\} \\
D_3 &= \{E_n, E_{n-1}, E_{n-2}, \cdots, E_5, E_4, E_2, E_1, E_3\} \\
D_4 &= \{E_n, E_{n-1}, E_{n-2}, \cdots, E_5, E_4, E_1, E_3, E_2\} \\
D_5 &= \{E_n, E_{n-1}, E_{n-2}, \cdots, E_5, E_4, E_1, E_2, E_3\} \\
D_6 &= \{E_n, E_{n-1}, E_{n-2}, \cdots, E_5, E_3, E_4, E_2, E_1\} \\
&\vdots \\
D_{n!-1} &= \{E_1, E_2, E_3, E_4, \cdots, E_{n-2}, E_{n-1}, E_n\}
\end{aligned}$$

3. According to the above sequences, each sequence owns a corresponding value called the factorial carry value $\{F_n, F_{n-1}, F_{n-2}, \cdots, F_5, F_4, F_3, F_2, F_1\}$. Using the factorial carry value, we can efficiently obtain any sequence. The factorial carry value is defined as follows:

$$\begin{array}{ccccccccc}
\{F_n, & F_{n-1}, & F_{n-2}, & \cdots & F_5, & F_4, & F_3, & F_2, & F_1\} \\
\uparrow & \uparrow & \uparrow & & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow \\
(n-1)! & (n-2)! & (n-3)! & \cdots & 4! & 3! & 2! & 1! & 0
\end{array}$$

For instance, suppose we want to get the sequence D_6. We can compute the factorial carry value $\{F_n, F_{n-1}, F_{n-2}, \cdots, F_5, F_4, F_3, F_2, F_1\}$ of D_6 as

$$6 = 0 \times (n-1)! + 0 \times (n-2)! + 0 \times (n-3)! + \\ \cdots + 0 \times 4! + 1 \times 3! + 0 \times 2! + 0 \times 1! + 0$$

So, the factorial carry value of D_6 is $\{0,0,0,\cdots,0,1,0,0,0\}$.

4. With the knowledge of the original sequence $D_0 = \{E_n, E_{n-1},\ E_{n-2},\ \cdots,\ E_5,\ E_4,\ E_3,\ E_2,\ E_1\}$ and the factorial carry value $\{0,0,0,\cdots,0,1,0,0,0\}$ of D_6, we can compute sequence D_6 as follows:

   Get E_n by introducing $F_n = 0$. Here, the remaining elements in the sequence are $\{E_{n-1}, E_{n-2}, \cdots, E_5, E_4, E_3, E_2, E_1\}$.

   Get E_{n-1} by introducing $F_{n-1} = 0$. Here, the remaining elements in the sequence are $\{E_{n-2}, \cdots, E_5, E_4, E_3, E_2, E_1\}$.

   Get E_{n-2} by introducing $F_{n-2} = 0$. Here, the remaining elements in the sequence are $\{E_{n-3}, \cdots, E_5, E_4, E_3, E_2, E_1\}$.

   $\vdots$

   Get E_5 by introducing $F_5 = 0$. Here, the remaining elements in the sequence are $\{E_4, E_3, E_2, E_1\}$.

   Get E_3 by introducing $F_4 = 1$. Here, the remaining elements in the sequence are $\{E_4, E_2, E_1\}$.

   Get E_4 by introducing $F_3 = 0$. Here, the remaining elements in the sequence are $\{E_2, E_1\}$.

   Get E_2 by introducing $F_2 = 0$. Here, the remaining element is $\{E_1\}$.

   Get E_1 by introducing $F_1 = 0$. Therefore, the sequence D_6 is $\{E_n,\ E_{n-1},\ E_{n-2},\ \cdots,\ E_5,\ E_3,\ E_4,\ E_2,\ E_1\}$.

In accordance with the above algorithm, let's take sequence D_{100} for example as follows:

1. Generate the original sequence D_0 as

$$D_0 = \{A, B, C, D, E, F\}.$$

2. Compute the factorial carry value of D_{100}.

$$100 = 0 \times 5! + 4 \times 4! + 0 \times 3! + 2 \times 2! + 0 \times 1! + 0$$

Then, the factorial carry value of D_{100} is $\{0, 4, 0, 2, 0, 0\}$.

3. Compute sequence D_{100} with its factorial carry value $\{0, 4, 0, 2, 0, 0\}$.

$$\begin{array}{lll} 0 \quad --- & \{A, B, C, D, E, F\} & \longrightarrow A \\ 4 \quad --- & \{B, C, D, E, F\} & \longrightarrow F \\ 0 \quad --- & \{B, C, D, E\} & \longrightarrow B \\ 2 \quad --- & \{C, D, E\} & \longrightarrow E \\ 0 \quad --- & \{C, D\} & \longrightarrow C \\ 0 \quad --- & \{D\} & \longrightarrow D \end{array}$$

The permutation of sequence D_{100} is $\{A, F, B, E, C, D\}$.

3 The Proposed Knapsack Cryptosystem

In this section, a new secure knapsack cryptosystem based on the Merkle-Hellman public-key cryptosystem will be proposed. A message M will be encrypted the ciphertext and signed the digital signature. Then, the sender sends the ciphertext and the digital signature to the verifier. The verifier can also represent to decrypt the ciphertext and authenticate the signature. The procedure of the proposed cryptosystem contains four phases: the encryption phase, the decryption phase, the signature generation phase, and the signature verification phase. In the initial stage, each user chooses a super increasing sequence $B = \{b_1, b_2, \cdots, b_{1360}\}$ as secret key vector.

$$b_i > \sum_{j=1}^{i-1} b_j \quad (i = 1, 2, \cdots, 1360).$$

W and W' are secret modular multipliers, relatively prime to P.

$$\begin{aligned} P &> \sum_{i=1}^{1360} b_i \\ gcd(W,P) &= 1 \\ W \times W' &\equiv 1 \bmod P. \end{aligned}$$

Each user transfers super increasing sequence $B = \{b_1, b_2, \cdots, b_{1360}\}$ into a pseudo-random sequence $A = \{a_1, a_2, \cdots, a_{1360}\}$ as follows:

$$a_i = b_i \times W \bmod P \quad (i = 1, 2, \cdots, 1360).$$

Further, each user chooses a random 170×256 binary matrix H, a vector $R = (r_1, r_2, \cdots, r_{256})^T$ and a vector $HR = (hr_1, hr_2, \cdots, hr_{170})^T$ to satisfy the following equation:

$$H \cdot R = HR \bmod n$$

$$= \begin{pmatrix} h_{1,1} & \cdots & h_{1,256} \\ \vdots & \ddots & \vdots \\ h_{170,1} & \cdots & h_{170,256} \end{pmatrix} \cdot \begin{pmatrix} r_1 \\ \vdots \\ r_{256} \end{pmatrix} = \begin{pmatrix} hr_1 \\ hr_2 \\ \vdots \\ hr_{170} \end{pmatrix} = \begin{pmatrix} 2^0 \\ 2^1 \\ \vdots \\ 2^{169} \end{pmatrix} \bmod n$$

$$hr_i = 2^{i-1} = \sum_{j=1}^{256} h_{i,j} r_j \bmod n \quad (i = 1, 2, \cdots, 170)$$

Let $H(\cdot)$ be a one-way hash function. The proposed scheme involves two parties: the sender and the verifier. Let A be a sender and B be a verifier. Then, A and B's secret and public parameters are listed in Table 1.

3.1 Encryption Phase

The sender A executes the following steps to generate the ciphertext C of the message M.

Table 1: key table

A		B	
Public Key	Secret Key	Public Key	Secret Key
A_a, R_a	B_a, M_a, W_a, W_a', H_a	A_b, R_b	B_b, M_b, W_b, W_b', H_b

1. Compute the digest D of M as

$$D = H_{1024}(M).$$

   Let D denotes a 1024-bit message.

2. Generate D' from D. The method is as follows:

   (a) D has 1024 bits, which means there can be approximately 1.7×10^{308} variations of D.

   (b) Compute 170!, which approximates 10^{306}.

   (c) D' is derived as follows:

$$D' = D \bmod 170!.$$

   Here, D' is smaller than the integer 170!.

3. Compute the factorial carry value $U = \{u_1, u_2, \cdots, u_{170}\}$ of D'.

$$D' = u_1 \times 169! + u_2 \times 168! + \cdots + u_{169} \times 1! + u_{170} \times 0$$

4. Divide B's public key vector $A_b = \{a_{b1}, a_{b2}, \cdots, a_{b1360}\}$ into 8 subset public key vectors. Each subset public key vector has 170 elements.

$$A_b = \{(a_{b1}, a_{b2}, \cdots, a_{b170}),$$

$$(a_{b171}, a_{b172}, \cdots, a_{b340}),$$

$$(a_{b341}, a_{b342}, \cdots, a_{b510}),$$

$$(a_{b511}, a_{b512}, \cdots, a_{b680}),$$

$$(a_{b681}, a_{b682}, \cdots, a_{b850}),$$

$$(a_{b851}, a_{b852}, \cdots, a_{b1020}),$$

$$(a_{b1021}, a_{b1022}, \cdots, a_{b1190}),$$

$$(a_{b1191}, a_{b1192}, \cdots, a_{b1360})\}$$

5. Recombine each subset public key vector using $U = \{u_1, u_2, \cdots, u_{170}\}$ by means of the Permutation Combination Algorithm. A chooses each subset public key vector in the first 128 elements. Then, A will obtain 1024 elements $A_{bu} = \{au_{b1}, au_{b2}, \cdots, au_{b1024}\}$.

6. M is divided into $\{M_1, M_2, \cdots, M_j\}$. Each M_k is a 1024-bit message $(k = 1, 2, \cdots, j)$.

$$\begin{aligned} M_1 &= \{x_{1,1}, x_{1,2}, \cdots, x_{1,1024}\} \\ M_2 &= \{x_{2,1}, x_{2,2}, \cdots, x_{2,1024}\} \\ &\vdots \\ M_j &= \{x_{j,1}, x_{j,2}, \cdots, x_{j,1024}\} \end{aligned}$$

7. The corresponding ciphertext C_k is given as the product of A_{bu} and M_k $(k = 1, 2, \cdots, j)$.

$$\begin{aligned} C_1 &= \sum_{i=1}^{1024} au_{bi} \times x_{1,i} \\ C_2 &= \sum_{i=1}^{1024} au_{bi} \times x_{2,i} \\ &\vdots \\ C_j &= \sum_{i=1}^{1024} au_{bi} \times x_{j,i} \end{aligned}$$

Then, the set of the ciphertext $\{C_1, C_2, \cdots, C_j\}$ is a ciphertext C. A sends C and D' to B through the insecure channel.

3.2 Decryption Phase

After receiving C and D', B executes the following steps to derive M from C and D'.

1. Compute the factorial carry value $U = \{u_1, u_2, \cdots, u_{170}\}$ of D'.

$$D' = u_1 \times 169! + u_2 \times 168! + \cdots + u_{169} \times 1! + u_{170} \times 0$$

2. Divide B's secret key vector $B_b = \{b_{b1}, b_{b2}, \cdots, b_{b1360}\}$ into 8 subset public key vectors. Then, each subset secret key vector has 170 elements.

$$\begin{aligned} B_b = \{&(b_{b1}, b_{b2}, \cdots, b_{b170}),\\ &(b_{b171}, b_{b172}, \cdots, b_{b340}),\\ &(b_{b341}, b_{b342}, \cdots, b_{b510}),\\ &(b_{b511}, b_{b512}, \cdots, b_{b680}),\\ &(b_{b681}, b_{b682}, \cdots, b_{b850}),\\ &(b_{b851}, b_{b852}, \cdots, b_{b1020}),\\ &(b_{b1021}, b_{b1022}, \cdots, b_{b1190}),\\ &(b_{b1191}, b_{b1192}, \cdots, b_{b1360})\} \end{aligned}$$

3. Recombine each subset secret key vector using $U = \{u_1, u_2, \cdots, u_{170}\}$ by means of the Permutation Combination Algorithm. B chooses each subset secret key vector in the first 128 elements. Then, B will obtain 1024 elements $B_{bu} = \{bu_{b1}, bu_{b2}, \cdots, bu_{b1024}\}$. However, $B_{bu} = \{bu_{b1}, bu_{b2}, \cdots, bu_{b1024}\}$ is still a super increasing sequence. In other words, the elements of B_{bu} do not change order.

4. Divide C into $\{C_1, C_2, \cdots, C_j\}$. Each C_k is a 1024-bit ciphertext ($k = 1, 2, \cdots, j$).

5. Compute the recombine message Mre_k, which is given as the product of C_k and W' $(k = 1, 2, \cdots, j)$.

$$\begin{aligned} Mre_k &= C_k \times W' \bmod P \\ &= \sum_{i=1}^{1024} (au_{bi} \times x_{k,i}) \times W' \bmod P \\ &= \sum_{i=1}^{1024} (bu_{bi} \times W \times x_{k,i}) \times W' \bmod P \\ &= \sum_{i=1}^{1024} bu_{bi} \times x_{k,i} \bmod P \end{aligned}$$

6. The message $\{M_1, M_2, \cdots, M_j\}$ is then obtained through recombining each pre-recombining message $\{Mre_1, Mre_2, \cdots, Mre_j\}$ by using $U = \{u_1, u_2, \cdots, u_{170}\}$ through the Permutation Combination Algorithm.

Then, the set of $\{M_1, M_2, \cdots, M_j\}$ is the real message M.

3.3 Signature Generation Phase

Without loss of generality, A executes the following steps to sign a signature from M.

1. Divide binary matrix H_a into 256 subset vectors. Each subset vector has 170 elements.

$$\begin{aligned} H_a = [&(h_{1,1}, h_{2,1}, \cdots, h_{170,1})^T, \\ &(h_{1,2}, h_{2,2}, \cdots, h_{170,2})^T, \\ &\vdots \\ &(h_{1,256}, h_{2,256}, \cdots, h_{170,256})^T] \end{aligned}$$

2. Recombine the binary matrix H_a using $U = \{u_1, u_2, \cdots, u_{170}\}$ by means of the Permutation Combination Algorithm. A chooses each subset vector in the first

128 elements. Then, A will obtain 128×256-binary matrix H'_a.

$$H'_a = \begin{pmatrix} h'_{1,1} & \cdots & h'_{1,256} \\ \vdots & \ddots & \vdots \\ h'_{128,1} & \cdots & h'_{128,256} \end{pmatrix}$$

3. Compute the digest D_M of D as

$$D_M = H_{128}(D).$$

   Let D_M be a 128-bit message.

4. Generate inverse binary sequence $D'_M = (m_1, m_2, \cdots, m_{128})$ from D_M. I.e., if $D_M = 3 = (011)_2$ then $D'_M = (110)_2$.

5. Compute a signature $S = (s_1, s_2, \cdots, s_{256})$ as

$$S = D'_M H'_a = \begin{pmatrix} m_1, m_2, \cdots, m_{128} \end{pmatrix} \begin{pmatrix} h'_{1,1} & \cdots & h'_{1,256} \\ \vdots & \ddots & \vdots \\ h'_{128,1} & \cdots & h'_{128,256} \end{pmatrix}$$

$$s_j = \sum_{i=1}^{128} m_i h'_{i,j} \quad j = 1, 2, \cdots, 256$$

A sends the digital signature S to B through the insecure channel.

3.4 Signature Verification Phase

B executes the following steps to verify the validity of signature S.

1. Use A's public key R_a to compute the pre-recombine message Mre which is given as the product of S and R_a.

$$\begin{aligned} Mre &= S \cdot R_a \bmod n \\ &= \sum_{j=1}^{256} s_j \cdot r_j \bmod n \end{aligned}$$

$$
\begin{aligned}
&= \sum_{j=1}^{256}[\sum_{i=1}^{128} m_i \cdot h'_{i,j}]r_j \bmod n \\
&= \sum_{i=1}^{128} m_i[\sum_{j=1}^{256} h'_{i,j} \cdot r_j] \bmod n \\
&= \sum_{i=1}^{128} m_i \cdot hr'_i \bmod n
\end{aligned}
$$

2. Recombine the message Mre to obtain D_M using $U = \{u_1, u_2, \cdots, u_{170}\}$ by means of the Permutation Combination Algorithm. Then, B can check the validity of S of M through the following equation:

$$D_M \stackrel{?}{=} H_{128}(H_{1024}(M))$$

If the equation holds, the message M is authenticated and the digital signature S is valid.

4 Security Analysis

The security of the proposed cryptosystem is based on the cryptographic assumption of the intractability of the knapsack problem. Most knapsack cryptosystems can be easily broken by the famous low-density attack proposed by Lagarias and Odlyzko [12].

We now discuss the range of P. Suppose $Q = \{b_1, b_2, \cdots, b_n\} = \{2^0, 2^1, \cdots, 2^{n-1}\}$. We can obtain

$$
\begin{aligned}
\sum_{i=0}^{j-1} 2^i &= 2^j - 1, \\
2^j &> \sum_{i=0}^{j-1} 2^i.
\end{aligned}
$$

Q is a super increasing sequence and $\sum_{i=1}^{n} b_i = 2^n - 1$. In order to make P satisfy the proposed cryptosystem, P must be larger than 2^n. When $n = 1360$, $P \geq 2^{1360} \approx 2.5164 \times 10^{409}$.

The density of knapsack cryptosystem is defined as

$$d = \frac{n}{\log_2(max(b_i))} \quad (1 \leq i \leq n).$$

When the density d is smaller than 0.9408, the knapsack public key cryptosystem becomes vulnerable to the low-density attack [4]. So, the density d must be kept larger than 0.9408 to make the proposed cryptosystem stay secure. Let A_b be $\{a_{b1}, a_{b2}, \cdots, a_{b1360}\}$, and then the range of P is:

$$\begin{aligned}
0.9408 &\leq \frac{n}{\log_2(P-1)} \\
\log_2(P-1) &\leq \frac{1360}{0.9408} \\
\log_2(P-1) &\leq 1445.5782\cdots \\
P-1 &\leq 2^{1445.5782\cdots} \\
P &\leq 2^{1445.5782\cdots} + 1 \\
P &\leq 1.4534 \times 10^{435}
\end{aligned}$$

According to the above statement, the range of P of our cryptosystem is $(2.5164 \times 10^{409}, 1.4534 \times 10^{435})$. Observing the definition of density, we come to two ways of increasing the density: increasing n and decreasing $\log_2(max(a_{bi}))$. According to the above discussion, we define the range of P to be $(2.5164 \times 10^{409}, 1.4534 \times 10^{435})$. In other words, the density of the proposed cryptosystem did exceed 0.9408 to avoid the low-density attack.

5 Conclusions

In this paper, a new knapsack cryptosystem based on permutation combination algorithm has been proposed. The proposed cryptosystem stays away from the low-density attack by keeping the density high.

References

[1] L. Adleman. "On breaking generalized knapsack public key cryptosystems,". Internal Rep. TR-83-207, Univ. Southern Calif., Los Angeles, Mar 1983.

[2] C. C. Chang and M. S. Hwang, "Parallel computation of the generating keys for RSA cryptosystems," *IEE Electronics Letters*, vol. 32, no. 15, pp. 1365–1366, 1996.

[3] BENNY CHOR and RONALD L. RIVEST, "A knapsack-type public key cryptosystem based on arithmetic in finite fields," *IEEE Trans. Inform. Theory*, vol. 34, pp. 901–909, September 1988.

[4] M. J. Coster, B. A. LaMacchia, A. M. Odlyzko, and C. P. Schnorr, "An improved low-density subset sum algorithm," in *Advances in Cryptology, EUROCRYPT'91*, pp. 54–67, Lecture Notes in Computer Science, Vol. 547, 1991.

[5] Y. G. Desmedt, J. P. Vandewalle, and R. M. Govarets, "A critical analysis of the security of knapsack public-key algorithms," *IEEE Trans. Inform. Theory*, vol. IT-30, pp. 601–611, July 1984.

[6] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Trans. Inform. Theory*, vol. IT-22, pp. 644–654, Nov 1976.

[7] T. ElGamal, "A public-key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Transactions on Information Theory*, vol. IT-31, pp. 469–472, July 1985.

[8] M. S. Hwang, C. C. Chang, and K. F. Hwang, "An ElGamal-like cryptosystem for enciphering large messages," *IEEE Transactions on Knowledge and Data Engineering*, vol. 14, no. 2, 2002.

[9] M. S. Hwang, and C. C. Lee, "Research issues and challenges for multiple digital signatures," *International Journal of Network Security*, vol. 1, no. 1, pp. 1-7, 2005.

[10] M. S. Hwang, Eric J. L. Lu, and I. C. Lin, "A practical (t, n) threshold proxy signature scheme based on the RSA cryptosystem," *IEEE Transactions on Knowledge and Data Engineering*, vol. 15, no. 5, pp. 1–9, 2003.

[11] Kiyoko Katayanagi, Yasuyuki Murakami, and Masao Kasahara, "A new product-sum public-key cryptosystem using message extension," *IEICE Transaction on Fundamentals of Electronics, Communications and Computer Sciences, Special Section on Cryptography and Information Security*, vol. E84-A, pp. 2482–2487, October 2001.

[12] J. C. Lagarias and A. M. Odlyzko, "Solving low-density subset sum problems," in *Proceedings of 24rd Annu. Symp. Foundations of comput. Sci.*, pp. 1–10, 1983.

[13] R. Merkle and M. E. Hellman, "Hiding information and signatures in trapdoor knapsack," *IEEE Trans. Inform. Theory*, vol. IT-24, pp. 525–530, Sept 1978.

[14] R. Michael, and S. David, "Computers and Intractability: A guide to the theory of NP-completeness," W. H. Freeman & Co., San Francisco, 1979.

[15] M.O. Rabin, "Digitalized signatures and public-key functions as intractable as factorization," *Technical Report, MIT/LCS/TR212, MIT Lab., Computer Science Cambridge, MA, USA*, January 1979.

[16] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public key cryptosystems," *Communications of the ACM*, vol. 21, pp. 120–126, Feb. 1978.

[17] A. Shamir. "A fast signature scheme,". Laboratory for Computer Science Report RM-107, MIT, Cambridge, MA, july 1978.

[18] H. Shimizu. "On the security of kasahara-murakami public key cryptosystem,". tech. rep., IEICE Technical Report., Los Angeles, Nov. 1999.

[19] M. Sramka, "Cryptanalysis of the Cryptosystem Based on DLP $r = \alpha^a \beta^b$," *International Journal of Network Security*, vol. 6, no. 1, pp. 80-81, 2008.

[20] P. C. Su, E. H. Lu, and K. C. Henry, "A knapsack public-key cryptosystem based on elliptic curve discrete logarithm," *Applied Mathematics and Computation*, vol. 168, no. 1, pp. 40-46, 2005.

[21] Daisuke Suzuki, Yasuyuki Murakami, Ryuichi Sakai, and Masao Kasahara, "A new product-sum type public key cryptosystem based on reduced bases," *IEICE Transaction on Fundamentals of Electronics, Communications and Computer Sciences, Special Section on Cryptography and Information Security*, vol. E84-A, pp. 326–330, January 2001.

[22] B. Wang, Q. Wu, and Y. Hu, "A knapsack-based probabilistic encryption scheme," *Information Sciences*, vol. 177, no. 19, pp. 3981-3994, 2007.