Cryptanalysis of two knapsack public-key cryptosystems

Jingguo Bi[1], Xianmeng Meng [2], and Lidong Han[1]
{jguobi,hanlidong}@sdu.edu.cn
mengxm@sdfi.edu.cn

[1] Key Laboratory of Cryptologic Technology and Information Security, Ministry of Education, Shandong University, 250100 Jinan, China
[2]Dept. of Mathematics and Statistics, Shandong University of Finance ,Jinan 250014, P.R. China.

Abstract. In this paper, we cryptanalyze two knapsack cryptosystems. The first one is proposed by Hwang et al [4], which is based on a new permutation algorithm named Permutation Combination Algorithm. We show that this permutation algorithm is useless to the security of the cryptosystem. Because of the special super increasing structure, we can break this cryptosystem use the method provided by Shamir at Crypto'82. The second one is provided by Su et al [16], which is based on the elliptic curve discrete logarithm and knapsack problem. We show that one can recover the plaintext as long as he solve a knapsack problem.Unfounately, this knapsack problem can be solved by Shamir's method or the low density attack. Finally, we give a improved version of Su's cryptosystem to avoid these attacks.
Keywords:Knapsack;public-key cryptosystem;Cryptanalysis.

1 Introduction

The knapsack (or subset sum) problem is a well-known NP-complete problem. This problem is stated as follows: given positive integers $a_1, a_2, \ldots, a_n$ and s, whether there is a subset of the a_i that sums to s. That is equivalent to determine whether there are variables $m_1, \ldots, m_n$ such that

$$s = \sum_{i=1}^{n} m_i a_i, m_i \in \{0,1\}, 1 \leq i \leq n \qquad (1.1)$$

The density of the knapsack is defined as $d = n/\log_2 A$, where $A = \max_{1 \leq i \leq n} a_i$.

The hardness of knapsack problem inspired many public-key cryptosystem in the eighties, following the seminal work of Merkle and Hellman[9]. Although the underlying problem is NP-complete, it has surprisingly been broken by Shamir[15] because of the special structure of the private key. Later, many variants have been shown insecure for any practical parameters by lattice reduction techniques (see the survey [11][5][10]), such as Lagarias-Odlyzko reduction [7] pointed out that the general knapsack can be solved under the existence of

a SVP-oracle if the density $d < 0.646$, which was generalized by J.Coster et al.[2] to 0.9408. Actually, we can use the celebrated LLL algorithm[8] or other lattice reduction [13][14] instead of the SVP-Oracle in practical applications. However, all proposed knapsack schemes have been broken except Okamoto-Tanaka-Uchiyama(OTU)quantum knapsack cryptosystem from Crypto '00[12]. Either by special structure such as Merkle-Hellman cryptosystem[9] and Chor-Rivest cryptosystem[1][18], or attacked by low-density attack[7][2]. To make sure of the resistance of low-density attack, it is suggested that the density d must be lager than 0.9408 [2]. Recently, several knapsack cryptosystems with high density have been proposed [4][16][17][19].

In this paper, we analyze two knapsack cryptosystems [4][16]. In [4], Hwang et al.investigated a new permutation algorithm named Permutation Combination Algorithm, by exploiting this algorithm to avoid the low-density attack. We show that the permutation algorithm is useless to avoid the low-density attack and the density of knapsack vector is smaller than 0.9408. However, if we use the low-density method[7][2], the dimension of the lattice is 1025, which is too big to use LLL algorithm[8]. Because of the special super increasing structure, we can obtain equivalent private keys using Shamir's method. In [16], Su et al.presented the knapsack cryptosystem based on a application of the elliptic curve discrete logarithm problem. We show that one can recover the plaintext as long as he can solve a knapsack cryptosystem almost the same as Merkle-Hellman cryptosystem.

The rest of the paper is organized as follows. In Section 2, we review Shamir's method[15] of attacking the original Merkle-Hellman cryptosystem[9]. The low-density attack[7][2]for the general knapsack problem will be introduced in Section 3. In Section 4, we give the description of Hwang's algorithm and analyze it. In Section 5, we propose the description of Su's algorithm and provide our attack. Finally, we conclude this paper in Section 6.

2 Shamir's method

At Crypto'82, Adi Shamir [15] gave the first attack on the original knapsack cryptosystem. In this section, we review Shamir's attack on the basic Merkle-Hellman knapsack cryptosystem. Firstly, we give a brief description of the original Merkle-Hellman knapsack cryptosystem.

The sender chooses a super increasing sequence

$$B = (b_1, b_2, \ldots, b_n), \text{i.e. } b_j > \sum_{i=1}^{j-1} b_i, 2 \le j \le n$$

and two positive integers W and P, with $P > \sum_{i=1}^{n} b_i$, $(W, P) = 1$, computes

$$a_i^{'} \equiv b_i W \pmod{P}, 0 < a_i^{'} < P \qquad (2.1)$$

Then selects a permutation π of $\{1, 2, \ldots, n\}$ and defines

$$a_i = a_{\pi(i)}^{'} \quad 1 \le i \le n$$

The public key is the sequence of n positive integers $a_1, a_2, \ldots, a_n$ and the private key is the super increasing sequence B, the integers W, P and the permutation π.Typically, the size of each b_i is $n+i$ bits, for $1 \leq i \leq n$, the size of P is $2n+1$ bits. In the original Merkle-Hellman cryptosystem $n = 100$.
A message $m = (x_1, \ldots, x_n)$ is encrypted as

$$c = \sum_{i=1}^{n} x_i a_i$$

and the receiver computes

$$\begin{aligned} m &\equiv cW^{-1} \pmod{P} \\ &\equiv \sum_{i=1}^{n} x_i a_i W^{-1} \pmod{P} \\ &\equiv \sum_{i=1}^{n} x_i a'_{\pi(i)} W^{-1} \pmod{P} \\ &\equiv \sum_{i=1}^{n} x_i b_{\pi(i)} \pmod{P}. \end{aligned}$$

Since $P > \sum_{i=1}^{n} b_i$, we can obtain that

$$c = \sum_{i=1}^{n} x_i b_{\pi(i)}$$

The equation is easy to solve since the b_i form a super increasing sequence.

Let $U = W^{-1} \pmod{P}, 0 < U < M$, from equation(2.1)

$$a_i \equiv b_{\pi(i)} W \pmod{P}$$

We have

$$b_{\pi(i)} \equiv a_i U \pmod{P}$$

This means that there exists some integer k_i such that

$$a_i U - k_i P = b_{\pi(i)}$$

Hence

$$\frac{U}{P} - \frac{k_i}{a_i} = \frac{b_{\pi(i)}}{a_i P}$$

The cryptanalyst does not know $U, P, \pi, k_i,$ and b_i, but a_i. However, he can find the size of $a_i, 1 \leq i \leq n$ and U are the same as P's and $b_i \leq 2^{n+i}$. The first five factors of the super sequence satisfy

$$b_i \leq 2^{n+i}, 1 \leq i \leq 5$$

let $i_j = \pi^{-1}(j)$, then we obtain

$$|\frac{U}{P} - \frac{k_{i_j}}{a_{i_j}}| \leq \frac{2^{n+5}}{2^{4n+2}}, \quad 1 \leq j \leq 5 \qquad (2.2)$$

and subtract the $j = 1$ term from the others, we have

$$\left|\frac{k_{i_j}}{a_{i_j}} - \frac{k_{i_1}}{a_{i_1}}\right| \leq 2^{-3n+4}, 2 \leq j \leq 5$$

This implies that

$$\left|k_{i_j}a_{i_1} - k_{i_1}a_{i_j}\right| \leq 2^{n+6}, 2 \leq j \leq 5 \qquad (2.3)$$

Inequalities (2.3) show how unusual the a_{i_j} and k_{i_j} are. After all, the size of each of them is $2n$ bits, so the size of $k_{i_j}a_{i_1}$ and $k_{i_1}a_{i_j}$is $4n$ bits. But the size of the difference of two such terms to be $n + 6$ bits, which requires some very special structure. Shamir's main contribution was to notice that k_{i_j} can be found in polynomial time by invoking H.W.Lenstra's theorem that the integer programming problem in a fixed number of variables can be solved in polynomial time[15]. However, the cryptanalyst should invoke $\binom{5}{n}$ times of H.W.Lenstra's integer programming algorithm because he doesn't know the permutation π. This yields the $k_{i_j}, 1 \leq j \leq 5$. Once the k_{i_j} are found, one obtains an approximation to U/P from the equation(2.2) and constructs a pair $(U^{'}, P^{'})$ with $U^{'}/P^{'}$ close to U/P such that the weights obtained by

$$c_i \equiv a_i U^{'} \pmod{P^{'}}, 0 < c_i < P^{'}, 1 \leq i \leq n$$

form a super increasing sequence. So one can find a equivalent secret key in polynomial time. The concrete description of Shamir's attack can be found in [6].

3 Low-density attack

In this section, we will introduce the low density attack proposed by J.Coster et al.[2] at Eurocrypt'91 which is the modification of Lagarias and Odlyzko's attack[7].

The knapsack problem is stated as follows: For given positive integers $a_1, \ldots, a_n$ and s, find variables $e_1, \ldots, e_n$, with $e_i \in \{0, 1\}$, such that

$$\sum_{i=1}^{n} e_i a_i = s$$

Define the vectors $\mathbf{b}_1, \ldots, \mathbf{b}_{n+1}$ as follow:

$$\mathbf{b}_1 = (1, 0, \ldots, 0, Na_1)$$
$$\mathbf{b}_2 = (0, 1, \ldots, 0, Na_2)$$
$$\vdots$$
$$\mathbf{b}_n = (0, 0, \ldots, 1, Na_n)$$
$$\mathbf{b}_{n+1} = (\frac{1}{2}, \frac{1}{2}, \ldots, \frac{1}{2}, Ns)$$

where $N > \frac{1}{2}\sqrt{n}$. Let L be the lattice spanned by the vectors $\mathbf{b}_1, \ldots, \mathbf{b}_{n+1}$. Notice that the vector

$$\mathbf{e} = (e_1^{'}, \ldots, e_n^{'}, 0) \in \mathbf{L} \text{ where } e_i^{'} = e_i - \frac{1}{2}$$

J.coster et al.showed that when the density $d < 0.9408\ldots$, the vector $\mathbf{e}$ is the shortest vector in L. So one can find $\mathbf{e}$ if there exists a SVP oracle. In reality, we usually use LLL algorithm [8] or other lattice bases reduction algorithm [13][14] instead of the SVP oracle. However, from [3], it is thought that when the dimension is big enough (such as bigger than 400), one can not find the shortest vector for random lattice.

4 Attack of Hwang et al.'s cryptosystem

4.1 The description of Hwang et al.'s cryptosystem

Hwang's cryptosystem is based on the Merkle-Hellman cryptosystem. In initial stage, each user choose a super increasing sequence $B = (b_1, \ldots, b_{1360})$ as secret key vector.i.e.

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

W and P are two positive integers, such that

$$(W, P) = 1, P > \sum_{i=1}^{1360} b_i$$

Compute

$$a_i = b_i W \pmod{P}, 1 \le i \le 1360$$

The public key is $A = \{a_1, \ldots, a_{1360}\}$ and the private key is W, P and the super increasing sequence $B = \{b_1, \ldots, b_{1360}\}$. Obviously, there is no difference between the cryptosystem above and the original Merkle-Hellman except the omission of the permutation in initial stage.

The author investigated a permutation combination algorithm and wanted this algorithm to ensure the security of the cryptosystem. The permutation algorithm is as follows: Define an original sequence $D_0 = \{E_1, \ldots, E_n\}$, and Recombine all the elements of the original sequence D_0 which obtain $(n! - 1)$ sequences $D_1, \ldots, D_{n!-1}$. Notice for any m, $1 \leq m \leq n! - 1$, m can be written as

$$m = \sum_{i=1}^{n} m_i(n-i)!, 0 \leq m_i \leq n - i$$

Input $D_0 = \{E_1, \ldots, E_n\}$ and integer m
Output $D_m = \{E_1^{'}, \ldots, E_n^{'}\}$

1. Write m as $m = \sum_{i=1}^{n} m_i(n-i)!$
2. for$(1 \leq i \leq n)$
   if $(m_i == 0)$
   $E_i^{'} = E_i$
   else
   { $E_i^{'} = E_{i+m_i}$
   for$(1 \leq j \leq m_i)$
   $E_{i+j}^{'} = E_i$ }
3. output $D_m = \{E_1^{'}, \ldots, E_n^{'}\}$.

For example, Generate the original sequence $D_0 = \{A, B, C, D, E, F\}$. Compute the value of D_{100},

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

Then $D_{100} = \{A, F, B, E, C, D\}$.
Encryption:Message M, choose a hash function whose digest is 1024 bits.

$$D = H_{1024}(M) \pmod{170!}$$

So D can be written as

$$D = \sum_{i=1}^{170} u_i((170-i)!)$$

And then divide the public key vector $A = \{a_1, \ldots, a_{1360}\}$ into 8 subset public key vectors. Each key vector has 170 elements

$$A = \{(a_1, \ldots, a_{170}), (a_{171}, \ldots, a_{340}), \ldots, (a_{1191}, \ldots, a_{1360})\}$$

Recombine each subset public key vector using $U = \{u_1, \ldots, u_{170}\}$ by means of the Permutation Combination Algorithm. Then chooses the first 128 elements in each subset public key vector, thus, the sender obtains 1024 elements $A^{'} = \{a_1^{'}, \ldots, a_{1024}^{'}\}$. Then divides M into $M_1, \ldots, M_j$, each $M_k, 1 \leq k \leq j$ is a 1024 bits message,

$$M_k = \{x_{k,1}, \ldots, x_{k,1024}\}$$

The corresponding ciphertext C_k is given as

$$C_k = \sum_{i=1}^{1024} x_{k,i} a_i^{'} \quad (4.1)$$

The ciphertext is $C = \{C_1, \ldots, C_j\}$.Send C and D to the receiver.
Decryption:After receiving D, D can be written as

$$D = \sum_{i=1}^{170} u_i((170 - i)!)$$

Divide the private key vector $B = \{b_1, \ldots, b_{1360}\}$ into 8 subset public key vectors.Each key vector has 170 elements.

$$B = \{(b_1, \ldots, b_{170}), (b_{171}, \ldots, b_{340}), \ldots, (b_{1191}, \ldots, b_{1360})\}$$

Recombine each subset private key vector using $U = \{u_1, \ldots, u_{170}\}$ by means of the Permutation Combination Algorithm.Chooses the first 128 elements in each subset public key vector, the receiver obtains 1024 elements

$$B^{'} = \{b_1^{'}, \ldots, b_{1024}^{'}\}$$

Divide C into $C_1, \ldots, C_j$. Each C_k,$1 \leq k \leq j$ is a 1024-bit ciphertext.

$$\begin{aligned} M_k &\equiv C_k W^{-1} \pmod{P} \\ &\equiv \sum_{i=1}^{1024} a_i^{'} \times x_{k,i} \times W^{-1} \pmod{P} \\ &\equiv \sum_{i=1}^{1024} (b_i^{'} \times W \times x_{k,i}) \times W^{-1} \pmod{P} \\ &= \sum_{i=1}^{1024} b_i^{'} x_{k,i}. \end{aligned}$$

So the receiver solve this super increasing knapsack problem and then obtain the message M.

4.2 Attack of Hwang et al.'s cryptosystem

The author suppose the super increasing sequence

$$B = \{b_1, \ldots, b_n\} = \{2^0, 2^1, \ldots, 2^{n-1}\}$$

and $n = 1360$,$P \geq 2^{1360} \approx 2.5164 \times 10^{409}$
From (4.1), We can obtain the density

$$d = \frac{1024}{\log_2 P} \approx \frac{1024}{1360} = 0.7529 < 0.9408$$

(In Hwang's paper, they compute the density $d = \frac{1360}{\log_2 max(b_i)}$. That is not correct.)

So this cryptosystem becomes vulnerable to the low-density attack[7][2].But the dimension of the lattice is 1025, as we say in Section 3, we can not use LLL algorithm [8]and other lattice basis reduction [13][14] to find the shortest vector in the lattice.

However, we can still use Shamir's method to break this cryptosystem. Because there is no permutation between the b_i and a_i. Let

$$U = W^{-1} \pmod{P}, 0 < U < M$$

From

$$a_i \equiv b_i W \pmod{P}$$

We have

$$b_i \equiv a_i U \pmod{P}$$

and this means that there exists some integer k_i such that

$$a_i U - k_i P = b_i$$

Hence

$$\frac{U}{P} - \frac{k_i}{a_i} = \frac{b_i}{a_i P}$$

The length of $a_i, 1 \leq i \leq n$ and U are the same as the length of P and

$$\{b_1, \ldots, b_5\} = \{2^0, \ldots, 2^4\}$$

So consider the first five of the super sequence

$$b_i \leq 2^5, 1 \leq i \leq 5$$

Therefore, we obtain

$$|\frac{U}{P} - \frac{k_i}{a_i}| \leq \frac{2^5}{2^{2\times 1360}} \qquad (4.2)$$

and subtract the $i = 1$ term from the others, we have

$$\left|\frac{k_i}{a_i} - \frac{k_1}{a_1}\right| \leq 2^{6-2\times 1360}, 2 \leq i \leq 5$$

This implies that

$$|k_i a_1 - k_1 a_i| \leq 2^6, 2 \leq i \leq 5 \qquad (4.3)$$

Inequalities (4.3) show that how unusual the a_i and k_i are. After all, the size of each of them is 1360 bits, so the size of $k_i a_1, k_1 a_i$ is 2720 bits. But the size of the difference of two such terms to be 6 bits. We can obtain $k_i, 1 \leq i \leq 5$ by invoking H.W.Lenstra's integer programming algorithm only once. Once the k_i are found,

from equation(4.2) we obtain an approximation to U/P and constructs a pair $(U^{'}, P^{'})$ with $U^{'}/P^{'}$ close to U/P such that the weights obtained by

$$b_i^{'} \equiv a_i U^{'} \pmod{P^{'}}, 0 < b_i < P^{'}, 1 \leq i \leq n$$

form a super increasing sequence. So we can use Shamir's method to find a group of equivalent secret key ($P^{'}, W^{'}$ and a new super increasing sequence $B^{'}$). We can use this equivalent secret key to recover the messages because we can eavesdrop the permutation D.

5 Attack of Su et al.'s cryptosystem

Su's cryptosystem based on a application of the elliptic curve logarithm problem and knapsack problem.

5.1 The description of Su et al.'s cryptosystem

In initial stage, the user selects the elliptic curve domain parameters.
a) A curve $y^2 = x^3 + ax + b$ over F_p, where $4a^3 + 27b^2 \neq 0$.
b) A point $\alpha = (x_0, y_0)$ such that order$(\alpha) > p$ which is a large prime number.
c) A super increasing sequence $A^{'} = \{a_1, a_2, \ldots, a_n\}$, such that

$$a_1 \approx 2^n, a_i > \sum_{j=1}^{i-1} a_j, 2 \leq i \leq n, a_n \approx 2^{2n}, \sum_{i=1}^{n} a_i < p$$

Define the set

$$A = \{a_i \alpha \mid 1 \leq i \leq n\}$$

d) Select a permutation π of $\{1, 2, \ldots, n\}$ and positive integer e, d, such that $\gcd(p, e) = 1, ed \equiv 1 \pmod{p}$, Let

$$S = \{s_i \mid s_i = (ea_{\pi(i)} \pmod{p})\alpha, 1 \leq i \leq n\}$$

$$T = \{t_i \mid t_i \equiv ea_{\pi(i)} \pmod{p}, 1 \leq i \leq n\}$$

The public key is S, T, α and p. The private key is e, d, π and the super increasing sequence $A^{'}$.
Encryption: The sender encodes the plaintext message $M = (m_1, \ldots, m_n)$ to be sent as $x - y$ points Pm_i. So the message becomes $M^{'} = (Pm_1, \ldots, Pm_n)$, chooses a binary message $X = (x_1, x_2, \ldots, x_n)$ and a random positive integer $k, k < p$.
When $x_i = 1$, define $C_{m_i} = \{k\alpha, P_{m_j} + ks_i\}$, otherwise, add confusing data to P_{m_i}. The ciphertext is

$$c = C_{m_1} \parallel C_{m_2} \parallel \cdots C_{m_n} \parallel \sum_{i=1}^{n} x_i t_i \pmod{p}$$

Sends c to the receiver.
Decryption:The receiver computes

$$D = e \cdot C_{m_1} \parallel e \cdot C_{m_2} \parallel \cdots e \cdot C_{m_n} \parallel d \cdot \sum_{i=1}^{n} x_i t_i \pmod{p}$$

where

$$e \cdot C_{m_i} = P_{m_i} + ks_i - eka_{\pi(i)}\alpha = P_{m_i} + kea_{\pi(i)}\alpha - eka_{\pi(i)}\alpha = P_{m_i} \quad (5.1)$$

Since

$$d \cdot \sum_{i=1}^{n} x_i t_i \pmod{p} \equiv \sum_{i=1}^{n} x_i a_{\pi(i)} \pmod{p} = \sum_{i=1}^{n} x_i a_{\pi(i)}$$

The receiver can obtain $X = (x_1, \ldots, x_n)$ by solving the super increasing knapsack problem and obtains $P_{m_i}$from equation (5.1) for $x_i = 1$. Then decodes P_{m_i} to recover the message M.

5.2 Attack of Su et al.'s cryptosystem

We observe that the vector $X = \{x_1, \ldots, x_n\}$ is important. From this vector, the receiver can extract the correct C_{m_i} from the confusing data. Firstly, we give a known the vector X attack and this attack will be used in the following.
Known X Attack

If we know the vector $X = \{x_1, x_2, \ldots, x_n\}$, from equation (5.1), we know

$$e \cdot C_{m_j} = P_{m_i} + ks_i - eka_{\pi(i)}\alpha$$

Notice that

$$t_i = ea_{\pi(i)}$$

Then we can obtain that

$$e \cdot C_{m_j} = P_{m_i} + ks_i - eka_{\pi(i)}\alpha = P_{m_i} + ks_i - t_i k\alpha$$

Unfortunately, the set T is the public key in the cryptosystem. So we can recover the message as long as we know the vector X.
Find the Vector X

To find the vector X, we must solve a Merkle-Hellman-like algorithm. Now we have a super increasing sequence $A^{'} = \{a_1, \ldots, a_n\}$ satisfies

$$a_1 \approx 2^n, a_i > \sum_{j=1}^{i-1} a_j, 2 \leq i \leq n, a_n \approx 2^{2n}, \sum_{i=1}^{n} < p$$

and a permutation π, and the public key is the set

$$T = \{t_i \mid t_i \equiv ea_{\pi(i)} \pmod{p}, 1 \leq i \leq n\}$$

the private key is $A^{'}, e, \pi$. The density

$$d = \frac{n}{\log_2 P} \leq \frac{n}{2n} = \frac{1}{2}$$

So if the n is moderate, we can use low density attack [7][2]to find the vector X. If n is big, we can still find the vector X by using Shamir's method. Consequently, we can recover the message after performing a known X attack .

Finally, we give an improved version of this cryptosystem to avoid the known X attack. In Su's paper, when $x_i = 1$, define $C_{m_i} = \{k\alpha, P_{m_j} + ks_i\}$.If we redefine $C_{m_i} = \{ka_{\pi(i)}\alpha, P_{m_j} + ks_i\}$, this new cryptosystem will be safer than the original one. Even if one could find the vector X he can't recover P_{m_i}, because e is the secret key and it is hard to solve the elliptic curve discrete logarithm problem.

6 Conclusion

In this paper, we cryptanalyze two new cryptosystems based on knapsack problem. The security level of the two encryption schemes are overestimated. We can break Hwang et al.'s cryptosystem use Shamir's method and recover the plaintext of Su et al.'s cryptosystem.

References

1. B. Chor and R.L. Rivest. A knapsack-type public key cryptosystem based on arithmetic in finite fields. IEEE Trans. Inform. Theory, 34, 1988.
2. M.J. Coster, A. Joux, B.A. La Macchia, A.M. Odlyzko, C.P. Schnorr and J. Stern, An improved lowdensity subset sum algorithm, Computational Complexity.2 (1992) 97-186
3. N. Gama and P. Q. Nguyen. Predicting lattice reduction. In Advances in Cryptology C Proc. EUROCRYPT 08, Lecture Notes in Computer Science. Springer, 2008.
4. M.S.Hwang, C.C.Lee, S.F.Tzeng, A new knapsack public-key cryptosystem based on permutation combination algorithm, Information Journal of Applied Mathematics and Computer Sciences 5;1(2009)33-38
5. A.Joux and J.Stern, Lattice reduction:A toolbox for the cryptanalyst. Journal of Cryptology, 11:161-185, 1998.
6. J. C. Lagarias, "Performance Analysis of Shamirs Attack on the Basic Merkle-Hellman Knapsack Cryptosystem, " Proc. 11th Intern. Colloquium on Automata, Languages and Programming (ICALP), Lecture Notes in Computer Science, vol. 172, Springer-Verlag, Berlin, 1984, 312-323.
7. J.C.Lagarias and A.M.Odlyzko.Solving low-density subset sum problems. Jounal of the Association for Computing Machinery, January 1985.
8. A.K. Lenstra, H.W. Lenstra and L. Lovsz, Factoring polynomials with rational coefficients, Mathematische Annalen 261 (1982) 515-534.
9. R.C.Merkle, and M.E.Hellman, Hiding Information and Signatures in Trapdoor Knapsacks.IEEE Trans.inf.Theory vol.24, 1978, 525-530.
10. P.Q.Nguyen and J.Stern, The two faces of lattices in cryptology, Cryptography and Lattices Proc.CALC'01, LNCS, vol.2146, Springer-Verlag, 2001, 146-180.

11. A.M.Odlyzko, The rise and fall of knapsack cryptosystems. In Cryptology and Computional Number Theory, vol.42 of Symposina in Applied Mathematics, 1990, 75-88
12. T. Okamoto, K. Tanaka, and S. Uchiyama. Quantum Public-Key Cryptosystems. In Proc. of Crypto 00, LNCS. Springer-Verlag, 2000.
13. C.-P. Schnorr and M. Euchner. Lattice basis reduction: improved practical algorithms and solving subset sum problems. Math. Programming 66, 1994, 181-199.
14. C.-P. Schnorr and H. H. Horner. Attacking the Chor-Rivest cryptosystem by improved lattice reduction. In Proc. of Eurocrypt 95, volume 921 of Lecture Notes in Computer Science, pages 1-12. IACR, Springer, 1995.
15. A.Shamir, A Polynomial Time Algorithm for Breaking the Basic Merkle-Hellman Cryptosystem.IEEE Trans.inf.Theory vol.30, 1984, 699-704.
16. P. C. Su, E. H. Lu, and K. C. Henry, A knapsack public-key cryptosystem based on elliptic curves discrete logarithm, Applied Mathematics and Computation 168, (2005)40- 46
17. 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, 326-330, January 2001.
18. S. Vaudenay Cryptanalysis of the Chor-Rivest Cryptosystem. In Journal of Cryptology, vol. 14 (2001), 87-100.
19. B. Wang, Q. Wu, and Y. Hu, A knapsack-based probabilistic encryption scheme, Information Sciences, vol. 177, no. 19, 3981-3994, 2007.