Computers and Mathematics with Applications 59 (2010) 194–206

Contents lists available at ScienceDirect

Computers and Mathematics with Applications

journal homepage: www.elsevier.com/locate/camwa

Quadratic compact knapsack public-key cryptosystem

Baocang Wang *, Yupu Hu

The Ministry of Education Key Laboratory of Computer Networks and Information Security, Xidian University, Xi'an 710071, China

ARTICLE INFO

Article history:
Received 7 October 2008
Received in revised form 6 August 2009
Accepted 17 August 2009

Keywords:
Public-key cryptography
Knapsack problem
Low-density subset-sum attack
Lattice basis reduction

ABSTRACT

Knapsack-type cryptosystems were among the first public-key cryptographic schemes to be invented. Their NP-completeness nature and the high speed in encryption/decryption made them very attractive. However, these cryptosystems were shown to be vulnerable to the low-density subset-sum attacks or some key-recovery attacks. In this paper, additive knapsack-type public-key cryptography is reconsidered. We propose a knapsack-type public-key cryptosystem by introducing an easy quadratic compact knapsack problem. The system uses the Chinese remainder theorem to disguise the easy knapsack sequence. The encryption function of the system is nonlinear about the message vector. Under the relinearization attack model, the system enjoys a high density. We show that the knapsack cryptosystem is secure against the low-density subset-sum attacks by observing that the underlying compact knapsack problem has exponentially many solutions. It is shown that the proposed cryptosystem is also secure against some brute-force attacks and some known key-recovery attacks including the simultaneous Diophantine approximation attack and the orthogonal lattice attack.

1. Introduction

Knapsack cryptography [1] is an important class of public-key cryptosystems in the area of public-key cryptography. It involves no expensive modular exponentiations, which makes the encryption and decryption much more efficient than discrete-logarithm-based and factorization-based cryptosystems [2,3]. For a long time, knapsack-type cryptosystems were considered to be the most attractive and the most promising due to their high speed of encryption and decryption and NP-completeness nature. Many knapsack-type cryptosystems were developed in the history of knapsack public-key cryptography especially in the 1980s, and the cryptographic applications of some variants of the knapsack problem were also investigated [4–14]. However, almost all additive knapsack-type cryptosystems were shown to be vulnerable to low-density subset-sum attacks [15–17], GCD attack [18], simultaneous Diophantine approximation attack [19,20] or orthogonal lattice attack [21]. Refer to the survey paper [22] for the rise and fall of knapsack cryptosystems.

Several methods have been used to design knapsack cryptosystems achieving high density, especially non-injective cryptosystems with density greater than 1. One method is due to Koskinen [23] and another one is due to Chor, Rivest [24], and Okamoto et al. [25]. However, the two knapsack cryptosystems given in [23] suffer a drawback that the decryption is time-consuming, while the so-called low-weight knapsack cryptosystems Chor–Rivest [24] and Okamoto–Tanaka–Uchiyama [25] were also shown to be vulnerable to low-density attacks [26–29]. This paper defines a new easy quadratic compact knapsack problem. We construct a new knapsack-type public-key cryptosystem based on the knapsack problem. The cryptosystem achieves a high knapsack density (about 1.27 when the dimension of the public cargo vector $n = 100$) under the relinearization attack model. However, the security of the knapsack cryptosystem against low-density subset-sum attacks is not supported by the estimation of the density. We show that the underlying compact knapsack problem always has exponentially

* Corresponding author.
E-mail address: bcwang79@yahoo.com.cn (B. Wang).

0898-1221/$ – see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.camwa.2009.08.031

many solutions. Hence, it is completely computationally infeasible for the attacker to find all the solutions among which the attacker expects to pick out the plaintext. Thus, we show that the system is secure against the low-density subset-sum attacks. We also show that the system is secure against simultaneous Diophantine approximation and orthogonal lattice attacks.

The rest of the paper is organized as follows: Section 2 lists some useful concepts to understand the security analysis. In Section 3, we define an easy quadratic knapsack problem. The detailed description of the proposed cryptosystem is given in Section 4. Section 5 discusses the performance related issues and specifies the parameter selection. Section 6 analyzes the security of the proposed cryptosystem. Section 7 provides some concluding remarks.

2. Preliminaries

We recall some concepts about the lattice theory, the low-density subset-sum attack and the simultaneous Diophantine approximation problem. These concepts are useful to understand the security analysis of the proposed cryptosystem.

2.1. Notations

Throughout this paper, the following notations will be used. The greatest common divisor of two integers a and b is denoted as $\gcd(a, b)$. For $(a, b) \in \left(\mathbb{Z}^+\right)^2$, and an integer m, $m \bmod (a, b)$ denotes the 2-tuple $(m \bmod a, m \bmod b)$. Naturally, $u \not\equiv v (\bmod (a, b))$ means that $u \bmod a \neq v \bmod a$ or $u \bmod b \neq v \bmod b$. When x is a number, $|x|$ means its absolute value. For an n-dimensional vector $X \in R^n$, we write $\|X\|$ for the Euclidean norm of X. The notation $|A|$ denotes the cardinality of a set A. We write the binary length of an integer a as $|a|_2$. $\lceil r \rceil$ denotes the smallest integer greater than or equal to r.

2.2. Definition of lattice

A lattice is a discrete (additive) subgroup of $\mathbb{R}^n$. More precisely, a lattice consists of all integral linear combinations of a set of linearly independent vectors $\{v_i\}$, i.e.,

$$L = \left\{ \sum_{i=1}^{n} z_i v_i | z_i \in \mathbb{Z} \right\}.$$

The most important algorithmic problems in lattice theory include the shortest vector problem (SVP), the closest vector problem (CVP) and the smallest basis problem (SBP). The SVP is to search for the shortest non-zero vector in a given lattice L. The CVP states that, given a lattice L and a vector v, find a lattice vector s minimizing the length of the vector $v - s$. The SBP requires one to find a lattice basis to minimize the maximum of the lengths of its elements in a given lattice. No polynomial-time algorithm is known for the three problems. The best polynomial-time algorithms for solving SVP achieve only slightly sub-exponential factors, and are based on the LLL algorithm [30]. These problems are of special significance in complexity theory and cryptology.

2.3. Knapsack problems and density

The standard subset sum or 0-1 knapsack problem is defined as follows. Given a cargo vector $A = (a_1, \ldots, a_n)$ and a sum s, find which elements are put into the knapsack, that is, solve the following linear Diophantine equation for the binary variables $X = (x_1, \ldots, x_n)$,

$$\sum_{i=1}^{n} a_i x_i = s, \quad x_i \in \{0,\ 1\}. \tag{1}$$

The standard 0-1 knapsack problem has many variants. One of these variants is compact knapsack problem which given $A = (a_1, \ldots, a_n)$ and s, asks for $X = (x_1, \ldots, x_n)$ with $0 \leq x_i \leq 2^b - 1$ and $b \geq 1$ such that $\sum_{i=1}^{n} a_i x_i = s$. Another problem is quadratic knapsack or matrix cover problem which needs to solve a 0-1 quadratic Diophantine equation $s = XAX^T$, where $X = (x_1, \ldots, x_n), x_i \in \{0,\ 1\}$, and A is an n-dimensional square matrix. These problems had been used to construct knapsack-type public-key cryptosystems [1,4,31,9,14]. In this paper, we will define a new knapsack problem, simultaneous quadratic compact knapsack problem.

In a compact knapsack cryptosystem with public key $A = (a_1, \ldots, a_n)$, a message $M = (m_1, \ldots, m_n)$ with $m_i \in [0, k]$ is encrypted into

$$s = \sum_{i=1}^{n} a_i m_i. \tag{2}$$

An important characteristic of a knapsack cryptosystem is the density of the cryptosystem. The density for compact knapsack problems (2) is defined as $d = nb/\log_2 \max_{1 \leq i \leq n} a_i$ [32], where we set $b = \lceil \log_2(k + 1) \rceil$ in that b bits are needed to represent the $k + 1$ integers in $[0, k]$. We note that when $b = 1$, the definition immediately gives the density for 0-1 knapsack problems (1), $d = n/\log_2 \max_{1 \leq i \leq n} a_i$ [17,16]. Here we also point out that some other definitions appear in the

literature for the density of compact knapsack problems (2). For example, Katayangi and Murakami defined the density of (2) as $d = n\lceil \log_2(k+1)\rceil / \log_2\left(k\sum_{i=1}^{n} a_i\right)$ [33]. In fact, when n approaches infinity, the two definitions are asymptotically identical.

2.4. Low-density subset-sum attacks

The basic idea of low-density attacks is to solve the knapsack problem with lattice basis reduction algorithms. Given a cargo vector $A = (a_1, \ldots, a_n)$ and a sum $s = \sum_{i=1}^{n} a_i x_i$, the Lagarias–Odlyzko low-density attack [16] is outlined as follows.

Construct the following matrix

$$V = \begin{pmatrix} 1 & 0 & \cdots & 0 & \delta a_1 \\ 0 & 1 & \cdots & 0 & \delta a_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & \delta a_n \\ 0 & 0 & \cdots & 0 & -\delta s \end{pmatrix} = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \\ v_{n+1} \end{pmatrix}.$$

The integral combinations of the row vectors $\{v_i\}$ of the matrix V imply an $(n+1)$-dimensional lattice L. Note that the vector $v = (x_1, \ldots, x_n, 0) = x_1 v_1 + \cdots + x_n v_n + v_{n+1} \in L$, and v is short. Hence, the short vector v may be found with lattice basis reduction algorithms.

In their original paper [16], Lagarias and Adlyzko set $\delta = 1$ and showed that when the density d of a 0-1 knapsack problem is less than 0.645, the knapsack problem almost always can be solved with a single call to a lattice oracle. Coster et al. obtained a new bound 0.6463 by setting $\delta > \sqrt{n}$ [17]. The authors of [17] also derived a better bound 0.9408 by introducing two different lattices, which we will not detail in this paper. The above bounds only apply to 0-1 knapsack problems. For compact knapsack problems (2), Lee and Park set $\delta > k\sqrt{n}$ and argued that when $b = \lceil \log_2(k+1)\rceil$ is much larger than n and if $d < 1$, the compact knapsack problem is almost always solvable with a single call to a lattice oracle [32]. However, as far as the authors know, no general results are obtained for compact knapsack problems (2).

2.5. Simultaneous Diophantine approximation

The simultaneous Diophantine approximation problem states that, given $n+1$ real numbers $r_1, \ldots, r_n, \varepsilon > 0$, and an integer $Q > 0$, find integers $p_1, \ldots, p_n$, and q such that $0 < q \le Q$, and

$$\left| r_i - \frac{p_i}{q} \right| \le \frac{\varepsilon}{q}, \quad i = 1, \ldots, n.$$

There exists a solution to the simultaneous Diophantine approximation problem if $Q \ge \varepsilon^{-n}$. However, no efficient algorithm is known to find the solution. This problem is also related to lattice basis reduction and the solution can be approximated with lattice basis reduction algorithms.

We illustrate the reduction in a bit more details. Similarly to the above low-density subset-sum attack, the key point is to find a lattice to represent the simultaneous Diophantine approximation problem. Note that the integral linear combinations of the row vectors of the following matrix form a lattice L,

$$V = \begin{pmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \\ -r_1 & -r_2 & \cdots & -r_n & \varepsilon/Q \end{pmatrix} = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \\ a_{n+1} \end{pmatrix}.$$

One can employ lattice basis reduction algorithms and obtain a reduced basis of the lattice L. The shortest vector b in the reduced basis can be used to solve the simultaneous Diophantine approximation problem. Since $b \in L$, there exist integers $p_1, \ldots, p_n$, and q such that

$$b = p_1 a_1 + \cdots + p_n a_n + q a_{n+1} = (p_1 - q r_1, \ldots, p_n - q r_n, q\varepsilon/Q).$$

Because b is short, $p_i - q r_i$ is small for $i = 1, \ldots, n$. Hence, $|r_i - p_i/q|$ is small for each $i = 1, \ldots, n$. From the discussion above, the set of fractions $\{p_i/q\}$ with a common denominator q is approximate to $\{r_i\}$. This observation illustrates the relation between the lattice reduction algorithms and the simultaneous Diophantine approximation problem.

3. An easy simultaneous quadratic knapsack problem

Knapsack cryptosystems are always constructed by finding an easy knapsack problem and then transforming this easy knapsack into a seemingly hard knapsack problem. In this section, we give an easy quadratic knapsack problem. The cargo

vector defined in the easy knapsack problem is different from the super-increasing sequences [1], the cargo vectors used in Graham–Shamir cryptosystem [34], and the knapsack sequences [35] used for attacking a knapsack-type cryptosystem [12] based on Diophantine equations, and it can be thought as the generalization of the cargo vectors given in [31,14].

Now we consider two sets $I \subset \mathbb{Z}$ and $J = \{j = (j_1, j_2) | j_1, j_2 \in \mathbb{Z}^+\}$. We use J^T to denote the set $\{(j_2, j_1) | (j_1, j_2) \in J\}$. Given a $j = (j_1, j_2) \in J$, the set $I \bmod j = \{(i \bmod j_1, i \bmod j_2) | i \in I\}$. Generally, we have $\forall j \in J$, $|I \bmod j| \leq |I|$. It is easy to see that $|I \bmod j| = |I|$ if and only if different integers in I modulo j always produce different integer pairs, i.e., $\forall i_1 \neq i_2 \in I$, and $J = (j_1, j_2) \in J$, $i_1 \not\equiv i_2 \bmod (j_1, j_2)$.

Definition 1. If $\forall j \in J$, $|I \bmod j| = |I|$, we call the set I distinguishable (DIST) modulo J.

More concretely, we set $I = \{i^2 | i = 0, 1, \ldots, 15\}$ and $J = W \cup W^T$, where W is a set consisting of the following integer pairs: (1, 31), (1, 34), (1, 37), (1, 38), (1, 41), (1, 43), (1, 46), (1, 47), (1, 53), (1, 58), (1, 59), (1, 61), (1, 62), (1, 67), (1, 68), (1, 71), (1, 73), (1, 74), (1, 76), (1, 78), (1, 79), (1, 82), (1, 83), (1, 86), (1, 87), (1, 89), (1, 92), (1, 93), (1, 94), (1, 97), (2, 17), (2, 19), (2, 23), (2, 29), (2, 31), (2, 34), (2, 37), (2, 38), (2, 39), (2, 41), (2, 43), (2, 46), (2, 47), (3, 26), (3, 29), (3, 31), (4, 17), (4, 19), (4, 23), (6, 13). It is easy but tedious to verify that I is DIST modulo J. Take $(3, 31), (17, 2) \in J$ as an example. We compute $I \bmod (3, 31)$ as a table $\{(0, 0), (1, 1), (1, 4), (0, 9), (1, 16), (1, 25), (0, 5), (1, 18), (1, 2), (0, 19), (1, 7), (1, 28), (0, 20), (1, 14), (1, 10), (0, 8)\}$, $I \bmod (17, 2) = \{(0, 0), (1, 1), (4, 0), (9, 1), (16, 0), (8, 1), (2, 0), (15, 1), (13, 0), (13, 1), (15, 0), (2, 1), (8, 0), (16, 1), (9, 0), (4, 1)\}$. Hence, both cardinalities of $I \bmod (3, 31)$ and $I \bmod (17, 2)$ equal to $|I| = 16$. Definition 1 also says that if I is DIST modulo J, given j and $i^2 \bmod j$, we can uniquely determine the integer i. For example, given $i^2 \bmod (3, 31) = (0, 5)$, from the table generated via $I \bmod (3, 31)$, we look up the table and find that the 6th component matches (0, 5) (starting from the 0th entry (0,0)), so we can uniquely determine $i^2 = 6^2 = 36$, $i = 6$. Similarly, from $i^2 \bmod (17, 2) = (13, 1)$ we can uniquely determine $i^2 = 9^2 = 81$, $i = 9$. In fact, J contains all the 100 integer pairs $j = (j_1, j_2)$ with $j_1 j_2 < 100$ such that I is DIST modulo J. We also should note that for any non-empty subset G of J, I is also DIST modulo G.

In the rest of the paper, we always use I and J to denote the set $I = \{i^2 | i = 0, 1, \ldots, 15\}$ and the set containing the 100 integer pairs respectively. Now we look at an example.

Example 1. Solve the simultaneous quadratic knapsack problem

$$49x_1^2 + 48x_2^2 + 51x_3^2 = 14\,271, \qquad 53x_1^2 + 31x_2^2 + 62x_3^2 = 14\,879, \quad x_i^2 \in I.$$

To solve the problem, we need to compute $I \bmod (3, 31)$ and $I \bmod (2, 17)$. Note that $\gcd(48, 51) = 3$ and $\gcd(31, 62) = 31$. The two equations modulo 3 and 31 gives two congruences respectively,

$$49x_1^2 + 48x_2^2 + 51x_3^2 \equiv 14\,271 \bmod 3, \qquad 53x_1^2 + 31x_2^2 + 62x_3^2 \equiv 14\,897 \bmod 31.$$

Hence, $x_1^2 \equiv (0, 5) \bmod (3, 31)$. Accordingly, we get $x_1 = 6$. Note that $48x_2^2 + 51x_3^2 = 14\,271 - 49x_1^2$ and we obtain $16x_2^2 + 17x_3^2 = 4169$. Similarly, we get $x_2^2 + 2x_3^2 = 419$ from the second equation. The two equations modulo 17 and 2 give $x_2^2 \equiv 13 \bmod 17$, $x_2^2 \equiv 1 \bmod 2$, from which we know $x_2 = 9$. Solving $x_3 = 13$ is immediate.

We summarize the results shown in the example using a theorem.

Theorem 1. *Given two cargo vectors $A = (a_1, \ldots, a_n)$ and $B = (b_1, \ldots, b_n)$. Denote by c_i and d_i the gcd of the latter i components of A and B respectively, i.e., $c_i = \gcd(a_n, \ldots, a_{n-i+1})$, $d_i = \gcd(b_n, \ldots, b_{n-i+1})$. Without loss of generality, we assume that $c_n = d_n = 1$. Let $G = \{g_i = (g_{1i}, g_{2i}) = (c_{i-1}/c_i, d_{i-1}/d_i) | i = 2, \ldots, n\}$. If $G \subset J$, the simultaneous quadratic knapsack problem*

$$\sum_{i=1}^{n} a_i x_i^2 = s_1, \qquad \sum_{i=1}^{n} b_i x_i^2 = s_2, \quad x_i^2 \in I \tag{3}$$

can be solved in polynomial (about n) time. Furthermore, the problem has at most one solution about $X = (x_1, \ldots, x_n)$ with $x_i \in \{0, 1, \ldots, 15\}$.

Proof. The following facts are obvious. Firstly, I is DIST modulo G. Hence, given g_i and an integer pair (a, b) belonging to the table generated via I modulo g_i, we can determine a unique x_i such that $x_i^2 \in I$ and $(a, b) \equiv x_i^2 \bmod g_i$. Secondly, $c_i = \gcd(c_{i-1}, a_{n-i+1})$ and $d_i = \gcd(d_{i-1}, b_{n-i+1})$. Hence, $\gcd(c_{i-1}/c_i, a_{n-i+1}/c_i) = \gcd(d_{i-1}/d_i, b_{n-i+1}/d_i) = 1$, and $c_{i-1} | a_j$, $d_{i-1} | b_j$, when $n - i + 2 \leq j \leq n$. Assume that (3) has solutions. We use mathematical induction to prove that we can solve (3).

The two equations of (3) modulo c_{n-1} and d_{n-1} respectively give $a_1 x_1^2 \equiv s_1 \bmod c_{n-1}$, $b_1 x_1^2 \equiv s_2 \bmod d_{n-1}$. Hence, we have $x_1^2 \equiv a_1^{-1} s_1 \bmod c_{n-1}$, $x_1^2 \equiv b_1^{-1} s_2 \bmod d_{n-1}$. Note that $g_n = (g_{1n}, g_{2n}) = (c_{n-1}/c_n, d_{n-1}/d_n) = (c_{n-1}, d_{n-1})$. So we obtain $x_1^2 \equiv (a_1^{-1} s_1, b_1^{-1} s_2) \bmod g_n$, from which we uniquely determine $x_1^2 \in I$ and hence the value of $x_1 \in \{0, \ldots, 15\}$.

We assume that the values of $x_1, \ldots, x_i$, $1 \leq i \leq n - 2$ have been determined. So

$$\sum_{j=i+1}^{n} a_j x_j^2 = s_1 - \sum_{j=1}^{i} a_j x_j^2, \qquad \sum_{j=i+1}^{n} b_j x_j^2 = s_2 - \sum_{j=1}^{i} b_j x_j^2. \tag{4}$$

Note that the two equations of (4) modulo c_{n-i-1} and d_{n-i-1} respectively give

$$a_{i+1}x_{i+1}^2 \equiv s_1 - \sum_{j=1}^{i} a_j x_j^2 \bmod c_{n-i-1},$$

$$b_{i+1}x_{i+1}^2 \equiv s_2 - \sum_{j=1}^{i} b_j x_j^2 \bmod d_{n-i-1}.$$

So we have

$$\frac{a_{i+1}}{c_{n-i}}x_{i+1}^2 \equiv \frac{s_1 - \sum_{j=1}^{i} a_j x_j^2}{c_{n-i}} \bmod \frac{c_{n-i-1}}{c_{n-i}},$$

$$\frac{b_{i+1}}{d_{n-i}}x_{i+1}^2 \equiv \frac{s_2 - \sum_{j=1}^{i} b_j x_j^2}{d_{n-i}} \bmod \frac{d_{n-i-1}}{d_{n-i}},$$

from which we can uniquely determine x_{i+1}^2 modulo g_{n-i} and hence the value of x_{i+1},

$$x_{i+1}^2 \equiv \left(\left(\frac{a_{i+1}}{c_{n-i}}\right)^{-1} \frac{s_1 - \sum_{j=1}^{i} a_j x_j^2}{c_{n-i}}, \left(\frac{b_{i+1}}{d_{n-i}}\right)^{-1} \frac{s_2 - \sum_{j=1}^{i} b_j x_j^2}{d_{n-i}} \right) \bmod g_{n-i}.$$

After the values of $x_1, \ldots, x_{n-1}$ have been determined, we get two equations $a_n x_n^2 = s_1 - \sum_{j=1}^{n-1} a_j x_j^2$, and $b_n x_n^2 = s_2 - \sum_{j=1}^{n-1} b_j x_j^2$. Either equation gives the unique value of $x_n^2 = (s_1 - \sum_{j=1}^{n-1} a_j x_j^2)/a_n = (s_2 - \sum_{j=1}^{n-1} b_j x_j^2)/b_n$. Hence, we also can uniquely determine $x_n \in \{0, \ldots, 15\}$.

If and only if one of the following cases appears, we conclude that (3) has no solutions. Firstly, there exists an i such that either $c_{n-i} | s_1 - \sum_{j=1}^{i} a_j x_j^2$ or $d_{n-i} | s_2 - \sum_{j=1}^{i} b_j x_j^2$ does not hold. Secondly, all the values of $x_1, \ldots, x_{n-1}$ can be uniquely determined. However, the two values $(s_1 - \sum_{j=1}^{n-1} a_j x_j^2)/a_n$ and $(s_2 - \sum_{j=1}^{n-1} b_j x_j^2)/b_n$ are not identical. Thirdly, all the values of $x_1, \ldots, x_n$ can be uniquely determined, but they cannot match the equations of (3) simultaneously.

To determine each x_i, we need to solve two linear congruences and look up the table generated via $I \bmod g_{n-i+1}$. So the simultaneous quadratic knapsack problem (3) can be solved in polynomial (about n) time. If the problem has solutions, each x_i is uniquely determined. So the simultaneous quadratic knapsack problem has at most one solution. □

For given input A, B subject to the requirements of *Theorem 1* and s_1, s_2, the algorithm to solve (3) runs as follows.

Algorithm 1. 1. Compute $c_i = \gcd(a_n, \ldots, a_{n-i+1})$, $d_i = \gcd(b_n, \ldots, b_{n-i+1})$ for $i = 1, \ldots, n$ and $G = \{g_i = (g_{1i}, g_{2i}) = (c_{i-1}/c_i, d_{i-1}/d_i) | i = 2, \ldots, n\}$.
2. For $i = 2, \ldots, n$, compute and store Table i generated via $I \bmod g_i$.
3. Compute $x_1^2 \equiv (a_1^{-1}s_1, b_1^{-1}s_2) \bmod g_n$, and look up Table 1. If the k-th component of Table 1 matches $(a_1^{-1}s_1, b_1^{-1}s_2)$, store $x_1 = k$; Otherwise, output "No Solutions" and exit.
4. For $i = 2, \ldots, n-1$, decide whether c_{n-i+1} and d_{n-i+1} divide $r_{1i} = s_1 - \sum_{j=1}^{i-1} a_j x_j^2$ and $r_{2i} = s_2 - \sum_{j=1}^{i-1} b_j x_j^2$ respectively. If no, output "No Solutions" and exit; otherwise, calculate

$$x_i^2 \equiv \left(\left(\frac{a_i}{c_{n-i+1}}\right)^{-1} \frac{r_{1i}}{c_{n-i+1}}, \left(\frac{b_i}{d_{n-i+1}}\right)^{-1} \frac{r_{2i}}{d_{n-i+1}} \right) = (l_{1i}, l_{2i}) \bmod g_{n-i+1},$$

and look up Table i. If the k-th component of Table i matches (l_{1i}, l_{2i}), store $x_i = k$; Otherwise, output "No Solutions" and exit.
5. Decide whether a_n divides $r_{1n} = s_1 - \sum_{j=1}^{n-1} a_j x_j^2$, b_n divides $r_{2n} = s_2 - \sum_{j=1}^{n-1} b_j x_j^2$ and $r_{1n}/a_n = r_{2n}/b_n$ or not. If yes, set $x_n^2 = r_{1n}/a_n = r_{2n}/b_n$; Otherwise, output "No Solutions" and exit. It is easy to solve x_n from x_n^2. Store x_n.
6. Decide whether $\sum_{i=1}^{n} a_i x_i^2 = s_1$ and $\sum_{i=1}^{n} b_i x_i^2 = s_2$. If yes, output $X = (x_1, \ldots, x_n)$ and exit; Otherwise, output "No Solutions" and exit.

Now we begin to construct a public-key cryptosystem in the next section by using the above easy knapsack problem.

4. The proposed cryptosystem

The proposed asymmetric encryption algorithm consists of three sub-algorithms: key generation, encryption and decryption.

4.1. Key generation

The key generation algorithm runs as follows.

1. Randomly choose two cargo vectors $A = (a_1, \ldots, a_n)$ and $B = (b_1, \ldots, b_n)$ which satisfy the requirements of Theorem 1.
2. Randomly choose a 2-dimensional square matrix $C = (c_{ij})_{2\times 2}$ with determinant 1 and the length of its entries upper-bounded by a constant, that is, $|c_{ij}|_2 = O(1)$ and write the inverse of C as C^{-1}.
3. Compute
$$\begin{pmatrix}\hat{A}\\ \hat{B}\end{pmatrix} = \begin{pmatrix}\hat{a}_1 & \cdots & \hat{a}_n\\ \hat{b}_1 & \cdots & \hat{b}_n\end{pmatrix} = C\begin{pmatrix}A\\ B\end{pmatrix}.$$
4. Randomly choose two prime integers p and q slightly greater than $225\sum_{i=1}^{n}\hat{a}_i$ and $225\sum_{i=1}^{n}\hat{b}_i$ respectively, and compute $N = pq$.
5. Use the Chinese remainder theorem to generate a cargo vector $E = (e_1, \ldots, e_n)$,
$$e_i \equiv \hat{a}_i \bmod p, \qquad e_i \equiv \hat{b}_i \bmod q, \tag{5}$$
6. Randomly choose an invertible integer v over $\mathbb{Z}_N$.
7. Compute
$$f_i \equiv e_i v \bmod N. \tag{6}$$

- **Secret key:** N, p, q, C^{-1}, and $v^{-1} \bmod N$;
- **Public key:** F.

Remark 1. In fact, to decrypt a given ciphertext, we also need to compute and store the values of $c_i = \gcd(a_n, \ldots, a_{n-i+1})$, $d_i = \gcd(b_n, \ldots, b_{n-i+1})$, $i = 1, \ldots, n$, the set $G = \{g_i = (g_{1i}, g_{2i}) = (c_{i-1}/c_i, d_{i-1}/d_i) | i = 2, \ldots, n\}$, and $n-1$ tables generated via $I \bmod g_i$ for $i = 2, \ldots, n$. However, these values are easily computed from the public and secret keys. As in other knapsack cryptosystems, the public cargo vector F can be permutated and re-indexed for increased security.

We use Algorithm 2 to generate two cargo vectors A and B satisfying the requirements of Theorem 1.

Algorithm 2. 1. Randomly choose $n-1$ integer pairs $g'_i = (g'_{1i}, g'_{2i}) \in J, i = 2, \ldots, n$ with repetition permitted.
2. Randomly choose $2(n-1)$ numbers $s_1, \ldots, s_{n-1}$ and $t_1, \ldots, t_{n-1}$ satisfying the following requirements. (1). $\gcd(s_i, g'_{1j}) = 1$. (2). $\gcd(t_i, g'_{2j}) = 1$. (3). $\gcd(s_i, s_{i+1}) = 1$; (4). $\gcd(t_i, t_{i+1}) = 1$.
3. Let $a_1 = s_1, b_1 = t_1$. Compute
$$a_i = s_i \prod_{j=n-i+2}^{n} g'_{1j}, \qquad b_i = t_i \prod_{j=n-i+2}^{n} g'_{2j}, \quad i = 2, \ldots, n-1,$$
$$a_n = \prod_{j=2}^{n} g'_{1j}, \qquad b_n = \prod_{j=2}^{n} g'_{2j}.$$
4. Output $A = (a_1, \ldots, a_n), B = (b_1, \ldots, b_n)$, and exit.

We prove that the generated vectors A and B really satisfy the requirement of Theorem 1.

Theorem 2. *The generated vectors A and B satisfy the requirements of Theorem 1.*

Proof. Denote c_i and d_i as the gcd of the latter i components of A and B respectively. Hence, we only need to show that for each $i = 2, \ldots, n, g_i = (c_{i-1}/c_i, d_{i-1}/d_i) \in J$.

It is easy to verify that
$$c_1 = a_n = \prod_{j=2}^{n} g'_{1j}, \qquad d_1 = b_n = \prod_{j=2}^{n} g'_{2j},$$
$$c_i = \gcd\left(\prod_{j=2}^{n} g'_{1j}, \ldots, s_{n-i+1}\prod_{j=i+1}^{n} g'_{1j}\right) = \gcd\left(\prod_{j=2}^{n} g'_{1j}, \ldots, \prod_{j=i+1}^{n} g'_{1j}\right) = \prod_{j=i+1}^{n} g'_{1j},$$
$$d_i = \prod_{j=i+1}^{n} g'_{2j}, \quad i = 2, \ldots, n-1,$$
$$c_n = \gcd(c_{n-1}, s_1) = \gcd(g'_{in}, s_1) = 1, \qquad d_n = 1.$$

So for $i = 2, \ldots, n-1$, we have

$$g_i = \left(\frac{c_{i-1}}{c_i}, \frac{d_{i-1}}{d_i}\right) = \left(\frac{\prod_{j=i}^{n} g'_{1j}}{\prod_{j=i+1}^{n} g'_{1j}}, \frac{\prod_{j=i}^{n} g'_{2j}}{\prod_{j=i+1}^{n} g'_{2j}}\right) = \left(g'_{1i}, g'_{2i}\right) = g'_i \in J,$$

$$g_n = \left(\frac{c_{n-1}}{c_n}, \frac{d_{n-1}}{d_n}\right) = (c_{n-1}, d_{n-1}) = \left(g'_{1n}, g'_{2n}\right) = g'_n \in J,$$

as desired. □

4.2. Encryption

The message $M = (m_1, \ldots, m_n)$ with $|m_i|_2 = 4$, i.e., $m_i \in \{0, \ldots, 15\}$, is encrypted into,

$$c = \sum_{i=1}^{n} f_i m_i^2. \tag{7}$$

4.3. Decryption

To decipher a ciphertext c, the receiver does the followings.

1. Compute

$$t \equiv cv^{-1} \equiv \sum_{i=1}^{n} e_i m_i^2 \pmod{N}. \tag{8}$$

2. Compute $t_p \equiv t \pmod{p}$, $t_q \equiv t \pmod{q}$, and $(s_A, s_B)^T = C^{-1}\left(t_p, t_q\right)^T$.
3. Solve the following simultaneous quadratic compact knapsack problem to recover the plaintext by using Algorithm 1,

$$s_A = \sum_{i=1}^{n} a_i m_i^2, \qquad s_B = \sum_{i=1}^{n} b_i m_i^2, \quad m_i^2 \in I. \tag{9}$$

4.4. Why decryption works?

We show how to recover the plaintext by solving the quadratic compact knapsack problem (7) according to the knowledge of the secret key and Algorithm 1. From (8) and (5), we have $t_p \equiv t \equiv \sum_{i=1}^{n} \hat{a}_i m_i^2 \pmod{p}$, $t_q \equiv t \equiv \sum_{i=1}^{n} \hat{b}_i m_i^2 \pmod{q}$. From the size conditions $p > 225\sum_{i=1}^{n} \hat{a}_i$ and $q > 225\sum_{i=1}^{n} \hat{b}_i$, we obtain $t_p = \sum_{i=1}^{n} \hat{a}_i m_i^2$ and $t_q = \sum_{i=1}^{n} \hat{b}_i m_i^2$. Therefore,

$$\begin{pmatrix} s_A \\ s_B \end{pmatrix} = C^{-1}\begin{pmatrix} t_p \\ t_q \end{pmatrix} = C^{-1}\begin{pmatrix} \hat{A} \\ \hat{B} \end{pmatrix}\begin{pmatrix} m_1^2 \\ \vdots \\ m_n^2 \end{pmatrix} = \begin{pmatrix} A \\ B \end{pmatrix}\begin{pmatrix} m_1^2 \\ \vdots \\ m_n^2 \end{pmatrix},$$

that is, (9). According to Theorem 1, this is a simultaneous quadratic compact knapsack problem, and can be efficiently solved by using Algorithm 1.

5. Suggested parameters and performance

5.1. Suggested parameters

We should choose p and q slightly greater than $225\sum_{i=1}^{n} \hat{a}_i$ and $225\sum_{i=1}^{n} \hat{b}_i$ respectively. Hence, we always can assume that $p \approx 225\sum_{i=1}^{n} \hat{a}_i \approx 225\sum_{i=1}^{n} a_i$ and $q \approx 225\sum_{i=1}^{n} \hat{b}_i \approx 225\sum_{i=1}^{n} b_i$ in that the entries of C is bounded by $O(1)$. In Algorithm 2, we need to choose s_i and t_i carefully in order that the generated a_i and b_i always have the same binary length. In fact, we can choose those s_i and t_i with lengths

$$|s_i|_2 = \left|\prod_{j=2}^{n} g'_{1j}\right|_2 - \left|\prod_{j=n-i+2}^{n} g'_{1j}\right|_2, \qquad |t_i|_2 = \left|\prod_{j=2}^{n} g'_{2j}\right|_2 - \left|\prod_{j=n-i+2}^{n} g'_{2j}\right|_2.$$

So,

$$|a_1|_2 \approx |a_2|_2 \approx \cdots \approx |a_n|_2 = \left|\prod_{j=2}^{n} g'_{1j}\right|_2,$$

$$|b_1|_2 \approx |b_2|_2 \approx \cdots \approx |b_n|_2 = \left|\prod_{j=2}^{n} g'_{2j}\right|_2.$$

In implementation, $n = 100$ is suggested. We recommend to choose C as

$$C_1 = \begin{pmatrix} 2 & 1 \\ 3 & 2 \end{pmatrix} \quad \text{or} \quad C_2 = \begin{pmatrix} 1 & 2 \\ 2 & 5 \end{pmatrix}.$$

We note that the recommended parameter $n = 100$ happens to be identical to the cardinality of J, $|J| = 100$. However, this does not mean we cannot choose a larger n. In fact, we have stated in the first step of Algorithm 2 that the $n - 1$ integer pairs $g'_i = (g'_{1i}, g'_{2i}) \in J$ for $i = 2, \ldots, n$ are generated with repetition permitted. Hence, we can choose as many g'_i's as we want. If repetitions were not permitted, we only can generate at most $|J| = 100$ g'_i's, and hence n is at most $|J| + 1 = 101$. If $n > 101$, there must exist a $g'_i \in J$ that is chosen at least two times. We recommend $n = 100$ mainly due to security considerations. In Section 6.1, we show that when $n = 100$, the attacker needs to perform $n16^{n/2} \approx 2^{206.6}$ operations to recover the plaintext. In Section 6.2, we show that for a ciphertext, there are about 2^{151} preimages. So we think that $n = 100$ is sufficient to obtain a high level of security.

5.2. Public-key size

We first assume that the randomly chosen $g'_i = (g'_{1i}, g'_{2i})$ is uniformly distributed over J. Hence, the expected value for $g'_{1i}g'_{2i}$ should be the geometric mean of all the 100 products $g_{1i}g_{2i}$, which is about

$$g'_{1i}g'_{2i} \approx \sqrt[100]{\prod_{(g_{1i},g_{2i})\in J} g_{1i}g_{2i}} = \sqrt[50]{\prod_{(w_1,w_2)\in W} w_1 w_2} \approx 60.86. \tag{10}$$

The binary length of the public key of the proposed cryptosystem is

$$\sum_{i=1}^{n} \lceil \log_2 f_i \rceil \approx n|f_1|_2 \approx \cdots \approx n|f_n|_2 \approx n|N|_2.$$

Note that

$$f_i \approx N = pq \approx 225^2 \left(\sum_{i=1}^{n} \hat{a}_i\right)\left(\sum_{i=1}^{n} \hat{b}_i\right) \approx 225^2 n^2 \hat{a}_n \hat{b}_n$$

$$\approx 225^2 n^2 a_n b_n = 225^2 n^2 \prod_{i=2}^{n} g'_{1i}g'_{2i} \approx 225^2 n^2 60.86^{n-1}.$$

So the binary length of N is $|N|_2 = (n-1)|60.86|_2 + 2|225n|_2$, which is bounded by $O(n)$. The public-key size is about $100|225^2 n^2 60.86^{n-1}|_2 \approx 6157$ bits when $n = 100$.

5.3. Information rate

The information rate defined as the ratio of the binary length of the message to that of the ciphertext of the proposed cryptosystem is $\rho = 4n/\log_2 C_{\max}$. Note that

$$C_{\max} = 225 \sum_{i=1}^{n} f_i \approx 225nN \approx 225^3 n^3 60.86^{n-1}. \tag{11}$$

So the information rate is evaluated by $\rho \approx 4n/\log_2(225^3 n^3 60.86^{n-1})$. When $n = 100$, the information rate is about 0.63.

5.4. Computational complexity

The encryption of the cryptosystem only performs $O(n)$ multiplication and addition operations. The binary lengths of the involved integers are bounded by $O(n)$ and $O(1)$ respectively, so the complexity of the encryption is $O(n^2)$. The decryption needs a modular multiplication operation for two large numbers, whose lengths are bounded by $O(n)$. So the computational complexity for carrying out the modular multiplication operation is given as $O(n^2)$. Furthermore, to decipher a given ciphertext, the decryption algorithm has to perform a 2-dimensional matrix-vector multiplication, $O(n)$ table-query operations and $O(n)$ modular multiplications and divisions with small integers. So the computational complexity of the decryption algorithm is $O(n^2)$.

6. Security analysis

Like all the other knapsack cryptosystems, our system does not match the provable security goals, either. In this section, we only discuss some known attacks on the proposed cryptosystem. However, we do not consider some specialized attacks such as Shamir's attack [36] on the basic Merkle–Hellman cryptosystem [1] to recover the super-increasing sequences and Vaudenay's algebraic attack [37] on the Chor–Rivest cryptosystem [24].

6.1. On solving the quadratic compact knapsack problem

One straightforward way to break the system is to solve (7) for $M = (m_1, \ldots, m_n)$. The attacker can exhaust $\sum_{i=1}^{n} f_i m_i^2$ for $m_i^2 \in I$. This attack needs 16^n steps. A better method is to first compute $S_1 = \{\sum_{i=1}^{n/2} f_i m_i^2\}$, $S_2 = \{c - \sum_{j=n/2+1}^{n} f_j m_j^2\}$. Then the attacker looks up S_1 and S_2 and expects to find an element $s \in S_1 \cap S_2$. If there exists an element $s = \sum_{i=1}^{n/2} f_i m_i^2 = c - \sum_{j=n/2+1}^{n} f_j m_j^2$, then $c = \sum_{i=1}^{n} f_i m_i^2$ and m_i is extracted. This attack requires about $n16^{n/2}$ steps [22]. For a sufficiently large parameter n, the attack is computationally infeasible.

6.2. Relinearization attack

One of the reasons for the insecurities of some additive knapsack-type cryptosystems is that these systems are basically linear, as observed in [18]. In our cryptosystem, the encryption function (7) is nonlinear about the plaintext vector. Of course, the attacker can obtain a linear function just by setting $y_i = m_i^2$,

$$c = \sum_{i=1}^{n} f_i y_i, \quad y_i \in I. \tag{12}$$

The lattice basis reduction algorithms are always used to find a "small" solution to a linear equation. Naturally, the attacker views (12) as a compact knapsack problem and then launches a low-density subset-sum attack. However, (12) is not a standard compact knapsack problem. In fact, when he launches a low-density subset-sum attack, the problem that the attacker wants to solve is the following standard compact knapsack problem

$$c = \sum_{i=1}^{n} f_i y_i, \quad 0 \le y_i \le 225, \text{ i.e., } y_i \in \mathbb{Z}_{226}. \tag{13}$$

For a solution $Y = (y_1, \ldots, y_n)$ to (13), Y is also a solution to (12) only when every y_i is a square. Otherwise, Y contains little information about the plaintext and hence is useless for the attacker. In other words, in the relinearization attack model, we just select a small space I^n as the plaintext space from a big space $\mathbb{Z}_{226}^n$. The difference of the two sets $\mathbb{Z}_{226}^n - I^n$ is the redundant information added into the messages. A similar method has been used in the Chor–Rivest [24] and Okamoto–Tanaka–Uchiyama [25] schemes. In their schemes, the big space is $\{0, 1\}^n$, while the small space consists of those vectors whose Hamming weight is exactly h.

In the sequel, we begin to show the infeasibility of solving the compact knapsack problem (13) for the plaintext by using lattice reduction algorithms.

6.2.1. Evaluation of density

Now we begin to investigate the effects of the powerful low-density attacks on the security of the proposed system. When applied to a specific knapsack instance, the low-density attacks depend on the density of the knapsack. If we adopt the definition given in [32], we can estimate the density of (13) via

$$d = \frac{nb}{\log_2 \max\limits_{1 \le i \le n} a_i}.$$

When $n = 100$, $d \approx 1.3$. If we use the definition in [33], the density turns out to be

$$d = \frac{\sum\limits_{i=1}^{n} e_i}{\log_2 C_{\max}} \approx \frac{n\lceil \log_2 226 \rceil}{\log_2 \left(225^3 n^3 60.86^{n-1}\right)}.$$

When $n = 100$, the density of (13) is about 1.27, which is sufficiently high.

Now we cannot claim the security of the cryptosystem against the low-density subset-sum attacks due to the following considers. Firstly, no general results are obtained for low-density attacks on compact knapsack problems. We note that Coster et al.'s bound 0.9408 [17] only applies to 0-1 knapsack problems, and Lee and Park's attack applies to compact knapsacks with $b = \lceil \log_2(k+1) \rceil$ much larger than n and density $d < 1$ [32]. All these things say nothing about the compact knapsack problem (13). So we cannot conclude that the proposed knapsack cryptosystem is secure only by observing that

Table 1
An example to illustrate the non-injectivity.

n, C	$n = 9, C = C_1$
A, B	$A = (506731, 514454, 546346, 551242, 536486555458, 620806, 1012894, 1012894)$; $B = (506743, 506729, 502727, 501983, 502603, 546809, 566029, 685193, 923521)$
$\hat{A}, \hat{B}$	$\hat{A} = (1520205, 1535637, 1595419, 1604467, 1575575, 1657725, 1807641, 2710981, 2949309)$; $\hat{B} = (2533679, 2556820, 2644492, 2657692, 2614664, 2759992, 2994476, 4409068, 4885724)$
p, q, N	$p = 3737151991, q = 5344529191, N = 19973317907103269281$
E	(900058651690127277 2, 5505351675426338988, 10499027714189237482, 2509752144226035369, 6504240071281348953, 19487681762645276734, 7504741862919801584, 9508426486509793436, 17500715120317378711)
v, v^{-1}	$v = 9779036791, v^{-1} \bmod N = 2042462702$
F	(11983552636085612996, 10999467547886443030, 15792325467390277628, 10445813110882639381, 9252643203486974008, 17826100034189837380, 1136144594347297305, 1012216192024971939, 10263527667452230037)
M, c	(3, 7, 15, 8, 6, 9, 11, 13, 10), 7980531210038881739482

$d > 0.9408$ or $d > 1$. Secondly, some knapsack cryptosystems achieving high density are also shown vulnerable to low-density subset-sum attacks. For example, the Chor–Rivest [24] and Okamoto–Tanaka–Uchiyama cryptosystems [25] were shown also vulnerable to low-density attacks [26–29]. The success of the low-density subset-sum algorithms in attacking the Chor–Rivest and Okamoto–Tanaka–Uchiyama cryptosystems depends on the fact that the two schemes choose low-weight vectors as the valid plaintext vectors. To claim the security of the proposed cryptosystem against low-density subset-sum attacks, we need to establish a stronger security argument.

6.2.2. Non-injectivity

The basic idea of low-density subset-sum and hence low-weight attacks is to find one [26,27] or polynomially many [28] n-dimensional spheres covering the solution candidates. The success of low-weight attacks depends on the fact that we can use a "small" sphere to cover a lattice point. In our cryptosystem, we should note that (13) always has many (denote the number of solutions as t) solutions and the unique solution to (12) is not necessarily the shortest vector no matter what norms are used. The only difference between the solution $(y_1, \ldots, y_n)$ of (12) and other solutions of (13) is that every y_i is a square. However, the lattice basis reduction algorithms, say the LLL algorithm [30], only considers the size of the entries of a vector. Hence, the special structure of $y_i \in I$ cannot be used by the lattice reduction algorithms. So we can assume that the lattice reduction algorithms just find a random vector in the t solutions. In real life practice, the practical lattice reduction algorithms always can serve as a lattice oracle to output a shortest vector of a low-dimensional lattice. See [26–28]. In fact, when $d > 1$, the compact knapsack problem (13) has many solutions, i.e., the function (13) about Y is non-injective. It may be a difficult task to estimate the lower bound for the number of the solutions that (13) has. The authors only find the estimations of the upper bound in [26]. Generally speaking, the number t is the ratio of the cardinality of the set $\mathbb{Z}_{226}^n$ to the number of the possible ciphertexts,

$$t \approx \frac{226^n}{C_{\max} + 1} \approx \frac{226^n}{225^3 n^3 60.86^{n-1}}.$$

The number t exponentially increases in n. When $n = 100$, $t \approx 2^{151}$. So the encryption function is non-injective under the relinearization attack model. Hence, the low-density subset-sum attack can find the valid plaintext only with a non-negligible probability $1/t \approx 1/2^{151}$ when $n = 100$.

To summarize, we assume that the low-density subset-sum attack can find a solution to (13) and that $Y_M = (m_1^2, \ldots, m_n^2)$ is uniformly distributed over all the t solutions to (13). Hence, the attacker succeeds to obtain Y_M with a probability $P = 1/t$. We showed that t exponentially increases with n. Therefore the probability $P = 1/t$ is a negligible function about n.

When discussing the low-weight attacks, we do not prevent the reduction from solving (13) to the shortest vector problem over a lattice. However, we want to point out a fact about the low-weight cryptosystems and our cryptosystem. In a low-weight cryptosystem, the lattice point covered by a sphere, i.e., the shortest vector, is always the unique plaintext. In our cryptosystem, any sphere covering the plaintext lattice point always covers exponentially many solutions to (13). It may be possible to find polynomially many solutions to (13). But it is highly impractical to enumerate all of these solutions, let alone to find them.

6.2.3. An example

The discussion in Section 6.2.2 shows that even if a lattice oracle is available and the attacker can find a solution to the compact knapsack problem (13), the probability is still negligible. Now we use a small example to illustrate the non-injectivity of the knapsack cryptosystem. The parameters are given in Table 1.

In the example, we use Matlab to find that the compact knapsack problem, $c = \sum_{i=1}^{9} f_i y_i$ with $0 \le y_i \le 225$, has $t = 486$ solutions in total, which are not listed in Table 1 for space limitations. Hence, if a low-density attack can output a random solution among the 486 solutions, the attacker will succeed with a probability 1/486. We also note that

$M = (3, 7, 15, 8, 6, 9, 11, 13, 10)$, and that $Y_M = (9, 49, 225, 64, 36, 81, 121, 169, 100)$ with $\|Y\| = 118\,262$ is not the shortest solution among all the 486 solutions. In fact, in the 486 solutions, there exist 152 solutions enjoying an Euclidean norm smaller than 118 262. Hence, Y_M is not the shortest solution. In fact, the shortest solution is (77, 77, 84, 81, 99, 87, 5, 188, 134), which has an Euclidean norm 96170. Note that this is only a small example. If n is chosen relatively large, the number t will be exponentially large, as it was pointed out in Section 6.2.2.

6.3. Diophantine approximation attack

If the attacker obtains N and $v^{-1} \bmod N$, and then factors $N = pq$, he can break the cryptosystem. Now, we consider some known attacks to recover the modulus and the multiplier.

Some knapsack cryptosystems use size conditions to disguise an easy knapsack problem. In such a cryptosystem, every $m_i \in [0, 2^b - 1]$, the modulus is m, and the multiplier is w such that $\gcd(m, w) = 1$. An easy knapsack vector $(a_1, \ldots, a_n)$ is disguised as a seemingly hard knapsack sequence $B = (b_1, \ldots, b_n)$ by using the size condition $m > (2^b - 1)\sum_{i=1}^{n} a_i$ and a modular multiplication $b_i = wa_i (\bmod\ m)$. The size condition can be utilized by the simultaneous Diophantine approximation attack to obtain some useful information about (w, m). See [19,20] for more information about the relation between the simultaneous Diophantine approximation problem and cryptanalytics. Another Diophantine approximation attack due to Lagarias [19] also uses the size conditions to recover some information about the modulus and the multiplier.

The trapdoor of the proposed system is disguised using the Chinese remainder theorem, which involves no size conditions. The attacker cannot expect finding some information about the trapdoor by launching a simultaneous Diophantine approximation attack. So the proposed cryptosystem is also invulnerable to Lagarias' attack. However, the reader may doubt that the size conditions $p > \sum_{i=1}^{n} \hat{a}_i$ and $q > \sum_{i=1}^{n} \hat{b}_i$ have been used. We should observe that if the attacker wants to launch a simultaneous Diophantine approximation attack, he must peel off the outmost shuffles (5) and (6). We will show that this is also a difficult task.

We also point out that Shamir's attack [36] on the basic Merkle–Hellman cryptosystem [1] does not apply to our cryptosystem, either. This is because the basic Merkle–Hellman cryptosystem is constructed by using a super-increasing knapsack sequence and a size condition, while the public cargo vector of the proposed cryptosystem is not constructed from a super-increasing knapsack sequence.

6.4. Orthogonal lattice attack

Another powerful key-recovery attack on knapsack cryptosystem is orthogonal lattice attack [21]. The crucial observation is that the attacked cryptosystem uses smooth numbers in the secret cargo vector. That is, the entries of the secret cargo vector only have small factors. The proposed cryptosystem uses two cargo vectors A and B with some special structures, that is, the entries of A and B only have small factors. However, we should note that the cargo vectors A and B are scrambled by a matrix C. After the multiplication, $\left(\hat{A}, \hat{B}\right)^T = C\,(A, B)^T$, it is highly impossible that all the entries of $\hat{A}$ and $\hat{B}$ still only have small factors. Hence, the orthogonal lattice attack seems infeasible.

6.5. Known N attack

We assume that the exact value of N is known by the attacker. It is straightforward for the attacker to search for $v^{-1} \bmod N$ and then factor N to recover the trapdoor information. We note that anyone can do exhaustively search for the matrix C. According to Algorithm 2, the only distinction between the generated a_i, b_i and a random integer with the same binary length is: when i is small enough, the generated a_i, b_i just contain small prime factors, while a random integer may not be. However, after the two shuffles (5) and (6), the generated vector F will be indistinguishable from a randomly chosen n-dimensional vectors over $\mathbb{Z}_N$. Hence, the attacker will learn little information about a_i and b_i from the public cargo vector F and without factoring N. In fact the best way for the attacker to retrieve the trapdoor seems to factor N at first and then recover the secret vectors A and B.

However, we show that the modulus $N = pq$ must be kept secret. Otherwise, the attacker can factor $N = pq$ and then break the system. Note that $|N|_2 = |225^2 n^2 60.86^{n-1}|_2 \approx 616$ bits when $n = 100$. Hence, it may not be difficult enough for the attacker to factor the modulus N.

6.6. Known p and q attack

Now we consider such a scenario that the attacker has factorized the modulus $N = pq$. If we denote v_p and v_q as the remainder of v modulo p and q, v_p^{-1} and v_q^{-1} for the inverse of $v_p \pmod p$ and $v_q \pmod q$ respectively, and set $f_{ip} = f_i (\bmod p)$, $f_{iq} = f_i (\bmod q)$, (6) modulo p and q result in

$$f_{ip} \equiv v_p \hat{a}_i (\bmod p), \qquad f_{iq} \equiv v_q \hat{b}_i (\bmod q).$$

The attacker can easily compute f_{ip} and f_{iq}. If he recovers v_p^{-1} and v_q^{-1}, he also recovers $\hat{A}$ and $\hat{B}$, $\hat{a}_i \equiv v_p^{-1} f_{ip} (\bmod\ p)$, $\hat{b}_i \equiv v_q^{-1} f_{iq} (\bmod\ q)$.

Without loss of generality, we let

$$v_p^{-1} f_{ip} - l_i p = \hat{a}_i, \quad i = 1, \ldots, n. \tag{14}$$

Divide both sides of (14) by pv_p^{-1}, and we obtain

$$\frac{f_{ip}}{p} - \frac{l_i}{v_p^{-1}} = \frac{\hat{a}_i}{pv_p^{-1}} \tag{15}$$

Note that $p \approx 225 \sum_{j=1}^{n} \hat{a}_j \approx 225n\hat{a}_i$. So we have,

$$\left| \frac{f_{ip}}{p} - \frac{l_i}{v_p^{-1}} \right| \approx \frac{\hat{a}_i}{225n\hat{a}_i v_p^{-1}} \approx \frac{1}{225nv_p^{-1}}.$$

We can claim that $\{l_i/v_p^{-1}\}$ is a set of fractions with a common and relatively small denominator $v_p^{-1} < p$ approximating the set of fractions $\{f_{ip}/p\}$. Or more formally, we can assume that these fractions $\{l_i/v_p^{-1}\}$ are the simultaneous Diophantine approximations of the fractions $\{f_{ip}/p\}$. If there is an efficient algorithm for solving the problem, the attacker can retrieve the secret vector $\hat{A} = (\hat{a}_1, \ldots, \hat{a}_n)$. Using a similar method, he also can recover the vector $\hat{B} = \left(\hat{b}_1, \ldots, \hat{b}_n\right)$.

If $\hat{A}$ and $\hat{B}$ are recovered, anyone can do exhaustively search for C^{-1} in that the entries of C^{-1} are bounded by $O(1)$. Hence, one can recover the secret cargo vectors A and B, $(A, B)^T = C^{-1}\left(\hat{A}, \hat{B}\right)^T$. So the gcd's c_i and d_i are also obtained. The attacker obtains all the secret keys.

6.7. A word of caution

The proposed cryptosystem does not match any provable security objectives. Hence, it cannot be used directly in real life practice. As a public-key cryptographic primitive, the proposed cryptosystem needs further studies. We encourage the reader to examine the security of the proposed cryptosystem, and hope that some paddings can be made to the cryptosystem to make it satisfy some provable security goals if possible.

7. Conclusions

We proposed an easy quadratic knapsack problem and constructed a new knapsack cryptosystem. The cryptosystem enjoys a high knapsack density by adding some redundant information to the plaintext space, and thus it is secure against the low-density subset-sum attack. We also discussed other attacks such as the brute-force attacks and the simultaneous Diophantine approximation attacks. None of them seems to compromise the proposed cryptosystem. However, we failed to achieve any provable security goals. It may be interesting for further studies.

Acknowledgements

The authors would like to thank the anonymous reviewers for their valuable comments on this paper. This work was supported by the National Natural Science Foundation of China (No. 60803149), the National Grand Fundamental Research 973 Program of China (No. 2007CB311201), and the 111 Project (No. B08038).

References

[1] R.C. Merkle, M.E. Hellman, Hiding information and signatures in trapdoor knapsacks, IEEE Transactions on Information Theory IT-24 (1978) 525–530.
[2] D. Liu, D. Huang, P. Luo, Y. Dai, New schemes for sharing points on an elliptic curve, Computers and Mathematics with Applications 56 (2008) 1556–1561.
[3] W.-T Zhu, C.-K. Wu, Security of the redefined Liaws broadcasting cryptosystem, Computers and Mathematics with Applications 56 (2008) 1665–1667.
[4] G. Orton, A multiple-iterated trapdoor for dense compact knapsacks, in: Proceedings of Eurocrypt'94, in: LNCS, vol. 950, Springer-Verlag, Berlin, 1995, pp. 112–130.
[5] M. Morii, M. Kasahara, New public key cryptosystem using discrete logarithm over $GF(p)$, IEICE Transactions J71-D (2) (1988) 448–453.
[6] D. Naccache, J. Stern, A new public-key cryptosystem, in: Proceedings of Eurocrypt'97, in: LNCS, vol. 1233, Springer-Verlag, Berlin, 1997, pp. 27–36.
[7] R.M.F. Goodman, A.J. McAuley, New trapdoor-knapsack public-key cryptosystem, IEE Proceedings 132Pt.E (6) (1985) 282–292.
[8] V. Niemi, A new trapdoor in knapsacks, in: Proceedings of Eurocrypt'90, in: LNCS, vol. 473, Springer-Verlag, Berlin, 1991, pp. 405–411.
[9] R. Janardan, K.B. Lakshmanan, A public-key cryptosystem based on the matrix cover NP-complete problem, in: Proceedings of Crypto'82, Plemum, New York, 1983, pp. 21–37.
[10] M. Qu, S.t A. Vanstone, The knapsack problem in cryptography, in: Finite Fields: Theory, Applications, and Algorithms, Contemporary Mathematics, vol. 168, American Mathematics Society, 1994, pp. 291–308.
[11] J.P. Pieprzyk, On public-key cryptosystems built using polynomial rings, in: Proceedings of Eurorypt'85, in: LNCS, vol. 219, Springer-Verlag, Berlin, 1985, pp. 73–80.
[12] C.H. Lin, C.C. Chang, R.C.T. Lee, A new public-key cipher system based upon the Diophantine equations, IEEE Transactions on Computers 44 (1) (1995) 13–19.
[13] W.A. Webb, A public key cryptosystem based on complementing sets, Cryptologia XVI (2) (1992) 177–181.

[14] B. Wang, Q. Wu, Y. Hu, A knapsack-based encryption scheme, Information Sciences 177 (19) (2007) 3981–3994.
[15] E.F. Brickell, Solving low density knapsacks, in: Advances in Cryptology–Crypto 1983, Plenum, New York, 1984, pp. 24–37.
[16] J.C. Lagarias, A.M. Odlyzko, Solving low-density subset sum problems, Journal of the ACM 32 (1985) 229–246.
[17] M.J. Coster, A. Joux, B.A. LaMacchia, A.M. Odlyzko, C.P. Schnorr, J. Stern, Improved low-density subset sum Algorithm 2 (2) (1992) 111-128.
[18] E.F. Brickell, A.M. Odlyzko, Cryptanalysis: A survey of recent results, in: Contemporary Cryptology, The Science of Information Integrity, IEEE Press, New York, 1992, pp. 501–540.
[19] J.C. Lagarias, Knapsack public key cryptosystems and Diophantine approximation, in: Proceedings of Cypto'83, Plenum, New York, 1984, pp. 3–23.
[20] B. Wang, Y.H. Hu, Diophantine approximation attack on a fast public key cryptosystem, in: The 2nd Information Security Practice and Experience Conference, ISPEC 2006, in: LNCS, vol. 3903, Springer-Verlag, Berlin, 2006, pp. 25–32.
[21] P. Nguyen, J. Stern, Merkle–Hellman revisited: A cryptanalysis of the Qu-Vanstone cryptosystem based on group factorizations, in: Proceedings of Crypto'97, in: LNCS, vol. 1294, Springer-Verlag, Berlin, 1997, pp. 198–212.
[22] A.M. Odlyzko, The rise and fall of knapsack cryptosystems, in: Cryptology and Computational Number Theory, in: Proceedings of Symposia in Applied Mathematics, vol. 42, American Mathematics Society, Providence, RI, 1990, pp. 75–88.
[23] J.A. Koskinen, Non-injective knapsack public-key cryptosystems, Theoretical Computer Science 255 (2001) 401–422.
[24] B. Chor, R.L. Rivest, A knapsack-type public key cryptosystem based on arithmetic in finite fields, IEEE Transactions on Information Theory 34 (1988) 901–909.
[25] T. Okamoto, K. Tanaka, S. Uchiyama, Quantum public-key cryptosystems, in: Proceedings of Crypto'00, in: LNCS, vol. 1880, Springer-Verlag, Berlin, 2000, pp. 147–165.
[26] P. Nguyen, J. Stern, Adapting density attacks to low-weight knapsacks, in: Proceedings of Asiacrypt'05, in: LNCS, vol. 3788, Springer-Verlag, Berlin, 2005, pp. 41–58.
[27] K. Omura, K. Tanaka, Density attack to the knapsack cryptosystems with enumerative source encoding, IEICE Transactions on Fundamentals E84-A (1) (2001) 1564–1569.
[28] I. Tetsuya, K. Jun, K. Takeshi, S. Takeshi, Low-density attack revisited, Design, Codes and Cryptography 43 (2007) 47–59.
[29] N. Kunihiro, New definition of density on knapsack cryptosystems, in: Progress in Cryptology — AFRICACRYPT 2008, in: LNCS, vol. 5023, Springer-Verlag, Berlin, 2008, pp. 156–173.
[30] A.K. Lenstra, H.W. Lenstra Jr., L. Lovász, Factoring polynomials with rational coefficients, Mathematische Annualen 261 (1982) 513–534.
[31] B. Wang, Y. Hu, Knapsack-type public-key cryptosystem with high density, Journal of Electronics and Information Technology 28 (12) (2006) 2390–2393. (in Chinese).
[32] M.K. Lee, K. Park, Low-density attack of public-key cryptosystems based on compact knapsacks, Journal of Electrical Engineering and Information Science 4 (2) (1999) 197–204.
[33] K. Katayangi, Y. Murakami, A new product-sum public-key cryptosystem using message extension, IEICE Transactions on Fundamentals E84-A (10) (2001) 2482–2487.
[34] A. Shamir, R.E. Zippel, On the security of the Merkle–Hellman cryptographic scheme, IEEE Transactions on Information Theory 26 (1980) 339–340.
[35] C.S. Laih, M.J. Gau, Cryptanalysis of a Diophantine equation oriented public key cryptosystem, IEEE Transactions on Computers 46 (1997) 511–512.
[36] A. Shamir, A polynomial-time algorithm for breaking the basic Merkle–Hellman cryptosystem, IEEE Transactions on Information Theory 30 (1984) 699–704.
[37] S. Vaudenay, Cryptanalysis of the Chor–Rivest cryptosystem, Journal of Cryptology 14 (2001) 87–100.