information

MDPI

Article

PKCHD: Towards a Probabilistic Knapsack Public-Key Cryptosystem with High Density

Yuan Ping [1,2,*], Baocang Wang [1,3,*], Shengli Tian [1], Jingxian Zhou [2] and Hui Ma [1]

[1] School of Information Engineering, Xuchang University, Xuchang 461000, China; cb_fan@126.com (S.T.); bsdczy@163.com (H.M.)
[2] Information Technology Research Base of Civil Aviation Administration of China, Civil Aviation University of China, Tianjin 300300, China; yzzxtj@aliyun.com
[3] Key Laboratory of Computer Networks and Information Security, Ministry of Education, Xidian University, Xi'an 710071, China
* Correspondence: pyuan.lhn@xcu.edu.cn (Y.P.); bcwang79@aliyun.com (B.W.)

Received: 22 January 2019; Accepted: 19 February 2019; Published: 21 February 2019

Abstract: By introducing an easy knapsack-type problem, a probabilistic knapsack-type public key cryptosystem (PKCHD) is proposed. It uses a Chinese remainder theorem to disguise the easy knapsack sequence. Thence, to recover the trapdoor information, the implicit attacker has to solve at least two hard number-theoretic problems, namely integer factorization and simultaneous Diophantine approximation problems. In PKCHD, the encryption function is nonlinear about the message vector. Under the re-linearization attack model, PKCHD obtains a high density and is secure against the low-density subset sum attacks, and the success probability for an attacker to recover the message vector with a single call to a lattice oracle is negligible. The infeasibilities of other attacks on the proposed PKCHD are also investigated. Meanwhile, it can use the hardest knapsack vector as the public key if its density evaluates the hardness of a knapsack instance. Furthermore, PKCHD only performs quadratic bit operations which confirms the efficiency of encrypting a message and deciphering a given cipher-text.

Keywords: public key cryptography; knapsack problem; low-density attack; lattice reduction

1. Introduction

A public key cryptosystem (PKC), a concept introduced by Diffie and Hellman in their landmark paper [1], is a critical cryptographic primitive in the area of network and information security. Traditional PKCs such as RSA [2] and ElGamal [3] suffer from the same drawback of relatively low speed, which hampers the further applications of public-key cryptography and also motivates the cryptographers to design faster PKCs. Among the first public-key schemes, knapsack-type cryptosystems were invented as fast PKCs. Due to the high speed of encryption and decryption and their NP-completeness, they were considered to be the most attractive and the most promising for a long time. However, some attacks lowered the initial enthusiasm and even announced the premature death of trapdoor knapsacks.

Following the first knapsack system developed by Merkle and Hellman [4], many knapsack-type cryptosystems can be found. However, only a few of them are considered to be secure, including the most resistant one, the Chor–Rivest knapsack system [5,6]. In the literature, many techniques were developed and many trapdoors were found to hide information, i.e., using the 0–1 knapsack problem [4], compact knapsack problem [7], multiplicative knapsack problem [8,9], modular knapsack problem [10,11], matrix cover problem [12], group factorization problem [13,14], polynomials over $GF(2)$ [15], Diophantine

equations [16], complementing sets [17], and so on. However, almost all the additive knapsack-type cryptosystems are vulnerable to low-density subset sum attacks [18–20], GCD attack [21], simultaneous Diophantine approximation attack [22] or orthogonal lattice attack [14]. Additionally, Refs. [23,24] show the rise and fall of knapsack cryptosystems.

Three reasons clarify the insecurities of the additive knapsack-type cryptosystems. Firstly, as observed in [21], these systems are basically linear. Secondly, for some of them, the trapdoor information is easy to recover. In particular, some systems use the size conditions to disguise an easy knapsack problem that make them vulnerable to simultaneous Diophantine approximation attacks [22]. Thirdly, the densities of some systems are not high enough. Coster et al. [20] showed that, if the density is $<0.9408\cdots$, a single call to a lattice oracle will lead to polynomial time solutions.

Like the aforementioned, to design a secure knapsack-type PKC, we must ensure that

- in the system, the encryption function is nonlinear about the message vector;
- to disguise the easy knapsack problem, the size conditions should be excluded;
- the encryption function must be non-injective. A cipher-text must have so many preimages that it is computationally infeasible for the attacker to list all the preimages.

It is believed in [23] that, if someone invents a knapsack cryptosystem that fully exploits the difficulty of the knapsack problem, with a high density and a difficult-to-discover trapdoor, then it will be a system better than those based on integer factorization and discrete logarithms. Can such a knapsack-type PKC satisfying the requirements above be developed, or, in other words, may any efficient yet straightforward constructions have been overlooked? In this paper, we will try to provide an affirmative answer.

Based on a new easy knapsack-type problem, a probabilistic knapsack public-key cryptosystem with high density (PKCHD) is proposed, which has the following properties:

- PKCHD is a probabilistic knapsack-type PKC.
- The multivariate polynomial encryption function is nonlinear about the message vector, and its degrees are controlled by the randomly-chosen small integers.
- The secret key is disguised via Chinese remainder theorem (CRT) rather than the size conditions. Thus, PKCHD is secure against simultaneous Diophantine approximation attacks.
- The density of PKCHD is sufficiently high under the relinearization attack model. A cipher-text has too many plaintexts for the attacker to enumerate all of them in polynomial time.
- If its density evaluates the hardness of a knapsack instance, PKCHD can always use the hardest knapsack vector as the public-key.
- The attacker has to solve at least two hard number-theoretic problems, namely integer factorization and simultaneous Diophantine approximation problems, to recover the trapdoor information.
- PKCHD is more efficient than RSA [2] and ElGamal [3]. The encryption and the decryption of the system only perform $O(n^2)$ bit operations.

The rest of the paper is organized as follows. In Section 2, we give some preliminaries on concepts and definitions about lattices, low-density subset sum attacks, and simultaneous Diophantine approximation. The easy knapsack-type problems are presented in Section 3, as well as several examples to make the problems more understandable. The detailed description of the proposed PKCHD is given in Section 4. Section 5 discusses the performance related issues and specifies the parameter selection. Section 6 discusses several attacks on our system including key-recovery attacks, low-density attacks, and simultaneous Diophantine approximation attacks. The security of the system is carefully examined in this section. Section 7 gives some concluding remarks.

2. Preliminaries

Throughout this paper, the following notations will be used:

- $\mathbf{R}$, the field of real numbers.
- $\mathbf{Z}$, the ring of integers; $\mathbf{Z}^+$, the set of all positive integers.
- $\mathbf{Z}_n = \{0, \cdots, n-1\}$, the complete system of least nonnegative residues modulo n; $\mathbf{Z}_n^*$, the reduced residue system modulo n.
- $\gcd(a, b)$, the greatest common divisor of a and b; $\mathrm{lcm}(a, b)$, the least common multiple of a and b.
- If $\gcd(a, b) = 1$, $a^{-1} \bmod b$ denotes the inverse of a modulo b.
- $a|b$, a divides b.
- $a \bmod p$, the least nonnegative remainder of a divided by p.
- $a = b \bmod N$ means that a is the least nonnegative remainder of b modulo N; $a \equiv b \pmod N$ means that a and b are congruent modulo N.
- For $(a, b) \in (\mathbf{Z}^+)^2$, and an integer m, $m \bmod (a, b)$ denotes the 2-tuple ($m \bmod a$, $m \bmod b$).
- $u \not\equiv v \pmod{(a, b)}$ means that $u \bmod a \neq v \bmod a$ or $u \bmod b \neq v \bmod b$.
- $|A|$, the cardinality of a set A.
- $|a|_2$, the binary length of an integer a.
- $\lceil r \rceil$, the smallest integer greater than or equal to r.

Throughout this paper, we also adopt some customary parlance. For example, when we say a value is negligible, we mean that the value is a negligible function $v(k) : \mathbf{N} \mapsto [0, 1]$, i.e., for any polynomial $p(\cdot)$, there exists $k_0 \geq 1$ such that $v(k) < 1/p(k)$ for any $k > k_0$. The length of a vector means its norm (L_1, L_2 or L_∞ norm).

2.1. Lattice

A lattice is a discrete additive subgroup of $\mathbf{R}^n$. An equivalent definition is that a lattice consists of all integral linear combinations of a set of linearly independent vectors, i.e.,

$$L = \left\{ \sum_{i=1}^{d} z_i b_i \;\middle|\; z_i \in \mathbf{Z} \right\},$$

where $b_1, \cdots, b_d$ are linearly independent over $\mathbf{R}$. Such a set of vectors $\{b_i\}$ is called a lattice basis.

In the lattice theory, three important algorithmic problems are the shortest vector problem (SVP), the closest vector problem (CVP) and the smallest basis problem (SBP). The SVP asks for the shortest non-zero vector in a given lattice L. Given a lattice L and a vector v, the CVP is to find a lattice vector s minimizing the length of the vector $v - s$. Then, the SBP aims at finding a lattice basis minimizing the maximum of the lengths of its elements. The problems are of special significance in complexity theory and cryptology. The SVP can be approximated by solving SBP. No polynomial-time algorithm is known for the three problems. The best polynomial time algorithms to solve the SVP achieve only slightly sub-exponential factors, and are based on the LLL algorithm [25].

Before 1996, the lattice theory only applies to cryptanalysis [14,18–22,26–29], especially in breaking some knapsack cryptosystems. However, positive applications of the lattice theory in cryptology [30–33] have been witnessed in the last ten years. Some cryptographers even introduce the knapsack cryptosystems into the lattice-based cryptosystems due to the applications of lattice reduction algorithms in breaking the knapsack-type cryptosystems. For example, Sakurai [34] viewed the lattice-based cryptosystems as the revival of the knapsack trapdoors. More negative and positive applications of the lattice theory in cryptology can be found in [34,35].

The SVP and CVP are widely believed as difficult problems. However, interestingly, experimental results showed that lattice reduction algorithms behave much more nicely, especially in the low-dimensional (<300) lattices, than was expected from the worst-case proved bounds. When the dimension of a lattice is low, the lattice reduction algorithms can serve as a lattice oracle (SVP or CVP oracle). Therefore, to make a PKC invulnerable to lattice attacks, generally, the dimension is required to be sufficiently high (>500) without reducing the practicability, e.g., NTRU [32]. In this paper, a new method of constructing knapsack-type cryptosystem is presented. The dimension of the lattice underlying the cryptosystem is low (about 150), and it is still secure against lattice attacks under some reasonable assumptions.

2.2. Low-Density Subset Sum Attacks

Given a cargo vector $A = (a_1, \cdots, a_n)$ and an integer s, the 0–1 knapsack problem or more precisely the subset-sum problem is to determine a binary vector $X = (x_1, \cdots, x_n)$ such that the scalar product of A and X is s. More generally, we define the general knapsack problem or compact knapsack problem as to find a vector $X = (x_1, \cdots, x_n)$ with $x_i \in [0, 2^b - 1]$ such that

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

Note that Equation (1) is linear about the variable X. However, when the linearity restriction is removed and a new function f quadratic about X is defined such that $f(X) = s$, i.e., $XAX^T = \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}x_i x_j = s$, we call it a matrix cover problem. Especially when the matrix A is diagonal, $A = \mathrm{diag}(a_1, \cdots, a_n)$, the matrix cover problem turns out to find the vector $X = (x_1, \cdots, x_n)$ subject to $\sum_{i=1}^{n} a_i x_i^2 = s$. This problem is called a quadratic knapsack problem. These problems had been used to construct knapsack-type PKCs [4,7,12].

In a compact knapsack cryptosystem, the public key of the system is a cargo vector $A = (a_1, \cdots, a_n)$. A message $M = (m_1, \cdots, m_n)$ 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. A cryptosystem's density has a great effect on its security against lattice-based attacks such as low-density subset-sum attack and on whether it can be used to generate digital signatures for data origin authentication purposes. In a high density cryptosystem, almost all the messages can be signed. Informally, the density of a knapsack cryptosystem is defined as the fraction of the signable messages among all the messages [36], or the density is approximately the information rate, which is the ratio of the number of bits in plaintext message over the average number of bits in cipher-text [23]. Now, we provide the formal definition of density.

Definition 1 (Density [37])**.** *The density d of the compact knapsack problem (2) is defined by*

$$d = \frac{\sum_{i=1}^{n} e_i}{\log_2 \mathrm{C}_{\max}}, \tag{3}$$

where $\mathrm{C}_{\max} = k\sum_{i=1}^{n} a_i$ *is the maximum value of the cipher-text in the system and* $e_i = |m_i|_2 = \lceil \log_2(\mathrm{k}+1) \rceil$.

We want to give two remarks about the definition here. Firstly, $\lceil \log_2(k+1) \rceil$ bits are needed to represent the $k+1$ integers in $[0,k]$. Thus, we set $e_i = \lceil \log_2(k+1) \rceil$. Secondly, some different definitions can be found in the literature. For example, Orton [7] defined the density of Equation (2) as

$$d = \frac{n\lceil \log_2(k+1) \rceil}{\log_2 \max a_i}.$$

However, Ref. [37] gave a smaller density definition than that given in [7]. Thus, we adopt the smaller definition.

When the density d of a knapsack problem is too low, there exists an efficient reduction from the knapsack problem to the SVP over a lattice. Coster et al. [20] showed that, if $d < 0.9408\cdots$, which is the improvement of the earlier bound $0.6463\cdots$ [19], then the knapsack problem can be easily solved in a non-negligible probability with a single call to a lattice oracle.

Given a knapsack system $A = (a_1, \cdots, a_n)$ and a sum $s = \sum_{i=1}^{n} a_i x_i$; the basic idea of the low-density attack [20] runs as follows. The attacker constructs a matrix

$$V = \begin{pmatrix} 1 & 0 & \cdots & 0 & Na_1 \\ 0 & 1 & \cdots & 0 & Na_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & Na_n \\ \frac{1}{2} & \frac{1}{2} & \cdots & \frac{1}{2} & -Ns \end{pmatrix} = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \\ v_{n+1} \end{pmatrix}$$

at first using the public key, where $N > \sqrt{n}/2$. The integral combinations of the row vectors $v_1, \cdots, v_{n+1}$ of V form an $(n+1)$-dimensional lattice L. Suppose that $e = (e_1, \cdots, e_n)$ is a solution to $s = \sum_{i=1}^{n} a_i x_i$. Note that the vector

$$f = (f_1, \cdots, f_n, 0) = (e_1 + \frac{1}{2}, \cdots, e_n + \frac{1}{2}, 0) = e_1 v_1 + \cdots + e_n v_n + v_{n+1} \in L,$$

which contains enough information for the attacker to solve a solution to $s = \sum_{i=1}^{n} a_i x_i$. The length of f is relatively small. The short vector f can be found with non-negligible probability by using lattice basis reduction algorithms.

In fact, even if we design a knapsack system with the density close to 1 and $>0.9408\cdots$, we cannot claim that it is secure against low-density subset sum attacks. Let the length of the message vectors be bounded by r and $N(n,r)$ be the number of integral lattice points with length at most r in the n-dimensional sphere of radius r centered at the origin. Assume that the lattice points in the sphere have the same length and that the lattice reduction algorithms can find a lattice point in the sphere. Thus, the lattice point output by the lattice reduction algorithm is exactly the message vector with a probability $Pr = 1/N(n,r)$. However, if the density is slightly greater than $>0.9408\cdots$, $N(n,r)$ is bounded by a constant $O(1)$ or a polynomial function $O(p(n))$. In such a case, the probability $Pr = 1/N(n,r)$ is non-negligible. This is why Omura et al. [26] showed that the low-density attack can be applied to Chor–Rivest [5] and Okamoto–Tanaka–Uchiyama cryptosystems [38].

2.3. Simultaneous Diophantine Approximation

The simultaneous Diophantine approximation problem is a basic problem in Diophantine approximation theory, which has found uses both in cryptanalysis [22,28] and cryptography [39]. The problem is defined as follows.

Definition 2 (Simultaneous Diophantine approximation)**.** *The simultaneous Diophantine approximation problem is: given $n+1$ real numbers $r_1, \cdots, r_n, \epsilon > 0$, and an integer $Q > 0$, find integers $p_1, \cdots, p_n$ and $q : 0 < q \leq Q$, such that*

$$\left| r_i - \frac{p_i}{q} \right| \leq \frac{\epsilon}{q}.$$

Informally speaking, this problem asks for a set of fractions with a common and relatively small denominator approximating the given set of real numbers. There is a solution to the simultaneous Diophantine approximation problem if $Q \geq \varepsilon^{-n}$, but no efficient algorithm is found. However, when viewed as a problem involving lattices, the problem can be approximated by lattice basis reduction algorithms. Note that the integral linear combinations of the row vectors of the matrix

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

form a lattice L. Lattice basis reduction algorithms can be applied to the lattice L to output a reduced basis. The shortest vector b in the reduced basis can be used to approximate the simultaneous Diophantine approximation problem. Since $b \in L$, there exist integers $p_1, \cdots, p_n$ and q such that

$$b = \sum_{i=1}^{n} p_i a_i + q a_{n+1} = \left(p_1 - q r_1, \cdots, p_n - q r_n, \frac{q\epsilon}{Q} \right).$$

Since b is short, each $p_i - q r_i$ is small, which is equivalent to saying that $|r_i - p_i/q|$ is also small. Thus, $\{p_i/q\}$ is a set of fractions, with a common denominator q, approximating $\{r_i\}$. This informal demonstration reveals the relation between lattice reduction algorithms and the simultaneous Diophantine approximation problem.

3. Easy Knapsack-Type Problems

Knapsack-type PKCs always follows a common design morphology [9], that is:

- Construct an easy instance P[easy] from an intractable problem P.
- Shuffle P[easy] to make the resultant problem P[shuffle] seemingly-hard and indistinguishable from P.
- P[shuffle] is published as the encryption key. The information s by means of which P[shuffle] is reduced to P[easy] is kept as the secret key.
- The authorized receiver knowing s solves P[easy] to recover a message, whereas the task for the attacker is to solve P[shuffle].

In the knapsack public-key cryptography, several kinds of easy knapsack problems have been considered, e.g., super-increasing sequences [4], the cargo vectors used in the Graham–Shamir cryptosystem [40] and the knapsack sequences [41] used for attacking a knapsack-type cryptosystem [16] based on Diophantine equations. In this section, we propose several new easy knapsack problems, which can be viewed as the generalizations of those problems presented in [42,43].

3.1. An Easy Compact Knapsack Problem

Simultaneous compact knapsack problem is considered in this section: given the sums $(s_1, s_2) \in (\mathbf{Z}^+)^2$ and two cargo vectors $A = (a_1, \cdots, a_n)$, $B = (b_1, \cdots, b_n) \in (\mathbf{Z}^+)^n$, find a vector $X = (x_1, \cdots, x_n)$, such that $s_1 = \sum_{i=1}^n a_i x_i$, and $s_2 = \sum_{i=1}^n b_i x_i$. The problem has a solution only if $\gcd(a_1, \cdots, a_n) | s_1$ and $\gcd(b_1, \cdots, b_n) | s_2$. Without loss of generality, in this paper, we always assume that $\gcd(a_1, \cdots, a_n) = \gcd(b_1, \cdots, b_n) = 1$. The following theorem gives an easy instance of the simultaneous compact knapsack problem.

Theorem 1. *Given two cargo vectors $A = (a_1, \cdots, a_n)$ and $B = (b_1, \cdots, b_n)$. Denote by c_i and d_i the gcd of the first i components of A and B, respectively, i.e., $c_i = gcd(a_1, \cdots, a_i)$, $d_i = gcd(b_1, \cdots, b_i)$. If $2 \le k \le \lambda_i = lcm(c_{i-1}/c_i, d_{i-1}/d_i)$, $i = 2, \cdots, n$, the following simultaneous compact knapsack problem*

$$\sum_{i=1}^{n} a_i x_i = s_1, \tag{4}$$

$$\sum_{i=1}^{n} b_i x_i = s_2, \qquad 0 \le x_i \le k-1, \tag{5}$$

can be solved in polynomial (in n) time. Furthermore, the problem has at most one solution.

Proof. Note that $c_{n-1} | a_i, i = 1, \cdots, n-1$, so Equation (4) mod c_{n-1} gives $a_n x_n \equiv s_1 \pmod{c_{n-1}}$. Thus, we can invert a_n and obtain $x_n \equiv s_1 a_n^{-1} \pmod{c_{n-1}}$. Similarly, we get $x_n \equiv s_2 b_n^{-1} \pmod{d_{n-1}}$. Then, we can determine a unique $x_n \in \mathbf{Z}_{\lambda_n}$ according to CRT, where $\lambda_i = \mathrm{lcm}(c_{n-1}/c_n, d_{n-1}/d_n) = \mathrm{lcm}(c_{n-1}, d_{n-1}) \ge k$. If the unique x_n obtained is greater than $k-1$, we can conclude that the simultaneous compact knapsack problem has no solutions. Otherwise, we determine an x_n, $0 \le x_n \le k-1$.

Suppose that the values of $x_{i+1}, \cdots, x_n$, $i = n-1, \cdots, 2$ have been determined, then

$$\sum_{j=1}^{i} a_j x_j = s_1 - \sum_{j=i+1}^{n} a_j x_j, \tag{6}$$

and

$$\sum_{j=1}^{i} b_j x_j = s_2 - \sum_{j=i+1}^{n} b_j x_j. \tag{7}$$

Note that Equation (6) modulo c_{i-1} gives

$$a_i x_i \equiv s_1 - \sum_{j=i+1}^{n} a_j x_j \pmod{c_{i-1}}.$$

It is easy to verify that $\gcd(a_i, c_{i-1}) = c_i$ and $\gcd(a_i/c_i, c_{i-1}/c_i) = 1$. If $c_i \left| s_1 - \sum_{j=i+1}^n a_j x_j \right.$, we have

$$\frac{a_i}{c_i} x_i \equiv \frac{s_1 - \sum_{j=i+1}^n a_j x_j}{c_i} \bmod \frac{c_{i-1}}{c_i}; \tag{8}$$

otherwise, the simultaneous compact knapsack problems (4) and (5) have no solutions. By inverting a_i/c_i, we obtain according to Equation (8)

$$x_i \equiv \frac{s_1 - \sum_{j=i+1}^n a_j x_j}{c_i} \left(\frac{a_i}{c_i}\right)^{-1} \bmod \frac{c_{i-1}}{c_i}. \tag{9}$$

Similarly, we can deduce that problems (4) and (5) have no solutions or have a congruence

$$x_i \equiv \frac{s_2 - \sum_{j=i+1}^{n} b_j x_j}{d_i} \left(\frac{b_i}{d_i}\right)^{-1} \bmod \frac{d_{i-1}}{d_i}. \tag{10}$$

From (9) and (10), we can determine a unique $x_i \in \mathbf{Z}_{\lambda_i}$ according to the CRT, where $\lambda_i = \mathrm{lcm}(c_{i-1}/c_i, d_{i-1}/d_i) \geq k$. Thus, if (4) and (5) have solutions, we can determine a unique x_i: $0 \leq x_i \leq k-1$.

With the determined values of $x_2, \cdots, x_n$, we get

$$a_1 x_1 = s_1 - \sum_{j=2}^{n} a_j x_j \stackrel{\text{def}}{=} r_1,$$

and

$$b_1 x_1 = s_2 - \sum_{j=2}^{n} b_j x_j \stackrel{\text{def}}{=} r_2.$$

If $a_1 | r_1$ and $b_1 | r_2$, respectively, and the two quotients are identical, i.e.,

$$0 \leq \frac{r_1}{a_1} = \frac{r_2}{b_1} \stackrel{\text{def}}{=} r \leq k-1,$$

we set $x_1 = r$; otherwise, we deduce that the problems (4) and (5) have no solutions. Even if the unique values of $x_1, \cdots, x_n$ have been determined, we cannot claim that they are the solutions to (4) and (5). We need to verify whether $x_1, \cdots, x_n$ satisfy (4) and (5). If yes, then $X = (x_1, \cdots, x_n)$ is a solution to (4) and (5); otherwise, (4) and (5) have no solutions.

To determine each x_i, we need to solve two modular equations by using CRT. This problem can be solved only by computing $2n$ modular equations. Thus, the simultaneous compact knapsack problems (4) and (5) can be solved in polynomial (in n) time. If the problem has solutions, each x_i is uniquely determined according to CRT. Thus, the simultaneous compact knapsack problem has at most one solution. □

However, a high-density knapsack-type cryptosystem can not be designed based on this easy knapsack problem. It should be generalized in some other way.

3.2. Generalization of the Simultaneous Compact Knapsack Problem

Before generalizing the simultaneous compact knapsack problem, we first introduce some useful notations to make the discussion more convenient. Given $I \subset \mathbf{Z}$, $K \subset \mathbf{Z}^+$ and $J = \{j = (j_1, j_2) | j_1, j_2 \in \mathbf{Z}^+\}$, we use I^K to denote the set $\{i^k | i \in I, k \in K\}$. $\forall j = (j_1, j_2) \in J$, and $I^K \bmod j$ represents the set $\{i^k \bmod j = (i^k \bmod j_1, i^k \bmod j_2) | i \in I, k \in K\}$. Generally speaking, we have the following inequalities:

$$\forall j \in J, \qquad \left| I^K \bmod j \right| \leq \left| I^K \right| \leq |I| \times |K|.$$

The second "$\leq$" holds in that it is possible for different i_1, i_2 and k_1, k_2 to give an identical $i_1^{k_1} = i_2^{k_2}$, for example, $2^2 = 4^1$; of course, two different $i_1^{k_1}$ and $i_2^{k_2} \bmod j$ also can give rise to the same value.

Definition 3. *If $\forall j \in J$, $\left|I^K \bmod j\right| = \left|I^K\right| = |I| \times |K|$, we call set I a truly-distinguishable (T-DIST) modulo the set J under the indices of K; if $\forall j \in J$, $\left|I^K \bmod j\right| = \left|I^K\right| < |I| \times |K|$, we call the set I pseudo-distinguishable (P-DIST) modulo the set J under the indices of K; If $\exists j \in J$, $\left|I^K \bmod j\right| < \left|I^K\right|$, we call the set I indistinguishable (IND) modulo the set J under the indices of K. If different (i_1, k_1) and (i_2, k_2) result in the same $i_1^{k_1} \equiv i_2^{k_2} \pmod{j}$, we call the 3-tuples $((i_1, k_1), (i_2, k_2), j)$ a collision. In particular, the collisions in the case of P-DIST are called trivial collisions; The collisions in the case of IND are called non-trivial collisions.*

Theorem 2. *A set I is T-DIST (P-DIST, or IND respectively) modulo the set J under the indices of K iff I is T-DIST (P-DIST, or IND respectively) modulo the set J^T under the indices of K, where $J^T = \{(j_2, j_1) | (j_1, j_2) \in J\}$.*

Proof. It suffices to note that $\forall j = (j_1, j_2) \in J$, $i_1^{k_1} \bmod (j_1, j_2) = i_2^{k_2} \bmod (j_1, j_2)$ iff $i_1^{k_1} \bmod (j_2, j_1) = i_2^{k_2} \bmod (j_2, j_1)$. □

Consider the definitions, in the case of T-DIST, no collisions occur. Thus, given the $i^k \bmod j$, we can uniquely determine the corresponding (i, k). In the case of P-DIST, when a collision occurs, we only can determine a unique value r from $i^k \bmod j$. However, there exist at least two integer pairs (i_1, k_1) and (i_2, k_2) such that $i_1^{k_1} = i_2^{k_2} = r$. A collision occurs in the case of IND iff $(i_1, k_1) \neq (i_2, k_2)$, $i_1^{k_1} \neq i_2^{k_2}$ and $i_1^{k_1} \bmod j = i_2^{k_2} \bmod j$.

Theorem 3. *Given two cargo vectors $A = (a_1, \cdots, a_n)$, $B = (b_1, \cdots, b_n)$ and two sets $I, K \subset \mathbf{Z}^+$ with $|I|, |K| = O(1)$. Let c_i and d_i respectively denote the gcd of the first i components of A and B, and $J = \{(c_{i-1}/c_i, d_{i-1}/d_i) | i = 2, \cdots, n\}$. If I is T-DIST modulo the set J under the indices of K, the simultaneous Diophantine equations*

$$\sum_{i=1}^{n} a_i x_i^{k_i} = s_1, \qquad \sum_{i=1}^{n} b_i x_i^{k_i} = s_2, \tag{11}$$

with $x_i \in I$ and $k_i \in K$, can be solved in polynomial (in n) time. Furthermore, the problem has at most one solution in $X = (x_1, \cdots, x_n)$.

Proof. Note that $|I|, |K| = O(1)$, and we can construct a table of I Modulo J under the Indices of K in polynomial time. Its query operations can be carried out in polynomial time.

The proof of the theorem is analogous to that of Theorem 1. The only distinction is: in Theorem 1, we use CRT to determine a unique $x_i \in \mathbf{Z}_{\lambda_i}$; whereas, in Theorem 3, when we obtain a unique $x_i^{k_i} \bmod (c_{i-1}/c_i, d_{i-1}/d_i)$, we look up the table to construct and determine a unique x_i and $x_i^{k_i}$.

It can be concluded that, if the simultaneous Diophantine equations have solutions, there exists only one solution. The problem can be solved in polynomial (in n) time. □

Algorithm 1 formalizes the computational method of solving the simultaneous Diophantine Equation (11).

The requirement “T-DIST" is not necessary. In fact, if I is P-DIST modulo the set J under the indices of K, Theorem 3 and hence Algorithm 1 also work. In such a case, each $x_i^{k_i}$ is uniquely determined, whereas some values of x_i are not uniquely determined. Now, we give the following theorem.

Algorithm 1. Solving the simultaneous Diophantine equations

1 Construct a table $\mathcal{T}$ showing that I is T-DIST modulo J under the indices of K and store the table.
2 Compute $l_{1n} = s_1 a_n^{-1} \pmod{c_{n-1}}$, $l_{2n} = s_2 b_n^{-1} \pmod{d_{n-1}}$.
  1) Look up $\mathcal{T}$, decide an entry matching (l_{1n}, l_{2n}).
  2) If no, output "No Solutions" and exit;
  3) Otherwise, determine and store the values of x_n and $x_n^{k_n}$.
3 For $i = n-1, \cdots, 2$
  1) Decide whether c_i and d_i divide $r_{1i} = s_1 - \sum_{j=i+1}^{n} a_j x_j^{k_j}$ and $r_{2i} = s_2 - \sum_{j=i+1}^{n} b_j x_j^{k_j}$, respectively.
  2) If no, output "No Solutions" and exit;
  3) Otherwise, calculate $l_{1i} = \frac{r_{1i}}{c_i}\left(\frac{a_i}{c_i}\right)^{-1} \bmod \frac{c_{i-1}}{c_i}$, $l_{2i} = \frac{r_{2i}}{d_i}\left(\frac{b_i}{d_i}\right)^{-1} \bmod \frac{d_{i-1}}{d_i}$.
  If no entries in $\mathcal{T}$ match (l_{1i}, l_{2i}), exit with "No Solutions";
  Otherwise, determine and store the unique x_i and $x_i^{k_i}$.
4 Check whether $c_1 = a_1$ divides $r_{11} = s_1 - \sum_{j=2}^{n} a_j x_j^{k_j}$ and $d_1 = b_1$ divides $r_{21} = s_2 - \sum_{j=2}^{n} b_j x_j^{k_j}$ and $r_{11}/a_1 = r_{21}/b_1$
  1) If yes, set $x_1^{k_1} = \frac{r_{11}}{a_1} = \frac{r_{21}}{b_1}$;
  2) Otherwise, output "No Solutions" and exit.
  3) Solve x_1 from $x_1^{k_1}$, and store x_1 and $x_1^{k_1}$.
5 Decide whether $\sum_{i=1}^{n} a_i x_i^{k_i} = s_1$ and $\sum_{i=1}^{n} b_i x_i^{k_i} = s_2$.
  1) If yes, output $X = (x_1, \cdots, x_n)$ and exit;
  2) Otherwise, output "No Solutions" and exit.

Theorem 4. *Given two cargo vectors $A = (a_1, \cdots, a_n)$, $B = (b_1, \cdots, b_n)$ and two sets $I, K \subset \mathbf{Z}^+$ with $|I|, |K| = O(1)$. Denote by c_i and d_i the gcd of the first i components of A and B, respectively. Let $J = \{(c_{i-1}/c_i, d_{i-1}/d_i) | i = 2, \cdots, n\}$. If I is P-DIST modulo the set J under the indices of K, the simultaneous Diophantine equations*

$$\sum_{i=1}^{n} a_i x_i^{k_i} = s_1, \qquad \sum_{i=1}^{n} b_i x_i^{k_i} = s_2,$$

with $x_i \in I$ and $k_i \in K$, can be solved in polynomial (in n) time. Furthermore, it has at most one solution in $x_1^{k_1}, \cdots, x_n^{k_n}$.

4. The Proposed PKCHD Cryptosystem

This section derives the proposed PKCHD, a probabilistic knapsack-type cryptosystem. The public information consists of two sets $I, K \subset \mathbf{Z}^+$, $|I|, |K| = O(1)$, and $n \in \mathbf{Z}^+$, the dimension of a message vector. Let

$$\mu = \max i^k, \qquad i \in I \text{ and } k \in K. \tag{12}$$

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

4.1. Key Generation

Randomly choose two cargo vectors $A = (a_1, \cdots, a_n)$ and $B = (b_1, \cdots, b_n) \in (\mathbf{Z}^+)^n$, and denote by c_i and d_i the gcd of the first i components of A and B, respectively. Let $J = \{(c_{i-1}/c_i, d_{i-1}/d_i) | i = 2, \cdots, n\}$. The randomly-chosen A and B must satisfy the following condition:

Con: I is T-DIST modulo the set J under the indices of K.

Randomly choose two prime numbers $p \neq q$ such that

$$p \geq \mu \sum_{i=1}^{n} a_i, \qquad q \geq \mu \sum_{i=1}^{n} b_i. \tag{13}$$

Let $N = pq$. Compute the vector $E = (e_1, \cdots, e_n)$ according to CRT,

$$e_i \equiv a_i (\text{mod } p), \qquad e_i \equiv b_i (\text{mod } q). \tag{14}$$

Compute $w = e_n^{-1} (\text{mod } N)$. The public encrypting vector is $F = (f_1, \cdots, f_n) = (f_1, \cdots, f_{n-1}, 1)$ with each

$$f_i \equiv we_i (\text{mod } N). \tag{15}$$

The secret key consists of p, q and e_n. When decrypting a cipher-text, the receiver stores the values of c_i, d_i.

4.2. Encryption

Let $M = (m_1, \cdots, m_n)$, $m_i \in I$ be the message to be encrypted, and $G = (g_1, \cdots, g_n)$, $g_i \in K$ be a randomly chosen index vector. Using the public key F, cipher-text c is computed by

$$c = \sum_{i=1}^{n} f_i m_i^{g_i}. \tag{16}$$

4.3. Decryption

To decipher a cipher-text c, the receiver firstly computes s_p and s_q by

$$\begin{cases} s_p \equiv e_n c \equiv \sum_{i=1}^{n} e_i m_i^{g_i} \equiv \sum_{i=1}^{n} a_i m_i^{g_i} \, (\text{mod} p) \, , \\ s_q \equiv e_n c \equiv \sum_{i=1}^{n} e_i m_i^{g_i} \equiv \sum_{i=1}^{n} b_i m_i^{g_i} \, (\text{mod} q) \, . \end{cases} \tag{17}$$

From Equations (12) and (13), we know that

$$s_p = \sum_{i=1}^{n} a_i m_i^{g_i}, \qquad s_q = \sum_{i=1}^{n} b_i m_i^{g_i}. \tag{18}$$

According to the key generation algorithm and Theorem 3, we know that Equation (18) are easy simultaneous Diophantine equations. The receiver can recover the message M by solving Equation (18) according to Algorithm 1.

4.4. Remarks

Even though the parameter N is not an RSA integer, the system works. The "T-DIST" requirement for the cargo vectors A and B in **Con** is not necessary. In fact, if A and B meet the following requirement,

Con***:** I is P-DIST modulo the set J under the indices of K.

The cipher-text will not be uniquely deciphered. The sender can add some redundant information to the message vector so that the receiver can pick out the exact message from all the plaintexts he deciphers. Alternatively, both of them can agree on an encoding method by means of which the messages are encoded as plaintext vectors so that no collision occurs in all the encoded plaintext vectors.

4.5. A Practical Implementation

To implement the PKCHD in real-life practice, we choose $I = \{0, 1, \cdots, 7\}$, $K = \{1, 2, 3\}$ and $n = 150$. Thus, $\mu = \max i^k = 7^3 = 343$. Let W be a set consisting of the following pairs $(w_1, w_2) \in (\mathbf{Z}^+)^2$: (1,51), (1,65), (1,66), (2,33), (2,37), (2,39), (2,41), (2,43), (2,47), (3,17), (3,22), (3,25), (3,26), (3,29), (3,32), (4,23), (5,13), (5,16), (5,19), (6,11), (6,13), (7,11), (8,11), (9,11). We have the following theorem.

Theorem 5. *I is P-DIST modulo the set $J = W \cup W^T$ under the indices of K.*

Proof. According to Theorem 2, we only need to show that I is P-DIST modulo the set W under the indices of K, which can be proved by verifying that for every $(w_1, w_2) \in W$,

$$|I^K \bmod (w_1, w_2)| = |I^K| < |I| \times |K|.$$

Take (1,51) as an example,

$$I^K \bmod (1, 51) = \{(0, i) \big| i = 0, \cdots, 9, 16, 25, 27, 36, 49, 13, 23, 12, 37\}.$$

Thus, $|I^K \bmod (1, 51)| = |I^K| = 19 < |I| \times |K| = 24$. □

In fact, J gives all the 48 integer pairs $j = (u, v)$ with $uv < 100$ such that I is P-DIST modulo the set $\{(u, v)\}$ under the indices of $K = \{1, 2, 3\}$.

We randomly choose two cargo vectors $A = (a_1, \cdots, a_n)$ and $B = (b_1, \cdots, b_n)$ such that

$$(c_{i-1}/c_i, d_{i-1}/d_i) \in J = W \cup W^T, \qquad i = 2, \cdots, n,$$

where $c_i = \gcd(a_1, \cdots, a_i)$ and $d_i = \gcd(b_1, \cdots, b_i)$. According to Theorem 5, the generated vectors A and B meet the requirement of **Con***. We also generate RSA integers $N = pq$ with p, q primes and $p \geq 343 \sum_{i=1}^{n} a_i$, $q \geq 343 \sum_{i=1}^{n} b_i$. We compute the public vector F according to Equations (14) and (15).

The message M is split into $n = 150$ blocks with each block $m_i \in I$. When generating $G = (g_1, \cdots, g_n)$, we should note that, if $m_i = 2$, the corresponding $g_i \neq 2$. The cipher-text is computed as

$$c = \sum_{i=1}^{n} f_i m_i^{g_i}, \qquad m_i \in I \text{ and } g_i \in K. \tag{19}$$

The decryption is the same as Equations (17) and (18). However, if we compute $m_i^{g_i} = 4$, we should decipher m_i into 4 rather than 2. When confronted with some $m_i^{g_i} = 0$ or 1, we can uniquely determine $m_i = 0$ or 1 (Of course, g_i is not uniquely determined). Thus, the message can be uniquely recovered.

One observation that we also want to point out here is that the proposed implementation can be modified as a deterministic encryption algorithm. We can develop an encoding algorithm which encodes messages into an n-dimensional vector $Y = (y_1, \cdots, y_n)$ with every $y_i \in M^G = \{m_i^{g_i} \big| 0 \leq m_i \leq 7,\ 1 \leq g_i \leq 3\}$. In such a case, the decryption also works. After deciphering a cipher-text into a $Y \in (M^G)^n$, the receiver can decode Y to recover the message. Of course, the modification is of no special significance both in efficiency and for security. However, it will be very useful for us to discuss the low-density attacks on our system.

5. Performance and Parameter Specifications

This section specifies the parameter selection, analyzes the performance related issues, i.e., the key generation, the computational complexity of the encryption and decryption algorithms, the public key size and the information rate.

5.1. Parameter Specifications

p and q should be slightly greater than $\mu \sum_{i=1}^{n} a_i$ and $\mu \sum_{i=1}^{n} b_i$, respectively. When generating the public and secret keys, $|I|, |K| = O(1)$ is not necessarily required. However, this requirement does improve the efficiency of decryption. To decrypt a cipher-text, n table-query operations are needed by the receiver. If $|I|, |K| = O(1)$, the table only includes $|I| \times |K| = O(1)$ rows, which makes the table-query operations more efficient. In order to make the data sizes of the public and secret keys acceptable, we should require that $\forall i \in I, k \in K$, $|i|_2, |k|_2 = O(1)$. From Equations (12) and (13), we know that, if the lengths of i and k are relatively large, then the length of N and hence the lengths of the public and secret keys will be very large. It makes the proposed PKCHD system impractical.

If factoring the generated modulus N is hard, N can be published without compromising the security. However, if the sender knows N, he can encrypt a message vector M by

$$c = \sum_{i=1}^{n} f_i m_i^{g_i} (\text{mod } N), \tag{20}$$

which results in the reduction of the bit-length of the cipher-text. The public vector F can be permuted and re-indexed for increased security.

Remark. The public key size of the proposed system is about $(n-1)|N|_2$. Thus, the considerable public data size may be a burden for realizing the PKC. In fact, the public key of a PKC is stored in a certificate issued by the trusted third party. However, if the public key is too large, at the certificate, we can save a hashed value instead of the public key. To encrypt a message, the sender asks the intended receiver for the public key F. If the public key F' sent by the receiver matches the hashed value stored at the receiver's certificate, the sender conceives that the vector F' is exactly the public key F of the receiver and then uses it to encrypt the message. This method is suggested in [4] to compress the public key data size.

5.2. On Generating the Keys

Algorithm 2 generates the secret cargo vectors $A = (a_1, \cdots, a_n)$ and $B = (b_1, \cdots, b_n)$ subject to **Con***.

Algorithm 2. Generating the secret cargo vectors A, B

1 Given I and K, compute a set $J \subset (\mathbf{Z}^+)^2$ such that I is P-DIST modulo K under the indices of J.
2 Randomly choose n-1 integer pairs $(u_i, v_i) \in J$, $i = 1, \cdots, n$-1 with repetition permitted.
3 1) Randomly choose 2(n-1) numbers $s_2, \cdots, s_n$ and $t_2, \cdots, t_n$
with $\begin{cases} \gcd(s_i, u_j) = \gcd(t_i, v_j) = 1 \\ \gcd(s_i, s_{i+1}) = \gcd(t_i, t_{i+1}) = 1 \end{cases}$ for $i = 2, \cdots, n$-1
2) If $s_1 = t_1 = u_n = v_n = 1$, for $i = 1, \cdots, n$, we calculate $a_i = s_i \prod_{j=i}^{n} u_j$, $\quad b_i = t_i \prod_{j=i}^{n} v_j$
4 Output $A = (a_1, \cdots, a_n)$, $B = (b_1, \cdots, b_n)$.

Given I and K, the set J consisting of integer pairs can be generated by doing exhaustive computation for all the integer pairs (u, v) with the product uv bounded by a small constant (for example, 100). On the basis of Theorem 6, the generated vectors A and B really satisfy the requirement of **Con***.

Theorem 6. *Generated by Algorithm 2, the secret cargo vectors A and B are subject to* ***Con*****.*

Proof. Let c_i and d_i denote the gcd of the first i components of A and B, respectively. To prove that A and B are subject to **Con***, we only need to show that, for each $i = 2, \cdots, n$, the $(c_{i-1}/c_i, d_{i-1}/d_i)$ belong to the generated set J.

It is easy to verify that

$$\begin{aligned} c_i &= \gcd(a_1, \cdots, a_i) \\ &= \gcd\left(s_1 \prod_{j=1}^{n} u_j, \cdots, s_i \prod_{j=i}^{n} u_j\right) \\ &= \gcd\left(\prod_{j=1}^{n} u_j, \cdots, \prod_{j=i}^{n} u_j\right) = \prod_{j=i}^{n} u_j. \end{aligned}$$

Similarly,

$$d_i = \gcd(b_1, \cdots, b_i) = \prod_{j=i}^{n} v_j.$$

Therefore,

$$\left(\frac{c_{i-1}}{c_i}, \frac{d_{i-1}}{d_i}\right) = (u_{i-1}, v_{i-1}) \in J,$$

as desired. □

In Algorithm 2, s_i and t_i should be carefully chosen to guarantee that the generated a_i and b_i are not too large and always have the same binary length. For example, we can choose those s_i and t_i with lengths

$$|s_i|_2 = \left|\prod_{j=1}^{n} u_j\right|_2 - \left|\prod_{j=i}^{n} u_j\right|_2,$$

and

$$|t_i|_2 = \left|\prod_{j=1}^{n} v_j\right|_2 - \left|\prod_{j=i}^{n} v_j\right|_2.$$

Thus,

$$|a_i|_2 \approx |b_j|_2 \approx |b_1|_2 \approx |a_1|_2 \approx \left|\prod_{i=1}^{n-1} u_i\right|_2. \tag{21}$$

Note that p and q are slightly greater than $\mu \sum_{i=1}^{n} a_i = 343 \sum_{i=1}^{n} a_i$ and $\mu \sum_{i=1}^{n} b_i = 343 \sum_{i=1}^{n} b_i$, and that $u_i v_i < 100$. Then, for each f_i, the length is

$$\begin{aligned} |f_i|_2 &\approx |N|_2 \approx |p|_2 \cdot |q|_2 \approx |343 n a_1|_2 \cdot |343 n b_1|_2 \\ &\approx |343^2 n^2 a_1 \cdot b_1|_2 \approx 2|343n|_2 + \left|\prod_{i=1}^{n-1} u_i v_i\right|_2 \\ &< 2|343n|_2 + |100^{n-1}|_2 \\ &\approx 2|343n|_2 + (n-1)|100|_2, \end{aligned} \tag{22}$$

which is bounded by $O(n)$. If the selected (u_i, v_i) is uniformly distributed over the set $J = W \cup W^T$, the expected value of $u_i \cdot v_i$ is

$$u_i \cdot v_i \approx \sqrt[48]{\prod_{(u,v)\in J} uv} = \sqrt[24]{\prod_{(w_1,w_2)\in W} w_1 w_2} \approx 76.1.$$

Thus,

$$f_i \approx N \approx 343^2 \cdot n^2 \cdot 76.1^{n-1}. \tag{23}$$

The two estimations from Equations (22) and (23) are critical for examining the effects of the low-density subset sum attacks on the implementation of the proposed cryptosystem.

To defend against multiple transmission attacks, one way is frequently changing the secret/public keys. However, since the proposed PKCHD cryptosystem requires an RSA modulus, we prefer a slight modification to it in practical use. Here, we can randomly choose two coprime numbers p and q, calculate the modulus $N = pq$ and keep it secret. Notice that p and q are not necessarily primes.

5.3. Computational Complexity

In this section, we evaluate the computational complexity of the proposed PKCHD cryptosystem by analyzing the costs for encrypting a message and decrypting a cipher-text. Since the length of f_i is bounded by $O(n)$ (see Equation (22)), encrypting a message (Equation (16)) needs $n-1$ multiplications and additions, and n exponentiations. (1) Generally, the computation for the $n-1$ additions is inexpensive; (2) as pointed out earlier, the lengths of $m_i \in I$ and $g_i \in K$ are bounded by $O(1)$. It takes $O(n)$ bit operations to perform the n exponentiations. Naturally, the binary length of $m_i^{g_i}$ is also $O(1)$. (3) Meanwhile, $O(|f_i|_2 \times |m_i^{g_i}|_2) = O(n)$ bit operations are required to do the multiplication $f_i \times m_i^{g_i}$. Thus, the computational complexity for carrying out the $n-1$ multiplications is given by $O(n^2)$. Consequently, the computational complexity for message encryption is $O(n^2)$.

To decrypt a cipher-text, the receiver should do a modular multiplication in (17) and solve the easy simultaneous Diophantine equations in (18). For the modular multiplication, $O((|N|_2)^2) = O(n^2)$ bit operations are required. To solve the Diophantine Equations (18) for M, the receiver only needs $O(n)$ division, subtraction, multiplication and table-query operations. Generally, the $O(n)$ divisions and multiplications are the most costly. The bit lengths of the two integers involved in a division (or a multiplication) are respectively bounded by $O(n)$ and $O(1)$. Thus, the computational complexity for doing the $O(n)$ division, subtraction, multiplication and table-query operations is $O(n^2)$. Thence, the computational complexity of the decryption algorithm is also $O(n^2)$.

Compared with the traditional asymmetric encryption primitives RSA [2] and El Gamal [3], the proposed PKCHD cryptosystem has improvement in efficiency. For instance, both the encryption and decryption of the proposed PKCHD cryptosystem are only of quadratic bit complexity, whereas RSA [2] and El Gamal [3] reach cubic regarding the security parameter (If the length of the encryption exponentiation e of RSA is bounded by $O(1)$, for example, $e = 3$ or $2^{17}+1$, the encryption only performs $O(\log_2^2 N)$ bit operations). To make the comparison more concrete, we take the encryption of the proposed implementation, for example. If $n = 150$, from (23), we have

$$|f_i|_2 \approx \left|343^2 \cdot n^2 \cdot 76.1^{n-1}\right|_2 = 963.$$

Thus, about $(n-1)\,|f_i|_2 \left|m_i^{g_i}\right|_2 = 149 \cdot 963 \cdot 9 \approx 1.3 \times 10^6$ bit operations are required to finish the encryption. The computational cost is only about $1.3 \times 10^6 / 1024^2 \approx 1.24$ times that of a standard RSA-1024 modular multiplication.

5.4. Information Rate

The information rate ρ of a cryptosystem is defined as the ratio of the binary length of the message to that of the cipher-text. In the proposed PKCHD cryptosystem, the information rate turns out to be

$$\rho = \frac{3n}{\log_2 C_{\max}}.$$

We need to evaluate the binary length of $C_{\max}$. Note that

$$\begin{aligned} C_{\max} &= 343 \sum_{i=1}^{n} f_i \approx 343 \left[(n-1) f_1 + 1 \right] \\ &\approx 343 \, (n-1) \, f_1 \approx 343^3 \cdot (n-1) n^2 \cdot 76.1^{n-1}. \end{aligned} \tag{24}$$

Thus, the information rate is evaluated by

$$\rho \approx \frac{3n}{\log_2 \left[343^3 \cdot (n-1) n^2 \cdot 76.1^{n-1} \right]}.$$

When $n = 150$, the information rate ρ is about 0.46.

6. Security Analysis

Suppose that the attacker is trying to cryptanalyze the proposed PKCHD cryptosystem. Given a ciphertext c, the attacker has two methods to attack the proposed cryptosystem. The one is to solve the cracking problem [44], that is, determine the unique message vector $M = (m_1, \cdots, m_n)$ according to his knowledge about the public information and the enciphering function (16) such that (16) is satisfied for some small integers $g_1, \cdots, g_n$. The other method is to solve the trapdoor problem, that is, reverse the basic mathematical construction of the trapdoor in a PKC. If the attacker finds an efficient algorithm for the trapdoor problem, he will also have an algorithm for the cracking problem. This section investigates the hardness for the attacker to solve the cracking problem and the trapdoor problem. To make our discussion more concrete, we only consider the attacks on the implementation described in Section 4.

6.1. On Solving the Cracking Problem

6.1.1. Brute Force Attacks

One straightforward way to attack the system is to solve (19) for $M = (m_1, \cdots, m_n)$ directly. Let $M^G = \{m_i^{g_i} | 0 \le m_i \le 7, \ 1 \le g_i \le 3\}$. To determine whether (19) has a solution, and if so, to find it, the attacker can compute all the $\sum_{i=1}^{n} f_i m_i^{g_i}$ with $m_i^{g_i} \in M^G$. However, note that $\left| M^G \right| = 19$, so the brute force attack will take on the order of 19^n steps. A better method is to compute and sort each of the sets

$$S_1 = \left\{ \sum_{i=1}^{n/2} f_i m_i^{g_i} \middle| m_i^{g_i} \in M^G \right\}$$

and

$$S_2 = \left\{ c - \sum_{i=n/2+1}^{n} f_i m_i^{g_i} \middle| m_i^{g_i} \in M^G \right\},$$

and then scan S_1 and S_2, looking for a common element. If a common element $s = \sum_{i=1}^{n/2} f_i m_i^{g_i} = c - \sum_{i=n/2+1}^{n} f_i m_i^{g_i}$ is found, then $c = \sum_{i=1}^{n} f_i m_i^{g_i}$. The entire procedure takes on $n19^{n/2}$ steps [24]. For the proper parameters n, the attack is computationally infeasible.

6.1.2. Low-Density Attack

Low-density subset sum attacks only apply to a linear multivariate equation. Note that the encryption function (19) is nonlinear about the message vector M, so the low-density attacks cannot be used to cryptanalyze the proposed cryptosystem directly. The attacker can re-linearize the encryption function. By setting $y_i = m_i^{g_i} \in M^G$, the attacker obtains a linear function from the encryption function (19),

$$c = \sum_{i=1}^{n} f_i y_i, \quad y_i \in M^G. \tag{25}$$

Notice that the problem (25) is not a standard compact knapsack problem. Analogous to the case of the standard knapsack problem, the known best method for solving the problem (25) seems to be the "Brute Force Attacks" given by Ref. [24]. However, if the attacker wants to use low-density attacks to recover the corresponding message from a given cipher-text c, he cannot ensure that the solution to (25) belongs to M^G. The attacker can solve the problem (25) by solving the compact knapsack problem defined below,

$$c = \sum_{i=1}^{n} f_i y_i, \quad 0 \le y_i \le 343. \tag{26}$$

The attacker looks forward to finding a solution $Y = (y_1, \cdots, y_n)$ to (26) using the low-density attacks. Now we assume that the attacker has found such a solution Y to the compact knapsack problem (26). If every $y_i \in M^G$, then the attacker can simply solve n equations $y_i = m_i^{g_i}$ to recover the message M. Thus, we call the vector Y a message plaintext since it contains enough information about the message M. On the contrary, if there exists a $y_i \notin M^G$, then Y contains little information about M and hence is useless for the attacker to decipher the cipher-text. Because the vector Y is also a solution to (26), we call the vector Y a plaintext vector. In other words, in the relinearization attack model, we view the plaintext space as $\{0, \cdots, 343\}^n$ and the message plaintext space as $(M^G)^n$. The difference between the two sets $\{0, \cdots, 343\}^n - (M^G)^n$ is the redundant information added to the messages, or, equivalently, we pick out some elements as the message plaintexts from the whole plaintext space. This method has been used in the Chor–Rivest [5] and Okamoto–Tanaka–Uchiyama [38] schemes. In their schemes, only those vectors whose Hamming weight is exactly h are the message plaintexts.

Now, we begin to investigate the effects of the powerful low-density attacks on the security of the proposed PKCHD. When applied to a specific knapsack instance, the low-density attacks depend on the density of the knapsack. To estimate the density of the compact knapsack problem (26) using the definition of (3), we must evaluate all the $e_i = |m_i|_2$ and $C_{\max}$. The estimation of $C_{\max}$ is given in (24) and each $e_i = |m_i|_2 = \lceil \log_2 (343 + 1) \rceil = 9$, so the density is

$$d = \frac{9n}{\log_2 C_{\max}} \approx \frac{9n}{\log_2 \left[343^3 \cdot (n-1)n^2 \cdot 76.1^{n-1}\right]}. \tag{27}$$

If we choose $n = 150$, the density is about $1.38 > 0.9408 \cdots$.

If the public vector F is evaluated via (22), we can give the lower bound of the density. According to (22) and (24), we can evaluate

$$C_{\max} \approx 343\,(n-1)\,f_1 < 343^3 (n-1) n^2 100^{n-1}.$$

Thus, the density is lower-bounded by

$$d > \frac{9n}{\log_2 [343^3(n-1)n^2 100^{n-1}]}.$$

In the case of $n = 150$, the lower bound is about $1.3 > 0.9408\cdots$. If we adopt the definition of density given in [7], the estimation will be ever larger.

With an appropriate choice of the parameters, the PKCHD can obtain a high density even under the worst case scenario. However, we cannot claim its security against low-density subset-sum attacks only by an argument based on density. In the knapsack-type cryptographic history, so many cryptosystems have been broken by the powerful low-density attacks. Even those cryptosystems with high density such as Chor–Rivest [5] and Okamoto–Tanaka–Uchiyama [38] schemes were also shown to be vulnerable to low-density attacks [26,27]. Thus, we must be cautious to claim the proposed PKCHD's security against the low-density attacks. Other lattice-based attacks on the system also need to be well examined. If we have shown that the proposed cryptosystem is invulnerable to the known lattice attacks, we think that the security of the cryptosystem against the lattice-reduction-based attacks should be convincing.

6.1.3. On the Number of Plaintext Vectors That a Cipher-Text Has

The low-density subset-sum attacks always assume that the practical lattice reduction algorithms can serve as an SVP oracle at least in the cases of low-dimensional lattices. In fact, lattice reduction algorithms perform well in practice, and some current experimental records can be found in [27]. Thus, we assume that lattice reduction algorithms can obtain the shortest vector in a lattice with low dimension. Meanwhile, another fact is that the encryption function of the proposed PKCHD is non-injective under the relinearization attack model. Thence, for a given cipher-text c, $0 \le c \le 343\sum_{i=1}^{n} f_i$, there are many preimages Y such that (26) is satisfied. The lengths of the preimages are bounded by the length r of the vector $Y_{\max} = (343, \cdots, 343)$. Thus, all the preimages are the lattice points in the n-dimensional sphere of radius r centered at the origin. The number $N(n, r)$ of the lattice points in the sphere is exactly the number of the preimages corresponding to a given cipher-text c. Furthermore, all the preimages almost have the same length. No evidence shows that the message is the shortest vector among all the plaintext vectors. In fact, Refs. [42,43] have given a small example in which the message plaintext is not the shortest vector no matter what norms are used. Thus, the lattice reduction algorithms just find a random vector in the $N(n, r)$ preimages. We use an assumption to formalize what we have discussed.

Unif: Given a cipher-text c, the vector output by the lattice reduction algorithms is uniformly distributed over the $N(n, r)$ plaintext vectors.

Theorem 7. *Under the assumption **Unif**, the probability δ of the lattice algorithms finding out the message vector is negligible.*

Proof. Based on the assumption **Unif**, we can conclude that $\delta = 1/N(n, r)$. Therefore, $N(n, r)$ needs to be evaluated. Since Ref. [27] presented the estimation of the upper bound of $N(n, r)$, to complete this proof, the lower bound is required. Notice that the expected number $N(n, r)$ should be the ratio of the number of all the plaintext vectors to that of the possible cipher-texts, i.e.,

$$\begin{aligned} N(n, r) &\approx \frac{344^n}{343\sum_{i=1}^{n} f_i + 1} \approx \frac{344^n}{C_{\max}} \\ &\approx \frac{344^n}{343^3 \cdot (n-1)n^2 \cdot 76.1^{n-1}} > 2^n, \end{aligned}$$

for sufficiently large n. Obviously,

$$\delta = \frac{1}{N(n,r)} < \frac{1}{2^n}$$

is negligible. □

The evaluation of the number of the preimages that a cipher-text has is somewhat rough. However, it suffices to show the non-injectivity of the encryption function under the relinearization attack model. Thence, another way of evaluating the number of the preimages is presented. Note that any vector $Y \in \{0, 1, \cdots, \cdots, 343\}^n$ satisfying (26) must be a solution to the modular knapsack problem defined below,

$$c = \sum_{i=1}^{n} f_i y_i \pmod N, \quad 0 \le y_i \le 343.$$

It is easy to verify that this problem is equivalent to the following simultaneous compact knapsack problem,

$$ce_n \pmod p = \sum_{i=1}^{n} a_i y_i, \quad ce_n \pmod q = \sum_{i=1}^{n} b_i y_i.$$

To solve the problem, the method given in Theorem 1 is preferred. According to CRT, a unique y_i modulo $\lambda_i = \text{lcm}(c_{i-1}/c_i, d_{i-1}/d_i)$ can be determined. However, since $\lambda_i = \text{lcm}(c_{i-1}/c_i, d_{i-1}/d_i) = \text{lcm}(u_{i-1}, v_{i-1}) \le u_{i-1}v_{i-1} < 100$ and $0 \le y_i \le 343$, we can determine at least three values for each y_i. Finally, there are at least 3^n vectors $Y = (y_1, \cdots, y_n)$ for which a given cipher-text c can be determined. Of course, not all the vectors are the solutions to (26). However, even if a small amount of the vectors satisfy (26), it suffices to show that a given cipher-text c has exponentially many plaintext vectors.

Now, a small example (see Table 1) is used to illustrate what we have discussed. To simplify the discussion, we set $I = \{0, 1, 2, 3\}$, $K = \{1, 2, 3\}$, and $n = 9$. In this case, the cipher-text $c = 44190990551868$ has ten preimages Ys under the relinearization attack model. However, there exists only one message plaintext vector $Y_1 = (4, 27, 3, 27, 2, 27, 0, 1, 4)$ amongst all the ten preimages. The left nine preimages $Y_2, \cdots, Y_{10}$ are the plaintext vectors. Thus, we conclude that the low-density subset sum attack will find the message plaintext vector Y_1 with a probability $\delta = \frac{1}{10}$ under the assumption **Unif**. Additionally, the message plaintext vector Y_1 is not the shortest non-zero vector in the lattice involved in the low-density subset sum attack no matter what norms are used. If we use (20) to encrypt the message, the encryption function

$$c = \sum_{i=1}^{9} f_i y_i \pmod N = 192662536160, \quad 0 \le y_i \le 27$$

even has 237 preimages in all, which are not listed in Table 1 for space limitations. In this case, the parameter n is too small to achieve practical security. However, if a relatively large n (e.g., 150) is chosen, the number of the preimages of a given cipher-text will be very large. This is what we have claimed in the proof of Theorem 7.

Table 1. The non-injectivity of the encryption function under the relinearization attack model.

I	$\{0,1,2,3\}$
K	$\{1,2,3\}$
μ	27
n	9
A	10000, 6000, 7000, 5800, 5300, 5840, 8210, 6662, 5113
B	10000, 5000, 8000, 5500, 5100, 6150, 5830, 5335, 6007
p	999979
q	999983
N	999962000357
E	10000, 250000750, 999712012607, 75004225, 50004250, 499903507646, 594995715, 750303249963, 499757509985
e_9^{-1}	759237254392
F	661037209656, 7824090728, 451539481682, 866739311295, 192593114076, 586570143338, 753328582077, 356431315295, 1
M	$(2,3,3,3,2,3,0,1,2)$
G	$(2,3,1,3,1,3,2,3,2)$
c	44190990551868
Y	$(4,27,3,27,2,27,0,1,4)$, $(10,5,12,19,19,7,10,1,4)$ $(5,12,9,13,9,27,10,1,4)$, $(18,6,4,25,13,4,0,11,4)$ $(13,13,1,19,3,24,0,11,4)$, $(5,8,19,27,4,1,0,21,4)$ $(2,0,15,27,24,1,0,21,4)$, $(1,0,22,7,1,21,10,21,4)$ $(2,3,16,8,1,21,21,1,14)$, $(3,2,5,23,0,12,12,11,24)$

6.1.4. On Reducing to the CVP

Nguyen and Stern [27] found that the knapsack problem also can be reduced to the CVP. Note that the solutions of

$$\sum_{i=1}^{n} z_i f_i = 0 \tag{28}$$

form an $(n-1)$-dimensional linear space over $\mathbf{R}$. Thus, the integral solutions of (28) form an $(n-1)$-dimensional lattice L. Given a cipher-text c, we can compute by using an extended Euclidean algorithm integers $x_1, \cdots, x_n$ such that $c = \sum_{i=1}^{n} x_i f_i$. Let $Y = (y_1, \cdots, y_n)$ be a plaintext vector (not necessarily the message plaintext vector). Then the vector $u = (x_1 - y_1, \cdots, x_n - y_n)$ belongs to L such that

$$\sum_{i=1}^{n}(x_i - y_i)f_i = \sum_{i=1}^{n} x_i f_i - \sum_{i=1}^{n} y_i f_i = c - c = 0.$$

In addition, u is fairly close to the vector $X = (x_1, \cdots, x_n)$. Thus, the closest vector $u \in L$ to X is expected to be found by accessing the CVP-oracle. Thus, $X - u$ is a plaintext vector. However, we should observe that the success probability of the reduction depends on the number $N(n,r)$ of integer points in the $(n-1)$-dimensional spheres. According to Theorem 7, we can conclude that the closest vector output by the CVP-oracle is the exact message plaintext vector with a negligible probability.

Furthermore, the cryptanalysis of low-weight knapsacks [26,27] does not compromise the security of the system in which the low-weight vectors are not selected as message vectors. Until now, it is safe to claim the security of the cryptosystem against the known lattice-based attacks including low-density subset-sum attacks.

6.2. On Solving the Trapdoor Problem

When we discuss the cracking problem, we only consider the infeasibility of the attacker's solving (19) regardless of the structure of the public vector $F = (f_1, \cdots, f_n)$. In other words, the public vector $F = (f_i, \cdots, f_n)$ is considered to be indistinguishable from a randomly generated n-dimensional vector. However, (19) is only a seemingly-hard compact knapsack problem. If the public key reveals enough information for the attacker to reverse the basic mathematical construction of the trapdoor in the proposed PKCHD system, then he also can serve as an authorized receiver to decipher any cipher-text. Thus, the key recovery attacks on the cryptographic scheme also need to be carefully studied.

6.2.1. Simultaneous Diophantine Approximation Attack

Most of the knapsack-type cryptosystems use size conditions to disguise an easy knapsack problem. The designer randomly generates an easy knapsack problem, $y = \sum_{i=1}^{n} a_i x_i, x_i \in [0, 2^b - 1]$, and chooses a modulus m and a multiplier w, $\gcd(m, w) = 1$. He uses the size condition $m > (2^b - 1)\sum_{i=1}^{n} a_i$ to disguise the easy cargo vector $A = (a_1, \cdots, a_n)$ as a seemingly-hard knapsack sequence $B = (b_1, \cdots, b_n)$, $b_i = wa_i (\text{mod } m)$. The size condition can be utilized by the simultaneous Diophantine approximation attack to obtain some useful information about (w, m). See [22,28] for more information about the relationship between the simultaneous Diophantine approximation problem and cryptanalytics.

The trapdoor of the proposed PKCHD system is disguised using CRT, which involves no size conditions. Thus, launching a simultaneous Diophantine approximation attack cannot find valuable information about the trapdoor. Even though the size condition has been used in (13), the attacker must peel off the outmost shuffle in (14) and (15) if he wants to launch a simultaneous Diophantine approximation attack. Unfortunately, it is also a difficult task.

6.2.2. Known N Attack

The exact value of N is assumed to be known by the attacker, and he wants to learn some information about the secret key. A straightforward way is to search for e_n and factor N to recover the trapdoor information. To evaluate to what extent the attacker can succeed, we must decide whether the public key $F = (f_1, \cdots, f_n)$ and N provide the attacker with enough information to compromise the cryptosystem. If the public vector F is indistinguishable from a random-chosen n-dimensional vector F^* over $\mathbf{Z}_N$ (In fact, only the first $n - 1$ components of F^* are randomly chosen, and the last components of F^* must be 1. Otherwise, it makes no sense to say that the public vector F is indistinguishable from a random-chosen n-dimensional vector in that $f_n = 1$). We can conclude that the public key F and N provide no useful information for the attacker to recover the secret key. In other words, it is impossible for the attacker to retrieve the integer $e_n \in \mathbf{Z}_N$ from a random n-dimensional vector F.

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 are smooth integers (i.e., it only contains small prime factors), whereas a random integer may not be. However, the public vector F is scrambled by (14) and (15). At the same time, the smoothness of the two vectors A and B is also disguised. After the two shuffles (14) and (15), the only distinction disappears. Then, the generated vector F must be indistinguishable from those random n-dimensional vectors over $\mathbf{Z}_N$. Thus, the publication of N will not affect the security of the system. On the contrary, it will reduce the length of the cipher-text and improve on the transmitting efficiency.

The attacker cannot expect to recover the secret key by searching for the integer e_n to make all the $a_i = f_i e_i (\text{mod } p)$ and $b_i = f_i e_i (\text{mod } q)$ smooth simultaneously, where $i < n$ is a relatively small integer. In fact, the best way of retrieving the trapdoor seems to factor N at first and then recover the secret vectors A and B. It is easy to verify that $a_n w \equiv 1 (\text{mod } p)$ and $b_n w \equiv 1 (\text{mod } q)$, where $w = e_n^{-1} (\text{mod } N)$. If

we write a_n^{-1} and b_n^{-1} for the inverse of $a_n (\text{mod } p)$ and $b_n (\text{mod } q)$ respectively, and set $f_{ip} = f_i (\text{mod } p)$, $f_{iq} = f_i (\text{mod } q)$, $i = 1, \cdots, n-1$, (15) modulo p and q result in

$$f_{ip} \equiv a_n^{-1} a_i (\text{mod } p), \qquad f_{iq} \equiv b_n^{-1} b_i (\text{mod } q).$$

Note that the vectors A and B are of some special structure. Therefore, if the modulus N is factored, the attackers will get some useful information from the integers f_{ip} and f_{ip}. To examine the potential threats against the proposed PKCHD cryptosystem, we consider a stronger assumption, that is, the attacker had factorized the modulus N.

6.2.3. Known p and q Attack

Now, we consider such a scenario that the attacker has factorized the modulus $N = pq$. It is easy for the attacker to compute the f_{ip}'s and f_{iq}'s. Then, for the attacker, the left task is just to recover a_n and b_n in that other a_i and b_i can be easily reconstructed via

$$a_i \equiv a_n f_{ip} (\text{mod } p), \qquad b_i \equiv b_n f_{iq} (\text{mod } q).$$

In addition, the gcd's c_i and d_i are easily determined by using the Euclidean algorithm. Thus, the secret key is recovered.

(a) Structural attack: In fact, if the attacker obtains two pairs (a_i, f_{ip}) and (b_j, f_{jq}), he can determine the exact values of a_n and b_n. Note that a_1 and b_1 have special structures (See Algorithm 2). If the attacker wants to launch a structural attack, i.e., he does exhaustive search for all the possible integer pairs (a_1, b_1). Assume $n = 150$, the $n-1$ integer pairs (u_i, v_i) are randomly chosen with repetition permitted such that $(u_i, v_i) \in J = W \cup W^T$. For each i, (u_i, v_i) takes 48 possible values. Then, the number of possible choices for the pair (a_1, b_1) is given in the following theorem.

Theorem 8. *When $n = 150$, the number t of choices for generating (a_1, b_1) is $t = \binom{197}{47}$.*

Proof. If we denote the set $J = \{j_i | i = 1, \cdots, 48\}$ and look at each j_i as an apple with color i, then we are confronted with such an "apple" probability model: choose $n = 150$ apples from the 48 color of apples with repetition permitted.

Now, we consider a line on which 197 dots are scattered. We choose 47 dots among the 197 dots and view them as boards. We denote the 47 boards as b_i, $i = 1, \cdots, 47$ from left to right. The dots on the left of b_1 are the apples with color 1, and the dots on the right of b_{47} are the apples with color 48. These dots between board i and board $i+1$ are the apples with color $i+1$, for $i = 1, \cdots, 46$. Thus, every choice of the 47 board corresponds to a choice of the integer pair (a_1, b_1). We have $t = \binom{197}{47}$ choices in total. Thus, we complete the proof. □

Since $t = \binom{197}{47} \approx 2^{1025}$, apparently, it is computationally infeasible for the attacker to try all the possibilities.

(b) Simultaneous Diophantine approximation attack: Without loss of generality, we let

$$a_n f_{ip} - l_i p = a_i, \qquad i = 1, \cdots, n-1. \tag{29}$$

Divide the both sides of (29) by pa_n, and we obtain

$$\frac{f_{ip}}{p} - \frac{l_i}{a_n} = \frac{a_i}{pa_n}. \tag{30}$$

Note that $p \approx 343 \sum_{j=1}^{n} a_j \approx 343 n a_i \approx 343 n \sqrt{76.1^{n-1}}$. Thus, we have

$$\left| \frac{f_{ip}}{p} - \frac{l_i}{a_n} \right| = \frac{a_i}{p a_n} \approx \frac{1}{p} \approx \frac{1}{343 n \sqrt{76.1^{n-1}}}$$

from (21), (23) and (30). If we note again that $a_n \approx p/(343n)$, we can claim that $\{l_i/a_n\}$ is a set of fractions with a common and relatively small denominator a_n approximating the set of fractions $\{f_{ip}/p\}$. More formally, we can assume that these fractions l_i/a_n are the simultaneous Diophantine approximations of the fractions f_{ip}/p. If there is an efficient algorithm to solve the problem, the attacker can retrieve the secret vector $A = (a_1, \cdots, a_n)$. Using a similar method, he also can recover the vector $B = (b_1, \cdots, b_n)$. Thus, the gcd's c_i and d_i are also obtained.

Since the simultaneous Diophantine approximation problem is a widely-believed intractable problem, no efficient algorithm has been found for it. From the discussion above, it can be deduced that, to reconstruct the secret key, the attacker must search for the modulus N and then solve two hard number-theoretic problems, namely the integer factorization problem and the simultaneous Diophantine approximation problem. This is a property shared with the scheme presented in [39].

6.3. Generating the Hardest Knapsack Instances

It is general knowledge that the whole public key cryptography is based on the computational complexity theory. We may hope that the PKCs based on proven intractability assumptions, e.g., the knapsack problem, are unbreakable super-codes. However, the fact is not the case; many PKCs based on the NP-complete problems such as the knapsack problem and the multivariate quadratic polynomials [45] had been shown insecure. Fortunately, some PKCs based on unproven mathematics' assumptions remain unbroken. Following the work of [45], this phenomena can be explained as follows. The security of some of the integer-factorization-based PKCs or the discrete-logarithm-based PKCs is based not only on the hardness of factoring an integer or solving the discrete logarithm problem defined over some cyclic groups, but also on the key generation algorithms. For example, it may not be a difficult thing for factoring a randomly-chosen large integer in that the integer always contains some small prime factors. However, the RSA system does not use such easy-to-factor integers, and it always can select the hardest factorization problem as the basis for its security. The knapsack problem is shown to be NP-complete, but the computational complexity only deals with the worst-case complexity. If the use of the hardest knapsack instances is excluded in public key cryptography, we cannot expect a knapsack cryptosystem to be an unbreakable super-code. In fact, the knapsack problems with density $<0.9408\cdots$ is shown easy to solve [20]. Many cryptographers have pointed out that the knapsack instances with density greater than 1 cannot be used in public key cryptography in that the cipher-texts are not uniquely decipherable. Relatively, the room left for designing a secure knapsack cryptosystem is narrow. Further discussion about the relationship between knapsack cryptography and computational complexity refers to [36].

Schnorr and Euchner [29] had shown that the hardest knapsack instances are those with density $d \approx 1 + \log_2(n/2)/n$, which is slightly larger than 1. The density of the proposed PKCHD is given in (27). When n approaches infinity,

$$\lim_{n\to\infty} \frac{9n}{\log_2 [343^3 \cdot (n-1) n^2 \cdot 76.1^{n-1}]} = \frac{9}{\log_2 76.1} \approx 1.44,$$

and

$$\lim_{n\to\infty} \left(1 + \frac{\log_2(n/2)}{n} \right) = 1.$$

Thus, for a sufficiently large n, we always have

$$\frac{9n}{\log_2\left[343^3 \cdot (n-1)n^2 \cdot 76.1^{n-1}\right]} > 1 + \frac{\log_2(n/2)}{n}.$$

In other words, the proposed PKCHD cryptosystem always can use a knapsack problem with density $d > 1 + \log_2(n/2)/n$ as the encryption function. To generate the hardest knapsack problem, the cryptosystem can generate two larger primes p and q to make the density $d \approx 1 + \log_2(n/2)/n$.

To make a knapsack problem be the hardest, the cargo vector should be indistinguishable from the random vectors. In fact, we have shown that the public vector of the PKCHD system is indistinguishable from a randomly-chosen vector. Consequently, if the hardness of a knapsack instance is evaluated by its density, the PKCHD system always can use the hardest knapsack vector as the public key.

6.4. Provable Security Remarks

In public key cryptography, two typical methods are employed for security analysis. One is the provable security theory [46], the basic idea is to reduce the security of a PKC under some attack model to a mathematical hard problem. The other is to deliver the PKC to the cryptological community for attacks that is called enumerative security. Provable security has been widely accepted as a standard method for the security analysis of PKCs. However, due to the following considerations, in this study, we do not prefer provable security results about the proposed PKCHD cryptosystem. Firstly, we should note that almost all the provably secure PKCs are constructed from the number-theoretic problems, i.e., integer factorization and discrete logarithm problems. Secondly, provable security theory is not suitable for analyzing the security of those PKCs based on NP-complete problems. These PKCs are always constructed from an easy problem. Actually, the problem of reversing the encryption functions is only a seemingly-hard rather than a truly hard problem. It makes no sense to reduce the security of a PKC to a seemingly-hard problem. Thirdly, security analysis for a newly-designed trapdoor one-way function should be centered on the estimation of the hardness of reversing the encryption function and retrieving the trapdoor information. If no efficient algorithms have been found for a long time to compromise its security, we can assume its one-wayness and begin to consider adding paddings to it to make it obtain provable security objectives.

It will be a significant theoretical result if one can prove that reversing the encryption function is equivalent to solving the mathematical problems used in constructing the PKC. However, this is an extremely tough task [44].

7. Conclusions

Due to the performance advantages over other cryptosystems, the knapsack cryptosystems, as a typical class of PKCs, plays an important role in the wide variety of available cryptosystems. Especially, new knapsack-type cryptographic primitives have been developed in recent years, e.g., the non-injective knapsack cryptosystems [47], the knapsack Diffie–Hellman problem [48], and elliptic curve discrete logarithm based knapsack public-key cryptosystem [49].

In this paper, a probabilistic knapsack-type PKC, namely PKCHD, which uses CRT to disguise the easy knapsack sequence has been constructed with careful security analysis. Fortunately, no practical attacks have been found to comprise the PKCHD's security. However, the history that almost all additive knapsack-type cryptosystems were shown to be vulnerable to some attacks makes the designers confident. Thus, some novel attacks are to be investigated to make it more secure.

Author Contributions: Conceptualization, Y.P. and B.W.; methodology, Y.P. and B.W.; validation, Y.P., B.W., S.T. and J.Z.; formal analysis, Y.P., B.W. and J.Z.; investigation, S.T. and H.M.; resources, S.T. and H.M.; writing–original draft preparation, Y.P.; writing–review and editing, Y.P. and B.W.; supervision, B.W.; project administration, Y.P. and B.W.; funding acquisition, Y.P. and B.W.

Funding: This work is supported by the National Key R&D Program of China under Grant No. 2017YFB0802000, the National Natural Science Foundation of China under Grant No. U1736111, the Plan For Scientific Innovation Talent of Henan Province under Grant No. 184100510012, the Program for Science and Technology Innovation Talents in the Universities of Henan Province under Grant No. 18HASTIT022, the Key Technologies R&D Program of Henan Province under Grant No. 182102210123 and 192102210295, the Foundation of Henan Educational Committee under Grant No. 16A520025 and 18A520047, the Foundation for University Key Teacher of Henan Province under Grant No. 2016GGJS-141, the Open Project Foundation of Information Technology Research Base of Civil Aviation Administration of China under Grant No. CAAC-ITRB-201702, and the Innovation Scientists and Technicians Troop Construction Projects of Henan Province.

Acknowledgments: The authors would like to thank the anonymous reviewers for their carefulness and patience, and thank Sheng Tong for the proof of Theorem 8 and Fagen Li for paper preparation.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses or interpretation of data; in the writing of the manuscript or in the decision to publish the results.

References

1. Diffie, W.; Hellman, M.E. New Directions in Cryptography. *IEEE Trans. Inf. Theory* **1976**, *IT-22*, 644–654. [CrossRef]
2. Rivest, R.L.; Shamir, A.; Adleman, L.M. A Method for Obtaining Digital Signature and Public Key Cryptosystems. *Commun. ACM* **1978**, *21*, 120–126. [CrossRef]
3. ElGamal, T. A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. *IEEE Trans. Inf. Theory* **1985**, *IT-31*, 469–472. [CrossRef]
4. Merkle, R.C.; Hellman, M.E. Hiding Information and Signatures in Trapdoor Knapsacks. *IEEE Trans. Inf. Theory* **1978**, *IT-24*, 525–530. [CrossRef]
5. Chor, B.; Rivest, R.L. A Knapsack-Type Public Key Cryptosystem Based on Arithmetic in Finite Fields. *IEEE Trans. Inf. Theory* **1988**, *IT-34*, 901–909. [CrossRef]
6. Vaudenay, S. Cryptanalysis of The Chor–Rivest Cryptosystem. *J. Cryptol.* **2001**, *14*, 87–100. [CrossRef]
7. Orton, G. A Multiple-Iterated Trapdoor for Dense Compact Knapsacks. In *Advances in Cryptology–Eurocrypt 1994 (LNCS)*; Springer-Verlag: Perugia, Italy, 1995; Volume 950, pp. 112–130.
8. Morii, M.; Kasahara, M. New Public Key Cryptosystem Using Discrete Logarithm Over $GF(p)$. *IEICE Trans. Fund.* **1988**, *J71-D*, 448–453.
9. Naccache, D.; Stern, J. A New Public-Key Cryptosystem. In *Advances in Cryptology–Eurocrypt 1997 (LNCS)*; Springer-Verlag: Konstanz, Germany, 1997; Volume 1233, pp. 27–36.
10. Goodman, R.M.F.; McAuley, A.J. New Trapdoor-Knapsack Public-Key Cryptosystem. *IEE Proc.* **1985**, *132 Pt E*, 282–292.
11. Niemi, V. A New Trapdoor in Knapsacks. In *Advances in Cryptology–Eurocrypt 1990 (LNCS)*; Springer-Verlag: Aarhus, Denmark, 1990; Volume 473, pp. 405–411.
12. Janardan, R.; Lakshmanan, K.B. A Public-Key Cryptosystem based on The Matrix Cover NP-Complete Problem. In *Advances in Cryptology–Crypto 1982*; Plenum: New York, NY, USA, 1983; pp. 21–37.
13. Blackburn, S.R.; Murphy, S.; Stern, J. Weaknesses of A Public Key Cryptosystem based on Factorization of Finite Groups. In *Advances in Cryptology–Eurocrypt 1993 (LNCS)*; Springer-Verlag: Lofthus, Norway, 1994; Volume 765, pp. 50–54.
14. Nguyen, P.; Stern, J. Merkle-Hellman Revisited: A cryptanalysis of The Qu-Vanstone Cryptosystem based on Group Factorizations. In *Advances in Cryptology–Crypto 1997 (LNCS)*; Springer-Verlag: Santa Barbara, CA, USA, 1997; Volume 1294, pp. 198–212.

15. Pieprzyk, J.P. On Public-Key Cryptosystems Built Using Polynomial Rings. In *Advances in Cryptology–Eurocrypt 1985 (LNCS)*; Springer-Verlag: Linz, Austria, 1985; Volume 219, pp. 73–80.
16. Lin, C.H.; Chang, C.C.; Lee, R.C.T. A New Public-Key Cipher System based upon The Diophantine Equations. *IEEE Trans. Comput.* **1995**, *44*, 13–19. [CrossRef]
17. Webb, W.A. A Public Key Cryptosystem based on Complementing Sets. *Cryptologia* **1992**, *XVI*, 177–181. [CrossRef]
18. Brickell, E.F. Solving Low Density Knapsacks. In *Advances in Cryptology–Crypto 1983*; Plenum: New York, NY, USA, 1984; pp. 24–37.
19. Lagarias, J.C.; Odlyzko, A.M. Solving Low-Density Subset Sum Problems. *J. ACM* **1985**, *32*, 229–246. [CrossRef]
20. Coster, M.J.; LaMacchia, B.A.; Odlyzko, A.M.; Schnorr, C.P. An Improved Low-Density Subset Sum Algorithm. In *Advances in Cryptology–Eurocrypt 1991 (LNCS)*; Springer-Verlag: Brighton, UK, 1991; Volume 547, pp. 54–67.
21. Brickell, E.F.; Odlyzko, A.M. Cryptanalysis: A Survey of Recent Results. In *Contemporary Cryptology, The Science of Information Integrity*; IEEE Press: New York, NY, USA, 1992; pp. 501–540.
22. Lagarias, J.C. Knapsack Public Key Cryptosystems and Diophantine Approximation. In *Advances in Cryptology–Crypto 1983*; Plenum: New York, NY, USA, 1984; pp. 3–23.
23. Lai, M.K. Knapsack Cryptosystems: The Past and The Future. Available online: http://www.ics.uci.edu/~{}mingl/knapsack.html (accessed on 20 December 2003).
24. Odlyzko, A.M. The Rise and Fall of Knapsack Cryptosystems. *Am. Math. Soc. Proc. Symp. Appl. Math* **1990**, *42*, 75–88.
25. Lenstra, A.K.; Lenstra, H.W., Jr.; Lovász, L. Factoring Polynomials with Rational Coefficients. *Math. Ann.* **1982**, *261*, 513–534. [CrossRef]
26. Omura, K.; Tanaka, K. Density Attack to The Knapsack Cryptosystems with Enumerative Source Encoding. *IEICE Trans. Fund.* **2001**, *E84-A*, 1564–1569.
27. Nguyen, P.; Stern, J. Adapting Density Attacks to Low-Weight Knapsacks. In *Advances in Cryptology–Asiacrypt 2005 (LNCS)*; Springer-Verlag: Chennai, India, 2005; Volume 3788, pp. 41–58.
28. Wang, B.; Hu, Y. Diophantine Approximation Attack on A Fast Public Key Cryptosystem. In *The 2nd Information Security Practice and Experience Conference–ISPEC 2006 (LNCS)*; Springer: Hangzhou, China, 2006; Volume 3903, pp. 25–32.
29. Schnorr, C.P.; Euchner, M. Lattice Basis Reduction: Improved Practical Algorithms and Solving Subset Sum Problems. *Math. Progr.* **1994**, *66*, 181–191. [CrossRef]
30. Ajtai, M.; Dwork, C. A Public-Key Cryptosystem with Worst-Case/Average-Case Equivalence. In Proceedings of the 29th ACM STOC, El Paso, TX, USA, 4–6 May 1997; pp. 284–293.
31. Goldreich, O.; Goldwasser, S.; Halvei, S. Public-Key Cryptosystems from Lattice Reduction Problems. In *Advances in Cryptology–Crypto 1997 (LNCS)*; Springer-Verlag: Santa Barbara, CA, USA, 1997; Volume 1294, pp. 112–131.
32. Hoffstein, J.; Pipher, J.; Silverman, J.H. NTRU: A New High Speed Public Key Cryptosystem. In *Proceedings the of Algorithm Number Theory–ANTS III (LNCS)*; Springer-Verlag: Portland, OR, USA, 1998; Volume 1423, pp. 267–288.
33. Cai, J.Y.; Cusick, T.W. A lattice-based Public-Key Cryptosystem. *Inf. Comput.* **1999**, *151*, 17–31. [CrossRef]
34. Sakurai, K. A Progress Report on Lattice-based Public-Key Cryptosystems—Theoretical Security Versus Practical Cryptanalysis. *IEICE Trans. Inf. Syst.* **2000**, *E83-D*, 570–579.
35. Nguyen, P.; Stern, J. The Two Faces of Lattices in Cryptology. In *Proceedings of the Cryptography and Lattices–CaLC (LNCS)*; Springer-Verlag: Providence, RI, USA, 2001; Volume 2146, pp. 146–180.
36. Shamir, A. On The Cryptocomplexity of Knapsack Systems. In Proceedings of the Eleventh Annual ACM Symposium on Theory of Computing, Atlanta, GA, USA, 30 April–2 May 1979; pp. 118–129.
37. Katayangi, K.; Murakami, Y. A New Product-Sum Public-Key Cryptosystem Using Message Extension. *IEICE Trans. Fund.* **2001**, *E84-A*, 2482–2487.
38. Okamoto, T.; Tanaka, K.; Uchiyama, S. Quantum Public-Key Cryptosystems. In *Advances in Cryptology–Crypto 2000 (LNCS)*; Springer-Verlag: Santa Barbara, CA, USA, 2000; Volume 1880, pp. 147–165.
39. Wang, B.; Hu, Y. Public Key Cryptosystem based on Two Cryptographic Assumptions. *IEE Proc. Commun.* **2005**, *152*, 861–865.

40. Shamir, A.; Zippel, R.E. On The Security of The Merkle-Hellman Cryptographic Scheme. *IEEE Trans. Inf. Theory* **1980**, *26*, 339–340. [CrossRef]
41. Laih, C.S.; Gau, M.J. Cryptanalysis of A Diophantine Equation Oriented Public Key Cryptosystem. *IEEE Trans. Comput.* **1997**, *46*, 511–512. [CrossRef]
42. Eier, R.; Lagger, H. Trapdoors in Knapsack Cryptosystems. In *Cryptography–EUROCRYPT 1982 (LNCS)*; Springer: Berlin/Heidelberg, Germany, 1982; Volume 149, pp. 316–322.
43. Wang, B.; Wu, Q.; Hu, Y. A Knapsack-based Probabilistic Encryption Scheme. *Inf. Sci.* **2007**, *177*, 3981–3994. [CrossRef]
44. Koblitz, N. *Algebraic Aspects of Cryptography*; Springer-Verlag: Berlin, Germany, 1998.
45. Wolf, C. Multivariate Quadratic Polynomials in Public Key Cryptography. Ph.D. Thesis, Katholieke Universiteit Leuven, Leuven, Belgium, 2005. Available online: http://eprint.iacr.org/2005/393 (accessed on 1 November 2005).
46. Koblitz, N.; Menezes, A.J. Another Look at "Provable Security". *J. Cryptol.* **2007**, *20*, 3–37. [CrossRef]
47. Koskinen, J.A. Non-Injective Knapsack Public-Key Cryptosystems. *Theor. Comput. Sci.* **2001**, *255*, 401–422. [CrossRef]
48. Han, S.; Chang, E.; Dillon, T. Knapsack Diffie-Hellman: A New Family of Diffie-Hellman. *Cryptology ePrint Archive: Report 2005/347*. Available online: http://eprint.iacr.org/2005/347 (accessed on 22 August 2006).
49. Su, P.C.; Lu, E.; Chang, H. A Knapsack Public-Key Cryptosystem based on Elliptic Curve Discrete Logarithm. *Appl. Math. Comput.* **2005**, *168*, 40–46. [CrossRef]