INFORMATICA, 2010, Vol. 21, No. 4, 611–626

A Novel Combinatorial Public Key Cryptosystem *

Baocang WANG[1,2], Yupu HU[1]

[1] *Key Laboratory of Computer Networks and Information Security*
Xidian University, Xi'an 710071, China
[2] *State Key Laboratory of Information Security, Institute of Software*
Chinese Academy of Sciences, Beijing 100049, China
e-mail: bcwang79@yahoo.com.cn, yphu@mail.xidian.edu.cn

Received: February 2009; accepted: April 2010

Abstract. Combinatorial problems serve as an important resource for developing practical public key cryptosystems and several combinatorial cryptosystems have been proposed in the cryptographic community. In this paper, a combinatorial public key cryptosystem is proposed. The security of the proposed cryptosystem is dependent on a combinatorial problem involving matrices. The system features fast encryption and decryption. However, the system also suffers from some drawbacks. The ciphertext expansion is relatively large and the key sizes are somewhat larger than that of RSA. The security of the system is carefully examined by illustrating the computational infeasibilities of some attacks on the system.

Keywords: public key cryptography, combinatorial cryptosystem, integer factorization, lattice reduction, security.

1. Introduction

It is striking to note that nowadays the security of most of the widely-used public key cryptosystems (PKCs) is based on number-theoretic problems, such as factoring integers and finding discrete logarithms over some cyclic groups (Liu and Huang, 2010; Sun *et al.*, 2010; Wang *et al.*, 2009). The desire to have a wide variety of available cryptosystems so as not to put all cryptographic eggs in one number-theoretic basket motivates the cryptographers to design more PKCs. An interesting area in public key cryptography is to design PKCs based on some combinatorial problems.

Trapdoor knapsack, a concept proposed by Merkle and Hellman (1978), can be seen as the first practical realization of combinatorial cryptosystems. The basic Merkle–Hellman trapdoor knapsack was broken by Shamir (1984). Whereafter, many work has been done to realize secure and efficient trapdoor knapsacks. However, almost all additive trapdoor knapsacks are shown to be insecure, including the most resistant one, the Chor–Rivest

*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), the 111 Project (No. B08038), the Zhejiang Provincial Natural Science Foundation of China (No. Y1091085), the Fundamental Research Funds for the Central Universities (No. JY10000901009), and the Henan Provincial Foundations and Frontiers of Science and Technology Project (No. 092300410159).

knapsack system (Chor and Rivest, 1988; Vaudenay, 2001). The painful experience traumatized many cryptographers, and also lowered the initial enthusiasm for combinatorially based cryptosystems. See the survey papers (Lai, 2003; Odlyzko, 1990) for the rise and fall for trapdoor knapsacks.

In 1994, Fellows and Koblitz reignited the cryptographers' enthusiasm for combinatorial cryptosystems by developing a combinatorial–algebraic cryptosystem, Polly Cracker (Fellows and Koblitz, 1994; Koblitz, 1998), which was modified by Ly (2006). The modified Polly Cracker is called Polly Two in Ly (2006). The Polly Cracker and Polly Two cryptosystems attracted a lot of attention and the security of the cryptosystems was discussed in some well-written papers (Steinwandt and Geiselman, 2002; Hofheinz and Steinwandt, 2002; Steinwandt *et al.*, 2002; Ackerman *et al.*, 2006). More generally, we also can look the braid-based cryptographic primitives (Ko *et al.*, 2000; Ahshel *et al.*, 1999) and NTRU (Hoffstein *et al.*, 1998) as combinatorial–algebraic cryptosystems.

In this paper, we propose a new combinatorial PKC. The encryption of the proposed cryptosystem only involves several modular multiplications and additions. The computational costs for the user to decipher a ciphertext are also several modular multiplicative operations. The proposed cryptosystem suffers from some drawbacks. The information rate is somewhat lower than that of RSA and ElGamal, and the key sizes are relatively large. When developing the cryptosystem, we use the integer factorization problem to disguise the secret key. However, the security of the system is not dependent on the integer factorization intractability assumption but a combinatorial problem involving matrices.

The rest of the paper is organized as follows. In Section 2, we give the detailed description of the proposed cryptosystem; Section 3 analyzes the computational complexity of the system and specifies the parameter selection. The security of the system is discussed in Section 4. Section 5 gives some concluding remarks.

2. The Proposed Cryptosystem

Throughout this paper, we use $\mathbf{R}$ and $\mathbf{Z}_n = \{0, \ldots, n-1\}$ to denote the field of real numbers and the complete system of the least nonnegative residues modulo n, respectively. We write $\gcd(a, b)$ for the greatest common divisor of a and b. If $\gcd(a, b) = 1$, $a^{-1} \bmod b$ denotes the inverse of a modulo b. We use $\langle b \rangle_p$ to mean the least nonnegative remainder of a divided by p. We use $a = b (\bmod N)$ to mean that a is the least nonnegative remainder of b modulo N and use $a \equiv b (\bmod N)$ to denote that a and b are congruent modulo N. The symbol $|A|$ represents the determinant of a square matrix A and $|a|$ means the binary length of an integer a.

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

2.1. *Key Generation*

The proposed encryption scheme involves matrices with dimension of an even integer n. In real life practice, we can choose $n = 4$. The basic steps to choose parameters run as follows.

Randomly generate a 1024 RSA modulus $N = pq$ with p and q primes and $|p| = |q| = 512$. Randomly choose an n-dimensional matrix $A = (a_{ij})_{n\times n}$ with $|a_{ij}| = 59$. We require that the matrix A is invertible over $\mathbf{R}$ and denote its inverse as A^{-1}. We randomly choose four matrices $C = (c_{ij})_{n\times n}$, $D = (d_{ij})_{n\times n}$, $E = (e_{ij})_{n\times n}$ and $F = (f_{ij})_{n\times n}$ with $c_{ij}, d_{ij}, e_{ij}, f_{ij} \in \mathbf{Z}_N$. The generated matrices C, D, E, and F satisfy the following conditions. For $i = 1, \ldots, n$, we require that p divides $c_{ij} + e_{i(n+1-j)}$ when i is an odd integer, and q divides $c_{ij} + e_{i(n+1-j)}$ when i is an even integer, that is,

$$\begin{cases} c_{ij} + e_{i(n+1-j)} = \alpha_{ij} p, & \text{when } i \text{ is odd,} \\ c_{ij} + e_{i(n+1-j)} = \alpha_{ij} q, & \text{when } i \text{ is even,} \end{cases} \tag{1}$$

where $\alpha_{ij} \in \mathbf{Z}_N$. We also require that p divides $d_{in} + f_{i1}$ and $d_{ij} + f_{i(j+1)}$ for $j = 1, \ldots, n-1$ when i is an odd integer, and q divides $d_{in} + f_{i1}$ and $d_{ij} + f_{i(j+1)}$ for $j = 1, \ldots, n-1$ when i is an even integer, that is,

$$\begin{cases} d_{in} + f_{i1} = \beta_{in} p, & \text{when } i \text{ is odd,} \\ d_{in} + f_{i1} = \beta_{in} q, & \text{when } i \text{ is even,} \\ d_{ij} + f_{i(j+1)} = \beta_{ij} p, j = 1, \ldots, n-1, & \text{when } i \text{ is odd,} \\ d_{ij} + f_{i(j+1)} = \beta_{ij} q, j = 1, \ldots, n-1, & \text{when } i \text{ is even,} \end{cases} \tag{2}$$

where $\beta_{ij} \in \mathbf{Z}_N$. Now we generate another matrix $A' = (a'_{ij})_{n\times n}$, where

$$\begin{cases} a'_{ij} = a_{ij} + \gamma_{ij} p, & \text{when } i \text{ is odd,} \\ a'_{ij} = a_{ij} + \gamma_{ij} q, & \text{when } i \text{ is even,} \end{cases} \quad \gamma_{ij} \in \mathbf{Z}_N. \tag{3}$$

To make the proposed cryptosystem well work, we also require that the matrices A', C, D, E and F are invertible modulo N. We denote the inverses of D, F modulo N as D^{-1} and F^{-1} respectively. We compute

$$\begin{aligned} B &= (b_{ij})_{n\times n} \equiv D^{-1} A' (\text{mod}\, N), \\ G &= (g_{ij})_{n\times n} \equiv D^{-1} C (\text{mod}\, N), \\ H &= (h_{ij})_{n\times n} \equiv F^{-1} E (\text{mod}\, N). \end{aligned} \tag{4}$$

The matrices of B, G and H and the modulus N are the public key; whereas the secret key consists of D, F, A^{-1}, p and q.

2.2. *Encryption*

The plaintext to be encrypted is M with $|M| = 450n$, which is divided into n blocks $m_1, \ldots, m_n$ with every block m_i, $|m_i| = 450$. To encrypt M, the sender randomly

chooses $2n$ integers $r_1, \ldots, r_n, s_1, \ldots, s_n \in \mathbf{Z}_N$. The sender computes the cipher-text (U, V) as follows,

$$U = \begin{pmatrix} u_1 \\ \vdots \\ u_n \end{pmatrix} \equiv B \begin{pmatrix} m_1 \\ \vdots \\ m_n \end{pmatrix} + G \begin{pmatrix} r_1 \\ \vdots \\ r_n \end{pmatrix} + \begin{pmatrix} s_1 \\ \vdots \\ s_n \end{pmatrix} (\text{mod } N),$$

and

$$V = \begin{pmatrix} v_1 \\ \vdots \\ v_n \end{pmatrix} \equiv H \begin{pmatrix} r_n \\ r_{n-1} \\ \vdots \\ r_2 \\ r_1 \end{pmatrix} + \begin{pmatrix} s_n \\ s_1 \\ \vdots \\ s_{n-2} \\ s_{n-1} \end{pmatrix} (\text{mod } N). \tag{5}$$

The ciphertext is the 2-tuple (U, V) and is sent to the intended receiver.

2.3. *Decryption*

Given a ciphertext vector (U, V), the receiver does the followings to recover the corresponding plaintext vector M. The receiver first computes $T = (t_1, \ldots, t_n)^T \equiv DU + FV (\text{mod } N)$, and sets $w_i = \langle t_i \rangle_p$, when i is odd, and $w_i = \langle t_i \rangle_q$, when i is even. Then he recovers the plaintext,

$$M = (m_1, \ldots, m_n)^T = A^{-1}(w_1, \ldots, w_n)^T. \tag{6}$$

2.4. *Why Decryption Works*

It is easy to verify that

$$T \equiv A'M + C \begin{pmatrix} r_1 \\ \vdots \\ r_n \end{pmatrix} + D \begin{pmatrix} s_1 \\ \vdots \\ s_n \end{pmatrix} + E \begin{pmatrix} r_n \\ r_{n-1} \\ \vdots \\ r_2 \\ r_1 \end{pmatrix} + F \begin{pmatrix} s_n \\ s_1 \\ \vdots \\ s_{n-2} \\ s_{n-1} \end{pmatrix} (\text{mod } N).$$

So for $i = 1, \ldots, n-1$,

$$\begin{aligned} t_i \equiv & \sum_{j=1}^{n} a_{ij}^{'} m_j + \sum_{j=1}^{n} [c_{ij} + e_{i(n+1-j)}] r_j + \sum_{j=1}^{n-1} [d_{ij} + f_{i(j+1)}] s_j \\ & + (d_{in} + f_{i1}) s_n (\text{mod } N). \end{aligned}$$

From (1), (2) and the construction of the matrix A', we know when i is odd, we have

$$t_i \equiv \sum_{j=1}^{n} a_{ij}m_j + p\left[\sum_{j=1}^{n}(\gamma_{ij}m_j + \alpha_{ij}r_j + \beta_{ij}s_j)\right](\text{mod}N),$$

and when i is even, we have

$$t_i \equiv \sum_{j=1}^{n} a_{ij}m_j + q\left[\sum_{j=1}^{n}(\gamma_{ij}m_j + \alpha_{ij}r_j + \beta_{ij}s_j)\right](\text{mod } N).$$

Accordingly, we get

$$w_i = \langle t_i \rangle_p \equiv \sum_{j=1}^{n} a_{ij}m_j (\text{mod}\, p), \quad \text{when } i \text{ odd },$$
$$w_i = \langle t_i \rangle_q \equiv \sum_{j=1}^{n} a_{ij}m_j (\text{mod}\, q), \quad \text{when } i \text{ even}.$$

Note that $|a_{ij}| = 59$, $|m_i| = 450$, $|p| = |q| = 512$. So we can conclude that

$$0 < \sum_{j=1}^{n} a_{ij}m_j < p, \qquad 0 < \sum_{j=1}^{n} a_{ij}m_j < q.$$

In other words, when i is odd, we have

$$w_i = \langle t_i \rangle_p = \sum_{j=1}^{n} a_{ij}m_j,$$

and when i is even, we have

$$w_i = \langle t_i \rangle_q = \sum_{j=1}^{n} a_{ij}m_j.$$

We can write the equation as

$$(w_1, \ldots, w_n)^T = A(m_1, \ldots, m_n)^T.$$

The plaintext is recovered by computing $M = A^{-1}(w_1, \ldots, w_n)^T$.

2.5. *An Example*

To make the proposed cryptosystem easy to understand, now we use a small example to illustrate the procedure. Firstly, the receiver generates two primes $p = 999979$ and

$q = 999631$, and computes their multiplication $N = pq = 999610007749$. Then five matrices are generated as follows,

$$A = \begin{pmatrix} 4 & 5 \\ 3 & 4 \end{pmatrix},$$
$$C = \begin{pmatrix} 248569123540 & 369785444609 \\ 838897460095 & 756483645235 \end{pmatrix},$$
$$E = \begin{pmatrix} 285078802953 & 627901470191 \\ 232874146978 & 159186111117 \end{pmatrix},$$
$$D = \begin{pmatrix} 156889736501 & 287632569851 \\ 652322010023 & 665130086997 \end{pmatrix},$$

and

$$F = \begin{pmatrix} 400904970557 & 667425952506 \\ 100154418780 & 137373484774 \end{pmatrix}.$$

It is easy to verify that the generated matrices C, D, E, and F satisfy the conditions (1) and (2). Then the receiver generates another matrix $A^{'} = (a_{ij}^{'})_{n\times n}$ according to (3),

$$A^{'} = A + \begin{pmatrix} 687356p & 355865p \\ 771023q & 560041q \end{pmatrix} = \begin{pmatrix} 687341565528 & 355857526840 \\ 770738492516 & 559834344875 \end{pmatrix}.$$

The receiver computes $D^{-1}(\mathrm{mod}\, N)$ and $F^{-1}(\mathrm{mod}\, N)$,

$$D^{-1} = \begin{pmatrix} 631418586541 & 354198766323 \\ 106207153194 & 865153161893 \end{pmatrix},$$
$$F^{-1} = \begin{pmatrix} 254565584792 & 195881254137 \\ 529308392657 & 71437726285 \end{pmatrix}.$$

Therefore, the receiver can compute the public matrices

$$B \equiv D^{-1}A^{'}(\mathrm{mod}\, N) = \begin{pmatrix} 41728074639 & 951212423002 \\ 416683313190 & 775553060663 \end{pmatrix},$$
$$G \equiv D^{-1}C(\mathrm{mod}\, N) = \begin{pmatrix} 641231320645 & 468099058568 \\ 95261839148 & 118089802391 \end{pmatrix},$$
$$H \equiv F^{-1}E(\mathrm{mod}\, N) = \begin{pmatrix} 907836643133 & 572493941408 \\ 431066561770 & 130333978646 \end{pmatrix}.$$

Assume that the plaintext to be encrypted is $M = (m_1,\ m_2)^T = (89436,\ 77201)^T$. The sender randomly generates $r_1 = 846320073629$, $r_2 = 635508996021$, $s_1 = 384620183512$, $s_2 = 201556638534 \in \mathbf{Z}_N$. Then he computes and sends the receiver

the ciphertext $(U,\ V)$,

$$U = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \equiv B \begin{pmatrix} m_1 \\ m_2 \end{pmatrix} + G \begin{pmatrix} r_1 \\ r_2 \end{pmatrix} + \begin{pmatrix} s_1 \\ s_2 \end{pmatrix} (\text{mod } N) = \begin{pmatrix} 824702160981 \\ 951670702040 \end{pmatrix},$$
$$V = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} \equiv H \begin{pmatrix} r_2 \\ r_1 \end{pmatrix} + \begin{pmatrix} s_2 \\ s_1 \end{pmatrix} (\text{mod } N) = \begin{pmatrix} 81144043583 \\ 305049716413 \end{pmatrix}.$$

To decipher the ciphertext $(U,\ V)$, the receiver firstly computes

$$T = \begin{pmatrix} t_1 \\ t_2 \end{pmatrix} \equiv DU + FV (\text{mod } N) = \begin{pmatrix} 599460154837 \\ 773188165575 \end{pmatrix}.$$

Then he computes $w_1 = \langle t_1 \rangle_p = 743749$ and $w_2 = \langle t_2 \rangle_q = 577112$. The plaintext is recovered by performing

$$M = \begin{pmatrix} m_1 \\ m_2 \end{pmatrix} = A^{-1} \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} = \begin{pmatrix} 4 & -5 \\ -3 & 4 \end{pmatrix} \begin{pmatrix} 743749 \\ 577112 \end{pmatrix} = \begin{pmatrix} 89436 \\ 77201 \end{pmatrix}.$$

3. Parameter Specifications and Performance

3.1. *Parameter Specifications*

When generating the parameters, we require that A is invertible over the field of $\mathbf{R}$. Generally speaking, the entries in the matrix A^{-1} are rational numbers but not necessarily integers. So the matrix A^{-1} cannot be efficiently represented. Three methods can be used to overcome the drawback. Firstly, we can choose the matrix A such that $|A| = 1$, in which case, the entries of the matrix A^{-1} are also integers. Secondly, for a randomly-chosen invertible matrix A, we denote its inverse as $A^{-1} = \frac{1}{|A|} A^*$, where A^* is the adjoint of A. When we do the computation (6) to decipher a ciphertext, we compute $A^*(v_1, v_2)^T$ and then divide the resultant vector by $|A|$. Thirdly, we can compute the inverse A^{-1} of A modulo p. Then (6) should be modified as

$$M = (m_1, \ldots, m_n)^T = A^{-1}(w_1, \ldots, w_n)^T (\text{mod}\, p). \tag{7}$$

We also require the matrices $A^{'}$, C, D, E and F are invertible modulo N. This is only for security considerations. In fact, if one of the matrices is non-invertible modulo N, at least one of the public matrices B, G and H will be non-invertible modulo N. Without loss of generality, we assume that B is non-invertible modulo N. So gcd$(|B|, N)$ will be greater than 1. If gcd$(|D|, N) \neq N$, then gcd$(|D|, N)$ will give a prime factor of N. So the security of the proposed cryptosystem will be reduced. In fact, such matrices are easy to generate. Note that a matrix A is invertible modulo N if and only if gcd$(|A|, N) = 1$. So for a small dimension n and a large RSA number $N = pq$, a randomly-chosen n-dimensional square matrix A is always invertible modulo N.

The suggested parameters are listed in Table 1.

Table 1
The suggested parematers with respect to security level

Security level	Moderate security	Higher security	Highest security
n	2	4	4
Length of N, $\|N\|$	1024	1024	2048

3.2. *Performance*

3.2.1. *Key Size*

We evaluate the public key and the secret key sizes. The public key consists of three n-dimensional square matrices B, G and H and the modulus N. So the public key size is about $(3n^2+1)$ times the binary length of the modulus N, i.e., $(3n^2+1)\times 1024 = 50176$ bits in the case of $n = 4$. The secret key consists of D, F, A^{-1}, p and q. So the secret key size can be evaluated via $2n^2 \times 1024 + n^2 \times 512 + 2 \times 512 = 41984$ bits. The public key size of the proposed cryptosystem is relatively large.

3.2.2. *Information Rate*

The information rate for a cryptosystem is defined as the ratio of the binary length l_m of the plaintext block to that of the ciphertext block l_c. In the proposed cryptosystem, the size of the plaintext block is $n \times 450 = 450n$ bits, whereas the size of the ciphertext block is about $2n \times 1024 = 2048n$ bits. So the information rate of the proposed cryptosystem is $r = l_m/l_c = 450n/2048n \approx 0.22$. In other words, the ciphertext expansion of the proposed cryptosystem is about 4.55:1.

3.2.3. *Computational Complexity*

In the proposed cryptosystem, only several modular multiplications and additions are performed during encryption and decryption. The most costly operations are the modular multiplications. So the computational complexity of the system to encrypt a plaintext and to decipher a ciphertext is given as $O(k^2)$, where $k = 1024$ is the security parameter. However, to make our evaluations more precise and concrete, we need to evaluate how many modular multiplications are needed during encryption and decryption.

Note the fact that it needs 1024^2 bit operations to do a multiplication modulo a 1024-RSA modulus N. We also assume that $n = 4$. The cryptosystem has to carry out three costly operations $B(m_1, \ldots, m_n)^T \bmod N$, $G(r_1, \ldots, r_n)^T \bmod N$ and $H(r_n, r_{n-1}, \ldots, r_1)^T \bmod N$ to encrypt a message, where $|m_i| = 450$, $|r_i| = |s_i| = 1024$. So it takes about $n^2 \times 1024 \times 450 + 2n^2 \times 1024 \times 1024 \approx 4 \times 10^7$ bit operations to do the computations. The computational cost for doing these is equivalent to that of $\frac{4\times 10^7}{1024^2} \approx 39$ 1024-modular multiplications.

The decryption algorithm needs to do the two costly computations $DU + FV \pmod N$ and $A^{-1}(w_1, \ldots, w_n)^T \pmod p$ (we assume that the receiver use (7) to decipher a ciphertext) to recover a plaintext. So the computational cost is about $2n^2 \times 1024 \times 1024 + n^2 \times 512 \times 512 \approx 3.77 \times 10^7$ bit operations, which is equivalent to computing $\frac{3.77\times 10^7}{1024^2} = 36$ 1024-modular multiplications.

3.3. *Implementation and Comparisons*

We make a comparison for our cryptosystem and the RSA system. When generating the key pairs, both our cryptosystem and RSA need to generate two strong RSA numbers. Additionally, our system needs to randomly choose several modulo N invertible matrices and RSA must pick out a public parameter e. However, a randomly-chosen n-dimensional square matrix A is always invertible modulo N, as pointed out earlier. So the computational time for the key generation of the proposed scheme is about the same as that of RSA. The RSA system is a trapdoor permutation, so its ciphertext expansion is 1:1. The ciphertext expansion of RSA is of course somewhat lower than that of our system. From the evaluations about the key sizes, we conclude that the key sizes of the proposal is larger than that of the RSA system. However, as far as the information rate and the key sizes concerned, our system remains practical. The advantage of our system over RSA should be its speed. The computational costs for the encryption algorithm and decryption algorithm of RSA is $1.5 \log_2 e$ and $1.5 \log_2 d$ modulo multiplications respectively. Of course, the RSA system can pick a small e to speed up the encryption; or it can choose small $d_p \equiv d(\mathrm{mod}\, p-1)$ and $d_q \equiv d(\mathrm{mod}\, q-1)$ to accelerate the decryption by using the Chinese Remainder Theorem. However, it is very difficult to make the encryption and the decryption very efficient simultaneously. Whereas both the encryption and the decryption algorithm of the proposed system is of quadratic bit complexity.

To illustrate the advantages and disadvantages of the proposed cryptosystem over RSA, both RSA and our cryptosystem are implemented on an Intel Pentium D Cpu 2.80GHz computer with 1Gb of RAM. The computations are performed in C++ in vs2008 and Windows XP environments, using Crypto++ 5.6.0 library. See `www:cryptopp.com`. When implementing RSA-1024, we randomly choose an encryption exponentiation e, and use the Chinese remainder theorem to accelerate the decryption. Our cryptosystem is implemented by choosing a higher security level, i.e., setting $n = 4$ and $|N| = 1024$. With respect to encryption/decryption efficiency, we measure their computational time in encrypting and decrypting a 1024-bit plaintext. Both RSA and the proposed cryptosystem are implemented 1024 times. Hence, the RSA algorithm encrypts and decrypts 1024×1024 bits plaintexts. We measure the consumed time for RSA to encrypt and decrypt a 1024-bit plaintext by the total time divided by 1024. The proposed cryptosystem encrypts and decrypts $1024 \times 4 \times 450$ bits plaintexts, and the computational time of the proposed cryptosystem in encrypting and decrypting a 1024-bit plaintext is estimated by the total time divided by 4×450. The implementation results are summarized in Table 2.

From the implementations, we can see that the proposed encryption scheme is more efficient on computational cost. But its key size and information rate are relatively large compared with the RSA cryptosystem. Hence, the proposed scheme is not suitable for resource-constraint environment like wireless communication in which the used device has limited memory storage and the transmission cost is very high.

Table 2
Implementation results for both RSA and our proposal

Cryptosystem	Key generation	Encryption	Decryption
RSA	70.283 ms	70.682 ms	18.023 ms
Our proposal	75.369 ms	1.134 ms	0.946 ms

4. Security

There are two methods used for the security analysis in public key cryptography. One is provable security theory. The basic idea of provable security is to reduce the security of a PKC under some attack model to a mathematically hard problem. Another method is to deliver the PKC to the literature for attacks. If the PKC is secure against all known attacks, we can assume its one-wayness. Here, we exclude provable security discussions about our system and just provide some known attacks on it.

Koblitz (1998) pointed out that in a PKC there are two types of one-way functions:

1. the encryption function (whose inversion is called the cracking problem);
2. the underlying function used to construct the trapdoor (the problem of reversing the basic mathematical construction of the trapdoor is called the trapdoor problem).

The basic requirements for a PKC to be secure are that both the cracking problem and the trapdoor problem are computationally infeasible. Now, we discuss several attacks aiming at solving the cracking problem and the trapdoor problem in our system.

4.1. *Lattice Attack*

As powerful cryptanalytic tools, lattice basis reduction algorithms had been used to attack many cryptographic primitives (Joux, 1998). Especially, when a cryptosystem involves linear equations, the cryptanalytic history tells us that it is always vulnerable to lattice attacks. So we must examine the security of the proposed cryptosystem against lattice attacks.

4.1.1. *Lattice Attack on the Cracking Problem*

To decihper a cipher-text, the attacker can solve the encryption function (5) for the plain-text M regardless of the special structure of the secret keys. The most powerful tool for solving linear multivariate equation with small variables is the lattice basis reduction algorithms, such as LLL (Lenstra *et al.*, 1982). However, the lattice attacks do not apply to cryptanalyze our cryptosystem due to the following considerations. Firstly, lattice basis reduction algorithms are applicable to the normed spaces. Note that our cryptosystem involves the arithmetics in $\mathbf{Z}_n$, which is not a normed space. So the lattice-based attacks are not applicable to find the solutions to (5). Secondly, even if the involved space is a normed space, the lattice attacks cannot break our cryptosystem. As we know, the lattice reduction algorithms are used to approximate the shortest vector in a lattice. However, in the proposed cryptosystem, the encryption function (5) is a system of linear multivariate

equations with large variables $r_1, \ldots, r_n, s_1, \ldots, s_n$. So we cannot expect constructing a lattice such that the plaintext and the random coins $r_1, \ldots, r_n, s_1, \ldots, s_n$ are the entries of the shortest vector of the lattice.

4.1.2. *Lattice Attack on the Trapdoor Problem*

The construction of the trapdoor of the system uses a similar structure to that of NTRU (Hoffstein *et al.*, 1998): using the multiplication for one element and the inverse of another element over some algebraic structure (the truncated polynomial ring in the case of NTRU and the matrix ring over $\mathbf{Z}_n$ in the case of the proposed cryptosystem). So the attacker looks forward to recovering the trapdoor information by constructing a lattice similar to the lattice defined by Coppersmith and Shamir (1997). However, the two algebraic structures underling the two cryptosystems are very different. It is not known how to construct such a lattice from the matrix ring over $\mathbf{Z}_n$. Even if the attacker can construct such a lattice, it is impossible for the attacker to obtain enough information to retrieve the secret key by learning the lattice. In fact, the proposed cryptosystem does not use a small element as its secret key. So the short vector in the lattice found by the attacker will contain little information about the secret key and hence will make no sense.

4.2. *Key Recovery Attacks*

The key recovery attacks aim at recovering from the public information the secret key used by the intended receiver to decipher a cipher-text. The secret key of the proposed cryptosystem consists of D, F, A^{-1}, p and q. In fact, if the attacker recovers the exact value of the matrices D and F, he can retrieve the matrices $A^{'}$, C and E. So he can obtain the exact values of p and q by computing $p = \gcd(c_{11} + e_{1n}, N)$ and $q = \gcd(c_{21} + e_{2n}, N)$, where the gcd's can be efficiently computed by the Euclidean algorithm. Note that $a_{ij} = a_{ij}^{'} (\mathrm{mod}\, p)$, when i is odd and that $a_{ij} = a_{ij}^{'} (\mathrm{mod}\, q)$, when i is even. So the attacker retrieves the matrix A. It is an easy thing to compute the inverse matrix A^{-1} of A. So the attacker can recover the secret key (D, F, A^{-1}, p, q) if he recovers D and F. So one method for the attacker to recover the secret key is to search for the matrices D and F subject to (1) and (2).

To recover the secret matrix D, the attacker can use the fact that the secret matrix $D^{-1} \mathrm{mod}\, N$ is a common multiplier for the public matrices B and G to recover the secret matrix D. Now we show that the attacker cannot use the fact to distil some useful information about the secret matrix D. The ring of matrices defined over ring $\mathbf{Z}_n$ is not an Euclidean domain. So we cannot expect developing an Euclidean-like algorithm to compute a common divisor for the two matrices B and G. In fact, for any invertible matrix O modulo N, O always can serve as such a common divisor for B and G, i.e., $B \equiv O(O^{-1}B) (\mathrm{mod}\, N)$ and $G \equiv O(O^{-1}G) (\mathrm{mod}\, N)$. Now we can conclude that it is infeasible for the attacker to distill the common multiplier D from the public matrices B and G.

Note that $H \equiv F^{-1}E (\mathrm{mod}\, N)$. It is also impossible to obtain the matrices E and F by decomposing H into the product of E and the inverse of F. In fact, for any invertible matrix O modulo N, we always can write H as $H \equiv O^{-1}(OH) (\mathrm{mod}\, N)$. So the individual matrix H provides no information about the matrices E and F.

As discussed above, to obtain some useful information about the secret key, the attacker must use the interrelations underlying the public matrices G and H. Formally, we define this problem as matrix combinatorial problem.

The Matrix Combinatorial Problem. Given two matrices G and H and an RSA modulus $N(=pq)$, decompose the matrices G and H into the forms of $G \equiv D^{-1}C(\mathrm{mod}\, N)$, $H \equiv F^{-1}E(\mathrm{mod}\, N)$ such that the entries of the matrices C, D, E and F are subject to (1) and (2).

This special matrix combinatorial problem on which the proposed cryptosystem is based is a new algebraic–combinatorial problem. However, we do not know the computational complexity for solving the combinatorial problem. So the readers are encouraged to study the problem and to provide attacks on the proposed cryptosystem. In the proposed cryptosystem, the integer factorization problem is also used to disguise the secret key. We do not know whether the prime factorization of the modulus N will lead to the solution to the matrix combinatorial problem and hence compromise the security of the system or not. If not, then we also can publicize the primes p and q without reducing the security. And also, we can cancel the requirements that the matrices $A^{'}$, C and E are invertible mod N.

4.3. *Algorithms for the Matrix Combinatorial Problem*

Now we provide two algorithms for the matrix combinatorial problem and analyze their computational complexities.

4.3.1. *Algorithm I*

Given two matrices G and H and an RSA modulus $N = pq$, the attacker wants to decompose the matrices G and H into the forms of $G \equiv D^{-1}C(\mathrm{mod}\, N)$, $H \equiv F^{-1}E(\mathrm{mod}\, N)$ such that the entries of the matrices C, D, E and F are subject to (1) and (2). One straightforward way for the attacker is to do exhaustive search for D and F. For any possible matrices D and F, the attacker computes $C \equiv DG(\mathrm{mod}\, N)$ and $E \equiv FH(\mathrm{mod}\, N)$ to obtain four matrices $C = (c_{ij})_{n\times n}$, $D = (d_{ij})_{n\times n}$, $E = (e_{ij})_{n\times n}$ and $F = (f_{ij})_{n\times n}$. Then he uses the Euclidean algorithm to compute the greatest common divisor of $c_{11} + e_{1n}$ and N, that is, $\gcd(c_{11} + e_{1n},\ N)$. If $\gcd(c_{11} + e_{1n},\ N) = 1$ or N, the attacker argues that the matrices D and F are invalid and should be discarded. Otherwise, $1 < \gcd(c_{11} + e_{1n},\ N) < N$, so he concludes that $\gcd(c_{11} + e_{1n},\ N)$ is a prime factor of N, denoted as p. The attacker can easily obtain the value of q just by doing a simple division, $q = N/p$. At this stage, the attacker possesses the following values $C = (c_{ij})_{n\times n}$, $D = (d_{ij})_{n\times n}$, $E = (e_{ij})_{n\times n}$, $F = (f_{ij})_{n\times n}$, and p, q. Then he can use these values to verify whether the conditions (1) and (2) are satisfied or not. If the conditions are satisfied, the attacker solves the matrix combinatorial problem.

Now we analyze the computational complexity of the algorithm. The algorithm needs to do exhaustive search for the matrices D and F whose elements have a binary length almost the same as that of N. Hence, it needs about $(N^{n^2})^2 = N^{2n^2} = 2^{2n^2 \log_2 N}$ operations to exhaust the matrices D and F. The algorithm also needs to perform the Euclidean

algorithm to calculate the greatest common divisor of $c_{11}+e_{1n}$ and N, which costs about $O(\log_2^3 N)$ operations. After p and q are obtained, the attacker needs to do $2n^2$ divisions to justify the conditions (1) and (2). The $2n^2$ divisions needs about $O(n^2 \log_2^2 N)$ operations. To summarize, we give the computational costs of the algorithm as the product of the aforementioned three costs, that is, $O(2^{2n^2 \log_2 N} n^2 \log_2^5 N)$, a computational complexity exponentially increasing with the input length $\log_2 N$.

4.3.2. *Algorithm II*

A better algorithm for the matrix combinatorial problem may exist. What follows presents such an algorithm.

The algorithm is intended to recover the matrices D and F by determining the row vectors of D and F with an exhaustive search approach. We denote the ith row vector of D and F as $D_i = (d_{i1},\ d_{i2}, \ldots, d_{in})$ and $F_i = (f_{i1},\ f_{i2}, \ldots, f_{in})$, respectively. The attacker does exhaustive search for D_1 and F_n. Then he computes

$$c_{11} = \sum_{i=1}^{n} d_{1i} g_{i1}, \qquad e_{1n} = \sum_{i=1}^{n} f_{ni} h_{in}.$$

From (1), we conclude that if D_1 and F_n is valid, then we must have $p | c_{11} + e_{1n}$. So the attacker can compute the greatest common divisor of $c_{11}+e_{1n}$ and N, $\gcd(c_{11}+e_{1n},\ N)$. If $1 < \gcd(c_{11}+e_{1n},\ N) < N$, the attacker sets $\gcd(c_{11}+e_{1n},\ N) = p$, and thus D_1 and F_n are also recovered. The attacker repeats this procedure to retrieve all the row vectors of D and F, that is, $(D_2,\ F_{n-1}), \ldots, (D_n,\ F_1)$. By doing this, the attacker recovers the matrices D and F. Hence, the combinatorial matrix problem is solved.

Now we analyze the computational complexity of the algorithm. Note that it takes $(N^n)^2 = N^{2n} = 2^{2n \log_2 N}$ bit operations to recover the row vectors D_i and F_{n+1-i}. The Euclidean algorithm to calculate the greatest common divisor of $c_{11} + e_{1n}$ and N costs about $O(\log_2^3 N)$ operations. Hence, the computational complexity to recover every vector pair $(D_i,\ F_{n+1-i})$ is given as $O(2^{2n \log_2 N} \log_2^3 N)$. Hence, the computational costs to recover the matrices D and F should be n times the complexity of $O(2^{2n \log_2 N} \log_2^3 N)$, that is, $O(n 2^{2n \log_2 N} \log_2^3 N)$. The algorithm performs better than *Algorithm I*. However, both algorithms are computationally infeasible.

4.4. *Provable Security Remarks*

The proposed public key cryptographic primitive does not match any provable security objectives. Hence, it cannot be used directly in real life practice. 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. As a public key cryptographic primitive, the proposed cryptosystem needs further studies. In fact, most of the newly-designed public key cryptographic primitives based on new intractability assumptions failed to obtain a provable security goal, for example, the NTRU algorithm proposed by Hoffstein *et al.* (1998), the braid-based public key encryption algorithm

proposed by Ko *et al.* (2000), the knapsack cryptosystems developed recently (Wang *et al.*, 2007; Wang and Hu, 2010), the digital signature schemes using noncommunicative algebraic structures (Sakalauskas, 2004, 2005), the key agreement protocol using conjugacy problem (Sakalauskas *et al.*, 2007). In the authors' point of view, provable security theory does not apply to analyze the security of those PKCs based on new intractable problems. 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. For example, the NTRU algorithm has survived the serious security scrutiny during the last decade, and now cryptographers begin to complete the provable security for the NTRU algorthm (Howgrave-Graham *et al.*, 2003). In some cases, some cryptosystems are built based on some new intractable problems which are not well understood in the literature. For example, the braid-based cryptosystem is based on the conjugator search problem defined over braid groups. However, this cryptographic construction were shown insecure in that it suffices to break the cryptosystem just by solving the braid decomposition problem other than the braid conjugator search problem (Kalka, 2006). This is why we argue that the security for a newly-designed public key cryptographic primitive should be centered on the estimation of the hardness of reversing the encryption function and retrieving the trapdoor information. 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.

5. Conclusions

This paper constructed a fast PKC from a new algebraic–combinatorial problem called the matrix combinatorial problem. The security of the system is carefully studied. No attacks have been found to compromise the security of the proposed cryptosystem. We also carefully examine the hardness of the matrix combinatorial problem by illustrating the computational infeasibilities of several algorithms for the problem. However, we pointed out that the proposed construction is not provably secure. We hope that the readers examine the security of our proposal and if possible provide some paddings for the cryptosystem in order to obtain a provable security argument.

Acknowledgments. The authors would like to thank the anonymous reviewers for their valuable comments on this paper. The authors also thank Tao Liu for the assistance in the implementation of the proposed cryptographic algorithm.

References

Ackermann, P., Kreuzer, M. (2006). Gröbner basis cryptosystems. *Appl. Algebra in Eng., Commun. Comput.*, 17(3–4), 173–194.

Anshel, I., Anshel, M., Goldfeld, D. (1999). An algebraic method for public key cryptography. *Math. Res. Lett.*, 6(3–4), 287–291.

Chor, B., Rivest, R.L. (1988). A knapsack-type public key cryptosystem based on arithmetic in finite fields. *IEEE Trans. Inform. Theory*, 34(5), 901–909.

Coppersmith, D., Shamir, A. (1997). Lattice attack on NTRU. In: *Advances in Cryptology-EuroCrypt'1997*, LNCS, Vol. 1233. Springer, Berlin, pp. 52–61.

Fellows, M.R., Koblitz, N. (1994). Combinatorial cryptosystems galore! *Contemp. Math.*, 168, 51–61.

Hoffstein, J., Pipher, J., Silverman, J.H. (1998). NTRU: a new high speed public key cryptosystem. In: *Proceedings of Algorithm Number Theory (ANTS III)*, LNCS, Vol. 1423. Springer, Berlin, pp. 267–288.

Hofheinz, D., Steinwandt, R. (2002). A differential attack on Polly Cracker. In: *IEEE International Symposium on Information Theory (ISIT 2002)*, Lausanne, Switzerland.

Howgrave-Graham, N., Silverman, J.H., Singer, A., Whyte, W. (2003). NAEP: provable security in the presence of decryption failures. Available at: `http://eprint.iacr.org/2003/172`.

Joux, A. (1998). Lattice reduction: a toolbox for the cryptanalyst. *J. Cryptol.*, 11(3), 161–185.

Kalka, A.G. (2006). Representation attacks on the braid Diffie–Hellman public key encryption. *Appl. Algebra Eng., Commun. Comput.*, 17(3–4), 257–266.

Ko, K.H., Lee, S.J., Cheon, J.H., Han, J.W., Kang, J.S., Park, C. (2000). New public-key cryptosystem using braid groups. In *Advances in Cryptology – Crypto'2000*, LNCS, Vol. 1880. Springer, Berlin, pp. 166–183.

Koblitz, N. (1998). *Algebraic Aspects of Cryptography*. Springer, Berlin.

Lai, M. K. (2003). Knapsack cryptosystems: the past and the future. online manuscript. Available at: `http://www.ics.uci.edu/~mingl/knapsack.html`.

Lenstra, A.K., Lenstra Jr., H.W., Lovász, L. (1982). Factoring polynomials with rational coefficients. *Math. Annua.*, 261(4), 513–534.

Lenstra, A.K., Lenstra, H.W., Manasse, M.S., Pollard, J.M. (1990). The number field sieve. In: *Proceedings 22nd ACM Symposium on Theory of Computing*, Baltimore, Maryland, pp. 564–572.

Liu, J., Huang, S. (2010). Identity-based threshold proxy signature from bilinear pairings. *Informatica*, 21(1), 41–56.

Ly, L.V. (2006). Polly two: a new algebraic polynomial-based public key scheme. *Appl. Algebra Eng., Commun. Comput.*, 17(3–4), 267–283.

Merkle, R.C., Hellman, M.E. (1978). Hiding information and signatures in trapdoor knapsacks. *IEEE Trans. Inform. Theory*, 24(5), 525–530.

Odlyzko, A.M. (1990). The rise and fall of knapsack cryptosystems. In: *Cryptology and Computational Number Theory, Proceedings of Symposia in Applied Mathematics*, Vol. 42. American Mathematics Society, Providence, pp. 75–88.

Sakalauskas, E. (2004). New digital signature scheme in Gaussian monoid. *Informatica*, 15(2), 251–270.

Sakalauskas, E. (2005). One digital dignature dcheme in demimodule over demiring. *Informatica*, 16(3), 383–394.

Sakalauskas, E., Tvarijonas, P., Raulynaitis, A. (2007). Key agreement protocol (KAP) using conjugacy and discrete logarithm problems in group representation level. *Informatica*, 18(1), 115–124.

Shamir, A. (1984). A polynomial-time algorithm for breaking the basic Merkle–Hellman cryptosystem. *IEEE Trans. Inform. Theory*, 30(5), 699–704.

Steinwandt, R., Geiselmann, W. (2002). Cryptanalysis of Polly Cracker. *IEEE Trans. Inform. Theory*, 48(11), 2990–2991.

Steinwandt, R., Geiselmann, W, Endsuleit, R. (2002). Attacking a polynomial-based cryptosystem: Polly Cracker. *Int. J. Inform. Sec.*, 1(3), 143–148.

Sun, X., Li, J., Yin, H., Chen, G. (2010). Delegatability of an identity based strong designated verifier signature scheme. *Informatica*, 21(1), 117–122.

Vaudenay, S. (2001). Cryptanalysis of the Chor–Rivest cryptosystem. *J. Crypt.*, 14, 87–100.

Wang, B., Hu, Y. (2010). Quadratic compact knapsack public-key cryptosystem. *Comput. Math. Appl.*, 59(1), 194–206.

Wang, B., Wu, Q, Hu, Y. (2007). A knapsack-based probabilistic encryption scheme. *Inform. Sci.*, 177(19), 3981–3994.

Wang, Z., Qian, H., Li, Z. (2009). Adaptively secure threshold signature scheme in the standard model. *Informatica*, 20(4), 591–612.

B.C. Wang received his BS degree in computational mathematics and their application software, the MS and PhD degrees in cryptology from Xidian University in 2001, 2004 and 2006 respectively. Currently, he is an associate professor in the Department of Telecommunication Engineerings, Xidian University. His research interests lie in cryptography and network security.

Y.P. Hu is the professor and the doctoral supervisor of Cryptology at Telecommunication Engineerings of Xidian University. His main research areas are cryptology and information security.

Nauja kombinatorinė viešojo rakto kriptosistema

Baocang WANG, Yupu HU

Straipsnyje pasiūlyta kombinatorinė viešojo rakto kriptosistema, kurios sauga priklauso nuo kombinatorinio uždavinio su matricomis sprendimo sudėtingumo. Kriptosistemos privalumai – greitas užšifravimas ir iššifravimas, o trūkumai – užšifruotas tekstas yra palyginus didelis, o raktai, palyginus su RSA, ilgesni. Straipsnyje analizuojama kriptosistema ir parodyta, kad kai kurios atakos skaitmeniškai yra neįvykdomos.