© 2013 *Vilnius University*

Cryptanalysis of a Public Key Cryptosystem Based on the Matrix Combinatorial Problem

Moon Sung LEE
Department of Mathematical Sciences, Seoul National University
Seoul, 151-747, Korea
e-mail: moolee@snu.ac.kr

Received: March 2011; accepted: June 2012

Abstract. In this paper, we present a cryptanalysis of a public key cryptosystem based on the matrix combinatorial problem proposed by Wang and Hu (2010). Using lattice-based methods finding small integer solutions of modular linear equations, we recover the secret key of this cryptosystem for a certain range of parameters. In experiments, for the suggested parameters by Wang and Hu, the secret key can be recovered in seconds.

Keywords: public key cryptosystem, matrix combinatorial problem, lattice, modular linear equation.

1. Introduction

Quite recently, Wang and Hu (2010) proposed a combinatorial public key cryptosystem which we refer to as the WH scheme. The scheme uses matrices of a small dimension as a secret key satisfying certain constraints and disguised using RSA modulus. Its security is based on a certain problem involving matrices named the matrix combinatorial problem whose computational complexity is unknown. The scheme has its advantage in the speed. The encryption and decryption can be done in a quadratic bit complexity.

To be secure, selecting parameters is very important. The WH scheme has a main parameter $(n, |N|)$ with an additional parameter $l_{\mathbf{A}}$ where n is a size of matrices, $|N|$ is a binary length of the RSA modulus N, and $l_{\mathbf{A}}$ is a binary length of entries of a certain secret matrix $\mathbf{A}$.

In this paper, we cryptanalyze the WH scheme and clarify conditions secure parameters should satisfy. We first show that the WH scheme can be completely broken if the factorization of the RSA modulus is given. This answers the question raised in Wang and Hu (2010) whether the factorization of N can be publicized to remove some requirements. Our result shows that the factorization of N should be kept secret. Moreover, we show that the RSA modulus can be factored, thus the scheme can be broken, if $l_{\mathbf{A}}$ is small. We also give a bound of $l_{\mathbf{A}}$ our attack can be applied. This clarifies the condition of parameter $l_{\mathbf{A}}$ which is not considered in Wang and Hu (2010).

Our cryptanalysis uses lattice-based methods finding small solutions of modular linear equations constructed from the public key exploiting the structure of the secret key. Since the suggested parameters use a small $l_{\mathbf{A}}$, they are insecure and the secret key can be recovered within seconds in experiments.

The rest of the paper is organized as follows. In the next section, we review known facts about matrices and lattices. In Section 3, we describe the WH scheme. Our cryptanalysis is presented in Section 4 with experimental results. Concrete parameters secure against our attack is given in Section 5, and we conclude in Section 6.

2. Preliminaries

2.1. *Notations*

Throughout this paper, we use $\mathbb{R}$, $\mathbb{Z}$, and $\mathbb{Z}_N = \{0, \ldots, N-1\}$ to denote the set of real numbers, the set of integers, and the integers modulo N. For any integers $a, b \in \mathbb{Z}$, we use $a \equiv b \pmod N$ to denote that a and b are congruent modulo N. The least nonnegative remainder of a modulo N is denoted as $a \bmod N$. We use $\gcd(a, b)$ to denote the greatest common divisor of a and b, and the symbol $|a|$ represents the binary length of a.

2.2. *Matrices*

We use uppercase bold letters to represent matrices, while lowercase bold letters are used to represent row vectors. We use $\mathbf{M}^T$ to denote the transpose of a matrix $\mathbf{M}$. A column vector is denoted as the transpose of a row vector. We use $\mathbf{I}_n$ and $\mathbf{0}_n$ to denote the identity matrix and the zero matrix of dimension n, respectively.

We define $\mathbf{P}_n$ and $\mathbf{J}_n$ as a cyclic permutation matrix and an exchange matrix of dimension n such that

$$\mathbf{P}_n = \begin{pmatrix} & & & 1 \\ 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{pmatrix} \quad \text{and} \quad \mathbf{J}_n = \begin{pmatrix} & & & 1 \\ & & 1 & \\ & \iddots & & \\ 1 & & & \end{pmatrix}. \tag{1}$$

These two matrices are permutation matrices that are used to permute columns or rows of a matrix. Premultiplying a matrix by $\mathbf{P}_n$ results in the elements of a matrix being circularly shifted down-warded by one position, while postmultiplication by $\mathbf{P}_n$ results in a left circular shift. On the other hand, premultiplication and postmultiplication by $\mathbf{J}_n$ reverses the rows and columns of a matrix, respectively. It is easy to see that $\mathbf{P}_n^n = \mathbf{J}_n^2 = \mathbf{I}_n$.

2.3. *Lattices*

An n-dimensional *lattice* is a set of integer combinations $\{\sum_{i=1}^n x_i \mathbf{b}_i \mid x_i \in \mathbb{Z}\}$ of n linearly independent vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{R}^m$. The set of vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n$ is called a *basis* for the lattice and is represented by a matrix $\mathbf{B}$ whose rows are basis vectors. We use $\mathcal{L}(\mathbf{B})$ to denote a lattice generated by $\mathbf{B}$.

Lattices can be used to solve modular linear equations whose solutions are small integers. For an equation $a_1x_1 + \cdots + a_nx_n \equiv 0 \pmod N$ where a_is and N are known, lattice-based methods are used to find small integer solutions $(x_1, \ldots, x_n)$ whenever $\prod_{i=1}^n X_i \leqslant N$, where the absolute value of x_i is bounded by X_i. Namely, certain lattice basis is constructed from a_is, X_is and N, then lattice basis reduction algorithms are used to find short vectors in the lattice. This folklore result is justified in Herrmann (2008; Appendix A). LLL (Lenstra *et al.*, 1982) and BKZ (Schnorr and Euchner, 1994) algorithms are mostly used lattice basis reduction algorithms in practice. LLL algorithm is implemented in several computer algebra systems including PARI/GP (PARI/GP, 2008) and magma (Bosma *et al.*, 1997). And BKZ algorithm is implemented in NTL (Shoup).

These algorithms can find quite short vectors in the lattice when the dimension n is small, say less than 50. In fact, it is considered to be easy to find shortest vectors in lattices of dimension less than 50 using enumeration techniques (Pohst, 1981; Gama *et al.*, 2010) or sieving methods (Ajtai *et al.*, 2001; Micciancio *et al.*, 2010). It is reported that the shortest vector can be found even in lattices of dimension ≈ 110 (Gama *et al.*, 2010).

It is also interesting that the equation $a_1x_1 + \cdots + a_nx_n \equiv 0 \pmod p$ can be solved with smaller X_i, where p is an unknown divisor of a known composite integer N (Herrmann and May, 2008).

3. The WH Scheme

In this section, we describe the WH scheme, a combinatorial public key cryptosystem due to Wang and Hu (2010).

3.1. *Key Generation*

The protocol involves matrices of an even dimension n and the RSA modulus $N = pq$. Depending on the security level, $(n, |N|) = (2, 1024), (4, 1024)$, and $(4, 2048)$ are suggested for parameters (Wang and Hu, 2010). In the following, we describe the key generation procedure as in Wang and Hu (2010) which involves a 1024-bit RSA modulus.

For the key generation, randomly select a 1024-bit RSA modulus $N = pq$ with primes p and q such that $|p| = |q| = 512$. Randomly choose an n-dimensional invertible matrix $\mathbf{A} = (a_{ij})_{n\times n}$ with $|a_{ij}| = 59$. Randomly choose four matrices $\mathbf{C} = (c_{ij})_{n\times n}$, $\mathbf{D} = (d_{ij})_{n\times n}$, $\mathbf{E} = (e_{ij})_{n\times n}$, and $\mathbf{F} = (f_{ij})_{n\times n}$ with $c_{ij}, d_{ij}, e_{ij}, f_{ij} \in \mathbb{Z}_N$ satisfying the following conditions,

$$\left.\begin{array}{l} c_{ij} + e_{i(n+1-j)} \equiv 0 \\ d_{in} + f_{i1} \equiv 0 \\ d_{ij} + f_{i(j+1)} \equiv 0, \quad j = 1, \ldots, n-1 \end{array}\right\} \pmod p, \quad \text{when } i \text{ is odd,} \tag{2}$$

$$\left.\begin{array}{l} c_{ij} + e_{i(n+1-j)} \equiv 0 \\ d_{in} + f_{i1} \equiv 0 \\ d_{ij} + f_{i(j+1)} \equiv 0, \quad j = 1, \ldots, n-1 \end{array}\right\} \pmod q, \quad \text{when } i \text{ is even,} \tag{3}$$

for $i, j = 1, \ldots, n$. Now generate another matrix $\mathbf{A}' = (a'_{ij})_{n\times n}$, where $a'_{ij} \in \mathbb{Z}_N$ and

$$\begin{array}{lll} a'_{ij} \equiv a_{ij} \pmod{p}, & \text{when } i \text{ is odd,} \\ a'_{ij} \equiv a_{ij} \pmod{q}, & \text{when } i \text{ is even.} \end{array} \tag{4}$$

An additional requirement is that the matrices $\mathbf{A}', \mathbf{C}, \mathbf{D}, \mathbf{E}$, and $\mathbf{F}$ should be invertible modulo N. Finally compute $\mathbf{B}, \mathbf{G}$, and $\mathbf{H}$ as follows,

$$\begin{array}{l} \mathbf{B} = (b_{ij})_{n\times n} = \mathbf{D}^{-1}\mathbf{A}' \bmod N, \\ \mathbf{G} = (g_{ij})_{n\times n} = \mathbf{D}^{-1}\mathbf{C} \bmod N, \\ \mathbf{H} = (h_{ij})_{n\times n} = \mathbf{F}^{-1}\mathbf{E} \bmod N. \end{array} \tag{5}$$

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

Note that $|a_{ij}| = 59$, which is relatively small compared to $|N|$. Since this has an important role in our cryptanalysis, we use $l_{\mathbf{A}}$ to denote that. Thus $l_{\mathbf{A}} = |a_{ij}| = 59$ for the 1024-bit RSA modulus (Wang and Hu, 2010).

3.2. *Encryption*

For the encryption, a plaintext M with $|M| = 450n$ is first divided into n blocks $m_1, \ldots, m_n$ with $|m_i| = 450$. After selecting random integers $r_1, \ldots, r_n, s_1, \ldots, s_n$ in $\mathbb{Z}_N$, the sender computes and sends the ciphertext $(\mathbf{u}, \mathbf{v})$ such that

$$\mathbf{u}^T = \begin{pmatrix} u_1 \\ \vdots \\ u_n \end{pmatrix} \equiv \mathbf{B} \begin{pmatrix} m_1 \\ \vdots \\ m_n \end{pmatrix} + \mathbf{G} \begin{pmatrix} r_1 \\ \vdots \\ r_n \end{pmatrix} + \begin{pmatrix} s_1 \\ \vdots \\ s_n \end{pmatrix} \pmod{N},$$

and

$$\mathbf{v}^T = \begin{pmatrix} v_1 \\ \vdots \\ v_n \end{pmatrix} \equiv \mathbf{H} \begin{pmatrix} r_n \\ r_{n-1} \\ \vdots \\ r_2 \\ r_1 \end{pmatrix} + \begin{pmatrix} s_n \\ s_1 \\ \vdots \\ s_{n-2} \\ s_{n-1} \end{pmatrix} \pmod{N}.$$

3.3. *Decryption*

Given a ciphertext $(\mathbf{u}, \mathbf{v})$, the receiver first computes $\mathbf{t}^T = (t_1, \ldots, t_n)^T = \mathbf{D}\mathbf{u}^T + \mathbf{F}\mathbf{v}^T \bmod N$, sets $w_i = t_i \bmod p$ when i is odd, and $w_i = t_i \bmod q$ when i is even. Then she/he recovers the plaintext M as

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

The decryption works as is shown in Wang and Hu (2010). Note that $l_{\mathbf{A}} = |a_{ij}| = 59$, $|m_i| = 450$, and $|p| = |q| = 512$, which makes the following holds,

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

when n is 2 or 4. This enables the decryption since the computed value $(w_1, \dots, w_n)^T$ is equal to $\mathbf{A}(m_1, \dots, m_n)^T$. Note also $\mathbf{A} = (a_{ij})_{n\times n}$ should have the positive integers as entries.

In the next section, we cryptanalyze the WH scheme.

4. Cryptanalysis of the WH Scheme

In Wang and Hu (2010; Section 4.2), the authors question whether the factorization of N compromises the security of the system or not. In this section, we first show that the factorization of $N = pq$ indeed compromises the security. And then, we present a method how to factor N using public information when $l_{\mathbf{A}}$ is small. The experimental results are presented in Section 4.3.

4.1. *Key Recovery When Factors of the Modulus Are Given*

Using the knowledge of p and q, we first construct modular linear equations which involve $\mathbf{A}$, and then lattice-based methods are used to find small solutions. Once $\mathbf{A}$ is obtained, remaining secrets can be computed easily.

For the ease of a presentation, we use $\mathbf{M}_{[1]}$ and $\mathbf{M}_{[2]}$ to denote sub-matrices of $\mathbf{M}$ formed by odd rows and even rows, respectively. This is useful since odd and even rows are related to p and q, respectively. With this notation, the equations (4,5) are rewritten as in the following,

$$\mathbf{D}_{[1]}\mathbf{B} \equiv \mathbf{A}'_{[1]} \equiv \mathbf{A}_{[1]}, \qquad \mathbf{D}_{[1]}\mathbf{G} \equiv \mathbf{C}_{[1]}, \qquad \mathbf{F}_{[1]}\mathbf{H} \equiv \mathbf{E}_{[1]} \pmod{p}\ , \tag{6}$$

$$\mathbf{D}_{[2]}\mathbf{B} \equiv \mathbf{A}'_{[2]} \equiv \mathbf{A}_{[2]}, \qquad \mathbf{D}_{[2]}\mathbf{G} \equiv \mathbf{C}_{[2]}, \qquad \mathbf{F}_{[2]}\mathbf{H} \equiv \mathbf{E}_{[2]} \pmod{q}\ . \tag{7}$$

Since modulo p equations have the same form as modulo q equations, we only focus on the modulo p equations. Using $\mathbf{B}$ is invertible, the following equation holds:

$$\mathbf{D}_{[1]} \equiv \mathbf{A}_{[1]}\mathbf{B}^{-1} \pmod{p}\ . \tag{8}$$

With permutation matrices $\mathbf{P}_n$ and $\mathbf{J}_n$ from (1), the equation (2) can be rewritten as

$$\mathbf{C}_{[1]} + \mathbf{E}_{[1]}\mathbf{J}_n \equiv \mathbf{0}, \qquad \mathbf{D}_{[1]} + \mathbf{F}_{[1]}\mathbf{P}_n \equiv \mathbf{0} \pmod{p}\ . \tag{9}$$

Multiplying by $\mathbf{P}_n^{-1}$, we obtain the following equation,

$$\mathbf{D}_{[1]}\mathbf{P}_n^{-1} + \mathbf{F}_{[1]} \equiv \mathbf{0} \pmod{p}\ . \tag{10}$$

Now let $\mathbf{Z} = \mathbf{B}^{-1}(\mathbf{G} - \mathbf{P}_n^{-1}\mathbf{H}\mathbf{J}_n) \bmod N$. Then,

$$\begin{aligned}\mathbf{A}_{[1]}\mathbf{Z} &\equiv \mathbf{A}_{[1]}\mathbf{B}^{-1}(\mathbf{G} - \mathbf{P}_n^{-1}\mathbf{H}\mathbf{J}_n) \overset{(8)}{\equiv} \mathbf{D}_{[1]}(\mathbf{G} - \mathbf{P}_n^{-1}\mathbf{H}\mathbf{J}_n)\\ &= \mathbf{D}_{[1]}\mathbf{G} - \mathbf{D}_{[1]}\mathbf{P}_n^{-1}\mathbf{H}\mathbf{J}_n \overset{(10)}{\equiv} \mathbf{D}_{[1]}\mathbf{G} + \mathbf{F}_{[1]}\mathbf{H}\mathbf{J}_n\\ &\overset{(6)}{\equiv} \mathbf{C}_{[1]} + \mathbf{E}_{[1]}\mathbf{J}_n \overset{(9)}{\equiv} \mathbf{0} \pmod p\ .\end{aligned}$$

Thus, the row vectors of $\mathbf{A}_{[1]}$ are solutions of the modular equation

$$\mathbf{x}\mathbf{Z} \equiv \mathbf{0} \pmod p\ , \quad \text{where } \mathbf{Z} = \mathbf{B}^{-1}(\mathbf{G} - \mathbf{P}_n^{-1}\mathbf{H}\mathbf{J}_n). \tag{11}$$

Clearly, $\mathbf{A}_{[2]}\mathbf{Z} \equiv \mathbf{0} \pmod q$ also holds.

From the public key, $\mathbf{Z}(= \mathbf{B}^{-1}(\mathbf{G} - \mathbf{P}_n^{-1}\mathbf{H}\mathbf{J}_n) \bmod N)$ is computed. Then, $\mathbf{Z}'$ is obtained using the Gaussian elimination with elementary column operations such that

$$\mathbf{Z}' = \begin{pmatrix} \mathbf{0}_{n'} & \mathbf{Z}'' \\ \mathbf{0}_{n'} & \mathbf{I}_{n'} \end{pmatrix}, \quad \text{where } n' = n/2. \tag{12}$$

In the above, $\mathbf{Z}'$ contains n' zero columns since $\mathbf{A}_{[1]}$ has n' independent rows which are solutions of (11), therefore solutions of $\mathbf{x}\mathbf{Z}' \equiv \mathbf{0} \pmod p$. To find small integer solutions of this equation, a lattice $\mathcal{L}(\mathbf{L})$ is constructed where a basis matrix $\mathbf{L}$ is

$$\mathbf{L} = \begin{pmatrix} -\mathbf{I}_{n'} & \mathbf{Z}'' \\ & p\mathbf{I}_{n'} \end{pmatrix}.$$

Using lattice basis reduction algorithms, short vectors in this lattice can be found. Those vectors are solutions of (11), therefore can be used as row vectors of $\mathbf{A}_{[1]}$.

Once $\mathbf{A}_{[1]}$ is obtained, $\mathbf{D}_{[1]}$ and $\mathbf{F}_{[1]}$ are obtained using (8) and (10), respectively. Using q instead of p in the above, the rest of the secret $\mathbf{A}_{[2]}, \mathbf{D}_{[2]}$, and $\mathbf{F}_{[2]}$ are obtained similarly. Combining obtained sub-matrices, the secret key $\mathbf{A}, \mathbf{A}^{-1}, \mathbf{D}$, and $\mathbf{F}$ are computed.

The observation here is that we do not need to recover the exact $\mathbf{A}$. As long as it has entries of binary length less than $l_{\mathbf{A}}$, it can be used as a secret key. Due to the small dimension n, we can always find short vectors, thus proper secret key.

Notice that the decryption procedure needs to be changed slightly since the obtained secret $\mathbf{A}$ has the negative values also. To simplify the discussion, we provide a simplified version which works in most of the time. Slight modification on the range of w_i which depends on $\mathbf{A}$ makes the decryption work always which is verified in experiments.

Decryption′. Given a ciphertext $(\mathbf{u}, \mathbf{v})$, the receiver first computes $\mathbf{t}^T = (t_1, \ldots, t_n)^T = \mathbf{D}\mathbf{u}^T + \mathbf{F}\mathbf{v}^T \bmod N$, sets $w_i \equiv t_i \pmod p$ with $w_i \in [-p/2, p/2]$ when i is odd, and $w_i \equiv t_i \pmod q$ with $w_i \in [-q/2, q/2]$ when i is even. Then she/he recovers the plaintext M as $(m_1, \ldots, m_n)^T = \mathbf{A}^{-1}(w_1, \ldots, w_n)^T$.

Our attack succeeds if we can find short vectors of the lattice $\mathcal{L}(L)$ of dimension n. Since n should be quite small due to the efficiency reason, using LLL (Lenstra *et al.*, 1982) and BKZ (Schnorr and Euchner, 1994) algorithms, such short vectors can be found easily. Thus, we claim the following.

Attack 1. *When n is small, say n is less than* 50, *the (equivalent) secret key of the WH scheme with parameter $(n, |N|)$ can be recovered in time polynomial in $|N|$ if the factorization of $N(= pq)$ is known.*

This solves a question raised in Wang and Hu (2010). To be secure, the factorization of N should be kept secret. However, it can be revealed in a certain condition as is described in the following.

4.2. *Factoring the Modulus*

In this section, we explain the method to factor N for certain parameters including the suggested in Wang and Hu (2010). Once the factorization of N is known, the secret key can be recovered as is claimed in Attack 1. Note that any solution $\mathbf{y}$ of the equation $\mathbf{x}\mathbf{Z}' \equiv \mathbf{0} \pmod{p}$ such that $\mathbf{y}\mathbf{Z}' \not\equiv \mathbf{0} \pmod{q}$ can be used to factor N.

The main observation is that $l_{\mathbf{A}}$ is set to be relatively small compared to $|N|$ in Wang and Hu (2010). This value should not be too small to avoid exhaustive search attacks, and should not be too large to avoid large ciphertext expansion ratio which is $\frac{2|N|}{|N|/2 - l_{\mathbf{A}} - \log_2 n - 1} \approx 4.55$ when $l_{\mathbf{A}} = 59$, $|N| = 1024$, and $n = 4$. Using the same notation as in the previous section including $n' = n/2$, we proceed as in the following.

First, recall that $\mathbf{Z}'$ can be computed from the public key and satisfies

$$\mathbf{A}_{[1]}\mathbf{Z}' \equiv \mathbf{0} \pmod{p}, \qquad \mathbf{A}_{[2]}\mathbf{Z}' \equiv \mathbf{0} \pmod{q}, \quad \text{where } \mathbf{Z}' = \begin{pmatrix} \mathbf{0}_{n'} & \mathbf{Z}'' \\ \mathbf{0}_{n'} & \mathbf{I}_{n'} \end{pmatrix}.$$

Now let $\mathbf{A} = (a_{ij})_{n\times n}$ and $\mathbf{Z}'' = (\alpha_{ij})_{n'\times n'}$. Note that α_{ij} are known values. Then the above equation implies the following:

$$\sum_{j=1}^{n'} a_{1j}\alpha_{j1} + a_{1(n'+1)} \equiv 0 \pmod{p},$$
$$\sum_{j=1}^{n'} a_{2j}\alpha_{j1} + a_{2(n'+1)} \equiv 0 \pmod{q}.$$

For the easy presentation, we use an additional variable $\alpha_{(n'+1)1}$, which will be set to 1 in the final equation. Moreover, let $x_j = a_{1j}$ and $y_j = a_{2j}$ for $j = 1, 2, \dots, n'+1$. Using these notations, the above two equations become

$$\sum_{j=1}^{n'+1} x_j\alpha_{j1} \equiv 0 \pmod{p}, \qquad \sum_{j=1}^{n'+1} y_j\alpha_{j1} \equiv 0 \pmod{q}. \tag{13}$$

Since we do not know p and q, we multiply above two equations to obtain a modular linear equation with a known modulus $N = pq$:

$$\sum_{j=1}^{n'+1} x_j y_j {\alpha_{j1}}^2 + \sum_{1 \leqslant i < j \leqslant n'+1} (x_i y_j + x_j y_i)\alpha_{i1}\alpha_{j1} \equiv 0 \pmod{N} .$$

Rearranging the above equation gives

$$\begin{aligned} x_{n'+1} y_{n'+1} {\alpha_{(n'+1)1}}^2 \equiv & -\sum_{j=1}^{n'} x_j y_j {\alpha_{j1}}^2 \\ & - \sum_{1 \leqslant i < j \leqslant n'+1} (x_i y_j + x_j y_i)\alpha_{i1}\alpha_{j1} \pmod{N} . \end{aligned} \tag{14}$$

Now to find x_j and y_j in (14), a lattice basis $\mathbf{L}$ is constructed:

$$\mathbf{L} = \begin{pmatrix} -1 & & & & & & & {\alpha_{11}}^2 \\ & \ddots & & & & & & \vdots \\ & & -1 & & & & & {\alpha_{n'1}}^2 \\ & & & -1 & & & & \alpha_{11}\alpha_{21} \\ & & & & -1 & & & \alpha_{11}\alpha_{31} \\ & & & & & \ddots & & \vdots \\ & & & & & & -1 & \alpha_{n'1}\alpha_{(n'+1)1} \\ & & & & & & & N \end{pmatrix} . \tag{15}$$

From (14), we know that the secret vector

$$\begin{aligned} \mathbf{x} = (& x_1 y_1, \ldots, x_{n'} y_{n'}, x_1 y_2 + x_2 y_1, x_1 y_3 + x_3 y_1, \ldots, \\ & x_{n'} y_{n'+1} + x_{n'+1} y_{n'}, x_{n'+1} y_{n'+1}) \end{aligned} \tag{16}$$

is in the lattice $\mathcal{L}(\mathbf{L})$ whose dimension d is $\frac{(n+2)(n+4)}{8} (= n' + \frac{(n'+1)n'}{2} + 1)$. Thus, if this vector is short, it can be found using lattice-based methods, namely lattice basis reduction (Lenstra *et al.*, 1982; Schnorr and Euchner, 1994) and enumeration (Pohst, 1981; Gama *et al.*, 2010) or sieving (Ajtai *et al.*, 2001; Micciancio *et al.*, 2010). In fact, this vector can be found by an enumeration of all lattice vectors of norm less than $\sqrt{d \cdot (2^{2l_{\mathbf{A}}+1})^2} \approx 2^{2l_{\mathbf{A}}}$. The Gaussian heuristic suggests that the number of lattice vectors of length less than r is approximately $\frac{\mathrm{vol}(\mathcal{B}_d(r))}{\det(L)} \approx \frac{r^d}{N}$. Thus, the estimated number of lattice vectors to be enumerated is about

$$\frac{(2^{2l_{\mathbf{A}}})^d}{N} = \frac{2^{2dl_{\mathbf{A}}}}{N} \approx 2^{2dl_{\mathbf{A}} - |N|}. \tag{17}$$

If $l_{\mathbf{A}} < \frac{|N|}{2d} = \frac{4|N|}{(n+2)(n+4)}$, this value is quite small and the vector $\mathbf{x}$ can be found easily at least for the small dimension d less than 50, which corresponds to n less than 16.

For the suggested parameter $(n, |N|, l_{\mathbf{A}}) = (4, 1024, 59)$, $2dl_{\mathbf{A}} - |N| = 2 \times 6 \times 59 - 1024 = -316$. Thus enumeration does not take much time and the secret vector $\mathbf{x}$ can be found easily as experiments in the next section shows.

Obtaining a short vector satisfying given constraints, factorization of N is easy since $\gcd(\sum_{j=1}^{n'} x_j\alpha_{j1} + x_{n'+1}, N) = p$ from (13). Our method to factor N is summarized:

- Method to factor N
  1. Compute $\mathbf{Z}(= \mathbf{B}^{-1}(\mathbf{G} - \mathbf{P}_n^{-1}\mathbf{H}\mathbf{J}_n) \bmod N)$ from the public key.
  2. Use the Gaussian elimination with elementary column operations to compute $\mathbf{Z}'$ in (12), thus obtain $\mathbf{Z}''$.
  3. From $\mathbf{Z}'' = (\alpha_{ij})$, construct a lattice basis $\mathbf{L}$ in (15).
  4. Reduce $\mathbf{L}$ using LLL, then BKZ algorithms.
  5. Using reduced basis, find a secret vector $\mathbf{x}$ in (16) doing enumeration[1] of all lattice vectors of norm less than $\sqrt{d \cdot (2^{2l_{\mathbf{A}}+1})^2}$ where $d = \frac{(n+2)(n+4)}{8}$.
  6. Factor N computing $\gcd(\sum_{j=1}^{n/2} x_j\alpha_{j1} + x_{n/2+1}, N)$.

Combining with Attack 1, we claim the following.

Attack 2. *When n is small, say n is less than* 16*, the (equivalent) secret key of the WH scheme with a parameter $(n, |N|, l_{\mathbf{A}})$ can be recovered in time polynomial in $|N|$ if $l_{\mathbf{A}} \lesssim \frac{4|N|}{(n+2)(n+4)}$.*

Since the suggested parameters use $n = 2$ or $n = 4$ with small $l_{\mathbf{A}}$, all the suggested parameters are subject to our attack. This is verified in experiments which is shown in the next section.

4.3. *Experimental Results*

We validate our proposed attack to the suggested parameters of the WH scheme with experiments. For each suggested parameters in Wang and Hu (2010), we generated 1,000 instances and tried to recover the secret key. We used the same parameter $l_{\mathbf{A}} = 59$ for $|N| = 1024$ as in Wang and Hu (2010). Since there is no indication for $l_{\mathbf{A}}$ when $|N| = 2048$, we used $l_{\mathbf{A}} = 2 \cdot 59 = 118$, which seems reasonable. The obtained secret key is tested to decrypt 100 ciphertexts generated from random messages, and accepted as a success if all ciphertexts are decrypted correctly.

In experiments, the lattice basis $\mathbf{L}$ in (15) is first reduced using LLL algorithm (Lenstra *et al.*, 1982). Then, instead of enumeration, small linear combinations of the first four vectors of a reduced basis are used to find the secret vector $\mathbf{x}$ in (16) where coefficients are chosen in $\{-5, \ldots, 5\}$ for simplicity. In the result, we could always find a secret vector $\mathbf{x}$. Moreover, the first vector in the reduced basis satisfied the constraints many times, as is shown in the column named "S.P. (1st vector)" in the Table 1, where the total success probability is shown in the column "S.P. (Total)".

[1] Enumeration of short vectors is implemented in magma (Bosma *et al.*, 1997).

Table 1

Success probability of the proposed attack to the suggested parameters in Wang and Hu (2010)

Security level	Parameter $(n, \vert N\vert, l_{\mathbf{A}})$	S.P. (1st vector)	S.P. (Total)
Moderate	$(2, 1024, 59)$	100%	100%
Higher	$(4, 1024, 59)$	73.2%	100%
Highest	$(4, 2048, 118)$	55%	100%

For the implementation, PARI/GP ver. 2.3.5 (PARI/GP, 2008) is used on a CPU Q9550 2.83 GHz with "qflll" for the lattice reduction. And secret keys are obtained within seconds in all cases. Experimental results are summarized in the Table 1 and this shows that the suggested parameters in Wang and Hu (2010) are completely insecure.

5. Discussion

In this section, we suggest concrete parameters for $|N| = 1024$ and $|N| = 2048$ assuming the Gaussian heuristic.

In our attack, all the steps terminate in polynomial time except the enumeration step when there are large number of short lattice vectors. The number of lattice vectors to be enumerated increases exponentially with $l_{\mathbf{A}}$ as is estimated in (17) assuming the Gaussian heuristic. Since our attack needs to enumerate all such vectors, it has time complexity at least $O(2^{2dl_{\mathbf{A}}-|N|})$. For the security parameter λ, we need $2dl_{\mathbf{A}} - |N| > \lambda$, thus:

$$l_{\mathbf{A}} > \frac{|N| + \lambda}{2d} = \frac{4(|N| + \lambda)}{(n+2)(n+4)}.$$

To maintain the main advantage of this cryptosystem which is speed, n should be small. Thus we suggest to use $n = 4$ or $n = 6$. According to NIST recommendations[2], RSA-1024 corresponds to $\lambda = 80$. And we need $l_{\mathbf{A}} > (\lambda + |N|)/(2d) = (80 + 1024)/12 = 92$ when $n = 4$. Considering possible future attacks, we suggest to use $(n, |N|, l_{\mathbf{A}}) = (4, 1024, 115)$. For the RSA-2048 which corresponds to $\lambda = 112$, $l_{\mathbf{A}}$ should be larger than 180 when $n = 4$. Thus we suggest to use $(n, |N|, l_{\mathbf{A}}) = (4, 2048, 225)$ for the higher security. If $n = 4$ is not comfortable, one can use $n = 6$ although this makes encryption and decryption slower. Our suggested parameters are listed in the Table 2.

6. Conclusions

In this paper, we cryptanalyzed the combinatorial public key cryptosystem proposed by Wang and Hu (2010). Exploiting the special structure and property of the secret key, we could recover secret keys within seconds for all the suggested parameters in Wang and Hu

[2] `http://www.keylength.com`.

Table 2

Suggested parameters of the WH scheme

Security	Parameter			Ciphertext
parameter λ	n	$\vert N\vert$	$l_{\mathbf{A}}$	expansion ratio
80	4	1024	115	5.2
112	4	2048	225	5.1
112	6	2048	135	4.6

(2010) using lattice-based methods. Since our attack is based on methods finding "small" integer solutions of modular linear equations, increasing certain parameter makes our attack fail, although it also increases ciphertext expansion ratio slightly.

This cryptosystem has its advantage in the speed of encryption and decryption which is quadratic. However, its relatively large ciphertext expansion ratio (> 4) and newly proposed difficulty assumption makes the scheme less attractive. Further research is needed to prove the assumption, the hardness of the matrix combinatorial problem.

Acknowledgements. This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (No. 2012R1A1A2039129) and by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (No. 2012-0001243). The author would like to thank the anonymous reviewers for their valuable comments which help to improve this paper.

References

Ajtai, M., Kumar, R., Sivakumar, D. (2001). A sieve algorithm for the shortest lattice vector problem. In: *Proceedings of 33rd ACM Symposium on Theory of Computing (STOC)*, pp. 601–610.

Bosma, W., Cannon, J., Playoust, C. (1997). The magma algebra system. I. The user language. *Journal of Symbolic Computation*, 24(3–4), 235–265.

Gama, N., Nguyen, P., Regev, O. (2010). Lattice enumeration using extreme pruning. In: Gilbert, H. (Ed.), *Advances in Cryptology (EUROCRYPT 2010)*, *LNCS*, Vol. 6110. Springer, Berlin, pp. 257–278.

Herrmann, M., May, A. (2008). Solving linear equations modulo divisors: on factoring given any bits. In: Pieprzyk, J. (Ed.), *Advances in Cryptology (ASIACRYPT 2008)*, *LNCS*, Vol. 5350. Springer, Berlin, pp. 406–424.

Lenstra, A., Lenstra, H.W., Lovsz, L. (1982). Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261(4), 515–534.

Micciancio, D., Voulgaris, P. (2010). Faster exponential time algorithms for the shortest vector problem. In: *Proceedings of ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 1468–1480.

The PARI Group Bordeaux PARI/GP, version `2.3.5` (2008).
Available from: `http://pari.math.u-bordeaux.fr/`.

Pohst, M. (1981). On the computation of lattice vectors of minimal length, successive minima and reduced bases with applications. *SIGSAM Bulleten*, 15(1), 37–44.

Schnorr, C.P., Euchner, M. (1994). Lattice basis reduction: improved practical algorithms and solving subset sum problems. *Mathematical Programming*, 66, 181–199.

Shoup, V., Number Theory C++ Library (NTL) version 5.5.2. `http://www.shoup.net/ntl/`.

Wang, B., Hu, Y. (2010). A novel combinatorial public key cryptosystem. *Informatica*, 21(4), 611–626.

M.S. Lee received his PhD degree in mathematical sciences from KAIST in 2009. He is currently a postdoctoral researcher in Seoul National University. His research interests include public key cryptanalysis and lattice-based methods.

Viešojo rakto šifravimo sistemos analizė grįsta matricų kombinatoriniu uždaviniu

Moon Sung LEE

Straipsnyje pasiūlyta viešojo rakto šifravimo sistema, grįsta matricų kombinatoriniu uždaviniu, kurią nagrinėjo Wang ir kt. Naudojant gardelinius metodus, apskaičiuojami modulinių tiesinių lygčių mažiausieji sveikieji sprendiniai ir surandamas šios šifravimo sistemos slaptasis raktas. Wang'o ir kt. pasiūlytiems parametrams slaptasis raktas apskaičiuojamas per kelias sekundes.