Implicit Polynomial Recovery and Cryptanalysis of a Combinatorial Key Cryptosystem

Jun Xu[1,2], Lei Hu[1], and Siwei Sun[1]

[1] State Key Laboratory of Information Security,
Institute of Information Engineering, Chinese Academy of Sciences,
Beijing 100093, China
[2] Graduate University of Chinese Academy of Sciences, Beijing 100049, China
{jxu,hu,swsun}@is.ac.cn

Abstract. A public key cryptosystem based on factoring and a combinatorial problem of matrices over $\mathbb{Z}_N$ proposed in 2010 is analyzed in this paper. We propose an efficient partial private key recovery attack on it by solving a problem of recovering implicit polynomials with small coefficients given their large roots and deriving the large roots from the public key. From the partial information of private key, we can decrypt any ciphertext of the cryptosystem by a simple computation. Our implicit polynomial recovery is an application of lattice basis reduction.

Keywords: Public Key Cryptography, Combinatorial Cryptosystem, Implicit Polynomial Recovery, Lattice, LLL Algorithm.

1 Introduction

Many asymmetric encryption schemes have been proposed after the discovery of public key cryptography, including the well known ones based on number theoretic problems like RSA and ElGamal. It is important for public key cryptography to research secure and fast asymmetric encryption cryptosystems relying on other hard mathematical problems such as lattice problems (e.g., Atjai-Dwork [1], GGH [5] and NTRU [6] public key cryptosystems) and combinatorial problems (e.g., knapsack trapdoors [8]).

Recently, a new combinatorial public key cryptosystem mixed with integer factorization problem were presented [13]. The authors of [13] thought that the security of the system is not dependent on the intractability of integer factorization but on a hard combinatorial problem involving matrices. Some attacks, especially lattice attacks and private key recovery attacks, were stressed and extensively discussed in [13], and the authors concluded that the lattice reduction algorithms do not work for this cryptosystem.

In this paper, we propose a partial private key recovery attack on the above cryptosystem [13] by using the means of lattice. We observe that a secret matrix A in the private key of the cryptosystem has relatively small entries compared with the RSA modulus N. By analyzing the relations between the public and secret matrices in the system, we derive elements in $\mathbb{Z}_N$ which are roots of

T.W. Chim and T.H. Yuen (Eds.): ICICS 2012, LNCS 7618, pp. 45–57, 2012.

some polynomials modulo N with the entries of A as their coefficients. Then we can construct some lattices and run the well known LLL algorithm [7] to recover the relatively small coefficients. This problem of recovering an implicit polynomial with small coefficients given its large roots is a dual of the problem of finding small roots of a polynomial with large coefficients, which is solved by Coppersmith in his seminal paper [3] in 1996. With the recovered matrix A, we can find out the factorization of N and partially recover information on the other secret matrices D and F in the private key. With this partially private key information, an attacker can recover the plaintext of any ciphertext by a very simple computation.

The paper is organized as follows. In Section 2 we give a description for the cryptographic system in [13]. We present our cryptanalysis on it in Section 3. The last section is the conclusion.

2 Description of the Public Key Encryption System

In this section we review the public key encryption scheme proposed in [13].

Key Generation: This cryptosystem involves $n \times n$ matrices over $\mathbb{Z}_N$, where n is an even integer and $N = pq$ is a random 1024-bit RSA modulus with two primes p and q of length of 512 bits. The authors of [13] suggest n is chosen as 2 or 4. Let Γ be the set of all $n \times n$ matrices over $\mathbb{Z}_N$ such that each entry in odd-numbered rows is multiples of p and each entry in the even-numbered rows is multiples of q. Define two $n \times n$ permutation matrices P_1 and P_2 as follows:

$$P_1 = \begin{bmatrix} 0 \cdots 0\ 1 \\ 0 \cdots 1\ 0 \\ \cdots \\ 1 \cdots 0\ 0 \end{bmatrix}, P_2 = \begin{bmatrix} 0 \cdots 0\ 1 \\ 1 \cdots 0\ 0 \\ \cdots \\ 0 \cdots 1\ 0 \end{bmatrix}.$$

Four matrices $C, D, E, F \in \mathbb{Z}^{n \times n}$ are chosen such that

$$C + EP_1 \in \Gamma, \quad D + FP_2 \in \Gamma. \tag{1}$$

Randomly generate an $n \times n$ invertible matrix A over $\mathbb{Z}$ whose all entries have "short" binary length of 59 bits, then generate another matrix $A' \in \mathbb{Z}_N^{n \times n}$ such that $A' - A \in \Gamma$. Randomly choose two invertible matrices D and F in $\mathbb{Z}_N^{n \times n}$, and compute

$$\begin{cases} B \equiv D^{-1}A' \pmod{N}, \\ G \equiv D^{-1}C \pmod{N}, \\ H \equiv F^{-1}E \pmod{N}. \end{cases} \tag{2}$$

Public Key: The RSA modulus N and three matrices B, G and H.
Private Key: The primes p, q and the matrices D, F and A.
Encryption: The plaintext m is coded into an n-dimensional column vector $(m_1, \cdots, m_n)^t$, where each entry m_i is of length of 450 bits. The sender randomly

chooses two n-dimensional vectors $r = (r_1, \cdots, r_n)^t$ and $s = (s_1, \cdots, s_n)^t$ over $\mathbb{Z}_N$. The ciphertext is a 2-tuple (u, v) given as follows:

$$\begin{cases} u \equiv Bm + Gr + s \pmod{N}, \\ v \equiv HP_1 r + P_2 s \pmod{N}. \end{cases} \tag{3}$$

Decryption: Given a ciphertext (u, v), the receiver computes the plaintext m as follows:

$$\begin{cases} t = (t_1, \ldots, t_n)^t \equiv Du + Fv (\bmod\ N), \\ w_i = t_i \bmod p \ \text{ when } i \text{ is odd}, \\ w_i = t_i \bmod q \ \text{ when } i \text{ is even}, \\ m = A^{-1}(w_1, \ldots, w_n)^t. \end{cases}$$

3 Attack on the Public Key Encryption Scheme

In this section we present a partial private key recovery attack on the scheme including: (i) revealing the primes p and q and the secret matrix A by using a lattice basis reduction method. This is done by implicit polynomials recovery; and (ii) getting partial information of the secret matrices D and F. With such partial private key information in hand, a ciphertxet-only attacker can decrypt any ciphertext of this cryptosystem by a simple operation only like the decryption process.

3.1 Recovering Relations on Secret Matrices and Factoring the RSA Modulus

From Formulas (1) and (2) in the key generation, we have

$$DG + FHP_1 \equiv C + EP_1 \pmod{N}, \text{and} \ \ DG + FHP_1 \in \Gamma. \tag{4}$$

Let D_i and F_i denote the i-th rows of D and F respectively. Then

$$\begin{cases} D_i G + F_i HP_1 \equiv 0 \pmod{p} & \text{for odd } i, \\ D_i G + F_i HP_1 \equiv 0 \pmod{q} & \text{for even } i. \end{cases} \tag{5}$$

Since $D + FP_2 \in \Gamma$ by (1), we have

$$\begin{cases} D_i P_2^{-1} + F_i \equiv 0 \pmod{p} & \text{for odd } i, \\ D_i P_2^{-1} + F_i \equiv 0 \pmod{q} & \text{for even } i. \end{cases} \tag{6}$$

By the two above equalities we obtain

$$\begin{cases} D_i (G - P_2^{-1} HP_1) \equiv 0 \pmod{p} & \text{for odd } i, \\ D_i (G - P_2^{-1} HP_1) \equiv 0 \pmod{q} & \text{for even } i. \end{cases} \tag{7}$$

Structure of the Matrix $G - P_2^{-1}HP_1$:

Let $m = n/2$. Clearly, $W := G - P_2^{-1}HP_1$ is a matrix which any attacker can know from the public key. By the first relation of (7), since $D (\bmod\ p)$ is

invertible over $\mathbb{Z}_p$ and chosen at random, and p is a large prime of 512 bits, with a probability very close 1 the remainder of W modulo p has rank m and its first m rows are linearly independent over $\mathbb{Z}_p$. Thus, with this probability we assume

$$W \equiv \begin{pmatrix} W_1 \\ T_1 W_1 \end{pmatrix} (\text{mod } p), \tag{8}$$

where $W_1 \in \mathbb{Z}_p^{m\times n}$ is of rank m over $\mathbb{Z}_p$ and $T_1 \in \mathbb{Z}_p^{m\times m}$. Similarly, with a probability very close 1 we have

$$W \equiv \begin{pmatrix} W_2 \\ T_2 W_2 \end{pmatrix} (\text{mod } q), \tag{9}$$

where the rank of $W_2 \in \mathbb{Z}_q^{m\times n}$ is m over $\mathbb{Z}_q$ and $T_2 \in \mathbb{Z}_q^{m\times m}$.

By the Chinese remainder theorem, there is an $m \times n$ matrix $\widetilde{W}$ over $\mathbb{Z}_N$ such that $\widetilde{W} = W_1 (\text{mod } p)$ and $\widetilde{W} = W_2 (\text{mod } q)$, and there is also an $m \times m$ matrix T over $\mathbb{Z}_N$ such that $T = W_1 (\text{mod } p)$ and $T = W_2 (\text{mod } q)$. Then from (8) and (9), we get

$$G - P_2^{-1} H P_1 \equiv \begin{pmatrix} \widetilde{W} \\ T\widetilde{W} \end{pmatrix} (\text{mod } N). \tag{10}$$

Relations on the Secret Matrix D:

Consider the block submatrices of D. Let P_3 denote an $n \times n$ permutation matrix which transforms a column vector $(x_1, x_2, \cdots, x_n)^t$ into $(x_1, x_3, \cdots, x_{n-1}, x_2, x_4, \cdots, x_n)^t$, and Δ be the set of all $n \times n$ matrices over $\mathbb{Z}_N$ such that all entries in the first m rows are multiples of p and all entries in the last m rows are multiples of q. Then, an $n \times n$ matrix D' over $\mathbb{Z}_N$ is in Γ if and only if $P_3 D' \in \Delta$.

By (7),

$$P_3 D (G - P_2^{-1} H P_1) \in \Delta. \tag{11}$$

Evenly partition $P_3 D$ into

$$P_3 D = \begin{pmatrix} D^{(1)} & D^{(2)} \\ D^{(3)} & D^{(4)} \end{pmatrix},$$

where $D^{(1)}, D^{(2)}, D^{(3)}, D^{(4)}$ are four $m \times m$ matrices over $\mathbb{Z}_N$. Plugging (10) into (11), we have

$$\begin{pmatrix} D^{(1)} & D^{(2)} \\ D^{(3)} & D^{(4)} \end{pmatrix} \begin{pmatrix} \widetilde{W} \\ T\widetilde{W} \end{pmatrix} \in \Delta,$$

and in other words,

$$\begin{cases} (D^{(1)} + D^{(2)} T)\widetilde{W} \equiv 0 \ (\text{mod } p), \\ (D^{(3)} + D^{(4)} T)\widetilde{W} \equiv 0 \ (\text{mod } q). \end{cases}$$

Since $\widetilde{W} \pmod p$ and $\widetilde{W} \pmod q$ are both of rank m, we have

$$\begin{cases} D^{(1)} \equiv -D^{(2)}T \pmod p, \\ D^{(3)} \equiv -D^{(4)}T \pmod q. \end{cases} \tag{12}$$

Relations Related on the Secret Matrix A:

By the key generation, we have $DB - A = A' - A \in \Gamma$ and

$$P_3DB - P_3A \in \Delta.$$

Let

$$P_3A = \begin{pmatrix} A^{(1)} & A^{(2)} \\ A^{(3)} & A^{(4)} \end{pmatrix}, B = \begin{pmatrix} B^{(1)} & B^{(2)} \\ B^{(3)} & B^{(4)} \end{pmatrix}$$

be the even partitions of matrices. The above relation says that

$$\begin{pmatrix} D^{(1)} & D^{(2)} \\ D^{(3)} & D^{(4)} \end{pmatrix} \begin{pmatrix} B^{(1)} & B^{(2)} \\ B^{(3)} & B^{(4)} \end{pmatrix} - \begin{pmatrix} A^{(1)} & A^{(2)} \\ A^{(3)} & A^{(4)} \end{pmatrix} \in \Delta,$$

and equivalently,

$$\begin{cases} D^{(1)}B^{(1)} + D^{(2)}B^{(3)} \equiv A^{(1)} \pmod p, \\ D^{(1)}B^{(2)} + D^{(2)}B^{(4)} \equiv A^{(2)} \pmod p, \\ D^{(3)}B^{(1)} + D^{(4)}B^{(3)} \equiv A^{(3)} \pmod q, \\ D^{(3)}B^{(2)} + D^{(4)}B^{(4)} \equiv A^{(4)} \pmod q. \end{cases} \tag{13}$$

Plugging (12) into (13), we have

$$\begin{cases} D^{(2)}(B^{(3)} - TB^{(1)}) \equiv A^{(1)} \pmod p, \\ D^{(2)}(B^{(4)} - TB^{(2)}) \equiv A^{(2)} \pmod p, \\ D^{(4)}(B^{(3)} - TB^{(1)}) \equiv A^{(3)} \pmod q, \\ D^{(4)}(B^{(4)} - TB^{(2)}) \equiv A^{(4)} \pmod q. \end{cases} \tag{14}$$

Again with a probability very close to 1, $A^{(1)}$ and $A^{(2)}$ are invertible over $\mathbb{Z}_p$ and over $\mathbb{Z}_q$.

Let $K = (B^{(3)} - TB^{(1)})^{-1}(B^{(4)} - TB^{(2)}) \pmod N$, which is an $m \times m$ matrix and can be computed from the public key N, G, H, B. Then the relations in (14) lead to:

$$\begin{cases} A^{(2)} \equiv A^{(1)}K \pmod p, \\ A^{(4)} \equiv A^{(3)}K \pmod q. \end{cases} \tag{15}$$

These are implicit relations related on the secret matrix A and the secret primes p and q.

Now we go to recover these implicit relations by using the fact that A is a relatively small matrix, and then find the secret primes. To simplify the notations and illustrate the principle, below we first consider the simplest case that $n = 2$.

The method is similar for other cases of n but involves higher dimensional lattices and further skills.

Recovering A and Factoring N when $n = 2$:

For $n = 2$, P_3 is the identity matrix,

$$P_3A = \begin{pmatrix} A^{(1)} & A^{(2)} \\ A^{(3)} & A^{(4)} \end{pmatrix} = A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}, P_3B = \begin{pmatrix} B^{(1)} & B^{(2)} \\ B^{(3)} & B^{(4)} \end{pmatrix} = B = \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix},$$

and $K = (b_{21} - Tb_{11})^{-1}(b_{22} - Tb_{12}) \pmod N$. From the pair of relations modulo p and q in (15), we get a relation modulo N as $(a_{12} - Ka_{11})(a_{22} - Ka_{21}) \equiv 0 \pmod N$, which is

$$a_{11}a_{21}K^2 - (a_{11}a_{22} + a_{12}a_{21})K + a_{12}a_{22} \equiv 0 \pmod N. \tag{16}$$

Note that the a_{ij} are of length of not exceeding 59 bits, the sizes of the coefficients in (16) (namely $a_{11}a_{21}, a_{11}a_{22}+a_{12}a_{21}, a_{12}a_{22}$) are not more than 119 bits, which are relatively small integers compared with the 1024-bit modulus N. This is a problem of recovering an implicit polynomial with small coefficients given its large roots. See Appendix A for its general description and a solution. It can be regarded as a dual of the problem of finding small roots of a polynomial with large coefficients, which had been solved by Coppersmith in his seminal paper [3] in 1996 by the well known lattice basis reduction method. For our problem, we can recover the small coefficients also by the lattice means as follows.

Construct a matrix as

$$\begin{pmatrix} 1 & 0 & -K^2 \pmod N \\ 0 & 1 & K \\ 0 & 0 & N \end{pmatrix}$$

and let L_1 be the three-dimensional lattice spanned by its rows. Run the LLL lattice basis reduction algorithm and get a short lattice vector (a, b, c) in the lattice. Obviously, all lattice vectors (a, b, c) satisfy that $aK^2 - bK + c \equiv 0 \pmod N$. Since the lattice is of very low dimension like 3, we almost always obtain the shortest vector (a, b, c) in L_1.

Note that $a_{12}/a_{11} \pmod N$ and $a_{22}/a_{21} \pmod N$ are two roots of (16), from this we know that $(a_{11}a_{21}, a_{11}a_{22} + a_{12}a_{21}, a_{12}a_{22})$ and (a, b, c) are proportional modulo N. They are relatively very small with respect to N, so they must be proportional in the usual sense. Assume that $(a_{11}a_{21}, a_{11}a_{22} + a_{12}a_{21}, a_{12}a_{22}) = t(a, b, c)$. This t is a small integer, we can exhaust to search it. In fact, $t = 1$ holds with a probability of about 39 percent (see the corollary in Appendix B).

For each small searched value of t such that ta, tb, tc are of length of not exceeding 119 bits, factor ta and tc as $ta = x_1x_3$ and $tc = x_2x_4$ with $x_1, \cdots, x_4$ of no more than 59 bits. We note that integers of lengths of less than two hundreds bits like 119 bits are very easy to factor by using the open source software like Shoup's number theoretical library NTL [11] or Magma [2]. From the complete decomposition of ta and tc, there may be several choices for these x_i, and hence

we test whether $x_1x_4 + x_2x_3 = tb$ holds or not. If yes, we let $(a_{11}, a_{12}, a_{21}, a_{22}) = (x_1, x_2, x_3, x_4)$ and compute $\gcd(a_{12} - Ka_{11}, N)$. This will generally get p by $\gcd(a_{12} - Ka_{11}, N) = p$.

Experimental Result for $n = 2$: We have implemented the above attack with the LLL algorithm on a PC with Intel(R) Core(TM) Quad CPU (2.83GHz, 3.25GB RAM, Windows XP). For $n = 2$, our experiment always successfully outputs the shortest vector (a, b, c), and amongst 100 instances randomly generated, there are 40 times that t is equal to 1. The time complexity is very low, all work including the lattice computation and decomposition test can be finished within ten seconds.

Recovering A and Factoring N when $n = 4$:

For $n = 4$, let

$$K = \begin{pmatrix} k_1 & k_2 \\ k_3 & k_4 \end{pmatrix}, A^{(i)} = \begin{pmatrix} a_i & b_i \\ a_i^{'} & b_i^{'} \end{pmatrix}, 1 \le i \le 4.$$

The relations in (15) then become:

$$\begin{cases} k_1a_1 + k_3b_1 - a_2 \equiv 0 \pmod{p}, \\ k_2a_1 + k_4b_1 - b_2 \equiv 0 \pmod{p}, \\ k_1a_3 + k_3b_3 - a_4 \equiv 0 \pmod{q}, \\ k_2a_3 + k_4b_3 - b_4 \equiv 0 \pmod{q}. \end{cases} \tag{17}$$

From the pair of relations modulo p and q in (17), we get relations modulo N as

$$\begin{cases} k_1^2(a_1a_3) + k_3^2(b_1b_3) + k_1k_3(a_1b_3 + a_3b_1) \\ -k_1(a_1a_4 + a_2a_3) - k_3(a_2b_3 + a_4b_1) + a_2a_4 \equiv 0 \pmod{N}, \\ k_2^2(a_1a_3) + k_4^2(b_1b_3) + k_2k_4(a_1b_3 + a_3b_1) \\ -k_2(a_1b_4 + a_3a_2) - k_4(b_2b_3 + b_1b_4) + b_2b_4 \equiv 0 \pmod{N}. \end{cases} \tag{18}$$

Obviously, we can get the same relations on $a_i^{'}$, $b_i^{'}$ $(1 \le i \le 4)$ as (18) from

$$\begin{cases} k_1a_1^{'} + k_3b_1^{'} - a_2^{'} \equiv 0 \pmod{p}, \\ k_2a_1^{'} + k_4b_1^{'} - b_2^{'} \equiv 0 \pmod{p}, \\ k_1a_3^{'} + k_3b_3^{'} - a_4^{'} \equiv 0 \pmod{q}, \\ k_2a_3^{'} + k_4b_3^{'} - b_4^{'} \equiv 0 \pmod{q}. \end{cases} \tag{19}$$

Construct two six-dimensional lattices L_2 and L_3 which are generated by the rows of the matrices

$$\begin{pmatrix} 1 & & & -k_1^2 \pmod{N} \\ & 1 & & -k_3^2 \pmod{N} \\ & & 1 & -k_1k_3 \pmod{N} \\ & & & k_1 \\ & & & k_3 \\ & & & N \end{pmatrix} \text{ and } \begin{pmatrix} 1 & & & -k_2^2 \pmod{N} \\ & 1 & & -k_4^2 \pmod{N} \\ & & 1 & -k_2k_4 \pmod{N} \\ & & & k_2 \\ & & & k_4 \\ & & & N \end{pmatrix}$$

respectively. Running the LLL algorithm, we get a reduced basis of L_2, $\{\alpha_1, \cdots, \alpha_6\}$, and a reduced basis of L_3, $\{\beta_1, \cdots, \beta_6\}$ (the vectors in a basis are listed in the increasing length order).

Note that the following vectors which we desire to find

$$\begin{cases} (a_1a_3, b_1b_3, a_1b_3 + a_3b_1, a_1a_4 + a_2a_3, a_2b_3 + a_4b_1, a_2a_4) \in L_2, \\ (a_1a_3, b_1b_3, a_1b_3 + a_3b_1, a_1b_4 + b_2a_3, b_2b_3 + b_4b_1, b_2b_4) \in L_3, \\ (a_1'a_3', b_1'b_3', a_2'b_3' + a_3'b_1', a_1'a_4' + a_2'a_3', a_2'b_3' + a_4'b_1', a_2'a_4') \in L_2, \\ (a_1'a_3', b_1'b_3', a_2'b_3' + a_3'b_1', a_1'b_4' + b_2'a_3', b_2'b_3' + b_4'b_1', b_2'b_4') \in L_3 \end{cases} \tag{20}$$

are of sizes less than $2^{118} \cdot \sqrt{3 \cdot 1^2 + 3 \cdot 2^2} < 2^{120}$, which are relatively very short vectors compared with most vectors in L_2 and L_3 with 1024-bit components.

Let

$$\begin{cases} (a_1a_3, b_1b_3, a_1b_3 + a_3b_1, a_1a_4 + a_2a_3, a_2b_3 + a_4b_1, a_2a_4) = x_1\alpha_1 + \cdots + x_6\alpha_6, \\ (a_1a_3, b_1b_3, a_1b_3 + a_3b_1, a_1b_4 + b_2a_3, b_2b_3 + b_4b_1, b_2b_4) = y_1\beta_1 + \cdots + y_6\beta_6, \end{cases} \tag{21}$$

where $x_i, y_i \in \mathbb{Z}$, $1 \le i \le 6$. In our experiment (see below), we observe that to search short vectors $x_1\alpha_1 + \cdots + x_6\alpha_6$ and $y_1\beta_1 + \cdots + y_6\beta_6$ in the lattices, only the first three or four coefficients in the two tuples of coefficients, $x_1, \cdots, x_6$ and $y_1, \cdots, y_6$ are not zero and they are always very small integers like ones with absolute values less than 50. This is reasonable because under the increasing length order, the last two vectors, α_5 and α_6, or β_5 and β_6, are obviously much longer than the first several vectors $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ and $\beta_1, \beta_2, \beta_3, \beta_4$. Do an exhaust lexicographical-like search for the integral coefficient vector $(x_1, \cdots, x_6)$ with x_6 and x_5 being set to 0 prior and all other components starting from 0 to a small number like 50 (in absolute value sense), and find all linear combinations $x_1\alpha_1 + \cdots + x_6\alpha_6$ of length less than 2^{120}. More precisely, we require that the first, second and last components of these vectors are all of length not exceeding 118 bits and the other components are of length not exceeding 119 bits.

Further, note that for the desired two vectors in (21), let $(c_1 \cdots, c_6)$ be either one of them, then $c_3^2 - 4c_1c_2 = (a_1b_3 + a_3b_1)^2 - 4a_1a_3b_1b_3 = (a_1b_3 - a_3b_1)^2$ and similarly $c_4^2 - 4c_1c_6, c_5^2 - 4c_2c_6$ are complete square numbers over integers. If one of $c_3^2 - 4c_1c_2, c_4^2 - 4c_1c_6, c_5^2 - 4c_2c_6$ is not a complete square number, then the vector $(c_1 \cdots, c_6)$ is not a desired vector of the form (21). Otherwise, by continuing to find square roots of the completely square numbers, we get the intended values for a_1b_3 and a_3b_1 from the values of $a_1b_3 + a_3b_1$ and $a_1b_3 - a_3b_1$, and similarly get the intended values for $a_1a_4, a_3a_2, b_1a_4, b_3a_2, a_1b_4, a_3b_2, b_1b_4, b_3b_2$. All these values obtained should be of length not exceeding 118 bits, if any one of such requirements invalidates, then the vector $(c_1 \cdots, c_6)$ can not be a desired vector of the form (21).

Search all linear combinations $x_1\alpha_1 + \cdots + x_6\alpha_6$ satisfying all requirements mentioned above, and let them form a set S_2. Similarly in the lattice L_3, search all vectors $y_1\beta_1 + \cdots + y_6\beta_6$ with the same restrictions and then form a set S_3. Typically in our experiment, the cardinalities of S_2 and S_3 are less than 500.

Now for the desired two vectors in (21), the first three components are pairwisely identical. This tells us that by simply finding "projective collisions" of S_2

and S_3, we will find all vectors in S_2 such that their first three components are equal to the corresponding components of some vectors in S_3. In our experiment, there are typically less than 200 such collisions. For any one of such three dimensional vectors, if it is a desired one which can be the projection of the vectors in (21), then the values obtained for $a_1a_3, b_1b_3, a_2a_4, b_2b_4$ and the intended values obtained for $a_1b_3, a_3b_1, a_1a_4, a_3a_2, b_1a_4, b_3a_2, a_1b_4, a_3b_2, b_1b_4, b_3b_2$ should satisfy many division relations like the following

$$a_1a_3 | \gcd(a_1b_3, a_1a_4, a_1b_4) \cdot \gcd(b_1a_3, a_2a_3, b_2a_3).$$

If any one of such relations invalidates, then we discard the collision. Otherwise, we can further try to find the values for $a_1, \cdots, a_4, b_1, \cdots, b_4$ as follows.

By using the fact that a_1 divides $\gcd(a_1a_3, a_1b_3, a_1a_4, a_1b_4)$ and many similar relations hold, factoring some of 14 products for

$$a_1a_3, b_1b_3, a_2a_4, b_2b_4, a_1b_3, a_3b_1, a_1a_4, a_3a_2, b_1a_4, b_3a_2, a_1b_4, a_3b_2, b_1b_4, b_3b_2,$$

which are all of lengths not exceeding 118 bits and are easy to factor as shown in the previous subsection about the case of $n = 2$, we will get at once the values for $a_1, \cdots, a_4, b_1, \cdots, b_4$ and distill out improper candidates that the values for $a_1, \cdots, a_4, b_1, \cdots, b_4$ are not less than 2^{59}. Finally, we get all proper candidates for $a_1, \cdots, a_4, b_1, \cdots, b_4$. In our experiment, there are typically less than 100 such candidates.

Now we have found out all candidate tuples for $(a_1, a_2, a_3, a_4, b_1, b_2, b_3, b_4)$. From any such tuple, we get the factorization of N by computing $p = \gcd(k_1a_1 + k_3b_1 - a_2, N)$. These candidate tuples are also suitable for $(a'_1, a'_2, a'_3, a'_4, b'_1, b'_2, b'_3, b'_4)$ since they both satisfy the completely same requirements, and they can not be distinguished. However, fortunately, there are only few such candidate tuples (let the number of candidate tuples be l), and that means there are at most $l(l-1)$ choices for the invertible matrix A. In our experiment, l is always less than 30. Thus, we have recovered the secret matrix A in the sense that we limit it into a small range. The $l(l-1)$ possibilities for A can be further removed out or namely we can further fix the choice after doing one or few proper ciphertect-only decryptions, see Subsection 3.3 below.

As mentioned in the case of $n = 2$, if the entries of the original $(a_1, a_2, a_3, a_4, b_1, b_2, b_3, b_4)$ are small and this vector has a multiple whose all entries are of length not exceeding 59 bits, then there may be several candidates for A, however, this happens with a much lower probability than in the case of $n = 2$ (See the proposition of Appendix B).

Experimental Result for $n = 4$: In the search of short vectors in L_2, the last two coefficients of the integral linear combinations $x_1\alpha_1 + \cdots + x_6\alpha_6$, x_5 and x_6, are always zero, and in many cases x_4 is also zero. While for other coefficients, they are always less than 50 in absolute values. A similar situation happens for L_3. The whole computation time for finding the $(a_1, a_2, a_3, a_4, b_1, b_2, b_3, b_4)$ including lattice computing and factorization is within two hours.

3.2 Partial Information Recovery of Secret Matrices D and F

When $n = 2$, p and q are factored out and the matrix A is found, the secret matrix D can be partially recovered as

$$\begin{cases} (D^{(1)}, D^{(2)}) \equiv (A^{(1)}, A^{(2)})B^{-1} \pmod{p}, \\ (D^{(3)}, D^{(4)}) \equiv (A^{(3)}, A^{(4)})B^{-1} \pmod{q}, \end{cases} \tag{22}$$

by (16). Although we do not completely know what is D, we have gotten its half information by (22). The similarity does for F by the fact that $D + FP_2 \in \Gamma$, and this suffices to mount a ciphertext-only attack. See Subsection 3.3.

For $n = 4$, p and q are revealed and there are at most $l^2 - l$ possibilities for A, recall the process of key generation, we have

$$\begin{cases} D_i \equiv A_i B^{-1} \pmod{p} & \text{when } i \text{ is odd}, \\ D_i \equiv A_i B^{-1} \pmod{q} & \text{when } i \text{ is even}. \end{cases} \tag{23}$$

Once we select some possibility for A, we can get half information of D by (23). Similarly, since $D + FP_2 \in \Gamma$, we can also obtain its half information of F.

3.3 Ciphertext-Only Attack

Recall the decryption process,

$$\begin{cases} t = (t_1, \ldots, t_n)^t \equiv Du + Fv \pmod{N}, \\ w_i = t_i \bmod p & \text{when } i \text{ is odd}, \\ w_i = t_i \bmod q & \text{when } i \text{ is even}. \end{cases}$$

When $n = 2$, we have

$$\begin{cases} w_i = \big((D_i \bmod p)\ u + (F_i \bmod p)\ v\big) \pmod{p} & \text{when } i \text{ is odd}, \\ w_i = \big((D_i \bmod q)\ u + (F_i \bmod q)\ v\big) \pmod{q} & \text{when } i \text{ is even}, \end{cases}$$

so the plaintext is completely recovered as $m = (m_1, \cdots, m_n)^t = A^{-1}(w_1, \ldots, w_n)^t$.

For $n = 4$, although there are probably $l(l-1)$ choices for the secret matrix A and for the partial information of D and F, we can try each possibility to decrypt the plaintext. If a meaningful information for the plaintext is recovered, then we find out a proper choice for the secret matrices A, D and F. Thus, by doing one or few proper ciphertext-only decryptions, we in fact recover the secret matrix A and fix the choice.

4 Conclusion

In this paper, we proposed an efficient partial private key recovery on the combinatorial public key cryptosystem recently proposed in [13]. The partial information recovery of private key is sufficient to decrypt any ciphertext of the cryptosystem in a simple computation.

We recover the partial information of private keys in the cryptosystem by solving a problem of recovering implicit polynomials with small coefficients given their large roots, and the large roots are derived from the public key. The problem of recovering an implicit polynomial with small coefficients can be regarded as a dual of the problem of finding small roots of a polynomial with large coefficients, and these two problems were solved respectively by Coppersmith in [3] in 1996 and in this paper by the lattice basis reduction method.

Acknowledgements: The work of this paper was supported by the National Natural Science Foundation of China (Grants 61070172 and 10990011), the Strategic Priority Research Program of Chinese Academy of Sciences under Grant XDA06010702, NBRPC 2011CB302400, and the State Key Laboratory of Information Security, Chinese Academy of Sciences.

References

1. Ajtai, M., Dwork, C.: A public-key cryptosystem with worst-case/average-case equivalence. In: Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing, pp. 284–293 (1997)
2. Bosma, W., Cannon, J., Playoust, C.: The Magma Algebra System I: The user language. Journal of Symbolic Computation 24, 235–265 (1997)
3. Coppersmith, D.: Finding a Small Root of a Univariate Modular Equation. In: Maurer, U.M. (ed.) EUROCRYPT 1996. LNCS, vol. 1070, pp. 155–165. Springer, Heidelberg (1996)
4. Coppersmith, D., Franklin, M., Patarin, J., Reiter, M.: Low-Exponent RSA with Related Messages. In: Maurer, U.M. (ed.) EUROCRYPT 1996. LNCS, vol. 1070, pp. 1–9. Springer, Heidelberg (1996)
5. Goldreich, O., Goldwasser, S., Halvei, S.: Public-Key Cryptosystems from Lattice Reduction Problems. In: Kaliski Jr., B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 112–131. Springer, Heidelberg (1997)
6. Hoffstein, J., Pipher, J., Silverman, J.H.: NTRU: A Ring-Based Public Key Cryptosystem. In: Buhler, J.P. (ed.) ANTS 1998. LNCS, vol. 1423, pp. 267–288. Springer, Heidelberg (1998)
7. Lenstra, A.K., Lenstra Jr., H.W., Lovász, L.: Factoring polynomials with rational coefficients. Mathematische Annalen 261, 515–534 (1982)
8. Merkle, R.C., Hellman, M.E.: Hiding Information and Signatures in Trapdoor Knapsack. IEEE Transaction on Information Theory 24, 525–530 (1978)
9. Nguyen, P.Q., Stern, J.: Cryptanalysis of the Ajtai-Dwork Cryptosystem. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 223–242. Springer, Heidelberg (1998)
10. Odlyzko, A.M.: The rise and fall of knapsack cryptosystems. Cryptology and Computational Number Theory 42, 75–88 (1990)
11. Shoup, V.: A library for doing number theory, `http://www.shoup.net/ntl`
12. Wang, B., Hu, Y.: Diophantine Approximation Attack on a Fast Public Key Cryptosystem. In: Chen, K., Deng, R., Lai, X., Zhou, J. (eds.) ISPEC 2006. LNCS, vol. 3903, pp. 25–32. Springer, Heidelberg (2006)
13. Wang, B., Hu, Y.: A Novel Combinatorial Public Key Cryptosystem. Informatica 21(4), 611–626 (2010)
14. Zwillinger, D.(editor in chief): CRC Standard Mathematical Tables and Formulae, 30th edn. CRC Press, Boca Raton (1996)

A Recovering Implicit Polynomials with Small Coefficients

Problem. Assume $f(x_1, \cdots, x_n) \in \mathbb{Z}[x_1, \cdots, x_s]$ is a polynomial with unknown coefficients and absolute values of these coefficients are relatively small compared to some large integer N. Given $a_1, \cdots, a_s \in \mathbb{Z}_N$ such that $f(a_1, \cdots, a_s) \equiv 0 \pmod{N}$. We expect to recover $f(x_1, \cdots, x_s)$.

Solution: Let $f(x_1, \cdots, x_s) = \sum_{(k_1, \cdots, k_s)} c_{k_1, \cdots, k_s} x_1^{k_1} \cdots x_s^{k_s}$. Then we have $\sum c_{k_1, \cdots, k_s} a_1^{k_1} \cdots a_s^{k_s} \equiv 0 \pmod{N}$. Construct a lattice L which is generated by the rows of the matrix $\begin{pmatrix} I & \alpha \\ 0 & N \end{pmatrix}$, where I is the identity matrix whose numbers of rows and columns are equal to the number of nonzero coefficients $c_{k_1, \cdots, k_s}$ with $(k_1, \cdots, k_s) \neq (0, \cdots, 0)$, and α is a column vector whose entry at the position labeled by $(k_1, \cdots, k_s)$ is equal to $a_1^{k_1} \cdots a_s^{k_s} \pmod{N}$. A lattice vector $(\cdots, \widetilde{a}_{k_1, \cdots, k_s}, \cdots, \widetilde{a}_{0, \cdots, 0})$ in L satisfies $\sum_{(k_1, \cdots, k_s) \neq (0, \cdots, 0)} \widetilde{a}_{k_1, \cdots, k_s} a_1^{k_1} \cdots a_s^{k_s} \equiv \widetilde{a}_{0, \cdots, 0} \pmod{N}$, it results in a solution for the problem. Running the LLL algorithm for L, we may find out a small solution for the problem.

B Probability That Several Random Integers Are Coprime

Proposition. Let N be a large positive integer and $l \geq 2$ be an integer. The probability that l integers which are chosen uniformly at random and independently in the interval $[1, N]$ are coprime is about $\prod_{\text{prime } r \leq N} (1 - \frac{1}{r^l})$, where the product is taken over all primes r not exceeding N. If $l = 2$, then this probability is about 0.6181. If $l = 8$, this probability is about 0.9959.

Proof: Set $S_N = \{1, 2, \cdots, N\}$ and let $S_N^l = S_N \times \cdots \times S_N$ be the Descartesian product of l copies of S_N. Then the set $S_N^l - \{(a_1, \cdots, a_l) \in S_N^l : \gcd(a_1, \cdots, a_l) = 1\}$ is equal to $\bigcup_{2 \leq k \leq N} (kS_{[\frac{N}{k}]})^l$. We restrict the index k in the union is square-free, that is, k is a product of distinct primes. For such integers, define $\rho_k = (-1)^u$ if k is a product of u distinct primes. By the inclusion-exclusion principle, the cardinality of the above union is equal to $\sum_{2 \leq k \leq N} (-\rho_k) \left[\frac{N}{k}\right]^l$, and hence, the number of pairs of coprime integers in S_N is equal to

$$\begin{aligned} & N^l + \sum_{2 \leq k \leq N} \rho_k \left[\tfrac{N}{k}\right]^l \approx \sum_{1 \leq k \leq N} \rho_k \left(\tfrac{N}{k}\right)^l \\ \approx\ & N^l (1 - \tfrac{1}{2^l} - \tfrac{1}{3^l} - \tfrac{1}{5^l} + \tfrac{1}{6^l} - \tfrac{1}{7^l} + \cdots) \\ \approx\ & N^l \prod_{\text{prime } r \leq N} (1 - \tfrac{1}{r^l}) \approx N^l \prod_{\text{prime } r} (1 - \tfrac{1}{r^l}). \end{aligned}$$

When $l = 2$, $\prod_{\text{prime } r} (1 - \frac{1}{r^l}) \approx 0.6181$. When $l = 8$, $\prod_{\text{prime } r} (1 - \frac{1}{r^l}) \approx 0.9959$.

Corollary. For a random matrix with integral entries independently and uniformly chosen in a large interval $[1, N]$, the probability that its two entries in each row are coprime is about $0.6181^2 \approx 0.3821$.