Cryptanalysis of a Knapsack Public Key Cryptosystem

Baocang Wang*†, Hui Liu†* and Yupu Hu†
*Computer Science Department of Zhoukou Normal University, Zhoukou 466001, China
†Key Laboratory of Computer Networks and Information Security,
Ministry of Education, Xidian University, Xi'an, 710071, China
Email: bcwang79@yahoo.com.cn, liuhui@zknu.edu.cn yphu@mail.xidian.edu.cn

***Abstract*—Murakami and Nasako proposed a knapsack public key cryptosystem in 2008. They claimed that their proposal is secure against some known attacks. In this paper, we propose a cryptanalytic attack on the cryptosystem. We use a heuristic method to show that the secret key can be recovered with lattice reduction algorithms. Hence, their construction is insecure.**

***Keywords*-public key cryptography; cryptanalysis; knapsack problem; lattice reduction**

I. Introduction

Since the proposal of the concept of public key cryptography by Diffie and Hellman [1], many attempts have been made to find pratical public key cryptosystems (PKCs). However, the security of the PKC's proposed so far, in most cases, depends on the difficulty of discrete logarithm problem or factoring problem. Shor showed that these problems can be easily solved by using quantum computers [2]. Hence, public key cryptosystems secure in quantum computing environments are expected. Knapsack problem is one that cannot be easily solved even using quantum computers. Hence, some cryptographers made some efforts to design secure knapsack public key cryptosystems [3].

In 2008, Murakami and Nasako constructed a new knapsack PKC [3]. The public key of the cryptosystem consists of two knapsack sequences that can be regarded as mutually independent knapsack sequences. They showed that their construction can be secure against the low-density attacks and key-recovery attacks.

In this paper, we propose a heuristic algorithm to cryptanalyze the cryptosystem. We show that the secret key can be recovered in two steps. In the first step, we solve two simultaneous Diophantine approximation problems by using lattice reduction algorithms to obtain eight parameters, which will be used to construct four linear equations with respect to the secret keys. In the second step, we construct a 4-dimensional lattice, and use lattice reduction algorithms to obtain a relatively short vector in the lattice. From the resultant short vector, we can recover the secret keys.

The paper is organized as follows. In section 2, we presents the lattice theory in a conceptual level, and review the Murakami-Nasako cryptosystem. The proposed cryptanalysis is detailed in Section 3. Section 4 concludes the work.

II. Preliminaries

A. Lattice

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

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

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

There are two important algorithmic problems in lattice theory: the shortest vector problem (SVP) and the smallest basis problem (SBP). The SVP asks for the shortest nonzero vector in a given lattice L. The SBP aims at finding a lattice basis minimizing the maximum of the lengths of its elements. The problems are of special significance in complexity theory and cryptology. The SVP can be approximated by solving SBP. No polynomial-time algorithm is known for the problems. The best polynomial time algorithms for solving SVP achieve only slightly sub-exponential factors, and are based on the LLL algorithm [4].

The SVP is widely believed as difficult problems. However, interestingly, experimental results showed that lattice reduction algorithms behave much more nicely especially in the low-dimensional (< 300) lattices than it was expected from the worst-case proved bounds. When the dimension of a lattice is low, the lattice reduction algorithms can serve as an SVP lattice oracle to output a relatively short vector in a lattice. The readers can find some current records for lattice computations at the end of [5].

B. Murakami and Nasako's Construction

In their cryptosystem, three parameters are needed, U, J, and $N = U + J$, where U is the binary length of plaintexts, and J is the binary size of random bits. To generate secret/public key pairs, the user firstly generates a super-increasing sequence $s_1, \cdots, s_U$ with $s_i > \sum_{j=1}^{i-1} s_j$ for $i = 2, \cdots, U$. Secondly, he randomly generates a positive integer $Z > \sum_{j=1}^{U} s_j$, and sets $s_i = Z$ for $i = U + 1, \cdots, N$. Thirdly, the user splits the sequence $s = (s_1, \cdots, s_N)$ into two parts $\alpha = (\alpha_1, \cdots, \alpha_N)$ and $\beta = (\beta_1, \cdots, \beta_N)$ such that $s_i = \alpha_i + \beta_i$ for

978-0-7695-3744-3/09 $25.00 © 2009 IEEE
DOI 10.1109/IAS.2009.299

$i=1,\ \cdots,\ N$. Generate four integers p, q, v, and w such that p and v, q and w are relatively prime respectively, i.e., $\gcd(v,\ p)=\gcd(w,\ q)=1$. We also require that

$$p > \sum_{\alpha_i>0}\alpha_i - \sum_{\alpha_i<0}\alpha_i, \qquad q > \sum_{\beta_i>0}\beta_i - \sum_{\beta_i<0}\beta_i. \quad (1)$$

The public key consists of $a=(a_1,\ \cdots,\ a_N)$, $b=(b_1,\ \cdots,\ b_N)$, U, J, and N with $a_i = v\alpha_i (\bmod\ p)$ and $b_i = w\beta_i (\bmod\ q)$ for $i=1,\ \cdots,\ N$. The secret key is (s, α, β, v, w, p, q).

To encrypt a U-bit-long plaintext $m_1,\ \cdots,\ m_U$, the sender randomly generates J bits $m_{U+1},\ \cdots,\ m_N$. The ciphertext $(C_p,\ C_q)$ is computed by $C_p=\sum_{i=1}^N a_i m_i$ and $C_q=\sum_{i=1}^N b_i m_i$.

Given a ciphertext $(C_p,\ C_q)$, we can recover the plaintext as follows. Firstly, compute $M_p = v^{-1}C_p (\bmod\ p) = \sum_{i=1}^N \alpha_i m_i$ and $M_q = w^{-1}C_q (\bmod\ q) = \sum_{i=1}^N \beta_i m_i$. Secondly, compute $M = M_p + M_q (\bmod\ Z) = \sum_{i=1}^U s_i m_i$. Note that $(s_1,\ \cdots,\ s_U)$ is a super-increasing sequence. Hence, it is easy to recover $m_1,\ \cdots,\ m_U$ from $M=\sum_{i=1}^U s_i m_i$.

III. The Proposed Cryptanalysis

In this section, we derive a heuristic key-recovery attack on the cryptosystem. We are only given the public sequences a and b, and want to retrieve the whole secret key (s, α, β, v, w, p, q).

Note that $v^{-1}a_i = \alpha_i (\bmod\ p)$ and $w^{-1}b_i = \beta_i (\bmod\ q)$ for $i=1,\ \cdots,\ N$. Hence, there must exist integers k_i and l_i such that $v^{-1}a_i - \alpha_i = k_i p$ and $w^{-1}b_i - \beta_i = l_i q$. If p, q, v^{-1}, and w^{-1} are recovered, the whole secret key is reconstructed in that $\alpha_i = v^{-1}a_i (\bmod\ p)$, $\beta_i = w^{-1}b_i (\bmod\ q)$ and $s_i = \alpha_i + \beta_i$. We bear in mind that α_i and β_i are much smaller than p and q respectively, so when we compute $\alpha_i = v^{-1}a_i (\bmod\ p)$, $\beta_i = w^{-1}b_i (\bmod\ q)$, we take α_i and β_i as the absolute least residue modulo p and q respectively. Hence, the secret key (s, α, β, v, w, p, q) is obtained. The proposed cryptanalytic algorithm consists of two phases. In the first phase, we determine the k_i and l_i such that $v^{-1}a_i - \alpha_i = k_i p$ and $w^{-1}b_i - \beta_i = l_i q$. In fact, four k_i's and l_i's are sufficient to derive the secret key in the second phase.

A. The first phase

At first, we choose four integer pairs $(a_i,\ b_i)$'s such that a_{i1}, a_{i2}, a_{i3}, a_{i4} are distinct, and b_{i1}, b_{i2}, b_{i3}, b_{i4} are also different. Without loss of generality, we assume that the four integer pairs are $(a_1,\ b_1)$, $(a_2,\ b_2)$, $(a_3,\ b_3)$, and $(a_4,\ b_4)$. Hence, we have eight integers $k_1,\ \cdots,\ k_4$ and $l_1,\ \cdots,\ l_4$ such that $v^{-1}a_i - \alpha_i = k_i p$ and $w^{-1}b_i - \beta_i = l_i q$, $i=1,\ \cdots,\ 4$. From $v^{-1}a_i - \alpha_i = k_i p$ we obtain $v^{-1}/p - k_i/a_i = \alpha_i/(pa_i)$. The equation when $i=2$ minus the one when $i=1$ results in $k_1/a_1 - k_2/a_2 = (a_1\alpha_2 - a_2\alpha_1)/(pa_1a_2)$. Hence, we have

$$\left|\frac{a_2}{a_1} - \frac{k_2}{k_1}\right| = \frac{|a_1\alpha_2 - a_2\alpha_1|}{pk_1a_1} < \frac{|\alpha_1|+|\alpha_2|}{k_1a_1}. \quad (2)$$

We note that $p > \sum_{\alpha_i>0}\alpha_i - \sum_{\alpha_i<0}\alpha_i$ is much larger α_i, and for randomly-chosen v, $a_i = v\alpha_i (\bmod\ p)$ always has about the same size with p. Therefore, the right side of (2) is very small. From $v^{-1}a_i - \alpha_i = k_i p$ and $p > a_i$, we can conclude that $k_i < a_i$. Hence, k_2/k_1 is a fraction with a smaller denominator $k_1 < a_1$ approximating the fraction a_2/a_1. Similarly, we argue that k_3/k_1, k_4/k_1 also approximate a_3/a_1 and a_4/a_1, respectively.

From the aforementioned analysis, we know that to determine $k_1,\ \cdots,\ k_4$, it suffices to search for a set of fractions $\{k_2/k_1,\ k_3/k_1,\ k_4/k_1\}$ sharing a common and relatively small denominator $k_1 < a_1$ approximating the set of publicly computable fractions $\{a_2/a_1,\ a_3/a_1,\ a_4/a_1\}$. This problem is called simultaneous Diophantine approximation problem, a basic problem in Diophantine approximation theory, which has found uses in both cryptanalysis [6] and cryptography [7].

To solve the simultaneous Diophantine approximation problem, we construct a lattice $L(P)$ generated by the row vectors of the following matrix

$$P = \begin{pmatrix} a_1 & 0 & 0 & 0 \\ 0 & a_1 & 0 & 0 \\ 0 & 0 & a_1 & 0 \\ -a_2 & -a_3 & -a_4 & \varepsilon \end{pmatrix} = \begin{pmatrix} p_2 \\ p_3 \\ p_4 \\ p_1 \end{pmatrix}, \quad (3)$$

where $\varepsilon > 0$ can be set arbitrarily small, for examle, $\varepsilon = 1/2^N$. Lattice basis reduction algorithms can be applied to the lattice $L(P)$ to output a relatively short vector v_P, which can be used to approximate the simultaneous Diophantine approximation problem. Since $v_P \in L(P)$, there exist integers k_2, k_3, k_4, k_1 such that $v_P = k_2p_2 + k_3p_3 + k_4p_4 + k_1p_1 = (k_2a_1 - k_1a_2,\ k_3a_1 - k_1a_3,\ k_4a_1 - k_1a_4,\ \varepsilon k_1)$. Since v_P is short, $|k_2a_1 - k_1a_2|$, $|k_3a_1 - k_1a_3|$, $|k_4a_1 - k_1a_4|$ are small. This is equivalent to saying that $|a_2/a_1 - k_2/k_1|$, $|a_3/a_1 - k_3/k_1|$, and $|a_4/a_1 - k_4/k_1|$ are also small. So $\{k_2/k_1,\ k_3/k_1,\ k_4/k_1\}$ is a set of fractions, with a common and small denominator k_1, approximating $\{a_2/a_1,\ a_3/a_1,\ a_4/a_1\}$. At this phase, the $k_1,\ k_2,\ k_3,\ k_4$ are recovered. Similarly, we can obtain $l_1,\ l_2,\ l_3,\ l_4$. To recover the secret key, in the second phase we will illustrate how to solve the equations $v^{-1}a_i - \alpha_i = k_i p$ and $w^{-1}b_i - \beta_i = l_i q$, $i=1,\ \cdots,\ 4$, for $p,\ q,\ v^{-1},\ w^{-1}$.

B. The second phase

From $v^{-1}a_i - \alpha_i = k_i p$ and $w^{-1}b_i - \beta_i = l_i q$ we get $s_i = \alpha_i + \beta_i = v^{-1}a_i - k_i p + w^{-1}b_i - l_i q$. We note that if s_i is randomly split into two parts α_i and β_i, the expected value of α_i and β_i should be $s_i/2$. Therefore, for $i = 1, \ \cdots, \ 4$,

$$p > \sum_{\alpha_i > 0} \alpha_i - \sum_{\alpha_i < 0} \alpha_i \approx \sum_{i=1}^{N} \frac{s_i}{2} \gg s_i.$$

Note that a_i and k_i have a size comparable to that of p. Hence, a_i and k_i are much larger than s_i. Similarly, we know b_i and l_i are much larger than s_i. Therefore, for $i = 1, \ \cdots, \ 4$,

$$\begin{aligned} \sqrt{\textstyle\sum_{i=1}^{4} a_i^2} &\gg \sqrt{\textstyle\sum_{i=1}^{4} s_i^2}, \\ \sqrt{\textstyle\sum_{i=1}^{4} k_i^2} &\gg \sqrt{\textstyle\sum_{i=1}^{4} s_i^2}, \\ \sqrt{\textstyle\sum_{i=1}^{4} b_i^2} &\gg \sqrt{\textstyle\sum_{i=1}^{4} s_i^2}, \\ \sqrt{\textstyle\sum_{i=1}^{4} l_i^2} &\gg \sqrt{\textstyle\sum_{i=1}^{4} s_i^2}. \end{aligned} \tag{4}$$

In other words, if we look $A = (a_1, \ \cdots, \ a_4)$, $K = (k_1, \ \cdots, \ k_4)$, $B = (b_1, \ \cdots, \ b_4)$, $L = (l_1, \ \cdots, \ l_4)$, and $S = (s_1, \ \cdots, \ s_4)$ as 4-dimensional vectors, S is much shorter than $A, \ K, \ B, \ L$ under the Euclidean norm. Hence, the basic idea of the second phase is to construct a lattice such that $(s_1, \ s_2, \ s_3, \ s_4)$ is a short vector in the lattice.

Now we construct the following matrix

$$Q = \begin{pmatrix} a_1 & a_2 & a_3 & a_4 \\ k_1 & k_2 & k_3 & k_4 \\ b_1 & b_2 & b_3 & b_4 \\ l_1 & l_2 & l_3 & l_4 \end{pmatrix} = \begin{pmatrix} A \\ K \\ B \\ L \end{pmatrix}, \tag{5}$$

and denote the lattice generated by the basis $A, \ K, \ B, \ L$ as $L(Q)$. We observe that $S = v^{-1}A + (-p)K + w^{-1}B + (-q)B \in L(Q)$. From (4), we also know that S is a very short vector in $L(Q)$. Hence, S is expected to be the shortest vector in $L(Q)$, and may be found with lattice reduction algorithms. If we obtain S, we can determine the integral combinatorial coefficients v^{-1}, $-p$, w^{-1}, and $-q$ by solving a system of equations. At this phase, we obtain the secret keys v^{-1}, $-p$, w^{-1}, and $-q$. Using the extended Euclidean algorithm, we can easily determine v and w. According to $\alpha_i = v^{-1}a_i (\text{mod } p)$ and $\beta_i = w^{-1}b_i (\text{mod } q)$, we can easily recovered α and β. We should note that α_i and β_i are taken as the absolute least residue modulo p and q respectively. From $s_i = \alpha_i + \beta_i$ we can determine s. Hence, the whole secret key $(s, \ \alpha, \ \beta, \ v, \ w, \ p, \ q)$ is recovered.

C. Summary

The proposed cryptanalysis is summarized as follows.

S1: From the public sequences a and b choose four integer pairs $(a_i, \ b_i)$, say, $(a_i, \ b_i)$ with $i = 1, \ \cdots, \ 4$, such that the four a_i's and b_i's are distinct.

S2: Construct a matrix P (3), and apply lattice reduction algorithms, for example LLL algorithm [4], to the lattice $L(P)$ generated by the row vectors of P to obtain the shortest vector v_P.

S3: Represent v_P as an integral combination of the row vectors of P by solving a system of equations, and obtain four positive integers $k_2, \ k_3, \ k_4, \ k_1$.

S4: From the values of $b_1, \cdots, b_4$, and S2, S3, obtain four positive integers $l_2, \ l_3, \ l_4, \ l_1$.

S5: From the known values, construct a matrix Q (5), and apply lattice reduction algorithms to output a short vector S in $L(Q)$.

S6: Represent S as an integral combination of the row vectors of Q, and obtain the combinatorial coefficients v^{-1}, $-p$, w^{-1}, and $-q$.

S7: Compute the inverses v of v^{-1} modulo p, and w of w^{-1}, modulo q.

S8: Compute α_i as the absolute least residue of $v^{-1}a_i$ modulo p, and β_i as the absolute least residue of $w^{-1}b_i$ modulo q, and set $s_i = \alpha_i + \beta_i$.

S9: Output the secret key $(s, \ \alpha, \ \beta, \ v, \ w, \ p, \ q)$.

D. Experimental results

We use a toy example presented in [3] to illustrate our cryptanalysis. We are given two knapsack sequences $a = (a_1, \cdots, a_{16}) = (3064, 2720, 2827, 616, 2121, 2994, 257, 1954, 2368, 1722, 387, 2191, 1602, 2600, 1048, 1959)$, and $b = (b_1, \cdots, b_{16}) = (1151, 1810, 1287, 1042, 275, 610, 2284, 2541, 2172, 2701, 583, 2290, 1033, 1045, 2036, 3417)$. Now we show how to use the proposed cryptanalysis to recover the secret key.

In the first phase, we choose 4 pairs of (a_i, b_i)'s $(a_1, b_1) = (3064, 1151)$, $(a_2, b_2) = (2720, 1810)$, $(a_3, b_3) = (2827, 1287)$, and $(a_8, b_8) = (1954, 2541)$, and construct the following matrices

$$P_a = \begin{pmatrix} 3064 & 0 & 0 & 0 \\ 0 & 3064 & 0 & 0 \\ 0 & 0 & 3064 & 0 \\ -2720 & -2827 & -1954 & 1/2^{16} \end{pmatrix},$$

$$P_b = \begin{pmatrix} 1151 & 0 & 0 & 0 \\ 0 & 1151 & 0 & 0 \\ 0 & 0 & 1151 & 0 \\ -1810 & -1287 & -2541 & 1/2^{16} \end{pmatrix}.$$

We use lattice reduction algorithms to the lattices $L(P_a)$ and $L(P_b)$ generated by the integral linear combinations of the row vectors of P_a and P_b respectively, and obtain the corresponding short vectors $v_{P_a} = (-400, -44, -400, 1228/2^{16}) \in L(P_a)$ and $v_{P_b} = (161, 21, 130, 770/2^{16}) \in L(P_b)$. By representing v_{P_a} and v_{P_b} as linear combinations of the row vectors of P_a and P_b respectively, we obtain the integral combinatorial coefficients $(k_2, k_3, k_8, k_1) = (1090, 1133, 783, 1228)$ and $(l_2, l_3, l_8, l_1) = (1211, 861, 1700, 770)$.

In the second phase, we construct a matrix (5),

$$Q = \begin{pmatrix} 3064 & 2720 & 2827 & 1954 \\ 1228 & 1090 & 1133 & 783 \\ 1151 & 1810 & 1287 & 2541 \\ 770 & 1211 & 861 & 1700 \end{pmatrix},$$

and use lattice reduction algorithm on the lattice $L(Q)$ generated by the row vectors of Q and obtain a short vector $S = (1, 3, 7, 216)$. By representing v_{P_a} and v_{P_b} as linear combinations of the row vectors of Q, we obtain the integral combinatorial coefficients $(v^{-1}, -p, w^{-1}, -q) = (1250, -3119, 2373, -3547)$. Hence, $p = 3119$, $q = 3547$, $v^{-1} = 1250$, and $w^{-1} = 2373$ are recovered. By using the extended Euclidean algorithm, we obtain $v = 1300$ and $w = 142$. From these values and the public knapsack sequences, we can compute $\alpha_i = v^{-1}a_i (\text{mod}\, p) = (-132, 290, -77, -393, 100, -300, -7, 323, 69, 390, 305, 268, 102, 2, 20, 335)$, and $\beta_i = w^{-1}b_i (\text{mod}\, q) = (133, -287, 84, 407, -73, 354, 116, -107, 365, 44, 129, 166, 332, 432, 414, 99)$. Finally, we just need to compute $\alpha + \beta$ to recover $s = (1, 3, 7, 14, 27, 54, 109, 216, 434, \cdots, 434)$. Now, we conclude that the whole secret key (s, α, β, v, w, p, q) is recovered.

IV. Conclusions

The knapsack cryptosystem proposed by Murakami and Nasako in 2008 is cryptanalized. We present a heuristic method to recover the secret key by using lattice reduction algorithms. Analysis shows that the secret key of the cryptosystem can be recovered. We think that their cryptosystem is insecure mainly because the construction of the public key is linear, that is, $s_i = \alpha_i + \beta_i = v^{-1}a_i - k_i p + w^{-1}b_i - l_i q$. Maybe some modifications can be obtained to make the cryptosystem resist our attack.

V. Acknowledgment

This work was supported by the National Natural Science Foundation of China (No. 60803149), the National Grand Fundamental Research 973 Program of China (No. 2007CB311201), and the 111 Project (No. B08038).

References

[1] W. Diffie and M.E. Hellman. New directions in cryptography, *IEEE Transaction on Information Theory*, 22(6): 644-654, 1976.

[2] P.W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, *SIAM Journal of Computing*, 26: 1484-1509, 1997.

[3] Y. Murakami and T. Nasako. A new trapdoor in knaspsack public-key cryptosystem with two sequences as the public key. Third 2008 International Conference on Convergence and Hybrid Information Technology, 2008, pp. 357-362.

[4] A.K. Lenstra, H. W. Lenstra Jr., and L. Lovász. Factoring polynomials with rational coefficients. *Mathematische Annualen*, 261: 513-534, 1982.

[5] P. Nguyen and J. Stern, Adapting density attacks to low-weight knapsacks, Asiacrypt 2005, LNCS 3788, Chennai, India: Springer-Verlag, 2005: 41-58.

[6] B. Wang, Y. Hu, Diophantine approximation attack on a fast public key cryptosystem. ISPEC 2006, LNCS 3903, Berlin: Springer-Verlag, 2006: 25-32.

[7] W. Baocang and H. Yupu, Public key cryptosystem based on two cryptographic assumptions, *IEE Proceedings of Communications*, 152(6): 861-865, 2005.