Cryptanalysis of a Knapsack Based Two-Lock Cryptosystem*

Bin Zhang[1,2], Hongjun Wu[1], Dengguo Feng[2], and Feng Bao[1]

[1] Institute for Infocomm Research, Singapore 119613
[2] State Key Laboratory of Information Security,
Graduate School of the Chinese Academy of Sciences,
Beijing, 100039, PRC
{stuzb,hongjun}@i2r.a-star.edu.sg

Abstract. In this paper we break a knapsack based two-lock cryptosystem proposed at ICICS'03 [7]. The two-lock cryptosystem is a commutative encryption algorithm that is very useful for the construction of the general t-out-of-n oblivious transfers and millionaire protocol. However, our analysis shows that the proposed knapsack based two-lock cryptosystem is extremely insecure. The serious flaw is that the sender in the two-lock cryptosystem can retrieve the secret key of the receiver fairly easily. We have implemented the attack on a Pentium 4 2.5 GHz processor. For the parameters given in [7], it takes only several minutes to break that knapsack based two-lock cryptosystem.

Keywords. Cryptanalysis, Two-lock cryptosystem, Knapsack problem.

1 Introduction

Cryptography plays an important role in today's digital world. Many cryptographic techniques have been developed to meet the various requirements arising from applications. Among them oblivious transfer is a very useful cryptographic primitive. The concept of oblivious transfer (OT) was first proposed by Rabin in [6]. In that paper, the sender has one bit secret message and would like the receiver to get it with probability, but the receiver does not want the sender to know whether the secret message being received or not. The 1-out-of-2 OT means that the sender has two secrets and would like the receiver to get one of them at the receiver's choice, meanwhile the receiver does not want the sender to know which secret bit being chosen. The concept of t-out-of-n OT is the generalization of that of 1-out-of-2 OT. The sender can not determine which t messages the receiver obtained, and the receiver can not learn the other $(n-t)$ messages. A millionaire protocol is used to solve the following problem. Two parties, each has a secret integer. Without revealing those two secret integers,

* Supported by National Natural Science Foundation of China (Grant No. 60273027), National Key Foundation Research 973 project (Grant No. G1999035802) and National Science Fund for Distinguished Young Scholars (Grant No. 60025205)

M. Jakobsson, M. Yung, J. Zhou (Eds.): ACNS 2004, LNCS 3089, pp. 303–309, 2004.

they could compare those two integers. In both the t-out-of-n oblivious transfers and millionaire protocol cases, the basic security requirement is that those two parties should not know each other's secret information.

The two-lock cryptosystems proposed in [7] can be used to efficiently construct the t-out-of-n oblivious transfer and/or millionaire protocol. The two-lock cryptosystem consists of two commutative encryption algorithms A and B. Let A and B denote the encryption algorithms belong to Alice and Bob, respectively. A and B satisfy $B_s(A_k(m)) = A_k(B_s(m))$ for any randomly chosen secret keys k and s. This two-lock cryptosystem operates as follows. If the sender Alice wants to send a secret message m to Bob, they communicate with each other as follows:

1. Alice sends to Bob: $Y = A_k(m)$.
2. Bob sends to Alice: $Z = B_s(Y)$.
3. Alice sends to Bob: $C = A_k^{-1}(Z)$.
4. Bob decrypts: $m = B_s^{-1}(C)$.

where $A_k^{-1}(\cdot)$ and $B_s^{-1}(\cdot)$ denote the decryption of $A_k(\cdot)$ and $B_s(\cdot)$ respectively. It is easy to see that at the end Bob can obtain the message m. A two-lock cryptosystem should meet the following security requirements: it should be computationally infeasible for an adversary to recover the keys k or s such that $C = A_k^{-1}(Z)$ or $Z = B_s(Y)$. And it should be computationally impossible for the two parties to recover each other's secret key. The very simple and efficient discrete logarithm based two-lock cryptosystem has been proposed in [1]. In [7], a new knapsack based two-lock cryptosystem was proposed.

In this paper, we show that the knapsack based two-lock cryptosystem proposed in [7] is extremely insecure. The sender in the two-lock cryptosystem can recover the receiver's secret key fairly easily.

This paper is organized as follows. In Section 2, we introduce the proposed knapsack based two-lock cryptosystem with some informal analysis. Our attack against this cryptosystem is given in Section 3. In Section 4, detailed experiment results of our attack are listed with some remarks. Section 5 concludes the paper.

2 The Knapsack Based Two-Lock Cryptosystem

We first recall the definition of knapsack problem. Let $a_1, \cdots, a_l, S, l$ be some integers. The knapsack or subset-sum problem is to determine, given positive integers $a_1, \cdots, a_l, S$, whether there is a subset of the $\{a_j\}$ that sums to S. This is equivalent to determine whether there are variables $x_1, \cdots, x_l \in \{0, 1\}$ satisfying $x_1a_1 + \cdots + x_la_l = S$. The density $d(a)$ of the knapsack vector $(a_1, \cdots, a_l)$ is defined as $d(a) = l/\log_2 \max\{a_1, \cdots, a_l\}$. The general knapsack problem is known to be NP-complete [4]. The first knapsack-based cryptosystem was proposed by Merkle and Hellman in 1978 [5], followed by a number of variants. Unfortunately, most of them were broken. The main reason is that although the general knapsack problem is hard, the knapsack algorithm being used in those cryptosystems may not be hard, and the cryptanalyst can deduce the original

solvable knapsack from the seemingly random knapsack. A good overview of these systems and their cryptanalysis can be found in [2,3].

The following describes the knapsack based two-lock cryptosystem proposed in [7]. Let t, k, n, l be secure parameters. Alice wish to send Bob a positive integer sequence $m = (m_1, \cdots, m_l) = (u_{1,1}, \cdots, u_{l,1}) + (v_{1,1}, \cdots, v_{l,1})$ where the binary length of m_i is n and $m_i \neq m_j (i \neq j)$. They begin their confidential communication as follows.

1. *Alice:* For $h = 1$ to t, randomly select positive integers e_h, M_h, f_h, N_h such that $M_h > k\max\{u_{1,h}, \cdots, u_{l,h}\}$, $N_h > k\max\{v_{1,h}, \cdots, v_{l,h}\}$, $(e_h, M_h) = 1$, $(f_h, N_h) = 1$ and $(M_h, N_h) = 1$. Compute $u_{j,h+1} = e_h u_{j,h} \bmod M_h$, $v_{j,h+1} = f_h v_{j,h} \bmod N_h$, for $j = 1$ to l. By Chinese remainder theorem, compute $(y_1, \cdots, y_l)$ such that $u_{j,t+1} = y_j \bmod M_t$ and $v_{j,t+1} = y_j \bmod N_t$, for $j = 1, \cdots, l$, i.e. $y_j = u_{j,t+1} N_t^{\phi(M_t)} + v_{j,t+1} M_t^{\phi(N_t)} \bmod M_t N_t$, where $\phi(\cdot)$ is the Euler function. Then select a random integer α and send $Y = (Y_1, \cdots, Y_l) = (y_1 - \alpha, \cdots, y_l - \alpha)$ to Bob.
2. *Bob:* Select a random nonsingular matrix $B = (b_{i,j})_{l \times l}$, where $b_{i,j} \in \{0, 1\}$ and the hamming weight of each column is k. Send $Z = (z_1, \cdots, z_l) = YB$ to Alice.
3. *Alice:* for $h = t$ to 1, compute $d_h = e_h^{-1} \bmod M_h$, $g_h = f_h^{-1} \bmod N_h$. Let $U_{i,t} = d_t(z_i + k\alpha) \bmod M_t$, $V_{i,t} = g_t(z_i + k\alpha) \bmod N_t$ for $i = 1, \cdots, l$. For $h = t-1, \cdots, 1$, calculate $U_{j,h} = d_h U_{j,h+1} \bmod M_h$, $V_{j,h} = g_h V_{j,h+1} \bmod N_h$, for $j = 1, \cdots, l$. Finally, send Bob $C = (c_1, \cdots, c_l) = (U_{1,1} + V_{1,1}, \cdots, U_{l,1} + V_{l,1})$.
4. *Bob:* Compute $(m_1, \cdots, m_l) = (c_1, \cdots, c_l) B^{-1}$.

In [7], the authors argue that if the adversaries intend to find a nonsingular matrix $(b'_{i,j})_{l \times l}$ form $Z = (z_1, \cdots, z_l)$ such that $z_j = b'_{1,j} y_1 + \cdots + b'_{l,j} y_l$, then they will be confronted with a random knapsack problem with density about $l/\log_2(M_t N_t)$. Let $l \geq 1000$, $t \geq 50$, $k = 128$, $n = 100$ and $M_t N_t \leq 2^{900}$, then the density $d(a) > 1$. However, as we will show below, it is the proposed sparse structure of the column vector of $B = (b_{i,j})_{l \times l}$ that leads to the failure of the knapsack based two-lock system.

3 Cryptanalysis of the Knapsack Based Two-Lock Cryptosystem

Our main idea is that if a dishonest Alice deceives Bob with a random-looking vector $(Y_1, \cdots, Y_l)$ of integers, then she can recover the matrix $B = (b_{i,j})_{l \times l}$ by solving easy knapsack problems due to the fact that Alice can choose all the information vectors sent to Bob at her choice. Thus if Alice chooses an easy knapsack and disguise it as a random-looking knapsack, then she can recover the original easy vector she sent to Bob from the vector encrypted by Bob using the knapsack-like encryption scheme. Since the basic security requirement of t-out-of-n oblivious transfers and millionaire protocol is that the two communication parties should not know the counterpart's secret key, the attack above indicates

that the proposed knapsack based two-lock cryptosystem is insecure for t-out-of-n oblivious transfers and millionaire protocol applications.

Now we are ready to give the description of our attack in detail. The problem we face is to restore the matrix $B = (b_{i,j})_{l\times l}$ from $Z = (z_1, \cdots, z_l)$ and $Y = (Y_1, \cdots, Y_l)$, where $Z = YB$ and $Y = (Y_1, \cdots, Y_l)$ is chosen at Alice's choice. We wish to recover each column vector $(b_{1,j}, \cdots, b_{l,j})^T$ of hamming weight 128 such that $z_j = b_{1,j}Y_1 + b_{2,j}Y_2 + \cdots + b_{l,j}Y_l$, for $j = 1, 2, \cdots, l$. It is obvious that the better the vector chosen by Alice, the easier it is to recover the matrix B. Since $M_tN_t \leq 2^{900}$, without loss of generality we take the binary representations of the integers Y_i to have 900 bit length. Our attack consists of three stages. At the first stage, Alice chooses a special integer vector $Y' = (Y_1', \cdots, Y_l')$. At the second stage, Alice disguises that special integer vector into a random-looking vector $Y = (Y_1, \cdots, Y_l)$. Finally, Alice recovers the matrix $B = (b_{i,j})_{l\times l}$ from $Z = (z_1, \cdots, z_l)$ and $Y = (Y_1, \cdots, Y_l)$, where Y is encrypted by Bob as $Z = YB$.

3.1 Our Attack

We take the scheme with parameters $l = 2000$, $k = 128$ and $2^{899} < M_tN_t < 2^{900}$ to demonstrate our algorithm. As stated above, the attack consists of three stages, i.e. choosing stage, disguising stage and recovering stage.

Stage 1. Choosing Special Integer Vector Y'. Choose integers $Y_1', \cdots, Y_{2000}'$ such that their binary representations being the row vectors of the following binary matrix (the rightmost bit is the least significant bit):

$$\begin{pmatrix} 0_{40\times 18} & 0_{40\times 18} & \cdots & 0_{40\times 18} & A^1_{40\times 18} \\ 0_{40\times 18} & 0_{40\times 18} & \cdots & A^2_{40\times 18} & 0_{40\times 18} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ A^{50}_{40\times 18} & 0_{40\times 18} & \cdots & 0_{40\times 18} & 0_{40\times 18} \end{pmatrix}_{2000\times 900} \tag{1}$$

Note that $0_{40\times 18}$ and $A^i_{40\times 18}$, $i = 1, \cdots, 50$, are sub-matrices of specified size, i.e. $0_{40\times 18}$ denotes zero matrix and $A^i_{40\times 18}$, $i = 1, \cdots, 49$, denote 40×18 matrices such that their row vectors are randomly chosen from the linear vector space $GF(2)^{18}$. $A^{50}_{40\times 18}$ denotes the matrix such that its leftmost 3 column vectors are the zero vectors and other elements are randomly chose from $GF(2)$ in such a way that the row vectors are all the non-zero vectors. The reason for such a choice of $A^{50}_{40\times 18}$ is stated in the following stage 2.

Stage 2. Disguising the Special Integer Vector Y'. We use the standard transformation to disguise an easy knapsack into a seemingly more complicated one, i.e. we first select a large integer W such that $(W, M_tN_t) = 1$ and let $Y_i = WY_i' \bmod M_tN_t$. Alice then sends the resultant integer vector to Bob. After receiving the encrypted vector from Bob, Alice reverses the procedure mentioned above using $Y_i' = Y_iW^{-1} \bmod M_tN_t$. Then Alice gets the easy knapsack and

recovers the secret key of Bob as stage 3 states. To guarantee recovering the original vector from the encrypted vector successfully, the following condition should be satisfied:

$$\sum_{i=1}^{l} b_{i,j} Y_i^{'} < M_t N_t \tag{2}$$

which results from

$$z_j = \sum_{i=1}^{l} b_{i,j} Y_i = \sum_{i=1}^{l} b_{i,j} (Y_i^{'} W \bmod M_t N_t) \tag{3}$$

Since there are on average at most 2 carriers from $A^i_{40\times 18}$ to $A^{i+1}_{40\times 18}$ when the row vectors are summed together, we put the leftmost 3 columns of $A^{50}_{40\times 18}$ to be zero vectors taking into account that $2^{899} < M_t N_t < 2^{900}$.

Stage 3. Recovering the Matrix B. The problem we now face is, given $z_j^{'} = z_j W^{-1} \mod M_t N_t$ and $Y_1^{'}, \cdots, Y_{2000}^{'}$, to find the column vector $(b_{1,j}, \cdots, b_{2000,j})^T$ of hamming weight 128 such that $z_j^{'} = b_{1,j} Y_1^{'} + b_{2,j} Y_2^{'} + \cdots + b_{2000,j} Y_{2000}^{'}$, for $j = 1, 2, \cdots, 2000$. First rewrite $z_j^{'}$ as binary representation

$$z_j^{'} = (z_{j,900}, \cdots, z_{j,773}, z_{j,772}, \cdots, z_{j,2}, z_{j,1})_2. \tag{4}$$

Then we get

$$\begin{aligned} z_j^{'} &= (z_{j,900}, \cdots, z_{j,19}, z_{j,18}, \cdots, z_{j,2}, z_{j,1})_2 \\ &= b_{1,j} Y_1^{'} + b_{2,j} Y_2^{'} + \cdots + b_{2000,j} Y_{2000}^{'} \\ &= b_{1,j} (0_1 \cdots 0_{18} 0_{19} \cdots 0_{864} 0_{865} \cdots 0_{882} *_{883} \cdots *_{900})_2 + \cdots + \\ &\quad b_{41,j} (0_1 \cdots 0_{18} 0_{19} \cdots 0_{864} *_{865} \cdots *_{882}\, 0_{883} \cdots 0_{900})_2 + \cdots + \\ &\quad b_{2000,j} (0_1 0_2 0_3 *_4 \cdots *_{18}\, 0_{19} \cdots 0_{864} 0_{865} \cdots 0_{882} 0_{883} \cdots 0_{900})_2, \end{aligned}$$

where the subscripts denote positions and the asterisks denote randomly chosen elements from $GF(2)$. We can see from above equation that the least significant bits $z_{j,18}, \cdots, z_{j,2}, z_{j,1}$ only depend on the sum of $b_{1,j} Y_1^{'} + \cdots + b_{40,j} Y_{40}^{'}$. The bits $z_{j,36}, \cdots, z_{j,20}, z_{j,19}$ depend on $b_{41,j} Y_{41}^{'} + \cdots + b_{80,j} Y_{80}^{'}$ and the carry from $b_{1,j} Y_1^{'} + \cdots + b_{40,j} Y_{40}^{'}$, ..., and so on. It is obvious from above observations that determining $(b_{1,j}, \cdots, b_{2000,j})^T$ is dependent on the determination of $(b_{1,j}, \cdots, b_{40,j})^T$, for given the knowledge of $(b_{1,j}, \cdots, b_{40,j})^T$, we can follow an iterative way to determine the remaining bits in $(b_{1,j}, \cdots, b_{2000,j})^T$. For the determination of $(b_{1,j}, \cdots b_{40,j})^T$, according to $128 \times 40/2000 = 2.56$, we use an exhaustive search through all the $2-3$ combinations of row vectors of $A^1_{40\times 18}$ to find out the '1' bits in $(b_{1,j}, \cdots, b_{40,j})^T$. The complexity of this procedure is about $\binom{40}{2} \approx 2^{9.60733}$ or $\binom{40}{3} \approx 2^{13.2703}$, which is absolutely negligible on an ordinary PC. We select the true combination of the row vectors, thus determine the '1' bits in $(b_{1,j}, \cdots, b_{40,j})^T$.

Now we give a full description of the attack.

1. *parameters:* $l = 2000$, $k = 128$, $2^{899} < M_t N_t < 2^{900}$, $(z_{900,j}, \cdots, z_{2,j}, z_{1,j}) = z'_j$, $(Y_1^{'}, \cdots, Y_{2000}^{'})$ as defined in the above matrix.
2. $(z_{18}^1, \cdots, z_1^1) = (z_{18,j}, \cdots, z_{1,j})$.
3. For $i = 1$ to 50, make an exhaustive search over the 40 row vectors of $A_{40\times 18}^i$ to find out about $2-3$ rows whose summation's least significant 18 bits are $(z_{18}^i, \cdots, z_1^i)$. Set the elements of $(b_{40(i-1)+1,j}, \cdots, b_{40i,j})$ corresponding to the selected rows as 1, others 0. If $i = 50$ stop else set $(z_{18}^{i+1}, \cdots, z_1^{i+1})$ be the bits in the position range $[18i + 1, 18(i + 1)]$ of $(z_{900,j}, \cdots, z_{1,j})_2 - \sum_{f=1}^{i}(b_{40f,j}(Y_{40f,j})_2 + \cdots + b_{40(f-1)+1,j}(Y_{40(f-1)+1,j})_2)$.

Complexity of the attack. From the above algorithm, we can recover the jth column vector $(b_{1,j}, \cdots, b_{2000,j})^T$ of B with $O(2^{20})$ operations on average, i.e. absolutely negligible amount of computations on an ordinary PC, and recover the matrix B with $O(2^{31})$ operations.

3.2 Some Remarks

In the algorithm above, we simply choose $2-3$ rows out of the 40 rows in order to clearly illustrate the main structure of our attack. In the experiments, we choose 5 rows out of the 40 rows in order to gain a high success probability. The complexity is also very small as shown in Section 4. In addition, we can also use the method above to attack the case that $l = 1000$, $k = 128$ and $2^{899} < M_t N_t < 2^{900}$, the complexity is $O(2^{27})$ if using partition 18×50 and 20×50. From the discussion above, we know that the insecurity of the proposed knapsack based two-lock cryptosystem is due to the sparseness of the columns of Bob's secret matrix B, which facilitates the attack with the growth of l if the partition of $Y_i^{'}$'s binary representations is properly chosen. Increasing the number of '1' bits in each column could enhance the resistance against our attack, but note that the size of the modulus constrains the number of '1' bits in each column. At Alice's side, modular arithmetic is carried out which means that the summation of $Y_i^{'}$s can not be larger than the modulus; otherwise Bob cannot decrypt correctly. So improving the knapsack based two-lock cryptosystem is nearly impossible.

4 Experiment Results

To check the actual performance as well as the correctness of our cracking algorithm, we have implemented our attack against the proposed knapsack based two-lock cryptosystem in C on the Pentium 4 2.5GHz processor. We use the stream cipher RC4 as the random noise source to supply the integers vector $(Y_1^{'}, Y_2^{'}, \cdots, Y_{2000}^{'})$ and the matrix $B = (b_{i,j})_{l\times l}$ where $l = 2000$. Then we simulate the process in the knapsack based two-lock cryptosystem to get the resulting vector $(z_1, z_2, \cdots, z_l)$. After obtaining $(z_1, z_2, \cdots, z_l)$, we apply our attack algorithm proposed in Subsection 3.1 to restore the matrix $B = (b_{i,j})_{l\times l}$ column-by-column. Instead of making an exhaustive search over the 3 out of 40 row vectors

of $A^i_{40\times 18}$to find out the correct rows, we made an exhaustive search over the 5 out of 40 row vectors to provide a higher success probability. The probability that there are at most five rows to be summed is

$$\frac{\sum_{i=0}^{5}\binom{40}{i}\binom{1960}{128-i}}{\binom{2000}{128}} \approx 0.961156, \tag{5}$$

where $i = 0$ corresponds to the case that $(b_{1,j}, \cdots, b_{40,j})^T$ happens to be a zero vector. In our experiments, there are on average

$$\binom{40}{5}/2^{18} \approx 2.5101, \tag{6}$$

wrong solutions corresponding to each value of $(z^i_{18}, \cdots, z^i_1)$. Actually, there are some columns where our algorithm only output one solution. We select the very solution with the minimum hamming weight. If some solutions have the same hamming weight, we check every possibility until we find the true key or an equivalent key. In our experiment, one 40-bit segment of one column of the secret key B can be recovered in about 3.6 milliseconds on average. We recovered the whole matrix key in about six minutes on the Pentium 4 2.5GHz processor PC. It is equivalent to about $2^{39.8}$ clock cycles. This experiment result is in expectation since the theoretical complexity given in Section 4 is $O(2^{31})$.

5 Conclusion

We have shown that the recently proposed knapsack based two-lock cryptosystem is insecure for oblivious transfers and millionaire protocol applications. It is an interesting problem to design new secure two-lock cryptosystems based on non-discrete logarithm problems.

References

1. F. Bao, R. Deng, P. Feng. An Efficient and Practical Scheme for Privacy Protection in E-commerce of Digital Goods. *2nd International Conference on Information and Communications Security-ICICS'00*, Springer-verlag, pp. 162-170, 2000.
2. E. F. Brickell, A.M.Odlyzko. Cryptanalysis: A Survey of Recent Results, *Proc. IEEE*, Vol.76, pp. 578-593, 1988.
3. W. Diffie. The First Ten Years of Public-Key Cryptography, *Proc. IEEE*, Vol.76, pp. 560-577, 1988.
4. M. R. Garey, D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, San Francisco, 1979.
5. R. C. Merkle, M. Hellman. Hiding Information and Signatures in Trapdoor Knapsack. *IEEE Transactions on Information theory*,Vol.24, No.5, pp. 525-530, 1978.
6. M. Rabin. How to Exchange Secrets by Oblivious Transfer, *Technical Report TR 81*, Aiken computation Laboratory, Harvard University, 1981.
7. Q. Wu, J. Zhang, and Y. Wang. Practical t-out-n Oblivious Transfer and Its Applications. *5th International Conference on Information and Communications Security-ICICS'03*, Springer-verlag, 2003, pp.226-237.