Cryptanalysis of a public key cryptosystem based on two cryptographic assumptions

A.M. Youssef

Abstract: Baocang and Yupu proposed a relatively fast public key cryptosystem. The authors claim that the security of their system is based on two number-theoretic hard problems, namely the simultaneous Diophantine approximation problem and the integer factorisation problem. In this article we present a polynomial time heuristic attack that enables us to recover the private key from the public key. In particular, we show that breaking the system can be reduced to finding a short vector in a lattice which can be achieved using the L^3-lattice reduction algorithm.

1 Introduction

Most public key schemes [1] are based on number-theoretic cryptographic assumptions such as the integer factorisation problem or discrete logarithm problem. One common feature among the majority of these systems is that they are constructed based only on one cryptographic assumption. Although these assumptions seem hard to break today, once an efficient algorithm for solving the underlying hard number-theoretic problems is developed, these systems based on the intractability assumptions could be easily broken.

To overcome this problem, Baocang and Yupu [2] proposed a new, and relatively fast, public key system based on multiple cryptographic assumptions. Baocang and Yupu argue that their proposed system is expected to be more secure because it seems unlikely that the multiple hard problems could be simultaneously solved in an efficient way.

In this paper we present a polynomial time heuristic attack that enables us to recover the private key from the public key. In particular, we show that breaking the system can be reduced to finding a short vector in a lattice which can be achieved using the L^3-lattice reduction algorithm [1,3].

The paper is organised as follows. In Section 2 we give a brief description for the system proposed in [2]. In Section 3, we describe our attack. Finally we give a numerical example using the same encryption-decryption example in [2].

2 Description of the Baocang–Yupu scheme

In this section we emphasise on the key generation step of the proposed public key scheme because of its relevance to our attack. Further details about the encryption and decryption operations and justification for the bounds on the parameters can be found in [2].

Let k be the security parameter and n the dimension of the matrices used during encryption and decryption.

The key generation algorithm runs as follows:

1. Randomly generate two strong primes p and q with p k-bits long and q 3k-bits long. Set $N = pq$.
2. m, e, b are random integers such that $m \approx q^{0.5}$, $e \approx q^{0.3}$, $b \approx q^{0.1}$, where m and e are upper bounds of plaintext blocks and error vectors, respectively.
3. Randomly choose a small n-dimensional mod q invertible matrix $\mathbf{B} = (b_{ij})$. The b_{ij}s satisfy the inequality $|b_{ij}| < b$.
4. Randomly choose n integers a_i, $i = 1, \ldots, n$, such that $q^{0.4} < a_i < q^{0.5}$ and calculate their inverse u_i mod q. Let $\mathbf{A} = diag(a_1, \ldots, a_n)$ and $\mathbf{U} = \mathbf{A}^{-1} = diag(u_1, \ldots, u_n)$. Compute $\mathbf{G} = \mathbf{UB} = \mathbf{A}^{-1}\,\mathbf{B}$ mod q.
Note that since $\mathbf{A}^{-1}$ is a diagonal matrix, then we have

$$b_{ij} = a_i g_{ij} \bmod q \tag{1}$$

5. Randomly choose an n-dimensional mod p invertible matrix $\mathbf{T} = (t_{ij})$. The matrix $\mathbf{H} = (h_{ij})$ is invertible mod N, where h_{ij} is constructed via Chinese remainder theorem [4], i.e.

$$\begin{aligned} h_{ij} &= g_{ij} \bmod q \\ h_{ij} &= t_{ij} \bmod p \end{aligned} \tag{2}$$

The 4-tuple ($\mathbf{H}$,N,m,e) forms the public key. The secret key consists of q and ($a_1, \ldots, a_n$).

The encryption operation is performed by first randomly choosing an n-dimensional vector $\mathbf{E} = (e_1, \ldots, e_n)$, with $e_i < e$, and splitting the plaintext $\mathbf{M}$ into n blocks $\mathbf{M} = (m_1, \ldots, m_n)$ with every block $m_i < m$. Then $\mathbf{M}$ is encrypted as $\mathbf{C} = (c_1, \ldots, c_n) = \mathbf{M} + \mathbf{GE}$ mod N.

3 Attacking the scheme

In this section we present a polynomial time heuristic attack that enables us to recover the private key from the public key. In particular, this attack enables us to factor N and recover the matrix $\mathbf{A}$ using the matrix $\mathbf{H}$ only. As mentioned in [2], once q and $\mathbf{A}$ are revealed, the system is totally broken.

The Lenstra–Lenstra–Lovász (LLL, or L^3) lattice reduction algorithm [3] is an algorithm which, given a lattice basis as input, outputs a basis with relatively

IEE Proceedings online no. 20060075
doi:10.1049/ip-ifs:20060075
Paper first received 24th April and in revised form 7th July 2006

The author is with the Concordia Institute for Information Systems Engineering, Concordia University, Montreal, Quebec, Canada, H3G 1M8
E-mail: youssef@ciise.concordia.ca

short vectors. The L^3 algorithm has found numerous applications in cryptanalysis of several public-key encryption schemes such as knapsack cryptosystems, RSA with particular settings and many other systems [5,6,7]. For a description of the L^3-lattice reduction algorithm and its applications in cryptanalysis, the reader is referred to [1,8].

The following lemma will be used in our attack:

Lemma 1: With the notation above we have

$$(a_i h_{ij} p) \text{mod } N = b_{ij}p, 1 \leq i,j \leq n.$$

Proof: From Equations (1) and (2) we have

$$\begin{aligned} a_i h_{ij} p \text{ mod } q &= (a_i(h_{ij} \text{ mod } q)p) \text{ mod } q \\ &= ((a_i g_{ij}) \text{ mod } q)p) \text{ mod } q \\ &= b_{ij} p \text{ mod } q. \end{aligned}$$

By noting that $|b_{ij}| \approx q^{0.1}$ and $p \approx q^{0.3}$, it is clear that $|b_{ij}\, p|$ is less than q and hence we have $b_{ij}\, p \text{ mod } q = b_{ij}p$.

The rest of the proof follows by using the Chinese remainder theorem [4] and noting that we have $(a_i\, h_{ij}\, p)$ mod $p = 0$. □

As was noted by one of the anonymous reviewers, from the above formulation, recovering the secret key $(a_1, \ldots, a_n)$ is reduced to n hidden number problems [9]. The basic steps in the attack are as follows:

1. For each row $(h_{i1}, h_{i2}, \ldots, h_{in})$, $1 \leq i \leq n$, in the matrix $\mathbf{H}$, use the L^3 algorithm to find a reduced basis $\mathbf{R}$ for the $(n+1)$-dimensional lattice $\mathbf{L}$ which is generated by the rows of the matrix

$$\begin{bmatrix} N & 0 & 0 & \cdots & 0 & 0 \\ 0 & N & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & N & 0 \\ h_{i1} & h_{i2} & h_{i3} & \cdots & h_{in} & 1 \end{bmatrix} \tag{3}$$

2. For each row $\mathbf{l} = (l_1, l_2, \ldots, l_n, l_{n+1})$ in R (starting from the shortest one) such that $l_{n+1} \neq N$ do the following:

- Evaluate $gcd(N, l_{n+1})$.
- If $gcd(N, l_{n+1}) \neq 1$, return $p = gcd(N, l_{n+1})$ and $a_i = l_{n+1}/gcd(N, l_{n+1})$.

3. Return failure

The following lemma is used to justify the success of the attack.

Lemma 2: The vector

$$\mathbf{x} = (b_{i1}p, b_{i2}p, \cdots, b_{in}p, a_i p)$$

is in $\mathbf{L}$ and has length less than $\sqrt{n(pq^{.1})^2 + (pq^{.5})^2} \approx pq^{0.5}$.

Proof: Using Lemma 1, the first part follows by noting that $\mathbf{x}$ is a linear combination of the rows of $\mathbf{L}$. The second part follows by noting that $|b_i| < b \approx q^{0.1}$ and $a_i \approx q^{0.5}$. □

Note that our lattice has dimension $(n+1)$ and volume N^n. From the lemma above, $\mathbf{x}$ is short compared with the $(n+1)^{th}$ root of the volume of the lattice. Hence, there is a good possibility that the L^3 algorithm will produce a reduced basis which includes the vector $\mathbf{x}$ [8], which is verified by our experimental results.

Let $\{\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_{n+1}\}$ denote the basis of the lattice $\mathbf{L}$ above. Let $C \in \mathbb{R}$ be such that $|\mathbf{r}_i|^2 \leq C$ for $i = 1, 2, \ldots, n+1$ and $|\mathbf{r}_i|$ denote the Euclidean norm of the basis vector $\mathbf{r}_i$. From [1], the number of arithmetic operations needed by the L^3 algorithm is $O((n+1)^4 logC)$, on integers of size $O((n+1)logC)$.

In order to reduce the attack complexity, once q is found, one can also apply the L^3 algorithm to the rows of the matrix $\mathbf{G} = \mathbf{H}$ mod q. In this case, for each row $(g_{i1}, g_{i2}, \ldots, g_{in}), 1 \leq i \leq n$, in the matrix $\mathbf{G}$, we use the L^3 algorithm to find a reduced basis $\mathbf{R}'$ for the $(n+1)$-dimensional lattice $\mathbf{L}'$ which is generated by the rows of the matrix

$$\begin{bmatrix} q & 0 & 0 & \cdots & 0 & 0 \\ 0 & q & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & q & 0 \\ g_{i1} & g_{i2} & g_{i3} & \cdots & g_{in} & 1 \end{bmatrix} \tag{4}$$

Similar to Lemma 2 above, it is easy to see that the vector

$$\mathbf{x}' = (b_{i1}, b_{i2}, \cdots, b_{in}, a_i)$$

is in $\mathbf{L}'$ and has length less than $\sqrt{n(q^{.1})^2 + (q^{.5})^2} \approx q^{0.5}$.

It is worth noting that the authors in [2] have discussed the lattice reduction attack and concluded that their proposed system is secure against this attack. This incorrect conclusion was drawn because they did not consider the right lattice to perform the attack.

It was also noted, by one of the anonymous reviewers, that our attack can be further optimised by multiplying all but the last column in the lattice bases (see equations 3, 4) by a factor of $q^{0.4}$. This balances the size of the entries in the target vector. Therefore, the norm of the target vector roughly stays the same while the determinant increases by a factor of $q^{0.4n}$.

4 Numerical example

In order to illustrate the steps in our cryptanalysis, we will use the same numerical example that was given in [2].

Let $p = 29$, $q = 29863$. Then $N = p \times q = 866027$. Let $\mathbf{A} = diag(98, 147, 125)$,

$$\mathbf{B} = \begin{bmatrix} 1 & -1 & 0 \\ 2 & 0 & -1 \\ -2 & 1 & -1 \end{bmatrix}, \text{ and } \mathbf{T} = \begin{bmatrix} 13 & 9 & 20 \\ 6 & 21 & 3 \\ 27 & 15 & 7 \end{bmatrix}$$

Then

$$\mathbf{G} = \begin{bmatrix} 21026 & 8837 & 0 \\ 8126 & 0 & 25600 \\ 24846 & 17440 & 12423 \end{bmatrix}, \text{ and}$$

$$\mathbf{H} = \begin{bmatrix} 588423 & 307467 & 656986 \\ 8126 & 776438 & 712649 \\ 622106 & 495248 & 400642 \end{bmatrix}$$

Consider the first row in $\mathbf{H}$. Using the L^3 algorithm (See algorithm 3.101 in [1]), the basis to be reduced is

$$\mathbf{L} = \begin{bmatrix} 866027 & 0 & 0 & 0 \\ 0 & 866027 & 0 & 0 \\ 0 & 0 & 866027 & 0 \\ 588423 & 307467 & 656986 & 1 \end{bmatrix}$$

The L^3-reduced basis (obtained using Maple 6.01) is

$$\begin{bmatrix} 29 & -29 & 0 & \mathbf{2842} \\ -8845 & 8845 & 0 & -783 \\ -56080 & -63372 & -29863 & -1048 \\ 78916 & 70399 & -179178 & -749 \end{bmatrix}$$

It is clear that the shortest vector (29, −29,0,2842) in the reduced basis above is in the form $(b_{11}p, b_{12}p,\ b_{13}p, a_1p)$.

Applying the lattice reduction algorithm to the other two rows of **H** we obtain the following two reduded bases:

$$\begin{bmatrix} 58 & 0 & -29 & \mathbf{4263} \\ -11774 & 0 & 5887 & 638 \\ -15229 & 59726 & -37180 & 531 \\ 62576 & 89589 & 118027 & 434 \end{bmatrix},$$

$$\begin{bmatrix} -58 & 29 & -29 & \mathbf{3625} \\ -13862 & 6931 & -6931 & 348 \\ 42980 & 38236 & -68099 & 1420 \\ -41315 & -113726 & -35589 & -962 \end{bmatrix}$$

Thus we obtain $\mathbf{A} = diag(2842/29, 4263/29, 3625/29) = diag(98, 147, 125)$ and

$$\mathbf{B} = \begin{bmatrix} 1 & -1 & 0 \\ 2 & 0 & -1 \\ -2 & 1 & -1 \end{bmatrix}$$

If **G** is to be used for the attack (after recovering q by any of the two attacks above), then the reduced bases obtained by considering the three rows in **G** are given by

$$\begin{bmatrix} 1 & -1 & 0 & \mathbf{98} \\ 305 & -305 & 0 & 27 \\ 14931 & 14932 & 0 & -49 \\ 0 & 0 & 29863 & 0 \end{bmatrix},$$

$$\begin{bmatrix} 2 & 0 & -1 & \mathbf{147} \\ -406 & 0 & 203 & 22 \\ 5891 & 0 & 11986 & -25 \\ 0 & 29863 & 0 & 0 \end{bmatrix},$$

$$\begin{bmatrix} -2 & 1 & -1 & \mathbf{125} \\ 478 & -239 & 239 & -12 \\ 9795 & 10034 & -10034 & 4 \\ -239 & -14812 & -15051 & 6 \end{bmatrix}$$

5 Conclusions

The system proposed by Baocang and Yupu is insecure. Using the L^3 lattice reduction algorithm, the system secret key can be derived from its known public key. We performed our attack on 10 random instances of the Baocang and Yupu system using the same security parameters suggested by Baocang and Yupu, i.e. $n = 16$ and $k = 256$. In all these 10 instances, we were able to fully recover the secret key in about 370 seconds (on average) using Maple 6.01, running on a DELL laptop with an Intel® Pentium® M processor, 1.6 GHz and 1.0 GB of RAM.

6 Acknowledgments

The author would like to thank the anonymous referees for their valuable comments. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada under Grant N00930.

7 References

1 Menezes, A.J., van Oorschot, P.C., and Vanstone, S.A.: 'Handbook of applied cryptographic research' (CRC Press, 1996)

2 Baocang, W., and Yupu, H.: 'Public key cryptosystem based on two cryptographic assumptions'. *IEE Proc., Commun.*, 2005, **152**Z, (6), pp. 861–865

3 Lenstra, A.K., Lenstra, H.W., and Lovász, L.: 'Factoring polynomials with rational coefficients'. *Math. Ann.*, 1982, **261**Z, pp. 515–534

4 Hardy, G.H., and Wright, E.M.: 'An introduction to the theory of numbers' (Oxford University Press, 1979, 5th edn.)

5 Nguyen, P., and Stern, J.: 'Cryptanalysis of the Ajtai-Dwork Cryptosystem'. Advances in Cryptology, Proc. CRYPTO '98, Santa Barbara, CA, USA, Aug. 1998, (*Lect. Notes Comput. Sci.,* **1462**), pp. 223–242

6 Nguyen, P., and Stern, J.: 'Cryptanalysis of a fast public key cryptosystem presented at SAC '97'. Proc. Workshop on Selected Areas in Cryptography 1998, Kingston, Canada, Aug. 1998, (*Lect. Notes Comput. Sci.,* **1556**), pp. 213–218

7 Youssef, A.M., and Gong, G.: 'Cryptanalysis of a public key cryptosystem proposed at ACISP 2000'. Proc. ACISP01, Sydney, Australia, July 2001, (*Lect. Notes Comput. Sci.,* **2119**), pp. 15–20

8 Nguyen, P., and Stern, J.: 'Lattice reduction in cryptology: An update'. Algorithmic Number Theory, Proc. ANTS-IV, July 2000, (*Lect. Notes Comput. Sci.,* **1838**), pp. 85–112

9 Boneh, D., and Venkatesan, R.: 'Hardness of computing the most significant bits of secret keys in Diffie-Hellman and related schemes'. Proc. CRYPTO '96, Santa Barbara, CA, USA, Aug. 1996, (*Lect. Notes Comput. Sci.,* **1109**), pp. 129–142