International Journal of Information & Network Security (IJINS)
Vol.1, No.4, October 2012, pp. 275~281
ISSN: 2089-3299

Attack on Fully Homomorphic Encryption over the Integers

Gu Chun-sheng*, **
* School of Computer Science and Technology, University of Science and Technology of China
** School of Computer Engineering, Jiangsu Teachers University of Technology

Article Info

Article history:

Received Jul 17th, 2012
Accepted Aug 26th, 2012

Keyword:

Fully Homomorphic Encryption
Cryptanalysis
Lattice Reduction
Approximate Matrix GCD

ABSTRACT

Recently, many fully-homomorphic encryption schemes have been constructed. However, the issue of the security of these fully homomorphic encryptions has not been carefully studied. By using lattice reduction algorithm, we firstly present an attack on the fully homomorphic encryption based on approximate GCD over the integers. Our result shows that the FHE in [4] is not secure for some parameter settings. Then, we define approximate matrix GCD problem, which is a generalization of approximate GCD. Finally, we construct an improvement FHE scheme based on approximate matrix GCD to avoid the lattice attack in this paper.

Corresponding Author:

Gu Chun-sheng,
School of Computer Engineering,
Jiangsu Teachers University of Technology,
1801 Zhongwu Main Road, Zhonglou District, Changzhou City, Jiangsu Province, China, 213001.
Email: guchunsheng@gmail.com

1. INTRODUCTION

Rivest, Adleman and Dertouzos [1] introduced a notion of privacy homomorphism. But until 2009, Gentry [2] constructed the first fully homomorphic encryptions based on ideal lattice, all previous schemes are insecure. Following the breakthrough of [2], there is currently great interest on fully-homomorphic encryptions [3-12]. In these schemes, the simplest one is certainly the one of van Dijk, Gentry, Halevi and Vaikuntanathan [4]. The public key of this scheme is a list of approximate multiples $\{x_i = q_i p + 2r_i\}_{i=1}^{\tau}$ for an odd integer p, where q_i, r_i is the uniform random integers over Z such that $|r_i| < 2^{\lambda-1}$. The secret key is p. To encrypt a message bit m, the ciphertext is evaluated as $c = \sum_{i \in T, T \subseteq [\tau]} x_i + 2r + m$, where $|r| < 2^{\lambda-1}$. To decrypt a ciphertext, compute the message bit $m = [c]_p \bmod 2$, where $[c]_p$ is an integer in $(-p/2, p/2)$.

To conveniently compare, we simply describe the known attacks considering in Section 5 and appendix B in [4]. Section 5 in [4] considered known attacks on the AGCD problem for two numbers (x_0, x_1) and many numbers $(x_0, \cdots, x_t)$. These attacks mainly discussed how to solve approximate GCD problem, i.e. the secret key p.

The appendix B.1 in [4] analyzed Nguyen and Stern's orthogonal lattice attack. Given $\vec{x} = (x_0, \ldots, x_t) = p\vec{q} + \vec{r}$, where $\vec{q} = (q_0, \ldots, q_t)$ and $\vec{r} = (r_0, \ldots, r_t)$, consider a t-dimensional lattice

$L_{\vec{x}}^{\perp}$ of integer vectors orthogonal to $\vec{x}$. It is easy to verify that any vector that is orthogonal to both $\vec{q}$ and $\vec{r}$, is in the lattice $L_{\vec{q},\vec{r}}^{\perp}$, it is also in $L_{\vec{x}}^{\perp}$. According to [4], the idea of attack is to reduce $L_{\vec{x}}^{\perp}$ to recover $t-1$ linearly independent vectors of $L_{\vec{q},\vec{r}}^{\perp}$, and further recover $\vec{q}$ and $\vec{r}$, and p. Then [4] discussed when $t > \gamma/(\eta-\rho)$, the lattice reduction algorithm can not find a $2^{\eta-\rho}$ approximate short vector in $L_{\vec{q},\vec{r}}^{\perp}$ on the worst-case.

Dijk et al. [4] also analyzed a similar attack by using the constraint $x_i - r_i = 0 \bmod p$, which paid close attention to how to solve for $\vec{r}$. [4] considered a following lattice.

$$M = \begin{pmatrix} x_1 & R_1 & & & \\ x_2 & & R_2 & & \\ & & & \ddots & \\ x_t & & & & R_t \end{pmatrix}.$$

But one needs to find t linearly independent short vectors of the lattice M to obtain the success of this attack. That is, each l_1 norm among t vectors is at most $p/2$. When t is large, solving these vectors is very difficult by using lattice reduction algorithm.

Instead of applying linear system $x_i - r_i = 0 \bmod p$, Coppersmith's method looks at quadratic system $(x_i - r_i)^2 = 0 \bmod p^2$ and $(x_i - r_i)(x_j - r_j) = 0 \bmod p^2$, etc, and finds one of the r_i and thereof p and all other r_i's by solving some small vectors in new lattices.

In a word, the attacks considering in Section 5 and appendix B in [4] is how to recover the secret key p, and the security analysis depends on the worst-case performance of the currently known lattice reduction algorithms.

The lattice in this paper is very similar to the lattice M. However, our attack only requires find one short vector with certain condition, and not to solve t short vectors. Moreover, our attack merely recovers the plaintext bit from a ciphertext and depends upon the average-case performance of the lattice reduction algorithms. On the other hand, if suppose $\vec{x} = (c, x_0, \ldots, x_t) = p\vec{q} + 2\vec{r} + m$ with a ciphertext c, then our attack in some sense is similar to solving a short vector of orthogonal lattice $L_{\vec{q}}^{\perp}$, which is different from the lattices $L_{\vec{x}}^{\perp}$ or $L_{\vec{q},\vec{r}}^{\perp}$ considering in Section 5 and appendix B in [4].

Currently, many fully-homomorphic encryption schemes [2-12] have been designed. However, the issue of the security of these fully homomorphic encryptions has not been carefully studied. Chen and Nguyen [18] presented a $2^{3\rho/2}$ time algorithm for the AGCD problem, which is improved to 2^{ρ} time in [11]. In [13], Gu and Gu proved that the FHEs in [3, 6] are not secure for the practical parameter settings by using lattice reduction algorithm. Our main observation is that one can directly obtain the plaintext from a ciphertext by using lattice reduction algorithm, without using the secret key for some parameter settings of the FHE in [4]. Our attack is different from the known attacks considering in [4]. Because the attacks they considered are to solve the secret key. So, our result shows the FHE in [4] is not secure for some practical parameters. Section 2 gives some notations and definitions, and the lattice reduction algorithms. Section 3 constructs a new lattice based on the public key, and presents a polynomial time algorithm to directly obtain the plaintext from a ciphertext. Section 4 presents an improvement FHE scheme. Section 5 concludes this paper.

2. Preliminaries

2.1 Notations

In this paper, we follow the parameter setting of [4]. Let λ be a security parameter, $[\lambda] = \{1, \ldots, \lambda\}$ be a set of integers. Let γ be bit-length of the integers in the public key, η the bit-length of the secret key, ρ the bit-length of the noise, and τ the number of integers in the public key. To conveniently describe, we concretely set $\rho = \lambda$, $\eta = 4\lambda^2$, $\gamma = \lambda^5$, and $\tau = \gamma + \lambda$ throughout this paper.

2.2 Lattice Reduction Algorithm

Given a basis of the lattice $b_1,\ldots,b_n$, one of the most famous problems of the algorithm theory of lattices is to find a short nonzero vector. Currently, there is no polynomial time algorithm for solving a shortest nonzero vector in a given lattice. The most celebrated LLL reduction finds a vector whose approximating factor is at most $2^{(n-1)/2}$. In 1987, Schnorr [14] introduced a hierarchy of reduction concepts that stretch from LLL reduction to Korkine-Zolotareff reduction which obtains a polynomial time algorithm with $(4k^2)^{n/2k}$ approximating factor for lattices of any rank. The running time of Schnorr's algorithm is poly(size of basis)*HKZ(2*k*), where HKZ(2*k*) is the time complexity of computing a 2*k*-dimensional HKZ reduction, and equal to $O(k^{k/2+o(k)})$. If we use the probabilistic AKS algorithm [15], HKZ(2*k*) is about $O(2^{2k})$.

Theorem 2.1 (Theorem 2.6 [14]) Every block $2k$-reduced basis $b_1,\ldots,b_{mk}$ of lattice L satisfies $\|b_1\| \le \sqrt{\gamma_k}\,\beta_k^{\frac{m-1}{2}}\lambda_1(L)$, where β_k is another lattice constant using in Schnorr's analysis of his algorithm. Shnorr [13] showed that $\beta_k \le 4k^2$, and Ajtai improved this bound to $\beta_k \le k^{\varepsilon}$ for some positive number $\varepsilon > 0$. Recently, Gama Howgrave, Koy and Nguyen [16] improved the approximation factor of the Schnorr's 2*k*-reduction to $\|b_1\| / \lambda_1(L) \le \sqrt{\gamma_k}(4/3)^{(3k-1)/4}\beta_k^{n/2k-1}$, and proved the following result via Rankin's constant.

Theorem 2.2 (Theorem 2, 3 [16]) For all $k \ge 2$, Schnorr's constant β_k satisfies: $k/12 \le \beta_k \le (1+k/2)^{2\ln 2+1/k}$. Asymptotically it satisfies $\beta_k \le 0.1\times k^{2\ln 2+1/k}$. In particular, $\beta_k \le k^{1.1}$ for all $k \le 100$.

Theorem 2.3 ([17]). For lattice L, the first vector b_1 output by LLL is satisfied to the ratio $\|b_1\| / \lambda(L) \approx (1.02)^n$ on the average.

3. Lattice Attack on FHE

To simplicity, we first refer the FHE in [4], then construct a new lattice based on the public key and an arbitrary ciphertext to recover the plaintext from the ciphertext by applying LLL lattice reduction algorithm.

3.1. Fully Homomorphic Encryption

KeyGen(λ). The secret key is a random odd η-bit integer: $p \xleftarrow{\Psi} (2\square+1)\cap[2^{\eta-1},2^{\eta})$. Select $q_0,\ldots,q_\tau \xleftarrow{\Psi} \square\cap[0,2^\gamma/p)$ with the largest odd integer q_0. Select $r_0,\ldots,r_\tau \xleftarrow{\Psi} \square\cap[-2^\rho,2^\rho]$, compute $x_0 = q_0 p + 2r_0$ and $x_i = [q_i p + 2r_i]_{x_0}$ for $i \in [\tau]$. Output the public key $pk =< x_0, x_1,\ldots,x_\tau >$ and the secret key $sk =< p >$.

Encrypt(pk, $m \in \{0,1\}$). Select a random subset $T \subseteq [\tau]$ and $r \xleftarrow{\Psi} \square\cap[-2^\rho,2^\rho]$, and output ciphertext $c = \left[m + 2r + \sum_{i\in T} x_i\right]_{x_0}$.

Decrypt(sk, c). Output $m' = \left[[c]_p\right]_2$.

To implement fully homomorphic encryption scheme, [4] applied the Gentry's standard bootstappable technique.

3.2. Lattice Attack Based on the Public Key

Given a list of approximate multiples of p:

$$\{x_i = q_i p + r_i : q_i \in \square\cap[0,2^\gamma/p), r_i \in \square\cap(-2^\rho,2^\rho)\}_{i=0}^{\tau}, \text{ find } p.$$

Dijk et al. [4] showed that the security of the FHE is equivalent to solving the approximate GCD problem. Chen and Nguyen [18] presented a new AGCD algorithm running in $2^{3\rho/2}$ polynomial-time operations, which is essentially the $3/4$ -th root of that of GCD exhaustive search.

According to FHE, we know that an arbitrary ciphertext has the form $c = qp + 2r + m$. The ideal of attack is very simple, that is, one is how to remove qp in c by adding small noise. When completing this, it is easy to recover the plaintext bit m in c . To do this, we define following Diophantine inequality equation problem.

Definition 3.1. (Diophantine Inequality Equation (DIE)). Given a list of integers $\{x_i = q_i p + r_i : q_i \in \mathbb{Z} \cap [0, 2^{\gamma}/p), r_i \in \mathbb{Z} \cap (-2^{\rho}, 2^{\rho})\}_{i=0}^{\tau}$, solve the Diophantine inequality equation $\left|\sum_{i=0}^{\tau} y_i x_i\right| < p/8$ subject to $|y_i| < p/(8\tau 2^{\rho})$ and at least one non-zero y_i .

Suppose there is an oracle to solve DIE, then one can obtain the plaintext bit in an arbitrary ciphertext of FHE [4]. Since $|y_i| < p/(8\tau 2^{\rho})$, $\left|\sum_{i=0}^{\tau} y_i r_i\right| < p/8$, that is, $\sum_{i=0}^{\tau} y_i x_i$ is only the sum of noise terms, without non-zero multiple of p . So, one can correctly decide the plaintext bit of a ciphertext according to the parity of $\sum_{i=0}^{\tau} y_i x_i$.

However, it is not difficult to see that DIE is a generalization of the knapsack problem. So, there is unlikely an efficient algorithm for DIE unless P=NP. However, this does not mean that there is not a polynomial time algorithm for special DIE.

Given the public key $pk = < x_0, x_1, \ldots, x_{\tau} >$ and a ciphertext c , we randomly choose a subset T from $[\tau]$ such that $|T| = \lambda^3$. Without generality of loss, assume $T = [\lambda^3]$ and $c = qp + 2r + m$ with $|2r| \le 2^{\rho}$. We construct a new lattice L as follows:

$$L = \begin{pmatrix} c & 0 & \cdots & 0 & 0 \\ -x_1 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ -x_{\lambda^3} & 0 & \cdots & 1 & 0 \\ -x_0 & 0 & \cdots & 0 & 1 \end{pmatrix}, L_1 = \begin{pmatrix} c & 1 & 0 & \cdots & 0 & 0 \\ -x_1 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ -x_{\lambda^3} & 0 & 0 & \cdots & 1 & 0 \\ -x_0 & 0 & 0 & \cdots & 0 & 1 \end{pmatrix}.$$

On the one hand, the size of the shortest vector of lattice L is less than $\sqrt{\lambda^3 + 2}\,|c|^{1/(\lambda^3+2)} \approx 2^{\lambda^2}$ according to the parameter setting. On the other hand, there is a non-zero solution $\left|\sum_{i=0}^{\lambda^3} y_i x_i + yc\right| \le 2^{\lambda^2}$ with $|y_i| \le 2^{\lambda^2}$ and $|y| \le 2^{\lambda^2}$ by using pigeon hole principle. This is because $|c|, |x_i| \le 2^{\lambda^5}$, the number of all distinct y_i, y subject to $|y|, |y_i| \le 2^{\lambda^2}$ is $(2^{\lambda^2})^{\lambda^3+2} > 2^{\lambda^5}$, that is, there is at least a non-zero solution for the equation $\left|\sum_{i=0}^{\lambda^3} y_i x_i + yc\right| \le 2^{\lambda^2}$. Thus, if one finds a non-zero small solution vector, then one gets the plaintext bit with probability at least 1/2 (y is an odd integer).

To conveniently decide, we use a variant lattice L_1 of L , and call LLL algorithm for lattice L_1 . Assume $b = (b_0, b_1, \ldots, b_{\lambda^3+1})$ is the first vector of the L_1 's basis output by LLL. If $\|b\|_{\infty} < p/(8\lambda^3 2^{\lambda})$ and $\mathrm{mod}(b_1, 2) = 1$, then $m = \mathrm{mod}(b_0, 2)$. In our experiment, we notice that b_1 may be an even integer, but the several vectors following the first vector (such as the second vector, or the third vector, et al.) often satisfy the above condition. That is, the first coordinate of vector is odd and its norm is small. So, as long as one gets one solution of the above form, one can correctly decide the plaintext bit. In fact, LLL can also be called many times for distinct subset T . So, we have the following result by applying the block lattice reduction.

Theroem 3.1. Suppose the parameters of FHE [4] $\lambda \le 100$, $\rho = \lambda$, $\eta = 5\lambda^2$, $\gamma = \lambda^5$, and $\tau = \gamma + \lambda$, then there is a running time $2^{\theta\lambda}, (\theta \le 1)$ algorithm recovering plaintext from ciphertext.

Proof: According to Theorem 2.1, 2.2, we know $\|b_1\| / \lambda_1(L) \le \sqrt{\gamma_k}(4/3)^{(3k-1)/4} \beta_k^{n/2k-1}$ and $\beta_k \le k^{1.1}$ for all $k \le 100$. If we choose $k = \lambda, n = \lambda^3$, then $\|b_1\| \approx \lambda^{1.1\times\lambda^3/2\lambda} \times \lambda_1(L) \approx 2^{3.66\lambda^2}\lambda_1(L) \le 2^{4.66\lambda^2} << 2^{\eta}$. By using AKS algorithm [15, 19], solving each block sub-lattice costs time $2^{\delta\lambda}, \delta < 1$, and the number of blocks is at most $\lambda^{O(1)}$. So, the total running time of algorithm is $2^{\theta\lambda}, \theta \le 1$.■

The cost $2^{\theta\lambda}, \theta \le 1$ breaking FHE in [4] is smaller than $2^{1.5\lambda}$ in [18].

Theorem 3.2. Suppose the average-case performance of LLL is true, namely, Theorem 1.3 holds. Then, for the parameters $\lambda \le 100$, $\rho = \lambda$, $\eta = 4\lambda^2$, $\gamma = \lambda^5$, and $\tau = \gamma + \lambda$, the FHE in [4] is insecure.

Proof: For the above lattice L_1, we have

$$\|b\| \le (1.02)^{\lambda^3+2}\lambda(L_1) \le (1.02)^{100\lambda^2+2}\lambda(L_1) \approx 7.2^{\lambda^2}\lambda(L_1) << 2^{4\lambda^2}$$

.■

Moreover, we verify for $\lambda = 3, 4, \ldots, 10$ the correctness and the efficiency of attack by computational experiment applying NTL [20].

4. Improvement of FHE [4]

The reason why the lattice attack is successful is that the secret key p is a large integer. If we replace p by a matrix, then the above attack dose not work.

Before giving improvement scheme, we firstly define approximate matrix GCD problem over the integers, which is a generalization of approximate GCD problem.

Definition 4.1 (approximate matrix GCD over the integers). Given a list matrices $B_i = R_i A + 2r_i \square I \in \square_p^{2\times 2}$, where $R_i \in \square_p^{2\times 2}$, $\det(A) = p$, r_i is a small integer, I is identity matrix, find the matrix A.

4.1 Construction

Key Generating Algorithm (KeyGen)

(1) Select a random matrix $T \in Z^{2\times 2}$ with $\|T\|_\infty = 2^{O(\lambda^2)}$ such that $p = \det(T) = 2^{O(\lambda^2)}$ and $p \bmod 2 = 1$. Compute $A \in Z^{2\times 2}$ with $AT = pI$, where I is identity matrix.

(2) Generate $\tau = O(\lambda \log \lambda)$ matrices $\{B_i = (R_i A + 2r_i \square I) \bmod p\}_{i=1}^{\tau}$, where $R_i \in \square_p^{2\times 2}$ is an uniformly random matrix and $|r_i| \le 2^{\lambda}$ and r_i is integer.

(3) Output the public key $pk = (p, B_i, i \in [\tau])$ and the secret key $sk = (p, T)$.

Encryption Algorithm (Enc)

Given the public key pk and a bit $m \in \{0,1\}$. Evaluate ciphertext $C = (\sum_{i\in[\tau]} k_i B_i + (m + 2r)I) \bmod p$ where $|k_i| \le 2^{\lambda}$ and r is integer.

Add Operation (Add)

Given the public key pk and ciphertexts C_1, C_2, output new ciphertext $C = (C_1 + C_2) \bmod p$.

Multiplication Operation (Mul)

Given the public key pk and ciphertexts C_1, C_2, output new ciphertext $C = (C_1 \times C_2) \bmod p$.

Decryption Algorithm (Dec)

Given the secret key sk and ciphertext C, decipher $M = (C \times T) \bmod p \bmod 2$, and the plaintext m is the element $m = M_{1,1}$ of the first row and the first column of M.

It is not difficult to verify that the improvement scheme is a somewhat homomorphic encryption. Now, one may use the Gentry's standard bootstrappable technique to implement fully homomorphic encryption.

In addition, we can also choose two random primes $p, q = 2^{O(\lambda^2)}$ with $p = a^2 + b^2$ i.e. $p \equiv 1 \bmod 4$. Set $n = pq$ and $T = \begin{pmatrix} a & b \\ -b & a \end{pmatrix}$, $A = \begin{pmatrix} a & -b \\ b & a \end{pmatrix}$ with $AT = \begin{pmatrix} p & 0 \\ 0 & p \end{pmatrix} = pI$. Now, we can replace p with $n = pq$, and use this matrix A to generate the public key $pk = (n, B_i, i \in [\tau])$. We observe that the security of this modification depends on the hardness of factoring $n = pq$.

4.2 Efficiency and Security

Efficiency. The size of the public key is $O(\lambda^3 \log \lambda)$, the size of the secret key is $O(\lambda^2)$, the expansion rate of ciphertext to plaintext is $O(\lambda^2)$. To implement FHE, one only needs to add ciphertexts of the secret key to the public key.

Security. Currently, we can not reduce the security of the improvement scheme to the approximate matrix GCD problem. So, we suppose that the approximate matrix GCD problem is hard in this paper. In the following, we only consider another possible attack for the scheme.

It is not feasible to use brute force attack by guessing noise term r because $|r| = O(2^{\lambda})$. A possible attack is to solve the following equation

$$\begin{cases} TB_1 = r_1 T \bmod p \\ TB_2 = r_2 T \bmod p \\ \quad\vdots \\ TB_\tau = r_\tau T \bmod p \end{cases}$$

However this system consists of quadratic equations when r_i is unknown. So, to solve this equation, we also require to guess r_i. As well as we know, attacking this scheme is not feasible by using algebraic equation method.

Moreover, this scheme can also avoid the attack in this paper because B_i is an approximate multiple of the secret key A.

5. Conclusion

This paper presents a lattice attack for the FHE in [4] by directly calling LLL algorithm. This attack mainly recovers plaintext from ciphertext. Our result shows that the FHE is not secure for some parameter settings in [4]. According to our experiment, the lattice attack can be avoided by taking larger parameter $\gamma = \lambda^6$. But, the scheme is less practical in this case.

Avoid the above lattice attack, we present an improvement FHE scheme based on approximate matrix GCD over the integers. However, we do not know the hardness of approximate matrix GCD problem. Thus, the security of the improvement FHE scheme remains open.

ACKNOWLEDGEMENTS

This work is supported by National Natural Science Foundation of China Grants (No. 61142007), by Foundation of Jinagsu Province 'Qinglang Project' (No.KYQ09002), and by Foundation of Jiangsu Teachers University of Technology (No. KYY11055).

REFERENCES

[1] R. Rivest, L. Adleman, and M. Dertouzos. On data banks and privacy homomorphisms. In Foundations of Secure Computation, pages 169-180, 1978.
[2] C. Gentry. Fully homomorphic encryption using ideal lattices. STOC 2009, pp. 169-178, 2009.
[3] Nigel P. Smart and Frederik Vercauteren. Fully homomorphic encryption with relatively small key and ciphertext sizes. PKC 2010, LNCS 6056, pp. 420–443.
[4] M. van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan. Fully homomorphic encryption over the integers. Eurocrypt 2010, LNCS 6110, pp. 24-43.
[5] D. Stehle and R. Steinfeld. Faster Fully Homomorphic Encryption. Asiacrypt 2010, LNCS 6477, pp. 377-394.
[6] Craig Gentry and Shai Halevi. Implementing Gentry's fully-homomorphic encryption scheme. Eurocrypt 2011, LNCS 6632, pp. 129–148.
[7] C. Gentry and S. Halevi, Fully homomorphic encryption without squashing using depth-3 arithmetic circuits, FOCS 2011, pp. 107-109.
[8] Zvika Brakerski and Vinod Vaikuntanathan. Fully homomorphic encryption from ring-LWE and security for key dependent messages, CRYPTO 2011, pp. 505-524.
[9] Zvika Brakerski and Vinod Vaikuntanathan. Efficient Fully Homomorphic Encryption from (Standard) LWE, FOCS 2011, pp. 97-106.
[10] Z. Brakerski, C. Gentry, and V. Vaikuntanathan. Fully homomorphic encryption without bootstrapping. In ITCS 2012.
[11] J. S. Coron, A. Joux, A. Mandal, D. Naccache, and M. Tibouchi. Public Key Compression and Modulus Switching for Fully Homomorphic Encryption over the Integers. EUROCRYPT 2012, LNCS 7237, pp. 446-464.
[12] J. S. Coron, A. Mandal, D. Naccache, and M. Tibouchi. Fully homomorphic encryption over the integers with shorter public-keys. CRYPTO 2011, LNCS 6841, pp. 487-504.
[13] Gu Chunsheng, Gu Jixing. Cryptanalysis of the Smart-Vercauteren and Gentry-Halevi's Fully Homomorphic Encryption, International Journal of Security and Its Applications, Vol.6, No.2, 2012,pp.103-108.
[14] C.P. Schnorr. A hierarchy of polynomial time lattice basis reduction algorithms. Theoretical Computer Science, 53, pp. 201-224, 1987.
[15] M. Ajtai, Ravi Kumar, and D. Sivakumar. A sieve algorithm for the shortest lattice vector problem. STOC 2001, pp 601-610.
[16] N. Gama, N. Howgrave-Graham, H. Koy, and P. Q. Nguyen. Rankin's constant and blockwise lattice reduction. CRYPTO 2006, pp. 112–130.
[17] P.Q. Nguyen and D. Stehle, LLL on the average, proc. ANTS VII, 2006, LNCS 4076, pp. 238-256.
[18] Y. Chen and P. Q. Nguyen. Faster algorithms for approximate common divisors: Breaking fully homomorphic encryption challenges over the integers, EUROCRYPT 2012, LNCS 7237, pp. 502-519.
[19] D. Micciancio P. Voulgari, Faster exponential time algorithms for the shortest vector problem, SODA 2010, pp. 1468-1480.
[20] V. Shoup. NTL: A Library for doing Number Theory. http://shoup.net/ntl/, Version 5.5.2, 2009.

BIOGRAPHY OF AUTHORS

Gu Chun-sheng received his Ph.D. Degree from University of Science and Technology of China in 2005. Since 2008 he has been an associate professor in the School of Computer Engineering, Jiangsu Teachers University of Technology. His research interests are in the cryptanalysis and design of cryptography.