Cryptanalysis of Smart-Vercauteren and Gentry-Halevi's Fully Homomorphic Encryption

Gu Chun-sheng[1,2] and Gu Ji-xing[3]

[1] *School of Computer Engineering, Jiangsu Teachers University of Technology, China Changzhou 213001*
[2] *School of Computer Science and Technology, University of Science and Technology of China, China Hefei 230027*
[3] *Institute of Image Communication & Information Processing, Shanghai Jiaotong University, China Shanghai 200240*

guchunsheng@gmail.com

Abstract

For the fully homomorphic encryption schemes in [3, 6], this paper presents attacks to solve an equivalent secret key and directly recover plaintext from ciphertext for lattice dimensions n=2048 with lattice reduction algorithm. Given the average-case behavior of LLL in [8] is true, then their schemes are also not secure for n=8192.

Keywords: *Fully Homomorphic Encryption, Cryptanalysis, Principal Ideal Lattice, Lattice Reduction*

1. Introduction

Rivest, Adleman and Dertouzos [1] first presented the concept of homomorphic encryption, which had been widely used in cryptography. It became insecure by 2009 when Gentry [2] constructed the first fully homomorphic encryptions based on ideal lattice. After the scheme of [2], Smart and Vercauteren [3] presented an refined FHE with smaller ciphertext and key by leveraging principal ideal lattice. Dijk, Gentry, Halevi and Vaikuntanathan [4] proposed a simple fully homomorphic encryption scheme over the integers, whose security depends on the hardness of solving approximate GCD over the integers. Stehle and Steinfeld [5] improved Gentry's fully homomorphic scheme and obtained a faster fully homomorphic scheme. Similar to [3], Gentry and Halevi [6] implemented Gentry's scheme by applying principal ideal lattice. The security of FHE's in [3, 6] depends on the hardness assumption of finding small principal ideal lattice, given its HNF form or two elements form. This paper will present two lattice attacks for FHE's in [3, 6].

1.1 Our Results

In the following attack, we use the concrete parameters of FHE in [3, 6]. By using polynomial time block lattice reduction algorithm [7], we solve an equivalent secret key for n=2048 of FHE's in [3, 6]. Considering the average case behavior of LLL [8], the ratio $\|b_1\| / \lambda(L)$ is about $(1.02)^n$, i.e. $\|b_1\| \le (1.02)^n \lambda_1(L) << 2^{380} \lambda_1(L)$ for n=8192, where 380 is the bit-size of the coefficients in the generator polynomial of [6]. As a result, our first result shows the FHE's in [3, 6] are not secure for n=8192.

Our second result is to directly recover plaintext from ciphertext for n=2048, 8192 by applying average case behavior of the LLL algorithm [8].

1.2 Organization

Section 2 gives some notations and the lattice reduction algorithms. Section 3, 4 analyze the security of FHE's in [3, 6]. Section 5 presents an attack by directly recovering plaintext from ciphertext for FHE's [3, 6]. Section 6 concludes this paper and gives two open problems.

2. Preliminaries

2.1 Notations

Let n be security parameter, $[n]=\{0,1,...,n\}$. Let R be a ring of integer polynomials modulo $f_n(x)$, i.e., $R=\mathbb{Z}[x]/f(x)$, where $f_n(x)$ is an irreducible polynomial of degree n over the integers. Let R_p denote the polynomial ring $\mathbb{Z}_p[x]/f(x)$ over modulo p. For $\forall u\in R$, we denote by $\|u\|_\infty$ the infinity norm of u, $\bar{u}=[u_0,...,u_{n-1}]$ the coefficient vector of u, $[u]_2$ the polynomial of u's coefficients modulo 2. For the ring R, its expansion factor is n, that is, $\|u\times v\|_\infty \le n\cdot\|u\|_\infty\cdot\|v\|_\infty$, where $\times$ is multiplication over R.

2.2 Lattice Reduction Algorithm

Given a basis of lattice $b_1,...,b_n$, 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 algorithm [9] finds a vector whose approximating factor is at most $2^{(n-1)/2}$. In 1987, Schnorr [10] introduced a hierarchy lattice reduction that generalizes 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)$\times$HKZ($2k$), where HKZ($2k$) is equal to $O(k^{k/2+o(k)})$, which is the time complexity of computing a $2k$-dimensional HKZ reduction. If the probabilistic algorithm in [11] is used, HKZ($2k$) is about $O(2^{2k})$. To guarantee computation feasible, we will choose $k=16$ in the following.

Theorem 2.1 (Theorem 2.6 [10]) Every block $2k$-reduced basis $b_1,...,b_{mk}$ 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 algorithm analysis.

Gama Howgrave, Koy and Nguyen [7] improved the approximation factor of Schnorr's $2k$-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 [7]) 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 ([8]). Suppose the average case behavior of LLL is true, then the first vector b_1 of LLL reduced basis is satisfied to $\|b_1\|/\lambda(L)\approx(1.02)^n$ on the average for L.

3. Cryptanalysis of Smart-Vercauteren's Scheme

3.1 Fully Homomorphic Encryption

Key Generation Algorithm (KeyGen).

(1) Choose a random polynomial $u(x)=\sum_{i=0}^{n-1}u_ix^i\in Z[x]$ such that $\|u(x)\|_\infty$ is a η-bit integer,

$u(x) = 1 \bmod 2$, and $p = \det(Rot(u(x)))$ is a prime.

(2) Compute $d(x) = \gcd(u(x), f_n(x))$ over $F_p[x]$. Assume $\alpha \in F_p$ is the unique root of $d(x)$.

(3) Apply the XGCD-algorithm over $Q[x]$ to obtain $v(x) = \sum_{i=0}^{n-1} v_i x^i \in Z[x]$ such that $u(x) \times v(x) = p \bmod f_n(x)$.

(4) Set $\beta = (v(x) \bmod x) \bmod (2p)$.

(5) Output the public key $pk = (p, \alpha)$, the secret key $sk = (p, \beta)$.

Encryption Algorithm (Enc). Given the public key pk and $m \in \{0,1\}$, choose a small random polynomial $r(x)$ with $\|r(x)\|_\infty$ is a μ-bit integer. Output a ciphertext $c = (2r(\alpha) + m) \bmod p$.

Decryption Algorithm (Dec). Given the secret key sk and a ciphertext c, decipher a bit $m = (c - \lfloor c \times \beta / p + 0.5 \rfloor) \bmod 2$.

All other details of FHE are omitted (see [3]).

3.2 Cryptanalysis of Smart-Vercauteren's FHE

According to KeyGen, $\gamma = u(x)$ is an element of prime norm in the number field K defined by $f_n(x)$, and α is a root of $f_n(x) \bmod p$. Namely, we get the prime ideal $I = \gamma \square Z[x] = p \square Z[x] + (x - \alpha) \square Z[x]$, and $u(\alpha) = 0 \bmod p$.

On the surface, one must get the secret key $v(x)$ to attack FHE. In fact, if one obtains a small multiple $w(x) = \delta(x) \times v(x)$ of $v(x)$, where $\delta(x)$ is a polynomial with small integer coefficent, one can directly decrypt a ciphertext. Since $C(x) = c + q(x) \times \gamma$, so $\delta(x) \times (C(x) - c) = \lfloor c \square w(x) + 0.5h \rfloor \times \gamma = q'(x) \times \gamma$, where $q'(x) = \delta(x) \times q(x)$. So, $[\delta(x) \times (C(x) - c)]_2 = q'(x)$ via $[\gamma]_2 = 1$. So, one randomly selects a small coefficient polynomial $C(x)$, generates a ciphertext $c = C(\alpha) \bmod p$, and solves $[\delta(x)]_2$ by the above equation. Once one knows $w(x)$ and $[\delta(x)]_2$, one can decipher arbitrary ciphertext with small noise. Thus, we only need to present an algorithm which generates a suitable polynomial $w(x)$.

Theorem 3.1. Given a principal ideal π in either two element (p, α) or HNF representation, there is a polynomial time algorithm which finds $w(x) = \delta(x) \times v(x)$ over $\square$ such that $\|\delta(x)\|_\infty \le \sqrt{\gamma_k} (4/3)^{(3k-1)/4} \beta_k^{n/2k-1}$.

Proof. Since α is a root of $f_n(x) = x^n + 1$ over modulo p, so $x^n + 1 = (x - \alpha) \square g(x) \bmod p$. It is easy to verify $g(x) = t(x) \square v(x) \bmod p$. Assume $g(x) = x^{n-1} + g_{n-2} x^{n-2} + \cdots + g_0$. One constructs the following lattice M.

$$M = \begin{pmatrix} g_0 & g_1 & \cdots & g_{n-2} & 1 \\ -1 & g_0 & \cdots & g_{n-3} & g_{n-2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ -g_1 & -g_2 & \cdots & -1 & g_0 \\ p & 0 & \cdots & 0 & 0 \\ 0 & p & \cdots & 0 & 0 \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & p \end{pmatrix}.$$

To reduce lattice M, one calls the lattice reduction algorithm in [7,10]. By Theorem 2.1, 2.2, one gets $w(x) = \delta(x) \times v(x)$ such that $\|\delta(x)\|_\infty \le \|\delta(x)\|_2 \le \sqrt{\gamma_k} (4/3)^{(3k-1)/4} \beta_k^{n/(2k)-1}$. Recall that $w(x) \in R$ since $u(x) \times v(x) = p \bmod f_n(x)$.■

When $n=2048$, $k=16$ and $\eta>298$, $\sqrt{\gamma_k}(4/3)^{(3k-1)/4}\beta_k^{n/(2k)-1}<2^{\eta-12}$. Now, if $\|w(x)\times C(x)\|_\infty < p/2$, then one can correctly recover the bit in a ciphertext.

Example 3.1 Let $n=4$, $u(x)=159+8x+4x^2+2x^3=[159\ 8\ 4\ 2]$, $p=641407153$, $v(x)=4027071-204800x-91520x^2-40898x^3$. $u(x)$ and $f_4(x)=1+x^4$ are factored as follows:

$$[159\ 8\ 4\ 2]=2*[[[26912186\ 1]\ 1]\ [[522671888\ 1]\ 1]\ [[91823081\ 1]\ 1]]\bmod 641407153 \quad (3\text{-}1)$$

$$[1\ 0\ 0\ 0\ 1]=[[[26912186\ 1]\ 1]\ [[258567259\ 1]\ 1]\ [[382839894\ 1]\ 1]\ [[614494967\ 1]\ 1]]\bmod 641407153 \quad (3\text{-}2)$$

So, $\alpha=p-26912186=614494967$ and the public key is $pk=(p,\alpha)$. According to pk, one computes $g(x)=[382839894\ 343459750\ 614494967\ 1]$. To obtain $w(x)$, one constructs lattice M and calls the LLL algorithm for M. In fact, one finds the exact solution $v(x)$ for this small example. Without loss of generality, assume $w(x)=\delta(x)\square v(x)=[1\ -1\ 1\ 4]\square v(x)=[4896893\ 3824893\ 4303943\ 15954106]$. To be simplicity, we compute $\alpha^2 \bmod p=343459750$ and $\alpha^3 \bmod p=382839894$.

To find $[\delta(x)]_2$, one first computes a ciphertext

$$\begin{aligned} c=a(x)(\alpha)&=(2r(x)+m(x))(\alpha) \ \bmod(p)\\ &=(3*382839894+4*343459750+5*614494967+9) \ \bmod(p)\\ &=463576302 \end{aligned}$$

$$\begin{aligned} d&=\lfloor 463576302/p\times[4896893\ 3824893\ 4303943\ 15954106]+[0.5\ \ 0.5\ \ 0.5\ \ 0.5]\rfloor\\ &=[3539224\quad -2764437\quad 3110670\quad 11530812] \end{aligned}$$

Since $d \bmod 2=[\delta(x)]_2\times[a(x)]_2 \bmod 2$, $[\delta(x)]_2=d \bmod 2\times([a(x)]_2)^{-1}\bmod 2=[1\ 1\ 1\ 0]$.

Thus, one decrypts a ciphertext by using equivalent secret key $w(x)$, $[\delta(x)]_2$.■

4. Cryptanalysis of Gentry-Halevi's Scheme

4.1 Fully Homomorphic Encryption

Key Generation Algorithm (KeyGen).

(1) Choose a random polynomial $u(x)=\sum_{i=0}^{n-1}u_i x^i \in Z[x]$, where each entry u_i is a η-bit integer, and $p=\det(Rot(u(x)))$ is an odd integer.

(2) Apply the XGCD-algorithm over $Q[x]$ to obtain $v(x)=\sum_{i=0}^{n-1}v_i x^i \in Z[x]$ such that $u(x)\times v(x)=p \bmod f_n(x)$.

(3) Check that $u(x)$ is a good generating polynomial. Here $u(x)$ is good if the Hermite normal form of $J=Rot(u(x))$ has the following form.

$$HNF(J)=\begin{pmatrix} p & 0 & 0 & 0 & 0\\ -\alpha & 1 & 0 & 0 & 0\\ -\alpha^2 \bmod p & 0 & 1 & 0 & 0\\ -\alpha^3 \bmod p & 0 & 0 & 1 & 0\\ & & & \ddots & \\ -\alpha^{n-1} \bmod p & 0 & 0 & 0 & 1 \end{pmatrix}$$

(4) Output the public key $pk=(p,\alpha)$, and the secret key $sk=(p,v(x))$.

Encryption Algorithm (Enc). Given the public key pk and a bit $m\in\{0,1\}$, choose a random polynomial $r(x)$ with $\|r(x)\|_\infty=1$. Output a ciphertext $c=(2r(\alpha)+m)\bmod p$.

Decryption Algorithm (Dec). Given the secret key sk and a ciphertext c, choose an odd coefficient v_i from $v(x)$, decipher the bit $m = (c - \lfloor c \times v_i / p + 0.5 \rfloor) \bmod 2$.
All other details of FHE are omitted (see [6]).

In FHE [6], $u(x)$ is an arbitrary good generating polynomial and p can be a composite number. Moreover, the decryption algorithm only uses modulo operation over the integers.

4.2 Cryptanalysis of Gentry-Halevi's FHE

By the decryption algorithm in [6], a ciphertext vector is $\bar{c} = (c,0,...,0)$. Hence, $[\bar{c} \times Rot(v)]_p = [c \cdot (v_0, v_1, ..., v_{n-1})]_p = ([cv_0]_p, [cv_1]_p, ..., [cv_{n-1}]_p)$. On the other hand, we have $[\bar{c} \times Rot(v) / p] = [\bar{a} \times Rot(v) / p] = \bar{a} \times Rot(v) / p$, where $[\cdot]$ is fractional part, and $\bar{a} = 2\bar{r} + b \cdot \bar{e}_1$ with small vectors $\bar{r}$ and $\bar{e}_1 = (1,0,...,0)$. So, $[\bar{c} \times Rot(v)]_p = \bar{a} \times Rot(v) = 2\bar{r} \times Rot(v) + b \cdot \bar{v}$. That is, $([cv_0]_p, [cv_1]_p, ..., [cv_{n-1}]_p) = b \cdot \bar{v} \bmod 2$ for any decryptable ciphertext c.

We apply the same method in Section 3, which finds a small multiple $w(x) = \delta(x) \times v(x)$ of the secret key $v(x)$. When all the entries of $\bar{a} \times Rot(w(x))$ are less than $p/2$, we may recover the message bit in a ciphertext c as follows: $b = 1$ if $([cw_0]_p, [cw_1]_p, ..., [cw_{n-1}]_p) = \bar{w} \bmod 2$, otherwise $b = 0$. Thus, we finds $w(x) = \delta(x) \times v(x)$ over $\mathbb{Z}[x]$ with $\|\delta(x)\|_\infty \le \sqrt{\gamma_k} (4/3)^{(3k-1)/4} \beta_k^{n/2k-1}$ by Theorem 3.1.

When $n = 2048$, $k = 16$ and $\eta = 380$, we can recover the message bit in a ciphertext by the above method. Furthermore, we may also recover the message bit in a ciphertext for $n = 8196$, $\eta = 380$ by Theorem 2.3.

Example 4.1 Let $n = 4$, $u(x) = 127 + 11x + 121x^2 + 12x^3 = [127\ 11\ 121\ 12]$, $p = 17*55827209$, $v(x) = [3944101 - 388356 \quad 3694147 \quad 317]$. We evaluate $\alpha = 836836133$, $\alpha^2 \bmod p = 31797930$, $\alpha^3 \bmod p = 692833054$. It is not difficult to verify that the above attack works. ■

5. Recovering Plaintext from Ciphertext

Given the public key $pk = (p, \alpha)$ of FHE's [3, 6] and plaintext bit $m \in \{0,1\}$, the encryption algorithm outputs a ciphertext $c = (2r(\alpha) + m) \bmod p$. A new lattice is constructed as follows.

$$B = \begin{pmatrix} c & 1 & 0 & 0 & \cdots & 0 \\ (2\alpha) \bmod p & 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ (2\alpha^{n-1}) \bmod p & 0 & 0 & 0 & 1 & 0 \\ p & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Calling LLL obtains a reduced basis C of B such that $U \times B = C$. By Theorem 2.3, the first vector C_1 of C is satisfied $\|C_1\|_2 \le (1.02)^n \times n^{1/2} \times 2^\mu$, where $\mu = \|r(x)\|_\infty$. It is easy to see that $\|U_1\|_\infty \le (1.02)^n \times n^{1/2} \times 2^\mu$. When $n \le 8192$, $(1.02)^n \times n^{1/2} \times 2^\mu << 2^{380}$. and U_{11} with probability 1/2 is odd, we get the plaintext $m = C_{11} \bmod 2$. In fact, other rows U_i of U also is feasible, if $\|U_i\|_\infty \le (1.02)^n \times n^{1/2} \times 2^\mu$ and $|C_{i1}| < (1.02)^n \times n^{1/2} \times 2^\mu$. When U_{11} is not odd, we can generate another lattice B replacing c with $k \cdot c$, recall LLL to get a reduced basis C of B, and check whether or not U_{11} is odd, where k is a small odd.

In order to implement FHE, the ciphertexts of the secret key bit in [6] merely use $\mu = 1$. So, one can use the above method to decipher these ciphertexts.

6. Conclusion and Open Problems

We analyze the security of the schemes in [3, 6]. We mainly show that their schemes are not secure for $n \leq 8192$ in [3, 6] by applying lattice reduction algorithm. However, we notice that the block lattice reduction algorithm needs too much time and space for large dimension and large integer. We plan to further improve the above lattice attack method to really break the FHE challenge in [6].

One more open problem is to construct a new FHE by hiding a principal ideal lattice to avoid the above lattice reduction attack.

References

[1] R. Rivest, L. Adleman and M. Dertouzos, "On data banks and privacy homomorphisms", In Foundations of Secure Computation, pp. 169-180, **(1978)**.
[2] C. Gentry, "Fully homomorphic encryption using ideal lattices", STOC 2009, pp. 169-178, **(2009)**.
[3] N. P. Smart and F. Vercauteren, "Fully homomorphic encryption with relatively small key and ciphertext sizes", PKC 2010, LNCS 6056, pp. 420-443, **(2010)**.
[4] M. van Dijk, C. Gentry, S. Halevi and V. Vaikuntanathan, "Fully homomorphic encryption over the integers", Eurocrypt 2010, LNCS 6110, pp. 24-43, **(2010)**.
[5] D. Stehle and R. Steinfeld, "Faster Fully Homomorphic Encryption", Asiacrypt 2010, LNCS 6477, pp. 377-394, **(2010)**.
[6] C. Gentry and S. Halevi, "Implementing Gentry's fully-homomorphic encryption scheme", EUROCRYPT 2011, LNCS 6632, pp. 129-148, **(2011)**.
[7] N. Gama, N. Howgrave-Graham, H. Koy and P. Q. Nguyen, "Rankin's constant and blockwise lattice reduction", CRYPTO 2006, pp. 112–130, **(2006)**.
[8] P. Q. Nguyen and D. Stehle, "LLL on the average", ANTS VII, 2006, LNCS 4076, pp. 238-256, **(2006)**.
[9] H. W. Lenstra Jr., A. K. Lenstra and L. Lov´asz, "Factoring polynomials with rational coefficients", Mathematische Annalen 261, pp. 515-534, **(1982)**.
[10] C. P. Schnorr, "A hierarchy of polynomial time lattice basis reduction algorithms", Theoretical Computer Science, 53, pp. 201-224, **(1987)**.
[11] Miklos Ajtai, R. Kumar and D. Sivakumar, "A sieve algorithm for the shortest lattice vector problem", STOC 2001, pp. 601-610, **(2001)**.

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.

Gu Ji-xing , received the B. S. degree from Tongji University in 1998 , and the M. S. degree from Shanghai Jiaotong University in 2001 , respectively. He majored in Communication & Information System and achieved the Ph.D. degree from Shanghai Jiaotong University in 2007. He is currently working on software R&D for some leading products. His main areas of research are media security, streaming scheduling, image communication , microwave IC's and circuit design.