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

Gu Chunsheng
School of Computer Engineering
Jiangsu Teachers University of Technology
Changzhou, China, 213001
guchunsheng@gmail.com

**Abstract:** For the fully homomorphic encryption schemes in [SV10, GH11], this paper presents attacks to solve equivalent secret key and directly recover plaintext from ciphertext for lattice dimensions $n$=2048 by using lattice reduction algorithm. According to the average-case behavior of LLL in [NS06], their schemes are also not secure for $n$=8192.
**Keywords:** Fully Homomorphic Encryption, Cryptanalysis, Principal Ideal Lattice, Lattice Reduction

## 1. Introduction

Homomorphic encryption has many applications in cryptography. Rivest, Adleman and Dertouzos [RAD78] first presented this concept. But until 2009, Gentry [Gen09] constructed the first fully homomorphic encryptions based on ideal lattice, all previous schemes are insecure. After the scheme of [Gen09], Smart and Vercauteren presented an optimization FHE scheme with smaller ciphertext and key [SV10] by using principal ideal lattice. Dijk, Gentry, Halevi, and Vaikuntanathan [vDGHV10] proposed a simple fully homomorphic encryption scheme over the integers, whose security depends on the hardness of finding an approximate integer GCD. Stehle and Steinfeld [SS10] improved Gentry's fully homomorphic scheme and obtained to a faster fully homomorphic scheme. Gentry and Halevi [GH11] implemented Gentry's scheme also by applying principal ideal lattice. Currently, the security of the FHEs in [SV10, GH11] depends on the hardness assumption of finding small principal ideal lattice, given its HNF form or two elements form. In this paper, we solve an equivalent secret key for $n$=2048, 8192 of the FHEs in [SV10, GH11] by using lattice reduction algorithm. Moreover, we also present an attack to directly recover plaintext from ciphertext by using the public key.

### 1.1 Our Cryptanalysis

In our cryptanalysis, we use for the concrete parameters of the FHE [SV10, SV10]. We show that their schemes are not secure for lattice dimension *n*=2048 by using block lattice reduction algorithm [GHKN06]. According to the average case behavior of LLL [NS06], the ratio $\|b_1\|/\lambda(L)$ is about $(1.02)^n$, namely, $\|b_1\| \le (1.02)^n \lambda_1(L) << 2^{380}\lambda_1(L)$, where 380 is the bit-size of the coefficients in the generator polynomial of [GH11]. Thus, their schemes are also 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 [NS06].

### 1.2 Organization

The remaining of this paper is organized as follows: In Section 2, we give some notations and definitions, and the lattice reduction algorithms, and then in Section 3, 4, we respectively analyze the security of the Smart-Vercauteren's scheme and the Gentry-Halevi's scheme. In Section 5, we present an attack by directly recovering plaintext from ciphertext in [SV10, GH11]. Finally, we conclude this paper and give several open problems.

## 2. Preliminaries

### 2.1 Notations

Let $n$ be a security parameter, $[n]=\{0,1,...,n\}$. Let $R$ be the ring of integer polynomials modulo $f_n(x)$, i.e., $R=\mathbb{Z}[x]/f(x)$, where $f_n(x)$ is an integer monic and irreducible polynomial of degree $n$. 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 $u$, $\vec{u}=[u_0,...,u_{n-1}]$ 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 the ring $R$.

### 2.2 Lattices

A lattice in $\mathbb{R}^m$ is the set of all integral combination of $n$ linearly independent vectors $b_1,...,b_n$ in $\mathbb{R}^m$ ($m \ge n$), namely $L=L(b_1,...,b_n)=\{\sum_{i=1}^n x_i b_i, x_i \in Z\}$, usual denoted as a matrix $B$. Any such $n$-tuple of vectors $b_1,...,b_n$ is called a basis of the lattice $L$. Every lattice has an infinite number of lattice bases. Two lattice bases $B_1, B_2 \in \mathbb{R}^{m\times n}$ are equivalent if and only if $B_1 = B_2 U$ for some unimodular matrix $U \in \mathbb{Z}^{n\times n}$. The volume of a lattice $L$ is the determinant of any basis of $L$, namely $vol(L) = \det(L) = \sqrt{B^T B}$. For every full-rank lattice $L$, there is a unique Hermite normal form (HNF) basis which given any basis of $L$ can be efficiently computed by using Gaussian elimination. The HNF usually uses as the public key of the lattice-based public key cryptography.

### 2.3 Ideal Lattices

In this paper, we take $f_n(x) = x^n + 1$ with $n$ a power of $2$. Let $I$ be a principal ideal of $R$, namely, it only has a single generator. For the coefficient vector $\vec{u} = (u_0, u_1, ..., u_{n-1})^T$ of $u \in R$, we define the cyclic rotation $rot(\vec{u}) = (-u_{n-1}, u_0, ..., u_{n-2})^T$, and the corresponding circulant matrix $Rot(u) = (\vec{u}, rot(\vec{u}), ..., rot^{n-1}(\vec{u}))^T$. $Rot(u)$ is called the rotation basis of the ideal lattice $(u)$. For $\forall f, u \in R$, $[f]_u$ is the coefficient vector of $f$ modulo the rotation basis of $u$, namely, $\vec{f} \bmod Rot(u)$. So, we consider each element of $R$ as being both a polynomial and a vector.

We focus on principal ideals of $R_p$ in this paper since the scheme in [SV10, GH11] only used the principal ideals.

### 2.4 Lattice Reduction Algorithm

Given a basis of the 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 reduction finds a vector whose approximating factor is at most $2^{(n-1)/2}$. In 1987, Schnorr [Sch87] 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 [AKS01], HKZ(2$k$) is about $O(2^{2k})$. In the following, we will choose $k=16$ to guarantee computation to be feasible.

**Theorem 2.1 (Sch87 Theorem 2.6)** 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 $\beta_k$ is another lattice constant using in Schnorr's analysis of his algorithm.

Shnorr [Sch87] 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 [GHKN06] 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 (GHKN06 Theorem 2, 3)** For all $k \ge 2$, Schnorr's constant $\beta_k$ 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 $k \le 100$.

**Observation 2.3 (NS06).** For a lattice $L$, The first vector $b_1$ output by LLL algorithm is satisfied to the ratio $\|b_1\|/\lambda(L) \approx (1.02)^n$ on the average.

# 3. Cryptanalysis of Smart-Vercauteren's Scheme

## 3.1 Fully Homomorphic Encryption (FHE)

For completeness, we here give the somewhat homomorphic encryption (SHE) and the fully homomorphic encryption (FHE) in [SV10].

**Key Generation Algorithm (SHE-KeyGen).**

(1) Choose a random polynomial $u(x)=\sum_{i=0}^{n-1}u_i x^i \in Z[x]$, such that $\|u(x)\|_\infty$ is a $\eta$-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 a bit $m\in\{0,1\}$, choose a small random polynomial $r(x)$ with $\|r(x)\|_\infty$ is a $\mu$-bit integer. Output the ciphertext $c=(2r(\alpha)+m) \bmod p$.

**Add Operation (Add).** Given the public key $pk$, and two ciphertexts $c_1,c_2$, evaluate the ciphertext $c=(c_1+c_2) \bmod p$.

**Multiplication Operation (Mul).** Given the public key $pk$ and two ciphertexts $c_1,c_2$, evaluate a new ciphertext $c=(c_1\times c_2) \bmod p$.

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

**Key Generation Algorithm (FHE-KeyGen).**

(1) Choose $s_1$ uniformly random integers $\beta_i$ in $[-p,p]$ such that there is a subset $S$ of $s_2$ elements with $\sum_{i\in S}\beta_i=\beta$.

(2) Define $sk_i=1$ if $i\in S$ and $sk_i=0$ otherwise.

(3) Encrypt the bits $sk_i$ under the SHE to get $\vec{\beta}_i=Enc(sk_i,pk)$.

(4) Output the public key $pk=(p,\alpha,s_1,s_2,\{\vec{\beta}_i,\beta_i\}_{i=1}^{s_1})$, the secret key $sk=(p,\beta)$.

To implement FHE, Smart and Vercauteren constructed Recrypt algorithm by introducing the sparse subset sum problem. Here we omit this algorithm.

### 3.2 Cryptanalysis of Smart-Vercauteren's FHE

In this subsection, we merely give an algorithm recovering message bit and postpone to Section 5 recovering the private key.

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

The security of the scheme above depends on the hardness of solving the following small principal ideal problem.

**Definition 3.1 (Small Principal Ideal Problem (SPIP)).** Given a principal ideal $\pi$ in either two element or HNF representation, compute a small generator of the ideal.

On the surface, we need to get the private key $v(x)$ to attack the scheme. In fact, if we can get a small multiple $w(x) = \delta(x) \times v(x)$ of the secret key $v(x)$, where $\delta(x)$ is a small integer polynomial, then we can directly decrypt a ciphertext. Since $C(x) = c + q(x) \times \gamma$ according to [SV10], we have $\delta(x) \times (C(x) - c) = \lfloor c \bullet w(x) + 0.5h \rfloor \times \gamma = q'(x) \times \gamma$, where $q'(x) = \delta(x) \times q(x)$, namely, $[\delta(x) \times (C(x) - c)]_2 = q'(x)$ via $[\gamma]_2 = 1$. Thus, we may select a small polynomial $C(x)$, evaluate its corresponding ciphertext $c = C(\alpha) \bmod p$, and then solve $[\delta(x)]_2$ by the above equation. Once one knows $w(x)$ and $[\delta(x)]_2$, one can decipher arbitrary ciphertext with small error term. Now, we only need to give an algorithm which generates a suitable polynomial $w(x)$.

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

**Proof.** Since $\alpha$ is a root of $f_n(x) = x^n + 1$ over modulo $p$, so we can factor $x^n + 1 = (x - \alpha) \bullet g(x) \bmod p$. It is easy to verify $g(x) = t(x) \bullet v(x)$ over modulo $p$. Without loss of generality, assume $g(x) = x^{n-1} + g_{n-2}x^{n-2} + \cdots + g_0$. We need to row reduce the following matrix

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

We call the lattice reduction algorithm in [Sch87, GHKN06] to get $w(x) = \delta(x) \times v(x)$ 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 here $w(x) \in R$ since $u(x) \times v(x) = p \bmod f_n(x)$.■

Thus, if $\|w(x) \times C(x)\|_\infty < p/2$, we can correctly recover the message bit in a ciphertext. According to Theorem 2.2, when $n = 2048$, $k = 16$, we can recover the message bit in an encrypted ciphertext if $\sqrt{\gamma_k}(4/3)^{(3k-1)/4}\beta_k^{n/2k-1} < 2^{\eta-12}$, namely $\eta > 298$.

**Example 3.1** Let $n = 4$, $u(x) = 159 + 8x + 4x^2 + 2x^3 = [159\ 8\ \ 4\ \ 2]$, $p = \det(Rot(u(x))) = 641407153$, $v(x) = 4027071 - 204800x - 91520x^2 - 40898x^3$.

We factor $u(x)$ and $f_4(x) = 1 + x^4$ over modulo $p$ as follows:

$$\begin{aligned} &[159\ 8\ 4\ 2] \\ &= 2*[[[26912186\ 1]\ 1]\ [[522671888\ 1]\ 1]\ [[91823081\ 1]\ 1]]\bmod 641407153 \end{aligned} \tag{3-1}$$

$$\begin{aligned} &[1\ 0\ 0\ 0\ 1] \\ &= [[[26912186\ 1]\ 1]\ [[258567259\ 1]\ 1]\ [[382839894\ 1]\ 1]\ [[614494967\ 1]\ 1]]\bmod 641407153 \end{aligned} \tag{3-2}$$

So, we evaluate $\alpha = p - 26912186 = 614494967$, and output the public key $pk = (p, \alpha)$. By $pk$, we can evaluate $g(x) = [382839894\ 343459750\ 614494967\ 1]$. Now, we construct the corresponding matrix $M$ and call the LLL algorithm for it to obtain $w(x)$. In fact, we get the exact solution $v(x)$ for this example. Without loss of generality, assume $w(x) = \delta(x) \bullet v(x) = [1\ \ -1\ \ 1\ \ 4] \bullet v(x) = [4896893\ 3824893\ 4303943\ 15954106]$. To be simplicity, we compute $\alpha^2 \bmod p = 343459750$ and $\alpha^3 \bmod p = 382839894$.

To solve $[\delta(x)]_2$, we first compute a ciphertext

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

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

Thus, according to $d \bmod 2 = [\delta(x)]_2 \times [a(x)]_2 \bmod 2$, we have

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

Now, we can decrypt a ciphertext by using $w(x)$ and $[\delta(x)]_2$.□

## 4. Cryptanalysis of Gentry-Halevi's Scheme

In this section, we first present the SHE and the FHE in [GH11], and then mainly analyze the security of FHE for their practical parameters.

### 4.1 Fully Homomorphic Encryption (FHE)

**Key Generation Algorithm (SHE-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 $\eta$-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 $sk=(p,v(x))$.

In fact, the SHE of Gentry and Halevi's scheme are similar as that in [SV10], except that $u(x)$ is an arbitrary good generating polynomial and $p$ does not need to be a prime. Moreover, here the decryption algorithm only uses modulo operation over integers.

Similar to that of [SV10], Gentry and Halevi [GH11] also introduced the sparse subset sum problem to squash the depth of decryption circuit. Here we omit the concrete details.

### 4.2 Cryptanalysis of Gentry-Halevi's FHE

For the decryption algorithm in [GH11], recall that the ciphertext vector $\vec{c}=(c,0,\ldots,0)$. Hence, $\left[\vec{c}\times Rot(v)\right]_p=\left[c\bullet(v_0,v_1,\ldots,v_{n-1})\right]_p=(\left[cv_0\right]_p,\left[cv_1\right]_p,\ldots,\left[cv_{n-1}\right]_p)$. On the other hand, according to [GH11], we have $\left[\vec{c}\times Rot(v)/p\right]=\left[\vec{a}\times Rot(v)/p\right]=\vec{a}\times Rot(v)/p$, where $\left[\bullet\right]$ is fractional part, and $\vec{a}=2\vec{r}+b\bullet\vec{e}_1$ with small vector $\vec{r}$ and $\vec{e}_1=(1,0,\ldots,0)$. So, $\left[\vec{c}\times Rot(v)\right]_p=\vec{a}\times Rot(v)=2\vec{r}\times Rot(v)+b\bullet\vec{v}$. Thus, for any decryptable ciphertext $c$, we have an equation $(\left[cv_0\right]_p,\left[cv_1\right]_p,\ldots,\left[cv_{n-1}\right]_p)=b\bullet\vec{v} \bmod 2$.

Therefore, we can use the same method as Section 3 above which evaluates a small multiple $w(x)$ of the secret key $v(x)$ such that $w(x)=\delta(x)\times v(x)$. When all the entries in $\vec{a}\times Rot(w(x))$ are less than $p/2$, we may recover the message bit in a ciphertext $c$ as follows: if $(\left[cw_0\right]_p,\left[cw_1\right]_p,\ldots,\left[cw_{n-1}\right]_p)=\vec{w} \bmod 2$ then $b=1$, otherwise $b=0$. Thus, we also merely need to present an efficient algorithm which 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 applying same method in Theorem 3.1.

So, we can recover the message bit in an encrypted ciphertext by using $k=16$ for the parameters $n=2048$, $\eta=380$ in [GH11]. Furthermore, if we use the sampling reduction algorithm in [Sch03] under their same assumption, we can further attack the scheme for the parameters $n = 8196$, $\eta = 380$ in [GH11].

**Example 4.1** Let $n = 4$, $u(x) = 127 + 11x + 121x^2 + 12x^3 = [127\ \ 11\ \ 121\ \ 12]$, $p = 949062553 = 17*55827209$, $v(x) = [3944101\ \ -388356\ \ -3694147\ \ 317303]$. We evaluate $\alpha = 836836133$, $\alpha^2 \bmod p = 317979309$, $\alpha^3 \bmod p = 692833054$. It is not difficult to verify that the above attack method works. □

## 5. Recovering Plaintext from Ciphertext

According to the FHE of [SV10, GH11], given the public key $pk = (p, \alpha)$ and plaintext $m \in \{0,1\}$, the encryption algorithm outputs ciphertext $c = (2r(\alpha) + m) \bmod p$. Thus, we construct a new lattice as follows.

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

We now call LLL algorithm to obtain a reduced basis $C, U$ such that $U \times B = C$. According to Observation 2.3, we know $\|C_1\|_2 \le (1.02)^n \times n^{1/2} \times 2^{\mu}$, where $\mu = \|r(x)\|_\infty$. It is not difficult to verify that $\|U_1\|_\infty \le (1.02)^n \times n^{1/2} \times 2^{\mu}$. Thus, $(1.02)^n \times n^{1/2} \times 2^{\mu} << 2^{380}$ for $n \le 8192$ 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 $|C_{i1}| < (1.02)^n \times n^{1/2} \times 2^{\mu}$.

We notice that the ciphertext of the secret key bit in [GH11] only uses $\mu = 1$ to implement FHE. So, one can use the above method to decipher these ciphertexts.

## 6. Conclusion and Open Problems

We have analyzed the security of the schemes in [SV10, GH11]. In fact, we mainly show that their schemes are not secure for $n \le 8192$ in [SV10, GH11] by applying lattice reduction algorithm. Concretely speaking, we have proved that their schemes are not secure for $n = 2048$ from the theoretical view. Moreover, according to average performance of LLL, we can attack the FHE for $n = 8192$. However, we notice that the block lattice reduction algorithm needs too much time and space for large dimension and large integer. Thus, in the following, we will improve the above lattice attack method to really break the FHE challenge in [GH11] from computational view.

Another open problem is to construct a new FHE by hiding a principal ideal lattice to resist the lattice reduction attack.

## Reference


[AKS01] Miklos Ajtai, Ravi Kumar, and D. Sivakumar. A sieve algorithm for the shortest lattice vector problem. In STOC, pages 601{610. ACM, 2001.

[Coh93] H. Cohen, A Course in Computational Algebraic Number Theory, Graduate Texts in Mathematics, 138. Springer, 1993.

[vDGHV10] M. van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan. Fully homomorphic encryption over the integers. In Proc. of Eurocrypt, volume 6110 of LNCS, pages 24-43. Springer, 2010.

[Gen01] C. Gentry. Key Recovery and Message Attacks on NTRU-Composite. Eurocrypt'01, LNCS 2045, pages 182-194.

[Gen09] C. Gentry. Fully homomorphic encryption using ideal lattices. In Proc. of STOC, pages 169-178, 2009.

[GH11] Craig Gentry and Shai Halevi. Implementing Gentry's fully-homomorphic encryption scheme. In Kenneth Paterson, editor, Advances in Cryptology — EUROCRYPT 2011, volume 6632 of Lecture Notes in Computer Science, pages 129–148, Berlin, Heidelberg, New York, 2011. Springer Verlag. Cryptology ePrint Archive: Report 2010/520: http://eprint.iacr.org/2010/520.

[GHKN06] N. Gama, N. Howgrave-Graham, H. Koy, and P. Q. Nguyen. Rankin's constant and blockwise lattice reduction. In CRYPTO, pages 112–130, 2006.

[GN08a] N. Gama and P. Nguyen, Finding Short Lattice Vectors within Mordell's Inequality, In Proc. of the ACM Symposium on Theory of Computing STOC'08, pp. 208–216, 2008.

[GN08b] N. Gama and P.Q. Nguyen, Predicting lattice reduction, in Proc. EUROCRYPT 2008, LNCS 4965, Springer-Verlag, pp. 31–51, 2008.

[GS02] C. Gentry, M. Szydlo. Cryptanalysis of the Revised NTRU Signature Scheme. Eurocrypt’02, LNCS 2332, pages 299-320.

[HPS98] J. Hoffstein, J. Pipher, J. H. Silverman. NTRU: A Ring-Based Public Key Cryptosystem. LNCS 1423, pages 267-288, 1998.

[LLL82] H.W. Lenstra Jr., A.K. Lenstra and L. Lov´asz, Factoring polynomials with rational coefficients, Mathematische Annalen 261, pp. 515–534, 1982.

[Mic07] D. Micciancio Generalized compact knapsaks, cyclic lattices, and efficient one-way functions. Computational Complexity, 16(4):365-411.

[NS00] P. Nguyen and J. Stern, Lattice Reduction in Cryptology: An Update, in Proc. of Algorithm Number Theory (ANTS IV), LNCS 1838, pages 85-112. Springer-Verlag, 2000.

[NS06] P.Q. Nguyen and D. Stehle, LLL on the average, proc. Of ANTS VII, 2006, LNCS 4076, pp. 238-256.

[RAD78] R. Rivest, L. Adleman, and M. Dertouzos. On data banks and privacy homomorphisms. In Foundations of Secure Computation, pages 169-180, 1978.

[Sch87] C.P. Schnorr, A hierarchy of polynomial time lattice basis reduction algorithms. Theoretical Computer Science, 53, pp. 201–224, 1987.

[SS10] D. Stehle and R. Steinfeld. Faster Fully Homomorphic Encryption. Cryptology ePrint Archive: Report 2010/299: http://eprint.iacr.org/2010/299.

[SV10] Nigel P. Smart and Frederik Vercauteren. Fully homomorphic encryption with relatively small key and ciphertext sizes. In Public Key Cryptography – PKC 2010, volume 6056 of Lecture Notes in Computer Science, pages 420–443, Berlin, Heidelberg, New York, 2010. Springer Verlag.