이학석사 학위논문

Implementation of Ring-based Homomorphic Encryption

(환-기반 준동형 암호의 구현)

2014년 2월

서울대학교 대학원

수리과학부

전 민 영

Implementation of Ring-based Homomorphic Encryption

(환-기반 준동형 암호의 구현)

지도교수 천 정 희

이 논문을 이학석사 학위논문으로 제출함

2013년 12월

서울대학교 대학원

수리과학부

전 민 영

전 민 영의 이학석사 학위논문을 인준함

2013년 12월

위 원 장 _____ (인)

부 위 원 장 _____ (인)

위 원 _____ (인)

Implementation of Ring-based Homomorphic Encryption

A dissertation
submitted in partial fulfillment
of the requirements for the degree of
Master of Science
to the faculty of the Graduate School of
Seoul National University

by

Min Young Jun

Dissertation Director : Professor Jung Hee Cheon

Department of Mathematical Sciences
Seoul National University

February 2014

Abstract

Fully homomorphic encryption(FHE) was introduced by Rivest, Adleman, and Dertouzos that can homomorphically evaluate encrypted data. However, it turned out to be insecure. After then, it has been presented lots of improved, modified FHE schemes for security and efficiency. One of them, a FHE of L*ó*pez-Alt, Tromer, and Vaikuntanathan based on NTRU encryption scheme of Hoffstein, Pipher, and Silverman, is more efficient than the other schemes for a long time. Bos et al. suggest the more efficient FHE scheme, which is the fastest, than L*ó*pez-Alt et al. They remove the DSPR assumption and modulus switching procedure. Thus, in this paper, we look into the scheme and implement its practical version and optimization. We provide experimental results and analyze them as regards efficiency compared with original one whose parameters are presented by authors.

Key words: BLLN, NTRU, FHE, Implementation
Student Number: No. 2011-23211

Contents

Chapter 1

Introduction

Homomorphic encryption is a cryptosystem that one could add or multiply ciphertexts and another could decrypt the result. Partially homomorphic cryptosystem can evaluate ciphertexts using only one operation without decryption of them. e.g. RSA, ElGamal, Paillier. On the other hand, fully homomorphic encryption(FHE) introduced by Rivest, Adleman, and Dertouzos [12], which is not secure under the plaintext attack, can evaluate both addition and multiplication. Until 2009, the cryptosystem of Boneh, Goh and Nissim [6] is the best homomorphic encryption that evaluates a number of additions and one multiplication.

In 2009, Craig Gentry suggested the first fully homomorphic encryption using ideal lattices, which can evaluate a unlimited number of additions and multiplications of ciphertexts that consist of polynomials over a ring [4, 5]. It is more secure than former one. Since the noise of ciphertext increase during evaluations, he present how to modify the scheme so as to reduce the noise called bootstrapping process. Then the scheme can be converted into a fully homomorphic encryption. However, it is not efficient to apply the real world. After then, it has been presented that lots of improved, modified and another fully homomorphic encryption schemes which do not use ideal lattices.

FHE based on NTRU encryption of Hoffstein, Pipher, and Silverman [7] was presented by L*ó*pez-Alt, Tromer, and Vaikuntanathan [9], which is more

efficient and secure than the others. Bos et al. [1] suggest the more efficient FHE scheme, which is the fastest of existing schemes, than L*ó*pez-Alt et al. Their scheme removes the decisional small polynomial ratio assumption, it avoids modulus switching that makes the noise size diminish and its ciphertext is represented by a single ring element. They also provide a practical version with parameters and certain implementation results. In this paper, we investigate the BLLN scheme, implement it and analyze what makes the scheme faster and how faster than the other parameters of optimization procedure.

In chapter 2, we provide a basic notation and some of the assumption composed of the RLWE and DSPR with LTV as aspects of security. In chapter 3, we introduce the BLLN scheme including how to be a leveled somewhat homomorphic encryption and fully homomorphic encryption. It might be seem to have two part according to the way of homomorphic multiplication. In chapter 4, we set up parameters by proving security, implement it and carry out an optimization by defining and experimenting the $YASHE.Discard$. Furthermore, we present their results as tables and analyze them with optimization.

Chapter 2

Preliminaries

Fully homomorphic encryption can evaluate the addition and multiplication homomorphically between ciphertexts as $Enc(m_1 + m_2) = Enc(m_1) + Enc(m_2)$ and $Enc(m_1 \cdot m_2) = Enc(m_1) \cdot Enc(m_2)$ where $Enc(m_i)$ is the encryption of a message m_i. According to some properties, it could be divided into somewhat homomorphic encryption, leveled fully homomorphic encryption, and fully homomorphic encryption. Somewhat homomorphic encryption is a special form of fully homomorphic encryption whose noise increases during the evaluating process where each ciphertext has low-degree homomorphic operations. Leveled fully homomorphic encryption can evaluate the ciphertexts with high-degree L, which sometimes called depth, and fully homomorphic encryption can evaluate them regardless of depth or degree.

2.1 Basic Notation

We will use a quotient ring $R = \mathbb{Z}[x]/(\Phi_n(x))$, where $\Phi_n(x) = x^n + 1$ is a n-th cyclotomic polynomial with n a power of two. Denote a ℓ_∞ norm of f by $\|f\|_\infty = max_i\{|a_i| : f = \sum_{i=0}^{n-1} a_i x^i \in R\}$.

Lemma 2.1.1 ([5, 10]). *Let $n \in \mathbb{N}$, let $\phi(x)$ be a n-th cyclotomic polynomial of degree n and let $R = \mathbb{Z}[x]/(\phi(x))$. For any t, $s \in R$,*

$$\begin{aligned} \|s \cdot t(\mathrm{mod}\phi(x))\| &\leq \sqrt{n} \cdot \|s\| \cdot \|t\| \\ \|s \cdot t(\mathrm{mod}\phi(x))\|_\infty &\leq \sqrt{n} \cdot \|s\|_\infty \cdot \|t\|_\infty. \end{aligned}$$

We say that $\delta = sup\{\|f \cdot g\|_\infty/(\|f\|_\infty\|g\|_\infty) : f, g \in R\}$ is an expansion factor of R which is equal to n by lemma 2.1.1. For a polynomial $f \in R$, denote the $[f]_q \in R/qR = \mathbb{Z}_q/(x^n+1)$ by reducing coefficients into $\mathbb{Z}_q$ which have the range $(-\frac{q}{2}, \frac{q}{2}]$ for some integer q . For $\mathbf{v}, \mathbf{w} \in R^n$, the dot product is defined by $\langle \mathbf{v}, \mathbf{w} \rangle = \sum_{i=1}^{n} u_i \cdot w_i \in R$ where u_i and w_i are the ith component of that. It also holds for the quotient ring R_q.

A discrete Gaussian distribution $D_{\mathbb{Z}^n,\sigma}$ over $\mathbb{Z}^n$ with parameter σ is a probability distribution that assigns a probability proportional to $\exp(-\pi\|\mathbf{x}\|^2/\sigma^2)$ to each $\mathbf{x} \in \mathbb{Z}^n$. It is a product distribution of n independent copies of $D_{\mathbb{Z},\sigma}$ with mean 0 and standard deviation σ over the integers [8, 11]. It outputs a $(r\sqrt{n})$-bounded polynomial with high probability. The distribution $\chi = D_{\mathbb{Z}^n,\sigma}$ supported over R is called B-bounded if we have a B-bounded polynomial for all f sampled from χ, i.e. $\|f\|_\infty < B$ [9].

In this paper, we use a truncated Gaussian distribution χ that is B-bounded and statistically close to the discrete Gaussian. It takes sample whose norm is less than B so as to restrict the noise bound and growth during the homomorphic operations [9, 11].

2.2 The Ring-LWE Problem

It has been suggested that lots of FHE schemes based on the RLWE assumption and reductions of the worst case to average case. Lyubaskevsky, Peikert and Regev describe the Ring learning with error problem (Ring-LWE) [11]. It distinguishes which distribution is used for sampling and generates certain samples. It is an extension of the learning with error problem called LWE that based on a field rather than polynomial.

Definition 2.2.1 (Ring-LWE). For given security parameter λ, $n = n(\lambda) \in \mathbb{Z}$, let $\phi(x) = \phi_n(x) \in \mathbb{Z}[x]$ be a polynomial of degree n, let $q = q(\lambda) \geq 2$

be an integer, let $R = \mathbb{Z}/(\phi(x))$ and R/qR. Let $\chi = \chi(\lambda)$ be a distribution over R. The $Ring - LWE_{n,q,\chi}$ problem is to distinguish the following two distributions. In the first distribution one samples (a_i, b_i) uniformly from R_q^2. In second distribution one samples $(a_i, b_i = a_i \cdot s + e_i) \in R_q^2$ where s fixed for all samples is uniformly sampled from R_q, a_i's are uniformly from R_q and e_i's are from an error distribution χ. The Ring-LWE assumption is that the Ring-LWE problem is hard.

The shortest vector problem over the ideal lattice which is worst-case problem can be reduced to the Ring-LWE problem which is average-case.

Theorem 2.2.2 (**Worst-case to Average-case Reduction** [9, 11])**.** *Let $\Phi_n(x) = x^n + 1$ be a n-th cyclotomic polynomial of degree which is a power of two. Let $R = \mathbb{Z}/(x^n + 1)$, $q \equiv 1(\mathrm{mod} 2n)$ be a prime integer and χ be a $B = \omega(\sqrt{nlogn})$-bounded distribution. Then there is a randomized reduction from $n^{\omega(1)} \cdot (q/B)$-approximate worst-case SVP for ideal lattice over R to $Ring - LWE_{n,q,\chi}$.*

This theorem is useful for setting parameters.

2.3 The DSPR Problem with LTV

A. López-Alt, E. Tromer, and V. Vaikuntanathan [9] present a scheme which is based on NTRU with modifications and multikey homomorphism. It is provably secure based on standard problems in ideal lattices. The parameters, which is n, q, $\phi(x)$ of the degree n and χ, depend on security parameter λ. Its message space is $\{0, 1\}$ and all operations are evaluated on the quotient ring $R_q = \mathbb{Z}_q/(\phi(x))$. We simply introduce LTV scheme as follows.

Key Generation: Sample f_0, g from key distribution χ, then compute $f = [2f_0 + 1]_q$ such that $f \equiv 1 \,(\mathrm{mod}\, 2)$. If f is not invertible modulo q, re-sample f_0. Let $h = [2gf^{-1}]_q$ for inverse of f modulo q in R, then set the secret key f and public key h.

Encryption: Take a message m from $\{0,1\}$. Ciphertext is generated by computing $c = [m + 2e + hs]_q$ over the ring R with s, e sampled from error distribution χ.

Decryption: Compute $m' = [[fc]_q]_2 \in R$ with the secret key f.

It easy to check that as long as there is no wrap-around modulo q. Compute

$$[fc]_q = [fm + 2fe + fhs]_q = [fm + 2fe + 2gs]_q.$$

If there is no wrap-around modulo q then

$$[fc]_q(\text{mod}2) = [fm + 2fe + 2gs]_q = [fm]_q(\text{mod}2) = m.$$

This scheme is secure under the assumption of decisional small polynomial ratio problem.We simply call it DSPR problem.

Definition 2.3.1 (**DSPR Problem** [1, 9])**.** For given security λ, let $\Phi(x) \in \mathbb{Z}[x]$ be a polynomial of degree $n \in \mathbb{Z}$, let $q \in \mathbb{Z}$ be a prime integer. Let $R = \mathbb{Z}/(\phi(x))$, $R_q = R/qR$ and let χ is a distribution over the ring R. all of them depend on λ. The $DSPR_{n,q,\chi}$ problem is to distinguish between the following two distributions. In first distribution, one samples h uniformly at random over R_q. In second distribution, one samples $h = a/b$ where a and b are from the distribution χ. The $DSPR_{n,q,\chi}$ assumption is that the $DSPR_{n,q,\chi}$ problem is hard.

It holds only if $\sigma > \sqrt{q} \cdot \text{poly}(n)$ for cyclotomic polynomial $\phi(x)$ which makes secure against unbounded adversaries.

Stehl*é* and Steinfeld [13] propose a theorem for modified NTRU that changes the value 2 which could indicates the size of the message space to the value t where the t is in the $R_q^\times$, and $2e$ to te.

Theorem 2.3.2 ([1, 13])**.** *Let $\Phi_n(x) = x^n+1$ be a n-th cyclotomic polynomial of degree which is a power of two ≥ 4 and it splits into $k = k(q)$ irreducible factors modulo a prime $q \geq 5$. Let $R = \mathbb{Z}[x]/(x^n + 1)$ and $R_q = R/qR$. Let $U(R_q^\times)$ be the uniform distribution on $R_q^\times$ which is the set of invertible*

elements in R_q. Let $\chi = D^{\times}_{\mathbb{Z}^n,\sigma}$ be the spherical discrete Gaussian distribution on R_q, restricted to $R_q^{\times}$. Let $\epsilon \in (0, 1/3)$, $t \in R_q^{\times}$. Let a and b are depended on t with some distributions. Then the statistical distance $a/b \pmod q$ and $U(R_q^{\times})$ is bounded by

$$D = \begin{cases} 2^{20n} \cdot q^{-\frac{\lfloor \epsilon k \rfloor}{k}} \cdot 2n & \text{if } \sigma \geq 2n \cdot \sqrt{\log(8nq)} \cdot q^{\frac{1}{2}+\epsilon} \\ 2^{20n} \cdot q^{-2\epsilon n} & \text{if } \sigma \geq \sqrt{2n \log(8nq)} \cdot q^{\frac{1+k\epsilon}{2}} \text{ and } q \geq (2n)^{\frac{k}{1-2k\epsilon}}. \end{cases}$$

Then the DSPR problem is hard in R_q.

Chapter 3

Ring-based FHE Scheme

In this section we investigate the main scheme suggested by Bos et al. [1]. In section 3.2, a leveled somewhat homomorphic encryption of the scheme is presented and subsection 3.1.1 describes the basic encryption, decryption system and how the homomorphic operations are evaluated. In subsection 3.1.2, its security is proven with some assumptions. In section 3.2, we provide how the scheme can be converted to a fully homomorphic encryption.

3.1 Leveled Somewhat Homomorphic Encryption

Bos et al. present a couple of leveled somewhat homomorphic encryption according to homomorphic multiplication with two different evaluation keys. This keys affect the complexity of the scheme such that we could estimate which method of homomorphic multiplication is fast. We present two parts of the scheme in parallel. They call them "$YASHE$" and "$YASHE'$". A sub notation "1" stands for $YASHE$ and "2" stands for $YASHE'$.

3.1.1 BLLN Scheme

In this subsection, we describe two parts of the scheme. The first part is a basic cryptosystem including how the encrypted messages can be decryptable

and the second part explains how the homomophic operation can be computed according to the noise bound by some calculations. The basic cryptosystem is composed of the key generation, encryption and decryption as follows.

Key Generation: Sample f_0, g from key distribution χ_{key}, then compute $f = [tf_0 + 1]_q$ for $1 < t < q$. If f is not invertible modulo q, re-sample f_0. Let $h = [tgf^{-1}]_q$ for inverse of f modulo q in R, then we can generate a basic key set $\{f, h\}$ which is comprised of a secret key and public key respectively. Actually, the secret keys are f and f^{-1}.
The evaluation key set $\{evk_1, evk_2\}$ that is used for homomorphic multiplication is computed by

$$\begin{aligned} evk_1 &= [f^{-1}P_{\omega,q}\left(D_{\omega,q}\left(f\right) \otimes D_{\omega,q}\left(f\right)\right) + \mathbf{e} + h \cdot \mathbf{s}]_q \in R^{\ell^3_{\omega,q}}, \\ evk_2 &= [P_{\omega,q}\left(f\right) + \mathbf{e} + h \cdot \mathbf{s}]_q \in R^{\ell_{\omega,q}} \end{aligned}$$

where $\ell_{\omega,q} = \lfloor \log_w q \rfloor + 2$,

$$\begin{aligned} D_{\omega,q} : R \to R^{\ell_{\omega,q}}, \qquad f \mapsto ([f_0]_\omega, [f_1]_\omega, \cdots, [f_{\ell_{\omega,q}-1}]_\omega) = ([f_i]_\omega)_{i=0}^{\ell_{\omega,q}-1} \\ P_{\omega,q} : R \to R^{\ell_{\omega,q}}, \qquad f \mapsto ([f]_q, [f\omega]_q, \cdots, [f\omega^{\ell_{\omega,q}-1}]_q) = ([f\omega^i]_q)_{i=0}^{\ell_{\omega,q}-1} \end{aligned}$$

for $f = \sum_{i=0}^{\ell_{\omega,q}-1} f_i\omega^i$ with $f_i \in R$.

If $\omega = 2$, then these functions are called BitDecomp and PowerOFTwo [2].

As the function $D_{\omega,q}$ and $P_{\omega,q}$ are defined, the dot product between two vectors $D_{\omega,q}(f)$ and $P_{\omega,q}(g)$ is equal to the scalar product of f and g modulo q for some $f, g \in R$. This follows from the fact that

$$< D_{\omega,q}(f), P_{\omega,q}(g) >= \sum_{i=0}^{\ell_{\omega,q}-1} [f_i]_\omega [g\omega^i]_q \equiv g \sum_{i=0}^{\ell_{\omega,q}-1} [f_i]_\omega \omega^i \equiv f \cdot g \pmod{q}.$$

Encryption: For given integer t, we take a message from the space R/tR whose element is of the form $m + tR$. Simply, it can be written by $[m]_t$. Ciphertext that is the encryption of the message is computed by $\left[\left\lfloor \frac{q}{t} \right\rfloor [m]_t + e + hs\right]_q \in R$ for s, e sampled from error distribution χ_{err}.

Decryption: Compute $m = \left[\left\lfloor \frac{t}{q}[fc]_q \right\rceil\right]_t \in R$ to obtain the message with the secret key f.

The scheme is correctly decryptable when there exist $v \in R$ such that $\|v\|_\infty < \frac{(\triangle - q \pmod t)}{2}$ for $[fc]_q = [\triangle[m]_t + v]_q$, where $\triangle = \left\lfloor \frac{q}{t} \right\rfloor$. For some $a \in R$,

$$\begin{aligned} \frac{t}{q}[fc]_q &= \frac{t}{q} \triangle [m]_t + v \cdot \frac{t}{q} + ta \\ &= [m]_t - \frac{q \pmod t}{q}[m]_t + v \cdot \frac{t}{q} + ta. \end{aligned}$$

If $\|v\|_\infty < \frac{(\triangle - q \pmod t)}{2}$, then $\left\| - \frac{q \pmod t}{q}[m]_t + v \cdot \frac{t}{q} \right\|_\infty < \frac{1}{2}$. So, we could be correctly decrypt the ciphertext.

The homomorphic operations consisted of addition and multiplication are defined as follows.

Homomorphic Addition: Let $c_1, c_2 \in R$ be encrypted message $m_1, m_2 \in R/tR$. Compute an addition $[c_1 + c_2]_q$ that is a encrypted message $m_1 + m_2$ modulo t by adding coefficients of ciphertext componentwise.

Although the size of inherent noise increases during this operation, which has a sum of v_i and $r \in R$ where the r satisfies the equation as $[m_1]_t + [m_2]_t = [m_1 + m_2]_t + r$, $\|r\|_\infty \leq 1$ and the ciphertext c_i has a form of $[m_i]_t + v_i$, it could be decryptable.

Homomorphic Multiplication: Compute a ciphertext with multiplication of two ciphertexts. Passing by a key-switching procedure, we obtain a final ciphertext called an intermediate ciphertext, which might be correctly decryptable, is obtained.

According to the method of multiplication, two ways of homomorphic encryption are presented with different ciphertexts $\tilde{c}_{1,mul}$ and $\tilde{c}_{2,mul}$, intermediate ciphertexts. Through the key-switching procedure, the intermediate

ciphertext becomes a ciphertext decrypted under an original secret key f instead of f^2. The evaluation key evk_i is used for this procedure. If this step is omitted, the ciphertext could not be decryptable since its noise is too large to decrypt. Hence the key-switching process is essential and the intermediate ciphertext exists at all times.

The ciphertext $\tilde{c}_{1,mul}$ which is a multiplication of two ciphertexts c_1 and c_2 is computed by

$$\tilde{c}_{1,mul} = \left[\left\lfloor \frac{t}{q} P_{\omega,q}\left(c_1\right) \otimes P_{\omega,q}\left(c_2\right) \right\rceil\right]_q \in R^{\ell^2_{\omega,q}},$$

where the $\otimes$ is a tensor product which reduces the monomials $\ell^2_{\omega,q}$ to $\binom{\ell_{\omega,q}}{2}$. The way of the key-switching is obtained by

$$\begin{aligned}
\left[\left\langle D_{\omega,q}\left(\tilde{c}_{1,mul}\right), evk_1\right\rangle\right]_q &= \left[\left\langle D_{\omega,q}\left(\tilde{c}_{1,mul}\right), f^{-1} P_{\omega,q}\left(D_{\omega,q}\left(f\right) \otimes D_{\omega,q}\left(f\right)\right) + \mathbf{e} + h \cdot \mathbf{s}\right\rangle\right]_q \\
&= \left[f^{-1}\left\langle D_{\omega,q}\left(\tilde{c}_{1,mul}\right), P_{\omega,q}\left(D_{\omega,q}\left(f\right) \otimes D_{\omega,q}\left(f\right)\right)\right\rangle\right. \\
&\quad \left. + \left\langle D_{\omega,q}\left(\tilde{c}_{1,mul}\right), \mathbf{e}\right\rangle + h \cdot \left\langle D_{\omega,q}\left(\tilde{c}_{1,mul}\right), \mathbf{s}\right\rangle\right]_q.
\end{aligned}$$

In this case, the final ciphertext $c_{1,mul}$ composed of a vector of polynomials is

$$c_{1,mul} = \left[\left\langle D_{\omega,q}\left(\tilde{c}_{1,mul}\right), evk_1\right\rangle\right]_q$$

that could be correctly decryptable under the secret key f if

$$\begin{aligned}
\left[f c_{1,mul}\right]_q &= \left[\left\langle D_{\omega,q}\left(\tilde{c}_{1,mul}\right), P_{\omega,q}\left(D_{\omega,q}\left(f\right) \otimes D_{\omega,q}\left(f\right)\right)\right\rangle\right. \\
&\quad \left. + f\left\langle D_{\omega,q}\left(\tilde{c}_{1,mul}\right), \mathbf{e}\right\rangle + gt\left\langle D_{\omega,q}\left(\tilde{c}_{1,mul}\right), \mathbf{s}\right\rangle\right]_q \\
&= \left[(t/q) \cdot \left\langle P_{\omega,q}\left(c_1\right) \otimes P_{\omega,q}\left(c_2\right), D_{\omega,q}\left(f\right) \otimes D_{\omega,q}\left(f\right)\right\rangle\right. \\
&\quad \left. - r_c + f\left\langle D_{\omega,q}\left(\tilde{c}_{1,mul}\right), \mathbf{e}\right\rangle + gt\left\langle D_{\omega,q}\left(\tilde{c}_{1,mul}\right), \mathbf{s}\right\rangle\right]_q \\
&= \left[\triangle[m_1 m_2]_t + \tilde{v}_{1,mul} + f\left\langle D_{\omega,q}\left(\tilde{c}_{1,mul}\right), \mathbf{e}\right\rangle + gt\left\langle D_{\omega,q}\left(\tilde{c}_{1,mul}\right), \mathbf{s}\right\rangle\right]_q \\
&= \left[\triangle[m_1 m_2]_t + v_{1,mul}\right]_q
\end{aligned}$$

where

$$\begin{aligned}
r_c &= \frac{t}{q}\langle P_{\omega,q}(c_1) \otimes P_{\omega,q}(c_2), D_{\omega,q}(f) \otimes D_{\omega,q}(f)\rangle - \langle \tilde{c}_{1,mul}, D_{\omega,q}(f) \otimes D_{\omega,q}(f)\rangle \\
&= \left\langle \left(\frac{t}{q} P_{\omega,q}\left(c_1\right) \otimes P_{\omega,q}\left(c_2\right)\right) - \left\lfloor \frac{t}{q} P_{\omega,q}\left(c_1\right) \otimes P_{\omega,q}\left(c_2\right) \right\rceil, D_{\omega,q}\left(f\right) \otimes D_{\omega,q}\left(f\right) \right\rangle,
\end{aligned}$$

and $v_{1,mul}$ satisfies under the condition of decryption.

Another ciphertext of the multiplication $\tilde{c}_{2,mul}$ which is a multiplication of the ciphertexts c_1 and c_2 is computed by

$$\tilde{c}_{2,mul} = \left[\left\lfloor \frac{t}{q} c_1 c_2 \right\rceil\right]_q \in R,$$

and the key-switching process is followed by

$$\begin{aligned}
[\langle D_{\omega,q}(\tilde{c}_{2,mul}), evk_2\rangle]_q &= [\langle D_{\omega,q}(\tilde{c}_{2,mul}), P_{\omega,q}(f) + \mathbf{e} + h\cdot\mathbf{s}\rangle]_q \\
&= [\langle D_{\omega,q}(\tilde{c}_{2,mul}), P_{\omega,q}(f)\rangle + \langle D_{\omega,q}(\tilde{c}_{2,mul}), \mathbf{e}\rangle \\
&\quad + h\cdot\langle D_{\omega,q}(\tilde{c}_{2,mul}), \mathbf{s}\rangle]_q.
\end{aligned}$$

The final ciphertext $c_{2,mul}$ composed of a simply single polynomial is

$$c_{2,mul} = [\langle D_{\omega,q}(\tilde{c}_{2,mul}), evk_2\rangle]_q$$

that could be correctly decryptable under the secret key f if

$$\begin{aligned}
[fc_{2,mul}]_q &= [f\langle D_{\omega,q}(\tilde{c}_{2,mul}), P_{\omega,q}(f)\rangle + f\langle D_{\omega,q}(\tilde{c}_{2,mul}), \mathbf{e}\rangle + gt\langle D_{\omega,q}(\tilde{c}_{2,mul}), \mathbf{s}\rangle]_q \\
&= [f\langle D_{\omega,q}((t/q)c_1c_2), P_{\omega,q}(f)\rangle - f\cdot r_c + f\langle D_{\omega,q}(\tilde{c}_{2,mul}), \mathbf{e}\rangle \\
&\quad + gt\langle D_{\omega,q}(\tilde{c}_{2,mul}), \mathbf{s}\rangle]_q \\
&= [\triangle[m_1m_2]_t + \tilde{v}_{2,mul} + f\langle D_{\omega,q}(\tilde{c}_{2,mul}), \mathbf{e}\rangle + gt\langle D_{\omega,q}(\tilde{c}_{2,mul}), \mathbf{s}\rangle]_q \\
&= [\triangle[m_1m_2]_t + v_{2,mul}]_q
\end{aligned}$$

where

$$\begin{aligned}
f\cdot r_c &= f\cdot\left\langle \left(D_{\omega,q}\left(\frac{t}{q}c_1c_2\right)\right) - \left\lfloor D_{\omega,q}\left(\frac{t}{q}c_1c_2\right)\right\rceil, P_{\omega,q}(f)\right\rangle \\
&= \frac{t}{q}f^2\cdot c_1c_2 - f^2\cdot\left\lfloor\frac{t}{q}c_1c_2\right\rceil
\end{aligned}$$

and $v_{2,mul}$ satisfies under the condition of decryption.

3.1.2 Security of the Scheme

The security of the scheme $YASHE$ using the first way of multiplication is based on the IND-CPA under a "circular security" assumption and the RLWE assumption. The second scheme $YASHE'$ is based on the DSPR problem under the circular security and RLWE assumptions. For the circular security, replace f by distinct secret key f_j of $evk_{i,j}$ for $i \in \{1, 2\}$ whose number "1" and "2" stand for the $YASHE$ and $YASHE'$ respectively and j which is a level with $1 \leq j \leq L$, i.e. for given

$$\begin{aligned} evk_{1,j} &= [f_j^{-1} P_{\omega,q}\left(D_{\omega,q}\left(f_{j-1}\right) \otimes D_{\omega,q}\left(f_{j-1}\right)\right) + \mathbf{e} + h_j \cdot \mathbf{s}]_q \in R^{\ell^3_{\omega,q}}, \\ evk_{2,j} &= [P_{\omega,q}\left(f_{j-1}^2\right) + \mathbf{e} + h_j \cdot \mathbf{s}]_q \in R^{\ell_{\omega,q}}, \end{aligned}$$

the final ciphertext $c_{i,mul}$, which is the output of key-switching step with input ciphertext $\tilde{c}_{i,mul}$, is correctly decryptable under the secret key f_j where $\tilde{c}_{i,mul}$ is a multiplication of two $j-1$th level ciphertexts.

3.2 Fully Homomorphic Encryption

To obtain a fully homomorphic encryption from the somewhat homomorphic encryption, we have to decrease the noise of homomorphic evaluation. It it is too large, then the homomorphic properties has gone. A modulus switching or bootstrapping is a general way to reduce the noise of homomorphic computations. The noise growth during the homomorphic addition could be neglected compared with homomorphic multiplication. Therefore we just consider the multiplication. For given $fc_i = \triangle[m_i]_t + v_i (\mathrm{mod}\, q)$ with the ciphertext c_i, the noise v_i has the bound V as

$$\begin{aligned} \|v_i\|_\infty &< 2\delta t B_{key} B_{err} + \frac{1}{2} q \,(\mathrm{mod}\, t)\, \delta t B_{key} \\ &< 2\delta t B_{err} + \frac{1}{2}\delta t^2 = \delta t\left(2B_{err} + \frac{1}{2}t\right) = V, \end{aligned}$$

and the inherent noise $v_{i,mult}$ of homomorphic multiplication is

$$\begin{aligned}
\|v_{1,mult}\|_\infty &< \delta t(2+\delta\ell_{\omega,tB_{key}}\omega)V + \frac{\delta t^2}{2}(3+\delta\ell_{\omega,tB_{key}}) \\
&\quad + \frac{1}{8}(\delta\ell_{\omega,tB_{key}}\omega)^2 + \frac{1}{2} + \delta^2 t\ell^3_{\omega,q}\omega B_{err}B_{key} \\
&= \delta t(2+\delta\ell_{\omega,t}\omega)V + \frac{\delta t^2}{2}(3+\delta\ell_{\omega,t}) + \frac{1}{8}(\delta\ell_{\omega,t}\omega)^2 + \frac{1}{2} + \delta^2 t\ell^3_{\omega,q}\omega B_{err}, \\
\|v_{2,mult}\|_\infty &< \delta t(4+\delta tB_{key})V + \delta^2 t^2 B_{key}(B_{key}+t) + \delta^2 t\ell_{\omega,q}\omega B_{err}B_{key} \\
&= \delta t(4+\delta t)V + \delta^2 t^2(1+t) + \delta^2 t\ell_{\omega,q}\omega B_{err}.
\end{aligned}$$

Hence the noise increase from $O(\delta t^2)$ of $\|v_i\|_\infty$ to $O(\delta^3 t^4)$ of $\|v_{i,mul}\|_\infty$.

To make a fully homomorphic encryption of $YASHE$ with an arbitrary level L_1, set up parameters with the hypothesis of the theorem by Stehlé and Steinfeld and RLWE assumption, i.e., for $\epsilon \in (0,1)$ and $k \in (1/2,1)$, let $q = 2^{d^\epsilon}$ be a prime and let $\Phi_n(x) = x^n + 1$ be a cyclotomic polynomial of degree n which splits into k irreducible factors modulo q. Let χ_{key} be a discrete Gaussian distribution on R_q with deviation $\sigma_{key} \geq 2n\sqrt{\log(8nq)}\cdot q^k$, let χ_{err} be an asymptotically $\omega(\sqrt{2n\log(2n)})$-bounded Gaussian distribution on R. The inherent noise of a ciphertext regarding the depth L_1 circuit, organized in a leveled homomorphic multiplicative tree structure without any additions, is bounded by $C_1^{L_1}V_j + L_1C_1^{L_1-1}C_2$ where

$$\begin{aligned}
C_1 &= \delta t(2+\delta\ell_{\omega,tB_{key}}\omega) = O(\mathrm{poly}(n)\log(q)) \text{ since } \delta = n, \\
C_2 &= \frac{\delta t^2}{2}(3+\delta\ell_{\omega,tB_{key}}\omega) + \frac{1}{8}(\delta\ell_{\omega,tB_{key}}\omega)^2 + \frac{1}{2} + \delta^2 t\ell^3_{\omega,q}\omega B_{err}B_{key} \\
&\quad = O(\mathrm{poly}(n)\log(q)^3 q^k) \text{ and } V = O(\mathrm{poly}(n)\cdot q^k) \text{ for some } k \in (1/2,1)
\end{aligned}$$

by iterating the bound of $\|v_{1,mult}\|$ L_1 times as assumed that the inherent noise terms of all ciphertexts are considered to have the roughly same size $V_j = C_1V + L_1C_2$ for each level $j > 0$. To guarantee correctness of the scheme, following equality should be satisfied as

$$q = \Omega(L_1 \cdot \mathrm{poly}(n)^{L_1+1} \cdot \log(q)^{L_1+2} \cdot q^k)$$

and any circuit of depth can be estimated by

$$L_1 = O\left(\frac{(1-k)\log(q)}{\log(\log(q)) + \log(n)}\right).$$

When the level L_1 is greater than a depth $L_{dec} = O(\log(\log(q)) + \log(n))$ over $\mathbb{F}_2$ of the decryption circuit, it could be converted into a fully homomorphic encryption scheme from Gentry's Bootstrapping Theorem [5].

To obtain a fully homomorphic encryption of $YASHE'$ with a level L_2, select parameters in order to satisfy the hypothesis of the RLWE and DSPR assumptions, i.e. let $q \equiv 1 \,(\mathrm{mod} 2n)$ be a prime and let $\Phi_n(x) = x^n + 1$ be a cyclotomic polynomial of degree n which is a power of 2. Let χ_{key} be a discrete Gaussian distribution over the ring R, let χ_{err} be an asymptotically $\omega(\sqrt{2n\log(2n)})$-bounded Gaussian distribution on R. By evaluating the inherent noise bound of a ciphertext during the homomorphic operations of the depth L_2, an overall noise bound can be deduced by iterating the ℓ_∞ norm of $v_{2,mult}$. It has the bound by

$$C_1'^{L_2} \cdot V + L_2 C_1'^{L_2-1} C_2' \tag{3.2.1}$$

where

$$C_1' = \left(1 + \frac{4}{\delta t B_{key}}\right)\delta^2 t^2 B_{key} = O(\mathrm{poly}(n)) \text{ since } \delta = n, \tag{3.2.2}$$

$$\begin{aligned} C_2' &= \delta^2 t B_{key}(t(B_{key} + t) + \ell_{\omega,q}\omega B_{err}) \\ &= O(\mathrm{poly}(n)\log(q) \cdot q^k) \text{ for some } k \in (1/2, 1). \end{aligned} \tag{3.2.3}$$

To guarantee correctness of this scheme, we have that

$$q = \Omega(L_2 \cdot \mathrm{poly}(n)^{L_2+1} \cdot \log(q) \cdot q^k).$$

Therefore YASHE' can evaluate any circuit of depth

$$L_2 = O\left(\frac{(1-k)\log(q)}{\log(n)}\right).$$

When the level L_2 is greater than the depth L_{dec}, it can be converted into a fully homomorphic encryption scheme from Gentry's Bootstrapping Theorem [5].

Chapter 4

Implementation

In this chapter, the intermediate ciphertext $\tilde{c}_{2,mul}$ of $YASHE'$ is comprised of a single polynomial rather than a vector of polynomials of the intermediate ciphertext $\tilde{c}_{1,mul}$ used for $YASHE$. Since the evaluation key of $YASHE'$ consists of $\ell_{\omega,q}$ polynomials instead of $\ell^3_{\omega,q}$ of $YASHE$ which results in a simple key switching procedure, the $YASHE'$ is more practical than $YASHE$. Thus, we provide concrete parameters of $YASHE'$ in section 4.1 and an implementation result of it with special function $YASHE.Discard$ to optimize in section 4.2. Finally, we analyze the results including what makes the implementation fast and how to set parameters with optimization.

4.1 Parameter Selection

Since $YASHE'$ is secure under the Ring-LWE assumption and the DSPR assumption, we have parameters from these assumptions. The attack against RLWE, which can be seen a variant of the LWE, can be considered in the same manner of the LWE. Therefore the distinguishing attack [8] on LWE can also be applied to the RLWE problem. The distinguishing attack is to find the shortest nonzero vector in the dual lattice of $\lambda_q(A)$

$$\lambda_q^{\perp}(A) := \{y \in \mathbb{Z}^m \mid y \cdot A \equiv 0 \,(\mathrm{mod}\, q)\}$$

where A is derived from a sample of LWE as $(A, b = A \cdot \mathbf{s} + \mathbf{e})$, $A \leftarrow \mathbb{Z}_q^{m \times n}$, $\mathbf{s} \leftarrow \mathbb{Z}_{\mathbf{q}}^{\mathbf{n}}$, which is a secret, and $\mathbf{e} \leftarrow \chi_\sigma^{\mathbf{n}}$ where the χ_σ is a normal distribution

with mean 0 and standard deviation σ on $\mathbb{Z}$. If we find the shortest vector of it, one can distinguish the distribution of LWE samples from uniformly distribution. The advantage of the distinguishing attack is very close to $\epsilon = \exp(-\pi \cdot (\|y\| \cdot \mathbf{s}/q)^2)$. For security, it is sufficient to take a ℓ_∞ norm of the shortest vector y less than $\alpha \cdot q/\sigma$ where $\alpha = \sqrt{\log(1/\epsilon)/\pi}$. Actually, the size of the norm has the minimum of q and $\delta^m q^{n/m}$ where δ is called the root-Hermite factor by [8]. In case of $\delta^m q^{n/m}$, the optimal value of m is $\sqrt{n\log(q)/log(\delta)}$. By BKZ algorithm, we have a relation between the runtime in seconds and root-Hermite factor as $\lambda = 1.8/\log_2 \delta - 110$. Then, the relation

$$\alpha \cdot q/\sigma < 2^{2\sqrt{n \log_2(q) \log_2(\delta)}} \tag{4.1.1}$$

is obtained. Let q be a 127-bit prime and $n = 2^{12}$ which is a degree of polynomial $\phi_n(x)$ of quotient ring $R = \mathbb{Z}/(\Phi_n(x))$, all of which depends on security parameter λ. Fix the key distribution assumed to be bounded by $B_{key} = 1$ for evaluating the key switching step and the error distribution bounded by $B_{err} = 6\sigma_{err}$ where $\sigma_{err} = 8$ with Ring-LWE assumption. These parameters with $\omega = 2^{32}$ are presented and used for implementation of the scheme by Bos et al. [1]. The maximum of the depth L is 2 through the computation of the noise bound with L_2.

4.2 Implementation of the Scheme

We implemented the scheme $YASHE'$ in C++ with NTL library while Bos et al. [1] implemented directly in C which does not depend on any other number theory library. For given parameters of the section 4.1, we obtain the running times of the scheme on average values over 100 tests and implement it on an Intel Core i7, 3.4 GHz, 16GB RAM. The result of implementation at level 1 is that key generation runs in 152 ms, encryption runs in 21 ms, addition of ciphertexts in 44 μs, multiplication of ciphertexts including the key-switching in 29 ms, and decryption runs in 7 ms.

4.2.1 Optimization

Before optimization, we define the $YASHE.Discard$ function of Bos et al. [1], which has a input ciphertext c and output ciphertext c' as

$$c' = \mathbf{YASHE.Discard}_{\omega}(\mathbf{c}, \mathbf{i}) = \lfloor \omega^{-\mathbf{i}}\mathbf{c} \rfloor.$$

It is a truncating function by removing an insignificant multiple of ω-words of the ciphertext c. If ω^i-words are thrown away, $\omega^i c$ is equal to c except least i-th bit which is zero. If $fc = \triangle m + v (\mathrm{mod}\, q)$, then $\omega^i c' f = \triangle m + v' (\mathrm{mod}\, q)$ with $\|v'\|_\infty \leq \|v\|_\infty + \frac{1}{2}\delta\omega^i \|f\|_\infty$. Therefore, both cihpertext length and the number of components of the evaluation key are reduced per multiplication of the key-switching procedure. Since the noise increase with size $\frac{1}{2}\delta\omega^i\|f\|_\infty$, the inherent noise $\|v'\|_\infty$ of level L is bounded by $C_1'^L \cdot V' + LC_1'^{L-1}C_2'$ where C_1', C_2', and V are the same as (3.2.2), (3.2.3) and

$$\begin{aligned} \|v'\|_\infty &\leq \|v\|_\infty + \frac{1}{2}\delta\omega^i\|f\|_\infty \\ &\leq V + \frac{1}{2}\delta\omega^i\|f\|_\infty \leq V + \frac{1}{2}\delta\omega^i(1+t) = V'. \end{aligned} \tag{4.2.1}$$

If $C_1'^L \cdot V' + LC_1'^{L-1}C_2'$ is bounded by $\frac{\triangle}{2}$ then it could be correctly decryptable.

To sum up all conditions of the noise bound, we get approximately inequalities that represent relations of n, ω, q, t and ℓ by

$$\begin{aligned} n^2t^3\frac{n}{2}\omega^i &\leq \frac{q}{2t} \Rightarrow n^3t^4\omega^i \leq q \text{ if } i \geq 1 \\ n^2t^2(\ell_{\omega,q} \cdot \omega \cdot B_{err}) &\leq \frac{q}{2t} \Rightarrow 2n^2t^3(\ell_{\omega,q} \cdot \omega \cdot B_{err}) \leq q \\ \frac{\log(q) - 3}{2\sqrt{\log(q)\log(\delta)}} &\leq \sqrt{n}, \;\; \lfloor \log_\omega(q) \rfloor + 2 = \ell_{\omega,q}. \end{aligned}$$

These are convenience for evaluating the noise fast regarding parameters. The first and second equality is derived from (4.2.1), (3.2.2) and (3.2.3), and the third equality is induced from (4.1.1).

For given parameters from section 4.1, we present data of the scheme implemented by comparing the result which is already presented before starting the section 4.2 with using the $YASHE.Discard$ function. In given parameters, we percieve that the value i has the maximum 1 which means that ω-words can be discarded and one evaluation key diminish. Look into the table 4.1 that shows the running time of multiplication by millisecond.

	KeyGen	**Encrypt**	**Add**	**Mult**	**Decrypt**
i=0	143	21	0.044	29	7
i=1	136	20	0.044	28	6

Table 4.1: Running times of $\log \omega = 32$, $t = 1024$ for unit [ms]

We can see that the multiplication time decreases about 3 percent and key generation time decreases about 4 percent compared with $i = 0$ that dose not use the discarding function. Particularly, the addition time is the shortest than the others about 1000 times. Unless q and n change, the timings of encryption, decryption and addition do not fluctuate.

According the parameter ω, the number of elements of the evaluation key varies such that the running time changes. We derived the fact that the number of components is a reciprocal proportion to ω as the way of key generation. Therefore, we implement it by changing $\log \omega = 32$ to $\log \omega = 48$ to make more efficient. The variation of ω cause the components of evaluation key to drop to $\ell_{2^{48},q} = 4$ from $\ell_{2^{32},q} = 5$. However, we can not use ω more than 2^{64} to reduce them because the inherent noise of multiplication increase too large to decrypt. The result of implementation with $\log \omega = 48$ using $YASHE.Discard$ function is shown by Table 4.2.

	KeyGen	**Encrypt**	**Add**	**Mult**	**Decrypt**
i=0	117	19	0.044	24	6
i=1	116	20	0.044	23	5

Table 4.2: Running times of $\log \omega = 48$, $t = 1024$ for unit [ms]

In this case, the effect of the function is very slight as the running time

of key generation decreases about 0.85 percent and that of multiplication decreases about 4 percent. But there is some effects on ω as the one decreases about $14 \sim 18$ percent and the other decreases about 17 percent from table 4.1 and table 4.2.

Furthermore, we implement it by changing the parameter t which decides message space as the ring R_{256} instead of R_{1024}. Table 4.3 shows the experimental result of that parameter. It uses $\log \omega = 48$ rather than $\log \omega = 32$ because we already know the former makes this scheme faster than the latter.

	KeyGen	**Encrypt**	**Add**	**mul**	**Decrypt**
i=0	121	20	0.044	25	6
i=1	113	20	0.044	23	6

Table 4.3: Running times of $\log \omega = 48$, $t = 256$ for unit [ms]

The variation of t does not have any effects on running time from table 4.2 and table 4.3. But the time of key generation decreases about 6 percent and that of multiplication decreases about 8 percent by using this function.

In the similar manner, we use $\log \omega = 64$ to reduce procedure of key-switching, which can only use $\ell_{\omega,q} = 3$ components of evaluation key without discarding words. If not, we should use $\ell_{\omega,q} = 4$ components. To make the scheme faster, we may consider $YASHE.Discard$ function. However, using this function makes it impossible to decrypt correctly since the inherent noise is too large to decrypt. Its experimental result is provided by table 4.4.

	KeyGen	**Encrypt**	**Add**	**Mul**	**Decrypt**
i=0	97	19	0.044	22	6

Table 4.4: Running times of $\log \omega = 64$, $t = 256$ for unit [ms]

The data of the table is about 19 percent of key generation and about 12 percent of multiplication faster than the result of the table 4.3 without

discarding words.

From some experimental results, we can deduced that the value ω, which affects the process of key-switching with the number of components of evaluation key, and the fact that whether $YASHE.Discard$ function is used or not are the important factors for deciding the running times. When the pair of n and q consist of very small size, the maximal depth (or level) L_{max} is very low. Since bootstrapping requires the depth L to be about 50, we additively implement the scheme under this circumstances, which should be set as the optimal parameter $q = 2048$ bit prime, $t = 2$, $n = 2^{16}$, $\omega = 293$, $i = 1$, and $\ell = 8$.

	KeyGen	**Encrypt**	**Mul**	**Decrypt**
i=1	8.5	0.3	296	0.09

Table 4.5: Running times of $L = 50$, $\log \omega = 293$, $t = 2$ for unit [s]

Chapter 5

Conclusion

For fixed parameter q that is a 127 bit prime and polynomial degree $n = 2^{12}$, we provide the experimental data by implementing the BLLN scheme. Then, we performed optimization by adjusting parameters and using particular function $YASHE.Discard$. Its results is presented by tables. When the parameter t, which determines message space, is fixed by 1024, the running times are reduced about 18 percent of key generation and about 20 percent of multiplication. When t is 256, the one decreases about 19 percent and the other decreases about 12 percent. As a result, we can see which parameters make the scheme faster and how to choose these parameters we want to implement it as well as we can estimate how faster it is, depending on some technique of optimization.

Bibliography

[1] J. Bos, K. Lauter, J. Loftus, and M. Naehrig. Improved security for a ring-based fully homomorphic encryption scheme. Cryptology ePrint Archive, Report 2013/075, 2013. http://eprint.iacr.org/2013/075/.

[2] Z. Brakerski. Fully homomorphic encryption without modulus switching from classical GapSVP. In Advances in Cryptology - Crypto 2012, volume 7417 of Lecture Notes in Computer Science, pages 868-886. Springer, 2012.

[3] Z. Brakerski, C. Gentry, and V. Vaikuntanathan. Fully homomorphic encryption without bootstrapping. In ITCS, pages 309-325, 2012.

[4] C. Gentry. A fully homomorphic encryption scheme, Stanford University PhD thesis, 2009, http://crypto.stanford.edu/craig/craig-thesis.pdf.

[5] C. Gentry. Fully homomorphic encryption using ideal lattices. In Michael Mitzenmacher, editor, STOC, pages 169-178. ACM, 2009.

[6] S. Goldwasser and S. Micali. Probabilistic encryption and how to play mental poker keeping secret all partial information, in Proceedings of the 14th ACM Symposium on Theory of Computing|STOC 1982, ACM (1982), 365-377.

[7] J. Hoffstein, J. Pipher, and J. H. Silverman. NTRU: A ring-based public key cryptosystem. In J. Buhler, editor, ANTS, volume 1423 of Lecture Notes in Computer Science, pages 267-288. Springer, 1998.

[8] R. Lindner and C. Peikert. Better key sizes (and attacks) for LWE-based encryption. In Proceedings of the 11th international conference on

Topics in cryptology:CT-RSA 2011, CT-RSA'11, pages 319-339, Berlin, Heidelberg, 2011. Springer Verlag.

[9] A. L*ó*pez-Alt, E. Tromer, and V. Vaikuntanathan. On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption. In STOC, pages 1219-1234, 2012.

[10] V. Lyubashevsky and D. Micciancio. Generalized compact knapsacks are collision resistant. In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, ICALP (2), volume 4052 of Lecture Notes in Computer Science, pages 144-155. Springer, 2006.

[11] V. Lyubashvesky, C. Peikert, and O. Regev. on ideal lattices and learning with errors over rings. In EUROCRYPT, volume6110 of Lecture notes in Computer Science, page 1-23, 2010.

[12] R. Rivest, L. Adleman, and M. Dertouzos. On Data Banks and Privacy Homomorphisms, in Foundations of Secure Computation, Academic Press, New York (1978), 169-180.

[13] D. Stehl*é* and R. Steinfeld. Making NTRU as secure as worst-case problems over ideal lattices. In Advances in Cryptology - EUROCRYPT 2011 - 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques, volume 6632 of Lecture Notes in Computer Science, page 27. Springer, 2011.

국문초록

Rivest 외 3명은 암호화된 상태에서 계산 가능한 완전동형암호를 제시하였다. 그러나 이 암호체계는 안전하지 않는 것으로 판명되어 이후 이를 변형하거나 개선된 수많은 동형암호들이 소개되었다. 그 중 López-Alt 외 3명이 제안한 완전동형암호는 NTRU 암호체계에 기반한 것으로 다른 암호체계보다 더 효율적이다. Bos 외 3명은 그들의 스킴보다 효율성면에서 개선된 암호화 알고리즘을 개발하였다. 이는 López-Alt 외 3명의 암호체계에 기반된 DSPR 문제와 Modulus switching 단계를 제거한 결과로 본 논문은 이들이 제안한 암호알고리즘을 살펴봄과 동시에 구현 및 최적화한 결과를 분석하여 제시한다.

주요어휘: BLLN,NTRU, 동형암호, 구현
학번: No. 2011-23211

감사의 글

학위 기간동안 부족한 제자를 보살펴주시고 아낌없는 조언과 가르침을 주신 천정희 선생님께 감사의 말씀드립니다. 선생님께 학문에 대한 열정과 삶의 지혜를 배웠습니다. 그리고 석사 논문 심사를 해주신 이기암 교수님, 오병권 교수님께 감사드립니다.

논문작성에 있어서 많은 조언과 도움을 주신 이문성 박사님께 감사드립니다. 논문 교정을 함께 해준 미란언니, 현숙언니, chocolat au lait를 사주며 격려를 아끼지 않았던 충훈 오빠, 심사준비에 도움을 준 형태오빠, 한솔언니 그리고 길지 않은 시간을 함께 보낸 연구실 선후배 동료들에게 감사합니다. 대학원에서 서로를 의지하며 보낸 대학원 동기들, 혜림이, 누리언니, 도영이, 지연이, 미림이, 경민오빠, 찬우오빠에게도 감사합니다.

무엇보다도 저에게 가장 큰 힘이 되어주신 부모님, 학위과정에서의 조언과 격려를 많이 해주신 친척분들과 제가 가장 아끼는 동생에게도 감사의 말 전합니다. 마지막으로 학부교수님들을 비롯하여 이 자리까지 올 수있게 해준 모든 분들께 감사를 표합니다.

이학석사 학위논문

Implementation of Ring-based Homomorphic Encryption

(환-기반 준동형 암호의 구현)

2014년 2월

서울대학교 대학원

수리과학부

전 민 영

Implementation of Ring-based Homomorphic Encryption

(환-기반 준동형 암호의 구현)

지도교수 천 정 희

이 논문을 이학석사 학위논문으로 제출함

2013년 12월

서울대학교 대학원

수리과학부

전 민 영

전 민 영의 이학석사 학위논문을 인준함

2013년 12월

위 원 장 _____ (인)

부 위 원 장 _____ (인)

위 원 _____ (인)

Implementation of Ring-based Homomorphic Encryption

A dissertation
submitted in partial fulfillment
of the requirements for the degree of
Master of Science
to the faculty of the Graduate School of
Seoul National University

by

Min Young Jun

Dissertation Director : Professor Jung Hee Cheon

Department of Mathematical Sciences
Seoul National University

February 2014

Abstract

Fully homomorphic encryption(FHE) was introduced by Rivest, Adleman, and Dertouzos that can homomorphically evaluate encrypted data. However, it turned out to be insecure. After then, it has been presented lots of improved, modified FHE schemes for security and efficiency. One of them, a FHE of L*ó*pez-Alt, Tromer, and Vaikuntanathan based on NTRU encryption scheme of Hoffstein, Pipher, and Silverman, is more efficient than the other schemes for a long time. Bos et al. suggest the more efficient FHE scheme, which is the fastest, than L*ó*pez-Alt et al. They remove the DSPR assumption and modulus switching procedure. Thus, in this paper, we look into the scheme and implement its practical version and optimization. We provide experimental results and analyze them as regards efficiency compared with original one whose parameters are presented by authors.

Key words: BLLN, NTRU, FHE, Implementation
Student Number: No. 2011-23211

Contents

Chapter 1

Introduction

Homomorphic encryption is a cryptosystem that one could add or multiply ciphertexts and another could decrypt the result. Partially homomorphic cryptosystem can evaluate ciphertexts using only one operation without decryption of them. e.g. RSA, ElGamal, Paillier. On the other hand, fully homomorphic encryption(FHE) introduced by Rivest, Adleman, and Dertouzos [12], which is not secure under the plaintext attack, can evaluate both addition and multiplication. Until 2009, the cryptosystem of Boneh, Goh and Nissim [6] is the best homomorphic encryption that evaluates a number of additions and one multiplication.

In 2009, Craig Gentry suggested the first fully homomorphic encryption using ideal lattices, which can evaluate a unlimited number of additions and multiplications of ciphertexts that consist of polynomials over a ring [4, 5]. It is more secure than former one. Since the noise of ciphertext increase during evaluations, he present how to modify the scheme so as to reduce the noise called bootstrapping process. Then the scheme can be converted into a fully homomorphic encryption. However, it is not efficient to apply the real world. After then, it has been presented that lots of improved, modified and another fully homomorphic encryption schemes which do not use ideal lattices.

FHE based on NTRU encryption of Hoffstein, Pipher, and Silverman [7] was presented by L*ó*pez-Alt, Tromer, and Vaikuntanathan [9], which is more

efficient and secure than the others. Bos et al. [1] suggest the more efficient FHE scheme, which is the fastest of existing schemes, than L*ó*pez-Alt et al. Their scheme removes the decisional small polynomial ratio assumption, it avoids modulus switching that makes the noise size diminish and its ciphertext is represented by a single ring element. They also provide a practical version with parameters and certain implementation results. In this paper, we investigate the BLLN scheme, implement it and analyze what makes the scheme faster and how faster than the other parameters of optimization procedure.

In chapter 2, we provide a basic notation and some of the assumption composed of the RLWE and DSPR with LTV as aspects of security. In chapter 3, we introduce the BLLN scheme including how to be a leveled somewhat homomorphic encryption and fully homomorphic encryption. It might be seem to have two part according to the way of homomorphic multiplication. In chapter 4, we set up parameters by proving security, implement it and carry out an optimization by defining and experimenting the $YASHE.Discard$. Furthermore, we present their results as tables and analyze them with optimization.

Chapter 2

Preliminaries

Fully homomorphic encryption can evaluate the addition and multiplication homomorphically between ciphertexts as $Enc(m_1 + m_2) = Enc(m_1) + Enc(m_2)$ and $Enc(m_1 \cdot m_2) = Enc(m_1) \cdot Enc(m_2)$ where $Enc(m_i)$ is the encryption of a message m_i. According to some properties, it could be divided into somewhat homomorphic encryption, leveled fully homomorphic encryption, and fully homomorphic encryption. Somewhat homomorphic encryption is a special form of fully homomorphic encryption whose noise increases during the evaluating process where each ciphertext has low-degree homomorphic operations. Leveled fully homomorphic encryption can evaluate the ciphertexts with high-degree L, which sometimes called depth, and fully homomorphic encryption can evaluate them regardless of depth or degree.

2.1 Basic Notation

We will use a quotient ring $R = \mathbb{Z}[x]/(\Phi_n(x))$, where $\Phi_n(x) = x^n + 1$ is a n-th cyclotomic polynomial with n a power of two. Denote a ℓ_∞ norm of f by $\|f\|_\infty = max_i\{|a_i| : f = \sum_{i=0}^{n-1} a_i x^i \in R\}$.

Lemma 2.1.1 ([5, 10]). *Let $n \in \mathbb{N}$, let $\phi(x)$ be a n-th cyclotomic polynomial of degree n and let $R = \mathbb{Z}[x]/(\phi(x))$. For any t, $s \in R$,*

$$\begin{aligned} \|s \cdot t(\mathrm{mod}\phi(x))\| &\leq \sqrt{n} \cdot \|s\| \cdot \|t\| \\ \|s \cdot t(\mathrm{mod}\phi(x))\|_\infty &\leq \sqrt{n} \cdot \|s\|_\infty \cdot \|t\|_\infty. \end{aligned}$$

We say that $\delta = sup\{\|f \cdot g\|_\infty/(\|f\|_\infty\|g\|_\infty) : f, g \in R\}$ is an expansion factor of R which is equal to n by lemma 2.1.1. For a polynomial $f \in R$, denote the $[f]_q \in R/qR = \mathbb{Z}_q/(x^n+1)$ by reducing coefficients into $\mathbb{Z}_q$ which have the range $(-\frac{q}{2}, \frac{q}{2}]$ for some integer q . For $\mathbf{v}, \mathbf{w} \in R^n$, the dot product is defined by $\langle \mathbf{v}, \mathbf{w} \rangle = \sum_{i=1}^{n} u_i \cdot w_i \in R$ where u_i and w_i are the ith component of that. It also holds for the quotient ring R_q.

A discrete Gaussian distribution $D_{\mathbb{Z}^n,\sigma}$ over $\mathbb{Z}^n$ with parameter σ is a probability distribution that assigns a probability proportional to $\exp(-\pi\|\mathbf{x}\|^2/\sigma^2)$ to each $\mathbf{x} \in \mathbb{Z}^n$. It is a product distribution of n independent copies of $D_{\mathbb{Z},\sigma}$ with mean 0 and standard deviation σ over the integers [8, 11]. It outputs a $(r\sqrt{n})$-bounded polynomial with high probability. The distribution $\chi = D_{\mathbb{Z}^n,\sigma}$ supported over R is called B-bounded if we have a B-bounded polynomial for all f sampled from χ, i.e. $\|f\|_\infty < B$ [9].

In this paper, we use a truncated Gaussian distribution χ that is B-bounded and statistically close to the discrete Gaussian. It takes sample whose norm is less than B so as to restrict the noise bound and growth during the homomorphic operations [9, 11].

2.2 The Ring-LWE Problem

It has been suggested that lots of FHE schemes based on the RLWE assumption and reductions of the worst case to average case. Lyubaskevsky, Peikert and Regev describe the Ring learning with error problem (Ring-LWE) [11]. It distinguishes which distribution is used for sampling and generates certain samples. It is an extension of the learning with error problem called LWE that based on a field rather than polynomial.

Definition 2.2.1 (Ring-LWE). For given security parameter λ, $n = n(\lambda) \in \mathbb{Z}$, let $\phi(x) = \phi_n(x) \in \mathbb{Z}[x]$ be a polynomial of degree n, let $q = q(\lambda) \geq 2$

be an integer, let $R = \mathbb{Z}/(\phi(x))$ and R/qR. Let $\chi = \chi(\lambda)$ be a distribution over R. The $Ring - LWE_{n,q,\chi}$ problem is to distinguish the following two distributions. In the first distribution one samples (a_i, b_i) uniformly from R_q^2. In second distribution one samples $(a_i, b_i = a_i \cdot s + e_i) \in R_q^2$ where s fixed for all samples is uniformly sampled from R_q, a_i's are uniformly from R_q and e_i's are from an error distribution χ. The Ring-LWE assumption is that the Ring-LWE problem is hard.

The shortest vector problem over the ideal lattice which is worst-case problem can be reduced to the Ring-LWE problem which is average-case.

Theorem 2.2.2 (**Worst-case to Average-case Reduction** [9, 11])**.** *Let $\Phi_n(x) = x^n + 1$ be a n-th cyclotomic polynomial of degree which is a power of two. Let $R = \mathbb{Z}/(x^n + 1)$, $q \equiv 1(\mathrm{mod} 2n)$ be a prime integer and χ be a $B = \omega(\sqrt{nlogn})$-bounded distribution. Then there is a randomized reduction from $n^{\omega(1)} \cdot (q/B)$-approximate worst-case SVP for ideal lattice over R to $Ring - LWE_{n,q,\chi}$.*

This theorem is useful for setting parameters.

2.3 The DSPR Problem with LTV

A. López-Alt, E. Tromer, and V. Vaikuntanathan [9] present a scheme which is based on NTRU with modifications and multikey homomorphism. It is provably secure based on standard problems in ideal lattices. The parameters, which is n, q, $\phi(x)$ of the degree n and χ, depend on security parameter λ. Its message space is $\{0, 1\}$ and all operations are evaluated on the quotient ring $R_q = \mathbb{Z}_q/(\phi(x))$. We simply introduce LTV scheme as follows.

Key Generation: Sample f_0, g from key distribution χ, then compute $f = [2f_0 + 1]_q$ such that $f \equiv 1 \,(\mathrm{mod}\, 2)$. If f is not invertible modulo q, re-sample f_0. Let $h = [2gf^{-1}]_q$ for inverse of f modulo q in R, then set the secret key f and public key h.

Encryption: Take a message m from $\{0,1\}$. Ciphertext is generated by computing $c = [m + 2e + hs]_q$ over the ring R with s, e sampled from error distribution χ.

Decryption: Compute $m' = [[fc]_q]_2 \in R$ with the secret key f.

It easy to check that as long as there is no wrap-around modulo q. Compute

$$[fc]_q = [fm + 2fe + fhs]_q = [fm + 2fe + 2gs]_q.$$

If there is no wrap-around modulo q then

$$[fc]_q(\text{mod}2) = [fm + 2fe + 2gs]_q = [fm]_q(\text{mod}2) = m.$$

This scheme is secure under the assumption of decisional small polynomial ratio problem.We simply call it DSPR problem.

Definition 2.3.1 (**DSPR Problem** [1, 9])**.** For given security λ, let $\Phi(x) \in \mathbb{Z}[x]$ be a polynomial of degree $n \in \mathbb{Z}$, let $q \in \mathbb{Z}$ be a prime integer. Let $R = \mathbb{Z}/(\phi(x))$, $R_q = R/qR$ and let χ is a distribution over the ring R. all of them depend on λ. The $DSPR_{n,q,\chi}$ problem is to distinguish between the following two distributions. In first distribution, one samples h uniformly at random over R_q. In second distribution, one samples $h = a/b$ where a and b are from the distribution χ. The $DSPR_{n,q,\chi}$ assumption is that the $DSPR_{n,q,\chi}$ problem is hard.

It holds only if $\sigma > \sqrt{q} \cdot \text{poly}(n)$ for cyclotomic polynomial $\phi(x)$ which makes secure against unbounded adversaries.

Stehl*é* and Steinfeld [13] propose a theorem for modified NTRU that changes the value 2 which could indicates the size of the message space to the value t where the t is in the $R_q^\times$, and $2e$ to te.

Theorem 2.3.2 ([1, 13])**.** *Let $\Phi_n(x) = x^n+1$ be a n-th cyclotomic polynomial of degree which is a power of two ≥ 4 and it splits into $k = k(q)$ irreducible factors modulo a prime $q \geq 5$. Let $R = \mathbb{Z}[x]/(x^n + 1)$ and $R_q = R/qR$. Let $U(R_q^\times)$ be the uniform distribution on $R_q^\times$ which is the set of invertible*

elements in R_q. Let $\chi = D^{\times}_{\mathbb{Z}^n,\sigma}$ be the spherical discrete Gaussian distribution on R_q, restricted to $R_q^{\times}$. Let $\epsilon \in (0, 1/3)$, $t \in R_q^{\times}$. Let a and b are depended on t with some distributions. Then the statistical distance $a/b \pmod q$ and $U(R_q^{\times})$ is bounded by

$$D = \begin{cases} 2^{20n} \cdot q^{-\frac{\lfloor \epsilon k \rfloor}{k}} \cdot 2n & \text{if } \sigma \geq 2n \cdot \sqrt{\log(8nq)} \cdot q^{\frac{1}{2}+\epsilon} \\ 2^{20n} \cdot q^{-2\epsilon n} & \text{if } \sigma \geq \sqrt{2n \log(8nq)} \cdot q^{\frac{1+k\epsilon}{2}} \text{ and } q \geq (2n)^{\frac{k}{1-2k\epsilon}}. \end{cases}$$

Then the DSPR problem is hard in R_q.

Chapter 3

Ring-based FHE Scheme

In this section we investigate the main scheme suggested by Bos et al. [1]. In section 3.2, a leveled somewhat homomorphic encryption of the scheme is presented and subsection 3.1.1 describes the basic encryption, decryption system and how the homomorphic operations are evaluated. In subsection 3.1.2, its security is proven with some assumptions. In section 3.2, we provide how the scheme can be converted to a fully homomorphic encryption.

3.1 Leveled Somewhat Homomorphic Encryption

Bos et al. present a couple of leveled somewhat homomorphic encryption according to homomorphic multiplication with two different evaluation keys. This keys affect the complexity of the scheme such that we could estimate which method of homomorphic multiplication is fast. We present two parts of the scheme in parallel. They call them "$YASHE$" and "$YASHE'$". A sub notation "1" stands for $YASHE$ and "2" stands for $YASHE'$.

3.1.1 BLLN Scheme

In this subsection, we describe two parts of the scheme. The first part is a basic cryptosystem including how the encrypted messages can be decryptable

and the second part explains how the homomophic operation can be computed according to the noise bound by some calculations. The basic cryptosystem is composed of the key generation, encryption and decryption as follows.

Key Generation: Sample f_0, g from key distribution χ_{key}, then compute $f = [tf_0 + 1]_q$ for $1 < t < q$. If f is not invertible modulo q, re-sample f_0. Let $h = [tgf^{-1}]_q$ for inverse of f modulo q in R, then we can generate a basic key set $\{f, h\}$ which is comprised of a secret key and public key respectively. Actually, the secret keys are f and f^{-1}.
The evaluation key set $\{evk_1, evk_2\}$ that is used for homomorphic multiplication is computed by

$$\begin{aligned} evk_1 &= [f^{-1}P_{\omega,q}\left(D_{\omega,q}\left(f\right) \otimes D_{\omega,q}\left(f\right)\right) + \mathbf{e} + h \cdot \mathbf{s}]_q \in R^{\ell^3_{\omega,q}}, \\ evk_2 &= [P_{\omega,q}\left(f\right) + \mathbf{e} + h \cdot \mathbf{s}]_q \in R^{\ell_{\omega,q}} \end{aligned}$$

where $\ell_{\omega,q} = \lfloor \log_w q \rfloor + 2$,

$$\begin{aligned} D_{\omega,q} : R \to R^{\ell_{\omega,q}}, \qquad f \mapsto ([f_0]_\omega, [f_1]_\omega, \cdots, [f_{\ell_{\omega,q}-1}]_\omega) = ([f_i]_\omega)_{i=0}^{\ell_{\omega,q}-1} \\ P_{\omega,q} : R \to R^{\ell_{\omega,q}}, \qquad f \mapsto ([f]_q, [f\omega]_q, \cdots, [f\omega^{\ell_{\omega,q}-1}]_q) = ([f\omega^i]_q)_{i=0}^{\ell_{\omega,q}-1} \end{aligned}$$

for $f = \sum_{i=0}^{\ell_{\omega,q}-1} f_i\omega^i$ with $f_i \in R$.

If $\omega = 2$, then these functions are called BitDecomp and PowerOFTwo [2].

As the function $D_{\omega,q}$ and $P_{\omega,q}$ are defined, the dot product between two vectors $D_{\omega,q}(f)$ and $P_{\omega,q}(g)$ is equal to the scalar product of f and g modulo q for some $f, g \in R$. This follows from the fact that

$$< D_{\omega,q}(f), P_{\omega,q}(g) >= \sum_{i=0}^{\ell_{\omega,q}-1} [f_i]_\omega [g\omega^i]_q \equiv g \sum_{i=0}^{\ell_{\omega,q}-1} [f_i]_\omega \omega^i \equiv f \cdot g \pmod{q}.$$

Encryption: For given integer t, we take a message from the space R/tR whose element is of the form $m + tR$. Simply, it can be written by $[m]_t$. Ciphertext that is the encryption of the message is computed by $\left[\left\lfloor \frac{q}{t} \right\rfloor [m]_t + e + hs\right]_q \in R$ for s, e sampled from error distribution χ_{err}.

Decryption: Compute $m = \left[\left\lfloor \frac{t}{q}[fc]_q \right\rceil\right]_t \in R$ to obtain the message with the secret key f.

The scheme is correctly decryptable when there exist $v \in R$ such that $\|v\|_\infty < \frac{(\triangle - q \pmod t)}{2}$ for $[fc]_q = [\triangle[m]_t + v]_q$, where $\triangle = \left\lfloor \frac{q}{t} \right\rfloor$. For some $a \in R$,

$$\begin{aligned} \frac{t}{q}[fc]_q &= \frac{t}{q} \triangle [m]_t + v \cdot \frac{t}{q} + ta \\ &= [m]_t - \frac{q \pmod t}{q}[m]_t + v \cdot \frac{t}{q} + ta. \end{aligned}$$

If $\|v\|_\infty < \frac{(\triangle - q \pmod t)}{2}$, then $\left\| - \frac{q \pmod t}{q}[m]_t + v \cdot \frac{t}{q} \right\|_\infty < \frac{1}{2}$. So, we could be correctly decrypt the ciphertext.

The homomorphic operations consisted of addition and multiplication are defined as follows.

Homomorphic Addition: Let $c_1, c_2 \in R$ be encrypted message $m_1, m_2 \in R/tR$. Compute an addition $[c_1 + c_2]_q$ that is a encrypted message $m_1 + m_2$ modulo t by adding coefficients of ciphertext componentwise.

Although the size of inherent noise increases during this operation, which has a sum of v_i and $r \in R$ where the r satisfies the equation as $[m_1]_t + [m_2]_t = [m_1 + m_2]_t + r$, $\|r\|_\infty \leq 1$ and the ciphertext c_i has a form of $[m_i]_t + v_i$, it could be decryptable.

Homomorphic Multiplication: Compute a ciphertext with multiplication of two ciphertexts. Passing by a key-switching procedure, we obtain a final ciphertext called an intermediate ciphertext, which might be correctly decryptable, is obtained.

According to the method of multiplication, two ways of homomorphic encryption are presented with different ciphertexts $\tilde{c}_{1,mul}$ and $\tilde{c}_{2,mul}$, intermediate ciphertexts. Through the key-switching procedure, the intermediate

ciphertext becomes a ciphertext decrypted under an original secret key f instead of f^2. The evaluation key evk_i is used for this procedure. If this step is omitted, the ciphertext could not be decryptable since its noise is too large to decrypt. Hence the key-switching process is essential and the intermediate ciphertext exists at all times.

The ciphertext $\tilde{c}_{1,mul}$ which is a multiplication of two ciphertexts c_1 and c_2 is computed by

$$\tilde{c}_{1,mul} = \left[\left\lfloor \frac{t}{q} P_{\omega,q}\left(c_1\right) \otimes P_{\omega,q}\left(c_2\right) \right\rceil\right]_q \in R^{\ell^2_{\omega,q}},$$

where the $\otimes$ is a tensor product which reduces the monomials $\ell^2_{\omega,q}$ to $\binom{\ell_{\omega,q}}{2}$. The way of the key-switching is obtained by

$$\begin{aligned}
\left[\left\langle D_{\omega,q}\left(\tilde{c}_{1,mul}\right), evk_1\right\rangle\right]_q &= \left[\left\langle D_{\omega,q}\left(\tilde{c}_{1,mul}\right), f^{-1} P_{\omega,q}\left(D_{\omega,q}\left(f\right) \otimes D_{\omega,q}\left(f\right)\right) + \mathbf{e} + h \cdot \mathbf{s}\right\rangle\right]_q \\
&= \left[f^{-1}\left\langle D_{\omega,q}\left(\tilde{c}_{1,mul}\right), P_{\omega,q}\left(D_{\omega,q}\left(f\right) \otimes D_{\omega,q}\left(f\right)\right)\right\rangle\right. \\
&\quad \left. + \left\langle D_{\omega,q}\left(\tilde{c}_{1,mul}\right), \mathbf{e}\right\rangle + h \cdot \left\langle D_{\omega,q}\left(\tilde{c}_{1,mul}\right), \mathbf{s}\right\rangle\right]_q.
\end{aligned}$$

In this case, the final ciphertext $c_{1,mul}$ composed of a vector of polynomials is

$$c_{1,mul} = \left[\left\langle D_{\omega,q}\left(\tilde{c}_{1,mul}\right), evk_1\right\rangle\right]_q$$

that could be correctly decryptable under the secret key f if

$$\begin{aligned}
\left[f c_{1,mul}\right]_q &= \left[\left\langle D_{\omega,q}\left(\tilde{c}_{1,mul}\right), P_{\omega,q}\left(D_{\omega,q}\left(f\right) \otimes D_{\omega,q}\left(f\right)\right)\right\rangle\right. \\
&\quad \left. + f\left\langle D_{\omega,q}\left(\tilde{c}_{1,mul}\right), \mathbf{e}\right\rangle + gt\left\langle D_{\omega,q}\left(\tilde{c}_{1,mul}\right), \mathbf{s}\right\rangle\right]_q \\
&= \left[(t/q) \cdot \left\langle P_{\omega,q}\left(c_1\right) \otimes P_{\omega,q}\left(c_2\right), D_{\omega,q}\left(f\right) \otimes D_{\omega,q}\left(f\right)\right\rangle\right. \\
&\quad \left. - r_c + f\left\langle D_{\omega,q}\left(\tilde{c}_{1,mul}\right), \mathbf{e}\right\rangle + gt\left\langle D_{\omega,q}\left(\tilde{c}_{1,mul}\right), \mathbf{s}\right\rangle\right]_q \\
&= \left[\triangle[m_1 m_2]_t + \tilde{v}_{1,mul} + f\left\langle D_{\omega,q}\left(\tilde{c}_{1,mul}\right), \mathbf{e}\right\rangle + gt\left\langle D_{\omega,q}\left(\tilde{c}_{1,mul}\right), \mathbf{s}\right\rangle\right]_q \\
&= \left[\triangle[m_1 m_2]_t + v_{1,mul}\right]_q
\end{aligned}$$

where

$$\begin{aligned}
r_c &= \frac{t}{q}\langle P_{\omega,q}(c_1) \otimes P_{\omega,q}(c_2), D_{\omega,q}(f) \otimes D_{\omega,q}(f)\rangle - \langle \tilde{c}_{1,mul}, D_{\omega,q}(f) \otimes D_{\omega,q}(f)\rangle \\
&= \left\langle \left(\frac{t}{q} P_{\omega,q}\left(c_1\right) \otimes P_{\omega,q}\left(c_2\right)\right) - \left\lfloor \frac{t}{q} P_{\omega,q}\left(c_1\right) \otimes P_{\omega,q}\left(c_2\right) \right\rceil, D_{\omega,q}\left(f\right) \otimes D_{\omega,q}\left(f\right) \right\rangle,
\end{aligned}$$

and $v_{1,mul}$ satisfies under the condition of decryption.

Another ciphertext of the multiplication $\tilde{c}_{2,mul}$ which is a multiplication of the ciphertexts c_1 and c_2 is computed by

$$\tilde{c}_{2,mul} = \left[\left\lfloor \frac{t}{q} c_1 c_2 \right\rceil\right]_q \in R,$$

and the key-switching process is followed by

$$\begin{aligned}
[\langle D_{\omega,q}(\tilde{c}_{2,mul}), evk_2\rangle]_q &= [\langle D_{\omega,q}(\tilde{c}_{2,mul}), P_{\omega,q}(f) + \mathbf{e} + h\cdot\mathbf{s}\rangle]_q \\
&= [\langle D_{\omega,q}(\tilde{c}_{2,mul}), P_{\omega,q}(f)\rangle + \langle D_{\omega,q}(\tilde{c}_{2,mul}), \mathbf{e}\rangle \\
&\quad + h\cdot\langle D_{\omega,q}(\tilde{c}_{2,mul}), \mathbf{s}\rangle]_q.
\end{aligned}$$

The final ciphertext $c_{2,mul}$ composed of a simply single polynomial is

$$c_{2,mul} = [\langle D_{\omega,q}(\tilde{c}_{2,mul}), evk_2\rangle]_q$$

that could be correctly decryptable under the secret key f if

$$\begin{aligned}
[fc_{2,mul}]_q &= [f\langle D_{\omega,q}(\tilde{c}_{2,mul}), P_{\omega,q}(f)\rangle + f\langle D_{\omega,q}(\tilde{c}_{2,mul}), \mathbf{e}\rangle + gt\langle D_{\omega,q}(\tilde{c}_{2,mul}), \mathbf{s}\rangle]_q \\
&= [f\langle D_{\omega,q}((t/q)c_1c_2), P_{\omega,q}(f)\rangle - f\cdot r_c + f\langle D_{\omega,q}(\tilde{c}_{2,mul}), \mathbf{e}\rangle \\
&\quad + gt\langle D_{\omega,q}(\tilde{c}_{2,mul}), \mathbf{s}\rangle]_q \\
&= [\triangle[m_1m_2]_t + \tilde{v}_{2,mul} + f\langle D_{\omega,q}(\tilde{c}_{2,mul}), \mathbf{e}\rangle + gt\langle D_{\omega,q}(\tilde{c}_{2,mul}), \mathbf{s}\rangle]_q \\
&= [\triangle[m_1m_2]_t + v_{2,mul}]_q
\end{aligned}$$

where

$$\begin{aligned}
f\cdot r_c &= f\cdot\left\langle \left(D_{\omega,q}\left(\frac{t}{q}c_1c_2\right)\right) - \left\lfloor D_{\omega,q}\left(\frac{t}{q}c_1c_2\right)\right\rceil, P_{\omega,q}(f)\right\rangle \\
&= \frac{t}{q}f^2\cdot c_1c_2 - f^2\cdot\left\lfloor\frac{t}{q}c_1c_2\right\rceil
\end{aligned}$$

and $v_{2,mul}$ satisfies under the condition of decryption.

3.1.2 Security of the Scheme

The security of the scheme $YASHE$ using the first way of multiplication is based on the IND-CPA under a "circular security" assumption and the RLWE assumption. The second scheme $YASHE'$ is based on the DSPR problem under the circular security and RLWE assumptions. For the circular security, replace f by distinct secret key f_j of $evk_{i,j}$ for $i \in \{1, 2\}$ whose number "1" and "2" stand for the $YASHE$ and $YASHE'$ respectively and j which is a level with $1 \leq j \leq L$, i.e. for given

$$\begin{aligned} evk_{1,j} &= [f_j^{-1} P_{\omega,q}\left(D_{\omega,q}\left(f_{j-1}\right) \otimes D_{\omega,q}\left(f_{j-1}\right)\right) + \mathbf{e} + h_j \cdot \mathbf{s}]_q \in R^{\ell^3_{\omega,q}}, \\ evk_{2,j} &= [P_{\omega,q}\left(f_{j-1}^2\right) + \mathbf{e} + h_j \cdot \mathbf{s}]_q \in R^{\ell_{\omega,q}}, \end{aligned}$$

the final ciphertext $c_{i,mul}$, which is the output of key-switching step with input ciphertext $\tilde{c}_{i,mul}$, is correctly decryptable under the secret key f_j where $\tilde{c}_{i,mul}$ is a multiplication of two $j-1$th level ciphertexts.

3.2 Fully Homomorphic Encryption

To obtain a fully homomorphic encryption from the somewhat homomorphic encryption, we have to decrease the noise of homomorphic evaluation. It it is too large, then the homomorphic properties has gone. A modulus switching or bootstrapping is a general way to reduce the noise of homomorphic computations. The noise growth during the homomorphic addition could be neglected compared with homomorphic multiplication. Therefore we just consider the multiplication. For given $fc_i = \triangle[m_i]_t + v_i (\mathrm{mod}\, q)$ with the ciphertext c_i, the noise v_i has the bound V as

$$\begin{aligned} \|v_i\|_\infty &< 2\delta t B_{key} B_{err} + \frac{1}{2} q \,(\mathrm{mod}\, t)\, \delta t B_{key} \\ &< 2\delta t B_{err} + \frac{1}{2}\delta t^2 = \delta t\left(2B_{err} + \frac{1}{2}t\right) = V, \end{aligned}$$

and the inherent noise $v_{i,mult}$ of homomorphic multiplication is

$$\begin{aligned}
\|v_{1,mult}\|_\infty &< \delta t(2+\delta\ell_{\omega,tB_{key}}\omega)V + \frac{\delta t^2}{2}(3+\delta\ell_{\omega,tB_{key}}) \\
&\quad + \frac{1}{8}(\delta\ell_{\omega,tB_{key}}\omega)^2 + \frac{1}{2} + \delta^2 t\ell^3_{\omega,q}\omega B_{err}B_{key} \\
&= \delta t(2+\delta\ell_{\omega,t}\omega)V + \frac{\delta t^2}{2}(3+\delta\ell_{\omega,t}) + \frac{1}{8}(\delta\ell_{\omega,t}\omega)^2 + \frac{1}{2} + \delta^2 t\ell^3_{\omega,q}\omega B_{err}, \\
\|v_{2,mult}\|_\infty &< \delta t(4+\delta tB_{key})V + \delta^2 t^2 B_{key}(B_{key}+t) + \delta^2 t\ell_{\omega,q}\omega B_{err}B_{key} \\
&= \delta t(4+\delta t)V + \delta^2 t^2(1+t) + \delta^2 t\ell_{\omega,q}\omega B_{err}.
\end{aligned}$$

Hence the noise increase from $O(\delta t^2)$ of $\|v_i\|_\infty$ to $O(\delta^3 t^4)$ of $\|v_{i,mul}\|_\infty$.

To make a fully homomorphic encryption of $YASHE$ with an arbitrary level L_1, set up parameters with the hypothesis of the theorem by Stehlé and Steinfeld and RLWE assumption, i.e., for $\epsilon \in (0,1)$ and $k \in (1/2,1)$, let $q = 2^{d^\epsilon}$ be a prime and let $\Phi_n(x) = x^n + 1$ be a cyclotomic polynomial of degree n which splits into k irreducible factors modulo q. Let χ_{key} be a discrete Gaussian distribution on R_q with deviation $\sigma_{key} \geq 2n\sqrt{\log(8nq)}\cdot q^k$, let χ_{err} be an asymptotically $\omega(\sqrt{2n\log(2n)})$-bounded Gaussian distribution on R. The inherent noise of a ciphertext regarding the depth L_1 circuit, organized in a leveled homomorphic multiplicative tree structure without any additions, is bounded by $C_1^{L_1}V_j + L_1C_1^{L_1-1}C_2$ where

$$\begin{aligned}
C_1 &= \delta t(2+\delta\ell_{\omega,tB_{key}}\omega) = O(\mathrm{poly}(n)\log(q)) \text{ since } \delta = n, \\
C_2 &= \frac{\delta t^2}{2}(3+\delta\ell_{\omega,tB_{key}}\omega) + \frac{1}{8}(\delta\ell_{\omega,tB_{key}}\omega)^2 + \frac{1}{2} + \delta^2 t\ell^3_{\omega,q}\omega B_{err}B_{key} \\
&\quad = O(\mathrm{poly}(n)\log(q)^3 q^k) \text{ and } V = O(\mathrm{poly}(n)\cdot q^k) \text{ for some } k \in (1/2,1)
\end{aligned}$$

by iterating the bound of $\|v_{1,mult}\|$ L_1 times as assumed that the inherent noise terms of all ciphertexts are considered to have the roughly same size $V_j = C_1V + L_1C_2$ for each level $j > 0$. To guarantee correctness of the scheme, following equality should be satisfied as

$$q = \Omega(L_1 \cdot \mathrm{poly}(n)^{L_1+1} \cdot \log(q)^{L_1+2} \cdot q^k)$$

and any circuit of depth can be estimated by

$$L_1 = O\left(\frac{(1-k)\log(q)}{\log(\log(q)) + \log(n)}\right).$$

When the level L_1 is greater than a depth $L_{dec} = O(\log(\log(q)) + \log(n))$ over $\mathbb{F}_2$ of the decryption circuit, it could be converted into a fully homomorphic encryption scheme from Gentry's Bootstrapping Theorem [5].

To obtain a fully homomorphic encryption of $YASHE'$ with a level L_2, select parameters in order to satisfy the hypothesis of the RLWE and DSPR assumptions, i.e. let $q \equiv 1 \,(\mathrm{mod} 2n)$ be a prime and let $\Phi_n(x) = x^n + 1$ be a cyclotomic polynomial of degree n which is a power of 2. Let χ_{key} be a discrete Gaussian distribution over the ring R, let χ_{err} be an asymptotically $\omega(\sqrt{2n\log(2n)})$-bounded Gaussian distribution on R. By evaluating the inherent noise bound of a ciphertext during the homomorphic operations of the depth L_2, an overall noise bound can be deduced by iterating the ℓ_∞ norm of $v_{2,mult}$. It has the bound by

$$C_1'^{L_2} \cdot V + L_2 C_1'^{L_2-1} C_2' \tag{3.2.1}$$

where

$$C_1' = \left(1 + \frac{4}{\delta t B_{key}}\right)\delta^2 t^2 B_{key} = O(\mathrm{poly}(n)) \text{ since } \delta = n, \tag{3.2.2}$$

$$\begin{aligned} C_2' &= \delta^2 t B_{key}(t(B_{key} + t) + \ell_{\omega,q}\omega B_{err}) \\ &= O(\mathrm{poly}(n)\log(q) \cdot q^k) \text{ for some } k \in (1/2, 1). \end{aligned} \tag{3.2.3}$$

To guarantee correctness of this scheme, we have that

$$q = \Omega(L_2 \cdot \mathrm{poly}(n)^{L_2+1} \cdot \log(q) \cdot q^k).$$

Therefore YASHE' can evaluate any circuit of depth

$$L_2 = O\left(\frac{(1-k)\log(q)}{\log(n)}\right).$$

When the level L_2 is greater than the depth L_{dec}, it can be converted into a fully homomorphic encryption scheme from Gentry's Bootstrapping Theorem [5].

Chapter 4

Implementation

In this chapter, the intermediate ciphertext $\tilde{c}_{2,mul}$ of $YASHE'$ is comprised of a single polynomial rather than a vector of polynomials of the intermediate ciphertext $\tilde{c}_{1,mul}$ used for $YASHE$. Since the evaluation key of $YASHE'$ consists of $\ell_{\omega,q}$ polynomials instead of $\ell^3_{\omega,q}$ of $YASHE$ which results in a simple key switching procedure, the $YASHE'$ is more practical than $YASHE$. Thus, we provide concrete parameters of $YASHE'$ in section 4.1 and an implementation result of it with special function $YASHE.Discard$ to optimize in section 4.2. Finally, we analyze the results including what makes the implementation fast and how to set parameters with optimization.

4.1 Parameter Selection

Since $YASHE'$ is secure under the Ring-LWE assumption and the DSPR assumption, we have parameters from these assumptions. The attack against RLWE, which can be seen a variant of the LWE, can be considered in the same manner of the LWE. Therefore the distinguishing attack [8] on LWE can also be applied to the RLWE problem. The distinguishing attack is to find the shortest nonzero vector in the dual lattice of $\lambda_q(A)$

$$\lambda_q^{\perp}(A) := \{y \in \mathbb{Z}^m \mid y \cdot A \equiv 0 \,(\mathrm{mod}\, q)\}$$

where A is derived from a sample of LWE as $(A, b = A \cdot \mathbf{s} + \mathbf{e})$, $A \leftarrow \mathbb{Z}_q^{m \times n}$, $\mathbf{s} \leftarrow \mathbb{Z}_{\mathbf{q}}^{\mathbf{n}}$, which is a secret, and $\mathbf{e} \leftarrow \chi_\sigma^{\mathbf{n}}$ where the χ_σ is a normal distribution

with mean 0 and standard deviation σ on $\mathbb{Z}$. If we find the shortest vector of it, one can distinguish the distribution of LWE samples from uniformly distribution. The advantage of the distinguishing attack is very close to $\epsilon = \exp(-\pi \cdot (\|y\| \cdot \mathbf{s}/q)^2)$. For security, it is sufficient to take a ℓ_∞ norm of the shortest vector y less than $\alpha \cdot q/\sigma$ where $\alpha = \sqrt{\log(1/\epsilon)/\pi}$. Actually, the size of the norm has the minimum of q and $\delta^m q^{n/m}$ where δ is called the root-Hermite factor by [8]. In case of $\delta^m q^{n/m}$, the optimal value of m is $\sqrt{n\log(q)/log(\delta)}$. By BKZ algorithm, we have a relation between the runtime in seconds and root-Hermite factor as $\lambda = 1.8/\log_2 \delta - 110$. Then, the relation

$$\alpha \cdot q/\sigma < 2^{2\sqrt{n \log_2(q) \log_2(\delta)}} \tag{4.1.1}$$

is obtained. Let q be a 127-bit prime and $n = 2^{12}$ which is a degree of polynomial $\phi_n(x)$ of quotient ring $R = \mathbb{Z}/(\Phi_n(x))$, all of which depends on security parameter λ. Fix the key distribution assumed to be bounded by $B_{key} = 1$ for evaluating the key switching step and the error distribution bounded by $B_{err} = 6\sigma_{err}$ where $\sigma_{err} = 8$ with Ring-LWE assumption. These parameters with $\omega = 2^{32}$ are presented and used for implementation of the scheme by Bos et al. [1]. The maximum of the depth L is 2 through the computation of the noise bound with L_2.

4.2 Implementation of the Scheme

We implemented the scheme $YASHE'$ in C++ with NTL library while Bos et al. [1] implemented directly in C which does not depend on any other number theory library. For given parameters of the section 4.1, we obtain the running times of the scheme on average values over 100 tests and implement it on an Intel Core i7, 3.4 GHz, 16GB RAM. The result of implementation at level 1 is that key generation runs in 152 ms, encryption runs in 21 ms, addition of ciphertexts in 44 μs, multiplication of ciphertexts including the key-switching in 29 ms, and decryption runs in 7 ms.

4.2.1 Optimization

Before optimization, we define the $YASHE.Discard$ function of Bos et al. [1], which has a input ciphertext c and output ciphertext c' as

$$c' = \mathbf{YASHE.Discard}_{\omega}(\mathbf{c}, \mathbf{i}) = \lfloor \omega^{-\mathbf{i}}\mathbf{c} \rfloor.$$

It is a truncating function by removing an insignificant multiple of ω-words of the ciphertext c. If ω^i-words are thrown away, $\omega^i c$ is equal to c except least i-th bit which is zero. If $fc = \triangle m + v (\mathrm{mod}\, q)$, then $\omega^i c' f = \triangle m + v' (\mathrm{mod}\, q)$ with $\|v'\|_\infty \leq \|v\|_\infty + \frac{1}{2}\delta\omega^i \|f\|_\infty$. Therefore, both cihpertext length and the number of components of the evaluation key are reduced per multiplication of the key-switching procedure. Since the noise increase with size $\frac{1}{2}\delta\omega^i\|f\|_\infty$, the inherent noise $\|v'\|_\infty$ of level L is bounded by $C_1'^L \cdot V' + LC_1'^{L-1}C_2'$ where C_1', C_2', and V are the same as (3.2.2), (3.2.3) and

$$\begin{aligned} \|v'\|_\infty &\leq \|v\|_\infty + \frac{1}{2}\delta\omega^i\|f\|_\infty \\ &\leq V + \frac{1}{2}\delta\omega^i\|f\|_\infty \leq V + \frac{1}{2}\delta\omega^i(1+t) = V'. \end{aligned} \tag{4.2.1}$$

If $C_1'^L \cdot V' + LC_1'^{L-1}C_2'$ is bounded by $\frac{\triangle}{2}$ then it could be correctly decryptable.

To sum up all conditions of the noise bound, we get approximately inequalities that represent relations of n, ω, q, t and ℓ by

$$\begin{aligned} n^2t^3\frac{n}{2}\omega^i &\leq \frac{q}{2t} \Rightarrow n^3t^4\omega^i \leq q \text{ if } i \geq 1 \\ n^2t^2(\ell_{\omega,q} \cdot \omega \cdot B_{err}) &\leq \frac{q}{2t} \Rightarrow 2n^2t^3(\ell_{\omega,q} \cdot \omega \cdot B_{err}) \leq q \\ \frac{\log(q) - 3}{2\sqrt{\log(q)\log(\delta)}} &\leq \sqrt{n}, \;\; \lfloor \log_\omega(q) \rfloor + 2 = \ell_{\omega,q}. \end{aligned}$$

These are convenience for evaluating the noise fast regarding parameters. The first and second equality is derived from (4.2.1), (3.2.2) and (3.2.3), and the third equality is induced from (4.1.1).

For given parameters from section 4.1, we present data of the scheme implemented by comparing the result which is already presented before starting the section 4.2 with using the $YASHE.Discard$ function. In given parameters, we percieve that the value i has the maximum 1 which means that ω-words can be discarded and one evaluation key diminish. Look into the table 4.1 that shows the running time of multiplication by millisecond.

	KeyGen	**Encrypt**	**Add**	**Mult**	**Decrypt**
i=0	143	21	0.044	29	7
i=1	136	20	0.044	28	6

Table 4.1: Running times of $\log \omega = 32$, $t = 1024$ for unit [ms]

We can see that the multiplication time decreases about 3 percent and key generation time decreases about 4 percent compared with $i = 0$ that dose not use the discarding function. Particularly, the addition time is the shortest than the others about 1000 times. Unless q and n change, the timings of encryption, decryption and addition do not fluctuate.

According the parameter ω, the number of elements of the evaluation key varies such that the running time changes. We derived the fact that the number of components is a reciprocal proportion to ω as the way of key generation. Therefore, we implement it by changing $\log \omega = 32$ to $\log \omega = 48$ to make more efficient. The variation of ω cause the components of evaluation key to drop to $\ell_{2^{48},q} = 4$ from $\ell_{2^{32},q} = 5$. However, we can not use ω more than 2^{64} to reduce them because the inherent noise of multiplication increase too large to decrypt. The result of implementation with $\log \omega = 48$ using $YASHE.Discard$ function is shown by Table 4.2.

	KeyGen	**Encrypt**	**Add**	**Mult**	**Decrypt**
i=0	117	19	0.044	24	6
i=1	116	20	0.044	23	5

Table 4.2: Running times of $\log \omega = 48$, $t = 1024$ for unit [ms]

In this case, the effect of the function is very slight as the running time

of key generation decreases about 0.85 percent and that of multiplication decreases about 4 percent. But there is some effects on ω as the one decreases about $14 \sim 18$ percent and the other decreases about 17 percent from table 4.1 and table 4.2.

Furthermore, we implement it by changing the parameter t which decides message space as the ring R_{256} instead of R_{1024}. Table 4.3 shows the experimental result of that parameter. It uses $\log \omega = 48$ rather than $\log \omega = 32$ because we already know the former makes this scheme faster than the latter.

	KeyGen	**Encrypt**	**Add**	**mul**	**Decrypt**
i=0	121	20	0.044	25	6
i=1	113	20	0.044	23	6

Table 4.3: Running times of $\log \omega = 48$, $t = 256$ for unit [ms]

The variation of t does not have any effects on running time from table 4.2 and table 4.3. But the time of key generation decreases about 6 percent and that of multiplication decreases about 8 percent by using this function.

In the similar manner, we use $\log \omega = 64$ to reduce procedure of key-switching, which can only use $\ell_{\omega,q} = 3$ components of evaluation key without discarding words. If not, we should use $\ell_{\omega,q} = 4$ components. To make the scheme faster, we may consider $YASHE.Discard$ function. However, using this function makes it impossible to decrypt correctly since the inherent noise is too large to decrypt. Its experimental result is provided by table 4.4.

	KeyGen	**Encrypt**	**Add**	**Mul**	**Decrypt**
i=0	97	19	0.044	22	6

Table 4.4: Running times of $\log \omega = 64$, $t = 256$ for unit [ms]

The data of the table is about 19 percent of key generation and about 12 percent of multiplication faster than the result of the table 4.3 without

discarding words.

From some experimental results, we can deduced that the value ω, which affects the process of key-switching with the number of components of evaluation key, and the fact that whether $YASHE.Discard$ function is used or not are the important factors for deciding the running times. When the pair of n and q consist of very small size, the maximal depth (or level) L_{max} is very low. Since bootstrapping requires the depth L to be about 50, we additively implement the scheme under this circumstances, which should be set as the optimal parameter $q = 2048$ bit prime, $t = 2$, $n = 2^{16}$, $\omega = 293$, $i = 1$, and $\ell = 8$.

	KeyGen	**Encrypt**	**Mul**	**Decrypt**
i=1	8.5	0.3	296	0.09

Table 4.5: Running times of $L = 50$, $\log \omega = 293$, $t = 2$ for unit [s]

Chapter 5

Conclusion

For fixed parameter q that is a 127 bit prime and polynomial degree $n = 2^{12}$, we provide the experimental data by implementing the BLLN scheme. Then, we performed optimization by adjusting parameters and using particular function $YASHE.Discard$. Its results is presented by tables. When the parameter t, which determines message space, is fixed by 1024, the running times are reduced about 18 percent of key generation and about 20 percent of multiplication. When t is 256, the one decreases about 19 percent and the other decreases about 12 percent. As a result, we can see which parameters make the scheme faster and how to choose these parameters we want to implement it as well as we can estimate how faster it is, depending on some technique of optimization.

Bibliography

[1] J. Bos, K. Lauter, J. Loftus, and M. Naehrig. Improved security for a ring-based fully homomorphic encryption scheme. Cryptology ePrint Archive, Report 2013/075, 2013. http://eprint.iacr.org/2013/075/.

[2] Z. Brakerski. Fully homomorphic encryption without modulus switching from classical GapSVP. In Advances in Cryptology - Crypto 2012, volume 7417 of Lecture Notes in Computer Science, pages 868-886. Springer, 2012.

[3] Z. Brakerski, C. Gentry, and V. Vaikuntanathan. Fully homomorphic encryption without bootstrapping. In ITCS, pages 309-325, 2012.

[4] C. Gentry. A fully homomorphic encryption scheme, Stanford University PhD thesis, 2009, http://crypto.stanford.edu/craig/craig-thesis.pdf.

[5] C. Gentry. Fully homomorphic encryption using ideal lattices. In Michael Mitzenmacher, editor, STOC, pages 169-178. ACM, 2009.

[6] S. Goldwasser and S. Micali. Probabilistic encryption and how to play mental poker keeping secret all partial information, in Proceedings of the 14th ACM Symposium on Theory of Computing|STOC 1982, ACM (1982), 365-377.

[7] J. Hoffstein, J. Pipher, and J. H. Silverman. NTRU: A ring-based public key cryptosystem. In J. Buhler, editor, ANTS, volume 1423 of Lecture Notes in Computer Science, pages 267-288. Springer, 1998.

[8] R. Lindner and C. Peikert. Better key sizes (and attacks) for LWE-based encryption. In Proceedings of the 11th international conference on

Topics in cryptology:CT-RSA 2011, CT-RSA'11, pages 319-339, Berlin, Heidelberg, 2011. Springer Verlag.

[9] A. L*ó*pez-Alt, E. Tromer, and V. Vaikuntanathan. On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption. In STOC, pages 1219-1234, 2012.

[10] V. Lyubashevsky and D. Micciancio. Generalized compact knapsacks are collision resistant. In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, ICALP (2), volume 4052 of Lecture Notes in Computer Science, pages 144-155. Springer, 2006.

[11] V. Lyubashvesky, C. Peikert, and O. Regev. on ideal lattices and learning with errors over rings. In EUROCRYPT, volume6110 of Lecture notes in Computer Science, page 1-23, 2010.

[12] R. Rivest, L. Adleman, and M. Dertouzos. On Data Banks and Privacy Homomorphisms, in Foundations of Secure Computation, Academic Press, New York (1978), 169-180.

[13] D. Stehl*é* and R. Steinfeld. Making NTRU as secure as worst-case problems over ideal lattices. In Advances in Cryptology - EUROCRYPT 2011 - 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques, volume 6632 of Lecture Notes in Computer Science, page 27. Springer, 2011.

국문초록

Rivest 외 3명은 암호화된 상태에서 계산 가능한 완전동형암호를 제시하였다. 그러나 이 암호체계는 안전하지 않는 것으로 판명되어 이후 이를 변형하거나 개선된 수많은 동형암호들이 소개되었다. 그 중 López-Alt 외 3명이 제안한 완전동형암호는 NTRU 암호체계에 기반한 것으로 다른 암호체계보다 더 효율적이다. Bos 외 3명은 그들의 스킴보다 효율성면에서 개선된 암호화 알고리즘을 개발하였다. 이는 López-Alt 외 3명의 암호체계에 기반된 DSPR 문제와 Modulus switching 단계를 제거한 결과로 본 논문은 이들이 제안한 암호알고리즘을 살펴봄과 동시에 구현 및 최적화한 결과를 분석하여 제시한다.

주요어휘: BLLN,NTRU, 동형암호, 구현
학번: No. 2011-23211

감사의 글

학위 기간동안 부족한 제자를 보살펴주시고 아낌없는 조언과 가르침을 주신 천정희 선생님께 감사의 말씀드립니다. 선생님께 학문에 대한 열정과 삶의 지혜를 배웠습니다. 그리고 석사 논문 심사를 해주신 이기암 교수님, 오병권 교수님께 감사드립니다.

논문작성에 있어서 많은 조언과 도움을 주신 이문성 박사님께 감사드립니다. 논문 교정을 함께 해준 미란언니, 현숙언니, chocolat au lait를 사주며 격려를 아끼지 않았던 충훈 오빠, 심사준비에 도움을 준 형태오빠, 한솔언니 그리고 길지 않은 시간을 함께 보낸 연구실 선후배 동료들에게 감사합니다. 대학원에서 서로를 의지하며 보낸 대학원 동기들, 혜림이, 누리언니, 도영이, 지연이, 미림이, 경민오빠, 찬우오빠에게도 감사합니다.

무엇보다도 저에게 가장 큰 힘이 되어주신 부모님, 학위과정에서의 조언과 격려를 많이 해주신 친척분들과 제가 가장 아끼는 동생에게도 감사의 말 전합니다. 마지막으로 학부교수님들을 비롯하여 이 자리까지 올 수있게 해준 모든 분들께 감사를 표합니다.