# 완전동형암호 알고리즘 TFHE 스킴의 핵심 연산 연구

오유리*, 이희승**, 최창림**, 박성천*, 김동규**
*한국전자통신연구원, **한양대학교 융합전자공학부
* e-mail : *oyr5624@etri.re.kr, scpark@etri.re.kr*
**e-mail : *hslee0426@hanyang.ac.kr, crchoi@hanyang.ac.kr , DQKIM@hanyang.ac.kr*

# Research of Core Operation in TFHE scheme

Hee-Seung Lee**, Chang Rim Choi**, *Dong Kyue Kim**
Department of Electronic Engineering
Hanyang University

Yu-Ri Oh*, Seong-Cheon Park*
Electronics and Telecommunications Research Institute

## Abstract

Due to the need of new cryptography of post-quantum era, there are a lot of research about Fully Homomorphic Encryption (FHE) schemes. In order to commercialize FHE schemes, the computation time of FHE schemes must be reduced. In this paper, we analyze the detailed structure of operation in Fast Fully Homomorphic Encryption over the Torus (TFHE) scheme which is the fastest scheme among FHE schemes. Afterwards, we analyze that the core operation of TFHE scheme is the external product, and describe the direction of future research.

## I. 서론

현재 널리 쓰이고 있는 공개키 알고리즘들은 양자 컴퓨터가 상용화 될 경우 Shor Algorithm에 의해 쉽게 깨질 수 있다. 이로 인해 양자내성암호에 대한 필요성이 높아졌다. 뿐만 아니라 암호화된 데이터를 연산처리 하기 위해서는 암호화된 데이터를 복호화하는 과정에서 정보의 유출이 빈번히 발생하고 있다. 따라서 양자 컴퓨터에서도 안전하고 암호화된 데이터 상에서 연산을 처리할 수 있는 암호 기술로써 완전동형암호에 대한 연구가 활발히 진행되고 있다.

동형암호는 평문과 암호문의 동형 성질로 인해 암호문 상태에서도 연산이 가능한 차세대 암호기술이다. 하지만 동형암호는 암호문 간의 일정 횟수 이상 계속 연산을 수행하면 암호강도를 위해 추가한 노이즈가 연산 과정에서 크기가 커져서 메시지를 침범하거나 또는 거듭되는 암호문 간의 연산으로 메시지 크기도 커져서 이를 관리하기 위해 연산 횟수를 제한하고 있다. 완전동형암호는 연산 과정에서 축적된 암호문의 노이즈 크기를 줄여주는 부가적인 부트스트래핑(bootstrapping) 과정을 통해 동형암호의 연산을 중단됨이 없이 지속 가능하게 하는 동형암호체계를 말한다[1]. 완전동형암호는 다양한 스킴이 존재하는데 그 중에서 TFHE 스킴은 Boolean 연산으로써 경량 연산 관점에서 주목받고 있는

스킴이다. 그럼에도 TFHE 스킴은 실용화하기에는 느린 연산 처리 성능으로 인해 연산 처리 속도를 개선하는 연구가 필요한 스킴이다[2].

우리는 TFHE 스킴의 연산 속도를 개선하기 위해서 스킴과 공개된 SW라이브러리로써 Palisade[3], TFHE 코드[4]를 분석하여 연산 코스트가 높은 부분을 특정하였다. 본 논문에서는 TFHE 스킴의 연산 구조를 소개하며 TFHE 스킴이 실용적 수준이 되기 위해 연산 처리 속도를 향상시켜야 하는 주요 연산에 대해서 소개한다. 그 전에 앞서 TFHE 스킴을 이해하는데 필요한 기본적인 알고리즘을 설명한다.

## II. TFHE 스킴의 암호화 방식

TFHE 스킴의 암호화에 적용되는 대수 구조에는 크게 TLWE, TRLWE, TRGSW 3가지 형태가 적용된다. 약어 T는 토러스(Torus)를 의미한다. 즉 3가지 형태는 각각 토러스 구조 위에서 수행되는 LWE, RLWE, RGSW를 의미한다. 토러스 구조란 모듈러 1 공간상에서의 구조로써 [0,1] 범위인 실수들의 집합을 의미하며, 기호 R은 실수, Z는 정수일 때 T = R/Z = R mod 1이다. 아래 그림1은 토러스 구조의 예시로 q는 $Z_q$의 모듈러 q공간 상을 말하며 p는 평문의 비트 수를 의미하고 델타는 p/q를 의미한다.

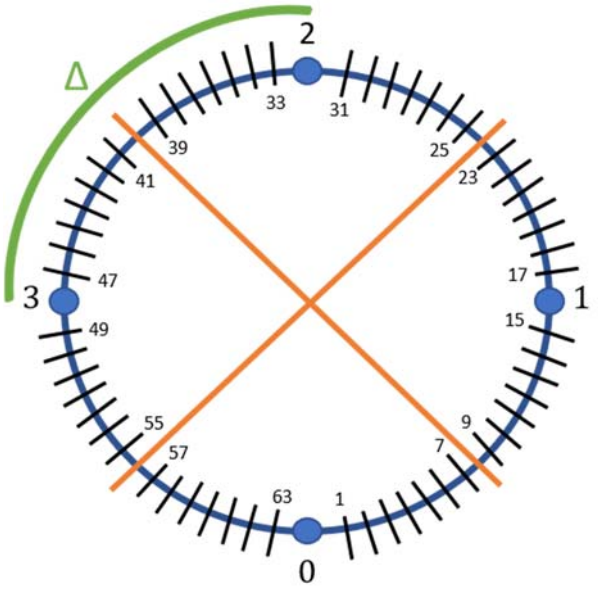


그림 1. 토러스 구조(q=64, p=4)

### 2.1 TLWE

TLWE의 기본인 LWE(Learning with Error)란 노이즈가 추가된 선형 방정식으로부터 생성된 벡터를 균등한 랜덤벡터로부터 구별하는 문제를 말한다. LWE는 크게 Search LWE와 Decisional LWE 두 가지로 구분이 된다. 이 중 Decisional LWE는 동형암호가 기반으로 하는 수학적 어려움이다[5]. LWE 암호화를 통해 1비트의 평문 m은 n개의 원소로 이루어진 $\vec{a}$ 벡터 하나와 $b = \vec{a} \times \vec{s} + m + e$ ( $\vec{s}$ : 비밀키 벡터)를 만족하는 원소 b 1개가 함께 있는 구조가 된다.

### 2.2 TRLWE

TRLWE의 기본인 RLWE의 R은 환(ring)을 의미하며 RLWE 문제는 LWE의 대수적 변종으로 벡터 대신 환의 원소를 이용한다. 즉 TRLWE는 수학적으로 T[X]=R[X]/( $X^N + 1$ ) mod 1을 만족한다[6]. 다항식 평문 M은 RLWE 암호화를 통해 N개의 계수를 가진 각각 N−1 차 다항식 2개가 함께 있는 구조의 암호문이 된다.

### 2.3 TRGSW

TRGSW의 기본인 GSW는 Gentry등 제안자 3인의 이름을 따서 만든 구조로 LWE와 RLWE는 곱셈 연산이 바로는 불가능하여 이를 곱셈 연산이 가능하도록 제안한 구조이다[7]. 즉 TRGSW는 2.2의 TRLWE에 적용한 환과 토러스 개념을 GSW에 적용한 구조이다. 다항식 평문M은 TRGSW 암호화를 통해 N개의 계수를 가진, 파라미터 $l$ 값에 따라 $2l \times 2$ 개의 다항식이 함께 있는 구조의 암호문이 된다.

## III. 세부 핵심 연산 구조 및 분석

TFHE 스킴은 Boolean 연산 특성으로 인해 여러 종류의 완전동형암호 알고리즘 중에서 가장 경량 동형연산이라는 특성이 장점이다. Boolean연산은 Logic gate 수준의 동형연산을 처리하게 되고 Logic gate 수준의 동형연산 1회가 완료될 때 마다 부가적인 부트스트래핑 동형연산을 즉시 수행하여 완전동형암호 연산를 완성한다. 하지만 Logic gate 수준의 동형연산이 완료될 때 마다 부가적인 부트스래핑 동형연산을 필요하므로 최종 연산결과를 얻기까지 지연시간이 부가되는 단점이 있다.

이때 Boolean 연산인 AND, OR 등의 Logic gate들은 NAND gate를 통해 모두 처리할 수 있으므로 TFHE 스킴의 연산 속도를 향상시키기 위해서는 NAND gate 연산에 걸리는 시간을 줄이게 되면 달성가능하다. 따라서 NAND gate로 처리되는 TFHE 연산 속도를 향상시키기 위한 목적으로 관련 연산을 세부적으로 분석하였다. 세부 구조와 전체적인

분석 결과는 아래 그림2와 같으며 세부 연산 별 소요 시간은 아래 표1과 같다.

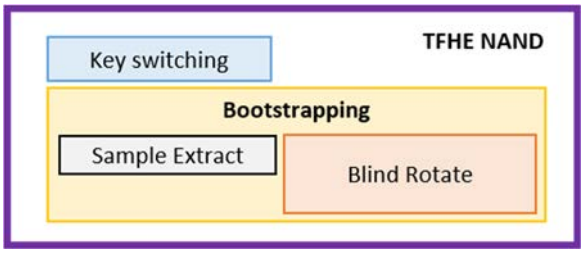


그림 2. NAND gate 연산 세부 구조

| | 소요 시간 (ms) |
|---|---|
| TFHE NAND | 15 |
| Key switching | 1.5 |
| Boostrapping | 13.5 |
| Sample Extract | ≈0 |
| Blind Rotate | ≈13.5 |

표1. 세부 연산 별 소요시간

연산 처리 속도를 분석하기 위해 palisade 라이브러리[3]와 TFHE 라이브러리[4]를 Intel(R) Core(TM) i7-10700 CPU 2.90 GHz PC에서 측정하였다.

NAND gate 연산이 한 번 진행되는 데에는 약15ms가 소요된다. 이때 NAND gate 연산은 키 스위칭(Key Switching)과 부트스트래핑을 포함하고 있고 각각 연산에 약1.5ms와 13.5ms가 소요된다. 보다 상세하게 부트스트래핑을 분석한 결과 연산 구성은 Sample Extract와 Blind Rotate 연산으로 나뉜다. Blind Rotate는 앞서 설명한 TRLWE와 TRGSW 구조가 되며 부트스트래핑 소요 시간의 13.5ms 대부분을 차지한다. Blind Rotate연산은 암호강도를 위해 사전 설정된 횟수 t (분석 라이브러리는 630회)만큼의 CMUX 연산을 수행하며, CMUX 연산 1회는 약 0.022ms가 소요되어, t(분석 라이브러리는 630회) × 0.022ms = 약 13.5ms로 분석되었다. 결과적으로 NAND gate연산에서 대부분의 연산 시간을 소요하는 것은 CMUX 연산임을 확인하였다.

## Ⅳ. 결론 및 향후 연구 방향

TFHE 스킴의 연산 속도를 개선하기 위해 NAND gate 연산을 특정하고 공개된 라이브러리 SW를 활용하여 연산량을 분석하였다. 본 논문에서는 TFHE의 느린 연산 성능을 개선하기 위해서는 수학적 원리를 바탕으로 분석한 연산 구조들 중에서 NAND gate 연산 처리 시간을 향상시키는 것이 TFHE 스킴 전체 성능을 향상하는데 높은 비중을 차지한다는 결론에 도달하였다.

향후 연구 방향으로는 분석한 TFHE 스킴의 느린 성능의 원인을 제공하는 연산들에 대해 HW로 구현하여 연산 속도를 향상시키는 연구를 이어갈 계획이다.

**감사의 글**

본 연구는 2021년도 한국전자통신연구원 내부연구사업 융합기획선행연구사업 지원으로 수행되었음. (21YT1200,단말 FHE 경량 HW 기술 개발)

## 참고문헌

[1] JS Yoo, J Yoon. LWE 와 완전동형암호에 대한 분석 및 동향. Review of KIISC, 2020.

[2] Chillotti, I., Gama, N., Georgieva, M., Izabachene, M. Tfhe: fast fully homomorphic encryption over the torus. Journal of Cryptology, 33(1), 34-91. 2020.

[3] https://palisade-crypto.org/

[4] https://tfhe.github.io/tfhe/

[5] Micciancio and O Regev. Lattice-based cryptography. Post Quantum Cryptography, pp. 147-191. Springer, February 2009.

[6] BM Case, S Gao, G Hu, Q Xu. Fully homomorphic encryption with k-bit arithmetic operations. Cryptology ePrint Archive, 2019.

[7] C. Gentry, A. Sahai, and B. Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In Crypto'13, 2013.