TFHE 파라미터의 최적화에 대한 연구

하승진[1], 주유연[1], 남기빈[1], 백윤흥[1]
[1]서울대학교 전기·정보공학부, 서울대학교 반도체 공동연구소

sjha@sor.snu.ac.kr, yyjoo@sor.snu.ac.kr, kvnam@sor.snu.ac.kr, ypaek@snu.ac.kr

A Study on the Optimisation of the TFHE Parameters

Seungjin Ha[1], Youyeon Joo[1], Kevin Nam[1], Yunheung Paek[1]
[1]Dept. of Electrical and Computer Engineering and Inter-University Semiconductor Research Center(ISRC), Seoul National University

요 약

본 논문에서는 TFHE(Fast Homomorphic Encryption over the Torus) 파라미터의 중요성과 그 파라미터가 동형암호 연산의 성능에 미치는 영향을 다룬다. 본 연구는 TFHE의 핵심 구성 요소인 TLWE, TRLWE, TRGSW 샘플의 파라미터 설정이 어떻게 보안 수준, 정확도, 처리 속도에 영향을 미치는지 분석한다. 이를 통해, 정확도와 처리 속도 같은 성능과 보안 수준 사이의 균형을 이루기 위한 파라미터 조정의 중요성을 강조하고, TFHE 파라미터를 사용하는 방법에 대한 구체적인 가이드라인을 제공한다. 본 논문은 동형암호 기술의 효율성을 극대화하고, 보다 안전하고 효율적인 데이터 처리 방법을 개발하는 데 기여할 것으로 기대된다.

1. 서론

동형암호(Homomorphic Encryption, HE)는 암호화된 데이터에 대해 아무런 복호화 없이 직접 연산을 수행할 수 있는 기술로, 데이터의 보안을 유지하면서도 다양한 처리를 가능하게 한다. 이러한 특성 덕분에 동형암호는 클라우드 컴퓨팅, 의료 정보 분석, 금융 데이터 처리 등 개인정보 보호가 중요한 분야에서 각광받고 있다.

동형암호 연산을 수행할 때는 다양한 동형암호 파라미터들이 연산 성능에 상당한 영향을 미친다. 예를 들어, 다항식 차수(Polynomial Modulus Degree) N을 높이면 암호문의 크기가 증가하고 연산 속도는 감소하지만, 계수 모듈러스(Coefficient Modulus) Q를 증가시키면 더 복잡하거나 정밀한 동형암호 연산이 가능해진다. 이처럼 원하는 연산 결과를 얻기 위해서는 동형암호 파라미터를 신중하게 설정해야 한다. 예컨대, CKKS(Cheon-Kim-Kim-Song) 암호 체계[1]는 좋은 파라미터 선택에 대해 잘 정리되어 있으며, Microsoft SEAL 라이브러리[2]는 다양한 다항식 차수 N에 대하여 적절한 계수 모듈러스 Q를 예시로 제시하고 있다.[3]

동형암호 체계 중 하나인 TFHE(Fast Homomorphic Encryption over the Torus)[4]는 다른 동형암호 체계와 달리 임의의 연산을 지원할 수 있다는 점에서 특히 유용하다. CKKS와 같이 근사값을 계산해야 하는 제한된 기능 지원 대신, TFHE는 보다 높은 연산 정확도를 제공한다. 본 논문에서는 TFHE에서는 어떠한 파라미터들이 사용되고, 이들 파라미터를 조절함으로써 어떤 부분에 영향을 미칠 수 있는지, 또한 원하는 보안 수준과 정확도의 연산 결과를 얻기 위해 파라미터를 어떻게 설정해야 하는지를 알아보려고 한다.

현재 TFHE 라이브러리[5]에서는 80/128-bit 보안 수준 파라미터를 제공하고 있지만, 이는 선택의 폭이 제한된 상태이다. 이에 따라 본 논문은 다양한 애플리케이션에서의 요구사항을 충족시킬 수 있도록 TFHE 파라미터 설정에 관해 보다 명확한 가이드라인을 제공함으로써, 더 안전하고 효율적인 동형암호 개발에 중요한 역할을 할 것이다.

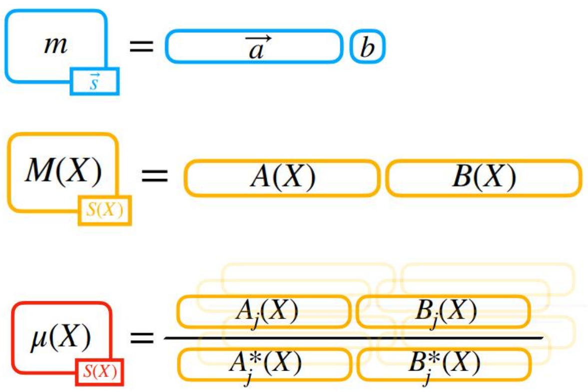
2. TFHE 파라미터

TFHE 암호 체계는 TLWE, TRLWE, TRGSW 샘플을 포함하며, 각 샘플은 동형 암호 연산 과정에서 서로 다른 역할을 수행한다. 이 샘플들은 연산 도중 서로 다른 형태로 변환되며, 이러한 유연성은

다양한 연산을 가능하게 하고 데이터의 보안을 유지하는 데 중요한 기여를 한다. 이들 각각의 사용은 암호화된 데이터에 대한 복잡한 연산을 효율적으로 처리할 수 있게 하며, TFHE의 다기능성과 높은 보안 수준을 보장한다. 이 샘플들은 다양한 파라미터를 활용하여 데이터를 안전하게 암호화하고 연산을 수행한다. [4][5]

<그림 1>은 TFHE 암호화 과정에서 사용되는 TLWE, TRLWE, 그리고 TRGSW 샘플의 구조를 나타내며, 각각의 샘플 유형이 어떻게 암호문을 구성하는지 보여준다. TLWE 샘플은 벡터 $\vec{a}$와 오류항 e를 b에 포함한 단순한 형태로, 각 성분이 평문 메시지 m과 관련된 연산을 수행한다. TRLWE와 TRGSW 샘플은 다항식 $A(X)$와 $B(X)$, 혹은 $A_j(X)$와 $B_j(X)$로 표현되며, 여기서 사용되는 다항식들은 평문 메시지 $M(X)$ 또는 $\mu(X)$를 다룬다.

<그림 1> TFHE의 각 샘플 유형의 암호문 [6]
(위에서부터 차례대로 TLWE, TRLWE, TRGSW)



<표 1> TFHE의 각 샘플 유형의 파라미터

TLWE	TRLWE	TRGSW
n(차원)	N(다항식 차수)	l(레이어의 수)
α(에러 표준 편차)	k(다항식의 수)	Bg(비트 파라미터)
	α(에러 표준 편차)	

TLWE(Torus Learning With Errors) 샘플은 가장 기본적인 암호화 단위로, 간단한 노이즈가 추가된 선형 연산, 예를 들어 덧셈 및 스칼라 곱셈을 수행하는 데 사용된다. 또한, 부트스트래핑 과정과 키 전환 메커니즘에 중요하게 사용된다. 부트스트래핑은 암호화된 데이터의 노이즈를 줄이는 데 쓰이며, 키 전환은 암호화된 데이터를 한 키에서 다른 키로 변환하는 과정에 필요하다.

TLWE 샘플에서 사용하는 파라미터는 n(차원)과 α(에러 표준 편차)가 있다. n은 TLWE 샘플의 복잡도와 보안 수준을 결정한다. n이 크면 클수록 공격자가 필요로 하는 계산량이 증가하여 더 높은 보안을 제공하지만, 암호문의 크기와 처리 복잡도를 증가시켜 계산 시간이 늘어난다. α는 에러의 표준 편차로, 암호화 과정에서 발생하는 노이즈의 양을 결정하여 암호화된 데이터의 정확성과 보안성에 영향을 준다. α가 작을수록 생성되는 노이즈가 줄어들어 복호화 시 데이터는 더 정확하게 복원되지만, 너무 작은 노이즈 값은 암호문의 노이즈 패턴을 분석하기 쉬워져 보안성이 감소할 수 있다.

TRLWE(Torus Ring Learning With Errors) 샘플은 TLWE와 유사하게 노이즈가 추가된 연산을 수행하지만, 다항식 형태로 이루어진 데이터에 대해 사용된다. 특히, 복잡한 연산이 필요한 경우에 사용되며, 동형 암호화된 데이터의 더 복잡한 처리와 효율적인 연산을 가능하게 한다. 부트스트래핑 과정에서도 다항식 연산을 지원하고 효율성을 높이는 중요한 역할을 하여, 암호화된 데이터의 유용성을 높이는 데 사용된다.

TRLWE 샘플에서 사용하는 파라미터는 N(다항식 차수), k(다항식의 수), α(에러 표준 편차)가 있다. N은 다항식의 차수로, 이 값이 높을수록 하나의 암호문에 더 많은 데이터를 효과적으로 인코딩할 수 있으므로 데이터 처리 능력이 향상되지만 더 큰 차수의 다항식 연산은 처리 시간을 늘리고, 결과적으로 전체 연산 속도를 느리게 만든다. k는 사용되는 다항식의 수이다. k가 높을수록 암호문의 구조가 더 복잡해지고, 공격자가 암호문을 분석하기 어려워져 보안성이 증가하지만, 동시에 암호화 및 복호화 과정에서 더 많은 계산을 필요로 하므로 계산 복잡도도 증가한다. α는 TLWE와 마찬가지로, 에러의 표준 편차가 작을수록 더 정확한 계산이 가능하다.

마지막으로 TRGSW(Torus Ring GSW) 샘플은 특히 조건문 처리와 같은 복잡한 논리 연산과 블라인드 회전(Blind Rotations)을 수행하는 데 필수적이다. 이 샘플은 부트스트래핑 과정에서도 매우 중요하며, 암호화된 데이터에 대해 복잡한 논리 회로를 구축할 수 있게 해준다. 이러한 기능은 동형 암호 시스템에서 다양한 알고리즘을 구현하는 데 중요한 역할을 한다.

TRGSW 샘플에서 사용하는 파라미터는 l(레이어의 수)와 Bg(비트 파라미터)가 있다. l은 TRGSW

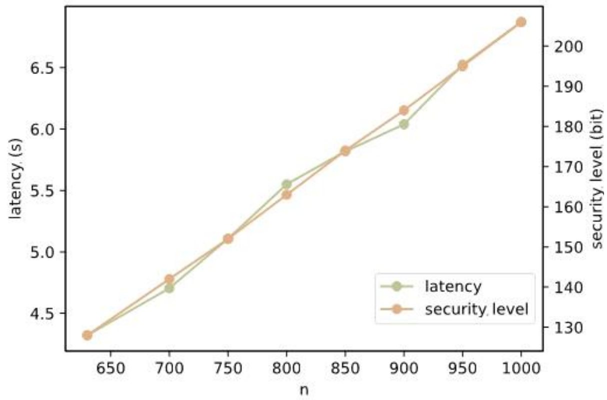
샘플에서 사용되는 다항식의 수로, 더 많은 l은 더 강력한 암호화를 가능하게 하지만 처리 시간이 증가한다. Bg는 TRGSW 샘플의 각 층(layer)에서 사용되는 정수의 크기를 결정하며, 이는 암호화된 데이터의 처리 및 분석을 위해 각 층에서 수행되는 계산의 정확도와 관련이 있다. TRGSW 샘플의 '각 층'은 암호문 내에서 다양한 계산 정밀도 레벨을 제공하며, 이러한 다중 레이어 구조는 복잡한 연산을 수행할 때 필요한 유연성과 정밀도를 제공한다. Bg가 크면 더 큰 범위의 정수를 포함함으로써 공격자가 사용하는 분석 기술을 효과적으로 방해할 수 있어 보안이 강화되지만, 큰 정수를 처리해야 하므로 계산 시간이 증가한다.

3. TFHE 파라미터의 trade-off

TFHE는 파라미터 선택이 보안 수준(security level), 정확도(accuracy), 그리고 처리 속도(latency)에 상당한 영향을 미치는데, 각 파라미터가 이러한 요소들과 관련되어 있어 파라미터를 결정할 때 파라미터의 trade-off를 고려해야 한다.[7][8] 보안 수준 측정의 경우 Lattice Estimator[9]의 classical primal attack에 대한 BKZ cost model을 활용하였으며, 처리 속도의 경우 TFHE 라이브러리[5]에서 16비트 정수 두 개를 full adder 회로로 덧셈을 10번 시도한 평균값으로 측정하였다. 정확도의 경우 이때 평문상에서 기대되는 결과값과 동형암호 덧셈 후 결과값을 비교하여 측정하였다.

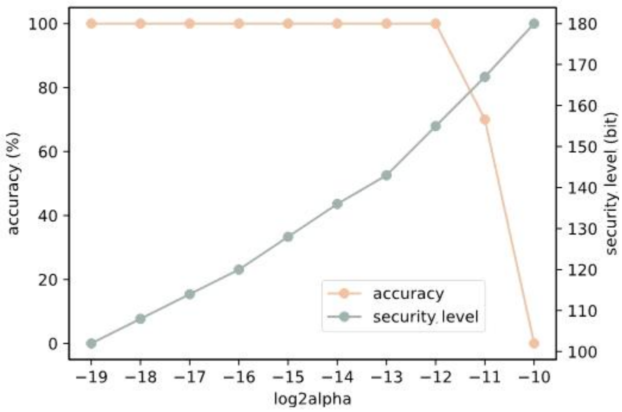
처리속도와 보안수준의 trade-off. 먼저, 처리 속도와 보안 수준 사이의 trade-off를 고려해야 한다. TLWE 샘플의 n은 TLWE 샘플의 차원을 나타내며, 이 값이 클수록 공격자가 복호화를 시도할 때 더 많은 계산을 필요로 하기 때문에 보안 수준은 향상되지만, 계산에 필요한 시간과 자원도 증가한다. TRLWE 샘플의 N 역시 다항식의 차수가 높을수록 복잡한 연산을 효과적으로 처리할 수 있으나, 처리 시간과 메모리 사용량이 증가한다. 또한, N이 클수록 보안 수준도 일반적으로 향상된다.

<그림 4> n에 따른 처리 속도와 보안 수준의 변화



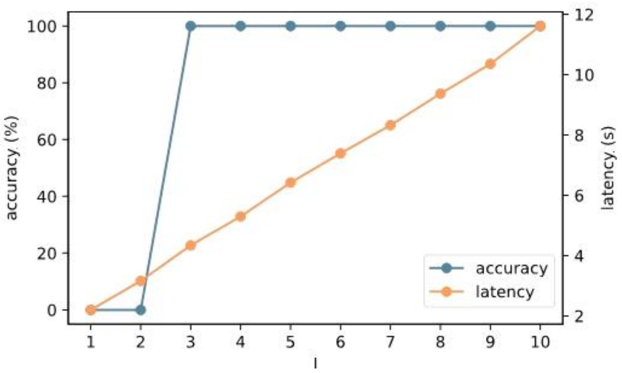
정확도와 보안 수준의 trade-off. TLWE 샘플의 α는 암호화된 데이터의 정확도와 보안성을 직접적으로 조절한다. α가 작을수록 노이즈가 줄어들어 복호화 시 데이터의 정확도가 향상되지만, 너무 낮은 값은 공격자가 오류를 분석하기 쉬워져 보안 위험이 증가할 수 있다. TRLWE의 α 역시 TLWE와 같은 trade-off가 적용되며, α를 적절히 조절함으로써 보안과 정확도 사이의 균형을 맞출 수 있다.

<그림 5> α에 따른 정확도와 보안 수준의 변화



정확도와 처리 속도의 trade-off. TRGSW 샘플의 l이 크면 부트스트래핑이나 복잡한 논리 연산을 수행할 때 세밀한 제어가 가능해지나, 연산 속도가 느려진다.

<그림 6> l에 따른 정확도와 처리 속도의 변화



4. 결론

본 논문에서는 TFHE(Fast Homomorphic Encryption over the Torus)의 다양한 파라미터가 동형암호 연산의 보안 수준, 정확도, 그리고 처리 속도에 미치는 영향을 면밀히 분석하였다. 연구 결과, TLWE, TRLWE, TRGSW 샘플들이 각기 다른 방식으로 동형암호 연산에 참여하며, 특히 파라미터 설정이 정확도, 처리 속도와 같은 성능과 보안 수준에 결정적인 영향을 미치는 것을 확인할 수 있었다. 이는 파라미터 선택에 있어 보안 수준과 성능 사이의 균형을 적절히 맞추는 것이 필수적임을 시사한다. 또한, 본 논문은 파라미터 설정에 대한 심도 있는 가이드라인을 제공하여, 다양한 애플리케이션에 적합한 동형 암호 솔루션 개발에 기여할 것으로 기대된다. 앞으로의 연구에서는 더 다양한 파라미터 조합과 최적화 방법을 모색하여, TFHE의 성능을 한층 더 향상시킬 수 있는 방안을 탐구할 예정이다. 이러한 연구 결과는 TFHE 기술의 실용적인 적용 범위를 넓히고, 보다 효율적이고 안전한 정보 처리가 가능하게 함으로써 동형암호 기술의 발전에 중요한 역할을 할 것이다.

5. ACKNOWLEDGEMENT

이 논문은 2024년도 BK21 FOUR 정보기술 미래인재 교육연구단에 의하여 지원되었음. 이 논문은 2024년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No.RS-2023-00277060, 개방형 엣지 AI 반도체 설계 및 SW 플랫폼 기술개발). 이 논문은 2024년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (RS-2023-00277326). 이 논문은 2024년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (IITP-2023-RS-2023-00256081).

참고문헌

[1] Cheon, Jung Hee, et al. "Homomorphic encryption for arithmetic of approximate numbers." Advances in Cryptology – ASIACRYPT 2017: 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part I 23. Springer International Publishing, 2017.

[2] Chen, Hao, Kim Laine, and Rachel Player. "Simple encrypted arithmetic library-SEAL v2. 1." Financial Cryptography and Data Security: FC 2017 International Workshops, WAHC, BITCOIN, VOTING, WTSC, and TA, Sliema, Malta, April 7, 2017, Revised Selected Papers 21. Springer International Publishing, 2017.

[3] Lee, Eunsang, et al. "Low-complexity deep convolutional neural networks on fully homomorphic encryption using multiplexed parallel convolutions." International Conference on Machine Learning. PMLR, 2022.

[4] Chillotti, Ilaria, et al. "Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds." Advances in Cryptology – ASIACRYPT 2016: 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part I 22. Springer Berlin Heidelberg, 2016.

[5] Chillotti, Ilaria, et al. "TFHE: fast fully homomorphic encryption over the torus." Journal of Cryptology 33.1 (2020): 34-91.

[6] https://cdn.fhe.org/slides/tfhe_deep_dive_ilaria_chillotti.pdf

[7] Klemsa, Jakub. "TFHE Parameter Setup for Effective and Error-Free Neural Network Prediction on Encrypted Data." Intelligent Computing: Proceedings of the 2021 Computing Conference, Volume 3. Springer International Publishing, 2021.

[8] Bergerat, Loris, et al. "Parameter optimization and larger precision for (T) FHE." Journal of Cryptology 36.3 (2023): 28.

[9] https://github.com/malb/lattice-estimator/