클라우드 컴퓨팅 환경에서의 동형암호기술 적용에 대한 연구

장지원*, 남기빈*, 조명현*, 백윤흥*

*서울대학교 전기·정보공학부, 반도체공동연구소

jwchang@sor.snu.ac.kr, kvnam@sor.snu.ac.kr, mhcho@sor.snu.ac.kr, ypaek@snu.ac.kr

A Study on the Applying Fully Homomorphic Encryption in the Cloud Computing Environment

Jiwon Chang*, Kevin Nam*, Myunghyun Cho*, and Yunheung Paek*

*Dept. of Electrical and Computer Engineering and Inter-University Semiconductor Research Center (ISRC),

Seoul National University

요 약

클라우드가 보편적으로 활용되면서 클라우드 서버에 정보를 저장하거나 연산을 하는 일은 일상이 되었다. 그러나, 이러한 클라우드 컴퓨팅 서비스가 급격히 증가하면서, 개인정보보호와 데이터 보안성, 기밀성 및 시스템의 안정성에 대한 우려가 높아지고 있다. 클라우드는 데이터를 위탁받아 연산하는 과정에서 사용자들의 개인정보를 유출시킬 수 있는 문제점이 있다. 이러한 문제점을 해결하기 위한 방법 중 현재 가장 각광받고 있는 해결책은 바로 동형암호기술이다. 동형암호는 이전 암호체계와 다르게 사용자의 암호화된 데이터를 복호화하지 않고서도 연산할 수 있어서, 이를 이용하게 되면 사용자 데이터의 기밀성을 보장하면서도 원하는 결과를 얻을 수 있다. 그러나, 동형암호를 클라우드 컴퓨팅 환경에 적용하는데 가장 큰 장애물은 바로 연산 오버헤드가 대단히 크다는 점이다. 본 연구에서는 최신 동형암호 기술을 소개하고 연산속도를 증가시키기 위한 솔루션들에 대해 알아보고자 한다.

I. 서론

4차 산업 혁명 시대를 맞이하며 인공지능, IoT, 빅데이터, 클라우드는 현대 컴퓨팅 환경에서의 대표적인 연산 주체이자 객체가 되어가고 있다. 이들은 데이터의 활용 및 분석을 통해 여러 응용 분야에 적용되고 있다. 최근 급부상하고 있는 엣지 컴퓨팅과도 연관이 큰데, 이러한 컴퓨팅 환경이 이루어지기 위해서는 원격 연산 기술이 필요하다. 빅데이터를 관리하기 위해 데이터 센터를 가진 많은 기업들은 데이터 센터에 수집된 빅데이터와 개인으로부터 제공받은 데이터를 기반으로 클라우드와 엣지 노드들을 통해 사용자들에게 편리한 서비스를 제공하고 있다. 이러한 빅데이터 시대에는 수많은 데이터들이 이곳저곳을 오가며 처리되고 있다. 때문에, 계속해서 더 많은 기업들이 클라우드 컴퓨팅 환경을 통해 고객의 데이터를 수집하여 서비스를 제공하려 할 것이다. 클라우드 컴퓨팅 환경은 원격으로 연산이 이루어져 개인이 처리할 수 있는 데이터양보다 훨씬 더 많은 방대한 양의 데이터를 효율적이고 경제적으로 처리할 수 있다.

그러나, 클라우드 컴퓨팅 환경은 여러 문제점[1]을 지니고 있는데, 개인정보와 같은 민감한 데이터에 대한 보안성이 취약하다는 가장 큰 문제를 갖고 있다. 대량의 데이터 수집 및 처리

과정이 클라우드 내에서 이루어지기 위해서 사용자로부터 전달받은 데이터를 암호화하지 않은 상태에서 연산을 수행해야 한다, 이러한 과정에서 클라우드 내의 민감한 데이터의 변조 및 유출 위험성은 상시 존재한다. 즉, 내부자 위협(Insider Threat)이나 부채널 공격(Side-channel Attack)에 취약한 클라우드 컴퓨팅 환경을 사용자들이 직접 통제하거나 감시할 수 없기 때문에, 클라우드 서비스를 전적으로 믿어야 한다. 이러한 상황이 계속해서 증가하므로, 최근 EU에서는 GDPR(General Data Protection Regulation), 미국 캘리포니아 주에서는 CCPA(California Consumer Privacy Act) 소비자 프라이버시 보호법을 제정하는 등 국제적으로 개인정보보호에 대한 관심이 증대되고 있는 추세이다.

이러한 추세에 발맞추어 안전한 클라우드 컴퓨팅 환경을 제공하기 위한 많은 연구가 이루어지고 있다. 현재 가장 많이 적용되는 기술은 바로 Intel과 ARM에서 사용하고 있는 신뢰실행환경(Trusted Execution Environment) 기술이다. 가장 대표적인 예로, Intel 6세대 프로세서인 '스카이레이크'에서부터 포함된 Intel SGX는 Enclave 라는 격리된 신뢰실행환경을 하드웨어적으로 제공하여 소프트웨어적으로 구축된 보안 기법보다 더 강력한 보호 환경을 제공한다. 하지만, Intel SGX가 캐시 부채널 공격에 취약하다[2]는 연구결과가 발표되었다.

부채널 공격뿐만 아니라 내부자 위협에도 데이터의 안정성을 보장하는 기술은 바로 차세대 암호체계인 동형암호(Homomorphic Encryption) 기술이다. 동형암호는 데이터를 암호화한 상태에서도 복호화 없이 연산이 가능 암호기술로 양자컴퓨터 시대에도 안전한 암호기술이다.

동형암호는 클라우드 서비스에 대한 어떠한 신뢰 가정 없이 데이터 보안을 보장한다. 클라우드 서비스는 단순히 계산능력만 사용자들에게 제공하고 데이터에 대한 정보는 암호화되어 있어 알 수 없다. 때문에, 동형암호는 프라이버시를 보장하는 클라우드의 솔루션으로 적용하기에 가장 적합한 기술이다. 예를 들어, MLaaS(Machine Learning as a Service)에서 동형암호 기술은 Oblivious 뉴럴 네트워크 추론에 적용될 수 있다. 사용자들은 개인의 데이터를 암호화하여 클라우드로 전송하고, 클라우드 서버에서는 암호화된 데이터들을 암호화된 채로 머신러닝 모델을 이용한 연산을 통해 결과값을 다시 사용자들에게 보내준다. 이때 모든 중간 및 최종 결과에 대한 데이터 값들은 암호화되어있고 오직 비밀키를 가지고 있는 사용자에 의해서만 복호화될 수 있다.

하지만, 동형암호기술을 실제 클라우드 컴퓨팅에 상용화하기에는 아직 연산속도가 일반연산보다 50만배 이상 느리다는 문제점이 존재한다. 현재 이러한 동형암호의 연산 오버헤드를 줄이기 위해, Software 최적화, Hardware를 이용한 연산 가속, 새로운 scheme 제시 등 많은 연구가 이루어지고 있다.

본 연구에서는 동형암호에 대한 기본적인 설명과 연산속도를 빠르게 하기 위한 연구들을 소개하며 향후 연구 방향을 제시하고자 한다.

II. 동형암호기술 소개

기존 암호화 기술은 데이터가 암호화된 상태에서 연산, 탐색, 분석 등의 작업이 불가능하였다. 반면에, 1978년 Rivest, Adleman and Dertouzos[3]에 의해 처음 제안된 동형암호는 암호화된 상태에서도 데이터의 연산이 가능하여 이상적인 암호 기술로 인식되었으나, 안전성 문제를 해결하지 못한 채 30여 년간 미제로 남았었다. 2009년 Gentry[4]가 암호문에 대한 임의의 연산이 가능하고 안전성을 보장하는 동형암호가 제안되었고, 이를 토대로 많은 연구가 발전되고 있다.

Gentry가 제안한 동형암호는 암호문에 대해 제한된 횟수의 연산만을 수행할 수 있었는데, 이는 암호문에 대한 연산이 수행된 후 암호문 안에 존재하는 노이즈(noise)가 켜져 일정 노이즈 임계치를 넘게 되면 암호문을 평문으로 복호화가 불가능해지기 때문이다. 이를 제한동형암호(SHE, Somewhat Homomorphic

Encryption)라고 부른다. 이와 달리, 무제한으로 암호문에 대한 임의의 연산이 가능한 암호를 완전동형암호(FHE, Fully Homomorphic Encryption)라 부른다. 완전동형암호는 연산이 수행된 후 암호문에 대한 노이즈를 줄이는 재부팅(Bootstrapping) 과정을 통해 무한히 암호문에 대한 연산을 노이즈 임계치를 넘지 않으면서 수행한다.

이러한 완전동형암호를 바탕으로 발전된 여러 방안(Scheme) 중 가장 대표적인 3가지 방안은 BGV[5], BFV[6], CKKS(HEAAN)[7] 이다. 이들의 가장 큰 차이점은 연산 가능한 수의 범위이다. BGV와 BFV 방안은 오직 정수에 대한 동형암호연산이 가능하기에 많은 제약이 존재한다. 이러한 제약에서 벗어날 수 있도록, 2017년 실수에 대한 동형암호연산이 가능한 CKKS 방안이 우리나라에서 최초로 제안되었다. 이전까지는 많은 동형암호 라이브러리들이 주로 BFV 방안을 활용하여 개발되었는데, CKKS의 등장 이후로 CKKS 방안을 채택한 라이브러리들이 다수 등장하고 있으며, 많은 연구진들이 CKKS 방안을 활용한 연구를 진행 중이다. 대표적인 예로 Microsoft Research에서 개발 중인 SEAL[8] 동형암호 라이브러리는 최근에 CKKS 방안을 주로 제공하며 BFV 방안 같은 경우는 선택적으로 사용할 수 있도록 제공하고 있다.

앞서 소개한 것과 같이, 완전동형암호는 여러 연구를 통해 많은 성능향상을 일궈냈다. 하지만, 여전히 소프트웨어로만 구현한 동형암호는 실용화하기에 연산속도가 너무 느리다. 이 때문에 최근에는 하드웨어를 도입하여 동형암호의 연산 오버헤드를 감소시키기 위한 많은 연구가 진행되고 있다.

III. 하드웨어 기반 동형암호 가속기

동형암호기술이 많은 발전을 이뤘음에도 불구하고 여전히 연산 오버헤드가 상당하다. 소프트웨어 구현만으로는 한계가 존재하기 때문에, 최근 많은 연구진들은 GPU나 FPGA와 같은 하드웨어를 사용하여 동형암호의 연산속도를 몇백배 이상 줄일 수 있었다. 하드웨어 가속기는 병렬처리에 강하며 에너지 효율성이 높기 때문에 이와 같은 성능 개선을 이뤄낼 수 있었다. 하지만, 대부분의 연구들은 시뮬레이션을 통한 결과들뿐이다. 실제 하드웨어를 구현한 연구는 추가적인 블록을 설계한다거나 코어들의 동기화를 고려해야 하는 등 굉장히 복잡하고 도전적인 작업이 필요하다.

Operation type	Message	Ciphertext	Slowdown
Addition	2.1 ns	348.2 ns	168.2×
Multiplication	4.3 ns	155883.8 ns	36112.7×

<표 1> 동형암호 연산 속도 비교[9]

<표 1>에서 볼 수 있듯이, 동형암호연산에서 가장 오래 걸리는 연산은 암호문 간의 곱셈 연산이다. 그래서 대부분의 연구진들은 이 동형암호 곱셈 연산을 가속화 하는 방안을 연구하고 있다. 가장 대표적인 3가지 예를 확인해보면, 먼저 2019년 발표된 [10]에서는 FPGA를 활용하여 동형암호 곱셈 연산을 블록 단위의 파이프라이닝과 병렬처리를 통해 가속화 하였다. 하지만, BFV 방안을 사용하여서 실수에 대한 연산이 어려우며, 하나의 고정된 동형암호 파라미터에 대한 연구로 확장성이 부족하다.

반면, 2020년 Microsoft Research에서 발표한 [11]에서는 CKKS 방안을 채택하여 실수에 대한 연산이 가능하며, FPGA를 활용해 이전보다 많은 단계의 파이프라이닝과 모듈화를 통해 동형암호 곱셈 연산을 가속화 하였으며, 뿐만 아니라, 여러 동형암호 파라미터 세트를 제공하여 이전 연구보다 더 높은 병렬성과 확장성을 보였다.

마지막으로 [12] 연구에서는 GPU를 활용하여 동형암호기술을 뉴럴 네트워크 연산에 적용하여 가속화 하였다. 이 연구에서는 BFV 방안의 동형암호를 적용한 합성곱(CNN) 연산을 통해 MNIST 데이터 집합과 CIFAR-10 데이터 집합을 높은 성능으로 분류해내었다. MNIST의 경우 99%의 높은 정확도를 보이지만, CIFAR-10의 경우 77.55%의 낮은 정확도를 보여 앞으로 더 많은 연구가 필요하다.

IV. 결론

본 논문에서는 동형암호에 대한 기본적인 개념과 클라우드 컴퓨팅 환경에 적용되기 위한 여러 가지 연구에 대하여 알아보았다. 차세대 암호체계인 동형암호에 관한 연구는 앞으로도 많은 연구가 필요한 분야이며, 뉴럴 네트워크 연산을 포함하면서 하드웨어 가속기를 이용하는 연구가 향후 이어질 것으로 본다.

V. ACKNOWLEDGEMENT

이 논문은 2020년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구(NRF-2017R1A2A1A17069478)이며, 2020년도 두뇌한국21플러스사업에 의하여 지원되었음. 또한, 2020년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No.2018-0-00230, (IoT 총괄/1세부) IoT 디바이스 자율 신뢰보장 기술 및 글로벌 표준기반 IoT 통합보안 오픈 플랫폼 기술개발 [TrusThingz 프로젝트]).

[참고문헌]

[1] T. Dillon, C. Wu, and E. Chang, “Cloud Computing: Issues and Challenges,” in IEEE International Conference on Advanced Information Networking and Applications, 2010.

[2] O. Oleksenko, B. Trach, R. Krahn, M. Silberstein, and C. Fetzer. Varys: Protecting SGX enclaves from practical side-channel attacks. In 2018 USENIX Annual Technical Conference (USENIX ATC), 2018.

[3] R. L. Rivest, L. Adleman, and M. L. Dertouzos, “On Data Banks and Privacy Homomorphisms,” Foundations of Secure Computation, vol. 4, no. 11, 1978.

[4] C. Gentry, “Fully homomorphic encryption using ideal lattices,” in Proceedings of the 41st ACM Symposium on Theory of Computing (STOC 2009), pp. 169 - 178, 2009.

[5] Z. Brakerski, C. Gentry, and V. Vaikuntanathan. “Fully homomorphic encryption without bootstrapping.” In Innovations in Theoretical Computer Science (ITCS’12), 2012

[6] J. Fan and F. Vercauteren, “Somewhat Practical Fully Homomorphic Encryption,” The International Association for Cryptologic Research Cryptology ePrint Archive, vol. 2012, 2012

[7] J. Cheon, A. Kim, M. Kim, and Y. Song, “Homomorphic Encryption for Arithmetic of Approximate Numbers,” in International Conference on the Theory and Application of Cryptology and Information Security, 2017

[8] SEAL 2020. Microsoft SEAL (release 3.5.0). https://github.com/Microsof/SEAL. Microsoft Research, Redmond, WA.

[9] W. Jung, E. Lee, S. Kim, K. Lee, N. Kim, C. Min, J. Cheon and J. Ahn, “HEAAN Demystified: Accelerating Fully Homomorphic Encryption Through Architecture-centric Analysis and Optimization,” arXiv:2003.04510, 2020

[10] S. S. Roy, F. Turan, K. Jarvinen, F. Vercauteren, and I. Verbauwhede, “FPGA-Based High-Performance Parallel Architecture for Homomorphic Computing on Encrypted Data,” in IEEE International Symposium on High Performance Computer Architecture (HPCA), 2019.

[11] M. S. Riazi, K. Laine, B. Pelton, and W. Dai, “HEAX: HighPerformance Architecture for Computation on Homomorphically Encrypted Data in the Cloud,” arXiv:1909.09731, 2019

[12] A. A. Badawi, J. Chao, J. Lin, C. F. Mun, S. J. Jie, B. H. M. Tan, X. Nan, K. M. M. Aung, and V. R. Chandrasekhar, “The AlexNet moment for homomorphic encryption: HCNN, the first homomorphic CNN on encrypted data with GPUs,” arXiv:1811.00778, 2018.