배타적 프라이버시 보호 기술을 활용한 효율적인 동형 암호 연산 기술

이동주 [1], 백윤흥 [1]
[1] 서울대학교 전기정보공학부, 반도체공동연구소

djlee@sor.snu.ac.kr, ypaek@snu.ac.kr

Efficient Homomorphic Encryption Operations Utilizing Exclusive Privacy Preserving Technique

Dong-Ju Lee[1], Yun-Heung Paek[1]
[1]Dept. of Electrical and Computer Engineering and Inter-University Semiconductor Research Center(ISRC), Seoul National University

요 약

클라우드 컴퓨팅 서비스를 사용하기 위해 사용자가 데이터를 클라우드로 전송하는 과정에서 프라이버시 문제가 발생할 수 있다. 이를 해결하기 위해 동형암호를 적용한 프라이버시 보호 원격 컴퓨팅 기술이 연구되고 있다. 하지만 동형암호 연산은 큰 성능 부하가 발생하며, 본 논문은 특정 연산에 대하여 배타적 프라이버시 보호기술을 적용한 효율적인 동형암호 연산 기술을 제안한다.

1. 서론

현재 많이 사용되는 ChatGPT 나 여타 인공지능 기반 분석 서비스들은 막대한 컴퓨팅 리소스를 필요로 한다. 하지만 사용자가 로컬 디바이스에서 이러한 서비스를 직접 실행하기 위해서는 상당한 비용이 필요하다. 그래서 OpenAI, 구글, 아마존 등의 기업은 클라우드 컴퓨팅을 통해 이러한 리소스를 원격으로 이용할 수 있도록 제공한다. 결과적으로 사용자는 상대적으로 적은 비용으로 다양한 양질의 서비스를 이용할 수 있게 된다. 그러나 이러한 원격 서비스를 이용함에 있어서, 사용자는 자신의 데이터를 클라우드로 전송해야 하며 기밀이나 사적으로 민감한 데이터를 다뤄야하는 서비스의 경우 프라이버시 문제가 발생할 수 있다. 이를 방지하기 위해 동형암호(**H**omomorphic **E**ncryption), 다자간 연산(**M**ulti-**P**arty **C**omputation), 신뢰실행환경(**T**rusted **E**xecution **E**nvironment) 등 다양한 프라이버시 보호기법을 적용한 프라이버시 보호 원격 컴퓨팅(**P**rivacy **P**reserving **C**loud **C**omputing) 기술이 활발히 연구되고 있다[1], [2], [3].

2. HE-based PPCC

HE 는 데이터의 복호화 없이 암호문간 연산이 가능한 4 세대 암호화 기법이다. HE 를 활용한 PPCC 의 대략적인 개요는 다음과 같다. ①사용자는 HE 로 데이터를 암호화하여 클라우드에 전송한다. ②클라우드는 암호화된 데이터를 입력으로 받아 해당 서비스에 대한 연산을 수행한다. ③클라우드는 연산결과를 사용자에게 보내고 사용자는 데이터를 복호화 하여 원하는 결과를 얻는다.

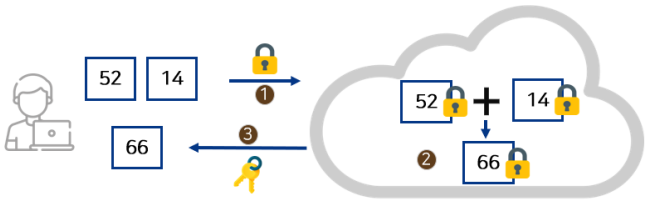

그림 1. 클라우드를 활용한 HE-based PPCC

하지만 HE 기반 PPCC 는 구현함에 있어서 문제점이 존재한다. HE 특성상 암호문의 크기가 커서 연산에 많은 시간이 소요되는데, If 문과 같이 암호화된 데이터에 대한 조건이 필요한 조건문의 경우 평문 연산과 다르게 모든 분기에 대해 연산해야 한다. 예를 들어 $x \leq 3$ 인 경우 x-1 을 연산하고 $x>3$ 인 경우 x+1 을 출력하는 함수의 경우, (조건 1: $x \leq 3$)의 경우 1 을 출력하고 (조건 2: $x>3$)의 경우 0 을 출력하는 Boolean 함수를 적용하고 x-1 을 연산하며 반대도 마찬가지로 적용하여 암호문을 더해주는 방식이다[4]. 이러한 일련의 과정으로 인해, 분기가 많아질수록 연산 부하는

매우 커지게 된다.

3. 배타적 프라이버시 보호 기술을 활용한 효율적인 동형암호 연산 기술

이러한 문제를 해결하기 위하여 본 논문은 HE PPCC 의 중간 연산 과정을 배타적 프라이버시 보호 기술을 적용하여 클라우드의 프라이버시를 침해하지 않게끔 사용자에게 보내 연산하는 방법을 제안한다. 하지만 사용자에게 암호문을 다시 전송하는 과정에서 통신 오버헤드가 발생하며 이를 해결하기 위해 HE 암호문대신 비교적 가벼운 암호체계로 암호화하여 전송하고 이를 HE 암호문으로 변형하여 연산하는 Transciphering 기술[5]을 사용할 수 있으며, 혹은 클라우드 내에 사용자의 TEE 를 구성하여 HE PPCC 의 중간 연산 과정을 클라우드 내 TEE 에 전송 후 연산하여 통신 오버헤드를 줄일 수 있다[6],[7]. TEE 는 클라우드와 분리된 물리 공간에서 사용자가 직접 자신의 코드를 단독적으로 수행할 수 있다. 다만 주목할 점은 이 연산 처리 방식이 기존과 다르게 클라우드 서비스의 일부 연산 코드가 사용자에게 노출된다는 점이다. 클라우드 서비스 모델은 클라우드의 중요한 자산이며 프라이버시 이므로, 사용자의 프라이버시를 보호하기 위해 클라우드의 프라이버시를 침해할 수 있는 상황이 발생할 수 있는 것이다. 우리는 이를 고려하여 클라우드의 프라이버시를 침해하지 않는 조건에 한하여 연산을 오프로딩 하는 방법을 소개하려 한다.

3-1 TEE 단독 연산의 한계

오프로딩 연산에 TEE 를 활용하는 경우, 모든 연산을 TEE 내부에서 처리할 수도 있지만, 사실상 그렇지 않다. 사용자에게 코드 정보를 모두 공개해야 한다는 점을 차치하더라도 TEE 기반의 연산기술은 부채널 공격에 취약하다고 알려져 있다. 운영 체제 수준의 공격자는 페이지 폴트를 유도하고, 액세스된 비트 및 캐시를 모니터링하며, 메모리 접근 시간을 확인하고 다른 방법을 통해 보호된 사용자 데이터의 내용을 추론할 수 있다[8], [9]. 또한 메모리의 한계로 인하여 메모리 이상의 연산을 처리하는 경우 부하가 발생해 성능이 매우 떨어진다[10]. 예를 들어 TEE 중 하나인 Intel 의 SGX 의 경우 메모리가 128MB 이하이다. 따라서 모든 연산을 TEE 내부에서 처리하기엔 보안 및 성능적인 한계가 존재한다.

3-2 배타적 프라이버시 보호 기술을 활용한 효율적인 동형암호 연산 기술

3-1 에서 기술했던 것과 같이, 리소스의 한계 때문에 HE 기반 PPCC 의 모든 연산을 TEE 로 오프로딩해 연산할 수 없다. 따라서 분기문과 같이 동형암호의 특성상 비 효율적으로 많은 시간을 필요로 하는 연산에 대해서 Transciphering 기술 혹은 TEE 를 활용하여 오프로딩해 연산하는 방식을 채택한다. 예를 들어 머신러닝 알고리즘 중 의사결정나무(Decision Tree) 알고리즘의 경우 그 특성상 수많은 비교 분기문을 포함하고 있다. 기존 연구들은 해당 알고리즘을 HE 를 적용하여 PPCC 로 효율적으로 연산하기 위한 방법을 제시하였다[11],[12]. 하지만 여전히 모든 leaf node 에 대한 경로에 대해 연산하여야 한다는 점 때문에 평문 연산에 비해 많은 시간이 소요된다.

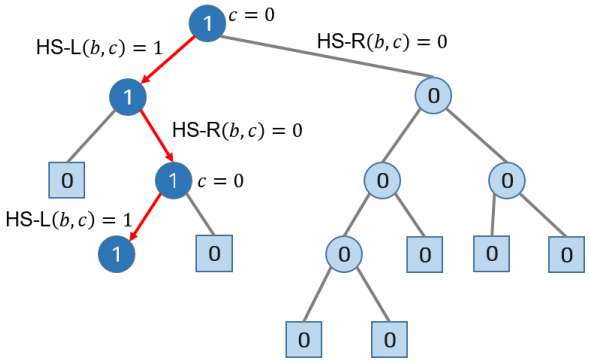

그림 2 HE 를 활용한 의사결정나무 알고리즘 예시[8]

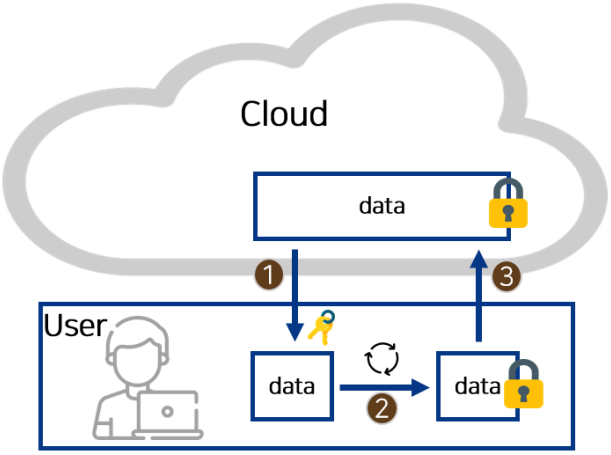

그림 3 배타적 프라이버시 보호기술을 활용한 동형암호 연산 예시

이러한 분기 연산에 대해 클라우드 서비스는 프라이버시가 침해되지 않는 부분의 연산에 필요한 코드와 입력값을 사용자에게 전송하고, 사용자는 입력값을 복호화한다. 하지만 클라우드가 코드와 입력값을 전송하게 된다면, 사용자는 클라우드의 온전한 코드와 해당코드의 입력 값을 알 수 있고, 이를 악용하여 클라우드의 서비스를 추출해낼 수 있다. ①따라서 클라우드는 입력값을 전송하기 전에 해당 암호문에 rotation 함수와 곱셈함수를 적용하여 입력값의 순서를 뒤바꾸어 보내게 된다. 현재 널리 활용되는 대표적인 HE 스킴인 CKKS 의 경우 입력데이터에 대한 SIMD(Single Instruction Multiple Data) 연산이 가능한데, 하나의 암호문에 여러 데이터를 담아 각각에 대한 병

렬 연산을 수행한다. Rotation 함수를 적용하면 데이터 간의 순서가 cyclic 하게 밀리는 구조이다. 이는 비트 간 shift 연산과 유사하며 예를들어 그림 4 와 같이 Rotation 2 함수를 적용하는 경우 데이터가 왼쪽으로 두 칸 만큼 이동하는 방식이다. 데이터의 순서를 바꾸는 예시로 6 개의 데이터 중 첫 번째부터 세 번째 데이터에 1, 나머지 데이터에 0 을 곱하고 Rotation 3 을 적용한 암호문과, 마찬가지로 뒤의 세 개의 데이터에도 동일한 방식을 적용한 암호문을 더하게 되면, 앞의 세 개와 뒤의 세 개의 데이터의 순서가 바뀐 암호문을 생성할 수 있다.

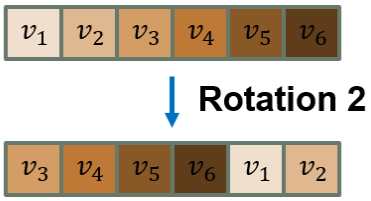

그림 4 CKKS 암호문 rotation 예시

②사용자는 클라우드에게 받은 데이터를 복호화 하여 순서가 뒤바뀐 데이터를 연산한다. 사용자는 데이터의 순서가 바뀌었고 그 경우의 수가 매우 크기 때문에(데이터의 Slot 이 8192 인 경우 8192! 의 경우의 수) 클라우드가 본래 연산하려던 원래의 값을 역 추론하기 어렵다. ③사용자는 TEE 내부에서 병렬적으로 연산한 값을 다시 암호화하여 클라우드에 전송하고, 클라우드는 원래 순서에 맞게 역으로 함수를 적용하여 되돌린다. 데이터가 다시 암호화되기 때문에 클라우드는 사용자가 어떤 분기를 거쳐 연산했는 지 알 수 없다. 이를 통해 사용자와 클라우드 모두 각자의 프라이버시를 배타적으로 보호할 수 있으며, 효율적인 분기 연산이 가능하다.

4. 결론

클라우드 컴퓨팅 서비스를 이용하기 위하여 사용자는 클라우드에 데이터를 전송해야 하며 이로 인해 사용자의 데이터 프라이버시 문제가 대두되고 있다. 이를 해결하기 위해 프라이버시 보호 원격 컴퓨팅 기술이(PPCC) 활발히 연구되고 있으며, 본 논문은 동형암호를 적용한 PPCC 에 대하여 배타적 프라이버시 보호 기술을 적용한 효율적인 연산 방식을 제안한다. 클라우드는 tranciphering 기술을 활용하거나, 신뢰실행환경(TEE)을 구성하여, 사용자에게 서비스의 코드와 순서가 바뀐 입력 값을 전송하고, 사용자는 해당 값을 복호화 한다. 사용자는 데이터를 연산하고, 다시 암호화하여 클라우드에 전송한다. 클라우드는 데이터를 원래 순서로 다시 되돌려 연산을 지속한다. 이 과정에서 사용자는 입력 값의 순서가 바뀌어 클라우드가 본래 연산했어야 할 값을 역 추론할 수 없으며, 클라우드 역시 데이터가 암호화되어 수신되기 때문에 사용자의 데이터를 알 수 없다. 이를 통해 상호 프라이버시를 보호하면서도 효율적인 분기 연산이 가능하다. 사용자가 복호화 및 암호화를 진행함으로써 자연스럽게 암호문의 노이즈가 초기화되어 Bootstrapping(암호문 재부팅)의 효과 역시 있을 것으로 기대되며, 이에 대한 성능 향상은 제안한 방식의 부가적인 이점이라 할 수 있다.

5. ACKNOWLEDGEMENT

이 논문은 2023 년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구이며 (IITP-2023-RS-2023-00256081), 2024 년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구이며 (RS-2023-00277326), 2024 년도 BK21 FOUR 정보기술 미래인재 교육연구단에 의하여 지원되었음. 본 연구는 반도체 공동연구소 지원의 결과물임을 밝힙니다. 이 연구를 위해 연구장비를 지원하고 공간을 제공한 서울대학교 컴퓨터 연구소에 감사드립니다.

참고문헌

[1] Gilad-Bachrach, Ran, et al. "Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy." International conference on machine learning. PMLR, 2016.

[2] Juvekar, Chiraag, Vinod Vaikuntanathan, and Anantha Chandrakasan. "{GAZELLE}: A low latency framework for secure neural network inference." 27th USENIX security symposium (USENIX security 18). 2018.

[3] Lee, Junghyun, et al. "Precise approximation of convolutional neural networks for homomorphically encrypted data." IEEE Access (2023).

[4] Cheon, Jung Hee, et al. "Numerical method for comparison on homomorphically encrypted numbers." International conference on the theory and application of cryptology and information security. Cham: Springer International Publishing, 2019.

[5] Cho, Jihoon, et al. "Transciphering framework for approximate homomorphic encryption." *International Conference on the Theory and Application of Cryptology and Information Security*. Cham: Springer International Publishing, 2021.

[6] Wang, Wenhao, et al. "Toward scalable fully homomorphic encryption through light trusted computing assistance." *arXiv preprint arXiv:1905.07766* (2019).

[7] Drucker, Nir, and Shay Gueron. "Achieving trustworthy Homomorphic Encryption by combining it with a Trusted Execution Environment." *J. Wirel. Mob. Networks Ubiquitous Comput. Dependable Appl.* 9.1 (2018): 86-99.

[8] Y. Xu, W. Cui, and M. Peinado, “Controlled-channel attacks: Deterministic side channels for untrusted operating systems,” in IEEE Symposium on Security and Privacy (SP). IEEE, 2015, pp. 640–656.
[9] W. Wang, G. Chen, X. Pan, Y. Zhang, X. Wang, V. Bindschaedler, H. Tang, and C. A. Gunter, “Leaky cauldron on the dark land: Understanding memory side-channel hazards in sgx,” in Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. ACM, 2017, pp. 2421–2434.
[10] M. Taassori, A. Shafiee, and R. Balasubramonian, “Vault: Reducing paging overheads in sgx with efficient integrity verification structures,” in Proceedings of the Twenty-Third International Conference on Architectural Support for Programming Languages and Operating Systems. ACM, 2018, pp. 665–678.
[11] Cong, Kelong, et al. "Sortinghat: Efficient private decision tree evaluation via homomorphic encryption and transciphering." *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*. 2022.
[12] Frery, Jordan, et al. "Privacy-Preserving Tree-Based Inference with Fully Homomorphic Encryption." *Cryptology ePrint Archive* (2023).