Journal of the The Korean Institute of Defense Technology Vol. 02, No. 1, pp. 005-008, 2020

국방 클라우드 컴퓨팅 암호화 효율적 구현 방안 연구

A study on efficient implementation method of defense cloud computing encryption

양현상*,1)

HyeonSang Yang*,1)

[초 록]

본 연구는 클라우드 컴퓨팅 암호화를 위해 동형암호에 관해 기존 문헌들의 연구 결과를 토대로 동형암호의 장단점에 대한 분석을 근거하여 동형암호 적용방법을 통해 효율적인 구현을 위한 방안을 제시하고자 한다. 이를 위해 동형암호를 IT 기술 구현 및 표준화 하려는 방법이 필요하다. 본 연구에서는 동형암호 개념 이해하고 동형암호 구현을 위한 장·단점을 분석하고 효율적 구현 방법을 소개하고자 한다.

[ABSTRACT]

This study aims to propose a method for efficient implementation through a method of applying isomorphic encryption based on the analysis of the strengths and weaknesses of isomorphic encryption based on the research results of existing literature on isotype encryption for cloud computing encryption. For this, it is necessary to implement and standardize homomorphic cryptography in IT technology. In this study, we intend to understand the concept of homomorphic cryptography, analyze the advantages and disadvantages of implementing homomorphic cryptography, and introduce efficient implementation methods.

Key Words : 클라우드 컴퓨팅(cloud computing), 암호화(encryption), 완전동형암호(fully homomorphic encryption), 동형암호(homomorphic encryption)

1. 서 론

최근 다양한 IT 기기의 사용이 확산하면서 클라우드 컴퓨팅이 보편화 되고 있다. 클라우드 컴퓨팅은 정보가 인터넷 서버에 영구적으로 저장하고 컴퓨터와 같은 IT 기기 등과 같은 클라이언트에는 임시 보관되는 환경을 의미한다.

사용자의 정보를 인터넷 서버에 저장하고, 각종 IT 장비를 통해서 언제든지 이용할 수 있게 하는 것이다.

클라우드 컴퓨팅을 도입하여 컴퓨터 시스템을 관리하기 위한 장비 구매, 소프트웨어 비용 등 큰 비용과 시간을 줄일 수 있다는 장점이 있다. 하지만 서비스를 제공하는 앞장에서는 제한 없이 접근할 수 있어야 하고, 서버에 저장된 자료가 있으므로 해커 등에 취약할 수밖에 없다.

1) 한화시스템 사업기획팀/고려사이버대 국방기술융합학과 겸임교수
(Hanwha System Business Division/Double Professor, Department of Defense Covergence Technology, The Cyber University of Korea)
* Corresponding author, E-mail: yhs10386@naver.com

Received: January 30, 2020 Revised:
Accepted: February 18, 2020

이런 취약점들을 보완하기 위해 사용자 스스로가 자료를 암호화해서 서버에 저장하는 것이 좋은 것이다. 이러면 암호화된 자료를 찾거나 자료를 통계 처리할 수밖에 없다.

또한, 클라우드 컴퓨팅 환경에서 데이터 처리는 상당수 서버에서 이루어지기 때문에 일반적으로 암호를 이용하여 클라우드 서버에서 사전에 복호화하는 과정이 있어야 한다. 그러기 위해서는 사전 복호화를 해야 한다.

이때 복호화하는 과정에서 임시로 클라우드 서버에 복호화된 자료가 임의로 저장하거나 외부로 유출될 수 있는 위험이 그대로 있다.[1]

최근에 빈번하게 발생하고 있는 정보 유출이 난 공공기관 보안 기술이 개인정보 보호하는 데 충분하지 않다는 것을 알 수 있다.

모든 개인정보는 암호화하고 인원은 인가를 받고 출입하는 등 접근 통제를 하도록 개인정보 보호법에 있고 이를 준수하게 되어 있다.

비밀키를 하여 유출 등을 예방해야 하고, 데이터베이스를 엄호한 상태로 통계적 처리를 하지 않았거나 식별이 어려운 암호체계를 필요하게 된다.

이에 따라 안정성을 보장하면서 원하는 기능을 다양하게 제공할 수 있는 효율적인 완전동형암호 기술 연구가 필요하다.

2. 관련 연구

본 연구는 완전동형암호에 관해 기존 문헌들의 연구 결과를 토대로 동형암호의 장단점에 대한 분석을 근거하여 동형암호 적용방법을 통해 효율적인 구현을 위한 검증 하고자 한다. 또한, 동형암호를 IT 기술 구현 및 표준화하려는 방법이 필요하다. 그래서 본 연구에서는 정확한 동형암호 개념 이해하고 완전동형암호 구현을 위한 장단점을 분석하고 효율적 구현 방법을 소개하고자 한다.

2.1 완전동형암호의 이해

2.1.1 동형암호 원리

'동형(Homomorphic)'이란, '같음(Same)' 뜻하는 고대 그리스어인, ὁμός(Homos)와 "모양(Shape)이나 형체(form)"를 뜻하는 μορφή(morphe) 의 결합에서 유래되었다. 한 걸음 더 나아가서, '동형'이라는 단어는 대수학(Algebra)에서 비롯된 단어로, 연산의 구조가 유지되는 함수를 가리킨다. 예를 들면, 지수 함수인 f(x) = ex는 동형함수입니다.

왜냐하면, ex+y = ex * ey 로서, 지수 함수는 양의 실수군에 대해 동형인 함수다. 그렇다면 '동형암호(Homomorphic Encryption)'는 무엇일까요? '동형(同形)암호'란 평문에 대한 연산을 수행한 후 암호화한 결과(암호문)와 각각의 암호문에 대하여 연산을 수행한 결과가 같은 값을 가지는 암호화 방식을 말한다. 시저(Ceasar) 암호를 가지고 동형암호 개념을 설명해 보겠다. 시저 암호는 하나의 알파벳을 특정한 알파벳과 자리를 바꾸는 치환 암호다.

정확한 것은 문자열 연결(Concatenation)에 대하여 부분적으로 동형이다. 13자리 알파벳과 자리를 바꾸는 시저 암호를 가지고 동형암호의 원리를 설명하면 다음과 같다.[2]

Encrypt(13, 'A') = 'N', Encrypt(13,'B') = 'O'...,

Encrypt(13, 'H') = U

"HELLO"와 "WORLD"를 각각 암호화해 문자열 연결을 한 뒤, 이 결과("URYYBJBE YQ")를 복호화 하면, "HELLOWORLD"를 얻을 수 있다.

```
var c1 = Encrypt(13, "HELLO");  // c1 = URYYB
var c2 = Encrypt(13, "WORLD"); // c2 = JBEYQ
var c3 = Concat (c1, c2); // c3 = URYYBJBEYQ
var c4 = Encrypt(13, "HELLO"||"WORLD")
            // c4 = URYYBJBEYQ
var p  = Decrypt(13, c3); // p  = HELLOWORLD
```

한편 동형암호는 중대한 약점을 갖고 있다. 동형암호는 일정 회수 이상 연산을 수행하면 잡신호가 발생해 더 연산을 수행할 수 없어 연산 횟수에 제한이 있다.

암호문의 잡신호를 작게 하는 재부팅(Boot strapping) 과정을 통해 동형암호의 약점인 연산 횟수 제한을 없앨 수 있는 암호화 방식을 '완전동형암호(Fully Homomorphic Encryption)'라고 한다.[4]

2.1.2 완동형암호 메시지 특성

완전동형암호를 덧셈 연산하고 이해하면 완전동형암호는 메시지를 특정한 두 개의 수로 나누어 나머지 두 개를 상대방에게 보내는 방식이다. 아래 예시에서 사용하는 평문은 10과 15이고, 사용하는 키는 4와 7입니다. 이때 사용한 연산은 덧셈 연산이다. 암호화는 모듈러 연산(mod4, mod7)이다.

10과 15의 덧셈 연산 결과를 암호화한 것과 평문인 10과 15를 각각 암호화한 후 덧셈 연산한 결과가 같은지 확인하면 된다. 예를 들어 메시지 10은 4와 7로 각각 나눠 나온 나머지 2와 3으로 암호화한다. 여기서 10과 15를 암호화해서 (2, 3)과 (3, 1) 이렇게 두 암호를 받았다고 가정한다.

두 암호 결과를 덧셈 연산한다. 같은 수로 나눈 나머지끼리 더해야 하므로 2는 3과 3은 1하고만 더할 수 있다. 그러면 암호 결과를 연산한 값은 (5, 4)가 된다.[4]

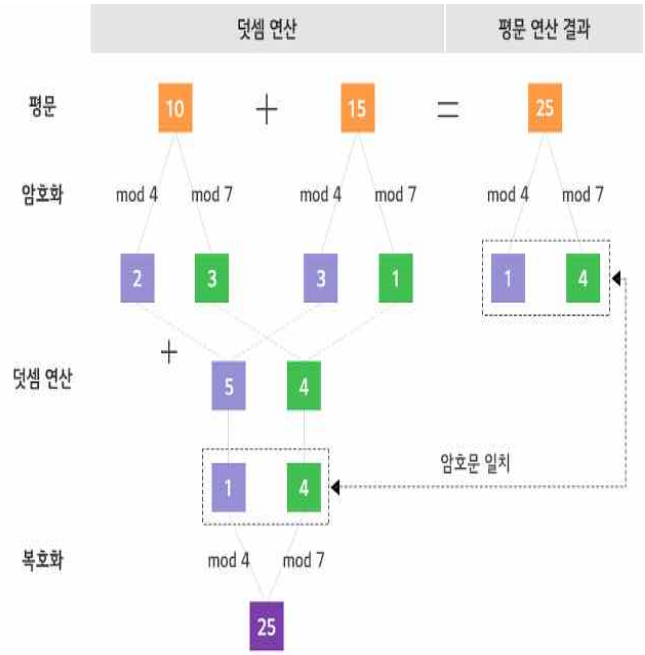

그림 1. 완전동형암호 예시

Fig. 1. Full homomorphic password example

2.1.3 완동형암호 복호화

(5, 4)를 복호화 해면 연산된 암호를 풀기 위해선 암호화할 당시 사용한 키를 알면 된다. 여기서 사용한 키는 4와 7이다. 그런데 5는 4로 한 번 더 나눠진다. 따라서 1이라고 생각해야 한다. 결국, 4로 나눴을 때 나머지가 1이고, 7로 나눴을 때 나머지가 4인 수를 구하면 되는데, 그 값은 25이다. 그런데 이 값은 원래 메시지인 10과 15를 더한 값과 같다. 따라서 이런

원리를 이용하면 암호가 걸린 상태에서도 통계적인 분석이 가능하게 되는 것이다.

2.2 동형암호 적용

2.2.1 동형암호 적용 활용

동형암호는 이미 1978년에 발표되었다. PKI(Public Key Infrastructure)의 핵심인 공개키 암호 알고리즘 RSA와 같은 해에 발표되었으며, RSA의 두 개 이니셜 'R'과 'A'의 주인공 Rivest, Addleman이 Dertouzous3과 함께 발표한 암호 알고리즘이다.

이론적으로 안전성이 증명되지 않아 오랫동안 관심에서 멀어져 있다가, 2009년 Gentry4(현재 IBM Thomas J. Watson Research Center 소속)에 의해 안전성이 증명된 동형암호가 발표되고, 2011년 MIT가 동형암호 기술을 10대 Emerging Technology로 선정하면서 학계와 산업계의 관심이 더욱 촉발되었다. 그렇다면 동형암호 기술이 최근 더욱 이슈화되고 있는 이유가 무엇일까요? 동형암호는 암호화된 데이터의 복호화 없이 암호문의 연산이 가능하다.

이러한 특징에 기반을 둔 완전동형암호의 장점은 암호화 후 통계 처리뿐만 아니라 검색, 기계 학습까지 가능하다는 점이다.[5]

특히, 해커가 데이터를 유출해도 볼 수 없다는 점에서 안전성이 주목받고 있다. 더욱이 기존 동형암호의 문제점인 평문 연산 대비 암호문의 연산 속도가 현저히 느린 점도 연산의 종류에 따라 최적화하는 기술이 발전되고 있다.

특히 의료계나 금융권은 오래전부터 쌓아온 방대한 데이터를 기반으로 유전자 정보 기반 정밀 치료나 신용 정보의 안전한 활용 등과 같은 새로운 기회 발굴에 활용하고자 하지만, 민감 정보를 포함한 개인정보의 빅데이터 활용에 대한 규제와 기술적인 어려움이 큰다는 점을 작용하여 전환점을 찾지 못하고 있다. 인공지능을 활용한 데이터 분석을 연구하는 학계와 이를 활용하려는 산업계가 다 같이 이러한 답답한 상황을 해결할 수 있는 도구로서, 동형암호 기술을 주목하게 된 것이다.

2.3 동형암호 기술 구현 방안

이외에도 표준화는 미국 국립보건원, 미국 MIT, 스위스 EPFL, 삼성, Gemplus, 인텔, SAP 등 정부, 학교, 스타트업 및 IT 기업 등이 포함되어 있다. 보안 업체가 주도한 기존 암호 기술과는 달리 동형암호는 데이터 분석 업체가 주도하고 있다.

MS는 인공지능(AI) 보호를 위해 동형암호를 채택했으며, IBM은 향후 5년간 연구 분야로 동형암호 기술을 꼽았다.

이와 함께 ENVIL, Cryptolab 등의 스타트업 기업들도 동형암호에 관심을 보이고 있다.[6]

2.3.1 동형암호의 장점과 단점

동형암호는 암호문을 복호화하지 않아도 검색, 통계 처리 및 기계 학습이 가능하고, 데이터를 처리하는 중간 과정에서 복호화하지 않아도 되므로, 데이터 유출 위험이 감소하는 장점이 있다.

단점은 기존 암호의 확장률(평문 대비 암호문이 커지는 비율)에 비해 10배에서 100배 정도 커질 수 있고, 암복호화 속도가 1ms인 RSA 대비 수십 ms가 소요되며, 암호문 곱셈 연산의 경우, 수백 ms가 소요된다. 한편 앞에서 언급했던 노이즈 감소를 위한 재부팅 시간이 필수적으로 필요하며, 2015년 기준 재부팅 시간이 0.02초 정도로 보고되었다.

2.3.2 복호화 없이 활용 가능한 완전동형암호

성능이나 소요 자원 측면에서 개선되어야 할 부분이 있지만, 개인정보를 처리하는 과정에서 복호화 없이 활용 가능한 장점을 활용한 완전동형암호의 활용 예시는 다음과 같다. 첫 번째 활용 분야는 생체 인식 분야다. 출입 통제에 사용하는 생체 정보를 안전하게 처리할 수 있다.

완전동형암호화된 지문 등의 생체 정보를 복호화하지 않고, 암호화된 채로 인증을 수행할 수 있다면, 생체 정보를 복호화해 평문으로 보관하거나, 검증을 위해 원본을 전송할 필요 없이 받은 암호문과 보관 중인 암호화된 생체 정보를 비교해 일치 여부를 확인할 수 있도록 할 수 있다.

완전동형암호 기술을 활용해 홍채 인증 기술을 개량 발전시켜 인증 시간을 0.25초로 앞당긴 시스템이 있다.[8]

두 번째 활용 분야는 금융 분야입니다. 완전동형암호는 컴퓨터가 할 수 있는 연산을 모두 수행할 수 있어 데이터의 기밀성을 보호하면서 금융 데이터를 동형 기계 학습의 훈련 단계에서 활용할 수 있다.

동형 기계 학습의 훈련 단계에서 암호화된 데이터 연산을 통해 예측•분류 모형을 얻고 그 모형으로부터 실시간 계산이 필요한 예측 단계에서는 함수 암호를 사용한다.

동형 기계 학습은 데이터의 기밀성을 보호할 수 있어 개인의 민감한 정보의 유출에 대한 보안 문제를 해결할 수 있으며 데이터의 손실 없이 신용도 평가 예측 모형의 고도화가 가능하다고 할 수 있다.

50만 명의 신용 데이터를 동형암호화된 상태에서 기계 학습을 수행해 개인정보를 보호하면서 신용 평가 모형의 신뢰성, 정확성, 안전성을 성공적으로 확보할 수 있음을 검증했다.[9]

3. 결 론

본 연구는 완전동형함호 구현을 위한 제한되는 동형함호의 구현을 연구하였다.

첫째, 완전동형암호화된 지문 등의 생체 정보를 복호화하지 않고, 암호화된 채로 인증을 수행할 수 있다면, 생체 정보를 복호화해 평문으로 보관하거나, 검증을 위해 원본을 전송할 필요 없이 받은 암호문과 보관 중인 암호화된 생체 정보를 비교

해 일치 여부를 확인할 수 있도록 할 수 있다.

둘째, 완전동형암호는 컴퓨터가 할 수 있는 연산을 모두 수행할 수 있어 데이터의 기밀성을 보호하면서 금융 데이터를 동형 기계 학습의 훈련 단계에서 활용할 수 있다.

따라서 본 연구를 통하여 완전동형함호 구현을 위한 제한되는 완전동형암호의 구현 방안을 살펴보았다. 완전동형암호를 효율적으로 구현할 수 있는 여러 알고리즘 설계 방식 및 구현 방법에 관한 연구에 많은 진전이 이루어지고 있다.

하지만, 아직 다양한 공격 시나리오에 대한 충분한 검증이 이루어지지 않았다. 키의 길이와 파라미터 크기를 확정하여 안전성을 평가하는 다양한 연구가 필요하다. 또한, 클라우드, 빅데이터 등 실제 응용 환경에서 어떻게 사용할 것인가에 대한 실제적인 적용에 관한 연구가 필요하다.

References

[1] T. Yu, S. Jajodia, Secure data management in decentralized systems, Springer Science and Business Media, vol. 33, pp. 355-380, Springer Press, 2007

[2] C. Gentry, "Fully homomorphic encryption using ideal lattices," Proceedings of the 41st ACM Symposium on Theory of Computing, pp.169-178, May 2009.

[3] A. Chatterjee, M. Kaushal, and I. Sengupta, "Accelerating sorting of fully homomorphic encrypted data," the 14th International Conference on Cryptology in India, LNCS 8250, pp.262-273, Dec. 2013.

[4] G.S. Cetin, Y. Doroz, B. Sunar, and E. Savas, "Low depth circuits for efficient homomorphic sorting," In IACR ePrint Arch. 2015/274, Mar. 2015

[5] J.H. Cheon, M. Kim, and M. Kim, "Optimized search-and-compute circuits and their application to query evaluation on encrypted data," IEEE Transactions on Information Forensics and Security, vol. 11, no. 1, pp. 188-199, Jan. 2016.

[6] 천정희,“개인정보가 보호되는 동형머신러닝” 제1회 가우스 콜로퀴움. 2018. 3.

[7] 천정희 외 2명, “개인정보가 보호되는 동형암호기반 금융 데이터분석” 한국금융정보학회. 2018 .2.

[8] 조은지 외 2명, “완전 동형암호 라이브러리의 성능 분석” 한국정보과학회. 2018 .2.

[9] 김주혁 외 1명, “최신 정보보호 이슈 및 국외 암호기술 연구 동향” 한국정보과학회. 2014. 10.

[10] 송유진, 박광용, “데이터베이스 아웃소싱을 위한 준동형성 암호기술”, 정보보호학회지, 제19권, 제3호, pp.80-89, 2009.

[11 김선주, 김진묵, 조인준, “모바일 클라우드 서비스 상에서 준동형 암호 기반의 형상 관리 방안”, 한국정보통신학회논문지, 제16권, 제10호,

[12] M. V. Dijk, C. Gentry, S. Halevi, and V.Vaikuntanathan, "fully homomorphic encryptionover the integers," Advances in Cryptology -EUROCRYPT, Lecture Notes in Computer Science, Vol.6110, pp.24-43, 2010.

[13] C. Gentry and S. Halevi, "Implementing gentry's fully homomorphic encryption scheme," Advances in Cryptology - EUROCRYPT, Lecture Notes in Computer Science, Vol.6632, pp.129-148, 2011.

[14] I. Chillotti, N. Gama, M. Georgieva, and M. Izabachène. Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds. pages 3-33, In Asiacrypt 2016.

[15] H. Chen, K. Laine, and R. Player, “Simple Encrypted Arithmetic Library - SEAL v2.1”, https://www.microsoft.com/en-us/research/wpcontent/uploads/2016/09/sealmanual-2.pdf,2017

[16] 서경진 , 김평 , 이윤호, “완전동형암호로 암호화된 데이터에 적합한 산술 가산기의 구현 및 성능향상에 관한 연구”. 정보보호학회논문지, 2017