LWE 기반의 순서 노출 암호화 기법

박재환, 한창희
서울과학기술대학교 전기정보공학과
reds98@seoultech.ac.kr, chahn@seoultech.ac.kr

Order-Revealing Encryption based on LWE

Jae Hwan Park, Changhee Hahn
Dept. of Electrical and Information Engineering, Seoul National University of Science and Technology

요 약

현재까지의 순서 노출 암호화 기법 (Order-Revealing Encryption, ORE) 중 안전성과 실용성을 동시에 만족시키는 기법은 Parmeter hiding ORE (18' ASIACRYPT)와 Efficient Multi-client ORE (21' ESORICS)이다. 하지만 두 기법은 이산 대수 문제를 기반으로 설계되었기 때문에 쇼어 알고리즘을 활용한 양자컴퓨터에 취약하다. 따라서 본 연구에서는 이러한 위협에 대비하고자 양자 컴퓨터에 내성을 갖는 Learning With Error (LWE) 문제를 기반으로 한 안전한 ORE 기법을 제안한다.

1. 서론

순서 노출 암호화 기법 (Order-Revealing Encryption, ORE)이란, 사용자가 서버에 저장해 둔 암호화된 데이터를 특정 함수를 이용하여 쿼리와 비교함으로써 순서를 노출시키는 기법이다. 현재까지의 ORE 기법 중 비교 단계에서 가장 적은 노출을 보이는 기법은 Cash et al. [1]과 Lv et al. [2]의 기법이다. 두 기법 [1,2]은 이산 대수 문제를 기반으로 기법을 구성하고 있다. 한편, 지속적인 양자 컴퓨터의 연구 및 상용화를 고려하여 암호화 기법 설계에 선제 대응이 필요하다. 하지만, 앞서 언급했듯이, 두 기법 [1,2]은 이산 대수 문제를 이용하여 기법을 구성하고 있는데 이는 양자 컴퓨팅 기반의 쇼어 알고리즘에 취약하다. 따라서, 본 연구에서는 양자 컴퓨터에 내성을 갖는 Learning with Error (LWE) 문제를 기반으로 ORE 기법을 제안한다. LWE는 곱해진 행렬 AS에 무작위로 뽑은 error를 더해줌으로써 행렬 T를 얻고, 공격자가 행렬 A와 T만이 주어졌을 때 행렬 S(비밀키)를 같은 크기의 무작위로 뽑은 행렬과 구별할 수 없다는 격자 암호 문제이다. 본 연구에서는 평문을 행렬 S로 활용하여 ORE 기법을 구성하였다. 본 논문에서 행렬은 A와 같이 대문자로 표기한다.

2. Property Preserving Hash

먼저 본 연구에 제시되는 ORE 기법에 기반이 되는 Property Preserving Hash (PPH)를 소개한다. PPH는 세 가지의 알고리즘으로 구성되어 있다: PPH.KeyGen, PPH.Hash, PPH.Test.

●PPH.KeyGen(1^{λ}, q, b): 이 알고리즘은 security parameter λ, modular 연산에 활용되는 q, 행렬의 크기를 결정하는 b를 입력으로 받고 역행렬이 존재하는 b×b 크기의 행렬 P와 D를 무작위로 뽑는다.

$$P \xleftarrow{\$} Z_q^{b\times b},\ D \xleftarrow{\$} Z_q^{b\times b} \quad (1)$$

그 뒤, 암호화 키 ek = (P, D, P^{-1}, D^{-1})과 테스트 키 tk = (P, D)를 반환한다.

●PPH.Hash(ek, m): 이 알고리즘은 암호화 키 ek와 메시지 m을 입력으로 받은 뒤 m을 이진법으로 변환하고 $1_{(2)}$를 더한 값과 뺀 값을 얻는다. 그 다음, 각각의 값들을 b×b 크기의 행렬로 변환하고 수식 (2)와 같이 hash 값 를 반환한다.

$$\vec{H} = (\overrightarrow{H_1} = D^{-1} + PM,\ \overrightarrow{H_2} = P^{-1} + M_{+1_{(2)}}D,\ \overrightarrow{H_3} = P^{-1} + M_{-1_{(2)}}D \quad (2)$$

●PPH.Test(tk, $\vec{H}$, $\overrightarrow{H'}$): 이 알고리즘은 두 개의 hash 값 $\vec{H}$, $\overrightarrow{H'}$과 tk를 입력으로 받아 수식 (3)을 연산한 뒤, $E_1 = E_2$ 또는 $E_1 = E_3$를 만족하면

1을 반환하고 아니면 0을 반환한다.

$$E_1 = \overrightarrow{H_1}D,\ E_2 = P\overrightarrow{H_2},\ E_3 = P\overrightarrow{H_3} \quad (3)$$

3. ORE based on LWE

본 연구에서 제시하는 ORE 기법에 대한 동작과정을 설명한다. ORE 기법은 다음의 네 가지 알고리즘으로 구성된다: ORE.KeyGen, ORE.Enc, ORE.Token, ORE.Comp. 함수 F는 유사 난수 생성기이다.

●ORE.KeyGen(1^{λ}, q, b): 이 알고리즘은 security parameter λ, modular 연산에 활용되는 q와 행렬의 크기를 결정하는 b를 입력으로 받고 PPH.KeyGen을 이용하여 ek = (P, D^{-1})와 hk = (P^{-1}, D)를 얻고 ek와 hk를 반환한다.

●ORE.Enc(ek, m): 이 알고리즘은 길이가 n인 이진법으로 표현된 메시지 m = (m_1, m_2, ..., m_n)과 ek를 입력으로 받아 수식 (4)와 같이 행렬 R를 정해진 크기에 알맞게 무작위로 뽑는다.

$$R \xleftarrow{\$} Z_q^{b\times b} \quad (4)$$

그 뒤, 모든 i = 1, ... , n에 대하여 수식 (5)를 연산한 뒤,

$$U_i = F(i, m_1 m_2 ... m_{i-1} \,||\, 0^{n-i+1}) + m_i \bmod 2^{\lambda},$$
$$V_i = \overrightarrow{H_1} \leftarrow PPH.Hash(ek, U_i) \quad (5)$$

집합 V = (V_1, V_2, ..., V_n)를 얻는다. 그 다음, 모든 i = 1, ... , n에 대하여 집합 V를 수식 (6)을 이용하여 연산한 뒤,

$$C_i = RV_i \quad (6)$$

집합 C = (C_1, C_2 ,..., C_n)과 ek = (RP, D^{-1})를 얻는다. 마지막으로 random permutation π : [n]→[n]를 이용하여 C_π = (RP, $C^{\pi}_{(1)}$, $C^{\pi}_{(2)}$,..., $C^{\pi}_{(n)}$)를 얻고 C_π를 반환한다.

●ORE.Token(hk, m'): 이 알고리즘은 메시지 m'과 hk를 입력으로 받아 수식 (7)과 같이 행렬 R_t를 정해진 크기에 알맞게 무작위로 뽑는다.

$$R_t \xleftarrow{\$} Z_q^{b\times b} \quad (7)$$

그 뒤, 모든 i = 1, ... , n에 대하여 수식 (8)을 연산한 뒤

$$U_i = F(i, m'_1 m'_2 ... m'_{i-1} \,||\, 0^{n-i+1}) + m'_i \bmod 2^{\lambda},$$
$$V_{i,1} = \overrightarrow{H_2} \leftarrow PPH.Hash(hk, U_i),\ V_{i,2} = \overrightarrow{H_3} \leftarrow PPH.Hash(hk, U_i) \quad (8)$$

집합 V_1 = ($V_{1,1}$,$V_{2,1}$,..., $V_{n,1}$)과 V_2 = ($V_{1,2}$,$V_{2,2}$,..., $V_{n,2}$)를 얻는다. 그 다음, 모든 i = 1, ... , n에 대하여 집합 V_1과 V_2를 수식 (9)를 이용하여 연산한 뒤 T = (($T_{(1,1)}$, $T_{(1,2)}$) ,..., ($T_{(n,1)}$, $T_{(n,2)}$))와 hk = (P^{-1}, D R_t)를 얻는다.

$$T_{i,1} = V_{i,1}R_t,\ T_{i,2} = V_{i,2}R_t \quad (9)$$

마지막으로 random permutation π : [n]→[n] 를 이용하여 T_π = (DR_t, ($T^{\pi}_{(1,1)}$, $T^{\pi}_{(1,2)}$) ,..., ($T^{\pi}_{(n,1)}$, $T^{\pi}_{(n,2)}$))를 얻고 T_π를 반환한다.

●ORE.Comp(C_π, T_π): 이 알고리즘은 c_π와 t_π를 입력으로 받아서 tk = (RP, DR_t)를 추출하고 ∀i, j ∈ [n]에 대하여 PPH.Test(tk, $C^{\pi}_{(i)}$, $t^{\pi}_{(j,1)}$)과 PPH.Test(tk, $C^{\pi}_{(i)}$, $T^{\pi}_{(j,2)}$)를 연산한다. 만약 ∀i, j ∈ [n]에 대하여 PPH.Test(tk, $C^{\pi}_{(i*)}$, $T^{\pi}_{(j*,1)}$)를 만족하는 i*과 j*이 존재한다면 1을 반환하는데, 이는 m > m'를 의미한다. 또한, PPH.Test(tk, $C^{\pi}_{(i*)}$, $T^{\pi}_{(j*,2)}$)를 만족하는 i*과 j*이 존재한다면 -1을 반환하는데, 이는 m < m'를 의미한다. 어떠한 경우에도 해당하지 않으면, 0을 반환하는데, 이는 m = m'를 의미한다.

제안기법은 LWE 문제를 이용하여 암호화하였다. C_π의 RP와 $C^{\pi}_{(i)}$에서 RP를 행렬 A로 $C^{\pi}_{(i)}$의 V_i를 행렬 S로 그리고 RD^{-1}를 error로 생각할 때, 이는 LWE 문제의 꼴과 같으므로 안전성을 보장하고 $T^{\pi}_{(i,1)}$과 $T^{\pi}_{(i,2)}$도 이와 유사하다. 또한, 행렬의 교환법칙은 항등함수가 아니면 성립하지 않는다는 조건을 이용하여 ORE.Enc과 ORE.Token으로부터 출력 되는 각각의 암호문들끼리는 서로 연산이 되지 않도록 설계하였고 probabilistic 특성 또한 만족한다.

4. 결론

본 연구에서는 쇼어 알고리즘을 활용한 양자 컴퓨터에 취약한 기존의 이산 대수 기반의 ORE 기법을 대신해 LWE 기반의 안전한 ORE 기법을 제시하였다. 또한 안전성을 분석을 통해 제안 기법이 양자 컴퓨터 환경에서 안전함을 보였다.

Acknowlegment

이 논문은 2023년도 교육부의 재원으로 한국연구재단 LINC 3.0 사업의 지원을 받아 수행된 연구임.

참고문헌

[1] Cash, David, et al. "Parameter-hiding order revealing encryption." ASIACRYPT, Brisbane, Australia, 2018, p. 181-210.

[2] Lv, Chunyang, et al. "Efficient Multi-client Order-Revealing Encryption and Its Applications." ESORICS: Darmstadt, Germany, 2021, p. 44-63.