격자 기반 암호 개요

(Lattice-Based Cryptography)

IT정보공학과 신명수

목차

1. Base
2. Lattice
3. Learning With Error (LWE)
4. Learning With Rounding (LWR)
5. LWE-based Encryption

PQC(Post Quantum Cryptography)의 필요성

- Shor 는 소인수분해 문제를 빠르게 처리할 수 있는 양자 알고리즘을 제안.
- 이러한 알고리즘이 적용가능한 양자컴퓨터가 개발되면 기존 암호화 시스템을 깨트릴 수 있으며, 현재 세계적으로 널리 쓰이고 있는 공개키 암호화 시스템인 RSA 또한 그 대상임.
- 양자컴퓨터의 계산능력에 내성을 가진 암호화 시스템이 필요하다.

주요 PQC 후보

- Lattice-Based : 격자 기반
- Code-Based : 부호 기반
- Hash-Based : 해쉬 기반
- Isogeny-Based : 아이소제니 기반
- Multivariate : 다변수 다항식 기반

2. Lattice

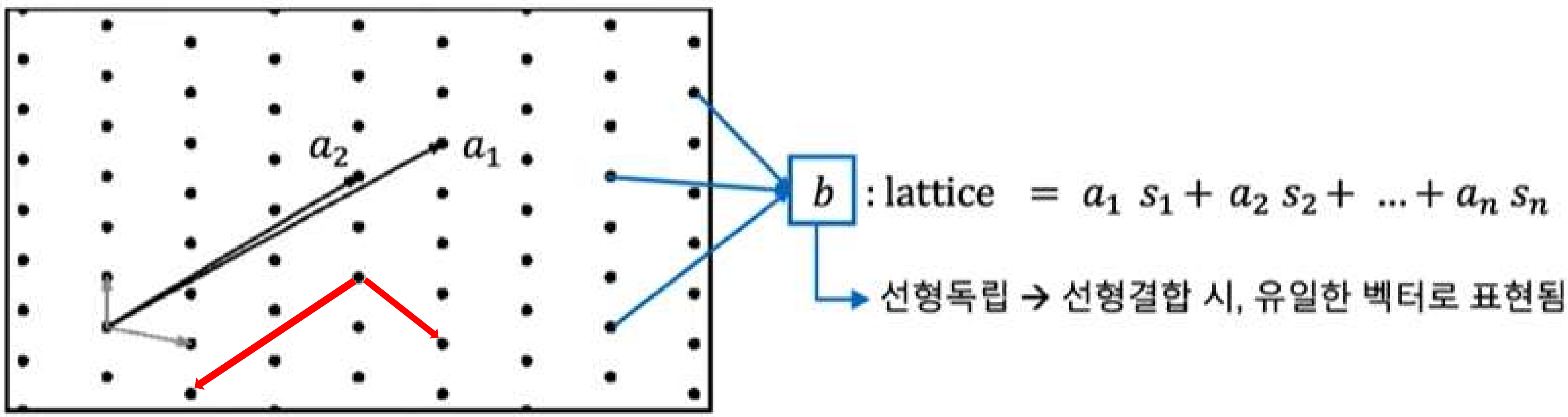
- $Lattice\ L = \{\sum_{i=1}^{n} a_i s_i \mid s_i \in Z\}$ $R^n\ basis : \{a_1, a_2, a_3 \dots \dots a_n\}$

  : R^n에서 정수 계수(s_i)를 갖는 모든 기저(basis)들의 선형결합
  = N차원 공간에 속하는 기저벡터들과 모든 정수의 선형조합.

- R^n의 이산적 덧셈 부분군 -> a_i로 vector space R^n과 L 생성.

- 기저(basis) : 모든 벡터들이 선형 독립인 n차원 벡터공간 내의 임의의 원소들을 표현하기 위해 필요한 최소한의 벡터.

1. Lattice



기저벡터들(a_i)이 선형결합되어 이루는 점들(b)의 집합

2. computational problems on lattices

- 고전적 격자 난제

 - SBP (Smallest basis problem) : 좋은 기저(=직교에 가장 가까운 기저)를 찾는 문제.

 - SVP (Shortest Vector Problem) : 격제 L이 주어졌을 때, 0이 아닌 최소 길이를 갖는 벡터를 찾는 문제. 차원이 클수록 찾기 어려우며 100차원 이하까지는 풀 수 있음.

 - CVP (Closest Vector Problem) : 격자 L과 한 점이 주어졌을 때, 그 점에서 가장 가까운 격자 벡터찾기. 둘러싸고 있는 벡터 2^n개의 후보를 갖고, 차원이 커질수록 찾기 어려움.

2. computational problems on lattices

- 거의 **직교인 기저**가 있으면 SVP, CVP 문제는 쉽게 풀린다.
  SVP -> 거의 직교인 기저라면, 기저벡터 중에 가장 짧은 벡터가 있다.
  CVP -> 정사영을 사용해 길이를 재면 가장 가까운 격자 벡터를 찾을 수 있다.
- 격자를 주려면 기저를 줘야하는데, 기저를 잘못주면 문제가 쉬워질 수 있음.
- 어떤 경우에 문제가 쉬워지는지 정확하게 측정하기 어려움.
- 매우 나쁜 기저를 줘도 한두번 연산으로 쉬운 기저가 나오기도 함.

3. Learning With Error (LWE)

- 최근의 난제 : WC = AC equiv(Worst Case Average Case equivalent)
  평균적인 케이스가 최악의 케이스와 동치 -> 어떤 상황에서도 난도를 유지.

- LWE (Learning With Error)
  - 현재 가장 널리 사용되는 난제
  - 작은 에러를 포함한 연립선형방정식의 해를 구하는 문제. (에러가 없으면 쉬운 문제)
  - A와 As + e 가 주어졌을 때, s를 구하는 문제.

3. Learning With Error (LWE)

- LWE (Learning With Error)
  - 현재 가장 널리 사용되는 난제
  - 작은 에러를 포함한 연립선형방정식의 해를 구하는 문제. (에러가 없으면 쉬운 문제)
  - A ($A \in \mathbb{Z}_p^{m \times n}$) 와 $b(= As + e)$ 가 주어졌을 때, s를 구하는 문제.

Mod 7

$$\begin{bmatrix} 0 & 5 & 2 & 3 \\ 1 & 3 & 6 & 9 \\ 3 & 0 & 8 & 5 \\ 4 & 7 & 9 & 3 \\ 1 & 0 & 6 & 5 \\ 4 & 9 & 2 & 7 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 6 \\ 1 \\ 0 \\ 5 \\ 2 \\ 3 \end{bmatrix}$$

라고 했을 때, $\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$ 찾기 쉽다.

3. Learning With Error (LWE)

- LWE (Learning With Error)
  - 현재 가장 널리 사용되는 난제
  - 작은 에러를 포함한 연립선형방정식의 해를 구하는 문제. (에러가 없으면 쉬운 문제)
  - A와 As + e 가 주어졌을 때, s를 구하는 문제. $e = \$\{-1, 0, 1\}$, $or\ e = \$GD(0, \sigma), (\sigma = 2.6)$

A　　　Noise vector e　　　Mod 7

$$\begin{bmatrix} 0 & 5 & 2 & 3 \\ 1 & 3 & 6 & 5 \\ 3 & 0 & 4 & 5 \\ 4 & 6 & 5 & 3 \\ 1 & 0 & 6 & 5 \\ 4 & 5 & 2 & 4 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 6 \\ 1 \\ 1 \\ 0 \\ 6 \end{bmatrix} = \begin{bmatrix} 6 \\ 1 \\ 0 \\ 5 \\ 2 \\ 3 \end{bmatrix} \qquad \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

찾기 어렵다.

3. Learning With Error (LWE)

- LWE (Learning With Error)
  - 현재 가장 널리 사용되는 난제
  - 작은 에러를 포함한 연립선형방정식의 해를 구하는 문제. (에러가 없으면 쉬운 문제)
  - A와 As + e 가 주어졌을 때, s를 구하는 문제. $e = \$\{-1, 0, 1\},\ or\ e = \$GD(0, \sigma), (\sigma = 2.6)$

- N이 100 이하일 때는 쉽지만, N이 200 이상일 때는 매우 어렵다.

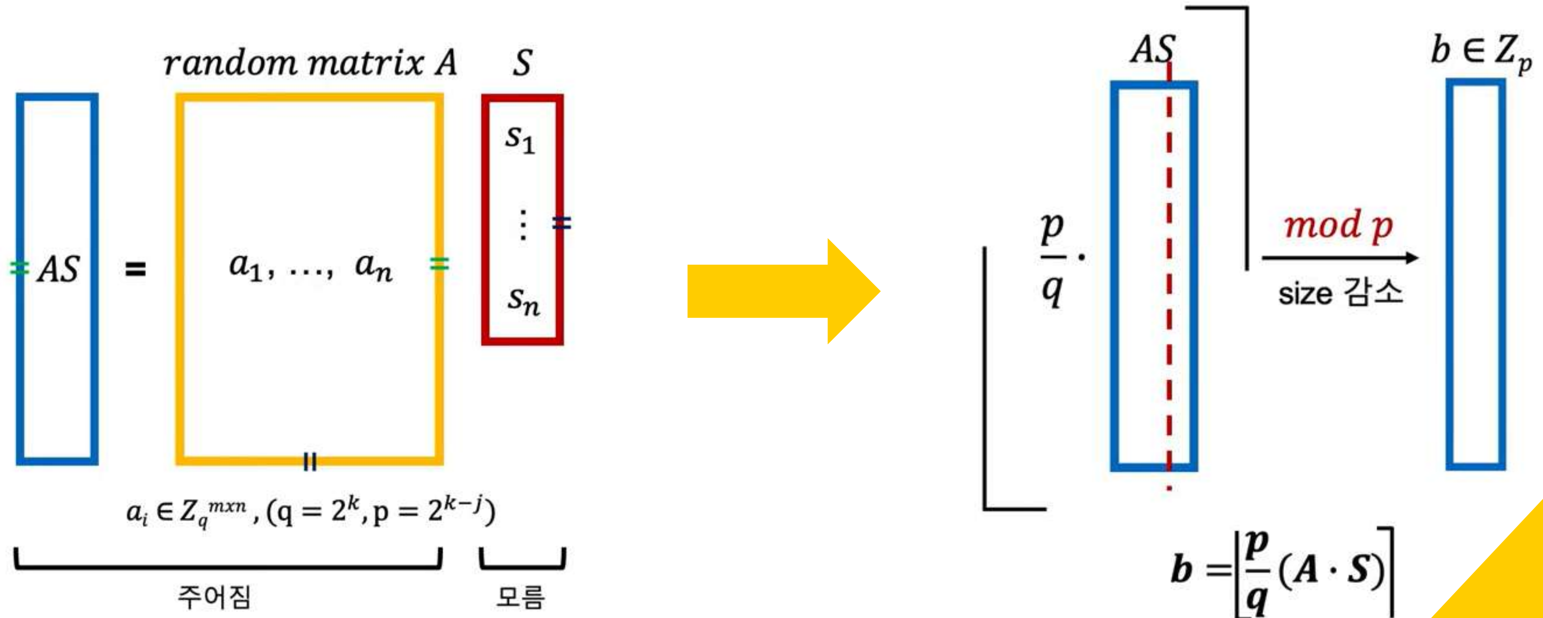
3. Learning With Error (LWE)

- m개의 sample (A, b)가 주어질 때, 계산된 LWE sample인지, random (A, u)인지 구분할 수 없게 됨.

- **$(\boldsymbol{A}, \boldsymbol{b} = \boldsymbol{A} \cdot \boldsymbol{S} + \boldsymbol{e} \ \boldsymbol{mod} \ \boldsymbol{q})$ 를 만족하는 $\boldsymbol{S}$가 존재하는지, random 선택된 것인지.**

4. Learning With Rounding (LWR)

- LWR sample인 (A, b)가 주어질 때, 비밀키 S 찾는 문제
  Rounding 을 통해 잘라내면 찾기 어려움
  - Rounding : 원래 값을 어느정도 유지하면서 자릿수를 원하는 만큼까지 줄이는 방법.



4. Learning With Rounding (LWR)

- **Rounding**

  q $\leq 2^{13}$: 1011100100111

  p $\leq 2^{10}$:1011100100

  -> 뒤쪽부터 (q-p) bit 만큼 잘라냄

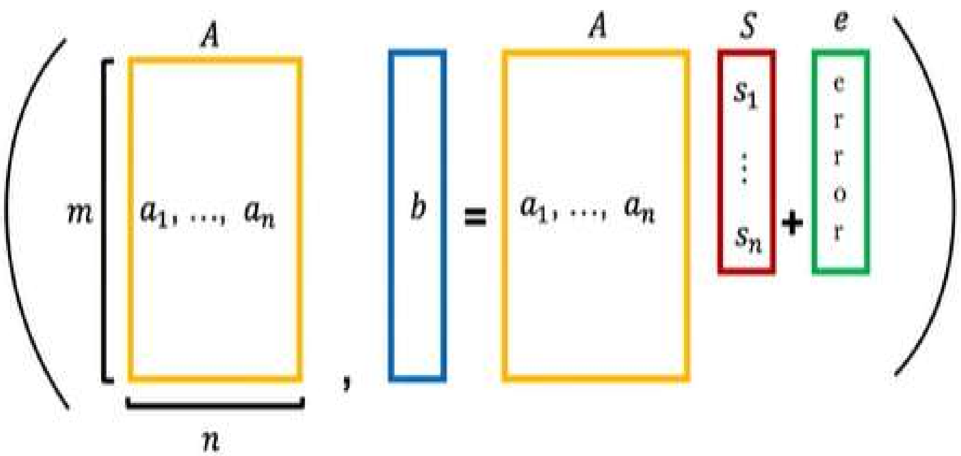
  많이 자를수록 S를 찾아내기 힘들지만, 복호화 시 오류 발생 가능성 증가

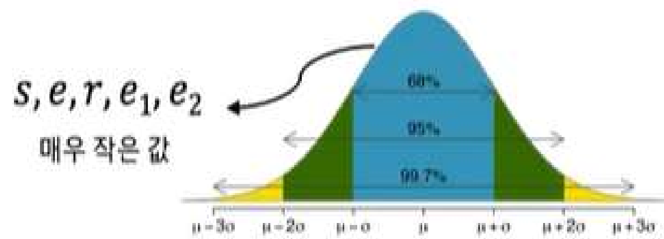
- m개의 sample (A, b)가 주어질 때, 계산된 LWR sample인지 mod p상에서의 random (A, u) 인지 구별하기 힘들다.

★ $\left(A, b = \left\lfloor \frac{p}{q}(A \cdot S) \right\rceil\right)$ **를 만족하는 S가 존재하는지, random 선택 된 것인지**

5. LWE-based Encryption

- Public key : $(A, b = As + e)$, secret key : S

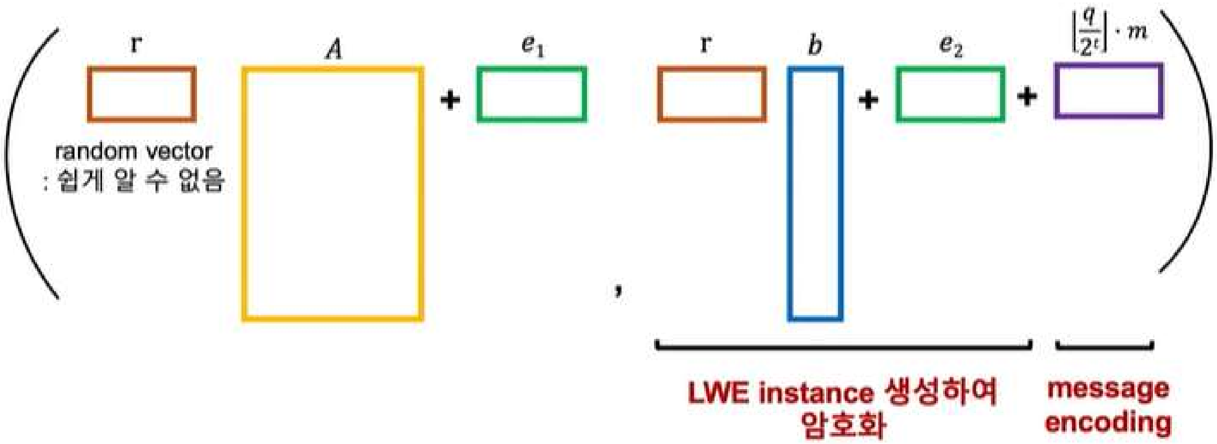




$*$ 모든 연산은 $mod\ q$

5. LWE-based Encryption

- Cipher text : $(C_1,\ C_2) = (r \cdot A + e_1, r \cdot b + e_2 + \left\lfloor \frac{q}{2^t} \right\rfloor \cdot m$



★ 송신자가 (C_1, C_2) 만들기 위해 가우시안분포에서 e_1, e_2를 뽑아서 사용 (message마다 새로 선택)

→ 수신자는 e_1, e_2 모름