Lattice-Based Cryptography

https://www.youtube.com/watch?v=4tWDiVu4lnU

한성대학교
HANSUNG UNIVERSITY

CryptoCraft LAB
https://crypto.modoo.at

Contents

CryptoCraft LAB

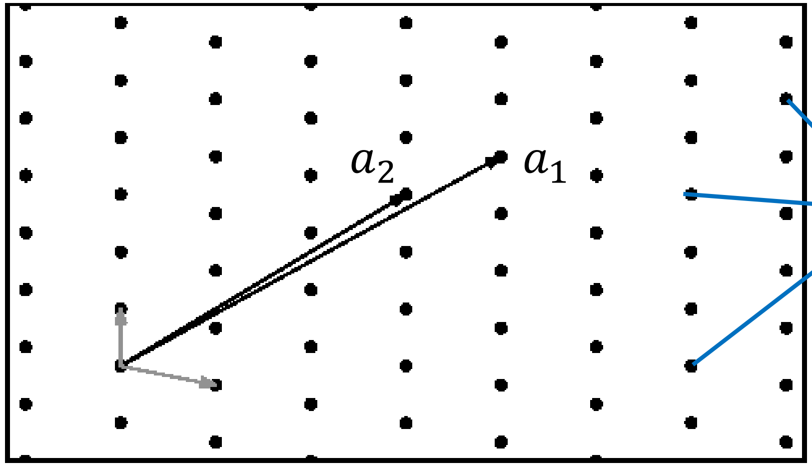
Lattice

$$\boldsymbol{Lattice\ L} = \left\{ \sum_{i=1}^{n} a_i\, s_i \,\middle|\, s_i \in Z \right\}$$

: **R**$^{\mathbf{n}}$에서 정수 계수(**S**$_{\mathbf{i}}$)를 갖는 모든 기저들(**a**$_{\mathbf{i}}$)의 선형 결합 **: R**$^{\mathbf{n}}$의 이산적 덧셈 부분군

$*\ \mathrm{R}^n$ 의 $basis: \{a_1, a_2, \dots, a_i\}$

→ a$_{\mathbf{i}}$ **로 vector space R**$^{\mathbf{n}}$ **과 L 생성**



기저벡터들(a_i)이 선형결합되어 이루는 점들(b)의 집합

$\boxed{b}$: lattice $= a_1\, s_1 + a_2\, s_2 + \dots + a_n\, s_n$

→ 선형독립 → 선형결합 시, 유일한 벡터로 표현됨

*기저(basis) : 모든 벡터들이 선형 독립인 n차원의 벡터공간 R^n 내의 임의의 원소들을 표현하기 위해 필요한 최소한의 벡터

→ 기저를 span하여 모든 벡터 생성 → vector space

computational problems on lattices

$matrix\ A$ $\quad S$

$$b = \begin{bmatrix} a_1, \ldots, a_n \end{bmatrix} \begin{bmatrix} s_1 \\ \vdots \\ s_n \end{bmatrix}$$

*1개의 seed로 생성 가능

$$b = A \cdot S$$

❖ A(L의 basis)가 주어지고, 선형결합 통해 A로 격자 위의 점들인 b를 생성할 때의 계수는 S_i

- **Shortest Vector Problem (SVP)**

  : 격자 위의 가장 짧은 0이 아닌 벡터는 무엇인가?

  : SVP가 어려우면 LWE도 어려움

- **Closest Vector Problem (CVP)**

  : 주어진 벡터와 가장 가까운 격자 위의 벡터는 무엇인가?

NP-hard

*NP 문제 : 비 결정론적 튜링머신으로 다항시간 내에 풀 수 있는 문제 → 시간이 오래 걸리는 문제

*모든 NP 문제들을 다른 어떤 문제인 A로 바꿀 수 있지만 해결할 수 없다면 NP-hard

Learning With Error (LWE) Problem

❖ m개의 sample (A, b) 이 주어질 때, 계산된 **LWE sample인지 random (A, u) 인지** 구분할 수 없게 됨

★ **$(A, b = A \cdot S + e \mod q)$ 를 만족하는 S가 존재하는지, random 선택 된 것인지**

Learning With Rounding (LWR) Problem

- **rounding**

  $q = 2^{13}$: 1011100100111

  $p = 2^{10}$: 1011100100

  → 뒤쪽부터 $(q - p)$ bit 만큼 잘라냄

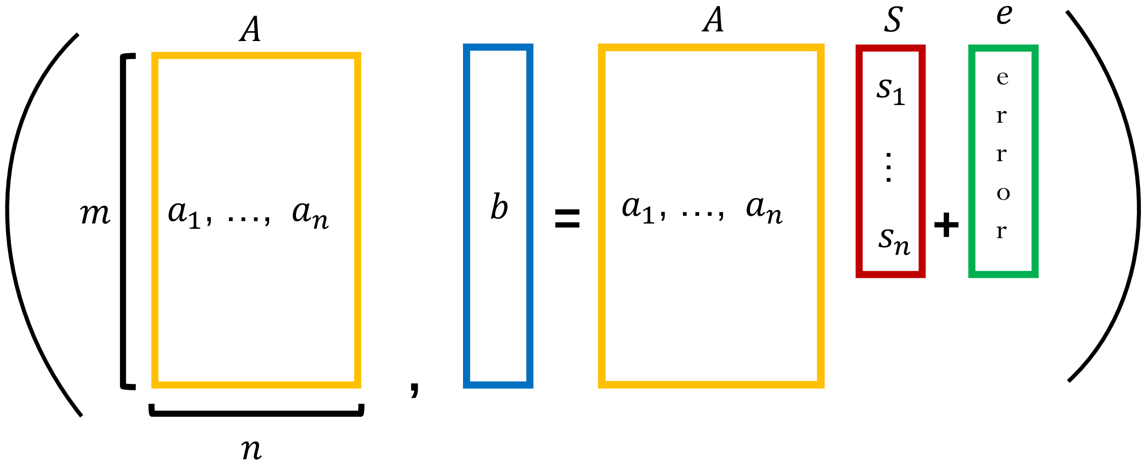
  ➢ **많이 자를수록 S를 찾아내기 어렵지만, 복호화 시 오류 발생 가능성이 증가**

  *modulus가 작을수록 효율성 증가, 오류율이 낮을수록 안전성 증가 → 보통 3bit

❖ m개의 sample (A, b) 이 주어질 때, 계산된 **LWR sample인지 mod p상에서의 random** (A, u) **인지 구분 불가**

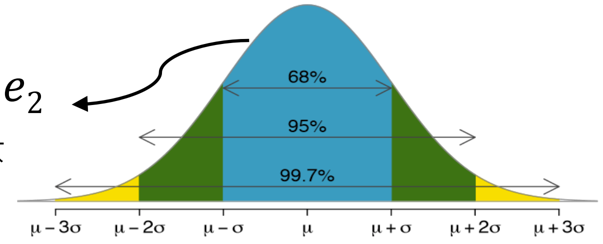
★ $\left(A, b = \left\lfloor \frac{p}{q}(A \cdot S) \right\rceil\right)$ **를 만족하는 S가 존재하는지, random 선택 된 것인지**

❖ public key : $(A, b = AS + e)$, secret key :S



s, e, r, e_1, e_2

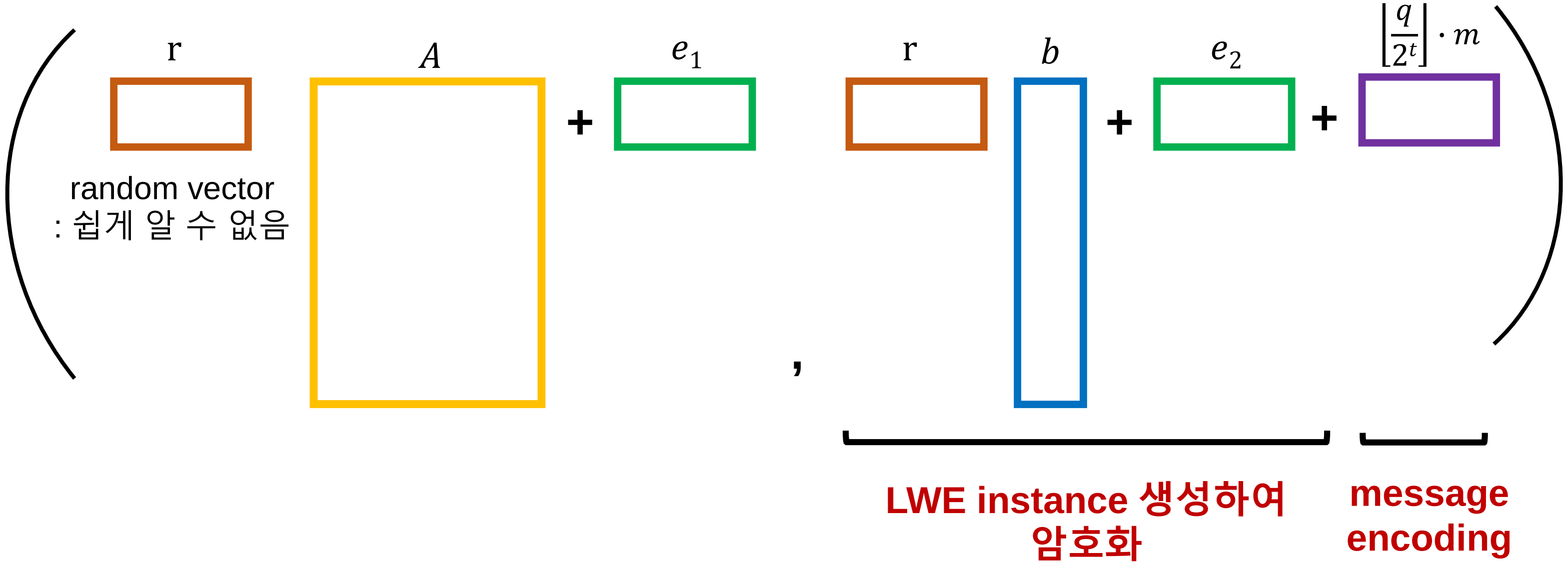
매우 작은 값



∗ 모든 연산은 $mod\ q$

LWE-based Encryption – encryption

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

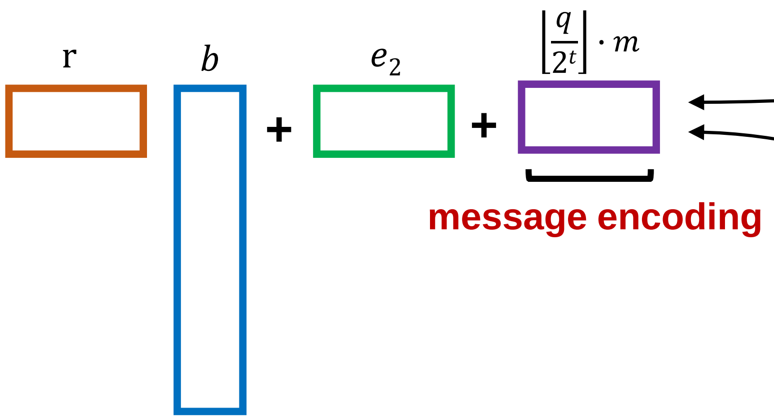


★ 송신자가 (C_1, C_2) 만들기 위해 가우시안분포에서 e_1, e_2를 뽑아서 사용 ($message$마다 새로 선택)

→ 수신자는 e_1, e_2 모름

LWE-based Encryption – message

❖ $t - bit$ message encoding : $\left\lfloor \frac{q}{2^t} \right\rfloor \cdot m$



$* \mathrm{t} = 1 \text{ and } \mathrm{q} = 2^{13}$

$if\ m = 1 \ \rightarrow 1000000000000$
error를 모두 더한 값 $\rightarrow$ 0000010110010
error + m $\rightarrow$ 1000010110010

$if\ m = 0 \ \rightarrow 0000000000000$
error를 모두 더한 값 $\rightarrow$ 0000010110010
error + m $\rightarrow$ 0000010110010

파급 x

$(\log_2 q)\ bit$

message	***error***
$t\ bit$	$(\log_2 q) - t\ bit$

*$\lfloor x \rfloor$ = max { $n \in$ Z : $n \leq x$ }

$if \quad q = 2^{13}; \left\lfloor \frac{q}{2^t} \right\rfloor \rightarrow \left\lfloor \frac{2^{13}}{2^t} \right\rfloor$: 2^{13-t}보다 작은 최대 정수 → $13 - t\ bit$로 표현 가능

* e 는 매우 작은 값 $from\ discrete\ gaussian$

→ 최상위비트($message$)까지 $error$가 파급되지 않음

→ m 유지 가능

★ **암호화, 복호화에 *good***

LWE-based Encryption – decryption

❖ Message bit recovery : rounding

- $0, \frac{q}{2}, \ldots, \frac{q}{2^t}$ 중, $C_2 - C_1 \cdot S$가 **가장 가까운 값으로 보냄**
- $C_2 - C_1 \cdot S$ 에 ***rounding constant*** 더하여 잘라냄
  - ➢ $error$는 매우 작은 값으로 $message$에 영향이 없기 때문에 정상적 복호화 가능

$$C_2 - C_1 \cdot S = \underbrace{(r \cdot e + e_2 - e_1 \cdot S)}_{\textbf{decryption error}} + \left\lfloor \frac{q}{2^t} \right\rfloor \cdot m$$

$$|error| < \frac{q}{2^{t+1}}$$

$error$가 매우 작은 값이어야 **정상적인 decryption** 가능

$* \ t = 1 \text{ and } q = 2^{13}$

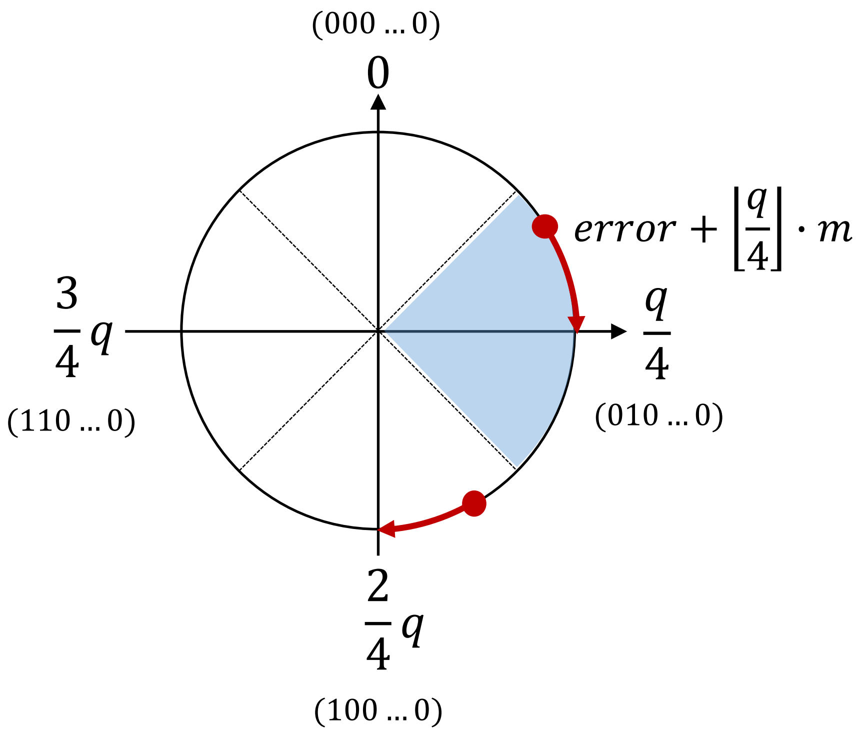
$if\ m = 1$ →	1 000010110010
$rounding\ constant$ →	0 100000000000
	1 100010110010

$if\ m = 0$ →	0 000010110010
$rounding\ constant$ →	0 100000000000
	0 100010110010

LWE-based Encryption – decryption

- example
  - $C_2 - C_1 \cdot S = (\mathrm{r} \cdot\ e + e_2 - e_1 \cdot \mathrm{S}) + \left\lfloor \frac{q}{4} \right\rceil \cdot m$

$* \, if \ t = 2 \ \rightarrow message : 2bit$



- **q = 2^{13} 경우, message bit recovery**

  $* \ t = 2 \ \rightarrow$ message bit : 상위 2bit 뒤쪽은 자름

  - 0에 가까우면 0으로 보냄 $\rightarrow 0 = 0000000000000$

    $\rightarrow$ ***message bit = 00***

  - $\frac{q}{4}$에 가까우면 2^{11}로 보냄 $\rightarrow 2^{11} = 0100000000000$

    $\rightarrow$ ***message bit = 01***

  - $\frac{2}{4}q$에 가까우면 2^{12}로 보냄 $\rightarrow 2^{12} = 1000000000000$

    $\rightarrow$ ***message bit = 10***

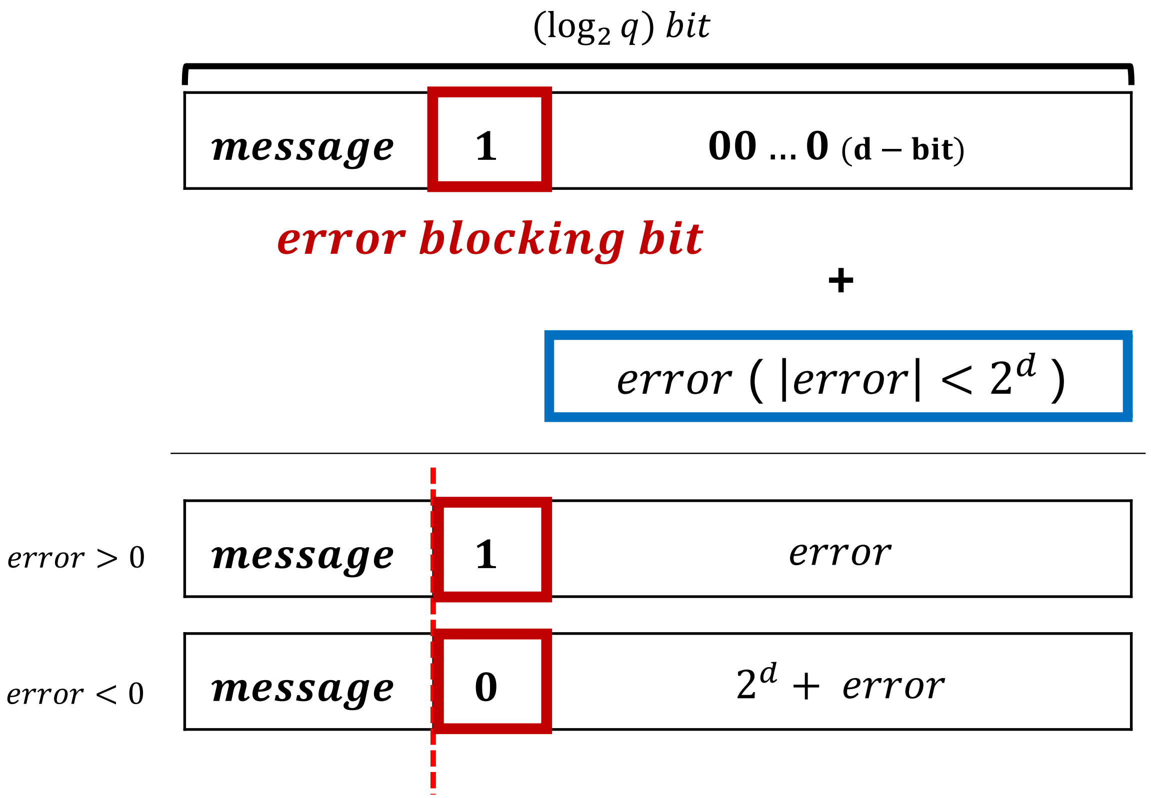
  - $\frac{3}{4}q$에 가까우면 $(1+2) * 2^{11} \rightarrow 2^{12} + 2^{11} = 1100000000000$

    $\rightarrow$ ***message bit = 11***

LWE-based Encryption – decryption

❖ EMBLEM encoding

- $rounding\ constant$ 더하는 과정 없이 $message$ 복구 가능 $(faster)$
- $error\ bit$ 범위 초과 시 정상적 복호화 불가



모든 $message\ block$마다 뒤에 1과 $0(d-bit)$을 붙임

→ 1||0^d

최대 $d-bit$ → $message$까지 $error$가 파급되지 않음

$error < 0$이지만 양수와 똑같이 더하고

$error\ blocking\ bit$자리에 1을 더함$(= 2^d + error)$

→ 0으로 바뀜

Q & A