

SHE

FHE \rightarrow 암호체계 (아직 후회적 방법 사용)

homomorphism \rightarrow FHE 처럼, ring element 를 다룰 수 있음
 \hookrightarrow bootstrapping 은 바로 FHE 이다.

• vector 기반 polynomial로 만들어서 (한번에?) 함수 있는 것.

• Grothomical polynomial $f(x) = x^n + 1$ (n 은 2^k 형태) $\rightarrow x^2+1, x^4+1, x^8+1, x^{16}+1$
 \hookrightarrow 빠른 회산 위해 $\mathbb{Z}_4 / \langle f(x) \rangle$ 이면 $x^4 = -1$
 (덧셈, 곱셈 정의 가능해서 Ring 정의 가능)

distribution 가능하면 예시 가장가려운 경우 여러개 뽑아서 예외 제제 (다항식) 생성

$(a_1, a_0) \rightarrow (n_1' + 0.001x^5 + -3.1x^2 + \dots, \dots \dots)$ 다항식 $a_i \in \mathbb{R}_q$ (uniform)

5가 noise에 영향을 리서 \mathbb{R}_q 에서 뽑았으나 γ (가까이한 예시)

효율적 binary 에서 뽑아서 noise 영향 \downarrow 미고 빠르게 계산 (그래도 안전, 하이 매튜 크기 때문)

$f, g, u \leftarrow X$ 해서 $ct = (c_0, c_1) = (a_0u + t_0 + m, a_1u + t_1 + f)$

저장 \hookrightarrow 쓰는 이따는, 공개돼있어서 안전하지 않다 암호를 잘 숨기는 회로 (a_0, a_1 이 공개돼있을 때 $+fu$ 로 숨김)

$\rightarrow (c_0, c_1) \cdot (1, s)$ 로 Dec 도 된다.

* 효율성 측면

① $\begin{bmatrix} c_0 = (a_0, a_1, \dots, a_n) \\ c_1 = (a'_0, a'_1, \dots, a'_n) \end{bmatrix} \rightarrow$ product (tensor 이기때문) $O(n^2)$

② $\begin{bmatrix} c_0 = \\ c_1 = \end{bmatrix} \rightarrow$ 또 $O(n^2)$ 이지만 FFT also 사용하면 $O(n)$ 이다. 그래서 RLWE 를 사용
 \hookrightarrow 곱셈 빠른 $O(n)$
 \uparrow 곱셈 변환 (Fast Fourier Transform!)

$$\mathbb{Z}/\langle 3 \rangle = \mathbb{Z}/3\mathbb{Z} \quad \begin{array}{l} \mathbb{Z} = \{0, 1, 2, \dots\} \\ 3\mathbb{Z} = \{0, 3, 6, 9, \dots\} \end{array}$$

$$1 \equiv 1 \in \mathbb{Z}/\langle 3 \rangle$$

$$\therefore \begin{pmatrix} 0 & 3 & 6 & \dots \\ 1 & 4 & 7 & \dots \\ 2 & 5 & 8 & \dots \end{pmatrix} \text{ 같이 생각된다.}$$

$$\text{ex) } \mathbb{Z}[x] / \langle x^4+1 \rangle = \mathbb{Z}[x] / (x^4+1) \mathbb{Z}[x]$$

$$= \{ a(x) \mid a(x) = (x^4+1) b(x) \text{ for some } b \in \mathbb{Z}[x] \}$$

$$\hookrightarrow x^5 + x + 1 \Rightarrow x(x^4+1) + 1 = 1$$

$$\mathbb{Z}[x] / \langle 3 \rangle \quad \left(\eta = 3 \times 2 + 1 = 1 \right) \text{ 인 것과 같은 원리}$$

\hookrightarrow 3을 0으로 취급했으니 $x^4+1=0$ 으로 생각

RLWE \Rightarrow 요즘도 이 기반으로
 사용 \therefore 미래-proof

시간표 확인

2월 16일 저녁! (칸)

이번주 workshop (아침 5~7시) 에 꼭 참석해주세요.

\hookrightarrow 과제는 요일날 한 시종!

● 합성곱

⇒ 두 함수를 곱해서 합함 $(f * g)(t) = \int_{-\infty}^{\infty} f(\tau) g(t - \tau) d\tau$ → g 를 반전시키고, t 만큼 이동 그후 f 와 하나씩 곱함

● 어디 사용?

↳ 행렬에서 합성곱 (DL), 신호필터, 라플라스 변환, 푸리에 변환

→ f, g 가 있는데 f 는 행렬, 원시신호, 이미지 / g 는 커널치, 필터로 표현

이것 f, g 를 합성곱하면 우리의 목적 값 얻을수 있다.

● 행렬의 합성곱

$$\begin{pmatrix} 0 & 3 & 2 & 1 \\ 2 & 1 & 0 & 2 \\ 1 & 3 & 2 & 0 \\ 2 & 0 & 3 & 3 \end{pmatrix} \times \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 6 & 7 \\ 10 & 8 \end{pmatrix}$$

입력 (4x4) 필터 (3x3) 결과 (피쳐맵)

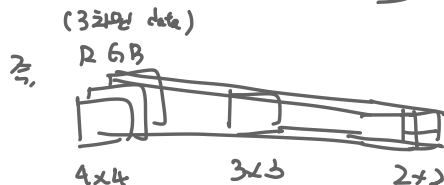
7x7

$0 \times 1 + 3 \times 0 + 2 \times 1$
 $2 \times 0 + 1 \times 1 + 0 \times 0 \rightarrow 6$
 $1 \times 1 + 3 \times 0 + 2 \times 1$

$$\begin{pmatrix} 6 & 7 \\ 10 & 8 \end{pmatrix} + 2 = \begin{pmatrix} 8 & 9 \\ 12 & 10 \end{pmatrix}$$

편향 (bias)

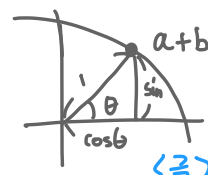
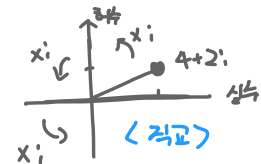
이미지는 RGB 입력해 $\begin{bmatrix} \text{red} \\ \text{green} \\ \text{blue} \end{bmatrix} \times \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \Rightarrow \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$ → 각각더해서 $\begin{pmatrix} 12 & 15 \\ 16 & 13 \end{pmatrix}$ 이된다.



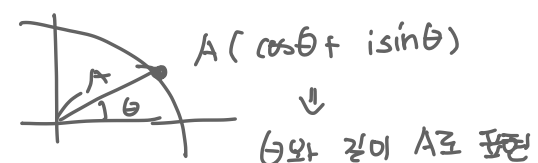
독립된 유리해미 이미지 사이즈 줄임

FFT (고속 푸리에 변환)

- 복소 평면 (실수 - x / 허수 - y)



$$\sqrt{a^2+b^2} = \sqrt{r^2(\cos^2\theta + \sin^2\theta)} = r$$



- 오일러 공식

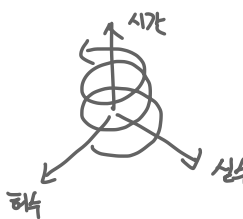
극좌표 $z = \cos\theta + i\sin\theta$ 을 미분 $\rightarrow \left(\frac{dz}{d\theta} = -\sin\theta + i\cos\theta\right) \times (-i) \Rightarrow -\frac{dz}{d\theta} i = \cos\theta + i\sin\theta = z$

정리하면 $\frac{dz}{z} = \frac{1}{-i} d\theta = i d\theta \Rightarrow$ 양변 적분 $\underline{\ln|z| = i\theta + C}$

θ가 0이면 z=1 임으로 C=0 이다. 즉 $\ln|z| = i\theta$ 이다

즉 $\underline{z = e^{i\theta} = \cos\theta + i\sin\theta}$

- 복소평면 극좌표계 3D 표현



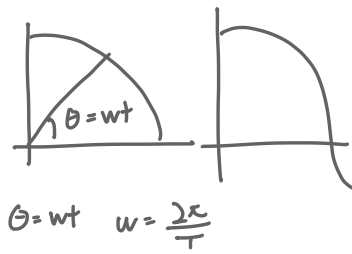
\Rightarrow 시간-실수 평면에 투영 $\Rightarrow \cos\theta$
 시간-허수 평면에 투영 $\Rightarrow \sin\theta$

$\hookrightarrow e^{\pi i} = -1 \rightarrow \underline{e^{\pi i} + 1 = 0}$

푸리에 급수 (수열의 합)

$$\Rightarrow \textcircled{1} f(t) = \sum_{n=-\infty}^{\infty} C_n e^{j \frac{2\pi n}{T} t} = \sum_{n=-\infty}^{\infty} C_n e^{jn\omega t} = \sum_{n=-\infty}^{\infty} C_n (\cos(n\omega t) + j \sin(n\omega t))$$

\nearrow 보편적 공식 \nearrow 가중치



$$\textcircled{2} C_n = \frac{1}{T} \int_{-T/2}^{T/2} f(t) e^{-j \frac{2\pi n}{T} t} dt$$

{ 이것으로 어떠한 주기함수도 표현할 수 있다!

↳ 왜? 함수의 직교성을 이용 \Rightarrow 모든 점을 표현 하려면 축 (dimension)의 개수를 무한히 늘린다.

↳ ex) $\vec{a} \cdot \vec{b} = 0$ 이고 2차원의 점임
 $\vec{a} \cdot \vec{b} \cdot \vec{c} = 0$ 이고 3차원의 점임

푸리에 급수 = (계수 C_n) \times (단순 주기함수 $e^{jn\omega t}$)

↳ 계수는 축, 주기함수와 직교성을 가지는 것을 증명한다? \Rightarrow 모든 주기함수 표현가능

↳ 생각할게... 내 뇌야랑 거리 멀어

①에 ② 대입하고 극한을 취해준다. (T 가 ∞ 이면 비국지함수도 국지함수로 볼 수 있다.)

$$\lim_{T \rightarrow \infty} f(t) = \lim_{T \rightarrow \infty} \sum_{n=-\infty}^{\infty} \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} f(t) e^{-j\frac{2\pi n}{T}t} dt e^{j\frac{2\pi n}{T}t} \dots \quad \text{리만하고 정리하면}$$

$$f(t) = \int_{-\infty}^{\infty} \underbrace{\int_{-\infty}^{\infty} f(t) e^{-j2\pi ut} dt}_{\text{계속의 역변환}} e^{j2\pi ut} du$$

↳ 계속의 역변환 이자 푸리에 변환 식이다.

• 푸리에 변환 VS 이산(고속) 푸리에 변환

⇒ 푸리에 급수 변환은 시간에 따라 연속적이어야 함. 그러나 디지털 신호는 이산적이라 이산 푸리에 변환 사용
 이산 푸리에 변환 (DFT)은 $O(N^2)$ 걸려서 고속 푸리에 변환 (FFT) 사용 $O(N \log N)$ 걸림.
 ↳ 한성공을 빨리함.

• 고속 푸리에 변환 원리 ($O(N \log N)$)

⇒ 전부 변환이 아니라 트렁크 신호만 골라서 변환시킴

⇒ N 개를 홀.짝 나눠 계산, ex) 100개 중 10개 골라내, 10개를 단순 연결해 제외된 신호들 예상

• 이산 푸리에 변환 (DFT)

$$x[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi nk/T} \quad (n \text{은 순번, } N \text{은 샘플 갯수}) \quad (A_n = \sum_{j=0}^{N-1} e^{-j2\pi jn/N} a_j \text{ 이렇게도 가능})$$

() $x[n]$ 은 실수 $x[k]$ 는 복소수. 진폭과 위상이 다른 N 개 성분들 모두 더해 원래 신호 얻음
 $x[k]$ 는 N 개만에 중심값은 대칭으로 걸림이다. 그래서 모든 합은 실수가 된다.

• 역변환 (IDFT)

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} x[k] e^{j2\pi nk/T} \quad \text{or} \quad a_n = \sum_{j=0}^{N-1} \frac{1}{N} e^{j2\pi jn/N} A_j$$

- 푸리에 변환은 Convolution을 곱으로 바꿔줌 ($F(a * b) = F(a) \cdot F(b)$)

즉기 N 인 수열 $\{a_j\}_{j=0}^{N-1}$, $\{b_j\}_{j=0}^{N-1}$ 의 convolution $\{c_n\}_{n=0}^{N-1} \Rightarrow c_n = \sum_{j=0}^{N-1} a_j b_{n-j}$

다른가지로 convolution에 DFT를 취하면 곱이됨 (증명 생략)

($b_{N-k} = b_{-k}$ 라 주기성)
그래서 뒤집어 씌웠다

* ($\{c_n\}_{n=0}^{N-1}$ 은 직접하면 $O(N^2)$ 이 걸리지만, 두수열에 DFT 취한후

곱하고 IDFT를 취하면 convolution 결과 얻음식 있다 \Rightarrow FFT (주로 Cooley-Tukey Algo)

($2^n = N$ 일때 가능하지만 만능변
형

$$A_n = \sum_{j=0}^{N/2-1} e^{-2\pi i (2j) n / N} a_{2j} + \sum_{j=0}^{N/2-1} e^{-2\pi i (2j+1) n / N} a_{2j+1}$$

$$= \sum_{j=0}^{N/2-1} e^{-2\pi i j n / (N/2)} a_{2j} + e^{-2\pi i n / N} \sum_{j=0}^{N/2-1} e^{-2\pi i j n / (N/2)} a_{2j+1}$$

n 대신 $n + N/2$ 도 넣어서 안됨

$e^{-2\pi i n / N}$ 만 복호 변환다

(그냥 1바퀴라 같음)

$$\therefore A_{n+N/2} = \sum_{j=0}^{N/2-1} e^{-2\pi i j n / (N/2)} a_{2j} - e^{-2\pi i n / N} \sum_{j=0}^{N/2-1} e^{-2\pi i j n / (N/2)} a_{2j+1}$$

(작은번재 DFT와 작은번재 DFT를 알고있다면 전체 DFT 계산하는데 $O(N)$ 이 걸림

$O(N \log N)$ 에 DFT 계산 가능