

● 논문 의미

automorphism 으로 plain text 도 생성,
canonical & Norm 생김, see A.9 appendix

=> AES 를 동형암호화 하였고 그 과정 사용된 테크닉들 소개

↳ 이는 BGV 성능을 대폭 향상

Automorphism, 갈루아군

↳ 자기자신으로 가는 사상 (행동, 공약) 들의 집합

$$\text{ex) } f(x) = (x^2 - 2)(x^2 - 3) \begin{pmatrix} \sqrt{2} & \times & = & \times & = \\ -\sqrt{2} & \times & = & \times & = \end{pmatrix} \Rightarrow 4\text{개다.}$$

↳ 6개의 automorphism 으로 평문 아크공을 성질 가지게 된다

$$x_1 \begin{pmatrix} m_1 & m_2 & m_3 & m_4 \end{pmatrix}$$

↓

$$x_2 \begin{pmatrix} m_2 & m_3 & m_4 & m_1 \end{pmatrix}$$

) =>

여러연산 801

AES Shift 연산 16를

$$x_3 \begin{pmatrix} m_1 & m_2 & . & . \\ m_2 & m_3 & . & . \end{pmatrix}$$

* 가결은 Cypher text 에 관심있을지라도 그런 따도 신경쓰지

마리 좋은 성질 만들어 준다 (very cool technique)

$$\Phi(x) = \prod_{i \in (\mathbb{Z}/m\mathbb{Z})^*} (x - z^i)$$

$F_z = F_1 \times F_2 \times F_3 \times F_4$ 인데 \Rightarrow 공간을 나누서 번갈아가며 효율적으로 쓸 수 있다.

$$\Phi_m = F \text{ 라고 하면 } \subseteq F_1 = (x - z^1) \quad F_2 = (x - z^2) \dots$$

성질을 만족하면 좋은데 이는 특정한 값들이 있는 것이다

$$(m = 0, 0 = \bigcirc) = m = 0 \text{ 인 경우 전체 계수 10이고 } + 0 \text{ 이다.}$$

Curriculum embedded Norm은

식에 $(z^1, z^2, z^3, z^4, \dots)$ 를 대입해서

이들 벡터를 가장 큰 값은 의미하고 이는 noise 특성하는데
반대,

$$\prod_{i=1}^2 \mathbb{Z}[x] / f_i \cong \mathbb{Z}[x] / \Phi_m(x) \rightarrow \mathbb{Z}[x] / \Phi_n(x) = A_d$$

(각본 것들 decomposition 한다) next space cirt space

Contribution (Plane 공간도 polynomial (Lattice) 을 위한 norm 설정)

poly 는 $E(a(x) + b(x))$, $E(a(x) \times b(x))$ 만족해야
우리가 정말 원하는 것은 아니다.

$$\text{2244-1} \quad \mathbb{Z}_m(x) = \prod_{i=1}^l f_i(\text{mod } p) \text{ 을 공간으로 } \mathbb{Z} / \mathbb{Z}_m(x)$$

2244-1 Vector를 설정.

$$\therefore A = (a_1(x), a_2(x), \dots, a_n(x))$$

↖ slot ↗ slot ↘ slot

$$B = (b_1(x), b_2(x), \dots, b_n(x))$$

$$\text{2244-2} \quad A+B \rightarrow a(x) + b(x) \rightarrow c_1(x) + c_2(x)$$

$$A \otimes B \rightarrow a(x) \times b(x) \rightarrow \text{mul}(c_1(x) \times c_2(x))$$

(불가능했던 plane에서 연산이 유효해졌고, 1차원 공간 잘 사용)

$$G_{xt} \in A_q \quad (\mathbb{Z}_q[x] / \Phi_m(x)) \quad \therefore \text{Double (RT 의미?)}$$

\Rightarrow 쪼개서 \downarrow 한 켤
양호한 공간론 정의

$$\mathbb{Z}_q[x] / \Phi_m(x) \simeq \prod_{i \in \mathbb{Z}_m^*} \mathbb{Z}_q[x] / (x - z^i) = \mathbb{Z}_q[x] / (x - z^i) \times \mathbb{Z}_q[x] / (x - z^i)^{m_i}$$

$$f \longmapsto (f \bmod (x - z^i), f \bmod (x - z^i)^{m_i}, \dots)$$

$$\downarrow \qquad \qquad \downarrow$$

$$f(z^i), \qquad f(z^i)^{m_i}$$

polynomial 을 그래프로 보는데 아티라 특별한 점들에서
evaluation 하는데 ($q \equiv 1 \pmod{m}$ 되면 $z \in \mathbb{Z}_q$ 에 들어감)

\hookrightarrow 특별한 성질

저런 성질 쓰면 $z \in \mathbb{Z}_q$ 라서 예쁘게 됨,

degree 낮음
(CRT, FFT)

$$(q = \prod_{i=1}^t p_i)$$

$$\mathbb{Z}_q \simeq \mathbb{Z}_{p_1} \times \mathbb{Z}_{p_2} \times \dots \times \mathbb{Z}_{p_t} \quad (\mathbb{Z}_q \text{ 을 작게 함})$$

$$a \longmapsto (a \bmod p_1, a \bmod p_2, \dots)$$

\Rightarrow 2개 쪼개는 (FFT, CRT) \Rightarrow Double CRT가 됨.

덧셈, 곱셈 가능

$$\therefore \mathbb{Z}_q[x] / \Phi_m(x) \simeq \prod_{i \in \mathbb{Z}_m^*} \prod_{j=1}^L \mathbb{Z}_{p_i}^{(j)} / (x - z^i)$$

(왜 z^i)
 \Rightarrow 쪼개고 / 작게 \downarrow
(저런 $p_i = 100$, $z \in \mathbb{Z}_{p_i}$)

$$[f(z^1)]_{p_1} \quad [f(z^1)]_{p_2} \quad [f(z^1)]_{p_3} \quad \dots$$

$$[f(z^2)]_{p_1}$$

$$[f(z^3)]_{p_1}$$

⋮

⇒ 비평계량

문제 (p1, p2, ..., pk) 를 mod 해야 했는데!

p1, p2, p3, ... 각각 각각으로 해시 계산 빨라진다.

$$O(n^2) \rightarrow O(n)$$



→



⇒

대부분 빨라짐.

Modulus Switching

$$q_L = \prod_{i=0}^j p_i, \quad q_{L-1} = \prod_{i=0}^{j-1} p_i \Rightarrow \frac{q_{L-1}}{q_L} = p_L$$

$$c_0 + c_1 \cdot s = m + \underbrace{2e}_{\rightarrow \text{크기만 증가}} \pmod{q_L}$$

\rightarrow 계수도 줄어든다.

$$\frac{1}{p_L} (c_0 + c_1 s) = m + \frac{2e}{p_L} \pmod{q_{L-1}}$$

\rightarrow scale 이라는 상수로 m은 건드리 않는다 $\Rightarrow + B_{\text{scale}}$
보정값이 붙음

Key Switching

$$c_0 + c_1 s = m + 2e [q]$$

$$c_0' + c_1' s = m' + 2e' [q]$$

다항식

\uparrow

$$c_0 c_0' + s(c_1 c_1' + c_1 c_0') + s^2(c_1 c_1') = m m' + 2\tilde{e} [q]$$

\rightarrow 해가 생겨서 문제

$$\text{Des} = x(1-s) \text{ 한다.}$$

$$d_0 + d_1 s + d_2 s^2 = m m' + 2\tilde{e} \approx \underbrace{\tilde{c}_0}_{\downarrow d_0 + D} + \underbrace{\tilde{c}_1 s}_{\downarrow d_1 + D} \pmod{q}$$

\rightarrow 해를 구하고 싶다

* 2차항(2차)에 한 $d_2 s^2 \approx D + Ds$ 이나 D 값은
linearization (key switching)

$$\underline{d_2 s^2} \approx 12 + 12s$$

↳ mss 로 생각해서 s 로 Enc 한다 생각한다.

$$\frac{d_2 s^2 + 2e + as - as}{\downarrow}$$

$$b - as$$

) s^2 을 공개 s 로 암호화

$$(m + 2e + as, a)$$

↓ $d_2 s^2 \rightarrow$ 해시리크 보이기 때문

$$(b + 2e + as, a)$$

↓

$$b - as \quad (b, a \in \text{evaluation key, relinearization key})$$

$$d_2 \left(\frac{s^2 + 2e + as - as}{b} \right) \quad a \leftarrow A_4$$

$$\therefore d_2 b - d_2 as$$

$$\rightarrow d_0 + d_1 s + d_2 s^2 = C_0 1 + C_1 s \pmod{q}$$

Var 1

$$\downarrow$$

$$d_0 + d_2 b$$

$$\downarrow$$

$$d_1 - d_2 a$$

(공개준다) (binary로)

Var 2

$$d_3 s^2 = \sum_{i=0}^n d_i 2^i s^2 \approx \sum_{i=0}^n d_i \left(\frac{2^i s^2 + 2e_i + a_i s - a_i s}{(b, a_i)} \right)$$

Var 3 (이번 논문)

$$d_2 s^2 \approx 12 + 12.5 \pmod{q}$$

→ 쿼리

$$P d_2 s^2 = d_2 (P_2 s^2) = \pmod{Pq}$$

$$\approx d_2 \cdot \frac{(P_2 s^2 + as + 2e - as)}{(b, a)} \pmod{Pq}$$

$$\text{Mod Switch} \left[\begin{array}{l} (d_2 b, b_2 a) \pmod{Pq} \\ \rightarrow (\text{Scale}) \pmod{q} \end{array} \right]$$

⇓

그냥 자체를 P로 분리고

error 살펴보고

다시 P로 내림

$$\Phi(x) = \prod_{i \in (\mathbb{Z}/m\mathbb{Z})^*} (x - z^i)$$

$F_z = F_1 \times F_2 \times F_3 \times F_4$ 인데 \Rightarrow 공간을 나누서 번갈아가며 효율적으로 쓸 수 있다.

$$\Phi_m = F \text{ 라고 하면 } \subseteq F_1 = (x - z^1) \quad F_2 = (x - z^2) \dots$$

성질을 만족하면 좋은데 이는 특수한 값들이 있는 것이다

$$(m = 0, 0 = \bigcirc) = m = 0 \text{ 인 경우 전체 계수 10이고 } + 0 \text{ 이다.}$$

Curriculum embedded Norm은

식에 $(z^1, z^2, z^3, z^4, \dots)$ 를 대입해서

이들 벡터를 가장 큰 값은 의미하고 이는 noise 특성하는데
보인다,

다시 리뷰

메시지 공간을 잘 나눌 줄 것이 중요!

식용현 \Rightarrow 잘 표현

Double CRT $\rightarrow \Phi_m(x)$ 을 나워서 \Rightarrow 3 들을 곱으로 생각 \rightarrow FFT \uparrow

$\hookrightarrow \therefore$ 속도가 빨라진다. (Memory 는 같음)

$n^2 \rightarrow n$ (매우 빨라짐)

간혹 계속 등 mod 연산 필요할 때는 $\Phi_m(x)$ 로 갔다가 다시 $T_1(x-3)$ 로 돌아온다.

$\left(\begin{array}{l} q \equiv 1 \pmod{m} \text{ 이면 } 3 \text{ 들이 모두 곱셈이 되어 } q \equiv 1 \pmod{n} \text{ 으로} \\ \text{잡아준다} \end{array} \right)$

\Rightarrow FFT, CRT 사용

Key Switching

$$1 + d_1 s + d_2 s^2 = \boxed{C_1} + C_2 s$$

$\hookrightarrow 1 + 1s$ 로 바꾸는 방법

•

1, $d_2 \left(\frac{s^2 + 2e + as}{b} - \frac{as}{a} \right)$, b, a 공개함, but error은 불음 단

2, $d_2 s^2 = \sum d_i \times 2^i s^2 \approx \sum d_i (2^i s^2 + 2e + as - as)$
 binary (binary or +-base)

↳ 왜 binary 쓰나? 너무 커져서 d_2 를 작게 쪼갬.
 (Coeff이 0 or 1을 가지기 때문에)
 $q \rightarrow \log q$ error는 줄어듦, key는 많이 만듦, 작은 error 많이 불음) 단

3. 연산은 p 곱함 \rightarrow 상대적 에러 갯수 \rightarrow 바뀜

$$P d_2 s^2 = d_2 (P s^2) \pmod{p^4}$$

$$\approx d_2 (P s^2 + as + 2e - as)$$

(key 1개 만듦, error 작음, 득마리 다 같음)

FFT (coeff) $\xrightarrow{\quad}$ value $\xleftarrow{\text{IFFT}}$ (coeff)

$$\vec{x} = (x_0, x_1, \dots, x_{n-1}) \xrightarrow{\text{FFT}} \overbrace{f(z^0), f(z^1), f(z^2), \dots}^n$$

$\sum x_i x^i$

$$\vec{y} = (y_0, y_1, \dots, y_{n-1}) \xrightarrow{\text{FFT}} f(z^0), f(z^1), f(z^2), \dots$$

$$(f \cdot g)(a) = f(a) \cdot g(a) \rightarrow (f(z^0)g(z^0), f(z^1)g(z^1), f(z^2)g(z^2), \dots)$$

$$\therefore \underbrace{2\text{FFT}}_{n \log n} + \underbrace{(O(n) \times \text{mult})}_{n \log n} + \underbrace{\text{IFFT}}_{n \log n} = \underbrace{O(n \log n)}_{\text{green wavy line}}$$

우리 동영 알고리즘 미리 고른걸 Table 0.6.14

$$\underline{n^2 \rightarrow n}$$

$$6, 3, 2, 1 \longleftrightarrow 11, 3+2^1, 3, 3-2^1$$

• QH $n \log n$ OI??

$S(n)$: FFT complex of polydeg = n

$$S(n) = 2 \underbrace{S\left(\frac{n}{2}\right)}_{\substack{\text{be, } b_o \\ n/2}} + n = 2\left(2 \cdot S\left(\frac{n}{4}\right) + \frac{n}{2}\right) + n$$

$$= 2^2 \times S\left(\frac{n}{4}\right) + n + n$$

$$= 2^2 \cdot S\left(\frac{n}{4}\right) + 2n$$

$$= 2^2 \times \left(2 \cdot S\left(\frac{n}{8}\right) + \frac{n}{4}\right) + 2n$$

$$= 2^3 S\left(\frac{n}{8}\right) + 3n$$

\vdots

$$= 2^k S\left(\frac{n}{2^k}\right) + kn$$

$$2^k = n \text{ or } \text{at least } 2n$$

$$\therefore = n S(1) + \underbrace{\log_2 n \times n}_{\text{degree 1 or less } 7b \text{ 1}}$$

$$\therefore \underline{S(n) = n \log n}$$

for j in range $\left(\frac{n}{2}\right)$

$$b[j] = b_e + w^j \cdot b_o$$

$$b\left[w + \frac{n}{2}\right] = b_e - w^j b_o$$

return b

FFT 코드로 이해하기

```
def FFT(P)
```

```
    n = len(P)
```

```
    if n == 1:
```

```
        return P
```

) base condition

```
    w = e $\frac{2\pi i}{n}$ 
```

// 복소평면 원 n등분

```
    P_e = [P_0, P_2, ..., P_{n-2}]
```

```
    P_o = [P_1, P_3, ..., P_{n-1}]
```

) -> 짝 홀 나눠

```
    b_e = FFT(P_e), b_o = FFT(P_o)
```

```
    b = [0] * n
```


-> 초기화

```
    for j in range(n/2): (j 짝 0부터 n/2-1 까지)
```

```
        b[j] = b_e[j] + wj b_o[j]
```

```
        b[j+n/2] = b_e[j] - wj b_o[j]
```

) -> 분점 대칭 이용

$w' = -w^j$ 

```
    return b
```

ex) $P(x) = 5 + 3x + 2x^2 + x^3$

• $n=1$ FFT(5) -> [5], FFT(2) -> [2], FFT(3) -> [3], FFT(1) -> [1]

• $n=2$ $P(x) = 5 + 2x$

$w = e^{2\pi i} = 1$

w^0

$b_e = 5$

$b[0] = b_e[0] + 1 \cdot b_o[0] = 7 \quad \therefore [7, 3]$

$b_o = 2$

$b[1] = b_e[0] - 1 \cdot b_o[0] = 3$

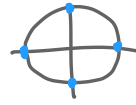
↳ Same way $3 + x \rightarrow [4, 2]$

- $n=4$

$$p(x) = 5 + 3x + 2x^2 + x^3$$

$$w = e^{\frac{2\pi i}{4}} = i$$

$$\begin{aligned} w^0 &= 1 & w^2 &= -1 \\ w^1 &= i & w^3 &= -i \end{aligned}$$



$$b_e = [1, 3]$$

$$b_o = [4, 2]$$

$$b[0] = [1, 3] + 1 [4, 2] = [11, 5]$$

$$b[2] = [1, 3] - 1 [4, 2] = [3, 1]$$

$$b[1] = [1, 3] + i [4, 2] = [1+4i, 3+2i]$$

$$b[3] = [1, 3] - i [4, 2] = [1-4i, 3-2i]$$

$$[11, 3+2i, 3, 3-2i]$$

$$p(x) = 1 + x + x^2 + x^3$$

$$\Rightarrow \text{FFT}(1) \rightarrow [1] \quad \text{FFT}(1+x) \rightarrow [2, 0]$$

when $n=4$

$$w = i$$

$$b[0] = [2, 0] + 1 [2, 0] = [4, 0]$$

$$b[2] = [2, 0] - 1 [2, 0] = [0, 0]$$

$$b[1] = [2, 0] + i [2, 0] = [2+2i, 0]$$

$$b[3] = [2, 0] - i [2, 0] = [2-2i, 0]$$

$$A(x) = 5 + 3x + 2x^2 + x^3$$

$$B(x) = 1 + x + x^2 + x^3$$

$$C(x) = A(x)B(x) = 5 + 8x + 10x^2 + 11x^3 + 6x^4 + 3x^5 + x^6 + 0x^7$$

$$n=8$$

$$n=4$$

$$\begin{pmatrix} P_e = 5 + 10x + 6x^2 + x^3 \\ P_o = 8 + 11x + 3x^2 + 0x^3 \end{pmatrix}$$

$$P_{e_e} = 5 + 6x \rightarrow [11, -1] \quad P_{o_e} = 8 + 3x \rightarrow [11, 5]$$

$$P_{e_o} = 10 + x \rightarrow [11, 9] \quad P_{o_o} = 11 + 0 \rightarrow [11, 11]$$

$$P_e \rightarrow b[0] = (11, -1) + (11, 9) \Rightarrow 11$$

$$b[2] \Rightarrow 0$$

$$b[1] = (11, -1) + i(11, 9) \Rightarrow -1 + 9i$$

$$b[3] \Rightarrow -1 - 9i$$

$$\Rightarrow [11, -1 + 9i, 0, -1 - 9i]$$

$$P_o \Rightarrow b[0] = (11, 5) + (11, 11) \Rightarrow 22$$

$$b[2] = \Rightarrow 0$$

$$b[1] = 11 + 5i + 11 + 11i \Rightarrow 11 + 11i$$

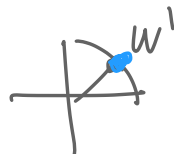
$$b[3] = \Rightarrow 11 - 11i$$

$$\Rightarrow [22, 11 + 11i, 0, 11 - 11i]$$

$$n=8$$

$$p_e \Rightarrow [11, -1+ai, 0, -1-ai]$$

$$p_o \Rightarrow [22, 11+11i, 0, 11-11i]$$



for $j = 0 \sim 3$

$$w = e^{\frac{\pi}{4}i} = (\sqrt{2}, \sqrt{2}i)$$

$$b[0] = 11 + (\sqrt{2}, \sqrt{2}i) 22 = ?$$

$$b[4] = 11 - (\sqrt{2}, \sqrt{2}i) 22 = ?$$

??

$$b[1] = (-1+ai) + i(11+11i) = -12+20i$$

$$b[5] = (-1+ai) - i(11+11i) = 10-2i$$

$$b[2] =$$

$$= 0$$

$$b[6] =$$

$$= 0$$

$$b[3] = (-1-ai) - 1(11-11i) = -12+2i$$

$$b[7] = (-1-ai) + (11-11i) = 10-20i$$

$$\therefore \Rightarrow [?, -12+20i, 0, -12+2i, ?, 10-2i, 0, 10-20i]$$

$$A(x) = 5 + 3x + 2x^2 + x^3 \Rightarrow [11, 3+2i, 3, 3-2i] \text{ ①}$$

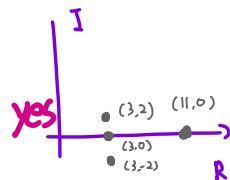
$$B(x) = 1 + x + x^2 + x^3 \Rightarrow [4, 0, 0, 0] \text{ ②}$$

$$C(x) = A(x)B(x) = 5 + 8x + 10x^2 + 11x^3 + 6x^4 + 3x^5 + x^6 + 0x^7$$

$$\Rightarrow [?, -12+20i, 0, -12+2i, ?, 10-2i, 0, 10-20i] \text{ ③}$$

1. A의 식이 로 바뀌었는데 계속 표현도식 \rightarrow 코표들 오지 않고 있습니다. 이는 다항식 $A(x) \rightarrow$ 복소평면의 점 4개

$(11, 0), (3, 2), (3, 0), (3, -2)$ 로 표현한 것 인가요? yes



그렇다면 복소평면에서 저 4개의 점을 지나는 그래프는

실제 실수로 그려진 다항식 $A(x)$ 의 그래프와 같은 것인가요? (1-2) yes

또한 복소평면에서 저 4개의 점을 연결하면 실수가 3인 지점에서

중복 값이 3개인데 복소 평면에서 가능한 것인가요? (1-3) yes

\rightarrow 피칸 용어를 잠시 배워볼 습니다.

2. 이전 page 에서 $b[0], b[4]$ 를 업데이트 해줄때 $w' = (\sqrt{2}, \sqrt{2}i)$ 이기

때문에 계산은 $11 + (\sqrt{2}, \sqrt{2}i) 22 = (22\sqrt{2} + 11, 22\sqrt{2}i)$ 이긴 ~~아닌~~

해줄 수 있는 것인가요? (2-1) yes

3. $A \times B \Rightarrow n^2$ 이 걸리기 때문에

$A \rightarrow$ 점으로 ①, $B \rightarrow$ 점으로 ② 래서 ① \odot ② = ③ 으로 $O(n)$

래서 ③을 FFT로 다시 C로 만들어 주서 $O(n \log n)$ 이 걸리

많이 시간이 단축된다고 이해 하였습니다. 이것이 맞나요? (3-1) $\rightarrow 3n \log n + n$

① \odot ② = ③ $O(n)$ 만에 연산 과정은 각각 점들은 어딘

연산을 해주어야 하나요? (점 4개 \times 점 4개 = 점 8개 이따위로 충분 합니다!) (3-2)

len 4, 4 다항식 곱하면
1개도니 2개 다항식을
해결할 한 값이 아니라
A, B를 1개씩 나눠서
점 4개씩 생성하고
그걸 곱해서 C의 점
4개씩 생성한 후 다시
순환하면 FFT를 하면



Reducible Youtube FFT => Very Good
