고려대 세미나

https://www.youtube.com/watch?v=C3NSmMTbN9Y&feature=share

장경배

Contents

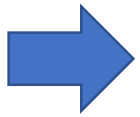
CryptoCraft LAB

McEliece

- 길이 ***k*** 의 메시지 ***m*** 을 암호화 하기 위해 Goppa code ***G*** 를 사용하여 길이 ***n*** 으로 선형확장 → 인코딩

(Message) x [Goppa code] = (codeword)

(***1*** x ***k***) (***k*** x ***n***) (***1*** x ***n***)



선형확장

(Linear expansion)

- 디코딩 과정에서는 ***G*** 에 해당하는 Parrity matrix ***H*** 가 사용됨 → Syndrome Decoding

codeword x [Parrity Check matrix.] = s, 오류가 없다면 s = 0

(***1*** x ***n*** (***n*** – ***k*** x ***n***)

McEliece

- 여기서 중요한 것은 생성된 codeword ***c*** 에 오류가 추가 되어도 수정할 수 있다는 점
  - Goppa code가 그 오류수정 역할을 수행 → 공개키로 사용된다.
  - 송신자들은 자신의 메시지와 Goppa code를 사용하여 codeword를 생성, 그 뒤에 오류 ***e*** 를 임의로 추가하여 원본 메세지를 암호화 한다. → ***mG*** + ***e*** = ***codeword*** (암호문)

- 하지만 Goppa code ***G*** 를 그대로 공개키로 사용하면 누구나 오류를 수정할 수 있음
  - 때문에 ***G*** 를 비밀스럽게 숨기는 과정이 존재

$$G' = S \cdot G \cdot P$$ → scramble 된 Goppa Matirx ***G'*** 를 공개키로 사용 (***S*** 는 가역, ***P*** 는 순열행렬)

- 마지막으로 수신자는 ***H*** 를 활용하여 수신된 암호문의 오류를 수정(Syndrome decoding)하여 원본 메세지를 획득한다.

Code – based Public-key Identification

- 1993년 Stern 은 Public-Key Identification Scheme 을 제안
  - Stern, A new identification scheme based on syndrome decoding. (1993)
  - 주어진 신드롬 값에 해당하는 low-weight 의 코드워드를 찾는 어려움에 기반
    - NP – complete

Code – based Public-key Identification

- Scheme

GF(2) 상의 [n, k] – random linear code 를 사용

모든 유저는 parity-check matirx $\boldsymbol{H}$ 와 code의 minimum distance 보다 약간 작은 정수 값 $\boldsymbol{w}$ 를 공유

각 유저는 weight $\boldsymbol{w}$ 의 n – bit 벡터 $\boldsymbol{s}$ 를 개인 secret key 로 가지게 되며

공개키는 신드롬 값인 $\boldsymbol{sH^T}$ 가 된다.

Zero-knowledge protocol 을 사용하여 $\boldsymbol{s}$ 값을 밝히지 않고도 자신이 $\boldsymbol{s}$ 를 알고있다는 사실을
증명함으로써 자신의 신원을 다른 사람에게 증명할 수 있음

Cryptanalysis Methods

- Code-based 암호 시스템을 공격하는 방법은 크게 2가지

  1. 공개키로 사용되는 generator matrix (**G'**)로 부터 original secret code (**G**)를 복구하는 방법

★ 2. linear code를 해독하는 문제 → ciphertext 로부터 plaintext를 복구하는 것

- 첫번째 방법인 Scramble 된 **G'**에서 secret **G**를 찾아내는 구조 공격(Structural attack) 또한 대표적, 하지만 두번째 항목의 **Information set decoding** 공격보다 훨씬 느림

- Information set decoding 의 목표는 주어진 $\boldsymbol{cH^T = s}$ 에서
  $\boldsymbol{H}$ 와 $\boldsymbol{s}$ 가 주어졌을 때, w- weight 의 벡터 $\boldsymbol{c}$ 를 찾아내는 것

- 다시말해서 해결하고자 하는 문제는 n개의 변수를 가지고있는 n-k개의 방정식의 선형 시스템에 대하여 해를 찾는 것이며, 여기서 무게 조건 때문에 해가 독특하다.

Information set decoding attack

- McEliece Goppa code ***G*** 를 그대로 공개키로 사용하면 누구나 오류를 수정할 수 있음
  - 때문에 G를 비밀스럽게 숨기는 과정이 존재

$$G' = S \cdot G \cdot P \rightarrow$$ scramble 된 Goppa Matirx ***G'*** 를 공개키로 사용 (***S*** 는 가역, ***P*** 는 순열행렬)

Infomation Set Decoding Attack

이러한 구조가 의미하는 것은 ***G'*** 로 생성한 **codeword**의 오류수정을 ***G*** 에 대한 ***H*** 가 수행한다는 것.

이 구조 때문에 **Information Set Decoding Attack** 이 가능

Information Set Decoding Attack

Information set decoding

- $cH^T = s \iff c'H'^T = s'$ where $\begin{bmatrix} H' = UHP \\ s' = sU^T \\ c' = cP \end{bmatrix}$

U = 가역행렬 (invertible)
P = 순열행렬 (Permutation)

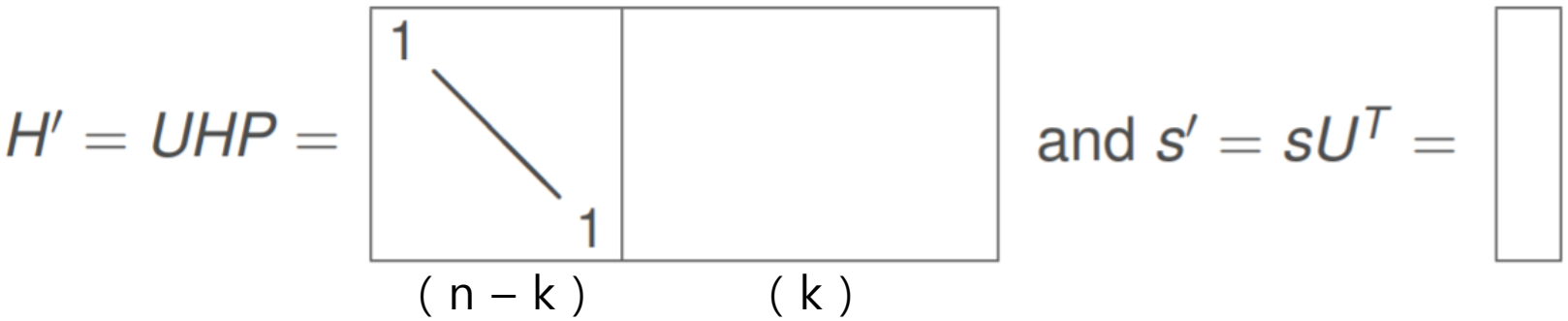
Proof

- $c'H'^T = (cP)(UHP)^T$
  $= (cP)P^T H^T U^T$ → 순열행렬과 전치행렬의 곱은 단위행렬
  $= cH^T U^T$
  $= sU^T$
  $= s'$

- 이 두가지 Syndrome decoding 계산은 동등함을 뜻한다.
- $CSD(H, s, w) \equiv CSD(UHP, sU^T, w)$ 가 동등한 것에 기반하여 **하나를 풀면 다른 한가지도 풀린다**
  → 코딩이론

Information Set Decoding Attack

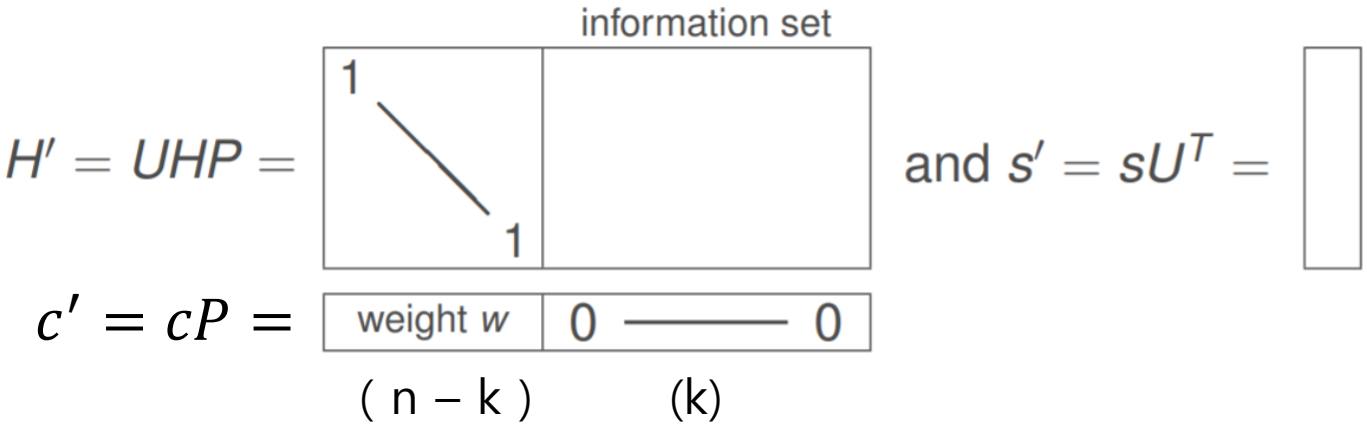
- 어떠한 U와 P를 사용해서라도 $CSD(H, s, w) \equiv CSD(UHP,\ s\mathrm{U}^{\mathrm{T}},\ w)$



- Gaussian elimination(가우스 소거법) 을 사용하여 H에서 위의 H′ 형태의 행렬을 형성한다.
- 위의 과정을 성공할 때 까지 P 와 U 를 변경하며 계산한다.

Information Set Decoding Attack

Step.



- c′ = weight w, 그리고 s′ 의 weight 도 w 이다.
- $s\mathrm{U}^{\mathrm{T}}$ 의 weight 가 w 라면 성공
  - ($s\mathrm{U}^{\mathrm{T}}$, 0) P^{-1}를 반환 → Original Syndrome decoding 에 사용될 수 있다.

Information Set Decoding Attack

Algorithm

- input : $H \in \{0,1\}^{(n-k)\times n}$, $s \in \{0,1\}^{n-k}$, integer $w > 0$
- output : $e \in \{0,1\}^n$ such that $eH^T = s$ and $\mathrm{wt}(e) = w$

Repeat :

choose a permutation matrix P

$$H' = UHP = \begin{array}{|c|c|} \hline 1 & \\ \diagdown & \\ 1 & \\ \hline \end{array} \quad \text{and } s' = sU^T = \boxed{}$$

(n – k) (k)

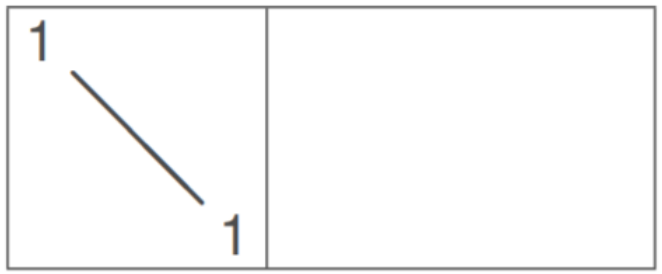
(Gaussian elimination)

if weight $(s\mathrm{U}^{\mathrm{T}})$ = w, return ($s\mathrm{U}^{\mathrm{T}}$, 0) P^{-1}

Information Set Decoding Attack

- Stern 의 Public-Key Identification
  - 주어진 신드롬 값에 해당하는 low-weight 의 코드워드를 찾는 어려움

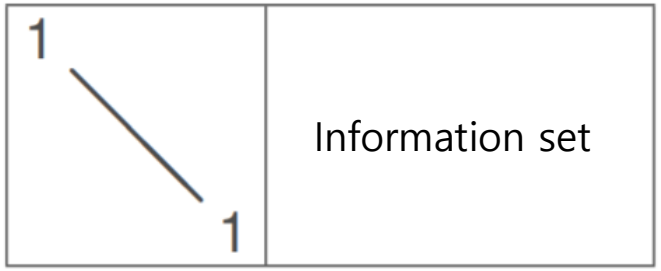
- Information set decoding 의 목표는 주어진 $cH^T = s$ 에서 w weight 의 c를 찾아내는 것
  - n개의 변수를 가지고있는 n-k개의 방정식의 선형 시스템의 해를 찾는 것이며,
    여기서 무게 조건 때문에 해가 독특하다.
  - 주어진 k 열의 오류 벡터가 zero 라면 error position은 남아있는 n-k 에 존재하게 된다.
    다시 말해서 k 에 해당하는 변수들이 선형시스템에 포함되지 않는다면,
    n-k 개의 변수를 가지고 있는 n-k 방정식의 선형 시스템을 해결함으로써 오류 벡터를 찾아낼 수 있다.

$H' = UHP =$ 

(n – k) (k)

and $s' = sU^T =$ 

Advanced Information Set Decoding Attack - Stern

$H' = UHP =$ 

and $s' = sU^T =$ 

(n – k) (k)

앞선 공격은 우측 information set 에 error 가 존재하지 않을 때 성립

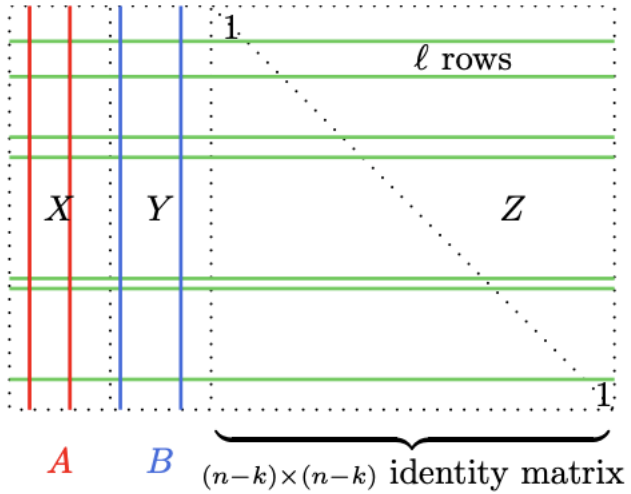
하지만 Stern의 공격법은
information set 에 작은 오류를 허용하면서 신드롬값 s'을 만족하는 벡터를 찾아냄

Advanced Information Set Decoding Attack - Stern

$2p$의 오류를 information set 에서 허용한다.

Stern 은 랜덤하게 ℓ 개의 행을 선택한다.

남은 k 열을 X 와 Y 두 그룹으로 나눈다.



1 단계 : Search

X 와 Y그룹에서 동일하게 p개의 컬럼 A, B를 선택

A, B의 열에서 ℓ 에 해당하는 합을 계산 → $\pi(A)$, $\pi(B)$

2 단계 : collision

만약 $\pi(A) = \pi(B)$ 를 만족한다면, AUB 안의 $2p$ 열들의 sum 을 계산

Sum 은 (n-k) – bit 벡터가 될 것이고, weight 가 $t - 2p$ 인지 확인

Information Set Decoding Attack

$$\mathrm{c}H^T = s$$

Parity check matrix ***H*** 와 특정 신드롬 값 ***s*** 가 주어졌을 때, w-weight 의 벡터 ***c*** 를 찾는 것

$$\mathrm{c}H'^T = 0$$

오류가 포함된 암호문 ***c*** 가 주어졌을 때, 암호문 ***c*** 로부터 오류를 수정해 원본 메세지 ***m*** 을 복구하는 것

Information set decoding attack

- McEliece system을 깨기 위해 앞서 언급한 low-weight 의 codeword를 약간 더 큰 linear-code 에서 찾는 공격법

C : $length\ -n$ 의 코드 $over\ F_2$

y : 코드워드 $c\ \in\ C$ 로부터 distance t 만큼 떨어진 코드워드

그렇다면 $\mathrm{y}\ -\ c$ 는 weight t 를 가지게 됨 → 오류벡터 e

만약, C 의 minimum distance 가 t 보다 크다면 weight - t 요소 $\mathrm{e}\ \notin C$, 즉 C 안에 포함될 수 없으며 반드시 $C + \{y\}$ 에 포함되어야 한다.

다시 말해서 $y - e$ 는 C 의 요소이며, y로부터 t 의 거리만큼 떨어져 있다.

Information set decoding attack

- McEliece system의 인코딩 과정을 살펴보면 자신의 메시지와 Goppa code를 사용하여 codeword $\boldsymbol{c}$ 를 생성, 그 뒤에 오류 $\boldsymbol{e}$ 를 임의로 추가하여 원본메세지를 암호화 한다. → $\boldsymbol{c} + \boldsymbol{e} = \boldsymbol{y}$ (암호문)

  $C : length\ -n$ 의 코드 $over\ F_2$

  y : 코드워드 $c \in C$ 로부터 distance t 만큼 떨어진 코드워드

* 코드 C 의 minimum distance 는 $2t+1$ 일때, 오류수정 가능 개수는 t

공격자는 C 에 대한 generator matrix G를 알고있기 때문에 y 를 generator 리스트에 추가하여

$C + \{y\}$ 에 대한 generator matrix를 형성할 수 있다.

$C + \{y\}$ 의 Syndrome Decoding시 유일하게 weight – t 의 codeword 가 존재하게 되는데,
이것이 $y - c$ 즉, e 가 된다.

즉, 앞선 Information set decoding 의 목표인 low-weight 의 코드워드를 조금 더 큰 linear code에서 찾음으로써 원본 c 를 획득하고 원본메세지 m을 복구할 수 있음

Information set decoding attack - Stern

새롭게 생성한 코드 $G'' = \begin{pmatrix} G' \\ C \end{pmatrix}$ 는 minimum distance t 를 가짐

목표 : 가장 낮은 weight - t 의 벡터를 찾는 것