Information Set Decoding Attack

장경배

https://youtu.be/SfAOSgpcJZo

한성대학교
HANSUNG UNIVERSITY

CryptoCraft LAB
https://crypto.modoo.at

Contents

CryptoCraft LAB

McEliece 시스템 복습

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

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

(***1 x k***) (***1 x n***) (Linear expansion)

(***k x n***)

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

McEliece 시스템 복습

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

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

→ ***mG′* + *e*** = **codeword**(암호문)

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

Infomation Set Decoding Attack

이러한 구조가 의미하는 것은 ***G′*** 로 생성한 **codeword**의 오류수정을 ***G*** 가 수행한다는 것.

이 구조 때문에 **Information Set Decoding Attack** 이 가능
→ 핵심은 원본코드 ***G*** 가 아닌 동일한 오류수정이 가능한 다른 ***G″*** 를 찾아내는 것

Information Set Decoding Attack

- 이러한 구조가 의미하는 것은 ***G′*** 로 생성한 **codeword**의 오류수정을 ***G*** 가 수행한다는 것.

- 이 구조 때문에 **Information Set Decoding Attack** 이 가능
  - 핵심은 원본 Goppa code ***G*** 가 아닌 동일한 오류수정이 가능한 다른 Goppa Matirx 를 찾아내는 것

Information set decoding

- Syndrome decoding → cH^T = s , codeword c에 G의 Parritycheck 행렬을 곱하여 syndrome 값을 획득
  - Codeword의 오류위치를 찾아주는 과정이다.
    - 오류의 개수만큼 weight 가 결정되고, 오류가 존재하지 않는다면 ***s*** 값은 0

- $cH^T = s \Longleftrightarrow c'H'^T = s'$ where $\begin{bmatrix} H' = UHP \\ S' = SU^T \\ c' = cp \end{bmatrix}$

U = non singular matrix
P = any permutation matrix

Information Set Decoding Attack

Information set decoding

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

U = any non singular matrix (invertible)
P = any permutation matrix

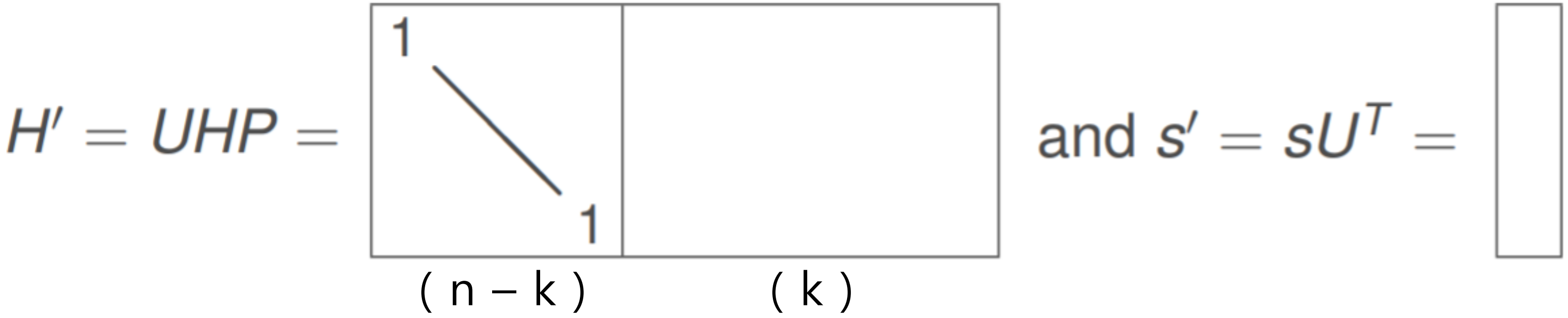
Proof

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

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

Information Set Decoding Attack

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

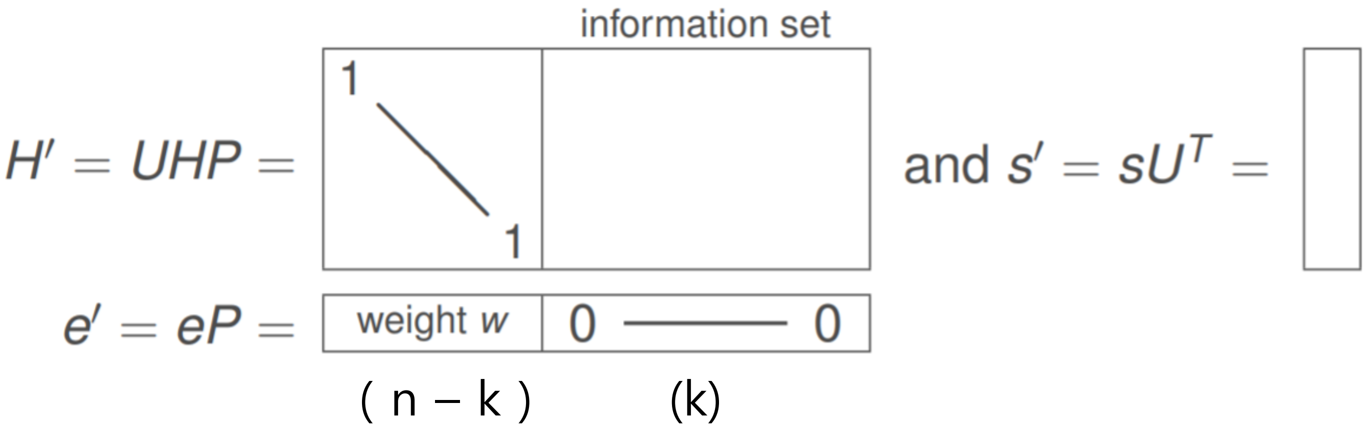


- Gaussian elimination(가우스 소거법) 을 사용하여 H에서 위의 H′ 형태의 행렬을 형성한다.
  (https://www.youtube.com/watch?v=2GKESu5atVQ)

- 위의 과정을 성공할 때 까지 P 와 U 를 변경하며 계산한다.
  - 왼쪽의 (n – k) 행렬이 Linear independent(선형독립)하다면
    행렬 뒤의 k 열은 information set을 형성한다.

Information Set Decoding Attack

Step.



- 운이 좋다면 오류 위치는 information set 밖에 존재한다.
  - e′ = 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

$$H' = UHP = \begin{array}{cc} \begin{array}{|c|c|} \hline \begin{matrix} 1 & & \\ & \ddots & \\ & & 1 \end{matrix} & \qquad\qquad \\ \hline \end{array} & \\ (\,n-k\,) \qquad\qquad (\,k\,) & \end{array} \text{ and } s' = sU^T = \boxed{}$$

(Gaussian elimination)

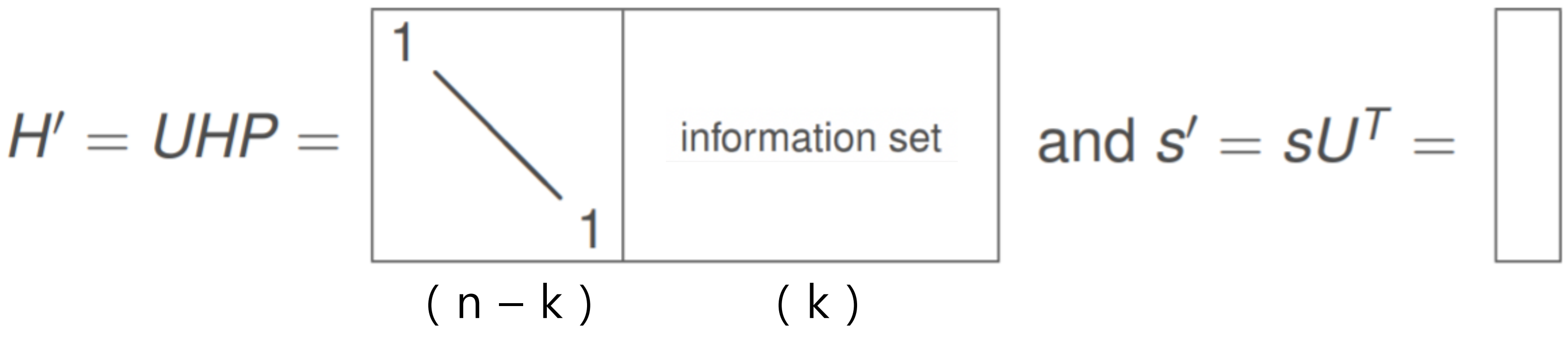
if weight ($s\mathrm{U}^{\mathrm{T}}$) = w, return ($s\mathrm{U}^{\mathrm{T}}$, 0) P^{-1}

Information Set Decoding Attack

- Information set decoding 의 목표는 주어진 $eH^T = s$ 에서 w weight 의 e를 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 은 여러 변형 버전이 있음.
  - 원리는 대부분 비슷한 것 같음
- Information set decoding 공격이 공격법 중 가장 효율적인 것 뿐이지, 확실한 공격법은 아님
  - Scramble 된 G′에서 secret G를 찾아내는 구조 공격(Structural attack) 또한 대표적, 하지만 훨씬 느림
  - Syndrome Decoding 에서 Brute force 또한 적용가능, 하지만 비현실적
- Information set decoding은 어찌 보면 효율적인 Brute force attack

Information Set Decoding Attack

- Information set decoding 의 목표는 주어진 $eH^T = s$ 에서 w weight 의 e를 찾아내는 것
- 다시말해서 해결하고자 하는 문제는 n개의 변수를 가지고있는 n-k개의 방정식의 선형 시스템에 대하여 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.



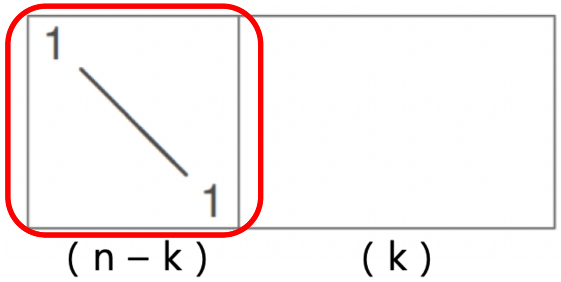
- 어려운 부분은 k – set 을 찾아내는 것이다.
최종적으로 Solving (n-k) equations, (n-k) variables and returning 1 iff error vector has weight w

Information Set Decoding Attack

step1. Gaussian Elimination

H에서 랜덤하게 n-k개의 열을 선택한다. → (n-k, n-k) 의 subset이 생겼다. → 선택된 열 ℓ subset에 대하여 행들의 선형조합을 통해 Gaussian Elimination 수행

*선형조합 : 벡터들을 스칼라 배와 벡터 덧셈을 조합하여 새로운 벡터를 얻는 연산

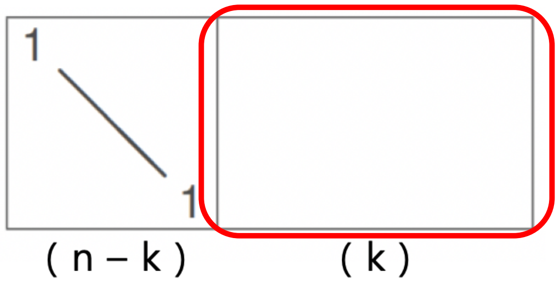


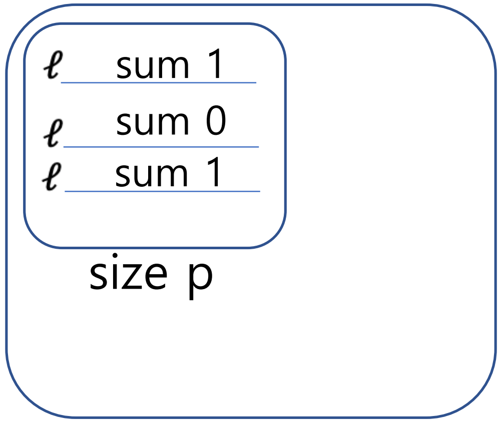
Information Set Decoding Attack

step2. Indexing

step1 에서 선택되지 않은 열의 index를 랜덤하게 쪼갠다. → X그룹과 Y그룹으로(same size)

X의 모든 size-p subset A에 대하여 sum(mod 2)을 ℓ 행에 매김으로서 ℓ-bit 벡터 $\pi(A)$ 획득
Y에도 동일하게 수행하여 $\pi(B)$ 획득





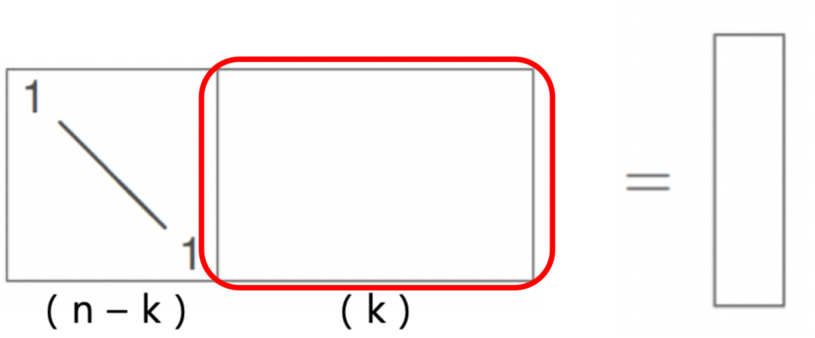
→ ℓ-bit 벡터 $\pi(A)$ = 101

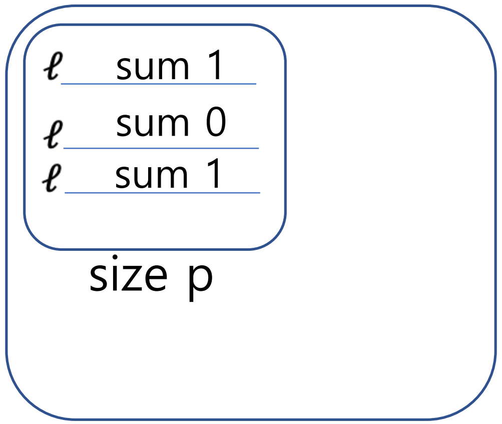
Information Set Decoding Attack

step3. collision

각 $\pi(A) = \pi(B)$ 에 대하여 A U B 의 2p 열들의 sum을 계산한다. → 이 합은 (n – k) bit 벡터가 됨

만약 이 sum의 Weight 가 w-2p 라면 이 A U B는 weight w의 codeword 를 형성





→ ℓ-bit 벡터 $\pi(A) = 101$

Quantum Information Set Decoding

- Grover 알고리즘은 데이터베이스 검색 뿐 아닌, 함수의 해를 찾는 분야에도 사용 된다.

- 이 관점에서 보아 Grover 알고리즘을 Information set decoding 에 적용하여 보자

  * 앞서 보았듯, Information set decoding 의 목표는 선형 시스템에 대하여 해를 찾는 것

- Grover 알고리즘은 size – k set 을 찾는데 사용된다.
  - 정확히는 size – k set 이 올바른지 검사하는 오라클에 적용된다.
  - 즉 선형 시스템의 n-k 다항식, n-k 변수를 푸는 것과, weight t 의 오류 벡터를 찾아 낸다.

감사합니다