3. Message Attack (ISD)

1. From Generic Decoding to Syndrome Decoding
2. Combinatorial Solutions: Exhaustive Search and Birthday Decoding
3. **Information Set Decoding: the Power of Linear Algebra**
4. Complexity Analysis
5. Lee and Brickell Algorithm
6. Stern/Dumer Algorithm
7. May, Meurer, and Thomae Algorithm
8. Becker, Joux, May, and Meurer Algorithm
9. Generalized Birthday Algorithm for Decoding
10. Decoding One Out of Many

Information Set Decoding: Using Linear Algebra

For any invertible $U \in \{0,1\}^{(n-k)\times(n-k)}$ and any permutation matrix $P \in \{0,1\}^{n\times n}$

$$\left(eH^T = s\right) \Leftrightarrow \left(e'H'^T = s'\right) \text{ where } \begin{cases} H' & \leftarrow & UHP \\ s' & \leftarrow & sU^T \\ e' & \leftarrow & eP \end{cases}$$

Information Set Decoding: Using Linear Algebra

For any invertible $U \in \{0,1\}^{(n-k)\times(n-k)}$ and any permutation matrix $P \in \{0,1\}^{n\times n}$

$$\left(eH^T = s\right) \Leftrightarrow \left(e'H'^T = s'\right) \text{ where } \begin{cases} H' & \leftarrow & UHP \\ s' & \leftarrow & sU^T \\ e' & \leftarrow & eP \end{cases}$$

Proof:
$$\begin{aligned} e'H'^T &= (eP)(UHP)^T \\ &= (eP)P^T H^T U^T \\ &= eH^T U^T \\ &= sU^T \\ &= s' \end{aligned}$$

Information Set Decoding: Using Linear Algebra

For any invertible $U \in \{0,1\}^{(n-k)\times(n-k)}$ and any permutation matrix $P \in \{0,1\}^{n\times n}$

$$\mathrm{CSD}(H, s, w) \equiv \mathrm{CSD}(UHP, sU^T, w)$$

Information Set Decoding: Using Linear Algebra

For any invertible $U \in \{0,1\}^{(n-k)\times(n-k)}$ and any permutation matrix $P \in \{0,1\}^{n \times n}$

$$\mathrm{CSD}(H, s, w) \equiv \mathrm{CSD}(UHP, sU^T, w)$$

In particular $H' = UHP = \left[\begin{array}{ccc|ccc} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \end{array}\right]$ and $s' = sU^T = \left[\begin{array}{c} \\ \\ \\ \end{array}\right]$

Information Set Decoding: Using Linear Algebra

For any invertible $U \in \{0,1\}^{(n-k)\times(n-k)}$ and any permutation matrix $P \in \{0,1\}^{n\times n}$

$$\mathrm{CSD}(H, s, w) \equiv \mathrm{CSD}(UHP, sU^T, w)$$

In particular $H' = UHP = \left[\begin{array}{c|c} \underbrace{\begin{matrix}1 & & \\ & \ddots & \\ & & 1\end{matrix}}_{n-k} & \qquad\qquad \end{array}\right]$ and $s' = sU^T = \left[\begin{array}{c} \\ \\ \end{array}\right]$

possible if the first $n-k$ columns of HP are independent

Information Set Decoding: Using Linear Algebra

For any invertible $U \in \{0,1\}^{(n-k)\times(n-k)}$ and any permutation matrix $P \in \{0,1\}^{n\times n}$

$$\mathrm{CSD}(H, s, w) \equiv \mathrm{CSD}(UHP, sU^T, w)$$

In particular

$$H' = UHP = \left[\;\underbrace{\begin{matrix}1 & & \\ & \ddots & \\ & & 1\end{matrix}}_{n-k}\;\middle|\;\overbrace{\underbrace{\begin{matrix}\qquad\qquad\qquad\end{matrix}}_{k}}^{\text{information set}}\;\right] \quad \text{and } s' = sU^T = \left[\;\right]$$

possible if the first $n-k$ columns of HP are independent

in which case the rightmost k positions form an information set

Information Set Decoding: Using Linear Algebra

For any invertible $U \in \{0,1\}^{(n-k)\times(n-k)}$ and any permutation matrix $P \in \{0,1\}^{n\times n}$

$$\mathrm{CSD}(H, s, w) \equiv \mathrm{CSD}(UHP, sU^T, w)$$

In particular $H' = UHP = \left[\begin{array}{c|c} \begin{matrix}1 & & \\ & \ddots & \\ & & 1\end{matrix} & \overbrace{}^{\text{information set}} \end{array}\right]$ and $s' = sU^T = \left[\right]$

$e' = eP = \left[\begin{array}{c|c} \text{weight } w & 0 \text{ ——— } 0 \end{array}\right]$

If we are lucky

– the error positions are out of the information set

Information Set Decoding: Using Linear Algebra

For any invertible $U \in \{0,1\}^{(n-k)\times(n-k)}$ and any permutation matrix $P \in \{0,1\}^{n\times n}$

$$\mathrm{CSD}(H, s, w) \equiv \mathrm{CSD}(UHP, sU^T, w)$$

In particular

$$\begin{array}{ccc} & & \text{information set} \\ H' = UHP = & \left[\begin{array}{c|c} \begin{matrix}1 & & \\ & \ddots & \\ & & 1\end{matrix} & \qquad\qquad\qquad \end{array}\right] & \text{and } s' = sU^T = \left[\begin{matrix} \\ \ \\ \ \end{matrix}\right] \\ e' = eP = & \left[\begin{array}{c|c} s' & 0 \ \text{———} \ 0 \end{array}\right] & \end{array}$$

If we are lucky

- the error positions are out of the information set
- easy to check because $e' = (s' \mid 0)$ and $\mathrm{wt}(s') = w$

Prange 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 $\mathrm{wt}(e) = w$

Prange 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:
    pick a permutation matrix P

Prange 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:

pick a permutation matrix P

compute $$UHP = \left[\begin{array}{ccc|c} 1 & & & \\ & \ddots & & \\ & & 1 & \end{array}\right]$$ (Gaussian elimination)

Prange 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:

pick a permutation matrix P

compute $$UHP = \left[\begin{array}{ccc|c} 1 & & & \\ & \ddots & & \\ & & 1 & \end{array}\right]$$ (Gaussian elimination)

if $wt(sU^T) = w$ then return $(sU^T, 0)P^{-1}$

Prange 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:

pick a permutation matrix P

compute $$UHP = \left[\begin{array}{ccc|c} 1 & & & \\ & \ddots & & \\ & & 1 & \end{array}\right]$$ (Gaussian elimination)

if $wt(sU^T) = w$ then return $(sU^T, 0)P^{-1}$

Each iteration costs about $n(n-k)$ column operations

Prange 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:

pick a permutation matrix P

compute $$UHP = \left[\begin{array}{ccc|c} 1 & & & \\ & \ddots & & \\ & & 1 & \end{array}\right]$$ (Gaussian elimination)

if $wt(sU^T) = w$ then return $(sU^T, 0)P^{-1}$

Each iteration costs about $n(n-k)$ column operations

Repeat until a solution has its non-zero coordinates “all left”

3. Message Attack (ISD)

1. From Generic Decoding to Syndrome Decoding
2. Combinatorial Solutions: Exhaustive Search and Birthday Decoding
3. Information Set Decoding: the Power of Linear Algebra
4. **Complexity Analysis**
5. Lee and Brickell Algorithm
6. Stern/Dumer Algorithm
7. May, Meurer, and Thomae Algorithm
8. Becker, Joux, May, and Meurer Algorithm
9. Generalized Birthday Algorithm for Decoding
10. Decoding One Out of Many