

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 $P \in \{0, 1\}^{n \times n}$

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

Information Set Decoding: Using Linear Algebra

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

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

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 $P \in \{0, 1\}^{n \times n}$

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

Information Set Decoding: Using Linear Algebra

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

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

In particular $H' = UHP =$

$\begin{matrix} 1 & \\ & \diagdown \\ & 1 \end{matrix}$	
---	--

and $s' = sU^T =$


--

Information Set Decoding: Using Linear Algebra

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

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

In particular $H' = UHP =$

<div style="text-align: center; padding: 10px;">1  1</div>	
---	--

 and $s' = sU^T =$

--

$\underbrace{\hspace{10em}}_{n-k}$

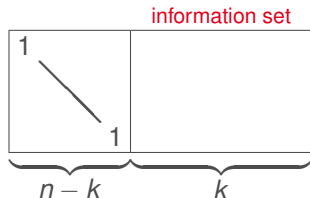
possible if the first $n-k$ columns of HP are independent

Information Set Decoding: Using Linear Algebra

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

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

In particular $H' = UHP =$



and $s' = sU^T =$

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 $P \in \{0, 1\}^{n \times n}$

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

In particular $H' = UHP =$

information set	
1 1	

 and $s' = sU^T =$

$e' = eP =$

weight w	0 ——— 0
------------	---------

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 $P \in \{0, 1\}^{n \times n}$

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

In particular $H' = UHP =$

information set	
1 1	

 and $s' = sU^T =$

$e' = eP =$

s'	$0 \text{ --- } 0$
------	--------------------

If we are lucky

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

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 $\text{wt}(e) = w$

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 $\text{wt}(e) = w$

repeat:

 pick a permutation matrix P

Prange Algorithm

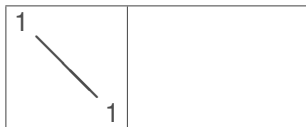
input: $H \in \{0, 1\}^{(n-k) \times n}$, $s \in \{0, 1\}^{n-k}$, integer $w > 0$

output: $e \in \{0, 1\}^n$ such that $eH^T = s$ and $\text{wt}(e) = w$

repeat:

pick a permutation matrix P

compute $UHP =$



(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 $\text{wt}(e) = w$

repeat:

pick a permutation matrix P

compute $UHP =$



(Gaussian elimination)

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

Prange Algorithm

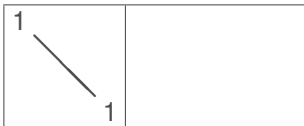
input: $H \in \{0, 1\}^{(n-k) \times n}$, $s \in \{0, 1\}^{n-k}$, integer $w > 0$

output: $e \in \{0, 1\}^n$ such that $eH^T = s$ and $\text{wt}(e) = w$

repeat:

pick a permutation matrix P

compute $UHP =$



(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

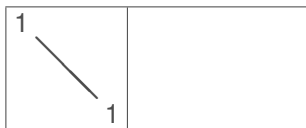
input: $H \in \{0, 1\}^{(n-k) \times n}$, $s \in \{0, 1\}^{n-k}$, integer $w > 0$

output: $e \in \{0, 1\}^n$ such that $eH^T = s$ and $\text{wt}(e) = w$

repeat:

pick a permutation matrix P

compute $UHP =$



(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