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

Lee and Brickell Algorithm

Idea: relax Prange algorithm to amortize the cost of the Gaussian elimination

Lee and Brickell Algorithm

Idea: relax Prange algorithm to amortize the cost of the Gaussian elimination

Allow error patterns of the form $e =$

$n-k$	k
weight $w-p$	weight p

At each iteration, we try the $\binom{k}{p}$ possible values for the right hand side block

(Prange Algorithm corresponds to $p = 0$)

Lee and Brickell Algorithm

Idea: relax Prange algorithm to amortize the cost of the Gaussian elimination

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

Lee and Brickell Algorithm

Idea: relax Prange algorithm to amortize the cost of the Gaussian elimination

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

repeat:
    pick a permutation matrix P

Lee and Brickell Algorithm

Idea: relax Prange algorithm to amortize the cost of the Gaussian elimination

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

repeat:

pick a permutation matrix P

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

Lee and Brickell Algorithm

Idea: relax Prange algorithm to amortize the cost of the Gaussian elimination

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

repeat:

pick a permutation matrix P

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

enumerate $\mathcal{L} = \{sU^T + e'H'^T \mid \mathrm{wt}(e') = p\}$

Lee and Brickell Algorithm

Idea: relax Prange algorithm to amortize the cost of the Gaussian elimination

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

repeat:

pick a permutation matrix P

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

enumerate $\mathcal{L} = \{sU^T + e'H'^T \mid \mathrm{wt}(e') = p\}$
if $s' \in \mathcal{L}$ has weight $w - p$ then return $(s', e')P^{-1}$

Lee and Brickell Algorithm

Idea: relax Prange algorithm to amortize the cost of the Gaussian elimination

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

repeat:

pick a permutation matrix P

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

enumerate $\mathcal{L} = \{sU^T + e'H'^T \mid \mathrm{wt}(e') = p\}$
if $s' \in \mathcal{L}$ has weight $w - p$ then return $(s', e')P^{-1}$

$\mathcal{K} = n(n-k) + \binom{k}{p}$ (Gaussian elimination + enumeration)

Lee and Brickell Algorithm – Complexity Analysis

For an error pattern $e = \overbrace{\boxed{\text{weight } w-p}}^{n-k}\overbrace{\boxed{\text{weight } p}}^{k}$, we have $\mathcal{P}_\infty = \dfrac{\binom{n-k}{w-p}\binom{k}{p}}{\binom{n}{w}}$

Lee and Brickell Algorithm – Complexity Analysis

For an error pattern $e = \overbrace{\boxed{\text{weight } w-p}}^{n-k}\overbrace{\boxed{\text{weight } p}}^{k}$, we have $\mathcal{P}_\infty = \dfrac{\binom{n-k}{w-p}\binom{k}{p}}{\binom{n}{w}}$

$\mathcal{N}_\infty = \dfrac{\binom{n}{w}}{\binom{n-k}{w-p}\binom{k}{p}}$ and $\mathcal{K} = n(n-k) + \binom{k}{p}$

Lee and Brickell Algorithm – Complexity Analysis

For an error pattern $e = \overbrace{\boxed{\text{weight } w-p}}^{n-k}\overbrace{\boxed{\text{weight } p}}^{k}$, we have $\mathcal{P}_\infty = \frac{\binom{n-k}{w-p}\binom{k}{p}}{\binom{n}{w}}$

$\mathcal{N}_\infty = \frac{\binom{n}{w}}{\binom{n-k}{w-p}\binom{k}{p}}$ and $\mathcal{K} = n(n-k) + \binom{k}{p}$

Never gains more than a polynomial factor over Prange algorithm

$$\mathrm{WF}_{\mathrm{LB}}(p) = \mathcal{N}_\infty \cdot \mathcal{K} = \frac{\binom{n}{w}}{\binom{n-k}{w-p}}\left(1 + \frac{n(n-k)}{\binom{k}{p}}\right) > \frac{\binom{n}{w}}{\binom{n-k}{w-p}} > \frac{\binom{n}{w}}{\binom{n-k}{w}} = \frac{1}{n(n-k)}\mathrm{WF}_{\mathrm{Prange}}$$

Lee and Brickell Algorithm – Complexity Analysis

For an error pattern $e = \overbrace{\boxed{\text{weight } w-p}}^{n-k}\overbrace{\boxed{\text{weight } p}}^{k}$, we have $\mathcal{P}_\infty = \dfrac{\binom{n-k}{w-p}\binom{k}{p}}{\binom{n}{w}}$

$\mathcal{N}_\infty = \dfrac{\binom{n}{w}}{\binom{n-k}{w-p}\binom{k}{p}}$ and $\mathcal{K} = n(n-k) + \binom{k}{p}$

Never gains more than a polynomial factor over Prange algorithm

$$\boxed{\mathrm{WF}_{\mathrm{LB}}(p) = \mathcal{N}_\infty \cdot \mathcal{K} = \frac{\binom{n}{w}}{\binom{n-k}{w-p}} \left(1 + \frac{n(n-k)}{\binom{k}{p}}\right)} > \frac{\binom{n}{w}}{\binom{n-k}{w-p}} > \frac{\binom{n}{w}}{\binom{n-k}{w}} = \frac{1}{n(n-k)} \mathrm{WF}_{\mathrm{Prange}}$$

Except for extravagant parameters, $p = 2$ is optimal

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