Code-Based Cryptography

Message Attacks (ISD)

Nicolas Sendrier

Code-Based Cryptography

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

Message Attack

A cryptogram for the McEliece encryption scheme has the following form

$$y = xG + e$$

Message Attack

A cryptogram for the McEliece encryption scheme has the following form

$$y = xG + e$$

cryptogram
(or ciphertext)

Message Attack

A cryptogram for the McEliece encryption scheme has the following form

$$y = xG + e$$

cryptogram
(or ciphertext)

message
(or cleartext)

Message Attack

A cryptogram for the McEliece encryption scheme has the following form

$$\underbrace{y}_{\substack{\text{cryptogram} \\ \text{(or ciphertext)}}} = \underbrace{x}_{\substack{\text{message} \\ \text{(or cleartext)}}}\underbrace{G}_{\text{public key}} + e$$

Message Attack

A cryptogram for the McEliece encryption scheme has the following form

$$y = xG + e$$

cryptogram
(or ciphertext)

message
(or cleartext)

public key

random error

Message Attack

A cryptogram for the McEliece encryption scheme has the following form

$$y = xG + e$$

cryptogram
(or ciphertext)

message
(or cleartext)

public key

random error

The adversary knows the cryptogram and the public key
and wishes to recover the message (or equivalently the error)

Message Attack

A cryptogram for the McEliece encryption scheme has the following form

$$y = xG + e$$

cryptogram (or ciphertext) — message (or cleartext) — public key — random error

The adversary knows the cryptogram and the public key
and wishes to recover the message (or equivalently the error)

Only an arbitrary generator matrix is known

$\rightarrow$ **generic decoding problem**

Generic Decoding

In contrast with the usual situation where the code is known in advance, a generic decoder takes a q-ary linear $[n, k]$ code as argument

Generic Decoding

In contrast with the usual situation where the code is known in advance, a generic decoder takes a q-ary linear $[n, k]$ code as argument

$G \in \mathbf{F}_q^{k\times n}$ a generator matrix

$\mathcal{C} = \langle G \rangle = \{xG \mid x \in \mathbf{F}_q^k\}$

Generic Decoding

In contrast with the usual situation where the code is known in advance, a generic decoder takes a q-ary linear $[n, k]$ code as argument

$G \in \mathbf{F}_q^{k \times n}$ a generator matrix
$\mathcal{C} = \langle G \rangle = \{xG \mid x \in \mathbf{F}_q^k\}$

$H \in \mathbf{F}_q^{(n-k) \times n}$ a parity check matrix
$\mathcal{C} = \langle H \rangle^{\perp} = \{c \in \mathbf{F}_q^n \mid cH^T = 0\}$

Generic Decoding

In contrast with the usual situation where the code is known in advance, a generic decoder takes a q-ary linear $[n, k]$ code as argument

$G \in \mathbf{F}_q^{k \times n}$ a generator matrix

$\mathcal{C} = \langle G \rangle = \{xG \mid x \in \mathbf{F}_q^k\}$

$H \in \mathbf{F}_q^{(n-k) \times n}$ a parity check matrix

$\mathcal{C} = \langle H \rangle^{\perp} = \{c \in \mathbf{F}_q^n \mid cH^T = 0\}$

Generic Decoder:

Generic Decoding

In contrast with the usual situation where the code is known in advance, a generic decoder takes a q-ary linear $[n, k]$ code as argument

$G \in \mathbf{F}_q^{k \times n}$ a generator matrix
$\mathcal{C} = \langle G \rangle = \{xG \mid x \in \mathbf{F}_q^k\}$

$H \in \mathbf{F}_q^{(n-k) \times n}$ a parity check matrix
$\mathcal{C} = \langle H \rangle^\perp = \{c \in \mathbf{F}_q^n \mid cH^T = 0\}$

Generic Decoder:

$$\begin{array}{rccc} \Phi : & \mathbf{F}_q^n \times \mathbf{F}_q^{k \times n} & \rightarrow & \mathbf{F}_q^k \\ & (y, G) & \mapsto & x \end{array}$$

Generic Decoding

In contrast with the usual situation where the code is known in advance, a generic decoder takes a q-ary linear $[n, k]$ code as argument

$G \in \mathbf{F}_q^{k\times n}$ a generator matrix
$\mathcal{C} = \langle G \rangle = \{xG \mid x \in \mathbf{F}_q^k\}$

$H \in \mathbf{F}_q^{(n-k)\times n}$ a parity check matrix
$\mathcal{C} = \langle H \rangle^\perp = \{c \in \mathbf{F}_q^n \mid cH^T = 0\}$

Generic Decoder:

$$\Phi : \quad \mathbf{F}_q^n \times \mathbf{F}_q^{k\times n} \quad \to \quad \mathbf{F}_q^k$$

$\Phi(xG + e, G) = x$ if e is "small"

"small" = of small Hamming weight

Generic Decoding

In contrast with the usual situation where the code is known in advance, a generic decoder takes a q-ary linear $[n, k]$ code as argument

$G \in \mathbf{F}_q^{k \times n}$ a generator matrix

$\mathcal{C} = \langle G \rangle = \{xG \mid x \in \mathbf{F}_q^k\}$

$H \in \mathbf{F}_q^{(n-k) \times n}$ a parity check matrix

$\mathcal{C} = \langle H \rangle^\perp = \{c \in \mathbf{F}_q^n \mid cH^T = 0\}$

Generic Decoder:

$$\Phi : \quad \mathbf{F}_q^n \times \mathbf{F}_q^{k \times n} \rightarrow \mathbf{F}_q^k$$

$\Phi(xG + e, G) = x$ if e is "small"

Generic Syndrome Decoder:

$$\begin{array}{rcl} \Psi : \quad \mathbf{F}_q^{n-k} \times \mathbf{F}_q^{(n-k) \times n} & \rightarrow & \mathbf{F}_q^n \\ (s, H) & \mapsto & e \end{array}$$

Generic Decoding

In contrast with the usual situation where the code is known in advance, a generic decoder takes a q-ary linear $[n, k]$ code as argument

$G \in \mathbf{F}_q^{k\times n}$ a generator matrix

$\mathcal{C} = \langle G \rangle = \{xG \mid x \in \mathbf{F}_q^k\}$

$H \in \mathbf{F}_q^{(n-k)\times n}$ a parity check matrix

$\mathcal{C} = \langle H \rangle^\perp = \{c \in \mathbf{F}_q^n \mid cH^T = 0\}$

Generic Decoder:

$$\Phi : \quad \mathbf{F}_q^n \times \mathbf{F}_q^{k\times n} \quad \to \quad \mathbf{F}_q^k$$

$\Phi(xG + e, G) = x$ if e is "small"

Generic Syndrome Decoder:

$$\Psi : \quad \mathbf{F}_q^{n-k} \times \mathbf{F}_q^{(n-k)\times n} \quad \to \quad \mathbf{F}_q^n$$

$\Psi(eH^T, H) = e$ if e is "small"

Generic Decoding

In contrast with the usual situation where the code is known in advance, a generic decoder takes a q-ary linear $[n, k]$ code as argument

$G \in \mathbf{F}_q^{k\times n}$ a generator matrix

$\mathcal{C} = \langle G \rangle = \{xG \mid x \in \mathbf{F}_q^k\}$

$H \in \mathbf{F}_q^{(n-k)\times n}$ a parity check matrix

$\mathcal{C} = \langle H \rangle^\perp = \{c \in \mathbf{F}_q^n \mid cH^T = 0\}$

Generic Decoder:

$$\Phi : \quad \mathbf{F}_q^n \times \mathbf{F}_q^{k\times n} \quad \to \quad \mathbf{F}_q^k$$

$$\Phi(xG + e, G) = x \text{ if } e \text{ is "small"}$$

Generic Syndrome Decoder:

$$\Psi : \quad \mathbf{F}_q^{n-k} \times \mathbf{F}_q^{(n-k)\times n} \quad \to \quad \mathbf{F}_q^n$$

$$\Psi(eH^T, H) = e \text{ if } e \text{ is "small"}$$

Those two kinds of decoders are equivalent

$\rightarrow$ **we will consider only syndrome decoding**

The Syndrome Decoding Problem

Syndrome Decoding Problem

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

The Syndrome Decoding Problem

Syndrome Decoding Problem — NP-hard

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

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

Find w columns of H adding to s (modulo 2)

$$H = \underbrace{\begin{bmatrix} h_1 & h_2 & \cdots & h_n \end{bmatrix}}_{n} \Big\updownarrow n-k \qquad s = \begin{bmatrix} \end{bmatrix}$$

The Syndrome Decoding Problem

Syndrome Decoding Problem — NP-hard

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

Find w columns of H adding to s (modulo 2)

$$H = \underbrace{\begin{bmatrix} h_1 & h_2 & \cdots & h_n \end{bmatrix}}_{n} \updownarrow n-k \qquad s = \begin{bmatrix} \end{bmatrix}$$

Find $1 \leq j_1 < j_2 < \cdots < j_w \leq n$ such that

$$h_{j_1} + h_{j_2} + \cdots + h_{j_w} = s$$

Single Solution *versus* Multiple Solution

Syndrome Decoding Problem

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

Single Solution *versus* Multiple Solution

Syndrome Decoding Problem

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

We denote $\mathrm{CSD}(H, s, w)$ the set of all solutions to the above problem

Single Solution *versus* Multiple Solution

Syndrome Decoding Problem

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

We denote $\mathrm{CSD}(H, s, w)$ the set of all solutions to the above problem
Fix n and k and let w grow

$\longrightarrow \dfrac{\binom{n}{w}}{2^{n-k}}$ solutions on average

$0 \longrightarrow w$

Single Solution *versus* Multiple Solution

Syndrome Decoding Problem

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

We denote $\mathrm{CSD}(H, s, w)$ the set of all solutions to the above problem
Fix n and k and let w grow

$\longrightarrow \dfrac{\binom{n}{w}}{2^{n-k}}$ solutions on average

Gilbert-Varshamov radius $\binom{n}{\tau_{\mathrm{GV}}} = 2^{n-k}$

Single Solution *versus* Multiple Solution

Syndrome Decoding Problem

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

We denote $\mathrm{CSD}(H, s, w)$ the set of all solutions to the above problem
Fix n and k and let w grow

at most one solution $\longleftarrow$ $\longrightarrow$ $\dfrac{\binom{n}{w}}{2^{n-k}}$ solutions on average

0 τ_{GV} w

Gilbert-Varshamov radius $\binom{n}{\tau_{\mathrm{GV}}} = 2^{n-k}$

Single Solution *versus* Multiple Solution

Syndrome Decoding Problem

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

We denote $\mathrm{CSD}(H, s, w)$ the set of all solutions to the above problem
Fix n and k and let w grow

exactly ~~at most~~ one solution $\longleftarrow$ $\longrightarrow$ $\frac{\binom{n}{w}}{2^{n-k}}$ solutions on average

0 — τ_{GV} — w

Gilbert-Varshamov radius $\binom{n}{\tau_{\mathrm{GV}}} = 2^{n-k}$

In cryptanalysis, we only consider situations where $\mathrm{CSD}(H, s, w) \neq \emptyset$

Single Solution *versus* Multiple Solution

Syndrome Decoding Problem

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

We denote $\mathrm{CSD}(H, s, w)$ the set of all solutions to the above problem
Fix n and k and let w grow

exactly
~~at most~~ one solution $\longleftarrow$ $\longrightarrow$ $\dfrac{\binom{n}{w}}{2^{n-k}}$ solutions on average

Gilbert-Varshamov radius $\binom{n}{\tau_{\mathrm{GV}}} = 2^{n-k}$

In cryptanalysis, we only consider situations where $\mathrm{CSD}(H, s, w) \neq \emptyset$

We expect $\approx \max\left(1, \binom{n}{w}/2^{n-k}\right)$ solutions

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

Exhaustive Search

Problem: find w columns of H adding to s (modulo 2)

$$H = \overbrace{\begin{bmatrix} h_1 & h_2 & \cdots & h_n \end{bmatrix}}^{n} \; n-k \qquad s = \begin{bmatrix} \; \end{bmatrix}$$

Exhaustive Search

Problem: find w columns of H adding to s (modulo 2)

$$H = \overbrace{\begin{bmatrix} h_1 & h_2 & \cdots & h_n \end{bmatrix}}^{n} \; n-k \qquad s = \begin{bmatrix} \; \\ \; \end{bmatrix}$$

Answer: enumerate all w-tuples $(j_1, j_2, \cdots, j_w)$ such that $1 \leq j_1 < j_2 < \ldots < j_w \leq n$ and check whether $s + h_{j_1} + h_{j_2} \cdots + h_{j_w} = 0$

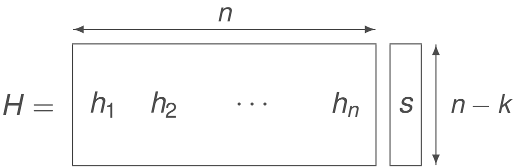
▸ How to enumerate nicely

Enumerate $\{s + eH^T \mid \mathrm{wt}(e) = w\} = \{s + h_{j_1} + \cdots + h_{j_w} \mid 1 \leq j_1 < \ldots < j_w \leq n\}$

$$H = \overbrace{\boxed{h_1 \quad h_2 \quad \cdots \quad h_n}}^{n} \quad \boxed{s} \quad \updownarrow n-k$$

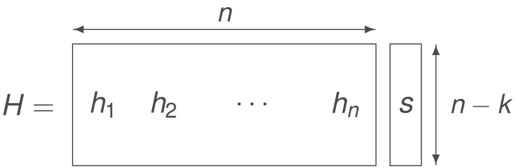
$$\boxed{\text{Enumerate } \{s + eH^T \mid \mathrm{wt}(e) = w\} = \{s + h_{j_1} + \cdots + h_{j_w} \mid 1 \leq j_1 < \ldots < j_w \leq n\}}$$

```
for j1 from 1 to n
```



$\boxed{\text{Enumerate } \{s + eH^T \mid \mathrm{wt}(e) = w\} = \{s + h_{j_1} + \cdots + h_{j_w} \mid 1 \le j_1 < \ldots < j_w \le n\}}$

```
for j1 from 1 to n
    for j2 from j1 + 1 to n
```



$$\boxed{\text{Enumerate } \{s + eH^T \mid \mathrm{wt}(e) = w\} = \{s + h_{j_1} + \cdots + h_{j_w} \mid 1 \leq j_1 < \ldots < j_w \leq n\}}$$

1: `for` j_1 `from` 1 `to` n

2: `for` j_2 `from` $j_1 + 1$ `to` n

$\ddots$

w: `for` j_w `from` $j_{w-1} + 1$ `to` n

$$\begin{array}{cc} & \overset{n}{\longleftrightarrow} \\ H = & \begin{bmatrix} h_1 & h_2 & \cdots & h_n \end{bmatrix} \; \boxed{s} \;\; \updownarrow n-k \end{array}$$

$\boxed{\text{Enumerate } \{s + eH^T \mid \mathrm{wt}(e) = w\} = \{s + h_{j_1} + \cdots + h_{j_w} \mid 1 \le j_1 < \ldots < j_w \le n\}}$

1:     `for` j_1 `from` 1 `to` n

2:        `for` j_2 `from` $j_1 + 1$ `to` n

$$\ddots$$

w: `for` j_w `from` $j_{w-1} + 1$ `to` n

$s_w \leftarrow s + h_{j_1} + h_{j_2} + \cdots + h_{j_w}$

$$H = \overbrace{\begin{bmatrix} h_1 & h_2 & \cdots & h_n \end{bmatrix}}^{n} \begin{bmatrix} s \end{bmatrix} \updownarrow n-k$$

$$\boxed{\text{Enumerate } \{s + eH^T \mid \mathrm{wt}(e) = w\} = \{s + h_{j_1} + \cdots + h_{j_w} \mid 1 \leq j_1 < \ldots < j_w \leq n\}}$$

1: for j_1 from 1 to n

2: for j_2 from $j_1 + 1$ to n

$\ddots$

w: for j_w from $j_{w-1} + 1$ to n

$s_w \leftarrow s + h_{j_1} + h_{j_2} + \cdots + h_{j_w}$

[if $s_w = 0$ then return $(j_1, j_2, \ldots, j_w)$] or [store$(s_w, (j_1, j_2, \ldots, j_w))$]

$$H = \overbrace{\begin{bmatrix} h_1 & h_2 & \cdots & h_n \end{bmatrix}}^{n} \begin{bmatrix} s \end{bmatrix} \; \Big\updownarrow \, n-k$$

Enumerate $\{s + eH^T \mid \mathrm{wt}(e) = w\} = \{s + h_{j_1} + \cdots + h_{j_w} \mid 1 \le j_1 < \ldots < j_w \le n\}$

$$H = \overbrace{\begin{bmatrix} h_1 & h_2 & \cdots & h_n \end{bmatrix}}^{n} \quad \begin{bmatrix} s \end{bmatrix} \; \Big\updownarrow n-k$$

```
1:  for j_1 from 1 to n
2:      for j_2 from j_1 + 1 to n
            ⋱
w:              for j_w from j_{w-1} + 1 to n
                    s_w ← s + h_{j_1} + h_{j_2} + ··· + h_{j_w}
                    [if s_w = 0 then return (j_1, j_2, ..., j_w)] or [store(s_w, (j_1, j_2, ..., j_w))]
```

Total cost is at most $w\binom{n}{w}$ column additions and $\binom{n}{w}$ tests

Enumerate $\{s + eH^T \mid \mathrm{wt}(e) = w\} = \{s + h_{j_1} + \cdots + h_{j_w} \mid 1 \leq j_1 < \ldots < j_w \leq n\}$

$$H = \overbrace{\begin{bmatrix} h_1 & h_2 & \cdots & h_n \end{bmatrix}}^{n} \; \begin{bmatrix} s \end{bmatrix} \quad n-k$$

1: `for` j_1 `from` 1 `to` n

2: `for` j_2 `from` $j_1 + 1$ `to` n

$\ddots$

w: `for` j_w `from` $j_{w-1} + 1$ `to` n

$s_w \leftarrow s + h_{j_1} + h_{j_2} + \cdots + h_{j_w}$

[`if` $s_w = 0$ `then return` $(j_1, j_2, \ldots, j_w)$] or [`store`$(s_w, (j_1, j_2, \ldots, j_w))$]

Total cost is at most $w\binom{n}{w}$ column additions ~~and $\binom{n}{w}$ tests~~

$$\boxed{\text{Enumerate } \{s + eH^T \mid \mathrm{wt}(e) = w\} = \{s + h_{j_1} + \cdots + h_{j_w} \mid 1 \le j_1 < \ldots < j_w \le n\}}$$

1: `for` j_1 `from` 1 `to` n

2: `for` j_2 `from` $j_1 + 1$ `to` n

$\ddots$

w: `for` j_w `from` $j_{w-1} + 1$ `to` n

$s_w \leftarrow s + h_{j_1} + h_{j_2} + \cdots + h_{j_w}$

[`if` $s_w = 0$ `then return` $(j_1, j_2, \ldots, j_w)$] or [`store`$(s_w, (j_1, j_2, \ldots, j_w))$]

$$H = \overbrace{\begin{pmatrix} h_1 & h_2 & \cdots & h_n \end{pmatrix}}^{n} \quad \begin{pmatrix} s \end{pmatrix} \updownarrow n-k$$

Total cost is about $w\binom{n}{w}$ column operations

$\boxed{\text{Enumerate } \{s + eH^T \mid \mathrm{wt}(e) = w\} = \{s + h_{j_1} + \cdots + h_{j_w} \mid 1 \le j_1 < \ldots < j_w \le n\}}$

$$H = \overbrace{\begin{bmatrix} h_1 & h_2 & \cdots & h_n \end{bmatrix}}^{n} \quad \begin{bmatrix} s \end{bmatrix} \; \updownarrow n-k$$

for j_1 from 1 to n

1: $\quad s_1 \leftarrow s + h_{j_1}$

for j_2 from $j_1 + 1$ to n

2:

$\ddots$

for j_w from $j_{w-1} + 1$ to n

w:

[if $s_w = 0$ then return $(j_1, j_2, \ldots, j_w)$] or [store$(s_w, (j_1, j_2, \ldots, j_w))$]

Instead, we may keep track of partial sums

Enumerate $\{s + eH^T \mid \mathrm{wt}(e) = w\} = \{s + h_{j_1} + \cdots + h_{j_w} \mid 1 \leq j_1 < \ldots < j_w \leq n\}$

$$H = \overbrace{\begin{bmatrix} h_1 & h_2 & \cdots & h_n \end{bmatrix}}^{n} \quad \underbrace{\begin{bmatrix} s \end{bmatrix}}_{n-k}$$

`for` j_1 `from` 1 `to` n

1: $s_1 \leftarrow s + h_{j_1}$

`for` j_2 `from` $j_1 + 1$ `to` n

2: $s_2 \leftarrow s_1 + h_{j_2} \quad (= s + h_{j_1} + h_{j_2})$

$\ddots$

`for` j_w `from` $j_{w-1} + 1$ `to` n

w:

[`if` $s_w = 0$ `then return` $(j_1, j_2, \ldots, j_w)$] or [`store`$(s_w, (j_1, j_2, \ldots, j_w))$]

Instead, we may keep track of partial sums

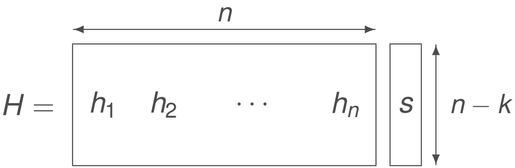
$$\boxed{\text{Enumerate } \{s + eH^T \mid \mathrm{wt}(e) = w\} = \{s + h_{j_1} + \cdots + h_{j_w} \mid 1 \le j_1 < \ldots < j_w \le n\}}$$

$$H = \begin{array}{c} \overset{n}{\longleftrightarrow} \\ \boxed{h_1 \quad h_2 \quad \cdots \quad h_n} \end{array} \; \boxed{s} \; \updownarrow n-k$$

```
     for j_1 from 1 to n
1:       s_1 ← s + h_{j_1}
         for j_2 from j_1 + 1 to n
2:           s_2 ← s_1 + h_{j_2}
                 ⋱
                 for j_w from j_{w-1} + 1 to n
w:                   s_w ← s_{w-1} + h_{j_w}   (= s + h_{j_1} + h_{j_2} + ··· + h_{j_w})
                     [if s_w = 0 then return (j_1, j_2, ..., j_w)] or [store(s_w, (j_1, j_2, ..., j_w))]
```

Instead, we may keep track of partial sums

$$\text{Enumerate } \{s + eH^T \mid \mathrm{wt}(e) = w\} = \{s + h_{j_1} + \cdots + h_{j_w} \mid 1 \le j_1 < \ldots < j_w \le n\}$$



for j_1 from 1 to n

1: $\quad s_1 \leftarrow s + h_{j_1}$

$\quad$ for j_2 from $j_1 + 1$ to n

2: $\quad\quad s_2 \leftarrow s_1 + h_{j_2}$

$\quad\quad\quad \ddots$

$\quad\quad\quad$ for j_w from $j_{w-1} + 1$ to n

w: $\quad\quad\quad\quad s_w \leftarrow s_{w-1} + h_{j_w}$

$\quad\quad\quad\quad$ [if $s_w = 0$ then return $(j_1, j_2, \ldots, j_w)$] or [store$(s_w, (j_1, j_2, \ldots, j_w))$]

Line i is executed about $\binom{n}{i}$ times

$\boxed{\text{Enumerate } \{s + eH^T \mid \mathrm{wt}(e) = w\} = \{s + h_{j_1} + \cdots + h_{j_w} \mid 1 \le j_1 < \ldots < j_w \le n\}}$

$\quad \texttt{for } j_1 \texttt{ from } 1 \texttt{ to } n$

1: $\qquad s_1 \leftarrow s + h_{j_1}$

$\qquad\quad \texttt{for } j_2 \texttt{ from } j_1 + 1 \texttt{ to } n$

2: $\qquad\qquad s_2 \leftarrow s_1 + h_{j_2}$

$\qquad\qquad\qquad \ddots$

$\qquad\qquad\qquad \texttt{for } j_w \texttt{ from } j_{w-1} + 1 \texttt{ to } n$

w: $\qquad\qquad\qquad\quad s_w \leftarrow s_{w-1} + h_{j_w}$

$\qquad\qquad\qquad\quad [\texttt{if } s_w = 0 \texttt{ then return } (j_1, j_2, \ldots, j_w)]$ or $[\texttt{store}(s_w, (j_1, j_2, \ldots, j_w))]$

$$H = \overbrace{\begin{bmatrix} h_1 & h_2 & \cdots & h_n \end{bmatrix}}^{n} \quad \begin{bmatrix} s \end{bmatrix} \;\; \updownarrow n - k$$

Line i is executed about $\binom{n}{i}$ times

$\rightarrow$ total of about $\binom{n}{1} + \binom{n}{2} + \cdots + \binom{n}{w}$ column additions

Enumerate $\{s + eH^T \mid \mathrm{wt}(e) = w\} = \{s + h_{j_1} + \cdots + h_{j_w} \mid 1 \leq j_1 < \ldots < j_w \leq n\}$

$$H = \left[\begin{array}{cccc|c} h_1 & h_2 & \cdots & h_n & s \end{array}\right] \quad \underbrace{}_{n-k \text{ rows}}^{n \text{ columns}}$$

```
   for j1 from 1 to n
1:     s1 <- s + h_j1
       for j2 from j1 + 1 to n
2:         s2 <- s1 + h_j2
               ...
               for jw from j(w-1) + 1 to n
w:                 sw <- s(w-1) + h_jw
                   [if sw = 0 then return (j1, j2, ..., jw)] or [store(sw, (j1, j2, ..., jw))]
```

Line i is executed about $\binom{n}{i}$ times

$\rightarrow$ total of about $\binom{n}{1} + \binom{n}{2} + \cdots + \binom{n}{w}$ column additions

dominated by $\binom{n}{w}$ when w is not too large

Back

Exhaustive Search

Problem: find w columns of H adding to s (modulo 2)

$$H = \begin{bmatrix} \overbrace{h_1 \quad h_2 \quad \cdots \quad h_n}^{n} \end{bmatrix} \Big\updownarrow n-k \qquad s = \begin{bmatrix} \; \\ \; \end{bmatrix}$$

Answer: enumerate all w-tuples $(j_1, j_2, \cdots, j_w)$ such that $1 \leq j_1 < j_2 < \ldots < j_w \leq n$ and check whether $s + h_{j_1} + h_{j_2} \cdots + h_{j_w} = 0$

▸ How to enumerate nicely

Requires *about* $\binom{n}{w}$ column operations

Exhaustive Search

Problem: find w columns of H adding to s (modulo 2)

$$H = \overbrace{\begin{bmatrix} h_1 & h_2 & \cdots & h_n \end{bmatrix}}^{n} \Big\updownarrow n-k \qquad s = \begin{bmatrix} \\ \\ \end{bmatrix}$$

Answer: enumerate all w-tuples $(j_1, j_2, \cdots, j_w)$ such that $1 \leq j_1 < j_2 < \ldots < j_w \leq n$ and check whether $s + h_{j_1} + h_{j_2} \cdots + h_{j_w} = 0$

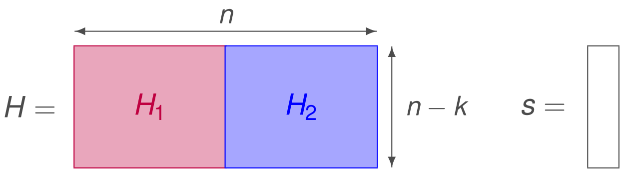
How to enumerate nicely

Requires *about* $\binom{n}{w}$ column operations

Note that we obtain all solutions

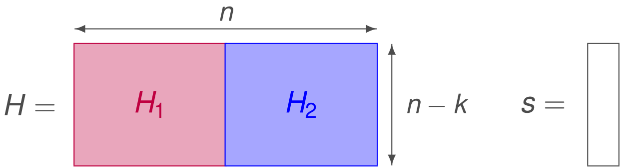
Birthday Decoding

Problem: find w columns of H adding to s (modulo 2)



Birthday Decoding

Problem: find w columns of H adding to s (modulo 2)



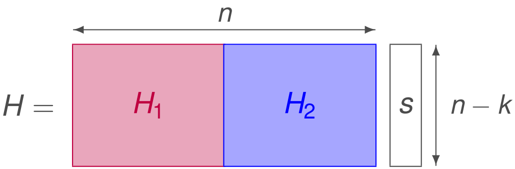
Answer: Split H into two equal parts and enumerate the two following sets

$$\mathcal{L}_1 = \left\{ e_1 H_1^T \mid \mathrm{wt}(e_1) = \frac{w}{2} \right\} \text{ and } \mathcal{L}_2 = \left\{ s + e_2 H_2^T \mid \mathrm{wt}(e_2) = \frac{w}{2} \right\}$$

If $\mathcal{L}_1 \cap \mathcal{L}_2 \neq \emptyset$, we have solution(s): $s + e_1 H_1^T + e_2 H_2^T = 0$

▸ Algorithm

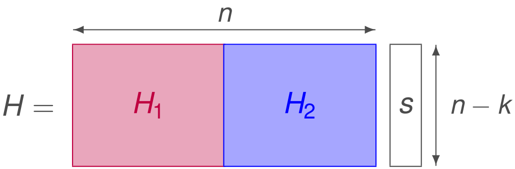
Compute $\mathcal{L}_1 \cap \mathcal{L}_2 = \{e_1 H_1^T \mid \mathrm{wt}(e_1) = \frac{w}{2}\} \cap \{s + e_2 H_2^T \mid \mathrm{wt}(e_2) = \frac{w}{2}\}$



Compute $\mathcal{L}_1 \cap \mathcal{L}_2 = \{e_1 H_1^T \mid \mathrm{wt}(e_1) = \frac{w}{2}\} \cap \{s + e_2 H_2^T \mid \mathrm{wt}(e_2) = \frac{w}{2}\}$

$\texttt{for all}$ e_1 of weight $w/2$

$\quad x \leftarrow e_1 H_1^T$; $T[x] \leftarrow T[x] \cup \{e_1\}$



Total cost: $\binom{n/2}{w/2}$

$|\mathcal{L}_1|$

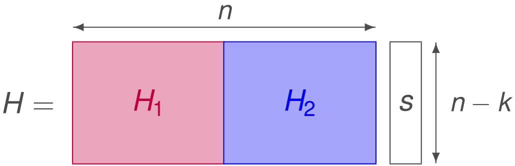
$\boxed{\text{Compute } \mathcal{L}_1 \cap \mathcal{L}_2 = \{e_1 H_1^T \mid \mathrm{wt}(e_1) = \tfrac{w}{2}\} \cap \{s + e_2 H_2^T \mid \mathrm{wt}(e_2) = \tfrac{w}{2}\}}$

`for all` e_1 of weight $w/2$

$\quad x \leftarrow e_1 H_1^T$; $T[x] \leftarrow T[x] \cup \{e_1\}$

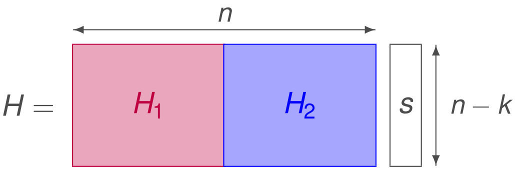
`for all` e_2 of weight $w/2$

$\quad x \leftarrow s + e_2 H_2^T$



Total cost: $\underbrace{\binom{n/2}{w/2}}_{|\mathcal{L}_1|} + \underbrace{\binom{n/2}{w/2}}_{|\mathcal{L}_2|}$

$$\boxed{\text{Compute } \mathcal{L}_1 \cap \mathcal{L}_2 = \{e_1 H_1^T \mid \mathrm{wt}(e_1) = \tfrac{w}{2}\} \cap \{s + e_2 H_2^T \mid \mathrm{wt}(e_2) = \tfrac{w}{2}\}}$$



$\texttt{for all}$ e_1 of weight $w/2$

  $x \leftarrow e_1 H_1^T$; $T[x] \leftarrow T[x] \cup \{e_1\}$

$\texttt{for all}$ e_2 of weight $w/2$

  $x \leftarrow s + e_2 H_2^T$

  $\texttt{for all}$ $e_1 \in T[x]$

    $\mathcal{I} \leftarrow \mathcal{I} \cup \{(e_1, e_2)\}$

Total cost: $\binom{n/2}{w/2} + \binom{n/2}{w/2} + \dfrac{\binom{n/2}{w/2}^2}{2^{n-k}}$

$$|\mathcal{L}_1| \qquad |\mathcal{L}_2| \qquad \frac{|\mathcal{L}_1| \cdot |\mathcal{L}_2|}{2^{n-k}}$$

Compute $\mathcal{L}_1 \cap \mathcal{L}_2 = \{ e_1 H_1^T \mid \mathrm{wt}(e_1) = \frac{w}{2} \} \cap \{ s + e_2 H_2^T \mid \mathrm{wt}(e_2) = \frac{w}{2} \}$

$\texttt{for all}$ e_1 of weight $w/2$

  $x \leftarrow e_1 H_1^T \; ; \; T[x] \leftarrow T[x] \cup \{e_1\}$

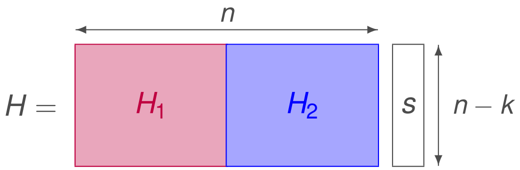
$\texttt{for all}$ e_2 of weight $w/2$

  $x \leftarrow s + e_2 H_2^T$

  $\texttt{for all}$ $e_1 \in T[x]$

    $\mathcal{I} \leftarrow \mathcal{I} \cup \{(e_1, e_2)\}$

$\texttt{return}$ $\mathcal{I}$



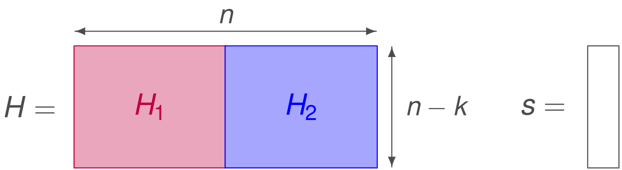
Total cost: $\binom{n/2}{w/2} + \binom{n/2}{w/2} + \frac{\binom{n/2}{w/2}^2}{2^{n-k}}$

$|\mathcal{L}_1| \qquad |\mathcal{L}_2| \qquad \frac{|\mathcal{L}_1| \cdot |\mathcal{L}_2|}{2^{n-k}}$

Back

Birthday Decoding

Problem: find w columns of H adding to s (modulo 2)



Answer: Split H into two equal parts and enumerate the two following sets

$$\mathcal{L}_1 = \left\{ e_1 H_1^T \mid \mathrm{wt}(e_1) = \frac{w}{2} \right\} \text{ and } \mathcal{L}_2 = \left\{ s + e_2 H_2^T \mid \mathrm{wt}(e_2) = \frac{w}{2} \right\}$$

If $\mathcal{L}_1 \cap \mathcal{L}_2 \neq \emptyset$, we have solution(s): $s + e_1 H_1^T + e_2 H_2^T = 0$

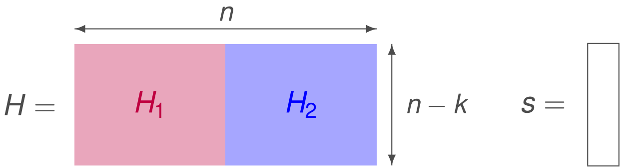
▸ Algorithm

Requires *about* $2\binom{n/2}{w/2} + \dfrac{\binom{n/2}{w/2}^2}{2^{n-k}}$ column operations

Can also be written $2L + L^2/2^{n-k}$ where $L = |\mathcal{L}_1| = |\mathcal{L}_2|$

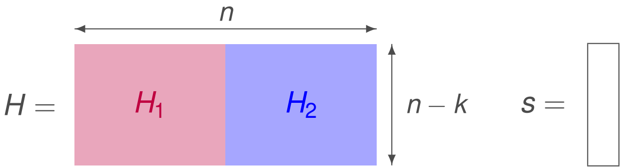
Birthday Decoding – Complexity

Problem: find w columns of H adding to s (modulo 2)



Birthday Decoding – Complexity

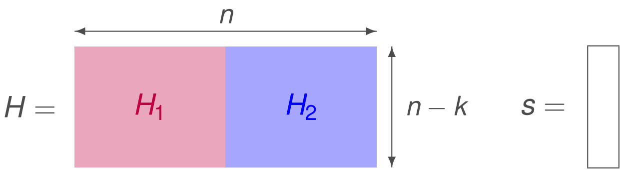
Problem: find w columns of H adding to s (modulo 2)



One particular error of Hamming weight w splits evenly with probability $\mathcal{P} = \dfrac{\binom{n/2}{w/2}^2}{\binom{n}{w}}$

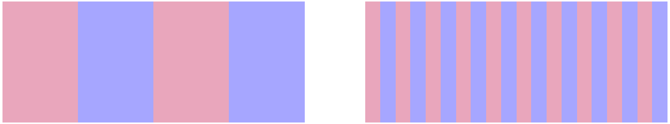
Birthday Decoding – Complexity

Problem: find w columns of H adding to s (modulo 2)



One particular error of Hamming weight w splits evenly with probability $\mathcal{P} = \dfrac{\binom{n/2}{w/2}^2}{\binom{n}{w}}$

We may have to repeat with H divided in several different ways



or more generally by picking the two halves randomly

Birthday Decoding – Complexity

Problem: find w columns of H adding to s (modulo 2)

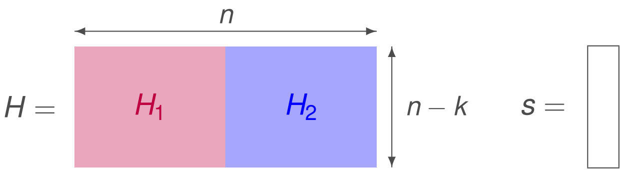
$$H = \overbrace{\begin{bmatrix} H_1 & H_2 \end{bmatrix}}^{n} \; n-k \qquad s = \begin{bmatrix} \; \end{bmatrix}$$

To obtain all solutions:

$$\mathcal{P} = \frac{\binom{n/2}{w/2}^2}{\binom{n}{w}}$$

Birthday Decoding – Complexity

Problem: find w columns of H adding to s (modulo 2)



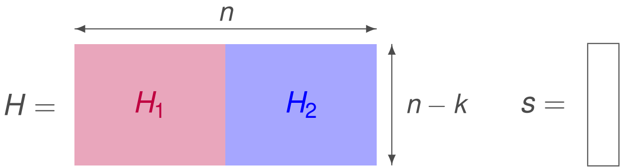
To obtain ~~all~~ most solutions:

repeat with $\approx \dfrac{1}{\mathcal{P}}$ different splitting: $\begin{cases} \text{1. compute } \mathcal{L}_1 \text{ and } \mathcal{L}_2 \\ \text{2. compute } \mathcal{L}_1 \cap \mathcal{L}_2 \end{cases}$

$$\mathcal{P} = \frac{\binom{n/2}{w/2}^2}{\binom{n}{w}}$$

Birthday Decoding – Complexity

Problem: find w columns of H adding to s (modulo 2)



To obtain ~~all~~ most solutions:

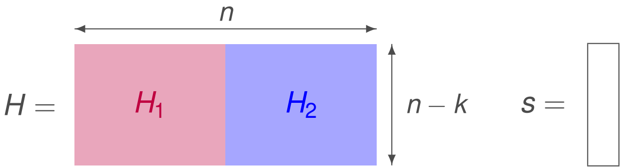
repeat with $\approx \dfrac{1}{\mathcal{P}}$ different splitting: $\begin{cases} 1.\ \text{compute } \mathcal{L}_1 \text{ and } \mathcal{L}_2 \\ 2.\ \text{compute } \mathcal{L}_1 \cap \mathcal{L}_2 \end{cases}$

$$\mathcal{P} = \frac{\binom{n/2}{w/2}^2}{\binom{n}{w}}$$

Total cost $\dfrac{2\binom{n/2}{w/2} + \binom{n/2}{w/2}^2 / 2^{n-k}}{\mathcal{P}} = \dfrac{2\binom{n}{w}}{\binom{n/2}{w/2}} + \dfrac{\binom{n}{w}}{2^{n-k}}$ operations

Birthday Decoding – Complexity

Problem: find w columns of H adding to s (modulo 2)



To obtain ~~all~~ most solutions:

$$\mathcal{P} = \frac{\binom{n/2}{w/2}^2}{\binom{n}{w}}$$

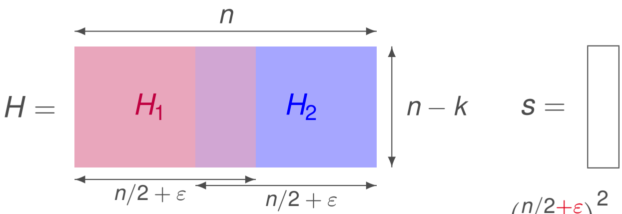
repeat with $\approx \frac{1}{\mathcal{P}}$ different splitting: $\begin{cases} 1.\ \text{compute } \mathcal{L}_1 \text{ and } \mathcal{L}_2 \\ 2.\ \text{compute } \mathcal{L}_1 \cap \mathcal{L}_2 \end{cases}$

Total cost $\dfrac{2\binom{n/2}{w/2} + \binom{n/2}{w/2}^2 / 2^{n-k}}{\mathcal{P}} = \dfrac{2\binom{n}{w}}{\binom{n/2}{w/2}} + \dfrac{\binom{n}{w}}{2^{n-k}}$ operations

$$\approx \sqrt[4]{8\pi w}\sqrt{\binom{n}{w}} + \#\mathit{Solutions}$$

Birthday Decoding – Complexity

Problem: find w columns of H adding to s (modulo 2)



To obtain ~~all~~ most solutions:

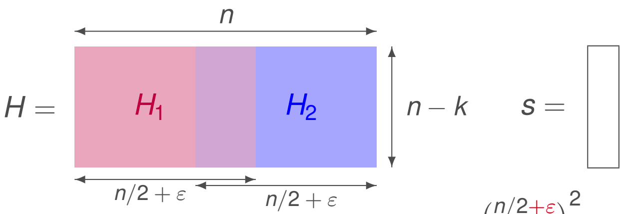
repeat with $\approx \dfrac{1}{\mathcal{P}}$ different splitting: $\left\{ \begin{array}{l} \text{1. compute } \mathcal{L}_1 \text{ and } \mathcal{L}_2 \\ \text{2. compute } \mathcal{L}_1 \cap \mathcal{L}_2 \end{array} \right.$

$$\mathcal{P} = \frac{\binom{n/2+\varepsilon}{w/2}^2}{\binom{n}{w}}$$

Relaxation: allow overlapping $\rightarrow$ H_1 and H_2 are wider by ε

Birthday Decoding – Complexity

Problem: find w columns of H adding to s (modulo 2)



To obtain ~~all~~ most solutions:

repeat with $\approx \dfrac{1}{\mathcal{P}}$ different splitting: $\begin{cases} 1.\ \text{compute } \mathcal{L}_1 \text{ and } \mathcal{L}_2 \\ 2.\ \text{compute } \mathcal{L}_1 \cap \mathcal{L}_2 \end{cases}$

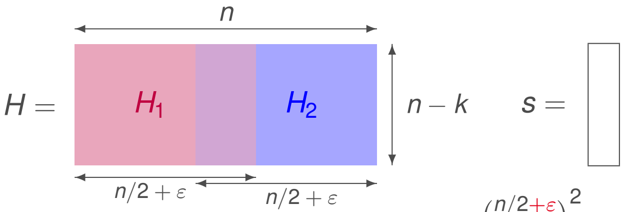
$$\mathcal{P} = \frac{\binom{n/2+\varepsilon}{w/2}^2}{\binom{n}{w}} \approx 1$$

Relaxation: allow overlapping $\rightarrow H_1$ and H_2 are wider by ε

We choose ε such that $\binom{n/2+\varepsilon}{w/2} \approx \sqrt{\binom{n}{w}} \rightarrow$ single repetition

Birthday Decoding – Complexity

Problem: find w columns of H adding to s (modulo 2)



To obtain ~~all~~ most solutions:

repeat with $\approx \frac{1}{\mathcal{P}}$ different splitting: $\begin{cases} \text{1. compute } \mathcal{L}_1 \text{ and } \mathcal{L}_2 \\ \text{2. compute } \mathcal{L}_1 \cap \mathcal{L}_2 \end{cases}$

$$\mathcal{P} = \frac{\binom{n/2+\varepsilon}{w/2}^2}{\binom{n}{w}} \approx 1$$

Relaxation: allow overlapping $\rightarrow$ H_1 and H_2 are wider by ε

We choose ε such that $\binom{n/2+\varepsilon}{w/2} \approx \sqrt{\binom{n}{w}}$ $\rightarrow$ single repetition

Total cost: $2\sqrt{\binom{n}{w}} + \binom{n}{w}/2^{n-k} = 2L + L^2/2^{n-k}$ with $L = \sqrt{\binom{n}{w}}$

(up to a small constant factor)

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

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

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

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 $s' = sU^T = \left[\ \ \right]$

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

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

information set

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

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 $H' = UHP = \begin{array}{c|c} & \text{information set} \\ \hline \begin{matrix} 1 & & \\ & \ddots & \\ & & 1 \end{matrix} & \end{array}$ and $s' = sU^T = \boxed{}$

$$e' = eP = \begin{array}{|c|c|} \hline \text{weight } w & 0 \text{ --- } 0 \\ \hline \end{array}$$

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

$$e' = eP = \begin{array}{|c|c|} \hline s' & 0 \cdots 0 \\ \hline \end{array}$$

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

ISD – Complexity Analysis

We will refer to Information Set Decoding (ISD) to designate is a family of algorithms similar to Prange algorithm

All variants of Information Set Decoding repeat a (large) number of times an independent iteration which has

- a constant (expected) cost $\mathcal{K}$
- a success probability $\mathcal{P}$

  $\rightarrow$ an expected number of iteration $\mathcal{N} = 1/\mathcal{P}$

The workfactor is $\mathcal{N} \cdot \mathcal{K}$

ISD – One Solution or All Solutions?

We consider the problem $\mathrm{CSD}(H, s, w)$ with $H \in \{0,1\}^{(n-k)\times n}$ and $s \in \{0,1\}^{n-k}$

ISD – One Solution or All Solutions?

We consider the problem $\mathrm{CSD}(H, s, w)$ with $H \in \{0,1\}^{(n-k)\times n}$ and $s \in \{0,1\}^{n-k}$

We assume that $\mathrm{CSD}(H, s, w) \neq \emptyset$ (*i.e.* $s \in \{eH^T \mid \mathrm{wt}(e) = w\}$)

$\rightarrow$ **there is always at least one solution**

ISD – One Solution or All Solutions?

We consider the problem $\mathrm{CSD}(H, s, w)$ with $H \in \{0,1\}^{(n-k)\times n}$ and $s \in \{0,1\}^{n-k}$

We assume that $\mathrm{CSD}(H, s, w) \neq \emptyset$ (*i.e.* $s \in \{eH^T \mid \mathrm{wt}(e) = w\}$)

$\rightarrow$ **there is always at least one solution**

1. If $\binom{n}{w} < 2^{n-k}$ (*i.e.* $w < \tau_{\mathrm{GV}}$) there is exactly one solution
2. If $\binom{n}{w} > 2^{n-k}$ (*i.e.* $w > \tau_{\mathrm{GV}}$) there are $\binom{n}{w}/2^{n-k}$ solutions (on average)

ISD – One Solution or All Solutions?

We consider the problem $\mathrm{CSD}(H, s, w)$ with $H \in \{0,1\}^{(n-k)\times n}$ and $s \in \{0,1\}^{n-k}$

We assume that $\mathrm{CSD}(H, s, w) \neq \emptyset$ (*i.e.* $s \in \{eH^T \mid \mathrm{wt}(e) = w\}$)

$\rightarrow$ **there is always at least one solution**

1. If $\binom{n}{w} < 2^{n-k}$ (*i.e.* $w < \tau_{\mathrm{GV}}$) there is exactly one solution
2. If $\binom{n}{w} > 2^{n-k}$ (*i.e.* $w > \tau_{\mathrm{GV}}$) there are $\binom{n}{w}/2^{n-k}$ solutions (on average)

First case (the most common) $\rightarrow$ no difference

ISD – One Solution or All Solutions?

We consider the problem CSD(H, s, w) with $H \in \{0,1\}^{(n-k)\times n}$ and $s \in \{0,1\}^{n-k}$

We assume that CSD$(H, s, w) \neq \emptyset$ (*i.e.* $s \in \{eH^T \mid \mathrm{wt}(e) = w\}$)

$\rightarrow$ **there is always at least one solution**

1. If $\binom{n}{w} < 2^{n-k}$ (*i.e.* $w < \tau_{\mathrm{GV}}$) there is exactly one solution
2. If $\binom{n}{w} > 2^{n-k}$ (*i.e.* $w > \tau_{\mathrm{GV}}$) there are $\binom{n}{w}/2^{n-k}$ solutions (on average)

First case (the most common) $\rightarrow$ no difference

Second case $\rightarrow$ finding only one solution should be easier

(intuitively by a factor $\binom{n}{w}/2^{n-k}$)

ISD – Probabilities

ISD performs many independent iterations. For one iteration, we denote

- $\mathcal{P}_\infty$ the probability to find one specific element of $\mathrm{CSD}(H, s, w)$

ISD – Probabilities

ISD performs many independent iterations. For one iteration, we denote

- $\mathcal{P}_\infty$ the probability to find one specific element of $\mathrm{CSD}(H, s, w)$
- $\mathcal{P}_1$ the probability to find any one element of $\mathrm{CSD}(H, s, w)$

  If $N = |\mathrm{CSD}(H, s, w)|$, we have

  $$\boxed{\mathcal{P}_1 = 1 - (1 - \mathcal{P}_\infty)^N \approx \min(1, N\mathcal{P}_\infty)} \text{ up to a small constant factor}$$

  or simply $\boxed{\mathcal{P}_1 = N\mathcal{P}_\infty}$ if N is not too large (which corresponds to intuition)

ISD – Probabilities

ISD performs many independent iterations. For one iteration, we denote

- $\mathcal{P}_\infty$ the probability to find one specific element of $\mathrm{CSD}(H, s, w)$
- $\mathcal{P}_1$ the probability to find any one element of $\mathrm{CSD}(H, s, w)$

  If $N = |\mathrm{CSD}(H, s, w)|$, we have

  $$\boxed{\mathcal{P}_1 = 1 - (1 - \mathcal{P}_\infty)^N \approx \min(1, N\mathcal{P}_\infty)} \text{ up to a small constant factor}$$

  or simply $\boxed{\mathcal{P}_1 = N\mathcal{P}_\infty}$ if N is not too large (which corresponds to intuition)

For the complexity analysis, there are two situations

- "$w < \tau_{\mathrm{GV}}$" or "$w > \tau_{\mathrm{GV}}$ and we want all solutions"
  $\rightarrow$ we expect to execute $\boxed{\mathcal{N}_\infty = 1/\mathcal{P}_\infty}$ iterations

ISD – Probabilities

ISD performs many independent iterations. For one iteration, we denote

- $\mathcal{P}_\infty$ the probability to find one specific element of $\mathrm{CSD}(H, s, w)$
- $\mathcal{P}_1$ the probability to find any one element of $\mathrm{CSD}(H, s, w)$

  If $N = |\mathrm{CSD}(H, s, w)|$, we have

  $$\boxed{\mathcal{P}_1 = 1 - (1 - \mathcal{P}_\infty)^N \approx \min(1, N\mathcal{P}_\infty)} \text{ up to a small constant factor}$$

  or simply $\boxed{\mathcal{P}_1 = N\mathcal{P}_\infty}$ if N is not too large (which corresponds to intuition)

For the complexity analysis, there are two situations

- "$w < \tau_{\mathrm{GV}}$" or "$w > \tau_{\mathrm{GV}}$ and we want all solutions"
  $\rightarrow$ we expect to execute $\boxed{\mathcal{N}_\infty = 1/\mathcal{P}_\infty}$ iterations
- "$w > \tau_{\mathrm{GV}}$ and we want only one solution"
  $\rightarrow$ we expect to execute $\boxed{\mathcal{N}_1 = \mathcal{N}_\infty / N = \frac{2^{n-k}}{\binom{n}{w}\mathcal{P}_\infty}}$ iterations

Prange Algorithm – Complexity Analysis

An error pattern is found if it has the following form $e = \begin{array}{cc} \overleftrightarrow{\quad n-k \quad} & \overleftrightarrow{\qquad k \qquad} \\ \boxed{\text{weight } w} & \boxed{0 \;\text{——}\; 0} \end{array}$

Prange Algorithm – Complexity Analysis

An error pattern is found if it has the following form $e = \big[\, \overbrace{\text{weight } w}^{n-k} \;\big|\; \overbrace{0 \text{ —— } 0}^{k} \,\big]$

It follows that $\mathcal{P}_\infty = \dfrac{\binom{n-k}{w}}{\binom{n}{w}}$ and $\mathcal{P}_1 = \dfrac{\binom{n-k}{w}}{\min(2^{n-k}, \binom{n}{w})}$

Prange Algorithm – Complexity Analysis

An error pattern is found if it has the following form $e = \big[\,\overbrace{\text{weight } w}^{n-k} \,\big|\, \overbrace{0 \;\text{———}\; 0}^{k}\,\big]$

It follows that $\mathcal{P}_\infty = \dfrac{\binom{n-k}{w}}{\binom{n}{w}}$ and $\mathcal{P}_1 = \dfrac{\binom{n-k}{w}}{\min(2^{n-k}, \binom{n}{w})}$

$\mathcal{K} = n(n-k)$ column operations (the Gaussian elimination dominates)

Prange Algorithm – Complexity Analysis

An error pattern is found if it has the following form $e = \overbrace{\boxed{\text{weight } w}}^{n-k}\overbrace{\boxed{0 \;\text{———}\; 0}}^{k}$

It follows that $\mathcal{P}_\infty = \dfrac{\binom{n-k}{w}}{\binom{n}{w}}$ and $\mathcal{P}_1 = \dfrac{\binom{n-k}{w}}{\min(2^{n-k}, \binom{n}{w})}$

$\mathcal{K} = n(n-k)$ column operations (the Gaussian elimination dominates)

Total workfactor is

- for all solutions $\boxed{\mathrm{WF}_{\text{Prange}} = n(n-k)\dfrac{\binom{n}{w}}{\binom{n-k}{w}}}$
- for one solution $n(n-k)\dfrac{\min(2^{n-k}, \binom{n}{w})}{\binom{n-k}{w}}$

indeed the values are identical when $\binom{n}{w} < 2^{n-k}$

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 $e = \overbrace{\boxed{\text{weight } w-p}}^{n-k}\overbrace{\boxed{\text{weight } p}}^{k}$

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 $UHP = \left[\begin{array}{c|c} \begin{matrix} 1 & & \\ & \ddots & \\ & & 1 \end{matrix} & H' \end{array}\right]$ (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, $0 \le p \le w$
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 $\mathcal{L} = \{sU^T + e'H'^T \mid \mathrm{wt}(e') = p\}$

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 $(s', e')P^{-1}$

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)

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 $e = \overset{\longleftarrow\; n-k \;\longrightarrow}{\boxed{\text{weight } w-p}}\overset{\longleftarrow\; k \;\longrightarrow}{\boxed{\text{weight } p}}$, 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

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

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

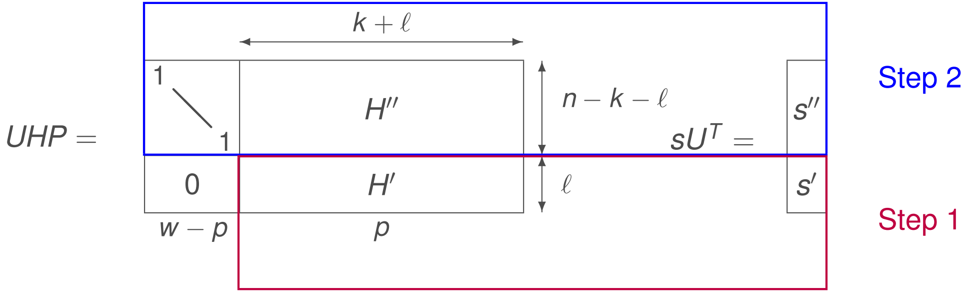
Stern Algorithm – Dumer Algorithm

Idea: combine Lee & Brickell algorithm and birthday decoding

$$UHP = \begin{array}{cc} \begin{array}{cc} & \overleftrightarrow{\quad k+\ell \quad} \end{array} & \\ \left[\begin{array}{c|c} \begin{matrix}1&&\\&\ddots&\\&&1\end{matrix} & H'' \\ \hline 0 & H' \end{array}\right] & \begin{array}{l} \updownarrow\, n-k-\ell \\[2ex] \updownarrow\, \ell \end{array} \\ \begin{array}{cc} w-p & \qquad p \end{array} & \end{array} \qquad sU^T = \left[\begin{array}{c} s'' \\ \hline s' \end{array}\right]$$

Stern Algorithm – Dumer Algorithm

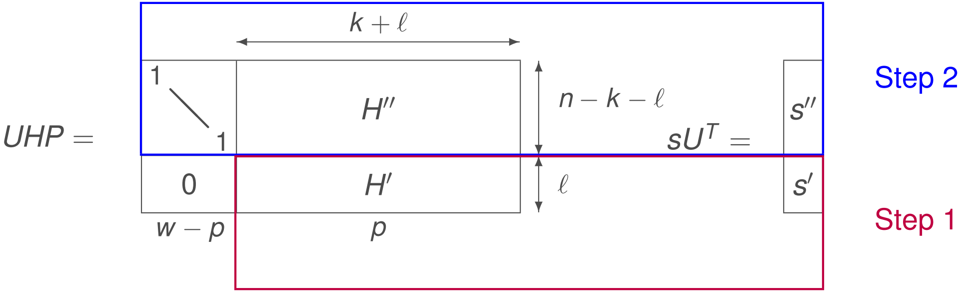
Idea: combine Lee & Brickell algorithm and birthday decoding



Step 1: Find all $e' \in \mathrm{CSD}(H', s', p)$
Step 2: Check $\mathrm{wt}(e' H''^T + s'') = w - p$

Stern Algorithm – Dumer Algorithm

Idea: combine Lee & Brickell algorithm and birthday decoding

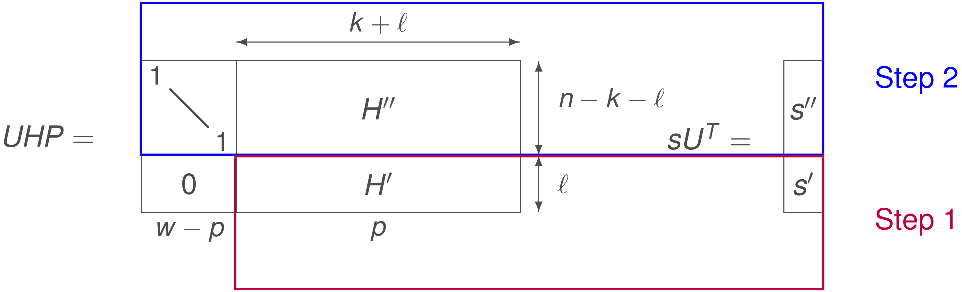


Step 1: Find all $e' \in \mathrm{CSD}(H', s', p)$
Step 2: Check $\mathrm{wt}(e' H''^T + s'') = w - p$

If step 1 is solved by enumeration $\rightarrow$ similar to Lee & Brickell

Stern Algorithm – Dumer Algorithm

Idea: combine Lee & Brickell algorithm and birthday decoding



Step 1: Find all $e' \in \mathrm{CSD}(H', s', p)$
Step 2: Check $\mathrm{wt}(e' H''^T + s'') = w - p$

If step 1 is solved by enumeration $\rightarrow$ similar to Lee & Brickell

If step 1 is solved by birthday decoding $\rightarrow$ Dumer Algorithm

Dumer Algorithm

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

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

Dumer Algorithm

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

repeat:
    pick a permutation matrix P

Dumer Algorithm

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

repeat:
 pick a permutation matrix P
 compute U, H', H'', s', s''

$$UHP = \begin{array}{|c|c|}\hline \begin{matrix}1 & \\ & \ddots \\ & & 1\end{matrix} & H'' \\ \hline 0 & H' \\ \hline \end{array} \; \updownarrow \ell$$

$$sU^T = \begin{array}{|c|}\hline s'' \\ \hline s' \\ \hline \end{array}$$

Dumer Algorithm

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

repeat:
- pick a permutation matrix P
- compute U, H', H'', s', s''
- solve $\mathrm{CSD}(H', s', p)$ *(birthday decoding)*

$$UHP = \left[\begin{array}{c|c} \begin{smallmatrix}1 & & \\ & \ddots & \\ & & 1\end{smallmatrix} & H'' \\ \hline 0 & H' \end{array}\right] \begin{array}{l} \\ \updownarrow \ell \end{array}$$

$$sU^T = \left[\begin{array}{c} s'' \\ \hline s' \end{array}\right]$$

Dumer Algorithm

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

repeat:
- pick a permutation matrix P
- compute U, H', H'', s', s''
- solve $\mathrm{CSD}(H', s', p)$ *(birthday decoding)*
- for all $e' \in \mathrm{CSD}(H', s', p)$
  - $e'' \leftarrow e' H''^T + s''$
  - if $\mathrm{wt}(e'') = w - p$
    - return $(e'', e')P$

$$UHP = \begin{array}{c}\left[\begin{array}{c|c} \begin{matrix}1 & & \\ & \ddots & \\ & & 1\end{matrix} & H'' \\ \hline 0 & H' \end{array}\right] \begin{array}{l} \\ \updownarrow \ell \end{array}\end{array}$$

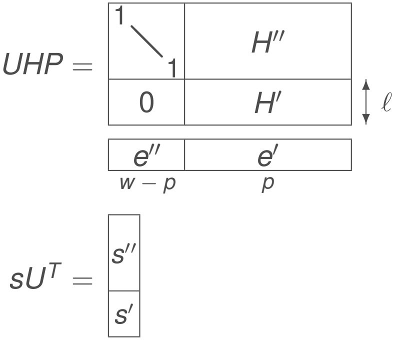
$$\left[\begin{array}{c|c} \underset{w-p}{e''} & \underset{p}{e'} \end{array}\right]$$

$$sU^T = \left[\begin{array}{c} s'' \\ \hline s' \end{array}\right]$$

Dumer Algorithm

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

repeat:
    pick a permutation matrix P
    compute U, H', H'', s', s''
    solve $\mathrm{CSD}(H', s', p)$ *(birthday decoding)*
    for all $e' \in \mathrm{CSD}(H', s', p)$
        $e'' \leftarrow e'H''^T + s''$
        if $\mathrm{wt}(e'') = w - p$
            return $(e'', e')P$



Note: Stern's algorithm (1989) was the first to use birthday decoding, Dumer's algorithm (1991) is only marginally better

We will refer now to the Stern/Dumer Algorithm

Stern/Dumer Algorithm – Complexity Analysis (1/2)

Iteration cost: $\mathcal{K} = n(n-k-\ell) + 2\sqrt{\binom{k+\ell}{p}} + \frac{\binom{k+\ell}{p}}{2^\ell} + \frac{\binom{k+\ell}{p}}{2^\ell}$

Stern/Dumer Algorithm – Complexity Analysis (1/2)

Iteration cost: $\mathcal{K} = \underbrace{n(n-k-\ell)}_{\text{Gaussian elimination}} + 2\sqrt{\binom{k+\ell}{p}} + \frac{\binom{k+\ell}{p}}{2^\ell} + \frac{\binom{k+\ell}{p}}{2^\ell}$

Stern/Dumer Algorithm – Complexity Analysis (1/2)

Iteration cost: $\mathcal{K} = \underbrace{n(n-k-\ell)}_{\text{Gaussian elimination}} + \underbrace{2\sqrt{\binom{k+\ell}{p}} + \frac{\binom{k+\ell}{p}}{2^\ell}}_{\text{Birthday decoding}} + \frac{\binom{k+\ell}{p}}{2^\ell}$

Stern/Dumer Algorithm – Complexity Analysis (1/2)

Iteration cost: $\mathcal{K} = \underbrace{n(n-k-\ell)}_{\text{Gaussian elimination}} + \underbrace{2\sqrt{\binom{k+\ell}{p}} + \frac{\binom{k+\ell}{p}}{2^\ell}}_{\text{Birthday decoding}} + \underbrace{\frac{\binom{k+\ell}{p}}{2^\ell}}_{\text{Final check}}$

Stern/Dumer Algorithm – Complexity Analysis (1/2)

In general, we can write

$$\mathcal{K} = K_0 \cdot n(n-k-\ell) + K_1 \cdot \sqrt{\binom{k+\ell}{p}} + K_2 \cdot \frac{\binom{k+\ell}{p}}{2^\ell}$$

where K_0, K_1, and K_2 are small (implementation dependent) constants

we will set $K_0 = K_1 = K_2 = 1$ to simplify the formula

Stern/Dumer Algorithm – Complexity Analysis (1/2)

We will simply write $\mathcal{K} = n(n-k-\ell) + \sqrt{\binom{k+\ell}{p}} + \frac{\binom{k+\ell}{p}}{2^\ell}$ up to a constant factor

Stern/Dumer Algorithm – Complexity Analysis (1/2)

We will simply write $\mathcal{K} = n(n-k-\ell) + \sqrt{\binom{k+\ell}{p}} + \frac{\binom{k+\ell}{p}}{2^\ell}$ up to a constant factor

Success probability: $\mathcal{P}_\infty = \frac{\binom{k+\ell}{p}\binom{n-k-\ell}{w-p}}{\binom{n}{w}}$ and $\mathcal{N}_\infty = \frac{1}{\mathcal{P}_\infty} = \frac{\binom{n}{w}}{\binom{k+\ell}{p}\binom{n-k-\ell}{w-p}}$

Stern/Dumer Algorithm – Complexity Analysis (1/2)

We will simply write $\mathcal{K} = n(n-k-\ell) + \sqrt{\binom{k+\ell}{p}} + \frac{\binom{k+\ell}{p}}{2^\ell}$ up to a constant factor

Success probability: $\mathcal{P}_\infty = \frac{\binom{k+\ell}{p}\binom{n-k-\ell}{w-p}}{\binom{n}{w}}$ and $\mathcal{N}_\infty = \frac{1}{\mathcal{P}_\infty} = \frac{\binom{n}{w}}{\binom{k+\ell}{p}\binom{n-k-\ell}{w-p}}$

Workfactor $\mathrm{WF_{SD}}(p,\ell) = \mathcal{N}_\infty \cdot \mathcal{K} = \frac{\binom{n}{w}}{\binom{n-k-\ell}{w-p}} \left(\frac{n(n-k+\ell)}{\binom{k+\ell}{p}} + \frac{1}{\sqrt{\binom{k+\ell}{p}}} + \frac{1}{2^\ell} \right)$

Stern/Dumer Algorithm – Complexity Analysis (1/2)

We will simply write $\mathcal{K} = n(n-k-\ell) + \sqrt{\binom{k+\ell}{p}} + \frac{\binom{k+\ell}{p}}{2^\ell}$ up to a constant factor

Success probability: $\mathcal{P}_\infty = \frac{\binom{k+\ell}{p}\binom{n-k-\ell}{w-p}}{\binom{n}{w}}$ and $\mathcal{N}_\infty = \frac{1}{\mathcal{P}_\infty} = \frac{\binom{n}{w}}{\binom{k+\ell}{p}\binom{n-k-\ell}{w-p}}$

$$\text{Workfactor } \mathrm{WF}_{\mathrm{SD}}(p,\ell) = \mathcal{N}_\infty \cdot \mathcal{K} = \frac{\binom{n}{w}}{\binom{n-k-\ell}{w-p}} \left(\frac{n(n-k+\ell)}{\binom{k+\ell}{p}} + \frac{1}{\sqrt{\binom{k+\ell}{p}}} + \frac{1}{2^\ell} \right)$$

(up to a constant factor)

Stern/Dumer Algorithm – Complexity Analysis (1/2)

We will simply write $\mathcal{K} = n(n-k-\ell) + \sqrt{\binom{k+\ell}{p}} + \frac{\binom{k+\ell}{p}}{2^\ell}$ up to a constant factor

Success probability: $\mathcal{P}_\infty = \frac{\binom{k+\ell}{p}\binom{n-k-\ell}{w-p}}{\binom{n}{w}}$ and $\mathcal{N}_\infty = \frac{1}{\mathcal{P}_\infty} = \frac{\binom{n}{w}}{\binom{k+\ell}{p}\binom{n-k-\ell}{w-p}}$

$$\text{Workfactor } \mathrm{WF}_{\mathrm{SD}}(p,\ell) = \mathcal{N}_\infty \cdot \mathcal{K} = \frac{\binom{n}{w}}{\binom{n-k-\ell}{w-p}} \left(\frac{n(n-k+\ell)}{\binom{k+\ell}{p}} + \frac{1}{\sqrt{\binom{k+\ell}{p}}} + \frac{1}{2^\ell} \right)$$

(up to a constant factor)

To be minimized over p and ℓ (positive integers)

Stern/Dumer Algorithm – Complexity Analysis (2/2)

The optimization parameters p and ℓ grow with the problem parameters (n, k, w)

$$\mathrm{WF}_{\mathrm{SD}}(p,\ell) = \frac{\binom{n}{w}}{\binom{n-k-\ell}{w-p}} \left(\frac{n(n-k+\ell)}{\binom{k+\ell}{p}} + \frac{1}{\sqrt{\binom{k+\ell}{p}}} + \frac{1}{2^\ell} \right)$$

Stern/Dumer Algorithm – Complexity Analysis (2/2)

The optimization parameters p and ℓ grow with the problem parameters (n, k, w)

For cryptographic parameters, the Gaussian elimination will never dominate

$$\mathrm{WF}_{\mathrm{SD}}(p, \ell) = \frac{\binom{n}{w}}{\binom{n-k-\ell}{w-p}} \left(\cancel{\frac{n(n-k+\ell)}{\binom{k+\ell}{p}}} + \frac{1}{\sqrt{\binom{k+\ell}{p}}} + \frac{1}{2^\ell} \right)$$

Stern/Dumer Algorithm – Complexity Analysis (2/2)

The optimization parameters p and ℓ grow with the problem parameters (n, k, w)

For cryptographic parameters, the Gaussian elimination will never dominate and we have a good estimate with

$$\mathrm{WF}_{\mathrm{SD}}(p,\ell) = \frac{\binom{n}{w}}{\binom{n-k-\ell}{w-p}} \left(\frac{1}{\sqrt{\binom{k+\ell}{p}}} + \frac{1}{2^\ell} \right)$$

Stern/Dumer Algorithm – Complexity Analysis (2/2)

The optimization parameters p and ℓ grow with the problem parameters (n, k, w)

For cryptographic parameters, the Gaussian elimination will never dominate and we have a good estimate with

$$\mathrm{WF}_{\mathrm{SD}}(p,\ell) = \frac{\binom{n}{w}}{\binom{n-k-\ell}{w-p}} \left(\frac{1}{\sqrt{\binom{k+\ell}{p}}} + \frac{1}{2^\ell} \right)$$

In most situations, the above formula is minimal when the addends are equal

$$\mathrm{WF}_{\mathrm{SD}} = \min_{0 \le p \le w} \frac{\binom{n}{w}}{\binom{n-k-\ell}{w-p}\sqrt{\binom{k+\ell}{p}}} \text{ with } \ell = \log_2 \sqrt{\binom{k+\ell}{p}}$$

Stern/Dumer Algorithm – Complexity Analysis (2/2)

The optimization parameters p and ℓ grow with the problem parameters (n, k, w)

For cryptographic parameters, the Gaussian elimination will never dominate and we have a good estimate with

$$\mathrm{WF}_{\mathrm{SD}}(p,\ell) = \frac{\binom{n}{w}}{\binom{n-k-\ell}{w-p}} \left(\frac{1}{\sqrt{\binom{k+\ell}{p}}} + \frac{1}{2^\ell} \right)$$

In most situations, the above formula is minimal when the addends are equal

$$\mathrm{WF}_{\mathrm{SD}} = \min_{0\le p\le w} \frac{\binom{n}{w}}{\binom{n-k-\ell}{w-p}\sqrt{\binom{k+\ell}{p}}} \text{ with } \ell = \log_2 \sqrt{\binom{k+\ell}{p}}$$

(up to a constant factor)

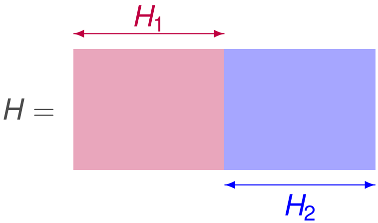
3. Message Attack (ISD)

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

Improved Birthday Decoding

Idea: Use the “representation technique” (Howgrave-Graham and Joux, 2010)

Let $\mathcal{L}_1 = \{e_1 H_1^T \mid \mathrm{wt}(e_1) = \frac{w}{2}\}$ and $\mathcal{L}_2 = \{s + e_2 H_2^T \mid \mathrm{wt}(e_2) = \frac{w}{2}\}$

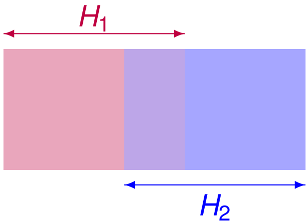


Improved Birthday Decoding

Idea: Use the “representation technique” (Howgrave-Graham and Joux, 2010)

Let $\mathcal{L}_1 = \{e_1 H_1^T \mid \mathrm{wt}(e_1) = \frac{w}{2}\}$ and $\mathcal{L}_2 = \{s + e_2 H_2^T \mid \mathrm{wt}(e_2) = \frac{w}{2}\}$

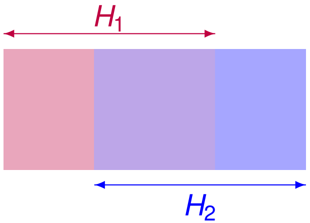
$H =$



Improved Birthday Decoding

Idea: Use the “representation technique” (Howgrave-Graham and Joux, 2010)

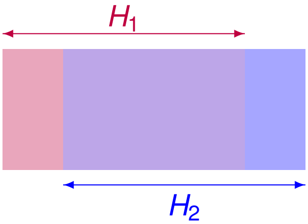
Let $\mathcal{L}_1 = \{e_1 H_1^T \mid \mathrm{wt}(e_1) = \frac{w}{2}\}$ and $\mathcal{L}_2 = \{s + e_2 H_2^T \mid \mathrm{wt}(e_2) = \frac{w}{2}\}$

$H =$ 

Improved Birthday Decoding

Idea: Use the "representation technique" (Howgrave-Graham and Joux, 2010)

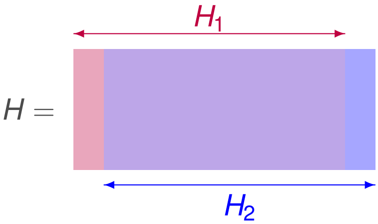
Let $\mathcal{L}_1 = \{e_1 H_1^T \mid \mathrm{wt}(e_1) = \frac{w}{2}\}$ and $\mathcal{L}_2 = \{s + e_2 H_2^T \mid \mathrm{wt}(e_2) = \frac{w}{2}\}$

$H =$ 

Improved Birthday Decoding

Idea: Use the "representation technique" (Howgrave-Graham and Joux, 2010)

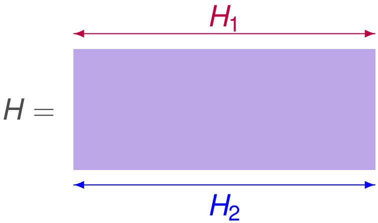
Let $\mathcal{L}_1 = \{e_1 H_1^T \mid \mathrm{wt}(e_1) = \frac{w}{2}\}$ and $\mathcal{L}_2 = \{s + e_2 H_2^T \mid \mathrm{wt}(e_2) = \frac{w}{2}\}$



Improved Birthday Decoding

Idea: Use the "representation technique" (Howgrave-Graham and Joux, 2010)

Let $\mathcal{L}_1 = \left\{e_1 H^T \mid \mathrm{wt}(e_1) = \frac{w}{2}\right\}$ and $\mathcal{L}_2 = \left\{s + e_2 H^T \mid \mathrm{wt}(e_2) = \frac{w}{2}\right\}$



Improved Birthday Decoding

Idea: Use the "representation technique" (Howgrave-Graham and Joux, 2010)

$$\text{Let } \mathcal{L}_1 = \{e_1 H^T \mid \mathrm{wt}(e_1) = \tfrac{w}{2}\} \text{ and } \mathcal{L}_2 = \{s + e_2 H^T \mid \mathrm{wt}(e_2) = \tfrac{w}{2}\}$$

Each $e \in \mathrm{CSD}(H, s, w)$ "represented" $\binom{w}{w/2}$ times as $e = e_1 + e_2$ with $e_1 H^T = s + e_2 H^T \in \mathcal{L}_1 \cap \mathcal{L}_2$

Improved Birthday Decoding

Idea: Use the "representation technique" (Howgrave-Graham and Joux, 2010)

$$\text{Let } \mathcal{L}_1 = \left\{e_1 H^T \mid \mathrm{wt}(e_1) = \tfrac{w}{2}\right\} \text{ and } \mathcal{L}_2 = \left\{s + e_2 H^T \mid \mathrm{wt}(e_2) = \tfrac{w}{2}\right\}$$

Each $e \in \mathrm{CSD}(H, s, w)$ "represented" $\binom{w}{w/2}$ times as $e = e_1 + e_2$ with $e_1 H^T = s + e_2 H^T \in \mathcal{L}_1 \cap \mathcal{L}_2$

We may decimate $\mathcal{L}_1$ and $\mathcal{L}_2$ while keeping the solutions in $\mathcal{L}_1 \cap \mathcal{L}_2$

Improved Birthday Decoding

Idea: Use the "representation technique" (Howgrave-Graham and Joux, 2010)

$$\text{Let } \mathcal{L}_1 = \left\{ e_1 H^T \mid \mathrm{wt}(e_1) = \tfrac{w}{2} \right\} \text{ and } \mathcal{L}_2 = \left\{ s + e_2 H^T \mid \mathrm{wt}(e_2) = \tfrac{w}{2} \right\}$$

Each $e \in \mathrm{CSD}(H, s, w)$ "represented" $\binom{w}{w/2}$ times as $e = e_1 + e_2$ with $e_1 H^T = s + e_2 H^T \in \mathcal{L}_1 \cap \mathcal{L}_2$

We may decimate $\mathcal{L}_1$ and $\mathcal{L}_2$ while keeping the solutions in $\mathcal{L}_1 \cap \mathcal{L}_2$

For any binary vector, let $\phi_r(x)$ denote the last r bits of x, we define

$$\mathcal{L}_1(r) = \left\{ e_1 H^T \mid \mathrm{wt}(e_1) = \tfrac{w}{2}, \phi_r(e_1 H^T) = 0 \right\}$$

$$\mathcal{L}_2(r) = \left\{ s + e_2 H^T \mid \mathrm{wt}(e_2) = \tfrac{w}{2}, \phi_r(s + e_2 H^T) = 0 \right\}$$

Improved Birthday Decoding

Idea: Use the “representation technique” (Howgrave-Graham and Joux, 2010)

$$\text{Let } \mathcal{L}_1 = \left\{ e_1 H^T \mid \mathrm{wt}(e_1) = \tfrac{w}{2} \right\} \text{ and } \mathcal{L}_2 = \left\{ s + e_2 H^T \mid \mathrm{wt}(e_2) = \tfrac{w}{2} \right\}$$

Each $e \in \mathrm{CSD}(H, s, w)$ “represented” $\binom{w}{w/2}$ times as $e = e_1 + e_2$ with $e_1 H^T = s + e_2 H^T \in \mathcal{L}_1 \cap \mathcal{L}_2$

We may decimate $\mathcal{L}_1$ and $\mathcal{L}_2$ while keeping the solutions in $\mathcal{L}_1 \cap \mathcal{L}_2$

For any binary vector, let $\phi_r(x)$ denote the last r bits of x, we define

$$\mathcal{L}_1(r) = \left\{ e_1 H^T \mid \mathrm{wt}(e_1) = \tfrac{w}{2}, \phi_r(e_1 H^T) = 0 \right\}$$

$$\mathcal{L}_2(r) = \left\{ s + e_2 H^T \mid \mathrm{wt}(e_2) = \tfrac{w}{2}, \phi_r(s + e_2 H^T) = 0 \right\}$$

Claim: if $2^r = \binom{w}{w/2}$ then any $e \in \mathrm{CSD}(H, s, w)$ is “represented in $\mathcal{L}_1(r) \cap \mathcal{L}_2(r)$” with probability $> 1/2$

Improved Birthday Decoding – Algorithm

$$H = \begin{bmatrix} H'' \\ \hline H' \end{bmatrix} \quad s = \begin{bmatrix} s'' \\ \hline s' \end{bmatrix}$$

r

Improved Birthday Decoding – Algorithm

$\texttt{for all } e_1 \in \boxed{\mathrm{CSD}(H', 0, w/2)}$

$\quad x \leftarrow e_1 H''^T \; ; \; T[x] \leftarrow T[x] \cup \{e_1\}$

$$H = \begin{bmatrix} H'' \\ \hline H' \end{bmatrix} \qquad s = \begin{bmatrix} s'' \\ \hline s' \end{bmatrix}$$

r

$$\sqrt{\binom{n}{w/2}} + \frac{\binom{n}{w/2}}{2^r}$$

first recursive call to CSD

solved by birthday decoding with complexity $\sqrt{\binom{n}{w/2}} + \frac{\binom{n}{w/2}}{2^r}$

Improved Birthday Decoding – Algorithm

$\texttt{for all } e_1 \in \mathrm{CSD}(H', 0, w/2)$

$\quad x \leftarrow e_1 H''^T \; ; \; T[x] \leftarrow T[x] \cup \{e_1\}$

$\texttt{for all } e_2 \in \boxed{\mathrm{CSD}(H', s', w/2)}$

$\quad x \leftarrow s + e_2 H''^T$

$$H = \begin{array}{|c|} \hline H'' \\ \hline H' \\ \hline \end{array} \qquad s = \begin{array}{|c|} \hline s'' \\ \hline s' \\ \hline \end{array}$$

r

$$\sqrt{\binom{n}{w/2}} + \frac{\binom{n}{w/2}}{2^r} + \sqrt{\binom{n}{w/2}} + \frac{\binom{n}{w/2}}{2^r}$$

second recursive call to CSD

solved with birthday decoding with complexity $\sqrt{\binom{n}{w/2}} + \frac{\binom{n}{w/2}}{2^r}$

Improved Birthday Decoding – Algorithm

$\texttt{for all } e_1 \in \mathrm{CSD}(H', 0, w/2)$

$\quad x \leftarrow e_1 H''^T \; ; \; T[x] \leftarrow T[x] \cup \{e_1\}$

$\texttt{for all } e_2 \in \mathrm{CSD}(H', s', w/2)$

$\quad x \leftarrow s + e_2 H''^T$

$\quad \texttt{for all } e_1 \in T[x]$

$\quad\quad \mathcal{I} \leftarrow \mathcal{I} \cup \{(e_1, e_2)\}$

$$H = \begin{array}{r} \\ r \updownarrow \end{array}\begin{bmatrix} H'' \\ \hline H' \end{bmatrix} \qquad s = \begin{bmatrix} s'' \\ \hline s' \end{bmatrix}$$

$$\sqrt{\binom{n}{w/2}} + \frac{\binom{n}{w/2}}{2^r} + \sqrt{\binom{n}{w/2}} + \frac{\binom{n}{w/2}}{2^r} + \frac{\binom{n}{w/2}^2}{2^{n-k+r}}$$

Keep the syndromes matching on the first $n - k - r$ bits

There are $\left(\frac{\binom{n}{w/2}}{2^r}\right)^2 \frac{1}{2^{n-k-r}}$ such syndromes and as many solutions

Improved Birthday Decoding – Algorithm

for all $e_1 \in \mathrm{CSD}(H', 0, w/2)$
  $x \leftarrow e_1 H''^T$; $T[x] \leftarrow T[x] \cup \{e_1\}$
for all $e_2 \in \mathrm{CSD}(H', s', w/2)$
  $x \leftarrow s + e_2 H''^T$
  for all $e_1 \in T[x]$
    $\mathcal{I} \leftarrow \mathcal{I} \cup \{(e_1, e_2)\}$
return $\mathcal{I}$

$$H = \begin{array}{c} \\ r \updownarrow \end{array} \begin{bmatrix} H'' \\ \hline H' \end{bmatrix} \qquad s = \begin{bmatrix} s'' \\ \hline s' \end{bmatrix}$$

$$\sqrt{\binom{n}{w/2}} + \frac{\binom{n}{w/2}}{2^r} + \frac{\binom{n}{w/2}^2}{2^{n-k+r}} \text{ column operations}$$

Improved Birthday Decoding – Algorithm

$\texttt{for all } e_1 \in \mathrm{CSD}(H', 0, w/2)$
$\quad x \leftarrow e_1 H''^T\ ;\ T[x] \leftarrow T[x] \cup \{e_1\}$
$\texttt{for all } e_2 \in \mathrm{CSD}(H', s', w/2)$
$\quad x \leftarrow s + e_2 H''^T$
$\quad \texttt{for all } e_1 \in T[x]$
$\quad\quad \mathcal{I} \leftarrow \mathcal{I} \cup \{(e_1, e_2)\}$
$\texttt{return } \mathcal{I}$

$$H = \begin{array}{c} \\ r\updownarrow \end{array}\begin{bmatrix} H'' \\ \hline H' \end{bmatrix} \qquad s = \begin{bmatrix} s'' \\ \hline s' \end{bmatrix}$$

$$\sqrt{\binom{n}{w/2}} + \frac{\binom{n}{w/2}}{2^r} + \frac{\binom{n}{w/2}^2}{2^{n-k+r}} \text{ column operations}$$

Replacing $2^r = \binom{w}{w/2}$ and using the identity $\dfrac{\binom{n}{w/2}}{\binom{w}{w/2}} = \dfrac{\binom{n}{w}}{\binom{n-w/2}{w/2}}$

Improved Birthday Decoding – Algorithm

$\texttt{for all } e_1 \in \mathrm{CSD}(H', 0, w/2)$
$\quad x \leftarrow e_1 H''^T \ ; \ T[x] \leftarrow T[x] \cup \{e_1\}$
$\texttt{for all } e_2 \in \mathrm{CSD}(H', s', w/2)$
$\quad x \leftarrow s + e_2 H''^T$
$\quad \texttt{for all } e_1 \in T[x]$
$\quad\quad \mathcal{I} \leftarrow \mathcal{I} \cup \{(e_1, e_2)\}$
$\texttt{return } \mathcal{I}$

$$H = \begin{array}{c} \\ r\updownarrow \end{array}\begin{bmatrix} H'' \\ \hline H' \end{bmatrix} \qquad s = \begin{bmatrix} s'' \\ \hline s' \end{bmatrix}$$

$$\sqrt{\binom{n}{w/2}} + \frac{\binom{n}{w/2}}{2^r} + \frac{\binom{n}{w/2}^2}{2^{n-k+r}} \text{ column operations}$$

Replacing $2^r = \binom{w}{w/2}$ and using the identity $\dfrac{\binom{n}{w/2}}{\binom{w}{w/2}} = \dfrac{\binom{n}{w}}{\binom{n-w/2}{w/2}}$

$$\sqrt{\binom{n}{w/2}} + \boxed{\frac{\binom{n}{w}}{\binom{n-w/2}{w/2}}} + \frac{\binom{n}{w}}{2^{n-k}} \frac{\binom{n}{w/2}}{\binom{n-w/2}{w/2}}$$

Improved Birthday Decoding – Algorithm

for all $e_1 \in \mathrm{CSD}(H', 0, w/2)$
    $x \leftarrow e_1 H''^T$; $T[x] \leftarrow T[x] \cup \{e_1\}$
for all $e_2 \in \mathrm{CSD}(H', s', w/2)$
    $x \leftarrow s + e_2 H''^T$
    for all $e_1 \in T[x]$
        $\mathcal{I} \leftarrow \mathcal{I} \cup \{(e_1, e_2)\}$
return $\mathcal{I}$

$$H = \begin{array}{c} \\ r \updownarrow \end{array} \begin{bmatrix} H'' \\ \hline H' \end{bmatrix} \qquad s = \begin{bmatrix} s'' \\ \hline s' \end{bmatrix}$$

$$\sqrt{\binom{n}{w/2}} + \frac{\binom{n}{w/2}}{2^r} + \frac{\binom{n}{w/2}^2}{2^{n-k+r}} \text{ column operations}$$

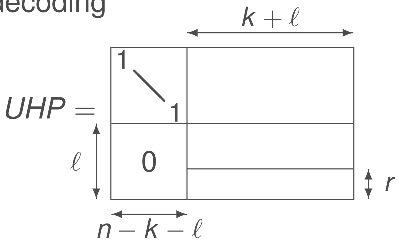
Replacing $2^r = \binom{w}{w/2}$ and using the identity $\dfrac{\binom{n}{w/2}}{\binom{w}{w/2}} = \dfrac{\binom{n}{w}}{\binom{n-w/2}{w/2}}$

$$\sqrt{\binom{n}{w/2}} + \boxed{\frac{\binom{n}{w}}{\binom{n-w/2}{w/2}}} + \frac{\binom{n}{w}}{2^{n-k}} \frac{\binom{n}{w/2}}{\binom{n-w/2}{w/2}}$$

Asymptotically, we have $\sqrt{\dfrac{\binom{n}{w}}{2^w}} \cdot 2^{o(w)}$ and we essentially gain a factor $2^{w/2}$

May, Meurer, and Thomae Algorithm

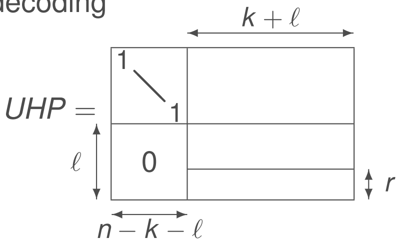
Idea: Dumer Algorithm with the improved birthday decoding



May, Meurer, and Thomae Algorithm

Idea: Dumer Algorithm with the improved birthday decoding

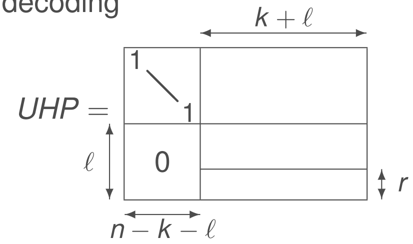
Number of iterations $\mathcal{N}_\infty = \frac{\binom{n}{w}}{\binom{n-k-\ell}{w-p}\binom{k+\ell}{p}}$



May, Meurer, and Thomae Algorithm

Idea: Dumer Algorithm with the improved birthday decoding

Number of iterations $\mathcal{N}_\infty = \dfrac{\binom{n}{w}}{\binom{n-k-\ell}{w-p}\binom{k+\ell}{p}}$



Iteration cost

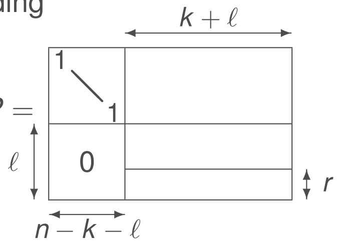
$$\mathcal{K} = n(n-k-\ell) + \sqrt{\binom{k+\ell}{p/2}} + \frac{\binom{k+\ell}{p}}{\binom{k+\ell-p/2}{p/2}} + \frac{\binom{k+\ell}{p}}{2^\ell}\frac{\binom{k+\ell}{p/2}}{\binom{k+\ell-p/2}{p/2}}$$

May, Meurer, and Thomae Algorithm

Idea: Dumer Algorithm with the improved birthday decoding

Number of iterations $\mathcal{N}_\infty = \dfrac{\binom{n}{w}}{\binom{n-k-\ell}{w-p}\binom{k+\ell}{p}}$

$UHP =$



Iteration cost

$$\mathcal{K} = \cancel{n(n-k-\ell)} + \cancel{\sqrt{\binom{k+\ell}{p/2}}} + \frac{\binom{k+\ell}{p}}{\binom{k+\ell-p/2}{p/2}} + \frac{\binom{k+\ell}{p}}{2^\ell}\frac{\binom{k+\ell}{p/2}}{\binom{k+\ell-p/2}{p/2}}$$

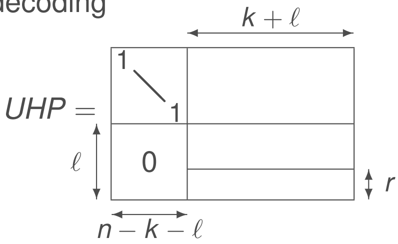
First two terms can be neglected (to be checked *a posteriori*)

May, Meurer, and Thomae Algorithm

Idea: Dumer Algorithm with the improved birthday decoding

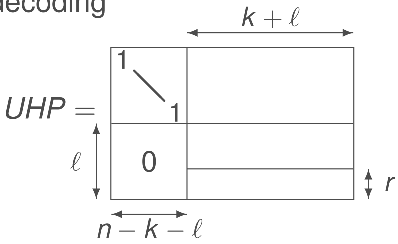
Number of iterations $\mathcal{N}_\infty = \dfrac{\binom{n}{w}}{\binom{n-k-\ell}{w-p}\binom{k+\ell}{p}}$

Iteration cost $\mathcal{K} = \dfrac{\binom{k+\ell}{p}}{\binom{k+\ell-p/2}{p/2}} + \dfrac{\binom{k+\ell}{p}}{2^\ell}\dfrac{\binom{k+\ell}{p/2}}{\binom{k+\ell-p/2}{p/2}}$



May, Meurer, and Thomae Algorithm

Idea: Dumer Algorithm with the improved birthday decoding



Number of iterations $\mathcal{N}_\infty = \dfrac{\binom{n}{w}}{\binom{n-k-\ell}{w-p}\binom{k+\ell}{p}}$

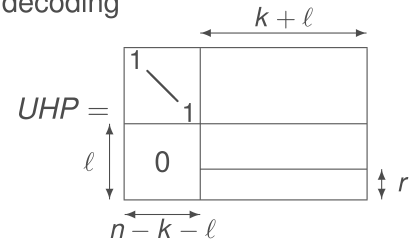
Iteration cost $\mathcal{K} = \dfrac{\binom{k+\ell}{p}}{\binom{k+\ell-p/2}{p/2}} + \dfrac{\binom{k+\ell}{p}}{2^\ell}\dfrac{\binom{k+\ell}{p/2}}{\binom{k+\ell-p/2}{p/2}}$

Workfactor is $\mathcal{N}_\infty \cdot \mathcal{K} = \dfrac{\binom{n}{w}}{\binom{n-k-\ell}{w-p}}\left(\dfrac{1}{\binom{k+\ell-p/2}{p/2}} + \dfrac{1}{2^\ell}\dfrac{\binom{k+\ell}{p/2}}{\binom{k+\ell-p/2}{p/2}}\right)$

minimal when the two terms are equal, *i.e.* $2^\ell = \binom{k+\ell}{p/2}$

May, Meurer, and Thomae Algorithm

Idea: Dumer Algorithm with the improved birthday decoding



Number of iterations $\mathcal{N}_\infty = \dfrac{\binom{n}{w}}{\binom{n-k-\ell}{w-p}\binom{k+\ell}{p}}$

Iteration cost $\mathcal{K} = \dfrac{\binom{k+\ell}{p}}{\binom{k+\ell-p/2}{p/2}} + \dfrac{\binom{k+\ell}{p}}{2^\ell}\dfrac{\binom{k+\ell}{p/2}}{\binom{k+\ell-p/2}{p/2}}$

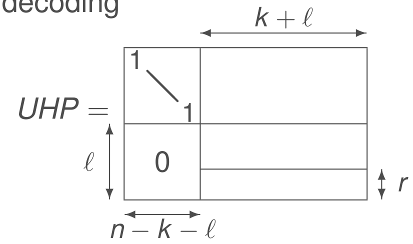
$$\mathrm{WF}_{\mathrm{MMT}} = \min_p \frac{\binom{n}{w}}{\binom{n-k-\ell}{w-p}\binom{k+\ell-p/2}{p/2}} \text{ with } \ell = \log_2 \binom{k+\ell}{p/2}$$

May, Meurer, and Thomae Algorithm

Idea: Dumer Algorithm with the improved birthday decoding

Number of iterations $\mathcal{N}_\infty = \dfrac{\binom{n}{w}}{\binom{n-k-\ell}{w-p}\binom{k+\ell}{p}}$

Iteration cost $\mathcal{K} = \dfrac{\binom{k+\ell}{p}}{\binom{k+\ell-p/2}{p/2}} + \dfrac{\binom{k+\ell}{p}}{2^\ell}\dfrac{\binom{k+\ell}{p/2}}{\binom{k+\ell-p/2}{p/2}}$



$$\mathrm{WF}_{\mathrm{MMT}} = \min_p \frac{\binom{n}{w}}{\binom{n-k-\ell}{w-p}\binom{k+\ell-p/2}{p/2}} \text{ with } \ell = \log_2 \binom{k+\ell}{p/2}$$

Asymptotic gain $\approx 2^{p/2}$ compared with Dumer's algorithm

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

Further Improvement of Birthday Decoding

$$\mathcal{L}_1(r, \varepsilon) = \{e_1 H^T \mid \mathrm{wt}(e_1) = \tfrac{w}{2} + \varepsilon, \phi_r(e_1 H^T) = 0\}$$

$$\mathcal{L}_2(r, \varepsilon) = \{s + e_2 H^T \mid \mathrm{wt}(e_2) = \tfrac{w}{2} + \varepsilon, \phi_r(s + e_2 H^T) = 0\}$$

Further Improvement of Birthday Decoding

$$\mathcal{L}_1(r,\varepsilon) = \{e_1 H^T \mid \mathrm{wt}(e_1) = \tfrac{w}{2} + \varepsilon, \phi_r(e_1 H^T) = 0\}$$
$$\mathcal{L}_2(r,\varepsilon) = \{s + e_2 H^T \mid \mathrm{wt}(e_2) = \tfrac{w}{2} + \varepsilon, \phi_r(s + e_2 H^T) = 0\}$$

Idea: two words of weight $\frac{w}{2}$ and length n are expected to have

$$\begin{cases} \frac{w^2}{4n} \text{ non-zero positions in common} \\ \text{a sum of weight } w - \frac{w^2}{2n} \end{cases}$$

Further Improvement of Birthday Decoding

$$\mathcal{L}_1(r,\varepsilon) = \{e_1 H^T \mid \mathrm{wt}(e_1) = \tfrac{w}{2} + \varepsilon, \phi_r(e_1 H^T) = 0\}$$
$$\mathcal{L}_2(r,\varepsilon) = \{s + e_2 H^T \mid \mathrm{wt}(e_2) = \tfrac{w}{2} + \varepsilon, \phi_r(s + e_2 H^T) = 0\}$$

Idea: if $\varepsilon = \frac{(w/2+\varepsilon)^2}{n}$, two words of weight $\frac{w}{2} + \varepsilon$ and length n are expected to have

$$\begin{cases} \varepsilon \text{ non-zero positions in common} \\ \text{a sum of weight } w \end{cases}$$

Further Improvement of Birthday Decoding

$$\mathcal{L}_1(r,\varepsilon) = \{e_1 H^T \mid \mathrm{wt}(e_1) = \tfrac{w}{2} + \varepsilon, \phi_r(e_1 H^T) = 0\}$$
$$\mathcal{L}_2(r,\varepsilon) = \{s + e_2 H^T \mid \mathrm{wt}(e_2) = \tfrac{w}{2} + \varepsilon, \phi_r(s + e_2 H^T) = 0\}$$

Idea: if $\varepsilon = \frac{(w/2+\varepsilon)^2}{n}$, two words of weight $\frac{w}{2} + \varepsilon$ and length n are expected to have

$$\begin{cases} \varepsilon \text{ non-zero positions in common} \\ \text{a sum of weight } w \end{cases}$$

Note also that there are $\binom{w}{w/2}\binom{n-w}{\varepsilon}$ different ways to write $e = e_1 + e_2$ with $\mathrm{wt}(e) = w$ and $\mathrm{wt}(e_1) = \mathrm{wt}(e_2) = \frac{w}{2} + \varepsilon$

Further Improvement of Birthday Decoding

$$\mathcal{L}_1(r,\varepsilon) = \{e_1 H^T \mid \mathrm{wt}(e_1) = \tfrac{w}{2} + \varepsilon, \phi_r(e_1 H^T) = 0\}$$

$$\mathcal{L}_2(r,\varepsilon) = \{s + e_2 H^T \mid \mathrm{wt}(e_2) = \tfrac{w}{2} + \varepsilon, \phi_r(s + e_2 H^T) = 0\}$$

Claim: Let $2^r = \binom{w}{w/2}\binom{n-w}{\varepsilon}$ and $\varepsilon = \frac{(w/2+\varepsilon)^2}{n}$
Any $e \in \mathrm{CSD}(H, s, w)$ is "represented in $\mathcal{L}_1(r,\varepsilon) \cap \mathcal{L}_2(r,\varepsilon)$" with probability $> 1/2$

Further Improvement of Birthday Decoding

$$\mathcal{L}_1(r,\varepsilon) = \{e_1 H^T \mid \mathrm{wt}(e_1) = \tfrac{w}{2} + \varepsilon, \phi_r(e_1 H^T) = 0\}$$
$$\mathcal{L}_2(r,\varepsilon) = \{s + e_2 H^T \mid \mathrm{wt}(e_2) = \tfrac{w}{2} + \varepsilon, \phi_r(s + e_2 H^T) = 0\}$$

Claim: Let $2^r = \binom{w}{w/2}\binom{n-w}{\varepsilon}$ and $\varepsilon = \frac{(w/2+\varepsilon)^2}{n}$
Any $e \in \mathrm{CSD}(H, s, w)$ is "represented in $\mathcal{L}_1(r,\varepsilon) \cap \mathcal{L}_2(r,\varepsilon)$" with probability $> 1/2$

Workfactor "simplifies" to

$$\sqrt{\binom{n}{w/2+\varepsilon}} + \frac{\binom{n}{w}}{\binom{n}{w/2+\varepsilon}} + \frac{\binom{n}{w}}{2^{n-k}}$$

(up to a polynomial factor)

Impact on MMT Algorithm Complexity

Instead of

$$\mathrm{WF}_{\mathrm{MMT}} = \min_p \frac{\binom{n}{w}}{\binom{n-k-\ell}{w-p}\binom{k+\ell-p/2}{p/2}} \text{ with } \ell = \log_2 \binom{k+\ell}{p/2}$$

(up to a constant factor)

Impact on MMT Algorithm Complexity

Instead of

$$\mathrm{WF}_{\mathrm{MMT}} = \min_p \frac{\binom{n}{w}}{\binom{n-k-\ell}{w-p}\binom{k+\ell-p/2}{p/2}} \text{ with } \ell = \log_2 \binom{k+\ell}{p/2}$$

(up to a constant factor)

We set $\varepsilon = \frac{(w/2+\varepsilon)^2}{n}$, and the workfactor reduces to

$$\mathrm{WF} = \min_p \frac{\binom{n}{w}}{\binom{n-k-\ell}{w-p}\binom{k+\ell}{p/2+\varepsilon}} \text{ with } \ell = \log_2 \binom{k+\ell}{p/2+\varepsilon}$$

(up to a polynomial factor)

Impact on MMT Algorithm Complexity

Instead of

$$\mathrm{WF}_{\mathrm{MMT}} = \min_p \frac{\binom{n}{w}}{\binom{n-k-\ell}{w-p}\binom{k+\ell-p/2}{p/2}} \text{ with } \ell = \log_2 \binom{k+\ell}{p/2}$$

(up to a constant factor)

We set $\varepsilon = \frac{(w/2+\varepsilon)^2}{n}$, and the workfactor reduces to

$$\mathrm{WF} = \min_p \frac{\binom{n}{w}}{\binom{n-k-\ell}{w-p}\binom{k+\ell}{p/2+\varepsilon}} \text{ with } \ell = \log_2 \binom{k+\ell}{p/2+\varepsilon}$$

(up to a polynomial factor)

This is the embryo of the next improvement of ISD

Becker, Joux, May, and Meurer Algorithm (1/2)

Idea: what happens if we let ε grows (much) beyond $w^2/4n$?

$$\mathcal{L}_1(r,\varepsilon) = \left\{e_1 H^T \mid \mathrm{wt}(e_1) = \frac{w}{2} + \varepsilon, \phi_r(e_1 H^T) = 0\right\}$$

$$\mathcal{L}_2(r,\varepsilon) = \left\{s + e_2 H^T \mid \mathrm{wt}(e_2) = \frac{w}{2} + \varepsilon, \phi_r(s + e_2 H^T) = 0\right\}$$

Becker, Joux, May, and Meurer Algorithm (1/2)

Idea: what happens if we let ε grows (much) beyond $w^2/4n$?

$$\mathcal{L}_1(r,\varepsilon) = \{e_1 H^T \mid \mathrm{wt}(e_1) = \tfrac{w}{2} + \varepsilon, \phi_r(e_1 H^T) = 0\}$$

$$\mathcal{L}_2(r,\varepsilon) = \{s + e_2 H^T \mid \mathrm{wt}(e_2) = \tfrac{w}{2} + \varepsilon, \phi_r(s + e_2 H^T) = 0\}$$

The workfactor becomes $\sqrt{L} + \frac{L}{2^r} + \frac{L^2}{2^{n-k+r}}$ with $L = \binom{n}{w/2+\varepsilon}$ and $2^r = \binom{w}{w/2}\binom{n-w}{\varepsilon}$

Becker, Joux, May, and Meurer Algorithm (1/2)

Idea: what happens if we let ε grows (much) beyond $w^2/4n$?

$$\mathcal{L}_1(r,\varepsilon) = \{e_1 H^T \mid \mathrm{wt}(e_1) = \tfrac{w}{2} + \varepsilon, \phi_r(e_1 H^T) = 0\}$$
$$\mathcal{L}_2(r,\varepsilon) = \{s + e_2 H^T \mid \mathrm{wt}(e_2) = \tfrac{w}{2} + \varepsilon, \phi_r(s + e_2 H^T) = 0\}$$

The workfactor becomes $\sqrt{L} + \dfrac{L}{2^r} + \dfrac{L^2}{2^{n-k+r}}$ with $L = \binom{n}{w/2+\varepsilon}$ and $2^r = \binom{w}{w/2}\binom{n-w}{\varepsilon}$

We may also write $\sqrt{L} + \dfrac{1}{\mu}\dfrac{\binom{n}{w}}{L} + \dfrac{1}{\mu}\dfrac{\binom{n}{w}}{2^{n-k}}$

where $\mu = \dfrac{\binom{w/2+\varepsilon}{\varepsilon}\binom{n-w/2-\varepsilon}{w/2}}{\binom{n}{w/2+\varepsilon}}$ is the probability that two words of weight $w/2+\varepsilon$ and length n have a sum of weight w

BJMM Algorithm (2/2)

BJMM Algorithm, key features:

- increase ε leading to FIBD (Further Improved Birthday Decoding)
- make an additional level of recursive call to FIBD
  (improved birthday decoding makes two calls to smaller CSD problems)
- embed all this into Information Set Decoding framework

BJMM Algorithm (2/2)

BJMM Algorithm, key features:

- increase ε leading to FIBD (Further Improved Birthday Decoding)
- make an additional level of recursive call to FIBD
  (improved birthday decoding makes two calls to smaller CSD problems)
- embed all this into Information Set Decoding framework

Improves the workfactor

Algorithm and analysis are very elaborated

Comparison of the Various ISD Variants

$$\mathrm{WF} = 2^{c \cdot n(1+o(1))}$$

c a constant
(asymptotic exponent)

Comparison of the Various ISD Variants

	$c = \lim_{n\to\infty} \frac{\log_2 \mathrm{WF}}{n}$	
	$k = 0.5n$ $w = 0.11n$	
Enumeration	0.5	
Birthday Decoding	0.25	
Prange	0.1198	
Stern	0.1154	
Dumer	0.1151	
MMT	0.1101	
BJMM	0.1000	

$\mathrm{WF} = 2^{c \cdot n(1+o(1))}$

c a constant
(asymptotic exponent)

Comparison of the Various ISD Variants

	$c = \lim_{n\to\infty} \frac{\log_2 \mathrm{WF}}{n}$	
	$k = 0.5n$ $w = 0.11n$	$k = 0.8n$ $w = 0.03n$
Enumeration	0.5	0.2
Birthday Decoding	0.25	0.1
Prange	0.1198	0.0724
Stern	0.1154	0.0680
Dumer	0.1151	0.0679
MMT	0.1101	0.0638
BJMM	0.1000	0.0562

$\mathrm{WF} = 2^{c \cdot n(1+o(1))}$

c a constant
(asymptotic exponent)

Comparison of the Various ISD Variants

	$c = \lim_{n\to\infty} \frac{\log_2 \mathrm{WF}}{n}$	
	$k = 0.5n$ $w = 0.11n$	$k = 0.8n$ $w = 0.03n$
Enumeration	0.5	0.2
Birthday Decoding	0.25	0.1
Prange	0.1198	0.0724
Stern	0.1154	0.0680
Dumer	0.1151	0.0679
MMT	0.1101	0.0638
BJMM	0.1000	0.0562

$\mathrm{WF} = 2^{c \cdot n(1+o(1))}$

c a constant
(asymptotic exponent)

Remark that Birthday Decoding is comparatively better when k/n grows

3. Message Attack (ISD)

Generalized Birthday Algorithm

Proposed by D. Wagner in 2002, in a more general context

The Generalized Birthday Algorithm (GBA) of order a solves the following problem:

Instance:	2^a lists of vectors $\mathcal{L}_i \subset \{0,1\}^\ell$, $i = 1, 2, \ldots, 2^a$
Answer:	$x_i \in \mathcal{L}_i$, $i = 1, 2, \ldots, 2^a$ such that $x_1 + x_2 + \ldots + x_{2^a} = 0$

If the lists are large enough, then GBA runs in time $O\left(2^{\ell/(a+1)}\right)$

(the case $a = 1$ corresponds to the usual birthday paradox)

Generalized Birthday Algorithm

Proposed by D. Wagner in 2002, in a more general context

The Generalized Birthday Algorithm (GBA) of order a solves the following problem:

Instance:	2^a lists of vectors $\mathcal{L}_i \subset \{0,1\}^\ell$, $i = 1, 2, \ldots, 2^a$
Answer:	$x_i \in \mathcal{L}_i$, $i = 1, 2, \ldots, 2^a$ such that $x_1 + x_2 + \ldots + x_{2^a} = 0$

If the lists are large enough, then GBA runs in time $O\left(2^{\ell/(a+1)}\right)$

(the case $a = 1$ corresponds to the usual birthday paradox)

GBA can be applied to decoding

- it applies to instances of CSD with many solutions
- it aims at finding one solution only

Birthday Decoding Again

Let $H \in \{0,1\}^{(n-k)\times n}$, $s \in \{0,1\}^{n-k}$, and $w > 0$, we consider $\mathrm{CSD}(H, s, w)$ where

- there are many solutions: exact condition to be determined
- we only want one solution

$$H = \left[\begin{array}{c|c} H_1 & H_2 \end{array}\right]$$

$s = s_1 + s_2$ arbitrarily

Birthday Decoding Again

Let $H \in \{0,1\}^{(n-k)\times n}$, $s \in \{0,1\}^{n-k}$, and $w > 0$, we consider CSD(H, s, w) where

- there are many solutions: exact condition to be determined
- we only want one solution

$$H = \begin{array}{|c|c|} \hline \quad H_1 \quad & \quad H_2 \quad \\ \hline \end{array}$$

$s = s_1 + s_2$ arbitrarily

We build two lists of size L

$$\mathcal{L}_i \subset \{s_i + e_i H_i^T \mid \mathrm{wt}(e_i) = w/2\}, i \in \{1,2\}$$

Any element of $\mathcal{L}_1 \cap \mathcal{L}_2$ provides a solution

Birthday Decoding Again

Let $H \in \{0,1\}^{(n-k)\times n}$, $s \in \{0,1\}^{n-k}$, and $w > 0$, we consider CSD(H, s, w) where

- there are many solutions: exact condition to be determined
- we only want one solution

$$H = \begin{array}{|c|c|} \hline H_1 & H_2 \\ \hline \end{array}$$

$s = s_1 + s_2$ arbitrarily

We build two lists of size L

$$\mathcal{L}_i \subset \{s_i + e_i H_i^T \mid \mathrm{wt}(e_i) = w/2\}, i \in \{1,2\}$$

Any element of $\mathcal{L}_1 \cap \mathcal{L}_2$ provides a solution

We must have $|\mathcal{L}_1 \cap \mathcal{L}_2| = \dfrac{L^2}{2^{n-k}} \geq 1$

Birthday Decoding Again

Let $H \in \{0,1\}^{(n-k)\times n}$, $s \in \{0,1\}^{n-k}$, and $w > 0$, we consider CSD(H, s, w) where

- there are many solutions: exact condition to be determined
- we only want one solution

$$H = \begin{array}{|c|c|}\hline H_1 & H_2 \\ \hline\end{array}$$

$s = s_1 + s_2$ arbitrarily

We build two lists of size L

$$\mathcal{L}_i \subset \{s_i + e_i H_i^T \mid \mathrm{wt}(e_i) = w/2\}, i \in \{1,2\}$$

Any element of $\mathcal{L}_1 \cap \mathcal{L}_2$ provides a solution

We must have $|\mathcal{L}_1 \cap \mathcal{L}_2| = \dfrac{L^2}{2^{n-k}} \geq 1$

Choosing $L = 2^{(n-k)/2}$ the workfactor is $\boxed{O\left(2^{(n-k)/2}\right)}$

Birthday Decoding Again

Let $H \in \{0,1\}^{(n-k)\times n}$, $s \in \{0,1\}^{n-k}$, and $w > 0$, we consider $\mathrm{CSD}(H, s, w)$ where

- there are many solutions: $\binom{n/2}{w/2}^2 \geq 2^{n-k}$
- we only want one solution

$$H = \begin{array}{|c|c|} \hline \quad H_1 \quad & \quad H_2 \quad \\ \hline \end{array}$$

$s = s_1 + s_2$ arbitrarily

We build two lists of size L

$$\mathcal{L}_i \subset \{s_i + e_i H_i^T \mid \mathrm{wt}(e_i) = w/2\}, i \in \{1,2\}$$

Any element of $\mathcal{L}_1 \cap \mathcal{L}_2$ provides a solution

We must have $|\mathcal{L}_1 \cap \mathcal{L}_2| = \dfrac{L^2}{2^{n-k}} \geq 1$

Choosing $L = 2^{(n-k)/2}$ the workfactor is $O\left(2^{(n-k)/2}\right)$

L cannot exceed $\binom{n/2}{w/2}$, and thus we need $\binom{n/2}{w/2}^2 \geq 2^{n-k}$

Order 2 GBA for Decoding

$$H = \begin{array}{|c|c|c|c|} \hline \quad H_1 \quad & \quad H_2 \quad & \quad H_3 \quad & \quad H_4 \quad \\ \hline \end{array} \qquad s = s_1 + s_2 + s_3 + s_4$$

Order 2 GBA for Decoding

$$H = \begin{bmatrix} H_1 & H_2 & H_3 & H_4 \end{bmatrix} \qquad s = s_1 + s_2 + s_3 + s_4$$

Let $\mathcal{L}_i \subset \{s_i + e_i H_i^T \mid \mathrm{wt}(e_i) = w/4\}, i \in \{1, 2, 3, 4\}$

Order 2 GBA for Decoding

$$H = \left[\begin{array}{c|c|c|c} H_1 & H_2 & H_3 & H_4 \end{array}\right] \qquad s = s_1 + s_2 + s_3 + s_4$$

Let $\mathcal{L}_i \subset \{s_i + e_i H_i^T \mid \mathrm{wt}(e_i) = w/4\}, i \in \{1, 2, 3, 4\}$ of size $L = 2^\ell$, $\ell = (n-k)/3$

Order 2 GBA for Decoding

$$H = \begin{array}{|c|c|c|c|}\hline H_1 & H_2 & H_3 & H_4 \\ \hline\end{array} \qquad s = s_1 + s_2 + s_3 + s_4$$

Let $\mathcal{L}_i \subset \{s_i + e_i H_i^T \mid \mathrm{wt}(e_i) = w/4\}, i \in \{1,2,3,4\}$ of size $L = 2^\ell$, $\ell = (n-k)/3$

Let $\mathcal{L}_{1,2} \subset \{x_1 + x_2 \mid x_1 \in \mathcal{L}_i, x_2 \in \mathcal{L}_2, \phi_\ell(x_1 + x_2) = 0\}$ ($\phi_\ell(x)$ the last ℓ bits of x)

Order 2 GBA for Decoding

$$H = \begin{array}{|c|c|c|c|} \hline \quad H_1 \quad & \quad H_2 \quad & \quad H_3 \quad & \quad H_4 \quad \\ \hline \end{array} \qquad s = s_1 + s_2 + s_3 + s_4$$

Let $\mathcal{L}_i \subset \{s_i + e_i H_i^T \mid \mathrm{wt}(e_i) = w/4\}, i \in \{1,2,3,4\}$ of size $L = 2^\ell$, $\ell = (n-k)/3$

Let $\mathcal{L}_{1,2} \subset \{x_1 + x_2 \mid x_1 \in \mathcal{L}_i, x_2 \in \mathcal{L}_2, \phi_\ell(x_1 + x_2) = 0\}$ ($\phi_\ell(x)$ the last ℓ bits of x)

We define $\mathcal{L}_{3,4}$ similarly, we expect $|\mathcal{L}_{1,2}| = |\mathcal{L}_{3,4}| = L^2/2^\ell = L$

Order 2 GBA for Decoding

$$H = \begin{array}{|c|c|c|c|} \hline H_1 & H_2 & H_3 & H_4 \\ \hline \end{array} \qquad s = s_1 + s_2 + s_3 + s_4$$

Let $\mathcal{L}_i \subset \{s_i + e_i H_i^T \mid \mathrm{wt}(e_i) = w/4\}, i \in \{1,2,3,4\}$ of size $L = 2^\ell$, $\ell = (n-k)/3$

Let $\mathcal{L}_{1,2} \subset \{x_1 + x_2 \mid x_1 \in \mathcal{L}_i, x_2 \in \mathcal{L}_2, \phi_\ell(x_1 + x_2) = 0\}$ ($\phi_\ell(x)$ the last ℓ bits of x)

We define $\mathcal{L}_{3,4}$ similarly, we expect $|\mathcal{L}_{1,2}| = |\mathcal{L}_{3,4}| = L^2/2^\ell = L$

We expect $|\mathcal{L}_{1,2} \cap \mathcal{L}_{3,4}| = \dfrac{|\mathcal{L}_{1,2}| \cdot |\mathcal{L}_{3,4}|}{2^{n-k-\ell}} = L^4/2^{n-k+\ell} = 1$

Order 2 GBA for Decoding

$$H = \begin{array}{|c|c|c|c|} \hline H_1 & H_2 & H_3 & H_4 \\ \hline \end{array} \qquad s = s_1 + s_2 + s_3 + s_4$$

Let $\mathcal{L}_i \subset \{s_i + e_i H_i^T \mid \mathrm{wt}(e_i) = w/4\}, i \in \{1,2,3,4\}$ of size $L = 2^\ell$, $\ell = (n-k)/3$

Let $\mathcal{L}_{1,2} \subset \{x_1 + x_2 \mid x_1 \in \mathcal{L}_i, x_2 \in \mathcal{L}_2, \phi_\ell(x_1 + x_2) = 0\}$ ($\phi_\ell(x)$ the last ℓ bits of x)

We define $\mathcal{L}_{3,4}$ similarly, we expect $|\mathcal{L}_{1,2}| = |\mathcal{L}_{3,4}| = L^2/2^\ell = L$

We expect $|\mathcal{L}_{1,2} \cap \mathcal{L}_{3,4}| = \dfrac{|\mathcal{L}_{1,2}| \cdot |\mathcal{L}_{3,4}|}{2^{n-k-\ell}} = L^4/2^{n-k+\ell} = 1$

After computing $\mathcal{L}_1, \mathcal{L}_2, \mathcal{L}_3, \mathcal{L}_4, \mathcal{L}_{1,2}, \mathcal{L}_{3,4}$ we expect to find an element in $\mathcal{L}_{1,2} \cap \mathcal{L}_{3,4}$ from which we derive a solution to CSD(H, s, w)

Order 2 GBA for Decoding

$$H = \begin{array}{|c|c|c|c|} \hline H_1 & H_2 & H_3 & H_4 \\ \hline \end{array} \qquad s = s_1 + s_2 + s_3 + s_4$$

Let $\mathcal{L}_i \subset \{s_i + e_i H_i^T \mid \mathrm{wt}(e_i) = w/4\}, i \in \{1,2,3,4\}$ of size $L = 2^\ell$, $\ell = (n-k)/3$

Let $\mathcal{L}_{1,2} \subset \{x_1 + x_2 \mid x_1 \in \mathcal{L}_i, x_2 \in \mathcal{L}_2, \phi_\ell(x_1 + x_2) = 0\}$ ($\phi_\ell(x)$ the last ℓ bits of x)

We define $\mathcal{L}_{3,4}$ similarly, we expect $|\mathcal{L}_{1,2}| = |\mathcal{L}_{3,4}| = L^2/2^\ell = L$

We expect $|\mathcal{L}_{1,2} \cap \mathcal{L}_{3,4}| = \dfrac{|\mathcal{L}_{1,2}| \cdot |\mathcal{L}_{3,4}|}{2^{n-k-\ell}} = L^4/2^{n-k+\ell} = 1$

After computing $\mathcal{L}_1, \mathcal{L}_2, \mathcal{L}_3, \mathcal{L}_4, \mathcal{L}_{1,2}, \mathcal{L}_{3,4}$ we expect to find an element in $\mathcal{L}_{1,2} \cap \mathcal{L}_{3,4}$ from which we derive a solution to $\mathrm{CSD}(H, s, w)$

The computing effort is $O\left(2^{(n-k)/3}\right)$ possible only if $\binom{n/4}{w/4} \geq 2^{(n-k)/3}$

Order a GBA for Decoding

In general the order *a* GBA decoding will have a cost $O\left(2^{\frac{n-k}{a+1}}\right)$

It is possible only if $\binom{n/2^a}{w/2^a} \geq 2^{\frac{n-k}{a+1}}$

Order a GBA for Decoding

In general the order *a* GBA decoding will have a cost $O\left(2^{\frac{n-k}{a+1}}\right)$

It is possible only if $\binom{n/2^a}{w/2^a} \geq 2^{\frac{n-k}{a+1}}$

Asymptotically, the condition becomes $\binom{n}{w} \geq 2^{\frac{2^a}{a+1}(n-k)}$ up to a polynomial factor

This reflects the fact that higher order GBA requires higher values of *w*

Order a GBA for Decoding

In general the order a GBA decoding will have a cost $\boxed{O\left(2^{\frac{n-k}{a+1}}\right)}$

It is possible only if $\binom{n/2^a}{w/2^a} \geq 2^{\frac{n-k}{a+1}}$

Asymptotically, the condition becomes $\boxed{\binom{n}{w} \geq 2^{\frac{2^a}{a+1}(n-k)}}$ up to a polynomial factor

This reflects the fact that higher order GBA requires higher values of w

Finally, note that improvements of birthday decoding apply

This allows to lower the complexity in some cases

Comparing GBA and ISD

Information Set Decoding (all variants) and its complexity analysis can easily be adapted to the case where we seek one solution among many

In practice ISD is almost always more efficient

GBA is more efficient only when the code rate k/n is close to 1 and even then, it is only better for a limited range of values of w

3. Message Attack (ISD)

Decoding One Out of Many (DOOM)

N-Syndrome Decoding

Instance: $S \subset \{0,1\}^{n-k}$, $|S| = N$, $H \in \{0,1\}^{(n-k)\times n}$, an integer $w > 0$
Answer: $e \in \{0,1\}^n$ such that $eH^T \in S$ and $\mathrm{wt}(e) \leq w$

We will denote $\mathrm{CSD}_N(H, S, w)$ the set of all solutions to the above problem

Decoding One Out of Many (DOOM)

N-Syndrome Decoding

Instance: $S \subset \{0,1\}^{n-k}$, $|S| = N$, $H \in \{0,1\}^{(n-k)\times n}$, an integer $w > 0$

Answer: $e \in \{0,1\}^n$ such that $eH^T \in S$ and $\mathrm{wt}(e) \leq w$

We will denote $\mathrm{CSD}_N(H, S, w)$ the set of all solutions to the above problem

As for CSD_1, we will consider solvable instances

Decoding One Out of Many (DOOM)

N-Syndrome Decoding

Instance: $S \subset \{0,1\}^{n-k}$, $|S| = N$, $H \in \{0,1\}^{(n-k)\times n}$, an integer $w > 0$
Answer: $e \in \{0,1\}^n$ such that $eH^T \in S$ and $\mathrm{wt}(e) \leq w$

We will denote $\mathrm{CSD}_N(H, S, w)$ the set of all solutions to the above problem

As for CSD_1, we will consider solvable instances

Meaning that $S \subset \{eH^T \mid \mathrm{wt}(e) = w\}$

Decoding One Out of Many (DOOM)

N-Syndrome Decoding

Instance: $S \subset \{0,1\}^{n-k}$, $|S| = N$, $H \in \{0,1\}^{(n-k)\times n}$, an integer $w > 0$
Answer: $e \in \{0,1\}^n$ such that $eH^T \in S$ and $\mathrm{wt}(e) \leq w$

We will denote $\mathrm{CSD}_N(H, S, w)$ the set of all solutions to the above problem

As for CSD_1, we will consider solvable instances

Meaning that $S \subset \{eH^T \mid \mathrm{wt}(e) = w\}$

Improvement:

- we get the N solutions at the expense of a factor $\approx \sqrt{N}$
- or we get one solution with a gain of a factor $\approx \sqrt{N}$

Birthday Decoding With Multiple Instances

Solve $\mathrm{CSD}_N(H, S, w)$ with birthday decoding

$$\text{Let} \begin{cases} \mathcal{L}_1 = \{e_1 H_1^T \mid \mathrm{wt}(e_1) = w_1\} \\ \mathcal{L}_2 = \{s + e_2 H_2^T \mid s \in S, \mathrm{wt}(e_2) = w_2\} \end{cases}$$

$$H = \begin{array}{cc} n_1 & n_2 \\ \left[\begin{array}{c|c} H_1 & H_2 \end{array}\right] & \end{array}$$

$$n = n_1 + n_2,\ w = w_1 + w_2$$

Birthday Decoding With Multiple Instances

Solve $\mathrm{CSD}_N(H, S, w)$ with birthday decoding

$$\begin{array}{cc} & \begin{array}{cc} \overset{n_1}{\longleftrightarrow} & \overset{n_2}{\longleftrightarrow} \end{array} \\ H = & \left[\begin{array}{c|c} \quad H_1 \quad & \quad H_2 \quad \end{array}\right] \end{array}$$

$$n = n_1 + n_2, \; w = w_1 + w_2$$

Let $\begin{cases} \mathcal{L}_1 = \{e_1 H_1^T \mid \mathrm{wt}(e_1) = w_1\} \\ \mathcal{L}_2 = \{s + e_2 H_2^T \mid s \in S, \mathrm{wt}(e_2) = w_2\} \end{cases}$

We choose w_1 and w_2 such that

$$\frac{w_1}{n_1} = \frac{w_2}{n_2} \text{ and } |\mathcal{L}_1| = \binom{n_1}{w_1} = |\mathcal{L}_2| = N\binom{n_2}{w_2}$$

Birthday Decoding With Multiple Instances

Solve $\mathrm{CSD}_N(H, S, w)$ with birthday decoding

$$H = \begin{array}{cc} \overset{n_1}{\longleftrightarrow} & \overset{n_2}{\longleftrightarrow} \\ \boxed{\quad H_1 \quad} & \boxed{\quad H_2 \quad} \end{array}$$

$$n = n_1 + n_2,\ w = w_1 + w_2$$

Let $\begin{cases} \mathcal{L}_1 = \{e_1 H_1^T \mid \mathrm{wt}(e_1) = w_1\} \\ \mathcal{L}_2 = \{s + e_2 H_2^T \mid s \in S, \mathrm{wt}(e_2) = w_2\} \end{cases}$

We choose w_1 and w_2 such that

$$\frac{w_1}{n_1} = \frac{w_2}{n_2} \text{ and } |\mathcal{L}_1| = \binom{n_1}{w_1} = |\mathcal{L}_2| = N\binom{n_2}{w_2}$$

Claim: If $N \leq \binom{n}{w}$, we obtain all solutions of $\mathrm{CSD}_N(H, S, w)$

for a cost $\sqrt{N\binom{n}{w}} + \frac{N\binom{n}{w}}{2^{n-k}}$ (up to a polynomial factor)

DOOM-ISD

Solve $\mathrm{CSD}_N(H, S, w)$ when $S \subset \{eH^T \mid \mathrm{wt}(e) = w\}$ with Dumer Algorithm

The problem has N solutions and we only want one

$$
\begin{array}{cc|c}
\left[\begin{array}{c|c}
\begin{matrix}1 & & \\ & \ddots & \\ & & 1\end{matrix} & H'' \\
\hline
0 & H'
\end{array}\right] & & \updownarrow \ell \\
\begin{matrix} \quad w-p \quad & \quad\quad p \quad\quad \end{matrix} & &
\end{array}
$$

DOOM-ISD

Solve $\mathrm{CSD}_N(H, S, w)$ when $S \subset \{eH^T \mid \mathrm{wt}(e) = w\}$ with Dumer Algorithm

The problem has N solutions and we only want one

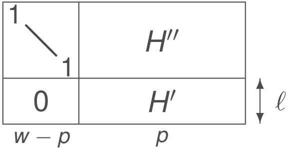
A specific solution requires $\mathcal{N}_\infty = \dfrac{\binom{n}{w}}{\binom{n-k-\ell}{w-p}\binom{k+\ell}{p}}$ iterations

$$\begin{array}{ccl} \left[\begin{array}{c|c} \begin{matrix}1 & & \\ & \ddots & \\ & & 1\end{matrix} & H'' \\ \hline 0 & H' \end{array}\right] & \begin{array}{l} \\ \\ \updownarrow \ell \end{array} \\ \begin{array}{cc} w-p \quad & \quad\quad p \end{array} & \end{array}$$

DOOM-ISD

Solve $\mathrm{CSD}_N(H, S, w)$ when $S \subset \{eH^T \mid \mathrm{wt}(e) = w\}$ with Dumer Algorithm

The problem has N solutions and we only want one



A specific solution requires $\mathcal{N}_\infty = \dfrac{\binom{n}{w}}{\binom{n-k-\ell}{w-p}\binom{k+\ell}{p}}$ iterations

For one solution only, we expect $\mathcal{N}_1 = \mathcal{N}_\infty / N$ iterations as long as $N \leq \mathcal{N}_\infty$

DOOM-ISD

Solve $\mathrm{CSD}_N(H, S, w)$ when $S \subset \{eH^T \mid \mathrm{wt}(e) = w\}$ with Dumer Algorithm

The problem has N solutions and we only want one

A specific solution requires $\mathcal{N}_\infty = \dfrac{\binom{n}{w}}{\binom{n-k-\ell}{w-p}\binom{k+\ell}{p}}$ iterations

$$\begin{array}{cc|c}
\left[\begin{array}{c|c} \begin{matrix} 1 & & \\ & \ddots & \\ & & 1 \end{matrix} & H'' \\ \hline 0 & H' \end{array}\right] & \updownarrow \ell \\
\quad w-p \qquad\qquad p &
\end{array}$$

For one solution only, we expect $\mathcal{N}_1 = \mathcal{N}_\infty / N$ iterations as long as $N \leq \mathcal{N}_\infty$

Iteration cost: $\mathcal{K} = \sqrt{N\binom{k+\ell}{p}} + \dfrac{N\binom{k+\ell}{p}}{2^\ell}$ as long as $N \leq \binom{k+\ell}{p}$

DOOM-ISD

Solve $\mathrm{CSD}_N(H, S, w)$ when $S \subset \{eH^T \mid \mathrm{wt}(e) = w\}$ with Dumer Algorithm

The problem has N solutions and we only want one

A specific solution requires $\mathcal{N}_\infty = \dfrac{\binom{n}{w}}{\binom{n-k-\ell}{w-p}\binom{k+\ell}{p}}$ iterations

$$\begin{array}{cc c}
\begin{array}{|c|c|}\hline \begin{matrix}1 & & \\ & \ddots & \\ & & 1\end{matrix} & H'' \\ \hline 0 & H' \\ \hline \end{array} & \updownarrow \ell \\
\quad w-p \qquad\qquad p &
\end{array}$$

For one solution only, we expect $\mathcal{N}_1 = \mathcal{N}_\infty / N$ iterations as long as $N \le \mathcal{N}_\infty$

Iteration cost: $\mathcal{K} = \sqrt{N\binom{k+\ell}{p}} + \dfrac{N\binom{k+\ell}{p}}{2^\ell}$ as long as $N \le \binom{k+\ell}{p}$

$$\rightarrow \quad \mathrm{WF}_{\mathrm{DOOM}} = \min_{0 \le p \le w} \frac{\binom{n}{w}}{\binom{n-k-\ell}{w-p}\sqrt{N\binom{k+\ell}{p}}} \text{ with } \ell = \log_2 \sqrt{N\binom{k+\ell}{p}}$$

DOOM-ISD

Solve $\mathrm{CSD}_N(H, S, w)$ when $S \subset \{eH^T \mid \mathrm{wt}(e) = w\}$ with Dumer Algorithm

The problem has N solutions and we only want one

$$\begin{array}{cc|c}
\left[\begin{array}{c|c} \begin{matrix}1 & & \\ & \ddots & \\ & & 1\end{matrix} & H'' \\ \hline 0 & H' \end{array}\right] & \updownarrow \ell \\
\begin{matrix} \quad w-p \quad & \quad\quad p \quad\end{matrix} &
\end{array}$$

A specific solution requires $\mathcal{N}_\infty = \dfrac{\binom{n}{w}}{\binom{n-k-\ell}{w-p}\binom{k+\ell}{p}}$ iterations

For one solution only, we expect $\mathcal{N}_1 = \mathcal{N}_\infty / N$ iterations as long as $N \leq \mathcal{N}_\infty$

Iteration cost: $\mathcal{K} = \sqrt{N\binom{k+\ell}{p}} + \dfrac{N\binom{k+\ell}{p}}{2^\ell}$ as long as $N \leq \binom{k+\ell}{p}$

$$\rightarrow \mathrm{WF}_{\mathrm{DOOM}} = \min_{0 \leq p \leq w} \frac{\binom{n}{w}}{\binom{n-k-\ell}{w-p}\sqrt{N\binom{k+\ell}{p}}} \text{ with } \ell = \log_2 \sqrt{N\binom{k+\ell}{p}}$$

$\rightarrow$ gain of a factor $\approx \sqrt{N}$ as long as $N \leq \min\left(\mathcal{N}_\infty, \binom{k+\ell}{p}\right)$

DOOM-GBA

A gain is also possible with an Order 2 GBA Decoding when $N = |S| = \binom{n/3}{w/3}$

$$H = \begin{array}{|c|c|c|} \multicolumn{1}{c}{n/3} & \multicolumn{1}{c}{n/3} & \multicolumn{1}{c}{n/3} \\ \hline \quad H_1 \quad & \quad H_2 \quad & \quad H_3 \quad \\ \hline \end{array}$$

DOOM-GBA

A gain is also possible with an Order 2 GBA Decoding when $N = |S| = \binom{n/3}{w/3}$

$$H = \begin{array}{|c|c|c|} \multicolumn{1}{c}{\overset{n/3}{\longleftrightarrow}} & \multicolumn{1}{c}{\overset{n/3}{\longleftrightarrow}} & \multicolumn{1}{c}{\overset{n/3}{\longleftrightarrow}} \\ \hline H_1 & H_2 & H_3 \\ \hline \end{array}$$

$\mathcal{L}_i \subset \{e_i H_i^T \mid \mathrm{wt}(e_i) = w/3\}, i \in \{1, 2, 3\}$ and $\mathcal{L}_4 = S$

DOOM-GBA

A gain is also possible with an Order 2 GBA Decoding when $N = |S| = \binom{n/3}{w/3}$

$$H = \begin{array}{|c|c|c|} \multicolumn{1}{c}{\overset{n/3}{\longleftrightarrow}} & \multicolumn{1}{c}{\overset{n/3}{\longleftrightarrow}} & \multicolumn{1}{c}{\overset{n/3}{\longleftrightarrow}} \\ \hline \quad H_1 \quad & \quad H_2 \quad & \quad H_3 \quad \\ \hline \end{array}$$

$\mathcal{L}_i \subset \{e_i H_i^T \mid \text{wt}(e_i) = w/3\}, i \in \{1,2,3\}$ and $\mathcal{L}_4 = S$

From $x_i \in \mathcal{L}_i, i \in \{1,2,3,4\}$ such that $x_1 + x_2 + x_3 + x_4 = 0$ we obtain

$$e_1 H_1^T + e_2 H_2^T + e_3 H_3^T + s = 0, s \in S$$

and we have $e = (e_1, e_2, e_3) \in \text{CSD}_N(H, S, w)$

DOOM-GBA

A gain is also possible with an Order 2 GBA Decoding when $N = |S| = \binom{n/3}{w/3}$

$$H = \begin{array}{|c|c|c|} \multicolumn{1}{c}{n/3} & \multicolumn{1}{c}{n/3} & \multicolumn{1}{c}{n/3} \\ \hline \quad H_1 \quad & \quad H_2 \quad & \quad H_3 \quad \\ \hline \end{array}$$

$\mathcal{L}_i \subset \{e_i H_i^T \mid \mathrm{wt}(e_i) = w/3\}, i \in \{1,2,3\}$ and $\mathcal{L}_4 = S$

From $x_i \in \mathcal{L}_i, i \in \{1,2,3,4\}$ such that $x_1 + x_2 + x_3 + x_4 = 0$ we obtain

$$e_1 H_1^T + e_2 H_2^T + e_3 H_3^T + s = 0, s \in S$$

and we have $e = (e_1, e_2, e_3) \in \mathrm{CSD}_N(H, S, w)$

Workfactor is $\boxed{\binom{n/3}{w/3} \approx \sqrt[3]{\binom{n}{w}}}$ up to a polynomial factor

DOOM-GBA

A gain is also possible with an Order 2 GBA Decoding when $N = |S| = \binom{n/3}{w/3}$

$$H = \begin{array}{|c|c|c|} \multicolumn{1}{c}{\overset{n/3}{\longleftrightarrow}} & \multicolumn{1}{c}{\overset{n/3}{\longleftrightarrow}} & \multicolumn{1}{c}{\overset{n/3}{\longleftrightarrow}} \\ \hline \quad H_1 \quad & \quad H_2 \quad & \quad H_3 \quad \\ \hline \end{array}$$

$\mathcal{L}_i \subset \{e_i H_i^T \mid \mathrm{wt}(e_i) = w/3\}, i \in \{1, 2, 3\}$ and $\mathcal{L}_4 = S$

From $x_i \in \mathcal{L}_i, i \in \{1, 2, 3, 4\}$ such that $x_1 + x_2 + x_3 + x_4 = 0$ we obtain

$$e_1 H_1^T + e_2 H_2^T + e_3 H_3^T + s = 0, s \in S$$

and we have $e = (e_1, e_2, e_3) \in \mathrm{CSD}_N(H, S, w)$

Workfactor is $\binom{n/3}{w/3} \approx \sqrt[3]{\binom{n}{w}}$ up to a polynomial factor

To be compared with $\sqrt{\binom{n}{w}}$ with the birthday decoding, gaining a factor $\approx \sqrt{N}$

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

Code-Based Cryptography