QSI Spring School on PQ Crypto, Porto

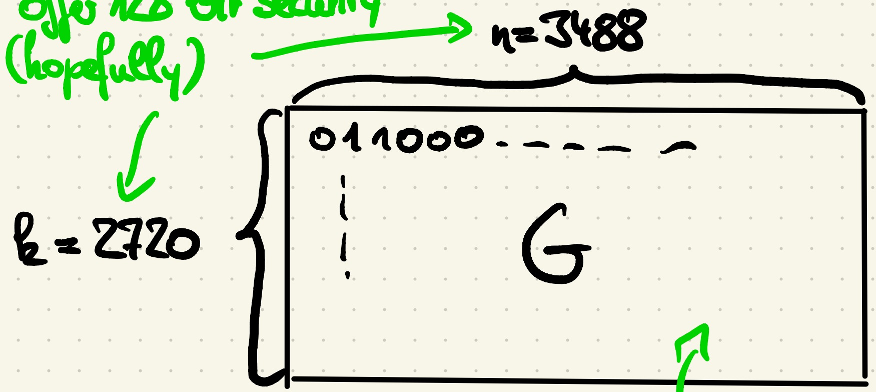
Lecture: Codes (Part I: Constructions)

Alexander May, Ruhr-University Bochum

Teaser for McEliece (Robert McEliece '78, Standard Classic McEliece '24(?))

public key pk

offer 128 bit security (hopefully)



Size of G:

2720·3488 bit

≈ 9MB

(compare with RSA-4096)

generator matrix over $\mathbb{F}_2$

128-bit security: Best algorithm requires $\geq 2^{128}$ steps classically, quantumly less.

Encryption: $c = m \cdot G + e$

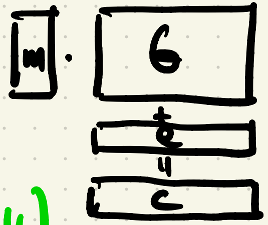
ciphertext

pk

message

added error, small no. of ones (64)

Notion:



Linear Code

We omit binary, since we always work with the binary field $\mathbb{F}_2$.

Def: A (binary) linear code C is a subspace of $\mathbb{F}_2^n$.
Let $k = \dim(C)$. Any basis $G \in \mathbb{F}_2^{k \times n}$ is called a generator matrix.

Notice that $C = \{ xG \mid x \in \mathbb{F}_2^k \}$ and therefore $|C| = 2^k$.

Great for crypto: We compactly represent 2^k codewords from C with only $k \cdot n$ bits.

Example: Repetitioncode $R(3)$

$$G = \begin{pmatrix} 111 & & & 0 \\ & 111 & & \\ 0 & & \ddots & \\ & & & 111 \end{pmatrix} \in \mathbb{F}_2^{k \times 3k} \qquad (x_1 x_2 \dots x_k) \cdot G = x_1 x_1 x_1 x_2 x_2 x_2 \dots x_k x_k x_k$$

Hamming Distance

Def: Let $x \in \mathbb{F}_2^n$. We define the **support** of x as

$$\mathrm{supp}(x) := \{ i \in \mathbb{N} \mid x_i \neq 0 \}.$$

The **hamming weight** of x is defined as

$$w(x) := |\mathrm{supp}(x)|.$$

The **distance** of $x, y \in \mathbb{F}_2^n$ is defined as

$$d(x,y) := w(x+y).$$

the set of 1-positions

$\mathrm{supp}(0110) = \{2,3\}$

no. of 1-positions

$\mathrm{wt}(0110) = 2$

no. of different positions

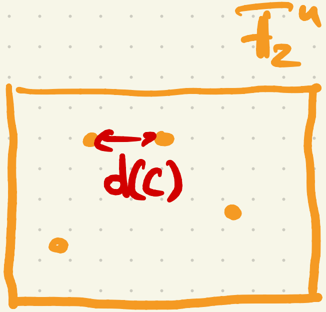
$d(0110, 1000) = 3$

Distance of code

Def: Let $C \subseteq \mathbb{F}_2^n$. Then the distance of C is defined as

$$d(C) = \min_{c \neq c' \in C} \{d(c,c')\}$$

Determine $d(C)$ in $\Theta(|C|^2)$.



We call $\frac{d(C)}{n}$ relative distance and $\frac{k}{n}$ rate of C.

Theorem: Let $C \subseteq \mathbb{F}_2^n$ be a linear code. Then

$$d(C) = \min_{c \in C \setminus \{0\}} \{w(c)\}. \leftarrow \text{Determine in } \Theta(|C|).$$

$\{0\} \leftarrow$ zero vector

Proof: „$\geq$" Let $c \neq c' \in C$ with minimal distance.

$$\rightarrow d(C) = d(c,c') = d(c+c', c'+c')$$
$$= d(c+c', 0) = w(c+c')$$
$$\geq \min_{c \in C \setminus \{0\}} \{c\}$$

$$\min_{c \in C \setminus \{0\}} \{c\} = \min_{c \neq 0 \in C} \{d(c,0)\}$$
$$\leq \min_{c \neq c' \in C} \{d(c,c')\}$$
$$= d(C)$$

Hamming ball

Notation: A linear code $C \leq \mathbb{F}_2^n$ with $\dim(C)=k$ and $d(C)$ is denoted $[n,k,d]$-code. $\mathcal{R}(3)$ is an $[3k,k,3]$-code.

Def: Let $x \in \mathbb{F}_2^n$ and $r \in \mathbb{N}$. The Hamming ball with center x and radius r is

$$\mathcal{B}^n(x,r) = \{ y \in \mathbb{F}_2^n \mid d(x,y) \leq r \}.$$

We define the volume of $\mathcal{B}^n(x,r)$ as $V^n(r) := |\mathcal{B}^n(x,r)|$

$\mathcal{B}^4(0110,1) = \{0110, 1110, 0010, 0100, 0111\}$, $V^4(1) = 1+4 = 5$

Theorem: $V^n(r) = \sum_{i=0}^{r} \binom{n}{i}$

Proof: There are $\binom{n}{i}$ vectors in distance i of some center.

Entropy and Binomials

Suppresses constants

suppress low-order (polynomial) terms

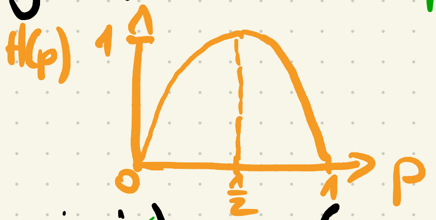
Notation: $3n^2+2n+1 = \Theta(n^2)$, $2n^3\cdot 2^n = \tilde{\Theta}(2^n)$

Fact (Stirling): $n! = \tilde{\Theta}\left(\left(\frac{n}{e}\right)^n\right)$

Note than $n^n = 2^{n\log n}$. Growth of $n!$ is faster than exponential in n.

Def: Let $p<1$. Then

$$H(p) := -p\cdot\log p - (1-p)\cdot\log(1-p).$$ ← Binary entropy.



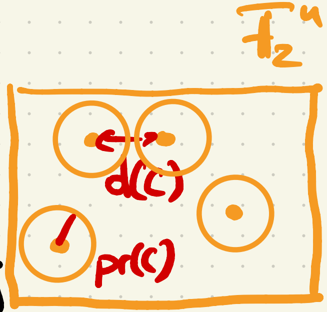
Theorem: $\binom{n}{i} = \tilde{\Theta}\left(2^{H(\frac{i}{n})\cdot n}\right)$

Proof:

$$\binom{n}{i} = \frac{n!}{(n-i)!\;i!} \overset{\text{Stirling}}{=} \tilde{\Theta}\left(\frac{n^n}{(n-i)^{n-i}\cdot i^i}\cdot\frac{\cancel{e^{n-i}}\cdot\cancel{e^i}}{\cancel{e^n}}\right) = \tilde{\Theta}\left(\frac{n^n}{\left(n(1-\frac{i}{n})\right)^{n-i}\cdot\left(n\cdot\frac{i}{n}\right)^i}\right)$$

$$= \tilde{\Theta}\left(\left(\frac{1}{(1-\frac{i}{n})^{1-\frac{i}{n}}\cdot(\frac{i}{n})^{\frac{i}{n}}}\right)^n\right) = \tilde{\Theta}\left(2^{H(\frac{i}{n})\cdot n}\right)$$

Packing radius & unique decoding



Def: Let $C \subseteq \mathbb{F}_2^n$ be a code. C's packing radius is

$$pr(C) := \max_{r \in \mathbb{N}} \{ B^n(c,r) \text{ are disjoint for all } c \in C \}$$

Corollary: $pr(C) = \lfloor \frac{d-1}{2} \rfloor$.

Promise problem: In crypto applications we are often in $B^n(c, \lfloor \frac{d-1}{2} \rfloor)$ by construction.

Note: Every point in $B^n(c, pr(C))$ allows for unique decoding to c.

$d(R(3)) = 3 \Rightarrow pr(R(3)) = 1$

R(3) allows for correcting one error (per block) via majority decision.

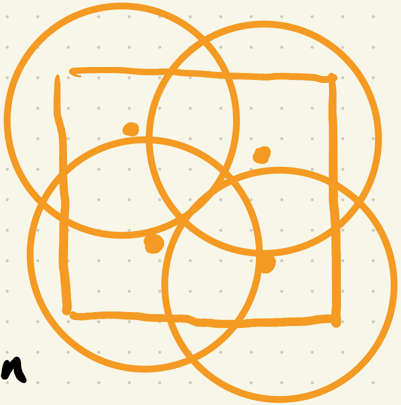
$010 \rightsquigarrow 000$, $011 \rightsquigarrow 111$

GV bound (Gilbert-Varshamov)

Theorem: There exists an $[n,k]$-code with distance d satisfying

$$H\Big(\underset{\text{relative distance}}{\frac{d}{n}}\Big) \geq 1-\underset{\text{rate}}{\frac{k}{n}}.$$

Proof sketch: Let $C \subseteq \mathbb{F}_2^n$ have maximal $k=\dim(C)$ among all linear codes with distance d. Then $\bigcup_{c\in C} B^n(d-1,c)=\mathbb{F}_2^n$.



(Otherwise we can add $v\in \mathbb{F}_2^n \setminus \bigcup_{c\in C} B^n(d-1,c)$ to the basis)

$$\Rightarrow 2^n = \Big|\bigcup_{c\in C} B^n(d-1,c)\Big| \leq \sum_{c\in C} |B^n(d-1,c)| = |C|\cdot V^n(d-1)$$

$$\approx 2^k \cdot 2^{H(\frac{d}{n})\cdot n}$$

$$\Rightarrow H\Big(\frac{d}{n}\Big) \geq 1-\frac{k}{n}.$$

Fact: Codes with random $G\in\mathbb{F}_2^{k\times n}$ achieve max. distance $H(\frac{d}{n})\approx 1-\frac{k}{n}$.

Inner Product

Def: Let $x, y \in \mathbb{F}_2^n$ with inner product

$$\mathbb{F}_2^n \times \mathbb{F}_2^n \longmapsto \mathbb{F}_2 , \quad (x,y) \mapsto \langle x,y \rangle = \sum_{i=1}^{n} x_i y_i$$

Facts: ① Symmetry: $\langle x,y \rangle = \langle y,x \rangle$

② Bilinearity $\langle x+y, z \rangle = \langle x,z \rangle + \langle y,z \rangle$

③ Scalar Associativity: $\langle \alpha x, y \rangle = \alpha \langle x,y \rangle = \langle x, \alpha y \rangle$ for $\alpha \in \mathbb{F}_2$

Def: We call x,y orthogonal if $\langle x,y \rangle = 0$.

Exercise: Show that every $x \in \mathbb{F}_2^n \setminus 0$ is orthogonal to half of the vectors in $\mathbb{F}_2^n$.

$\uparrow$ Lot of orthogonality in $\mathbb{F}_2^n$.

Orthogonal Complement

Def: Let $C \subseteq \mathbb{F}_2^n$ be a linear code. We denote the orthogonal complement of C by

$$C^\perp = \{ x \in \mathbb{F}_2^n \mid \langle c, x \rangle = 0 \text{ for all } c \in C \}.$$

Example: Let C be generated by $G = \begin{pmatrix} 1011 \\ 1001 \end{pmatrix}$. Elements $x \in C^\perp$ satisfy

$$\left| \begin{array}{l} \langle 1011, x \rangle = 0 \\ \langle 1001, x \rangle = 0 \end{array} \right| \Leftrightarrow \left| \begin{array}{l} x_1 + x_3 + x_4 = 0 \\ x_1 + x_4 = 0 \end{array} \right| \Leftrightarrow \left| \begin{array}{l} x_3 = 0 \\ x_1 + x_4 = 0 \end{array} \right|$$

Why does the basis suffice?

$$\Rightarrow C^\perp = \{0000, 1001, 0100, 1101\}$$

generated by $\begin{pmatrix} 1001 \\ 0100 \end{pmatrix}$ ← Always a linear code?

Dual Code

Theorem: $C^{\perp} \subseteq \mathbb{F}_2^n$ is a linear code, called the dual code of C.

Proof: ① $0^n \in \mathbb{F}_2^n$

← We show that $C^{\perp}$ is a subspace.

② Let $x, y \in C^{\perp}$. Then we have for all $c \in C$

$$\langle x+y, c\rangle = \underbrace{\langle x, c\rangle}_{0} + \underbrace{\langle y, c\rangle}_{0} = 0 \Rightarrow x+y \in C^{\perp}.$$

Theorem: Let C, D be linear codes with $C \subseteq D$. Then $D^{\perp} \subseteq C^{\perp}$.

Proof:

$$x \in D^{\perp} \Rightarrow \langle x, d\rangle = 0 \text{ for all } d \in D$$

$$\Rightarrow \langle x, c\rangle = 0 \text{ for all } c \in C$$

$$\Rightarrow x \in C^{\perp}$$

Parity Check Matrix

Def: Let $C \subseteq \mathbb{F}_2^n$ be a linear code. Any matrix P is called parity check matrix of C if $C = \{x \in \mathbb{F}_2^n \mid P \cdot x^t = 0\}$. ↖ May define C via G or P.

Let $f: \mathbb{F}_2^k \to \mathbb{F}_2^n$, $x \mapsto Px^t$.
Then $\ker(f) = C$.

Theorem: Let C be generated by $G \in \mathbb{F}_2^{k \times n}$.

① $C^\perp = \{x \in \mathbb{F}_2^n \mid Gx^t = 0\}$, i.e., G is parity check matrix of $C^\perp$.

② $\dim(C^\perp) = n - \dim(C) = n - k$

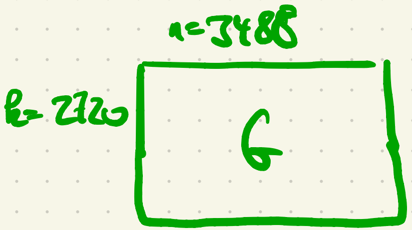
③ $C^{\perp\perp} = C$

Proof: left as an exercise

Some McEliece implications

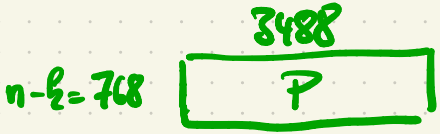
Recall: McEliece's pk is generator matrix G.

Problem: pretty large.



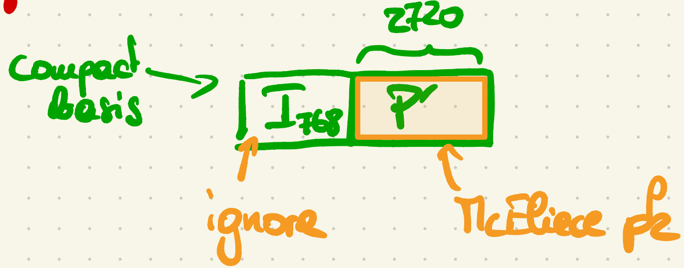
Ideas: ① Take parity check matrix P.

Saves already factor of $\frac{2720}{768} \approx 4$ in size.



② Take systematic form of P.

Saves another 768^2 bits.



Exercise: Show that a random matrix from $\mathbb{F}_2^{n \times n}$ is invertible with probability $\prod_{i=0}^{n-1} 1-\frac{1}{2^{n-i}} > 0.288$.

Equivalent codes

Def: Let C have basis $G \in \mathbb{F}_2^{k \times n}$. C' is **equivalent** to C if there exists an invertible matrix $S \in \mathbb{F}_2^{k \times k}$ and a permutation matrix $P \in \mathbb{F}_2^{n \times n}$ such that

$$C' \text{ is generated by } \overset{\text{another basis of } G}{\underset{\text{performs row operations, e.g. Gauss elimination}}{\boxed{S \cdot G}}} \cdot P. \leftarrow \text{permutates columns, i.e., codeword positions}$$

Notice: Equivalent codes have the same parameters $[n,k,d]$.

Exercise: Show that any C has an equivalent code with generator/parity check matrix in systematic form.

From generator to parity (and vice versa)

Theorem: Let C be generated by $G = [I_k \mid \overset{\mathbb{F}_2^{k\times(n-k)}}{A}] \in \mathbb{F}_2^{k\times n}$.
Then $P = [\underset{\mathbb{F}_2^{(n-k)\times k}}{A^t} \mid I_{n-k}] \in \mathbb{F}_2^{(n-k)\times n}$ is a parity check matrix for C.

Proof: Let C' be the code with parity check matrix P. We show $C' = C$ via

① $C' \leq C$: For every row g_i of G we have $P g_i^t = 0$, since its j-th entry is

$$(a_{1j} \dots a_{kj}\, 0 \dots \underset{\uparrow j}{1} \dots 0) \cdot (0 \dots \underset{\uparrow i}{1} \dots 0\, a_{i1} \dots a_{in-k})$$

$$= a_{ij} + a_{ij} = 0.$$

② $\dim(C') = \dim(C)$: $(C')^\perp$ has generator $P \in \mathbb{F}_2^{(n-k)\times n}$

$$\Rightarrow \dim(C') = n - \underbrace{\dim(C'^\perp)}_{n-k} = k = \dim(C).$$

Goppa Code

McEliece parameters : $n = 3488$, $t = 64$, $m = 12$ $tm = 12 \cdot 64 = 768 = n - k$

① m defines the large field $\mathbb{F}_{2^m} = \mathbb{F}_{2^{12}}$ with 4096 elements.

② t defines the degree of the irreducible Goppa polynomial $g(x) \in \mathbb{F}_{2^m}[x]$, i.e.,

$$g(x) = \sum_{i=0}^{t} g_i \cdot x^i , \quad g_i \in \mathbb{F}_{2^m} .$$

implies $n \leq 2^m$

③ n defines the number of distinct Goppa points $L = \{\alpha_1, \dots, \alpha_n\} \leq \mathbb{F}_{2^m}$.

Definition: A Goppa code C of length n is

elegant, but not suited for defining a parity check matrix

$$C(L,g) = \left\{ c \in \mathbb{F}_2^n : \sum_{i=1}^{n} \frac{c_i}{x - \alpha_i} = 0 \mod g(x) \right\}$$

Exercise: Check that $C(L,g)$ is a code, i.e., a subspace of $\mathbb{F}_2^n$.

Towards a Parity Check Matrix

cancel

Recall: $g(x) = g_0 + g_1 x + \dots + g_t x^t$

Observe that $\frac{1}{x-\alpha_i} = -\frac{g(x) - \overset{=0}{g(\alpha_i)}}{x-\alpha_i} \cdot g^{-1}(\alpha_i) \mod g(x)$. Let $g(x) = \sum_{i=0}^{t} g_i x^i$.

$$\frac{g(x)-g(\alpha_i)}{x-\alpha_i} = \frac{g_1(x-\alpha_i) + \dots + g_t(x^t-\alpha_i^t)}{x-\alpha_i} = g_1 + g_2(x+\alpha_i) + g_3(x^2+\alpha_i x+\alpha_i^2) +$$

$$\dots + g_t(x^{t-1} + \alpha_i x^{t-2} + \dots + \alpha_i^{t-1})$$

Codeword $c = \sum_{i=1}^{n} c_i \frac{g(x)-g(\alpha_i)}{x-\alpha_i} g^{-1}(\alpha_i)$ has coefficients

Identity: $P' c = 0^t$.

$x^{t-1}: \sum_{i=1}^{n} c_i g_t g^{-1}(\alpha_i)$

$x^{t-2}: \sum_{i=1}^{n} c_i (g_{t-1} + \alpha_i g_t) g^{-1}(\alpha_i)$

$\vdots$

$x^0: \sum_{i=1}^{n} c_i (g_1 + \alpha_i g_2 + \alpha_i^2 g_3 + \dots + \alpha_i^{t-1} g_t) \cdot g^{-1}(\alpha_i)$

$$P' = \begin{pmatrix} g_t & g_t & & g_t \\ g_{t-1}+\alpha_1 g_t & g_{t-1}+\alpha_2 g_t & \cdots & g_{t-1}+\alpha_n g_t \\ \vdots & \vdots & \ddots & \vdots \\ g_1+\dots+\alpha_1^{t-1} g_t & g_1+\dots+\alpha_2^{t-1} g_t & \cdots & g_1+\alpha_n^{t-1} g_t \end{pmatrix} \cdot \begin{pmatrix} g^{-1}(\alpha_1) & & 0 \\ & \ddots & \\ 0 & & g^{-1}(\alpha_n) \end{pmatrix}$$

$$P' = \begin{pmatrix} g_t & g_t & & g_t \\ g_{t-1}+\alpha_1 g_t & g_{t-1}+\alpha_2 g_t & \cdots & g_{t-1}+\alpha_n g_t \\ \vdots & \vdots & \ddots & \vdots \\ g_1+\dots+\alpha_1^{t-1} g_t & g_1+\dots+\alpha_2^{t-1} g_t & \cdots & g_1+\alpha_n^{t-1} g_t \end{pmatrix} \cdot \begin{pmatrix} g^{-1}(\alpha_1) & & 0 \\ & \ddots & \\ 0 & & g^{-1}(\alpha_n) \end{pmatrix}$$

$$\Rightarrow P' = \begin{pmatrix} g_t & 0 & \cdots & 0 \\ g_{t-1} & g_t & \cdots & 0 \\ \vdots & \vdots & \ddots & \\ g_1 & g_2 & \cdots & g_t \end{pmatrix} \cdot \begin{pmatrix} 1 & 1 & \cdots & 1 \\ \alpha_1 & \alpha_2 & \cdots & \alpha_n \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_1^{t-1} & \alpha_2^{t-1} & \cdots & \alpha_n^{t-1} \end{pmatrix} \cdot \begin{pmatrix} g^{-1}(\alpha_1) & & 0 \\ & \ddots & \\ 0 & & g^{-1}(\alpha_n) \end{pmatrix}$$

↑ invertible, can be omitted

Secret Parity Check Matrix

$$\overline{P} = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ \alpha_1 & \alpha_2 & \cdots & \alpha_n \\ \vdots & & & \\ \alpha_1^{t-1} & \alpha_2^{t-1} & \cdots & \alpha_n^{t-1} \end{pmatrix} \cdot \begin{pmatrix} g^{-1}(\alpha_1) & & 0 \\ & \ddots & \\ 0 & & g^{-1}(\alpha_n) \end{pmatrix} \in \mathbb{F}_{2^m}^{t \times n}$$

Notice: L, g define $\overline{P}$.

McEliece Public Key

Let $\mathbb{F}_{2^m} = \mathbb{F}_2[y]/f(y)$ for some irreducible (over $\mathbb{F}_2$) f with $\deg(f) = m$.
Then elements $\beta \in \mathbb{F}_{2^m}$ can be written as $\beta = \beta_0 + \beta_1 y + \dots + \beta_{m-1} y^{m-1}$ with $\beta_i \in \mathbb{F}_2$.

Def: Let $\mathbb{F}_{2^m} = \mathbb{F}_2[y]/f(y)$. Then we call the map

$$\mathbb{F}_{2^m} \to \mathbb{F}_2^m, \quad \beta = \beta_0 + \beta_1 y + \dots + \beta_{m-1} y^{m-1} \longmapsto (\beta_0, \beta_1, \dots, \beta_{m-1})$$

the canonical embedding of $\mathbb{F}_{2^m}$ into $\mathbb{F}_2^m$ (with respect to f).

Apply the canonical embedding component-wise on $\overline{P}$:

$$\overline{P} \in \mathbb{F}_{2^m}^{t \times n} \xmapsto[\text{embedding}]{\text{canonical}} \hat{P} \in \mathbb{F}_2^{tm \times n} \xmapsto[\text{form}]{\text{systematic}} P = (I_{tm} \mid A)$$

$A \in \mathbb{F}_2^{tm \times (n-tm)}$

McEliece pk

transformations should hide secret L, g

Towards distance

Def: For $y \in \mathbb{F}_2^n$ we define the Goppa syndrome of y as $S_y(x) := \sum_{i=1}^{n} \frac{y_i}{x-\alpha_i} \mod g(x)$.

The support of y is defined as $\mathrm{supp}(y) = \{ i \in \{1, \dots, n\} \mid y_i = 1 \}$. ← y's 1-positions

The Goppa multiplier of $y \in \mathbb{F}_2^n$ is defined as $f_y(x) := \prod_{i \in \mathrm{supp}(y)} (x - \alpha_i)$.

Corollary: $c \in C(L,g) \Leftrightarrow \sum_{i=1}^{n} \frac{c_i}{x-\alpha_i} = 0 \mod g(x) \Leftrightarrow S_c(x) = 0$

Lemma: For $y \in \mathbb{F}_2^n$ we have $f'_y(x) = \sum_{i \in \mathrm{supp}(y)} \boxed{\prod_{\substack{j \in \mathrm{supp}(y) \\ j \neq i}} (x - \alpha_j)}$ ← derivative of i-th term

Proof: Apply product formula of differentiation.

$(uvw)' = u'vw + uv'w + uvw'$

Lemma: $c \in C(L,g) \Leftrightarrow f_c'(x) = 0 \mod g(x)$

Proof: From the previous corollary we have $c \in C(L,g) \Leftrightarrow s_c(x) = 0$.

Since $f_c(x) = \prod_{i \in supp(c)} (x - \alpha_i)$ and $g(x)$ is irreducible of $\deg(g) = t > 1$, we have

$$\gcd(f_c(x), g(x)) = 1.$$

Moreover, $s_c(x) \cdot f_c(x) = \sum_{i=1}^{n} \frac{c_i}{x-\alpha_i} \cdot \prod_{i \in supp(c)} (x-\alpha_i) = \sum_{i \in supp(c)} \prod_{\substack{j \in supp(c) \\ j \neq i}} (x - \alpha_j) = f_c'(x) \mod g(x)$.

It follows that $s_c(x) = 0 \mod g(x) \Leftrightarrow f_c'(x) = 0 \mod g(x)$. $\square$

Lemma: $C(L,g) = C(L,g^2)$

„$\supseteq$": Let $c \in C(L,g^2)$. Then $s_c(x) = 0 \mod g^2(x)$

$$\Rightarrow s_c(x) = 0 \mod g(x) \Rightarrow c \in C(L,g)$$

„$\subseteq$": Let $c \in C(L,g)$. Then $f_c'(x) = 0 \mod g(x)$.

Let $f_c'(x) = \sum_{i=1}^{n} i f_i x^{i-1}$. For even i we have $i f_i x^{\enclose{circle}{i-1}} = 0 \mod 2$. odd exponents vanish

$$\Rightarrow f_c'(x) = \sum_{i=0 \bmod 2}^{n} f_i \left(x^{\frac{i}{2}}\right)^2 = \left(\sum_{i=0 \bmod 2}^{n} f_i^{2^{m-1}} \cdot x^{\frac{i}{2}}\right)^2.$$

Recall that $a^2+b^2 = (a+b)^2$ and $a^{2^m} = a$ over $\mathbb{F}_{2^m}$.

Therefore $f_c'(x)$ is a square, implying that every irreducible factor of $f_c'(x)$ has to appear with even multiplicity. Thus $g^2(x) \mid f_c'(x)$

$$\Leftrightarrow f_c'(x) = 0 \mod g^2(x) \Leftrightarrow c \in C(L,g^2). \qquad \square$$

Goppa code distance

Theorem: Let $C(L,g)$ be a Goppa code with $\deg(g) = t$. Then $d(C) \geq 2t+1$.

Proof: Let $c \in C \setminus 0^n$ be a codeword of minimal weight $w(c) = d(C)$. We have

$$f_c'(x) = \sum_{i \in \mathrm{supp}(c)} \prod_{\substack{j \in \mathrm{supp}(c) \\ j \neq i}} (x - \alpha_i) = 0 \mod g^2(x).$$

Recall $C(L,g) = C(L,g^2)$.

$$\Rightarrow g^2(x) \mid f_c'(x) \Rightarrow \deg(f_c'(x)) = d(C) - 1 \geq \deg(g^2(x)) = 2t. \quad \square$$

QSI Spring School on PQ Crypto, Porto

Lecture : Codes (Part II : Deconstructions)

Alexander May, Ruhr-University Bochum

McEliece encryption/decryption

Goppa points Goppa polynomial

Gen: public key $P \in \mathbb{F}_2^{(n-k)\times n}$, secret key L, g

Enc: Embed m injectively in $\mathbb{F}_2^n$ with weight t: $m \mapsto e$

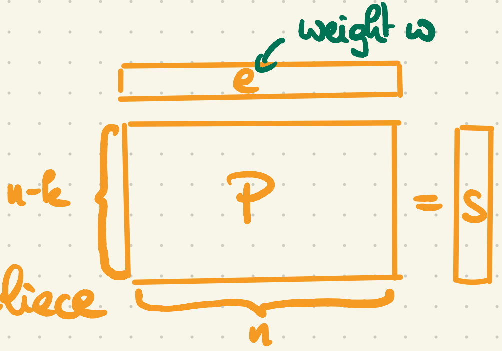
Encryption: $s = P \cdot e^t \in \mathbb{F}_2^{tm}$

syndrome

Dec: Recover e from syndrome s using the secret key L, g. (details later)

Invert the embedding to recover m from e.

Syndrome Decoding Problem

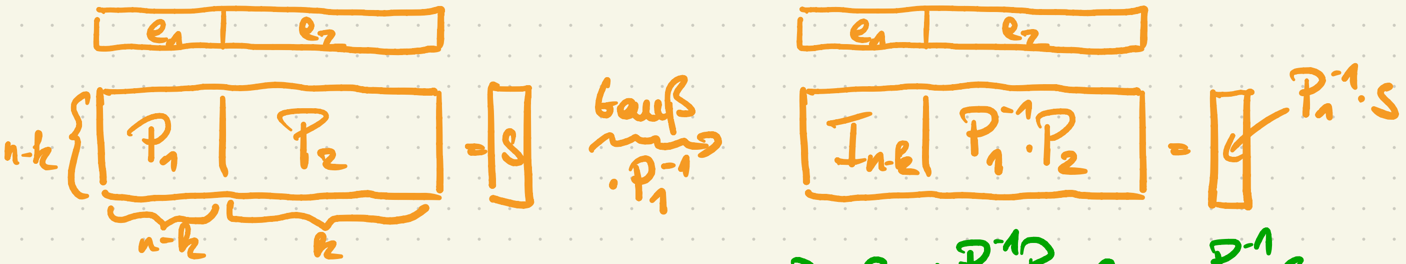


Given: $P \in \mathbb{F}_2^{(n-k)\times n}$, $s \in \mathbb{F}_2^{n-k}$, $\omega = t$ for McEliece

Find: $e \in \mathbb{F}_2^n$ with $\mathrm{wt}(e) = \omega$ and $s = P \cdot e^t$.

Information Set Decoding (Prange '62)

Idea:



$$\Rightarrow \; e_1 + P_1^{-1} \cdot P_2 \cdot e_2 = P_1^{-1} \cdot s$$

For $e_2 = 0^k$ we obtain $e_1 = P_1^{-1} \cdot s$, and therefore the solution $e = (e_1, 0^k)$.

Def: We call the first n-k columns of P an information set.

Pranges idea: Permute P s.t. the information set contains all ones of e.

Pranges ISD

Input: $P \in \mathbb{F}_2^{(n-k)\times n}$, $s \in \mathbb{F}_2^{n-k}$, $\omega = w(e)$

$$P \cdot H \cdot \overbrace{H^{-1} \cdot e}^{(e_1,e_2)} = s$$

① Repeat

② Repeat: Choose random permutation matrix $H \in \mathbb{F}_2^{n\times n}$. Let $PH = (\underbrace{P_1}_{n-k} | P_2)$.

Until P_1 is invertible.

Until $w(P_1^{-1} \cdot s) = \omega$. $e_1 = P_1^{-1} \cdot s$

Output: $e = H \cdot (P_1^{-1} \cdot s, 0^k)$ $H^{-1} \cdot e = (e_1, e_2)$

Complexity: ② has polynomial complexity.

① succeeds with probability $\frac{\binom{n-k}{\omega}}{\binom{n}{\omega}}$.

$$T = \tilde{O}\left(\frac{\binom{n}{\omega}}{\binom{n-k}{\omega}}\right), \quad M = \tilde{O}(1).$$

McEliece params 2^{143}

Grover Search ('96) / Amplitude Amplification ('97)

Let $\mathcal{A}$ be an algorithm with success probability p.

Theorem (Classical): On expectation we run $\frac{1}{p}$ instantiations of $\mathcal{A}$ until (first) success.

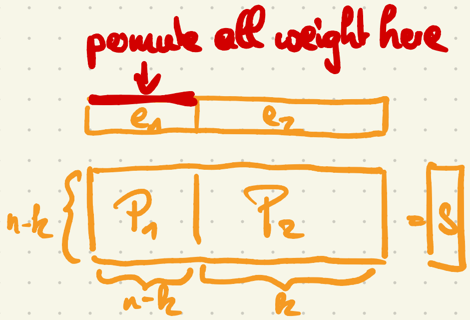
Proof: Expectation $E[X] = \frac{1}{p}$ of geometric distribution with $\Pr[X=n] = (1-p)^{n-1} \cdot p$.

Theorem (Quantum): On expectation we run $\sqrt{\frac{1}{p}}$ quantum instantiations of $\mathcal{A}$ until success.

(without proof)

Typical square root speedup, can be shown to be optimal.

Prange with Amplitude Amplification



Prange's runtime is dominated by finding π:

$$p = \Pr[\pi \text{ is good}] = \frac{\binom{n-k}{w}}{\binom{n}{w}}.$$

McEliece: $n = 3488,\ k = 2720,\ w = 64 \Rightarrow T_{\text{classic}} = \frac{1}{p} \approx 2^{143}$

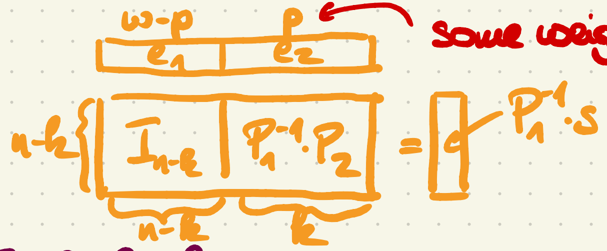
$$T_{\text{quantum}} = \sqrt{\frac{1}{p}} \approx 2^{72}$$

Note: Quantumly, McEliece has less than 80 bit security.

(But: Amplitude Amplification requires large quantum circuit depth.)

Lee-Brickell ISD ('88)

Idea: Relax the requirement that all error positions land in information set.



some weight outside

$$e_1 = P_1^{-1}\cdot s + P_1^{-1}\cdot P_2\cdot e_2$$

$$P\cdot H\cdot \overbrace{H^{-1}\cdot e}^{(e_1|e_2)} = s$$

Lee-Brickell algo

Input: $P\in \mathbb{F}_2^{(n-k)\times n}$, $s\in\mathbb{F}_2^{n-k}$, $\omega = w(e)$, $p \ll \omega$

```
① Repeat
    (1.1) Repeat: Choose permutation $H\in\mathbb{F}_2^{n\times n}$. Let $(P_1|P_2)=PH$. Until $P_1$ invertible.
    (1.2) For all $e_2\in\mathbb{F}_2^{k}(p)$        ← Brute force of $e_2$
  Until $w(\overbrace{P_1^{-1}\cdot s + P_1^{-1}\cdot P_2\cdot e_2}^{e_1}) = \omega - p$.
② Output $e = H(P_1^{-1}\cdot s + P_1^{-1}\cdot P_2\cdot e_2, e_2)$
```

Complexity Lee-Brickell

① $Pr(H \text{ good}) = \frac{\binom{n-k}{w-p}\cdot\binom{k}{p}}{\binom{n}{w}}$ better than Prange for $p>0$

(1.2) $|\mathbb{F}_2^k(p)| = \binom{k}{p}$ no cost in Prange

① · (1.2) $T = \frac{\binom{n}{w}}{\binom{n-k}{w-p}\cdot\cancel{\binom{k}{p}}}\cdot\cancel{\binom{k}{p}}$

omitting $\tilde{O}$ for readability

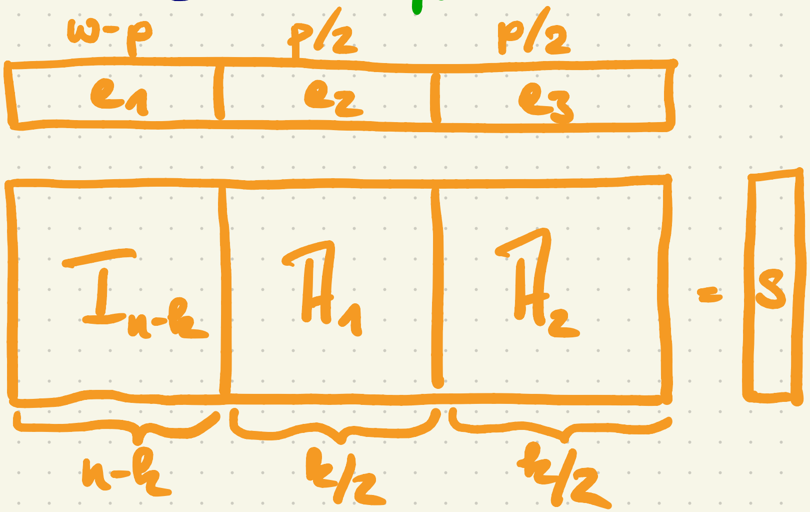
Since $w < \frac{n-k}{2}$, we maximize $\binom{n-k}{w-p}$ for the choice $p=0$

$\Rightarrow$ Minimal run time $T = \frac{\binom{n}{w}}{\binom{n-k}{w}}$ ← identical to Prange

Question: Lee-Brickell identical to Prange? What's the point?

Well, use MitM instead of Brute-Force.

MitM ISD (1st try)



MitM identity:

$$e_1 + H_1 \cdot e_2 + H_2 \cdot e_3 = s$$

3 unknowns, but only 2 sides

Solution 1: Remove unknown e_1 (next slide)

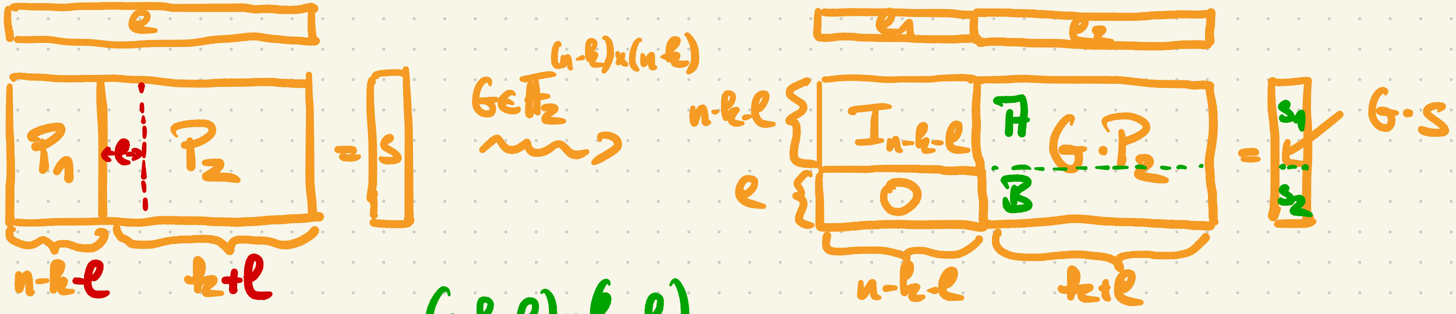
Solution 2 (LSH): Use approximate identity $H_1 \cdot e_2 \approx_{e_1} s + H_2 \cdot e_3$

locality sensitive hashing
(better, but a bit more advanced :))

identity on all but w-p positions

Leon's removal of e_1 ('88)

Leon's ℓ-window: Use semi-systematic form. Q: How to compute?



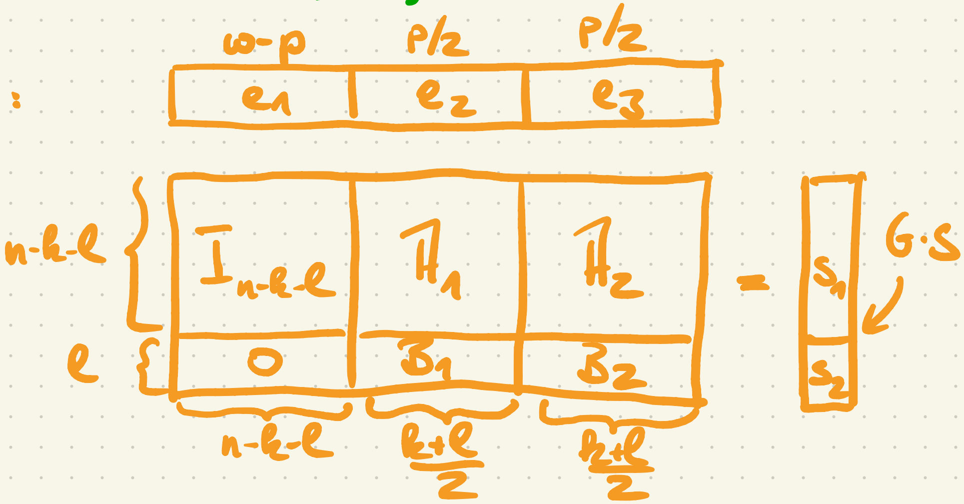
Let $G\cdot P_2 = \begin{pmatrix} \bar{H} \\ B \end{pmatrix}$ ← $(n-k-\ell)\times(k+\ell)$, ← $\ell\times(k+\ell)$ and $s = \begin{pmatrix} s_1 \\ s_2 \end{pmatrix} \begin{matrix} \}\, n-k-\ell \\ \}\, \ell \end{matrix}$. Then

(1) $e_1 = s_1 + \bar{H}\cdot e_2$

(2) $0 = s_2 + B\cdot e_2$ (we removed the annoying e_1)

Dumer-Stern ISD ('89)

Idea:



Identities: (1) $H_1 \cdot e_2 \approx_{e_1} s_1 + H_2 \cdot e_3$ approximate

(2) $B_1 \cdot e_2 = s_2 + B_2 \cdot e_3$ exact

Strategy: First check (2), then (1).

Dumer-Stern ISD

```
Input: $P \in \mathbb{F}_2^{(n-k)\times n}$, $s \in \mathbb{F}_2^{n-k}$, $\omega = w(e)$, $p \le \omega$, $\ell \le n-k$
① Repeat until success
    (1.1) Choose permutation $H \in \mathbb{F}_2^{n\times n}$.
                                                            We assume this is doable, otherwise repeat.
    (1.2) Compute semi-systematic form $G \cdot PH = \left(\begin{array}{c|cc} I_{n-k-\ell} & H_1 & H_2 \\ 0 & B_1 & B_2 \end{array}\right)$, $G \cdot s = \binom{s_1}{s_2}$.
    (1.3) For all $e_2 \in \mathbb{F}_2^{\frac{k+\ell}{2}}(\frac{p}{2})$: Compute L with entries $(B_1 e_2, e_2)$
    (1.4) For all $e_3 \in \mathbb{F}_2^{\frac{k+\ell}{2}}(\frac{p}{2})$:
        (1.4.1) For all $(s_2 + B_2 e_3, e_2) \in L$:          All $(e_2, e_3)$ satisfying (2).
            If $w(H_1 e_2 + H_2 e_3 + s_1) = \omega - p$, success.          Satisfy also (1)?
② Output: $e = H(\underbrace{H_1 e_2 + H_2 e_3 + s_1}_{e_1}, e_2, e_3)$
```

Complexity: ① $\Pr[H \text{ is good}] = \dfrac{\binom{n-k-\ell}{w-p}\cdot\binom{(k+\ell)/2}{p/2}^2}{\binom{n}{w}}$

$\boxed{1.3}$ $\left|\mathbb{F}_2^{\frac{k+\ell}{2}}\left(\frac{p}{2}\right)\right| = \binom{(k+\ell)/2}{p/2}$ $\quad\boxed{1.4}$ $\overbrace{\binom{(k+\ell)/2}{p/2}\cdot\binom{(k+\ell)/2}{p/2}\cdot 2^{-\ell}}^{\text{candidates in } \boxed{1.4.1}}$

$$\Rightarrow T = \frac{\binom{n}{w}}{\binom{n-k-\ell}{w-p}\cdot\binom{(k+\ell)/2}{p/2}^2}\cdot\binom{(k+\ell)/2}{p/2}\cdot\max\left\{1,\binom{(k+\ell)/2}{p/2}\cdot 2^{-\ell}\right\}$$

again omitting $\tilde{O}$

$$M = \binom{(k+\ell)/2}{p/2}$$

McEliece parameters: $n=3488$, $k=2720$, $w=64$

Prange for $\ell=k=0$: $T=2^{143}$, no memory

Optimized $p=10, \ell=46$: $T=2^{138}$, $M=2^{45}$

5 bit save for quite heavy memory

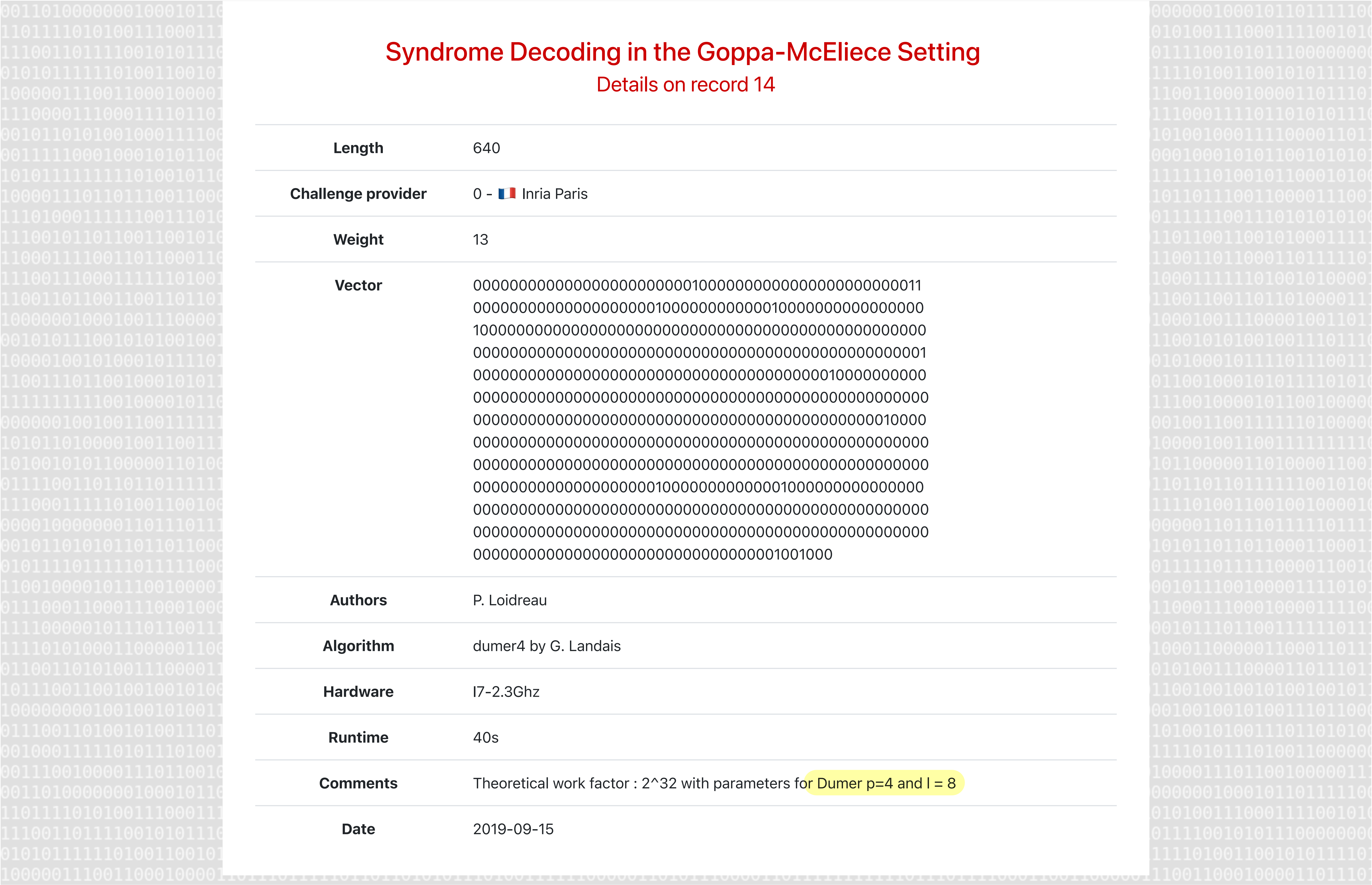
Syndrome Decoding in the Goppa-McEliece Setting

Hall of Fame

Length	Weight	Authors	Algorithm	Date	Details
1409	26	Shintaro Narisada, Hiroki Furue, Yusuke Aikawa, Kazuhide Fukushima, and Shinsaku Kiyomoto	MMT variant	2023-11-13	See details
1347	25	Daniel J. Bernstein, Tanja Lange, Christiane Peters	See https://isd.mceliece.org/1347.html for more information.	2023-02-24	See details
1284	24	Andre Esser, Alex May and Floyd Zweydinger	MMT variant	2021-08-16	See details
1223	23	Andre Esser, Alex May and Floyd Zweydinger	BJMM/MMT variant	2021-05-10	See details
1161	22	Shintaro Narisada, Kazuhide Fukushima, and Shinsaku Kiyomoto	Dumer	2021-01-10	See details
1101	21	Anders Nilson	Multi threads Dumer4, Gregory Landais impl.	2020-08-14	See details
1041	19	Shintaro Narisada, Kazuhide Fukushima, and Shinsaku Kiyomoto	Dumer	2020-08-11	See details
982	20	Noémie Bossard	Mutltithreaded Dumer4, Gregory	2020-	

			Landais original implementation	07-02	See details
923	19	Valentin Vasseur	Dumer	2020-03-17	See details
865	18	Valentin Vasseur	Dumer	2019-11-06	See details
808	17	Valentin Vasseur	Dumer	2019-11-06	See details
751	16	Valentin Vasseur	Dumer	2019-11-06	See details
695	14	Valentin Vasseur	Dumer	2019-09-22	See details
640	13	P. Loidreau	dumer4 by G. Landais	2019-09-15	See details
587	12	P. Loidreau	dumer4 by G. Landais	2019-09-15	See details
534	11	Francesco Tinarelli	-	2019-08-26	See details
482	11	Francesco Tinarelli	-	2019-08-24	See details
431	10	Francesco Tinarelli	-	2019-	

				08-24	See details
381	9	Francesco Tinarelli	-	2019-08-24	See details
333	8	Francesco Tinarelli	-	2019-08-23	See details
286	7	Julien Lavauzelle	Lee-Brickell	2019-08-13	See details
240	6	Julien Lavauzelle	Lee-Brickell	2019-08-13	See details
197	5	Julien Lavauzelle	Lee-Brickell	2019-08-13	See details
156	4	Julien Lavauzelle	Lee-Brickell	2019-08-13	See details
117	4	Julien Lavauzelle	Lee-Brickell	2019-08-13	See details
80	3	Julien Lavauzelle	Lee-Brickell	2019-08-13	See details
48	1	Aleksei Udovenko	-	2019-08-20	See details
20	1	Aleksei Udovenko	-	2019-	



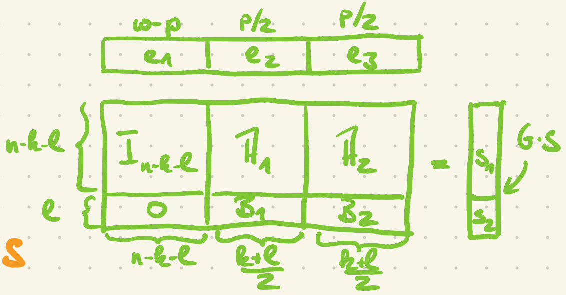
Syndrome Decoding in the Goppa-McEliece Setting

Details on record 14

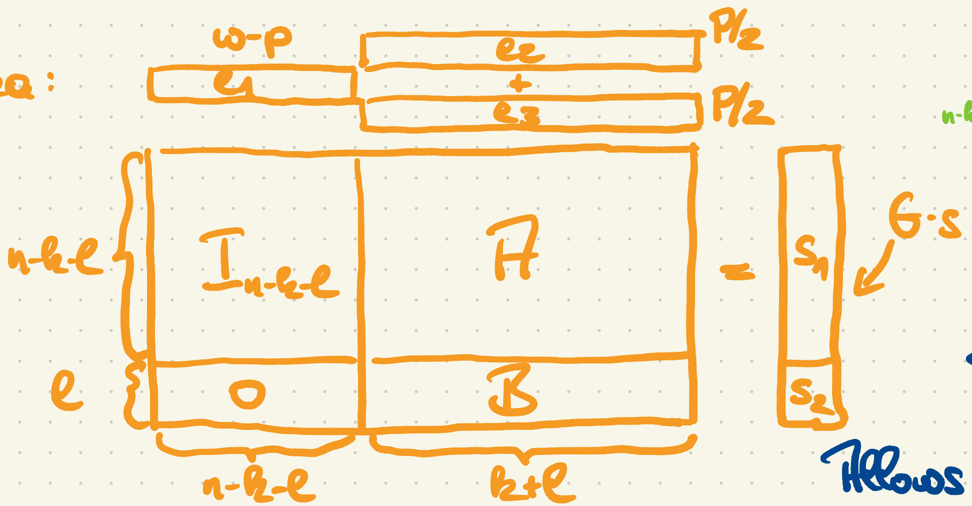
Length	640
Challenge provider	0 - 🇫🇷 Inria Paris
Weight	13
Vector	00000000000000000000000010000000000000000000000011 00000000000000000000100000000000010000000000000000 10000000000000000000000000000000000000000000000000 00000000000000000000000000000000000000000000000001 00000000000000000000000000000000000000010000000000 00000000000000000000000000000000000000000000000000 00000000000000000000000000000000000000000000010000 00000000000000000000000000000000000000000000000000 00000000000000000000000000000000000000000000000000 00000000000000000000100000000000001000000000000000 00000000000000000000000000000000000000000000000000 00000000000000000000000000000000000000000000000000 0000000000000000000000000000000001001000
Authors	P. Loidreau
Algorithm	dumer4 by G. Landais
Hardware	I7-2.3Ghz
Runtime	40s
Comments	Theoretical work factor : 2^32 with parameters for Dumer p=4 and l = 8
Date	2019-09-15

May-Meurer-Thomae (MMT '11)

Compare with Dumer-Stern



Idea:



$R = \binom{p}{p/2}$ representations

Allows to fix $\log(R) \approx p$ coordinates.

Identities: (1) $He_2 \approx_{e_1} s_1 + He_3$ approximate

(2) $Be_2 = s_2 + Be_3$ exact

Strategy: First check (2), then (1).

MMT

```
Input: $P \in \mathbb{F}_2^{(n-k)\times n}$, $s \in \mathbb{F}_2^{n-k}$, $\omega = w(e)$, $p \le \omega$, $\ell \le n-k$
① Repeat until success
    (1.1) Choose permutation $\Pi \in \mathbb{F}_2^{n\times n}$. Set $R = \binom{p}{p/2}$.
    (1.2) Compute semi-systematic form $G \cdot P\Pi = \begin{pmatrix} I_{n-k-\ell} & A \\ 0 & B \end{pmatrix}$, $G \cdot s = \binom{s_1}{s_2}$.
    (1.3) For all $e_2 \in \mathbb{F}_2^{k+\ell}(\frac{p}{2})$ : Compute $L_1 = \{(Be_2, e_2) \mid [Be_2]_p = 0\}$.   Requires $p \le \ell$.
    (1.4) For all $e_3 \in \mathbb{F}_2^{k+\ell}(\frac{p}{2})$: Compute $L_2 = \{(s_2 + Be_3, e_3) \mid [Be_3]_p = 0\}$.
        (1.4.1) For all $(Be_2, e_2, s_2 + Be_3, e_3) \in L_1 \times L_2$ with $Be_2 = s_2 + Be_3$
            If $w(A(e_2+e_3) + s_1) = \omega - \underbrace{w(e_2+e_3)}_{\le p}$, then success.
② Output: $e = \Pi(\underbrace{A(e_2+e_3) + s_1}_{e_1}, e_2 + e_3)$
```

Complexity: ① $\Pr[H \text{ is good}] = \dfrac{\binom{n-k-l}{w-p}\cdot\binom{k+l}{p}}{\binom{n}{w}}$

elements in L_1, L_2 already match in p coordinates

(1.3) + (1.4) $|L_1| = |L_2| = \dfrac{\binom{k+l}{p/2}}{\binom{p}{p/2}}$ (1.4.1) $|L_1|\cdot|L_2|\cdot 2^{l-p}$

$$\Rightarrow T = \frac{\binom{n}{w}}{\binom{n-k-l}{w-p}\cdot\binom{k+l}{p}} \cdot \frac{\binom{k+l}{p/2}}{\binom{p}{p/2}} \cdot \max\left\{ \frac{\binom{k+l}{p/2}}{\binom{p}{p/2}} \cdot 2^{l-p}, 1 \right\}$$

$$M = \frac{\binom{k+l}{p/2}}{\binom{p}{p/2}}$$

Here we assume that L_1, L_2 can be constructed in time $|L_1|, |L_2|$.

Exercise: Construct a MitM for L_1, L_2, good enough for McEliece params.

McEliece parameters: $n = 3488$, $k = 2720$, $w = 64$

Prange for $l = p = 0$: $T = 2^{143}$, no memory

Optimized $p = 18$, $l = 87$: $T = 2^{133}$, $M = 2^{54}$

yet another 5 bit

Dumer-Stern

$T = 2^{138}$, $M = 2^{45}$

Partial Key Exposure

Alexander May, Ruhr-University Bochum

Decoding Goppa Codes

Exercise: Implement it.

Theorem: Let $y = c + e$ for some $c \in C(L,g)$ with $w(e) \le t$.

Then c can be computed efficiently.

Proof: We have $s_y(x) = \sum_{i=1}^{n} \frac{y_i}{x-\alpha_i} = \underbrace{\sum_{i=1}^{n} \frac{c_i}{x-\alpha_i}}_{=0} + \sum_{i=1}^{n} \frac{e_i}{x-\alpha_i} = \sum_{i \in supp(e)} \frac{1}{x-\alpha_i} \mod g^2(x)$.

Multiplication by (the unknown) $f_e(x) = \prod_{i \in supp(e)} (x-\alpha_i) \mod g^2(x)$ yields

$$\underbrace{f_e(x)}_{\text{unknown}} \cdot \underbrace{s_y(x)}_{\text{known}} = \sum_{i \in supp(e)} \prod_{\substack{j \in supp(e) \\ j \neq i}} (x-\alpha_j) = \underbrace{f_e'(x)}_{\text{unknown}} \mod g^2(x).$$

We have $\deg(f_e) = t$ and $\deg(f_e') = t-1$.

Why only $2t-1$ unknowns?

Solve the $2t$ equations in the $2t-1$ unknown coeffs of $f_e(x)$ and $f_e'(x)$.

Factor $f_e(x)$ over $\mathbb{F}_{2^m}[x]$ in linear factors. Determine $supp(e)$ from α_i's. ■

Partial Key Exposure

Recall: McEliece secret key: $L=\{\alpha_1,\dots,\alpha_n\}$, $g(x)\in\mathbb{F}_{2^m}[x]$, $\deg(g)=t$

$n=3488$, $m=12$, $t=64$

public key: $P=(I_{tm}|A)\in\mathbb{F}_2^{tm\times n}$ $\quad tm=768$

ciphertext: $c=P\cdot e^t$ with $w(e)=t$

↖ embedding of m into $\mathbb{F}_2^n(t)$

Question: Can we reconstruct (L,g) from partial information?

Motivation Partial Key Recovery attack: Obtain partial information from side channels.

Secret Key Recovery from All Goppa points

Theorem: Given $P \in \mathbb{F}_2^{tm \times n}$ and $L = (\alpha_1, \alpha_2, \dots, \alpha_n) \in \mathbb{F}_{2^m}^n$. L suffices to compute g(x).

Then g(x) can be computed efficiently.

Proof: Compute a codeword $c \in \mathbb{F}_2^n$ with $P \cdot c = 0$. Exercise: Give an algorithm.

$$\Rightarrow f_c'(x) = \sum_{i \in supp(c)} \prod_{\substack{j \in supp(c) \\ j \neq i}} (x - \alpha_j) \quad \text{and} \quad g(x) \mid f_c'(x).$$

↖ known

Factor $f_c'(x) \in \mathbb{F}_{2^m}[x]$ into irreducible factors.

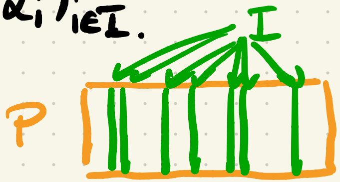
If there is a unique deg-t factor, output g(x).

Otherwise restart with a different codeword c. ■

Secret Key Recovery from $tm+1$ Goppa points

We know only points from I.

Theorem: (Kirshanova, May, '22) Given $P\in\mathbb{F}_2^{tm\times n}$, $I\subseteq\{1,\dots,n\}$, $|I|\geq tm+1$, $(\alpha_i)_{i\in I}$.
Then $g(x)$ can be computed efficiently.



Proof: Let $[P]_I\in\mathbb{F}_2^{tm\times|I|}$ denote the projection of P to the coords in I.

Compute $c'\in\mathbb{F}_2^{|I|}$ with $[P]_I\cdot c=0$. Again, how?

Expand c' with zeros to $c\in C(L,g)$ having $\mathrm{supp}(c)\subseteq I$.

$$\Rightarrow f_c'(x)=\sum_{i\in\mathrm{supp}(c)}\ \prod_{\substack{j\in\mathrm{supp}(c),\\ j\neq i}}(x-\alpha_j) \quad\text{and}\quad g(x)\mid f_c'(x).$$

only known α_j's from I

Find $g(x)$ from factoring $f_c'(x)$ over $\mathbb{F}_{2^m}[x]$. ■

(n, t, m)	$\ell = tm + 1$	$\lvert\mathcal{L}\rvert = 1$	$\ell = tm + 2$	$\lvert\mathcal{L}\rvert = 1$	Av. time
$(3488, 64, 12)$	769	97%	770	100%	18 sec
$(4608, 96, 13)$	1249	99%	1250	100%	54 sec
$(6960, 119, 13)$	1548	99%	1549	100%	91 sec
$(8192, 128, 13)$	1665	99%	1666	100%	105 sec

Table: Recovery of Goppa polynomial $g(x)$.

Recovery of Remaing Points

Exercise: $A \cdot x = b$ is solvable iff $\operatorname{rank}(A) = \operatorname{rank}(A|b)$.

Theorem: (Kirshanova, May, '22) Given $P \in \mathbb{F}_2^{tm \times n}$, $I \subseteq \{1,\dots,n\}$, $|I| \geq tm+1$, $(\alpha_i)_{i \in I}$, $g(x) \in \mathbb{F}_{2^m}[x]$. Then $L = (\alpha_1, \dots, \alpha_n)$ can be recovered efficiently.

Proof: Let us recover α_r for some $r \in \{1,\dots,n\} \setminus I$.

Assume for simplicity that $\operatorname{rank}([P]_I) = tm$. Solve linear equation

$$[P]_I \cdot c' = [P]_r \longleftarrow \text{r-th column of } P$$

$\operatorname{rank}([P]_I) = tm$
$= \operatorname{rank}([P]_{I \cup r})$

Expand c' with zeros to $c \in C(L,g)$ with $\operatorname{supp}(c) \subseteq I \cup \{r\}$.

$$\Rightarrow \sum_{i \in \operatorname{supp}(c)} \frac{1}{x - \underset{\text{known}}{\alpha_i}} = \frac{1}{x - \overset{\text{unknown}}{\alpha_r}} \mod \underset{\text{known}}{g(x)}$$

Compute $\left(\sum_{i \in \operatorname{supp}(c)} \frac{1}{x - \alpha_i} \right)^{-1} = x - \alpha_r \mod g(x)$, read of α_r. ■

(n, t, m)	$\ell = tm + 1$	time
$(3488, 64, 12)$	769	42 sec
$(4608, 96, 13)$	1249	130 sec
$(6960, 119, 13)$	1548	167 sec
$(8192, 128, 13)$	1665	183 sec

Notice: 2^{13}

Table: Experimental results for point recovery.

Support Splitting Algorithm

Setting: We know all Goppa points, but not their order.

Example McEliece: $n = 8192 = 2^{13} = 2^m \Rightarrow L = \mathbb{F}_{2^m}$

Question: Assume that we know $g(x)$ and $L = \{\alpha_1, \ldots, \alpha_n\}$, can we order L?

Theorem (Sendrier's Support Splitting '00): Let $P \in \mathbb{F}_2^{(n-k)\times n}$ the parity check matrix of C. Let $P' = S \cdot P \cdot \Pi$ for some invertible $S \in \mathbb{F}_2^{(n-k)\times(n-k)}$ and permutation $\Pi \in \mathbb{F}_2^{n\times n}$. Then one can (efficiently) find the permutation $\Pi \in \mathbb{F}_2^{n\times n}$.

Proof omitted. Nice algorithm, but tricky analysis.

Idea of attack (knowing $g(x)$):

- Let $\mathbb{F}_{2^m} = \{\alpha_1, \alpha_2, \dots, \alpha_n\}$.
- Construct matrix

$$H = \begin{pmatrix} 1 & 1 & & 1 \\ \alpha_1 & \alpha_2 & \dots & \alpha_n \\ \vdots & \vdots & & \vdots \\ \alpha_1^{t-1} & \alpha_2^{t-1} & & \alpha_n^{t-1} \end{pmatrix} \cdot \begin{pmatrix} g(\alpha_1) & & & 0 \\ & g(\alpha_2) & & \\ & & \ddots & \\ 0 & & & g(\alpha_n) \end{pmatrix} \in \mathbb{F}_{2^m}^{t \times n}.$$

- Apply canonical embedding $H \to P' \in \mathbb{F}_2^{tm \times n}$.
- Run Support Splitting on McEliece public key P and P' to find Π.
- Apply Π to recover L.

Notice: Brute-Force on $g(x) \in \mathbb{F}_{2^m}[x]$ costs $\tilde{O}(2^{tm})$ trials. $tm = 128 \cdot 13$

Open research question: Key security of McEliece.

Wrap Up (Lessons Learned)

- McEliece is quite an elegant encryption scheme. if you like (linear) algebra
- Lots of cryptanalysis in theory and practice: Prange, Dumer-Stern, MMT, ..
- Almost square root speedup quantumly.
- Secret key security ≫ Syndrome decoding security?
- Quite efficient Partial Key Exposure attacks.

Some advertisement: Try bochum-challeng.es.