Une attaque polynomiale du schéma de McEliece basé sur les codes géométriques

A. Couvreur\textsuperscript{1}  I. Márquez-Corbella\textsuperscript{1}  R. Pellikaan\textsuperscript{2}

\textsuperscript{1}INRIA Saclay & LIX
\textsuperscript{2}Department of Mathematics and Computing Science, TU/e.

Caramel Seminars - Thursday June 19, 2014
Introduction to Coding Theory

- An \([n, k]\) linear code \(C\) over \(\mathbb{F}_q\) is a \(k\)-dimensional subspace of \(\mathbb{F}_q^n\).
- Its size is \(M = q^k\), the information rate is \(R = \frac{k}{n}\) and the redundancy is \(n - k\).
- The generator matrix of \(C\) is a \(k \times n\) matrix \(G\) whose rows form a basis of \(C\), i.e.
  \[ C = \left\{ xG \mid x \in \mathbb{F}_q^k \right\}. \]
- The parity-check matrix of \(C\) is an \((n - k) \times n\) matrix \(H\) whose nullspace is generated by the codewords of \(C\), i.e.
  \[ C = \left\{ y \in \mathbb{F}_q^n \mid Hy^T = 0 \right\}. \]
- The hamming distance between \(x, y \in \mathbb{F}_q^n\) is \(d_H(x, y) = |\{i \mid x_i \neq y_i\}|.\)
- The minimum distance of \(C\) is
  \[ d(C) = \min \{d_H(c_1, c_2) \mid c_1, c_2 \in C \text{ and } c_1 \neq c_2\}. \]

**Figure:** If \(d(C) = 3\)

**Figure:** If \(d(C) = 4\)
Decoding Linear Codes

The Decoding problem:

**Input:** a Generator matrix \( G \in \mathbb{F}_q^{k \times n} \) of \( C \) and the received word \( y \in \mathbb{F}_q^n \)

**Output:** A closest codeword \( c \), i.e.

\[
    c \in C : d_H(c, y) = \min \{ d_H(\hat{c}, y) | \hat{c} \in C \}
\]

Decoding arbitrary linear codes: Exponential complexity

---

Decoding Special Classes of Codes

Efficient decoding algorithms up to half the minimum distance for:

1. Generalized Reed-Solomon codes
2. Goppa codes
3. Algebraic Geometry codes

**Polynomial complexity** \( \sim O(n^3) \)

- Peterson, Arimoto, 1960
- Berlekamp-Massy, 1963
- Skorobogatov-Vladut, 1990
- Sakata, 1990
- Feng-Rao, Duursma 1993
- Sudam, Guruswami, 1997
Public-Key Cryptosystems

Most PKC are based on number-theoretic problems

It can be attacked in polynomial time using Shor’s algorithm

Quantum Computer

RSA
ECDSA
DSA
HECC
ECC
McEliece introduced the first PKC based on Error-Correcting Codes in 1978.

Advantages:
1. Fast encryption (matrix-vector multiplication) and decryption functions.
2. Interesting candidate for post-quantum cryptography.

Drawback:
- Large key size.

R. J. McEliece.

A public-key cryptosystem based on algebraic coding theory.
**McEliece Cryptosystem**

$\rightarrow \ t \in \mathbb{N}^* \quad \Rightarrow \quad \text{Error-correcting capacity of } C$

Consider any triplet:

$C, \quad A_C (t)$

$\rightarrow \ [n, k]_q \ \text{linear code} \ \text{with an efficient decoding algorithm}$

$\Rightarrow \ \text{Let } G \text{ be a non structured generator matrix of } C.$

$\rightarrow \ \text{"Efficient" decoding algorithm} \ \text{for } C \ \text{which corrects up to } t \ \text{errors}.$
**McEliece Cryptosystem**

**Key Generation**

Given:
1. **McEliece Public Key:** $K_{pub} = (G, t)$
2. **McEliece Private Key:** $K_{secret} = \langle A \rangle$

**Encryption**

Encrypt a message $m \in \mathbb{F}_q^k$ as

$$y = mG + e$$

where $e$ is a random error vector of weight at most $t$.

**Decryption**

Using $K_{secret}$, the receiver obtain $m$. 

---

**Example**

**§5.1 Compute $B$**
PROPOSALS

GRS codes
Subcodes of GRS codes
Binary Reed-Muller codes
AG codes
Binary Goppa codes

Several Proposals

AG codes
Binary Goppa codes
The class of GRS codes was proposed by Niederreiter in 1986 for code-based PKC.

Sidelnikov-Shestakov in 1992 introduced an algorithm that breaks this proposal in polynomial time.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Key size</th>
<th>Security level</th>
</tr>
</thead>
<tbody>
<tr>
<td>[256, 128, 129]_{256}</td>
<td>67 ko</td>
<td>$2^{95}$</td>
</tr>
</tbody>
</table>
Berger and Loidreau in 2005 propose another version of the Niederreiter scheme designed to resist the Sidelnikov-Shestakov attack.

- **Main idea:** work with subcodes of the original GRS code.

**Attacks:**

- **Wieschebrink:** (2010)
  - Presents the first feasible attack to the Berger-Loidreau cryptosystem but is impractical for small subcodes.
  - Notes that if the square code of a subcode of a GRS code of parameters \([n, k]_q\) is itself a GRS code of dimension \(2k - 1\) then we can apply Sidelnikov-Shestakov attack.

- **M-Márquez-Pellikaan:** (2012) Give a characterization of the possible parameters that should be used to avoid attacks on the Berger-Loidreau cryptosystem.
Wieschebrick (2010) and Baldi et al. (2011) proposed other variants of the Niederreiter scheme.

 Attacks: Couvreur et al. (2013) provide a cryptanalysis of these schemes.
Binary Reed-Muller codes

- The class of **Binary Reed-Muller** codes was proposed by **Sidelnikov** in **1994** for code-based PKC.

- **Minder-Shokrollahi** in **2007** presents a sub-exponential time attack.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Key size</th>
<th>Security level</th>
</tr>
</thead>
<tbody>
<tr>
<td>$[1024, 176, 128]_2$</td>
<td>22.5 ko</td>
<td>$2^{72}$</td>
</tr>
<tr>
<td>$[2048, 232, 256]_2$</td>
<td>59, 4 ko</td>
<td>$2^{93}$</td>
</tr>
</tbody>
</table>
AG codes

In 1996 Janwa and Moreno propose to use AG codes for the McEliece cryptosystem.

This system was broken for:


GRS codes are Algebraic Geometry codes on the projective line.


4. We can retrieve the model of the curve (in polynomial time) by M-Martínez-Pellikaan-Ruano in 2013.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Key size</th>
<th>Security level</th>
</tr>
</thead>
<tbody>
<tr>
<td>$[171, 109, 61]_{128}$</td>
<td>16 ko</td>
<td>$2^{66}$</td>
</tr>
</tbody>
</table>

It is NOT broken.
The class of binary goppa codes was proposed by McEliece in 1977 for code-based PKC.

✓ McEliece with Goppa codes has resisted cryptanalysis so far!!

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Key size</th>
<th>Security level</th>
</tr>
</thead>
<tbody>
<tr>
<td>$[1024, 524, 101]_2$</td>
<td>67 ko</td>
<td>$2^{62}$</td>
</tr>
<tr>
<td>$[2048, 1608, 48]_2$</td>
<td>412 ko</td>
<td>$2^{96}$</td>
</tr>
</tbody>
</table>
Proposals

- GRS codes
- Subcodes of GRS codes
- Binary Reed-Muller codes
- AG codes
- Binary Goppa codes

Several Proposals

- §5.1 Compute on degenerate complexity examples
- Hermitian curves
- Suzuki curves

Conclusions
For all \( a, b \in \mathbb{F}_q^n \) we define:

- **Star Product:** \( a \ast b = (a_1 b_1, \ldots, a_n b_n) \in \mathbb{F}_q^n \)

- **Standard Inner Product:** \( \langle a, b \rangle = \sum_{i=1}^{n} a_i b_i \in \mathbb{F}_q \)

For all subsets \( A, B \subseteq \mathbb{F}_q^n \) we define:

- \( A \ast B = \{ a \ast b \mid a \in A \text{ and } b \in B \} \)

For \( B = A \implies A \ast A \) is denoted as \( A^{(2)} \)

- \( A \perp B \iff \langle a, b \rangle = 0 \quad \forall \; a \in A \text{ and } b \in B \)
Decoding by Error Correcting Pairs

Let $C$ be a linear code. We denote by:

- $k(C)$ = dimension of $C$
- $d(C)$ = minimum distance of $C$

Error-correcting pairs (ECP)

Let $C$ be an $\mathbb{F}_q$ linear code of length $n$. The pair $(A, B)$ of $\mathbb{F}_q$-linear codes of length $n$ is a $t$-ECP for $C$ over if the following properties hold:

- **E.1** $(A \ast B) \perp C$.
- **E.2** $k(A) > t$.
- **E.3** $d(B^\perp) > t$.
- **E.4** $d(A) + d(C) > n$.

An $[n, k]_q$ code which has a $t$-ECP over $\mathbb{F}_q$ has a decoding algorithm with complexity $\mathcal{O}(n^w)$. 

Let $C$ be a linear code. We denote by:

- $k(C)$ = dimension of $C$
- $d(C)$ = minimum distance of $C$
Let:

- $C$, $A$ and $B$ be linear subspaces of $\mathbb{F}_q^n$
- $y \in \mathbb{F}_q^n$ be the received word with error vector $e$

Compute:

$$K_y = \{ a \in A \mid \langle y, a \ast b \rangle = 0, \text{ for all } b \in B \}$$

**Remark: Condition 1**

If $A \ast B \subseteq C^\perp \implies K_y = K_e$

Let $J$ be a subset of $\{1, \ldots, n\}$, define:

$$A(J) = \{ a \in A \mid a_j = 0, \text{ for all } j \in J \}$$

**Lemma 1: Condition 3**

Let $I = \text{supp}(e)$ and $A \ast B \subseteq C^\perp$. If $d(B^\perp) > t \implies A(I) = K_y$
**Lemma 2: Condition 2**

If $l = \text{supp}(e)$ and $k(A) > t \implies \exists a \in K_y \setminus \{0\}$

**Lemma 3: Condition 4**

Let $a \in K_y \setminus \{0\}$ and define $J = \{j \mid a_j = 0\}$. Then:

1. If $d(B^\perp) > t$ then $l = \text{supp}(e) \subseteq J$
2. If $d(A) + d(C) > n$ then there exists a unique solution to:

\[
Hx^T = Hy^T \quad \text{such that } x_j \neq 0 \text{ for all } j \in J
\]
Decoding by Error-Correcting Pairs (ECP) III

1. Compute:
   \[ K_y = \{ a \in A \mid \langle y, a \ast b \rangle = 0, \text{ for all } b \in B \} \]

   Find the zero space of a set of linear equations over \( \mathbb{F}_q \).

2. If \( K_y = 0 \) ⟺ The received word has more than \( t \) errors
   → Else take a nonzero \( a \in K_y = A(I) \) and define \( J = \{ j \mid a_j = 0 \} \)

3. Find \( e \in \mathbb{F}_q^n \) by solving the following linear equation (which has a unique solution):
   \[ Hx^T = Hy^T \quad \text{such that} \quad x_j \neq 0 \text{ for } j \in J \]

   Solve linear equations over \( \mathbb{F}_q \).

Complexity: \( \sim \mathcal{O}(n^w) \)
Let
- \( \mathbf{a} = (a_1, \ldots, a_n) \) be an \( n \)-tuple of mutually distinct elements of \( \mathbb{F}_q \).
- \( \mathbf{b} = (b_1, \ldots, b_n) \) be an \( n \)-tuple of nonzero elements of \( \mathbb{F}_q \).
- \( k \in \mathbb{N} : k < n \)

The **GRS code** \( \text{GRS}_k(\mathbf{a}, \mathbf{b}) \) is defined by:

\[
\text{GRS}_k(\mathbf{a}, \mathbf{b}) = \{ \mathbf{b} \ast f(\mathbf{a}) = (b_1 f(a_1), \ldots, b_n f(a_n)) \mid f \in \mathbb{F}_q[X]_{<k} \}
\]
The GRS<sub>k</sub>(<a, b>) is an MDS code with parameters $[n, k, n - k + 1]_q$.

A generator matrix of GRS<sub>k</sub>(<a, b>) is given by

$$G_{a, b} = \begin{pmatrix}
    b_1 & \cdots & b_n \\
    b_1 a_1 & \cdots & b_n a_n \\
    \vdots & \ddots & \vdots \\
    b_1 a_1^{k-1} & \cdots & b_n a_n^{k-1}
\end{pmatrix} \in F_q^{k \times n}$$

The dual of a GRS code is again a GRS code. In particular:

$$\text{GRS}_{k}(a, b) \perp = \text{GRS}_{n-k}(a, c)$$

for some $c$ explicitly known.

The GRS<sub>k</sub>(<a, b>) \perp is an MDS code with parameters $[n, n - k, k + 1]_q$. 

The GRS codes are MDS codes, which means that they have maximum distance separability. This property is crucial for error correction in communication systems.
Note that: \( \text{GRS}_k(a, b) \ast \text{GRS}_l(a, c) = \text{GRS}_{k+l-1}(a, b \ast c) \)

Let

\[
A = \text{GRS}_{t+1}(a, b_1), \quad B = \text{GRS}_t(a, b_2)
\]

and

\[
C = \text{GRS}_{2t}(a, b_1 \ast b_2)\perp
\]

then \((A, B)\) is a \(t\)-ECP for \(C\).

- **E.1** \(A \ast B = \text{GRS}_{2t}(a, b_1 \ast b_2) = C\perp \Rightarrow (A \ast B) \perp C\)
- **E.2** \(k(A) > t\)
- **E.3** \(B\perp = \text{GRS}_{n-t}(a, c_2) \Rightarrow d(B\perp) = t + 1 > t\)
- **E.4** \(d(A) + d(C) = (n - t) + (2t + 1) > n\)
Conversely, let $C = \text{GRS}_{n-2t}(a, b)$ then

$$A = \text{GRS}_{t+1}(a, c) \quad \text{and} \quad B = \text{GRS}_t(a, 1)$$

is a $t$-ECP for $C$ where $c \in (\mathbb{F}_q \setminus \{0\})^n$ verifies that

$$C^\perp = \text{GRS}_{n-2t}(a, b)^\perp = \text{GRS}_{2t}(a, c).$$

Moreover an $[n, n-2t, 2t+1]_q$ code that has a $t$-ECP is a GRS code.
An AG code is defined by a triplet

\((\mathcal{X}, \mathcal{P}, E)\)

- \(\mathcal{X}\) is an algebraic curve of genus \(g\) over the finite field \(\mathbb{F}_q\).
- \(\mathcal{P} = (P_1, \ldots, P_n)\) is an \(n\)-tuple of mutually distinct \(\mathbb{F}_q\)-rational points of \(\mathcal{X}\).
- \(E\) is the evaluation map.

Algebraic Curve = Smooth, Projective and Geometrically Connected Curve

Whose defining equations are polynomials with coefficients in \(\mathbb{F}_q\).

\(D_\mathcal{P}\) denotes the divisor \(D_\mathcal{P} = P_1 + \cdots + P_n\).
Algebraic Geometry codes

- An AG code is defined by a triplet

$$(\mathcal{X}, \mathcal{P}, E)$$

- $\mathcal{X}$ is an algebraic curve of genus $g$ over the finite field $\mathbb{F}_q$

**Algebraic Curve** = **Smooth**, **Projective** and **Geometrically Connected** Curve

Whose defining equations are polynomials with coefficients in $\mathbb{F}_q$.

- $\mathcal{P} = (P_1, \ldots, P_n)$ is an $n$-tuple of mutually distinct $\mathbb{F}_q$-rational points of $\mathcal{X}$

$D_{\mathcal{P}}$ denotes the divisor $D_{\mathcal{P}} = P_1 + \cdots + P_n$
An AG code is defined by a triplet \((\mathcal{X}, \mathcal{P}, E)\).

\[ E \text{ is an } \mathbb{F}_q\text{-divisor of } \mathcal{X} \text{ such that } \text{supp}(E) \cap \text{supp}(D_P) = \emptyset \]
Divisors on Curves

A **divisor** \( D \) on \( \mathcal{X} \) is a formal finite sum:

\[
D = \sum_{P \in \mathcal{X}} n_P P \quad \text{with } n_P \in \mathbb{Z} \text{ and } P \in \mathcal{X}
\]

- If \( n_P \geq 0 \) for all \( P \in \mathcal{X} \) then \( D \) is an **Effective Divisor**, \( (D \geq 0) \).
- **Support of the divisor** \( D \): \( \text{supp}(D) = \{ P \mid n_P \neq 0 \} \)
- **Degree of the divisor** \( D \): \( \text{deg}(D) = \sum_{P \in \mathcal{X}} n_P \text{deg}(P) \)
**Algebraic Geometry codes III**

**Divisor of rational functions**

The divisor of \( f \in \mathbb{F}_q(\mathcal{X}) \) is defined to be:

\[
(f) = \sum_{P \text{ zero of } f} v_P(f)P - \sum_{P \text{ pole of } f} v_P(f)P
\]

\[
(f)_0 \quad (f)_\infty
\]
**Algebraic Geometry codes IV**

---

**Space of Rational Functions associated to the divisor \( E \)**

\[
L(E) = \{ f \in \mathbb{F}_q(x) \mid f = 0 \text{ or } (f) + E \geq 0 \}
\]

Intuitively: \( f \in L(E) \iff f \) has enough zeros and not too many poles

---

**Riemann-Roch Theorem**

\[
\dim(L(E)) \geq \deg(E) + 1 - g
\]

Furthermore, if \( \deg(E) > 2g - 2 \) then \( \dim(L(E)) = \deg(E) + 1 - g \)
A **L** **G** **E** **B** **R** **A** **I** **C** **G** **E** **O** **M** **E** **T** **R** **Y** **C** **O** **D** **E** **S** V

→ Let us consider the triplet:

\[ (\mathcal{X}, \mathcal{P}, E) \]

→ \( \mathcal{X} \) is an algebraic curve of genus \( g \) over the finite field \( \mathbb{F}_q \).

→ \( \mathcal{P} \) is an \( n \)-tuple of distinct \( \mathbb{F}_q \)-rational points of \( \mathcal{X} \).

→ \( E \) is an \( \mathbb{F}_q \)-divisor of \( \mathcal{X} \) such that \( \text{supp}(E) \cap \text{supp}(D_{\mathcal{P}}) = \emptyset \)

Since \( \text{supp}(E) \cap \text{supp}(D_{\mathcal{P}}) = \emptyset \) the following **evaluation map** is well defined:

\[
\text{ev}_{\mathcal{P}} : \quad L(E) \longrightarrow \mathbb{F}_q^n \\
\quad f \quad \longmapsto \quad \text{ev}_{\mathcal{P}}(f) = (f(P_1), \ldots, f(P_n))
\]

### Algebraic Geometry Code (AG code)

The **AG code** associated to the triplet \((\mathcal{X}, \mathcal{P}, E)\) is:

\[
\mathcal{C}_L(\mathcal{X}, \mathcal{P}, E) = \{ \text{ev}_\mathcal{P}(f) = (f(P_1), \ldots, f(P_n)) \mid f \in L(E) \}
\]
If \( \{ f_1, \ldots, f_k \} \) is a basis of \( L(E) \) then

\[
G = \begin{pmatrix}
    f_1(P_1) & \cdots & f_1(P_n) \\
    \vdots & \ddots & \vdots \\
    f_k(P_1) & \cdots & f_k(P_n)
\end{pmatrix} \in \mathbb{F}_q^{k \times n}
\]

is a generator matrix of the code \( C_L(\mathcal{X}, \mathcal{P}, E) \).

**Theorem I [Parameters of an AG code]**

Let \( C = C_L(\mathcal{X}, \mathcal{P}, E) \). If \( \deg(E) < n \) then

\[
k(C) \geq \deg(E) + 1 - g \quad \text{and} \quad d(C) \geq n - \deg(E)
\]

Moreover, if \( n > \deg(E) > 2g - 2 \) then \( k(C) = \deg(E) - g + 1 \).
**DUAL OF AN AG CODE**

Let:

- $\omega$ be a **differential form** with a simple pole and residue 1 at $P_j$ for all $j = 1, \ldots, n$.
- $K$ be the **canonical divisor** of $\omega$.

Then

$$C_L(X, P, E)^\perp = C_L(X, P, E_\perp)$$

with $E_\perp = D_P - E + K$ and $\deg(E_\perp) = n - \deg(E) + 2g - 2$

---

**THEOREM II [PARAMETERS OF THE DUAL OF AN AG CODE]**

Let $C = C_L(X, P, E)$. If $\deg(E) > 2g - 2$ then

$$k(C^\perp) \geq n - \deg(E) - 1 + g \quad \text{and} \quad d(C^\perp) \geq \deg(E) - 2g + 2$$

Moreover, if $n > \deg(E) > 2g - 2$ then $k(C^\perp) = n - \deg(E) - 1 + g$
Consider the AG code
\[ C = C_L \left( \mathcal{X}, \mathcal{P}, E \right) \]

**Theorem [Pellikaan - 1992]**

The pair of codes \((A, B)\) defined by
\[ A = C_L(\mathcal{X}, \mathcal{P}, F) \quad \text{and} \quad B = C_L(\mathcal{X}, \mathcal{P}, E - F) \]

with \(\deg(E) > \deg(F) \geq t + g\) is a \(t\)-ECP for \(C\).

Such a pair always exists whenever
\[ \deg(E) > 2g - 2 \quad \text{and} \quad t = t^* = \left\lfloor \frac{d^* - 1 - g}{2} \right\rfloor. \]

where \(d^* = \deg(E) - 2g + 2\) is the designed minimum distance of \(C\).
Corollary [MAIN COROLLARY]

Let $C = C_L(\mathcal{X}, \mathcal{P}, E)\perp$ and $B = C_L(\mathcal{X}, \mathcal{P}, E - F)$ with $\deg(F) \geq t + g$.

And let us define $A_0 = (B \ast C)\perp$. Then $(A_0, B)$ is a $t$-ECP for $C$.

In order to compute a $t$-ECP for $C = C_L(\mathcal{X}, \mathcal{P}, E)$, it suffices to compute a code of type $C_L(\mathcal{X}, \mathcal{P}, E - F)$ for some divisor $F$ with $\deg(F) \geq t + g$. 
**Context of the Cryptosystem**

**Public Key:**

\[ K_{\text{pub}} = G \quad \text{and} \quad t^* = \left\lfloor \frac{d^* - g - 1}{2} \right\rfloor \]

where:

- \( G \) is a generator matrix of the **public code**: \( C_{\text{pub}} = C_L(\mathcal{X}, \mathcal{P}, E)^\perp \)

- \( d^* = \deg(E) - 2g + 2 \) is the designed minimum distance of \( C_{\text{pub}} \)

**Future work!!!**

\( t^* \) seems reasonable if \( K_{\text{secret}} \) is based on ECP.

\[ t^* = \left\lfloor \frac{d^* - g - 1}{2} \right\rfloor \leq t = \left\lfloor \frac{d^* - 1}{2} \right\rfloor = \text{actual error-correction capability of } C \]
The $P$-Filtration

→ Let $P = P_1$ be a point of the $n$-tuple $P$.
→ We focus on the sequence of codes:

$$B_i := (C_L(\mathcal{X}, P, E - iP_1))_{i \in \mathbb{N}}$$

Which Elements of the Sequence Do We Know?

1. From a generator matrix of $C_{\text{pub}} = C_L(\mathcal{X}, P, E)^\perp$ one can compute $C_L(\mathcal{X}, P, E)$
   → Computed by Gaussian elimination.

2. $B_0 = C_L(\mathcal{X}, P, E)$.

3. $B_1$ is the set of codewords of the code $B_0$ which are zero at position $P_1$.
   → Computed by Gaussian elimination.

The codes $B_0$ and $B_1$ are known.
**Effective computation - Algorithm I**

**How to compute \( B_2 \)?**

→ If \( \frac{n}{2} > \deg(E) \), then \( B_1^{(2)} \subseteq F_q^n \).

→ If \( \deg(F - P) = \deg(E) - 1 \geq 2g + 1 \), then

\[
B_1^{(2)} = C_L(X, P, E - P_1)^{(2)} = C_L(X, P, 2E - 2P_1)
\]

Thus, \( B_2 \) is the solution space of the following problem

\[
z \in B_1 \quad \text{and} \quad z \ast B_0 \subseteq (B_1)^{(2)}
\]

**Proposition**

Let \( F, G \) be two divisors on \( X \) such that

\[
\deg(F) \geq 2g \quad \text{and} \quad \deg(G) \geq 2g + 1
\]

Then,

\[
C_L(X, P, F) \ast C_L(X, P, G) = C_L(X, P, F + G)
\]
Effective computation - Algorithm I

Theorem I: If we know $B_{s-1}$ and $B_s$ we can compute $B_{s+1}$

$B_{s+1}$ is the solution space of the following problem

$$z \in B_s \quad \text{and} \quad z \ast B_{s-1} \subseteq (B_s)^{(2)}$$

(2)

If $s \geq 1$ and $\frac{n}{2} > \deg(E) \geq 2g + s + 1$.

$(t^* + g)$ repeated applications of Theorem I determines the code $B_{t^*+g}$. 
Effective computation - Algorithm II

We can do better by decreasing the number of iterations and relaxing the parameters conditions ⇒ Algorithm II

→ Algorithm I:

\[ B_0 \supseteq B_1 \supseteq B_2 \supseteq B_3 \supseteq \ldots \supseteq B_{t^* + g - 1} \supseteq B_{t^* + g} \]

Solve \((t^* + g)\) systems of linear equations

→ Algorithm II:

\[ B_0 \supseteq B_1 \supseteq B_2 \supseteq B_4 \supseteq \ldots \supseteq B_{t^* + g} \supseteq B_{t^* + g} \]

Solve \(2 \lceil \log_2 (t^* + g) \rceil + 2\) systems of linear equations
Algorithm I vs. Algorithm II

→ Algorithm I:

\[ \mathcal{B}_0 \supseteq \mathcal{B}_1 \supseteq \mathcal{B}_2 \supseteq \mathcal{B}_3 \supseteq \ldots \supseteq \mathcal{B}_{t^*+g-1} \supseteq \mathcal{B}_{t^*+g} \]

Solve \((t^* + g)\) systems of linear equations

Theorem I: If we know \(\mathcal{B}_{s-1}\) and \(\mathcal{B}_s\) we can compute \(\mathcal{B}_{s+1}\)

\(\mathcal{B}_{s+1}\) is the solution space of the following problem

\[ \mathbf{z} \in \mathcal{B}_s \quad \text{and} \quad \mathbf{z} \star \mathcal{B}_{s-1} \subseteq \left( \mathcal{B}_s \right)^{(2)} \]

If \(s \geq 1\) and \(\frac{n}{2} > \deg(E) \geq 2g + s + 1\).
Algorithm I vs. Algorithm II

→ Algorithm II:

\[ B_0 \supseteq B_1 \supseteq B_2 \supseteq B_4 \supseteq \ldots \supseteq B_{\frac{t^*+g}{2}} \supseteq B_{t^*+g} \]

Solve \( 2 \left\lceil \log_2 (t^* + g) \right\rceil + 2 \) systems of linear equations

Theorem I: If we know \( B_{\left\lfloor \frac{s}{2} \right\rfloor} \) and \( B_{\left\lfloor \frac{s+1}{2} \right\rfloor} \) we can compute \( B_s \)

\( B_s \) is the solution space of the following problem

\[ \mathbf{z} \in B_s \quad \text{and} \quad \mathbf{z} \ast B_0 \subseteq B_{\left\lfloor \frac{s}{2} \right\rfloor} \ast B_{\left\lfloor \frac{s+1}{2} \right\rfloor} \]

If \( s \geq 1 \) and \( \frac{n}{2} > \deg(E) \geq 2g + s + 1. \)
The Attack

**Public Key:** \( \mathcal{K}_{\text{pub}} = C_{\text{pub}} = C_L(X, \mathcal{P}, E)^\perp \) and \( t = \left\lfloor \frac{d^* - g - 1}{2} \right\rfloor \)

**The Algorithm:** Suppose that \( n \geq \deg(E) \).

**Step 1.** Determine the values \( g \) and \( \deg(E) \) using the following Proposition.

**Proposition**

If \( 2g + 1 \leq \deg(E) < \frac{1}{2} n \).

Then, \( \deg(E) = k(C^{(2)}) - k(C) \) and \( g = k(C^{(2)}) - 2k(C) + 1 \)

**Step 2.** Compute the code \( B_{t^*+g} = C_L(X, \mathcal{P}, E - (t^* + g)P_1) \), using one of the algorithms described in §5.1.

**Step 3.** Deduce an ECP from \( B \).

**Corollary:** Let \( B \) of type \( C_L(X, \mathcal{P}, E - F) \) with \( \deg(F) \geq t^* + g \)

Let us define \( A_0 = (B \ast C)^\perp \). Then \((A_0, B)\) is a \( t \)-ECP for \( C = C_L(X, \mathcal{P}, E)^\perp \).
Unfortunately the codes

\[ B_i = C_L(X, P, E - iP_1) \]

are degenerated for \( i > 0 \).
**From degenerate to non degenerate II**

**Aim of this section**

How to computer another code

\[ \hat{B}_i = C_L(\mathcal{X}, \mathcal{P}, E - F') \]

with:

1. \( F' = F + (h) \) for some \( h \in \mathbb{F}_q(\mathcal{X}) \)
2. \( \text{supp}(F') \cap \text{supp}(D_P) = \emptyset \)

**Remark:** We do not need to compute \( h \) but just **prove its existence**.

\( \Rightarrow \) Them following result allows to compute a generator matrix of

\[ \hat{B}_{t^*+g} \] from the codes \( B_{t^*+g} \) and \( B_{t^*+g+1} \).
Let $G$ be a generator matrix of $\mathcal{B}_{t^*+g}$ of the form:

$$G = \begin{pmatrix} 0 & \mathbf{c}_1 \\ 0 & \mathbf{G}_1 \end{pmatrix},$$

where

$$\begin{cases} (0 | \mathbf{c}_1) \in \mathcal{B}_{t^*+g} \setminus \mathcal{B}_{t^*+g+1} \\ (0 | \mathbf{G}_1) = \text{gen. matrix of } \mathcal{B}_{t^*+g+1} \end{cases}$$

Then the following matrix is a generator matrix for $\hat{\mathcal{B}}_{t^*+g}$

$$\hat{G} = \begin{pmatrix} 1 & \mathbf{c}_1 \\ 0 & \mathbf{G}_1 \end{pmatrix}$$
The **costly part** of the attack is the computation of the code $B$.

We can apply one of the algorithms of §5.1

Computing:

1. a generator matrix of $C^{(2)}$
2. and then apply Gaussian elimination to such matrix

Costs:

\[ O \left( \binom{k}{2} n + \binom{k}{2} n^2 \right) \sim O \left( k^2 n^2 \right) \text{ operations in } \mathbb{F}_q. \]

Roughly speaking the cost of our attack is about $O \left( (\lambda + 1) n^4 \right)$ where:

1. $\lambda =$ Linear systems to solve depending on the chosen algorithm from §5.1
2. The term $(\lambda + 1)$ is the cost of computing a non-degenerated code.
→ We summarize in the following tables the average running times of our algorithm for several codes.

→ The attack has been implemented with MAGMA.

→ The work factor $w$ of and ISD attack is given. These work factors have been computed thanks to Christiane Peter’s Software

Remark: **ISD’s** average complexity is

$$O \left( k^2 n \frac{n}{(n-k)} \right) \text{ operations in } \mathbb{F}_q$$
**Example I: Hermitian curves**

**Hermitian Curve**

The **Hermitian curve** $\mathcal{H}_r$ over $\mathbb{F}_q$ with $q = r^2$ is defined by the affine equation

$$Y^r + Y = X^{r+1}$$

→ This curve has $P_\infty = (0 : 1 : 0)$ as the only point at infinity.

Take:
- $E = mP_\infty$
- $\mathcal{P}$ be the $n = q\sqrt{q} = r^3$ affine $\mathbb{F}_q$-rational points of the curve.

The following table considers different codes of type $C_L(\mathcal{H}_r, \mathcal{P}, E)^\perp$ with $n > \deg(E) > 2g - 2$.

<table>
<thead>
<tr>
<th>$q$</th>
<th>$g$</th>
<th>$n$</th>
<th>$k$</th>
<th>$t$</th>
<th>$w$</th>
<th>key size</th>
<th>time</th>
</tr>
</thead>
<tbody>
<tr>
<td>$7^2$</td>
<td>21</td>
<td>343</td>
<td>193</td>
<td>54</td>
<td>$2^{84}$</td>
<td>163 ko</td>
<td>74 s</td>
</tr>
<tr>
<td>$9^2$</td>
<td>36</td>
<td>729</td>
<td>404</td>
<td>126</td>
<td>$2^{182}$</td>
<td>833 ko</td>
<td>21 min</td>
</tr>
<tr>
<td>$11^2$</td>
<td>55</td>
<td>1331</td>
<td>885</td>
<td>168</td>
<td>$2^{311}$</td>
<td>2730 ko</td>
<td>67 min</td>
</tr>
</tbody>
</table>

**Table:** Comparison with Hermitian codes

$w$ computed with Christiane Peters software
Example II: Suzuki curves

Suzuki curves are curves $\mathcal{X}$ defined over $\mathbb{F}_q$ by the following equation

$$Y^q - Y = X^{q_0}(X^q - X) \quad \text{with} \quad q = 2q_0^2 \geq 8 \quad \text{and} \quad q_0 = 2^r$$

This curve has exactly:

- $q^2 + 1$ rational places
- A single place at infinity $P_\infty$.

Take:

- $E = mP_\infty$
- $\mathcal{P}$ be the $q^2$ rational points of the curve.

The following table considers several codes of type $C_L(\mathcal{X}, \mathcal{P}, E)^\perp$ with $n > \deg(E) > 2g - 2$.

<table>
<thead>
<tr>
<th>$q$</th>
<th>$g$</th>
<th>$n$</th>
<th>$k$</th>
<th>$t$</th>
<th>$w$</th>
<th>key size</th>
<th>time</th>
</tr>
</thead>
<tbody>
<tr>
<td>$2^5$</td>
<td>$124$</td>
<td>$1024$</td>
<td>$647$</td>
<td>$64$</td>
<td>$2^{110}$</td>
<td>$1220$ ko</td>
<td>$30$ min</td>
</tr>
</tbody>
</table>

Table: Comparison with Suzuki codes

$w$ computed with Christiane Peters software
We constructed a **polynomial-time** algorithm which breaks the McEliece scheme based on AG codes whenever

\[ 2 < t \leq \left\lfloor \frac{d^* - g - 1}{2} \right\rfloor \]

**Complexity:** \( O(n^4) \)

**Future work:** using the concept of Error-Correcting Arrays (ECA) or well-behaving sequence obtain an attack for

\[ t = \left\lfloor \frac{d^* - 1}{2} \right\rfloor \]
Thank you for your attention!