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

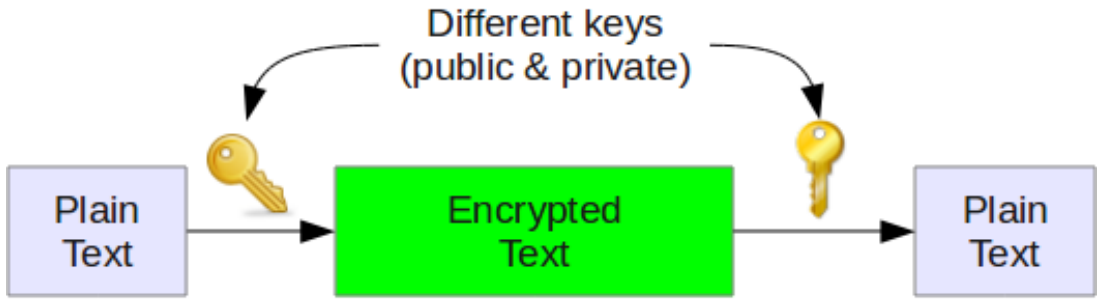
A. Couvreur[1] I. Márquez-Corbella [1] R. Pellikaan [2]

[1] INRIA Saclay & LIX

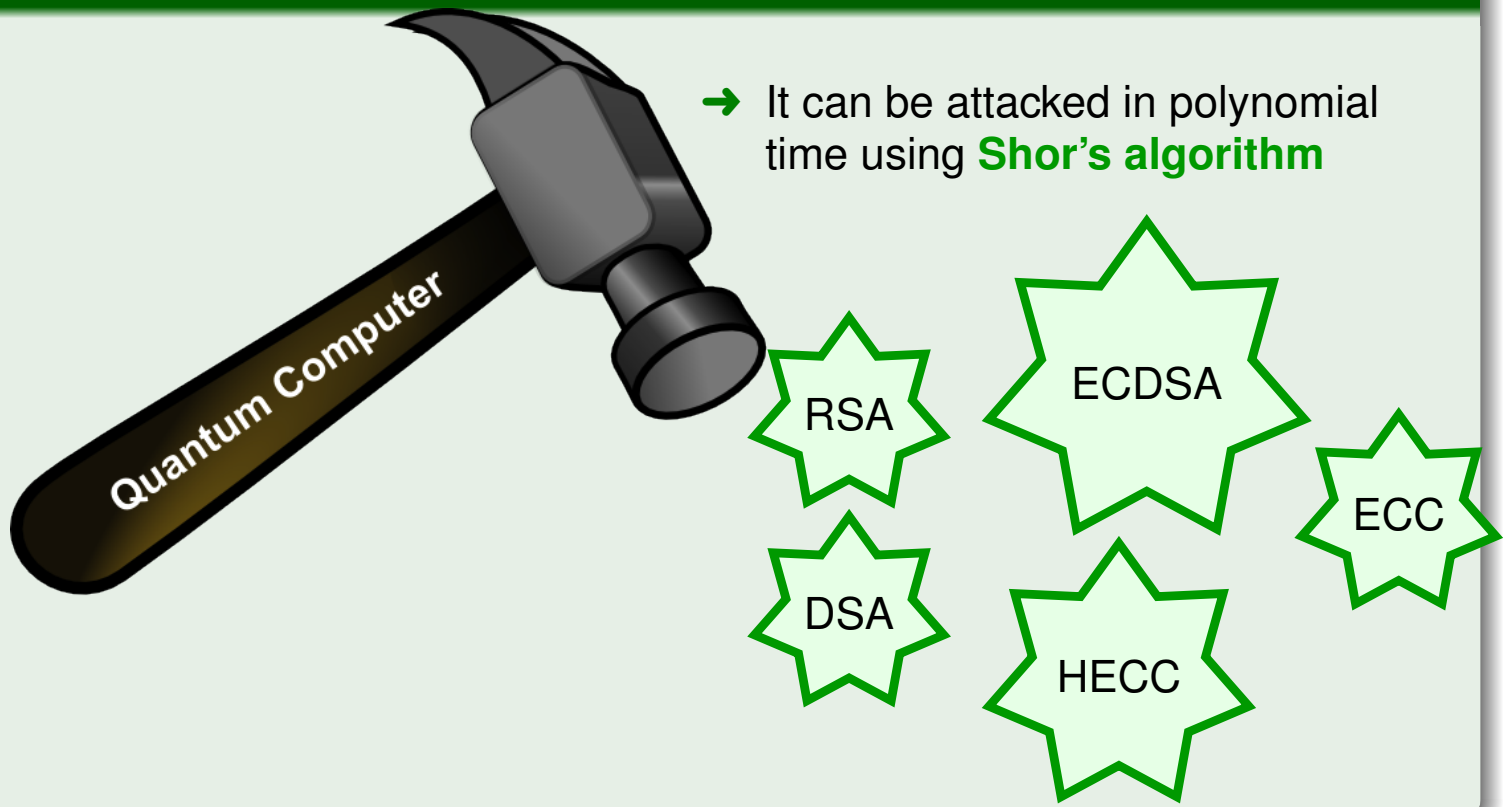
[2] Department of Mathematics and Computing Science, TU/e.

May 23, 2014

Public-Key Cryptosystems



Most PKC are based on Number-theoretic problems



McEliece cryptosystem

➜ McEliece introduced the first PKC based on Error-Correcting Codes in 1978.

Advantages:

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

Drawback:

➢ Large key size.

R. J. McEliece.
A public-key cryptosystem based on algebraic coding theory.
DSN Progress Report, 42-44:114-116, 1978.

McEliece Cryptosystem

➜ $t \in \mathbb{N}^* \implies$ **Error-correcting capacity** of $\mathcal{C}$

Consider any triplet:

$$\left(\mathcal{C}, \mathcal{A}_{\mathcal{C}}, t \right)$$

➜ $[n, k]_q$ **linear code** with an efficient decoding algorithm

➠ Let G be a non structured generator matrix of $\mathcal{C}$.

➜ ***"Efficient"* decoding algorithm** for $\mathcal{C}$ which corrects up to t errors.

McEliece Cryptosystem

Key Generation

Given:

1. **McEliece Public Key:** $\mathcal{K}_{pub} = (G, t)$
2. **McEliece Private Key:** $\mathcal{K}_{secret} = (\mathcal{A}_C)$

Encryption

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

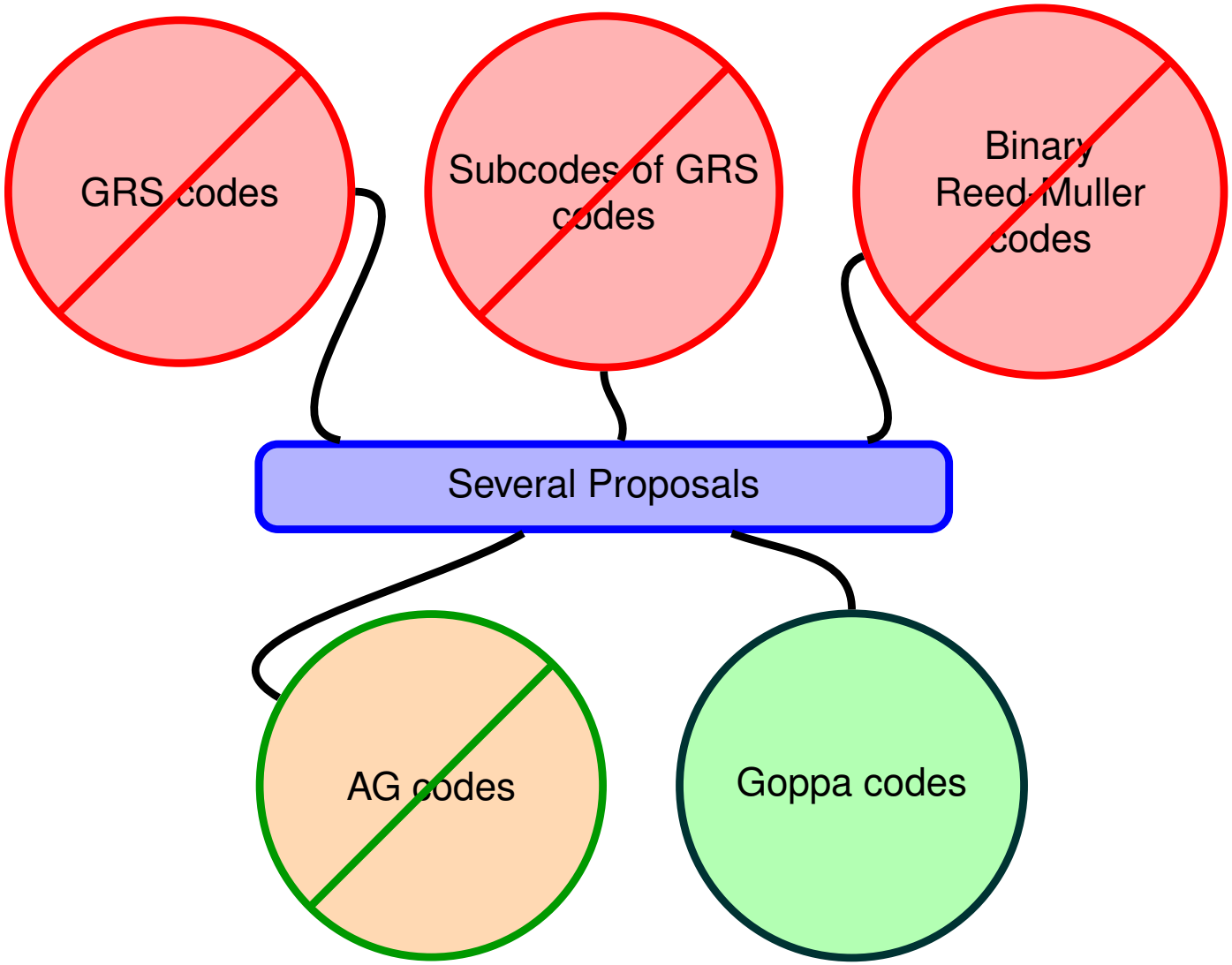
$$\mathbf{y} = \mathbf{m}G + \mathbf{e}$$

where $\mathbf{e}$ is a random error vector of weight at most t.

Decryption

Using $\mathcal{K}_{secret}$, the receiver obtain $\mathbf{m}$.

Proposals



GRS 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.

Parameters	Key size	Security level
$[256, 128, 129]_{256}$	67 ko	2^{95}

Subcodes of GRS codes I

- **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:** (2006, 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 $[n, k]$ is itself a GRS code of dimension $2k - 1$ then we can apply Sidelnikov-Shestakov attack.
  - **M-Mártinez-Pellikaan:** (2012) Give a characterization of the possible parameters that should be used to avoid attacks on the Berger-Loidreau cryptosystem.

Subcodes of GRS codes II

- **Wieschebrick** (2006) 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.

Parameters	Key size	Security level
$[1024, 176, 128]_2$	22.5 ko	2^{72}
$[2048, 232, 256]_2$	59, 4 ko	2^{93}

AG codes

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

- This system was broken for:

  1. Codes on curves of genus $g = 0$ by the **Sidelnikov-Shestakov** attack in 1992

     GRS codes are Algebraic Geometry codes on the projective line.

  2. Codes on curves of genus $g \leq 2$ by **Faure and Minder** in 2008.
  3. We can retrieve the structure of the code (in polynomial time) by **M-Martínez-Pellikaan-Ruano** in 2013

Parameters	Key size	Security level
$[171, 109, 61]_{128}$	16 ko	2^{66}

Compact variants

1. In 2005 **Gaborit** propose to use BCH codes.
   Size key: $\sim$ 1.5 ko, **Security level:** 2^{80}.
2. In 2009 **Berger, Cayrel, Gaborit and Otmani** propose to use alternant quasi-cyclic codes.
   Size key: $\sim$ 750 o, **Security level:** 2^{80}.
3. In 2009 **Misoczki and Baretto** propose to use alternant quasi-dyadic codes.
   Size key: $\sim$ 2.5 ko, **Security level:** 2^{80}.

✖ **Attacks:**

- ✖ **Otmani, Tillich and Dallot** in 2008.
- ✖ **Faugère, Otmani, Perret, Tillich** in 2010.
- ✖ **F. de Portzamparck, Faugère, Otmani, Perret, Tillich** in 2014.

Goppa codes

- 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!!

Parameters	Key size	Security level
$[1024, 524, 101]_2$	67 ko	2^{62}
$[2048, 1608, 48]_2$	412 ko	2^{96}

Notation

- For all $\mathbf{a}, \mathbf{b} \in \mathbb{F}_q^n$ we define:
  - **Star Multiplication:** $\mathbf{a} * \mathbf{b} = (a_1 b_1, \ldots, a_n b_n) \in \mathbb{F}_q^n$
  - **Standard Inner Multiplication:** $\langle \mathbf{a}, \mathbf{b} \rangle = \sum_{i=1}^{n} a_i b_i \in \mathbb{F}_q$

- For all subsets $A,\ B \subseteq \mathbb{F}_q^n$ we define:
  - $A * B = \langle \{\mathbf{a} * \mathbf{b} \mid \mathbf{a} \in A \text{ and } \mathbf{b} \in B\} \rangle$
    For $B = A \Longrightarrow A * A$ is denoted as $A^{(2)}$
  - $A \perp B \iff \langle \mathbf{a}, \mathbf{b} \rangle = 0 \quad \forall\, \mathbf{a} \in A \text{ and } \mathbf{b} \in B$

Decoding by Error Correcting Pairs

Let $\mathcal{C}$ be a linear code. We denote by:

- $k(\mathcal{C})$ = dimension of $\mathcal{C}$
- $d(\mathcal{C})$ = minimum distance of $\mathcal{C}$

Error-correcting pairs (ECP)

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

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

An $[n, k]$ code which has a t-ECP over $\mathbb{F}_{q^m}$ has a decoding algorithm with complexity $\mathcal{O}\left((nm)^3\right)$.

Generalized Reed-Solomon Codes I

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 $\mathbb{F}_q$.
- $k \in \mathbb{Z} \ : \ k < n$

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

$$\mathrm{GRS}_k(\mathbf{a}, \mathbf{b}) = \{\mathbf{b} * f(\mathbf{a}) = (b_1 f(a_1), \ldots, b_n f(a_n)) \mid f \in \mathbb{F}_q[X]_{<k}\}$$

Generalized Reed-Solomon Codes II

Parameters of $\mathrm{GRS}_k(\mathbf{a}, \mathbf{b})$

The $\mathrm{GRS}_k(\mathbf{a}, \mathbf{b})$ is an **MDS** code with parameters $[n, k, n-k+1]$.

➜ A generator matrix of $\mathrm{GRS}_k(\mathbf{a}, \mathbf{b})$ is given by

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

Dual of a GRS code

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

$$\mathrm{GRS}_k(\mathbf{a}, \mathbf{b})^{\perp} = \mathrm{GRS}_{n-k}(\mathbf{a}, \mathbf{c}) \text{ for some } \mathbf{c} \text{ explicitly known}$$

➜ The $\mathrm{GRS}_k(\mathbf{a}, \mathbf{b})^{\perp}$ is an MDS code with parameters $[n, n-k, k+1]$.

t-ECP for GRS I

Note that: $\mathrm{GRS}_k(\mathbf{a}, \mathbf{b}) * \mathrm{GRS}_l(\mathbf{a}, \mathbf{c}) = \mathrm{GRS}_{k+l-1}(\mathbf{a}, \mathbf{b} * \mathbf{c})$

Let

$$A = \mathrm{GRS}_{t+1}(\mathbf{a}, \mathbf{b}_1), \quad B = \mathrm{GRS}_t(\mathbf{a}, \mathbf{b}_2) \text{ and}$$
$$\mathcal{C} = \mathrm{GRS}_{2t}(\mathbf{a}, \mathbf{b}_1 * \mathbf{b}_2)^{\perp}$$

then (A, B) is a t-ECP for $\mathcal{C}$.

E.1 $A * B = \mathrm{GRS}_{2t}(\mathbf{a}, \mathbf{b}_1 * \mathbf{b}_2) = \mathcal{C}^{\perp} \Rightarrow (A * B) \perp \mathcal{C}$
E.2 $k(A) > t$
E.3 $B^{\perp} = \mathrm{GRS}_{n-t}(\mathbf{a}, \mathbf{c}_2) \Rightarrow d(B^{\perp}) = t + 1 > t$
E.4 $d(A) + d(\mathcal{C}) = (n - t) + (2t + 1) > n$

t-ECP for GRS II

Conversely, let $\mathcal{C} = \mathrm{GRS}_{n-2t}(\mathbf{a}, \mathbf{b})$ then

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

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

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

Moreover an $[n, n-2t, 2t+1]$ code that has a t-ECP is a GRS code.

Algebraic Geometry codes

➜ An AG code is defined by a triplet

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

➜ $\mathcal{X}$ 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 $D_{\mathcal{P}} = P_1 + \cdots + P_n$

Algebraic Geometry codes

➜ An AG code is defined by a triplet

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

➜ $\mathcal{X}$ 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 $D_{\mathcal{P}} = P_1 + \cdots + P_n$

Algebraic Geometry codes I

- An AG code is defined by a triplet

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

- E is an $\mathbb{F}_q$-divisor of $\mathcal{X}$ such that

$$\mathrm{supp}(E) \cap \mathrm{supp}(D_{\mathcal{P}}) = \emptyset$$

DIVISORS ON CURVES

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

$$D = \sum_{P \in \mathcal{X}(\mathbb{F}_q)} n_P P \text{ with } n_P \in \mathbb{Z} \text{ and } P \in \mathcal{X}(\mathbb{F}_q)$$

- If $n_P \geq 0$ for all $P \in \mathcal{X}(\mathbb{F}_q)$ then D is an **Effective Divisor**.
- **Support of the divisor D:** $\operatorname{supp}(D) = \{P \mid n_P \neq 0\}$
- **Degree of the divisor D:** $\deg(D) = \sum_{P \in \mathcal{X}(\mathbb{F}_q)} n_p \deg(P)$

Divisor of rational functions

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

$$(f) = \underbrace{\sum_{P \text{ zero of } f} \mathrm{v}_P(f)P}_{(f)_0} - \underbrace{\sum_{P \text{ pole of } f} \mathrm{v}_P(f)P}_{(f)_\infty}$$

Algebraic Geometry codes IV

Space of Rational Functions associated to the divisor E

$$L(E) = \{f \in \mathbb{F}_q(\mathcal{X}) \mid f = 0 \text{ or } (f) + E \geq 0\}$$

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

Riemman-Roch Theorem

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

Furthermore, if $\deg(E) > 2g - 2$ then $\dim(L(E)) = \deg(E) + 1 - g$

Algebraic Geometry codes V

➜ Let us consider the triplet:

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

➜ $\mathcal{X}$ 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 $\operatorname{supp}(E) \cap \operatorname{supp}(D_{\mathcal{P}}) = \emptyset$

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

$$\begin{array}{rccc} \mathrm{ev}_{\mathcal{P}} : & L(E) & \longrightarrow & \mathbb{F}_q^n \\ & f & \longmapsto & \mathrm{ev}_P(f) = (f(P_1), \ldots, f(P_n)) \end{array}$$

Algebraic Geometry Code (AG code)

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

$$\mathcal{C}_L(\mathcal{X}, \mathcal{P}, E) = \{\mathrm{ev}_{\mathcal{P}}(f) = (f(P_1), \ldots, f(P_n)) \mid f \in L(E)\}$$

Algebraic Geometry codes VI

➜ 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 $\mathcal{C}_L(\mathcal{X}, \mathcal{P}, E)$

Theorem I [Parameters of an AG code]

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

$$k(\mathcal{C}) \geq m + 1 - g \quad \text{and} \quad d(\mathcal{C}) \geq n - m$$

Moreover, if $n > m > 2g - 2$ then $k(\mathcal{C}) = m - g + 1$.

Dual of an AG code

Let:

- ω 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 ω.

Then

$$\mathcal{C}_L(\mathcal{X}, \mathcal{P}, E)^\perp = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, E^\perp)$$

with $E^\perp = D_\mathcal{P} - E + K$ and $\deg(E^\perp) = n - m + 2g - 2$

Theorem II [Parameters of the dual of an AG code]

Let $\mathcal{C} = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, E)$. If $\deg(E) = m > 2g - 2$ then

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

Moreover, if $n > m > 2g - 2$ then $k(\mathcal{C}^\perp) = n - m - 1 + g$

t-ECP for AG codes I

➜ Consider the AG code

$$\mathcal{C} = \mathcal{C}_L\left(\mathcal{X}, \mathcal{P}, E\right)^{\perp}$$

Theorem [Pellikaan - 1992]

The pair of codes (A, B) defined by

$$A = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, F) \quad \text{and} \quad B = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, E - F)$$

with $m > \deg(F) \geq t + g$ is a t-ECP for $\mathcal{C}$.

➪ Such a pair **always exists** whenever

$$m > 2g - 2 \quad \text{and } t = \left\lfloor \frac{d^* - 1 - g}{2} \right\rfloor.$$

t-ECP for AG codes II

Corollary [Main Corollary]

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

And let us define $A_0 = (B * \mathcal{C})^{\perp}$. Then (A_0, B) is a t-ECP for $\mathcal{C}$

In order to compute a t-ECP for $\mathcal{C} = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, E)$, it suffices to compute a code of type

$\mathcal{C}_L(\mathcal{X}, \mathcal{P}, E - F)$ for some divisor F with $\deg(F) \geq t + g$

Context of the Cryptosystem

Public Key:

$$\mathcal{K}_{\text{pub}} = G \quad \text{and} \quad t = \left\lfloor \frac{d^* - g - 1}{2} \right\rfloor$$

where:

- G is a generator matrix of the **public code**:

$$\mathcal{C}_{\text{pub}} = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, E)^{\perp}$$

- $d^* = m - 2g + 2$ is the designed minimum distance of $\mathcal{C}_{\text{pub}}$

➜ Our t seems reasonable if $\mathcal{K}_{\text{secret}}$ is based on ECP.

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

➜ **Future work!!!**

THE P-FILTRATION

CONSTRUCT $\mathcal{C}_L(\mathcal{X}, \mathcal{P}, E - F)$ WITH $\deg(F) \geq t + g$ FROM $\mathcal{C} = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, E)^\perp$

- Let P be a point of the n-tuple $\mathcal{P}$.
- We focus on the sequence of codes:

$$\mathcal{B}_i := (\mathcal{C}_L(\mathcal{X}, \mathcal{P}, E - iP))_{i \in \mathbb{N}}$$

WHICH ELEMENTS OF THE SEQUENCE DO WE KNOW?

1. From a generator matrix of $\mathcal{C}_{\text{pub}} = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, E)^\perp$ one can compute $\mathcal{C}_L(\mathcal{X}, \mathcal{P}, E)$
   - Computed by **Gaussian elimination**.
2. $\mathcal{B}_0 = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, E)$.
3. $\mathcal{B}_1$ is the set of codewords of the code $\mathcal{C}_L(\mathcal{X}, \mathcal{P}, E)$ which are zero at position P.
   - Computed by **Gaussian elimination**.

The codes $\mathcal{B}_0$ and $\mathcal{B}_1$ are **known**.

Effective computation - Algorithm I

How to compute $\mathcal{B}_2$?

- If $\frac{n}{2} > m$, then $\mathcal{B}_1^{(2)} \subsetneq \mathbb{F}_q^n$.
- If $\deg(F-P) = m-1 \geq 2g+1$, then

$$\mathcal{B}_1^{(2)} = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, E-P)^{(2)} = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, 2E-2P)$$

Thus, $\mathcal{B}_2$ is the solution space of the following problem

$$\mathbf{z} \in \mathcal{B}_1 \quad \text{and} \quad \mathbf{z} * \mathcal{B}_0 \subseteq (\mathcal{B}_1)^{(2)}. \tag{1}$$

Proposition

Let F, G be two divisors on $\mathcal{X}$ such that

$$\deg(F) \geq 2g \text{ and } \deg(G) \geq 2g+1$$

Then,

$$\mathcal{C}_L(\mathcal{X}, \mathcal{P}, F) * \mathcal{C}_L(\mathcal{X}, \mathcal{P}, G) = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, F+G)$$

EFFECTIVE COMPUTATION - ALGORITHM I

CONSTRUCT $\mathcal{C}_L(\mathcal{X}, \mathcal{P}, E - F)$ WITH $\deg(F) \geq t + g$ FROM $\mathcal{C} = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, E)^{\perp}$

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

$$\mathbf{z} \in \mathcal{B}_s \quad \text{and} \quad \mathbf{z} * \mathcal{B}_{s-1} \subseteq (\mathcal{B}_s)^{(2)}. \tag{2}$$

If $s \geq 1$ and $\frac{n}{2} > m \geq 2g + s + 1$.

$(t + g)$ repeated applications of **Theorem I** determines the code $\mathcal{B}_{t+g}$.

Algorithm 5.1: Let $\frac{n}{2} > m \geq 3g + t + 1$

Data: Generator matrices for the codes $\mathcal{B}_0$ and $\mathcal{B}_1$
Result: A generator matrix for the code $\mathcal{B}_{t+g}$
for $s = 2, \ldots, t + g$ **do**
  Compute $\mathcal{B}_s$ from the codes $\mathcal{B}_{s-1}$ and $\mathcal{B}_{s-2}$ using **Theorem I**.

Algorithm complexity: We solve $(t + g)$ systems of linear equations.

Effective computation - Algorithm II

We can do **better** by **decreasing** the number of iterations and **relaxing** the parameters conditions $\Rightarrow$ 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

➜ **Algorithm II:**

$$\mathcal{B}_0 \supseteq \mathcal{B}_1 \supseteq \mathcal{B}_2 \supseteq \mathcal{B}_4 \supseteq \ldots \supseteq \mathcal{B}_{\frac{t+g}{2}} \supseteq \mathcal{B}_{t+g}$$

Solve $2\lceil \log_2(t + g) \rceil + 2$ systems of linear equations

Algorithm I vs. Algorithm II

Construct $\mathcal{C}_L(\mathcal{X}, \mathcal{P}, E - F)$ with $\deg(F) \geq t + g$ from $\mathcal{C} = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, E)^{\perp}$

➜ **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

$$\mathbf{z} \in \mathcal{B}_s \quad \text{and} \quad \mathbf{z} * \mathcal{B}_{s-1} \subseteq (\mathcal{B}_s)^{(2)}.$$

If $s \geq 1$ and $\frac{n}{2} > m \geq 2g + s + 1$.

ALGORITHM I VS. ALGORITHM II

CONSTRUCT $\mathcal{C}_L(\mathcal{X}, \mathcal{P}, E - F)$ WITH $\deg(F) \geq t + g$ FROM $\mathcal{C} = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, E)^{\perp}$

➜ **Algorithm II:**

$$\mathcal{B}_0 \supseteq \mathcal{B}_1 \supseteq \mathcal{B}_2 \supseteq \mathcal{B}_4 \supseteq \ldots \supseteq \mathcal{B}_{\frac{t+g}{2}} \supseteq \mathcal{B}_{t+g}$$

Solve $2\lceil \log_2(t+g) \rceil + 2$ systems of linear equations

THEOREM I: IF WE KNOW $\mathcal{B}_{\lfloor \frac{s}{2} \rfloor}$ AND $\mathcal{B}_{\lfloor \frac{s+1}{2} \rfloor}$ WE CAN COMPUTE $\mathcal{B}_s$

$\mathcal{B}_s$ is the solution space of the following problem

$$\mathbf{z} \in \mathcal{B}_s \quad \text{and} \quad \mathbf{z} * \mathcal{B}_0 \subseteq \mathcal{B}_{\lfloor \frac{s}{2} \rfloor} * \mathcal{B}_{\lfloor \frac{s+1}{2} \rfloor}.$$

If $s \geq 1$ and $\frac{n}{2} > m \geq 2g + s + 1$.

The Attack

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

The Algorithm: Suppose that $\frac{n}{2} \geq m$.

Step 1. Determine the values g and m using the following Proposition.

Proposition

If $2g + 1 \leq m < \frac{1}{2}n$. Let $k_1 = k(\mathcal{C})$ and $k_2 = k(\mathcal{C}^{(2)})$

Then, $m = k_2 - k_1$ and $g = k_2 - 2k_1 + 1$

Step 2. Compute the code $\mathcal{B}_{t+g} = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, E - (t+g)P_1)$, using one of the algorithms described in §5.1

Step 3. Deduce an ECP from B.

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

Let us define $A_0 = (B * \mathcal{C})^{\perp}$. Then (A_0, B) is a t-ECP for $\mathcal{C} = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, E)^{\perp}$.

From degenerate to non degenerate I

Unfortunately the codes

$$\mathcal{B}_i = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, E - iP_1)$$

are **degenerated** for $i > 0$.

From degenerate to non degenerate II

Aim of this section

How to computer another code

$$\hat{\mathcal{B}}_i = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, E - F')$$

with:

1. $F' = F + (h)$ for some $h \in \mathbb{F}_q(\mathcal{X})$
2. $\mathrm{supp}(F') \cap \mathrm{supp}(D_{\mathcal{P}}) = \emptyset$

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

➜ Them following result allows to compute a generator matrix of

$$\hat{\mathcal{B}}_{t+g} \text{ from the codes } \mathcal{B}_{t+g} \text{ and } \mathcal{B}_{t+g+1}.$$

From degenerate to non degenerate III

Theorem

Let G be a generator matrix of $\mathcal{B}_{t+g}$ of the form:

$$G = \left(\begin{array}{c|c} 0 & \mathbf{c}_1 \\ \hline \mathbf{0} & G_1 \end{array}\right), \text{ where}$$

$$\begin{cases} \left(\begin{array}{c|c} 0 & \mathbf{c}_1 \end{array}\right) \in \mathcal{B}_{t+g} \setminus \mathcal{B}_{t+g-1} \\ \left(\begin{array}{c|c} \mathbf{0} & G_1 \end{array}\right) = \text{ gen. matrix of } \mathcal{B}_{t+g+1} \end{cases}$$

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

$$\hat{G} = \left(\begin{array}{c|c} 1 & \mathbf{c}_1 \\ \hline \mathbf{0} & G_1 \end{array}\right)$$

Complexity

- The **costly part** of the attack is the computation of the code B

  $\Longrightarrow$ We can apply one of the algorithms of §5.1

  Computing:

  1. a generator matrix of $\mathcal{C}^{(2)}$
  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.

Examples

- 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

$$O\left(k^2 n \frac{\binom{n}{t}}{\binom{n-k}{t}}\right) \text{ operations in } \mathbb{F}_q$$

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

$$\mathcal{C}_L(\mathcal{H}_r, \mathcal{P}, E)^\perp \text{ with } n > m > 2g - 2.$$

q	g	n	k	t	**w**	key size	time
7^2	21	343	193	54	2^{84}	163 ko	74 s
9^2	36	729	404	126	2^{182}	833 ko	21 min
11^2	55	1331	885	168	2^{311}	2730 ko	67 min

Table: Comparison with Hermitian codes

Example II: Suzuki curves

Suzuki curves

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

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

This curve has exactly:

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

Take:

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

The following table considers several codes of type

$$\mathcal{C}_L(\mathcal{X}, \mathcal{P}, E)^\perp \text{ with } n > m > 2g - 2.$$

q	g	n	k	t	**w**	key size	time
2^5	124	1024	647	64	2^{110}	1220 ko	30 min

Table: Comparison with Suzuki codes

Conclusions

- 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:** $O\left(n^4\right)$

- **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!