ERROR CORRECTING PAIR: A NEW APPROACH TO CODE-BASED CRYPTOGRAPHY

I. MÁRQUEZ-CORBELLA [1] R. PELLIKAAN [2]

[1] INRIA Saclay & LIX

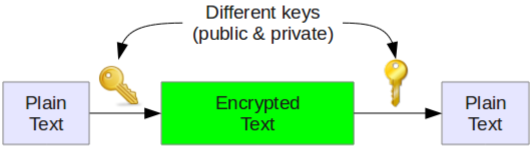
[2] Department of Mathematics and Computing Science, TU/e.

20^{th} Conference on **A**pplications of **C**omputer **A**lgebra (**ACA 2014**)

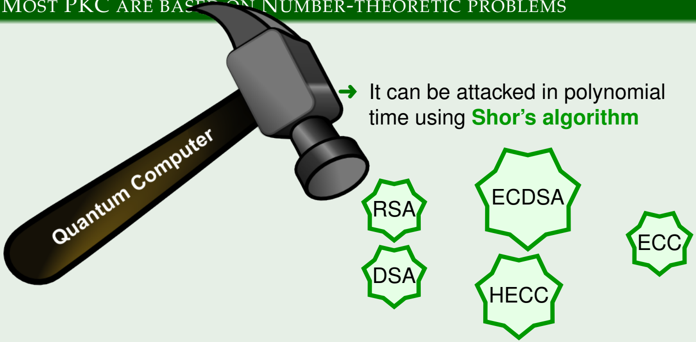
CACTC Session - (COMPUTER ALGEBRA IN CODING THEORY AND CRYPTOGRAPHY)

July 9-12, 2014

Public-Key Cryptosystems



Most PKC are based on Number-theoretic problems



➜ It can be attacked in polynomial time using **Shor's algorithm**

McEliece cryptosystem

➜ 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.
DSN Progress Report, 42-44:114-116, 1978.

McEliece Cryptosystem

➜ $t \in \mathbb{N}^*$ $\Longrightarrow$ **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:** $\mathcal{K}_{secret} = (\mathcal{A}_{\mathcal{C}})$

Encryption

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

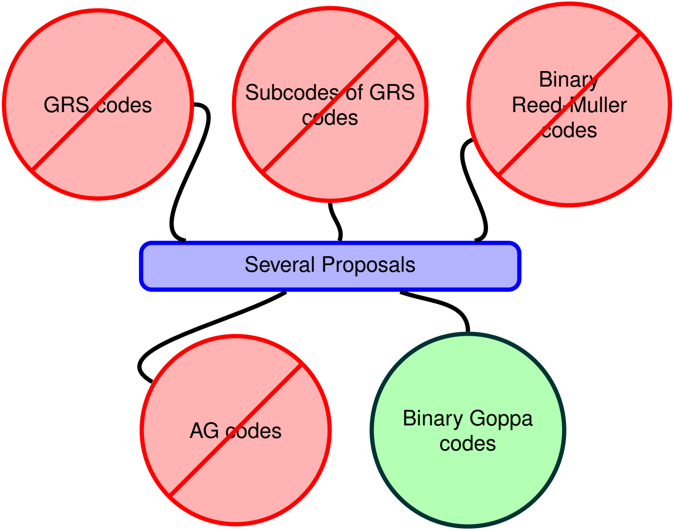
$$\mathbf{y} = \mathbf{m}G + \mathbf{e}$$

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

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:** (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-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** (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.

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. Genus $g = 0$: by the **Sidelnikov-Shestakov** attack in 1992

   GRS codes are Algebraic Geometry codes on the projective line.

2. Genus $g = 1$: by **Minder-Shokrollahi** in 2007.
3. Genus $g \leq 2$: by **Faure-Minder** in 2008.
4. Arbitrary genus by **Couvreur-M-Pellikaan** in 2014.
   We can retrieve the **model of the curve** (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}.

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

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

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

Decoding by Error-Correcting Pairs (ECP) I

Let:

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

Compute:

$$K_{\mathbf{y}} = \{\mathbf{a} \in A \mid \langle \mathbf{y}, \mathbf{a} * \mathbf{b} \rangle = 0, \text{ for all } \mathbf{b} \in B\}$$

Remark: Condition 1

$$\text{If } A * B \subseteq \mathcal{C}^{\perp} \implies K_{\mathbf{y}} = K_{\mathbf{e}}$$

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

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

Lemma 1: Condition 3

Let $I = \mathrm{supp}(\mathbf{e})$ and $A * B \subseteq \mathcal{C}^{\perp}$. If $d(B^{\perp}) > t \implies A(I) = K_{\mathbf{y}}$

Decoding by Error-Correcting Pairs (ECP) II

Lemma 2: Condition 2

If $I = \text{supp}(e)$ and $k(A) > t \implies \exists \mathbf{a} \in K_{\mathbf{y}} \setminus \{\mathbf{0}\}$

Lemma 3: Condition 4

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

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

$$H\mathbf{x}^T = H\mathbf{y}^T \text{ such that } x_j \neq 0 \text{ for all } j \in J$$

Decoding by Error-Correcting Pairs (ECP) III

1. Compute:

$$K_{\mathbf{y}} = \{\mathbf{a} \in A \mid \langle \mathbf{y}, \mathbf{a} * \mathbf{b} \rangle = 0, \text{ for all } \mathbf{b} \in B\}$$

> Find the zero space of a set of linear equations over $\mathbb{F}_q$

2. If $K_{\mathbf{y}} = \mathbf{0} \Longrightarrow$ **The received word has more than t errors**

➜ Else take a nonzero $\mathbf{a} \in K_{\mathbf{y}} = A(I)$ and define $J = \{j \mid a_j = 0\}$

3. Find $\mathbf{e} \in \mathbb{F}_q^n$ by solving the following linear equation (which has a **unique** solution):

$$H\mathbf{x}^T = H\mathbf{y}^T \quad \text{such that} \quad x_j \neq 0 \text{ for } j \in J$$

> Solve linear equations over $\mathbb{F}_q$

Complexity: $\sim \mathcal{O}\left(n^{\mathbf{w}}\right)$

Motivation

"At the heart of any public-key cryptosystem is a one-way function - a function $y = f(x)$ that is easy to evaluate but for which is computationally infeasible (one hopes) to find the inverse $x = f^{-1}(y)$".

N. Koblitz, A. Menezes.
The brave new world of bodacious assumptions in cryptography.
Notices Amer. Math. Soc. 57(3), 357-365 (2010).

Let $\mathcal{C}_t$ the class of linear codes over $\mathbb{F}_q$ that have a t-ECP over an extension of $\mathbb{F}_q$.

- This family have an **efficient decoding algorithm** $\Rightarrow$ they are appropriate for code-based cryptography.
- Most families of codes used in code-based cryptography belongs to $\mathcal{C}_t$.

  (Like GRS codes, Goppa codes, AG codes ...)
- We proposed to use the subclass of $\mathcal{C}_t$ formed by **those linear codes $\mathcal{C}$ whose error correcting pair is not easily reconstructed from** $\mathcal{C}$, i.e. we consider the following one way function:

$$\mathbf{x} = (A, B) \quad \longmapsto \quad \mathbf{y} = A * B,$$

  where (A, B) is a t-ECP.

Examples of the Existence of ECP

- t-ECP for **G**eneralized **R**eed-**S**olomon (**GRS**) codes
- t-ECP for **A**lgebraic-**G**eometric (**AG**) codes.
- t-ECP for **Alternant** codes.
- t-ECP for **Goppa** codes.
- t-ECP for **cyclic** codes.

Question:

- If a code has a ***t*-ECP** how difficult / easy is to **retrieve** such a pair?

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

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 $[n, k, n-k+1]_q$.

➜ 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:

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

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

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

Subcodes of GRS codes

Let: $\mathcal{C} \subseteq \mathcal{D}$

- ➜ $\mathcal{D}$ be a code that has (A, B) as t-ECP
- ➜ $\mathcal{C}$ be a subcode of $\mathcal{D}$

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

Algebraic Geometry codes

➜ An AG code is defined by a triplet

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

➜ $\mathcal{X}$ 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

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

Algebraic Geometry codes II

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

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

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

Algebraic Geometry Code (AG code)

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

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

Algebraic Geometry codes III

➜ If $\{f_1, \dots, f_k\}$ is a basis of $L(E)$ then

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

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 $\deg(E) < n$ then

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

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

Algebraic Geometry codes IV

Dual of an AG code

Let:

- ω be a **differential form** with a simple pole and residue 1 at P_j for all $j = 1, \dots, n$.
- 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 $\quad E^\perp = D_\mathcal{P} - E + K \quad$ and $\quad \deg(E^\perp) = n - \deg(E) + 2g - 2$

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

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

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

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

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

- Such a pair **always exists** whenever

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

where $d^* = \deg(E) - 2g + 2$ is the designed minimum distance of $\mathcal{C}$

t-ECP for AG codes II

Corollary [Main Corollary]

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

Polynomial time attack against McEliece based in AG codes - Retrieving an ECP

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**:

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

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

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

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

➜ **Future work!!!**

The P-Filtration

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

$$\mathcal{B}_i := (\mathcal{C}_L(\mathcal{X},\mathcal{P},E-iP_1))_{i\in\mathbb{N}}$$

Which Elements of the Sequence do We know?

1. From a generator matrix of $\mathcal{C}_{\mathrm{pub}} = \mathcal{C}_L(\mathcal{X},\mathcal{P},E)^\perp$ 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 $\mathcal{B}_0$ which are zero at position P_1.
   - Computed by **Gaussian elimination**.

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

Effective computation - Algorithm I

Proposition

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

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

Then,

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

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

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

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

$$\underbrace{C_L(\mathcal{X}, \mathcal{P}, E-2P_1)}_{\mathcal{B}_2} * \underbrace{\mathcal{C}_L(\mathcal{X}, \mathcal{P}, E-P_1)}_{\mathcal{B}_0} \subseteq \underbrace{\mathcal{C}_L(\mathcal{X}, \mathcal{P}, 2E-2P_1)}_{\mathcal{B}_1^{(2)}}$$

Effective computation - Algorithm I

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

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

$$\underbrace{\mathcal{C}_L(\mathcal{X}, \mathcal{P}, E - (s+1)P_1)}_{\mathcal{B}_{s+1}} * \underbrace{\mathcal{C}_L(\mathcal{X}, \mathcal{P}, E - (s-1)P_1)}_{\mathcal{B}_{s-1}} \subseteq \underbrace{\mathcal{C}_L(\mathcal{X}, \mathcal{P}, 2E - 2sP_1)}_{\mathcal{B}_s^{(2)}}$$

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

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

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

Polynomial time attack against McEliece based in AG codes - Retrieving an ECP

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

The Algorithm: Suppose that $\dfrac{n}{2} \geq \deg(E)$.

STEP 1. Determine the values g and $\deg(E)$ using the following Proposition.

Proposition

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

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

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

Complexity of the Attack:

➜ 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

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

➜ 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.

Example:

- 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

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

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

w computed with Christiane Peters software

Alternant codes

Let

- $\mathbf{a} = (a_1, \ldots, a_n)$ be an n-tuple of **mutually distinct** elements of $\mathbb{F}_{q^m}$.
- $\mathbf{b} = (b_1, \ldots, b_n)$ be an n-tuple of **nonzero** elements of $\mathbb{F}_{q^m}$.

➜ $\mathrm{GRS}_k(\mathbf{a}, \mathbf{b})$ be the GRS code over $\mathbb{F}_{q^m}$ of dimension k.

The **alternant code** $\mathrm{Alt}_r(\mathbf{a}, \mathbf{b})$ is the $\mathbb{F}_q$-linear restriction:

$$\mathrm{Alt}_r(\mathbf{a}, \mathbf{b}) = \mathbb{F}_q^n \cap (\mathrm{GRS}_r(\mathbf{a}, \mathbf{b}))^\perp$$

Parameters of $\mathrm{Alt}_r(\mathbf{a}, \mathbf{b})$

The $\mathrm{Alt}_r(\mathbf{a}, \mathbf{b})$ has parameters $[n, k, d]_q$ with:

$$k \geq n - mr \qquad \text{and } d \geq r + 1$$

Every $[n, k, d]$ linear code with $d \geq 2$ is an **alternant code**!

t-ECP FOR ALTERNANT CODES

Let $\mathcal{C} = \mathrm{Alt}_{2t}(\mathbf{a}, \mathbf{b})$. Then:

$$d(\mathcal{C}) \geq 2t + 1 \quad \text{and} \quad \mathcal{C} \subseteq (\mathrm{GRS}_{2t+1}(\mathbf{a}, \mathbf{b}))^{\perp}$$

Let

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

then (A, B) is a t-ECP over $\mathbb{F}_{q^m}$ for $\mathcal{C}$.

No known structural attacks against code-base PKC using Alternant codes

Goppa codes

Let

- $\mathbf{a} = (a_1, \ldots, a_n)$ be an n-tuple of **mutually distinct** elements of $\mathbb{F}_{q^m}$.
- g be a polynomial **with coefficients** in $\mathbb{F}_{q^m}$ such that

$$g(a_j) \neq 0 \text{ for all } j = 1, \ldots, n$$

The **Goppa code** $\Gamma(\mathbf{a}, g)$ is the $\mathbb{F}_q$-linear code defined by:

$$\Gamma(\mathbf{a}, g) = \left\{ \mathbf{c} \in \mathbb{F}_q^n \mid \sum_{j=1}^{n} \frac{c_j}{X - a_j} \equiv 0 \mod g(X) \right\}$$

t-ECP for Goppa codes

Goppa codes are alternant codes

Let

- $\mathbf{a} = (a_1, \ldots, a_n)$ be an n-tuple of **mutually distinct** elements of $\mathbb{F}_{q^m}$.
- g be a **Goppa polynomial** of degree r.
- $\mathbf{b} = (b_1, \ldots, b_n)$ be an n-tuple of **nonzero** elements of $\mathbb{F}_{q^m}$ such that
$$b_j = \frac{1}{g(a_j)}$$

Then: $\Gamma(\mathbf{a}, g) = \mathrm{Alt}_r(\mathbf{a}, \mathbf{b}) \Longrightarrow$ it has an $\left\lfloor \frac{r}{2} \right\rfloor$-ECP

No known structural attacks against code-base PKC using Binary Goppa codes

t-ECP FOR CYCLIC CODES

➜ **ECP** for cyclic codes were found **beyond half the BCH bound** by Duursma (1993) and Kötter (1996).

I. Duursma
Decoding codes from curves and cyclic codes.
Ph.D thesis, Eindhoven University of Technology (1993)

I. Duursma, R. Kötter.
Error-locating pairs for cyclic codes.
IEEE Trans. Inform. Theory, Vol.40, 1108–1121 (1994)

R. Kötter.
On algebraic decoding of algebraic-geometric and cyclic codes.
Ph.D thesis, Linköping University of Technology (1996).

Conclusions

- We propose for the **McEliece cryptosystem** the class of codes $\mathcal{C}_t$
  - with a t-ECP
  - but whose error-correcting pair is note easily reconstructed from a given generator matrix.

- That is: the security of the McEliece cryptosystem is not only based on the inherent intractability of bounded distance decoding but on the one-way function:

$$\mathbf{x} = (A, B) \quad \longmapsto \quad \mathbf{y} = A * B$$

- **First Question:** If a code has a t-ECP, how difficult/easy is to retrieve such a pair?
- **Second Question:** It is possible to distinguish a random code from one having a t-ECP?

Thank you for your attention!