Polynomial Time Attack against Wild McEliece over Quadratic Extensions

Alain Couvreur, Ayoub Otmani, Jean-Pierre Tillich

INRIA/École Polytechnique, Université de Rouen, INRIA

EUROCRYPT 2014, Copenhagen

Table of Contents

1. McEliece Encryption Scheme
2. Square codes and distinguishers
3. Our attack

McEliece scheme

- **Secret Key :** A generator matrix $\boldsymbol{G} \in \mathcal{M}_{k \times n}(\mathbb{F}_q)$ of a code $\mathscr{C}$ having an efficient t–correcting algorithm ;
- **Public Key :** $\boldsymbol{G'} := S\boldsymbol{G}P$, where $S \in \boldsymbol{GL}(k, \mathbb{F}_q)$ and P is an $n \times n$ permutation matrix ;
- **Encryption :** $m \in \mathbb{F}_q^k \quad \longmapsto \quad y \stackrel{\text{def}}{=} m\boldsymbol{G'} + e.$
- **Decryption :**
  $y \quad \longmapsto \quad yP^{-1} = mS\boldsymbol{G} + eP^{-1} \quad \longmapsto \quad mS \quad \longmapsto \quad m.$

Advantages and drawbacks

Advantages

- Post Quantum ;
- Efficient encryption and decryption (compared to RSA, El Gamal) : For instance, the original McEliece has
  - encryption $\approx$ 5 times quicker than RSA 1024 (with public exponent 17)
  - decryption $\approx$ 150 times quicker than RSA 1024.

Drawbacks

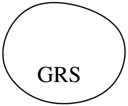
- Huge size of the keys : The original proposal (McEliece 1977) : $[1024, 524, 101]_2$ has a 67ko key (more than 500 times RSA 1024 for a similar security).

Definition (Generalized Reed–Solomon Codes (GRS))

Let

- $\boldsymbol{x} = (x_1, \ldots, x_n) \in \mathbb{F}_q^m$ with the x_i's pairwise distinct.
- $\boldsymbol{y} = (y_1, \ldots, y_n) \in \mathbb{F}_q^n$, with the y_i's nonzero.

$$\mathbf{GRS}_k(\boldsymbol{x}, \boldsymbol{y}) \stackrel{\text{def}}{=} \{(y_1 f(x_1), \ldots, y_n f(x_n)) \mid f \in \mathbb{F}_q[x], \deg f < k\}$$


GRS

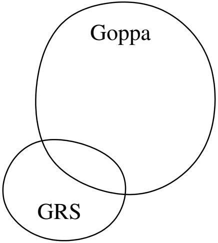
Definition

Let $x \in \mathbb{F}_{q^m}^n$ be a support and $\Gamma \in \mathbb{F}_{q^m}[x]$. The Goppa code $\mathscr{G}(x, \Gamma)$ is defined as

$\mathscr{G}(x, \Gamma) \stackrel{\text{def}}{=}$

$$\mathbf{GRS}_{\deg \Gamma}(x, y)^{\perp} \cap \mathbb{F}_q^n.$$

and $\forall i,\ y_i = \frac{1}{\Gamma(x_i)}$.



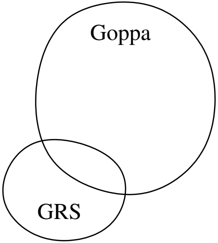
Definition

Let $x \in \mathbb{F}_{q^m}^n$ be a support and $\Gamma \in \mathbb{F}_{q^m}[x]$. The Goppa code $\mathscr{G}(x,\Gamma)$ is defined as

$$\mathscr{G}(x,\Gamma) \stackrel{\text{def}}{=}$$

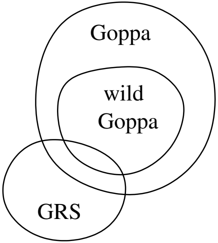
$$\mathbf{GRS}_{\deg \Gamma}(x,y)^{\perp} \cap \mathbb{F}_q^n.$$

and $\forall i,\ y_i = \frac{1}{\Gamma(x_i)}$.



Definition

When the Goppa polynomial Γ is of the form $\Gamma(z) = \gamma(z)^q$ for some squarefree $\gamma \in \mathbb{F}_{q^m}[z]$, the Goppa code is said to be *wild*.

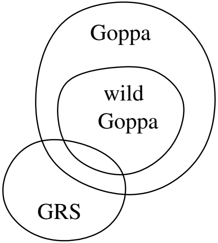


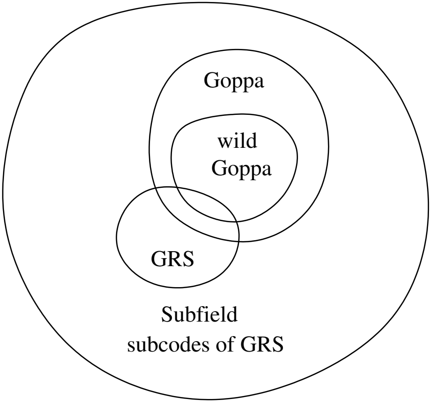
Definition

When the Goppa polynomial Γ is of the form $\Gamma(z) = \gamma(z)^q$ for some squarefree $\gamma \in \mathbb{F}_{q^m}[z]$, the Goppa code is said to be *wild*.

Wild Goppa codes have

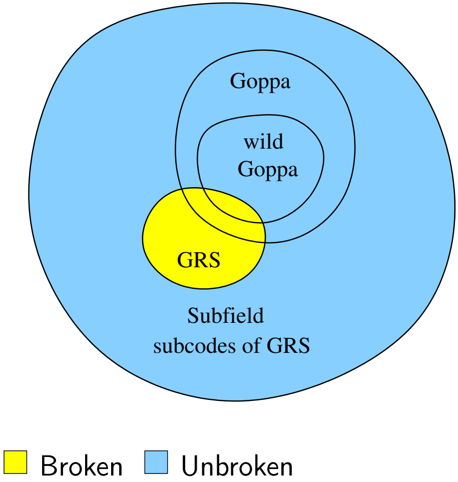
- Better correction capacity (Sugyiama et al. 1976)
- hence provide a higher security (Bernstein, Lange, Peters, 2010)

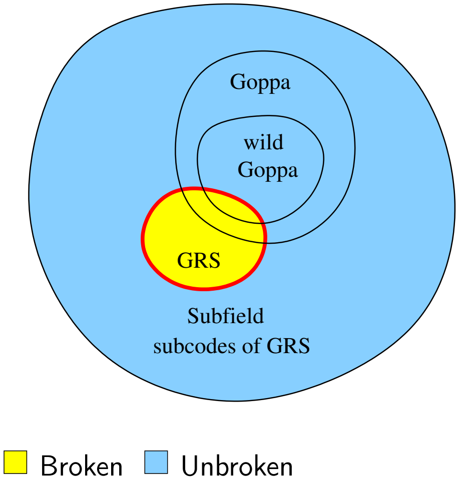



Goppa
wild
Goppa
GRS
Subfield
subcodes of GRS

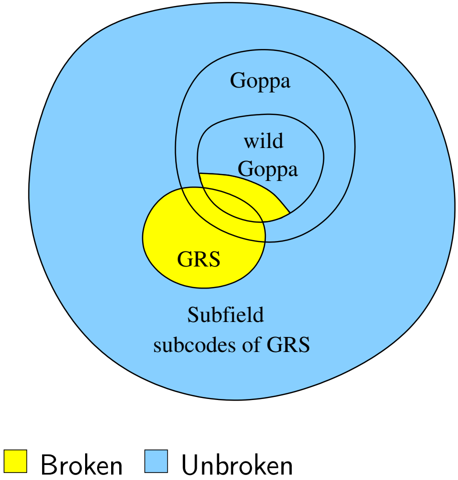
GRS codes are proposed for McEliece by Niederreiter (1986).

Sidelnikov, Shestakov (1992) give a key-recovery attack in $O(n^3)$.




Goppa
wild
Goppa
GRS
Subfield
subcodes of GRS
Broken
Unbroken

Our contribution :



Given two codes $\mathscr{A}, \mathscr{B}$ in $\mathbb{F}_q^n$,

$$\mathscr{A} \star \mathscr{B} \stackrel{\text{def}}{=} \mathrm{Span}_{\mathbb{F}_q} \{ \boldsymbol{a} \star \boldsymbol{b} \mid \boldsymbol{a} \in \mathscr{A},\ \boldsymbol{b} \in \mathscr{B} \}.$$

$\star$ denotes the component wise product : $a \star b \stackrel{\text{def}}{=} (a_1 b_1, \ldots, a_n b_n)$.

Proposition

$$\dim(\mathscr{A} \star \mathscr{A}) \leqslant \min \left\{ n, \binom{\dim \mathscr{A} + 1}{2} \right\}$$

Theorem (Cascudo, Cramer, Mirandola, Zémor. (In progress))

Let $\mathscr{A}$ be a random code of length n and dimension k such that $n > \binom{k+1}{2}$. Then for all integer $\ell < \binom{k+1}{2}$

$$\boldsymbol{Prob}\left(\dim(\mathscr{A} \star \mathscr{A}) \leqslant \binom{\dim \mathscr{A} + 1}{2} - \ell \right) = o(q^{-\ell}). \quad (k \to +\infty)$$

Distinguisher on GRS codes

Theorem

Let $\mathbf{x}, \mathbf{y} \in \mathbb{F}_q^n$ *be a support and a multiplier. Let* $k < n/2$*, then*

$$\mathbf{GRS}_k(\mathbf{x}, \mathbf{y})^{\star 2} = \mathbf{GRS}_{2k-1}(\mathbf{x}, \mathbf{y}^{\star 2})$$

and hence :

$$\dim \mathbf{GRS}_k(\mathbf{x}, \mathbf{y})^{\star 2} = 2k - 1.$$

Distinguisher on GRS codes

Theorem

Let $\mathbf{x}, \mathbf{y} \in \mathbb{F}_q^n$ be a support and a multiplier. Let $k < n/2$, then

$$\mathbf{GRS}_k(\mathbf{x}, \mathbf{y})^{\star 2} = \mathbf{GRS}_{2k-1}(\mathbf{x}, \mathbf{y}^{\star 2})$$

and hence :

$$\dim \mathbf{GRS}_k(\mathbf{x}, \mathbf{y})^{\star 2} = 2k - 1.$$

Application (Wieschebrink (2010))

An attack against Berger Loidreau proposal (2005) based on subcodes of low codimension of GRS codes.

Our Attack

Public key : $\mathscr{C}$: a Goppa code $\mathscr{G}(\boldsymbol{x}, \boldsymbol{\gamma}^q)$ over a quadratic extension $(m = 2)$.

Distinguisher by shortening

In general Goppa codes are not distinguishable by squares. But in the specific case of wild Goppa Codes over a quadratic extension :

Theorem (C-, Otmani, Tillich 2014)

$\mathscr{G}(x, \gamma^{q-1})$ shortened at a positions is distinguishable if $a \in \{a^-, \dots, a^+\}$:

$$\begin{aligned} a^- &= n - 2r(q+1) - 1 \\ a^+ &= \max\left\{ a \geqslant 0 \;\middle|\; \begin{array}{c} 3(n-a) - 4r(q+1) - 2 \leqslant \\ \min\left\{ n-a, \binom{n-a-2r(q-1)+r(r-2)}{2} \right\} \end{array} \right\} \end{aligned}$$

Remark

The interval $\{a^-, \dots, a^+\}$ is nonempty if :

when $q \geqslant$	9	19	37	64
$r >$	2	3	4	5

The heart of our attack

The heart of our attack

We know

$$\begin{array}{lcr} \mathscr{C}_0 \stackrel{\text{def}}{=} \mathscr{C} & \longleftrightarrow & \mathbb{F}_{q^2}[x] \\ \mathscr{C}_1 & \longleftrightarrow & x\mathbb{F}_{q^2}[x] \end{array}$$

$\mathscr{C}_1$ is obtained by computing the words having some entry set to zero (elementary linear algebra).

The heart of our attack

We know

$$\begin{array}{lcr} \mathscr{C}_0 \stackrel{\text{def}}{=} \mathscr{C} & \longleftrightarrow & \mathbb{F}_{q^2}[x] \\ \mathscr{C}_1 & \longleftrightarrow & x\mathbb{F}_{q^2}[x] \end{array}$$

$\mathscr{C}_1$ is obtained by computing the words having some entry set to zero (elementary linear algebra).

To compute $\mathscr{C}_2 \longleftrightarrow x^2\mathbb{F}_{q^2}[x]$, notice that

$$\mathscr{C} \star \mathscr{C}_2 \subseteq \mathscr{C}_1 \star \mathscr{C}_1.$$

The heart of our attack

We know

$$\begin{array}{lcr} \mathscr{C}_0 \stackrel{\text{def}}{=} \mathscr{C} & \longleftrightarrow & \mathbb{F}_{q^2}[x] \\ \mathscr{C}_1 & \longleftrightarrow & x\mathbb{F}_{q^2}[x] \end{array}$$

$\mathscr{C}_1$ is obtained by computing the words having some entry set to zero (elementary linear algebra).

To compute $\mathscr{C}_2 \longleftrightarrow x^2\mathbb{F}_{q^2}[x]$, notice that

$$\mathscr{C} \star \mathscr{C}_2 \subseteq \mathscr{C}_1 \star \mathscr{C}_1.$$

Hence, $\mathscr{C}_2$ can be computed as the set of solutions z of

$$\left\{ \begin{array}{ccc} z & \in & \mathscr{C}_1 \\ z \star \mathscr{C} & \subseteq & \mathscr{C}_1 \star \mathscr{C}_1 \end{array} \right. .$$

Our Attack

- **Step 1.** Compute

$$\mathscr{C} = \mathscr{C}_0$$

Our Attack

- **Step 1.** Compute

$$\mathscr{C} = \mathscr{C}_0 \supseteq \mathscr{C}_1$$

Our Attack

- **Step 1.** Compute

$$\mathscr{C} = \mathscr{C}_0 \supseteq \mathscr{C}_1 \supseteq \mathscr{C}_2 \supseteq \cdots$$

Our Attack

- **Step 1.** Compute

$$\mathscr{C} = \mathscr{C}_0 \supseteq \mathscr{C}_1 \supseteq \mathscr{C}_2 \supseteq \cdots \supseteq \mathscr{C}_{q+1}$$

Our Attack

- **Step 1.** Compute

$$\mathscr{C} = \mathscr{C}_0 \supseteq \mathscr{C}_1 \supseteq \mathscr{C}_2 \supseteq \cdots \supseteq \mathscr{C}_{q+1}$$

- **Step 2.** From $\mathscr{C}_{q+1}$, one can compute $\boldsymbol{x}^{\star(q+1)} = (x_0^{q+1}, x_1^{q+1}, \ldots, x_{n-1}^{q+1})$. (It uses the norm over $\mathbb{F}_{q^2}$.)

  Reapplying Step 1 and 2, one can also compute : $(\boldsymbol{x} - \boldsymbol{1})^{\star(q+1)} = ((x_0 - 1)^{q+1}, (x_1 - 1)^{q+1}, \ldots, (x_{n-1} - 1)^{q+1})$

Our Attack

- **Step 1.** Compute

$$\mathscr{C} = \mathscr{C}_0 \supseteq \mathscr{C}_1 \supseteq \mathscr{C}_2 \supseteq \cdots \supseteq \mathscr{C}_{q+1}$$

- **Step 2.** From $\mathscr{C}_{q+1}$, one can compute $\boldsymbol{x}^{\star(q+1)} = (x_0^{q+1}, x_1^{q+1}, \ldots, x_{n-1}^{q+1})$. (It uses the norm over $\mathbb{F}_{q^2}$.)

  Reapplying Step 1 and 2, one can also compute :
  $(\boldsymbol{x}-\boldsymbol{1})^{\star(q+1)} = ((x_0-1)^{q+1}, (x_1-1)^{q+1}, \ldots, (x_{n-1}-1)^{q+1})$

- **Step 3.** Deduce from $\boldsymbol{x}^{\star(q+1)}$ and $(\boldsymbol{x}-\boldsymbol{1})^{\star(q+1)}$ the support $\boldsymbol{x}$ up to Galois action.

Our Attack

- **Step 1.** Compute

$$\mathscr{C} = \mathscr{C}_0 \supseteq \mathscr{C}_1 \supseteq \mathscr{C}_2 \supseteq \cdots \supseteq \mathscr{C}_{q+1}$$

- **Step 2.** From $\mathscr{C}_{q+1}$, one can compute $\boldsymbol{x}^{\star(q+1)} = (x_0^{q+1}, x_1^{q+1}, \ldots, x_{n-1}^{q+1})$. (It uses the norm over $\mathbb{F}_{q^2}$.)

  Reapplying Step 1 and 2, one can also compute : $(\boldsymbol{x} - \mathbf{1})^{\star(q+1)} = ((x_0 - 1)^{q+1}, (x_1 - 1)^{q+1}, \ldots, (x_{n-1} - 1)^{q+1})$

- **Step 3.** Deduce from $\boldsymbol{x}^{\star(q+1)}$ and $(\boldsymbol{x} - \mathbf{1})^{\star(q+1)}$ the support $\boldsymbol{x}$ up to Galois action.

- **Step 4.** A bit more technique to deduce $\boldsymbol{x}$ and the Goppa Polynomial γ.

Complexity and running times

Complexity : $O(n^4\sqrt{n} + n^4(q^2 - n))$ (recall that $n \leqslant q^2$).

Table : Running times with an Intel® Xeon 2.27GHz

$[q, n, k, r]$	[29,781, 516,5] ☣	[29, 791, 575, 4] ☣	[29,794,529,5] ☣
Average time	16min	19.5min	15.5min

(q, n, k, r)	[31, 795, 563, 4] ☣	[31,813, 581,4] ☣	[31, 851, 619, 4] ☣
Average time	31.5min	31.5min	27.2min

(q, n, k, r)	[32,841,601,4] ☣	[31, 900, 228, 14]
Average time	49.5min	24min

Proposed parameters (Bernstein, Lange, Peters 2010)
Never proposed parameters

Complexity and running times

Complexity : $O(n^4\sqrt{n} + n^4(q^2 - n))$ (recall that $n \leqslant q^2$).

Table : Running times with an Intel® Xeon 2.27GHz

$[q, n, k, r]$	[29,781, 516,5] ☣	[29, 791, 575, 4] ☣	[29,794,529,5] ☣
Average time	16min	19.5min	15.5min

(q, n, k, r)	[31, 795, 563, 4] ☣	[31,813, 581,4] ☣	[31, 851, 619, 4] ☣
Average time	31.5min	31.5min	27.2min

(q, n, k, r)	[32,841,601,4] ☣	[31, 900, 228, 14]
Average time	49.5min	24min

Proposed parameters (Bernstein, Lange, Peters 2010)
Never proposed parameters (More than 2^{130} possible choices for γ and security $>$ 125 bits with respect to ISD)

Conclusion

- We broke McEliece based on Wild Goppa codes $\mathscr{G}(x, \gamma^{q-1})$ for
  - $m = 2$;
  - $\deg \gamma$ s.t. :

when $q \geqslant$	9	19	37	64
$r >$	2	3	4	5

Conclusion

- We broke McEliece based on Wild Goppa codes $\mathscr{G}(x, \gamma^{q-1})$ for
  - $m = 2$;
  - $\deg \gamma$ s.t. :

when $q \geqslant$	9	19	37	64
$r >$	2	3	4	5

- It is the first polynomial time key-recovery attack against a family of non trivial subfield subcodes of GRS codes.

Conclusion

- We broke McEliece based on Wild Goppa codes $\mathscr{G}(\mathbf{x}, \gamma^{q-1})$ for
  - $m = 2$;
  - $\deg \gamma$ s.t. :

when $q \geqslant$	9	19	37	64
$r >$	2	3	4	5

- It is the first polynomial time key-recovery attack against a family of non trivial subfield subcodes of GRS codes.
- From a distinguisher, we got an attack.

Conclusion

- We broke McEliece based on Wild Goppa codes $\mathscr{G}(\mathbf{x}, \gamma^{q-1})$ for
  - $m = 2$;
  - $\deg \gamma$ s.t. :

when $q \geqslant$	9	19	37	64
$r >$	2	3	4	5

- It is the first polynomial time key-recovery attack against a family of non trivial subfield subcodes of GRS codes.
- From a distinguisher, we got an attack.
- Question : are other distingushable codes breakable ? For instance high rate Goppa codes (distinguisher on the dual).

Thank you for your attention.