A distinguisher for high-rate McEliece Cryptosystems

J.C. Faugère (INRIA, SALSA project),
A. Otmani (Université Caen- INRIA, SECRET project),
L. Perret (INRIA, SALSA project),
J.-P. Tillich (INRIA, SECRET project)

May 28th, 2010

1. Algebraic approach for attacking the McEliece cryptosystem

- $\boldsymbol{x} = (x_1, \dots, x_n) \in \mathbb{F}_{q^m}^n$ with $x_i \neq x_j$ if $i \neq j$
- $\boldsymbol{y} = (y_1, \dots, y_n) \in \mathbb{F}_{q^m}^n$ with $y_i \neq 0$

For any $t < n$, let $\boldsymbol{H} \stackrel{\text{def}}{=} \begin{pmatrix} y_1 & y_2 & \cdots & y_n \\ y_1 x_1 & y_2 x_2 & \cdots & y_n x_n \\ \vdots & \vdots & & \vdots \\ y_1 x_1^{t-1} & y_2 x_2^{t-1} & \cdots & y_n x_n^{t-1} \end{pmatrix}$

Definition 1. *An alternant code is the kernel of an $\boldsymbol{H}$ of this type*

$$\mathcal{A}_t(\boldsymbol{x}, \boldsymbol{y}) = \left\{ \boldsymbol{v} \in \mathbb{F}_q^n | \boldsymbol{H} \boldsymbol{v}^T = \boldsymbol{0}. \right\}.$$

Goppa code : $\exists \Gamma$, polynomial of degree t such that $y_i = \Gamma(x_i)^{-1}$.

Decoding Alternant and Goppa codes

Proposition 1. [decoding alternant codes] *$t/2$ errors can be decoded in polynomial time as long as $\boldsymbol{x}$ and $\boldsymbol{y}$ are known.*

Proposition 2. [The special case of binary Goppa codes] *In the case of a binary Goppa code ($q = 2$), t errors can be decoded in polynomial time, if $\boldsymbol{x}$ and Γ are known.*

The problem

What is known: a basis of the code $\longrightarrow$ rows of a generator matrix $\boldsymbol{G} = (g_{ij})$ of size $k \times n$.

What we also know:

$$\boldsymbol{H}\boldsymbol{G}^T = \mathbf{0}. \tag{1}$$

What we want to find: $\boldsymbol{H}$

Find in the case of an alternant code $\boldsymbol{x}, \boldsymbol{y}$, and in the special case of a binary Goppa code $\boldsymbol{x}$ and Γ.

The algebraic system

$\boldsymbol{H}\boldsymbol{G}^T = \mathbf{0}$ translates to

$$\left\{\begin{array}{lcl} g_{1,1}Y_1 + \cdots + g_{1,n}Y_n & = & 0 \\ \vdots & & \vdots \\ g_{k,1}Y_1 + \cdots + g_{k,n}Y_n & = & 0 \\ g_{1,1}Y_1X_1 + \cdots + g_{1,n}Y_nX_n & = & 0 \\ \vdots & & \vdots \\ g_{k,1}Y_1X_1 + \cdots + g_{k,n}Y_nX_n & = & 0 \\ \vdots & & \vdots \\ g_{1,1}Y_1X_1^{t-1} + \cdots + g_{1,n}Y_nX_n^{t-1} & = & 0 \\ \vdots & & \vdots \\ g_{k,1}Y_1X_1^{t-1} + \cdots + g_{k,n}Y_nX_n^{t-1} & = & 0 \end{array}\right. \qquad (2)$$

where the $g_{i,j}$'s are known coefficients in $\mathbb{F}_q$ and $k \geq n - t\,m$.

Freedom of choice in (2)

Proposition 3. *Theoretically, the system has $2n$ unknowns but we can take arbitrary values for one Y_i and for three X_i's (as long as these values are different).*

Applications

When the number of unknowns is small, ex:

• Berger-Cayrel-Gaborit-Otmani proposal at AfricaCrypt'09 based on quasi-cyclic alternant codes

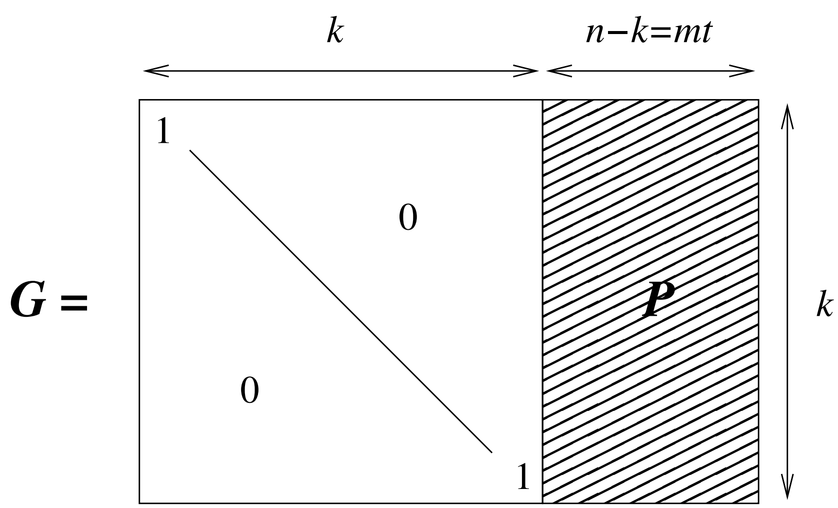
• Misoczki-Baretto at SAC'09 variant based on quasi-dyadic Goppa codes

$\Rightarrow$ algebraic system can be solved by (dedicated) Grobner basis techniques.

- breaks all parameters proposed in these articles ([Faugère-Otmani-Perret-Tillich;Eurocrypt 2010] with the exception of binary dyadic codes. Related to [Leander-Gauthier Umana; SCC2010]

2. A naive attack

W.l.o.g. we can assume that $\boldsymbol{G}$ is systematic in its k first positions.



Step 1 – expressing the $Y_iX_i^d$'s in terms of the $Y_jX_j^d$'s for $j \in \{k+1, \ldots, n\}$.

$\boldsymbol{P} = (p_{ij})_{\substack{1 \leq i \leq k \\ k+1 \leq j \leq n}}$. We can rewrite (2) as

$$\begin{cases} Y_i & = & \sum_{j=k+1}^{n} p_{i,j} Y_j \\ Y_i X_i & = & \sum_{j=k+1}^{n} p_{i,j} Y_j X_j \\ & \ldots & \\ Y_i X_i^{t-1} & = & \sum_{j=k+1}^{n} p_{i,j} Y_j X_j^{t-1} \end{cases} \tag{3}$$

for all $i \in \{1, \ldots, k\}$.

Step 2.– Exploiting $Y_i(Y_i X_i^2) = (Y_i X_i)^2$

$$\left\{\begin{array}{lcr} Y_i & = & \sum_{j=k+1}^{n} p_{i,j} Y_j \\ Y_i X_i & = & \sum_{j=k+1}^{n} p_{i,j} Y_j X_j \\ Y_i X_i^2 & = & \sum_{j=k+1}^{n} p_{i,j} Y_j X_j^2 \end{array}\right. \tag{4}$$

$$\Rightarrow \quad \left(\sum_{j=k+1}^{n} p_{i,j} Y_j\right)\left(\sum_{j=k+1}^{n} p_{i,j} Y_j X_j^2\right) = \left(\sum_{j=k+1}^{n} p_{i,j} Y_j X_j\right)^2$$

$$\Rightarrow \quad \sum_{j=k+1}^{n} \sum_{j'>j} p_{i,j} p_{i,j'} \left(Y_j Y_{j'} X_{j'}^2 + Y_{j'} Y_j X_j^2\right) = 0$$

Step 3. – Linearization

$$Z_{jj'} \stackrel{\text{def}}{=} Y_j Y_{j'} X_{j'}^2 + Y_{j'} Y_j X_j^2$$
$$\sum_{j=k+1}^{n} \sum_{j'>j} p_{i,j} p_{i,j'} Z_{jj'} = 0.$$

- $\binom{n-k}{2} \approx \frac{m^2t^2}{2}$ unknowns
- $k = n - mt$ equations

$\Rightarrow$ reveals $Z_{jj'}$ when $n - mt \geq \frac{m^2t^2}{2}$?

- This happens for the Courtois-Finiasz-Sendrier scheme, ex: $n = 2^{21}$,$t = 10$,$m = 21$ which has to choose small values of t.

This approach always fails...

$D_{\text{alternant}}$, resp. D_{Goppa} dimension of the linear solution space when $\boldsymbol{G}$ is the generator matrix of an alternant code, resp. Goppa code.

Experimental fact 1. *Let* $D_{\textit{rand}} \stackrel{\text{def}}{=} \binom{mt}{2} - k$, *with high probability*

$$
\begin{aligned}
D_{\textit{alternant}} &= \max\left(D_{\textit{rand}}, \frac{m(t-1)}{2}\left\{(2\ell+1)t - 2\frac{q^{\ell+1}-1}{q-1}\right\}\right) \\
&\quad \textit{for } \ell \stackrel{\text{def}}{=} \left\lfloor \log_q(t-1) \right\rfloor \\
D_{\textit{Goppa}} &= D_{\textit{alternant}} = \max\left(D_{\textit{rand}}, \frac{m(t-1)(t-2)}{2}\right) \textit{ for } t < q-1 \\
D_{\textit{Goppa}} &= \max\left(D_{\textit{rand}}, \frac{mt}{2}\left\{(2\ell+1)t - 2q^{\ell} + 2q^{\ell-1} - 1\right\}\right),
\end{aligned}
$$

for $t \geq q-1$ *and with* ℓ *s.t.* $q^{\ell} - 2q^{\ell-1} + q^{\ell-2} < t \leq q^{\ell+1} - 2q^{l} + q^{\ell-1}$

Table 1: $q = 2$ and $m = 10$

t	3	4	5	6	7	8	9	10	11
$\binom{mt}{2}$	435	780	1225	1770	2415	3160	4005	4950	5995
k	994	984	974	964	954	944	934	924	914
D_{rand}	0	0	251	806	1461	2216	3071	4026	5081
$D_{\text{alternant}}$	30	90	251	806	1461	2216	3071	4026	5081
$T_{\text{alternant}}$	30	90	220	400	630	910	1320	1800	2350
D_{Goppa}	180	380	700	1110	1610	2216	3071	4026	5081
T_{Goppa}	180	380	700	1110	1610	2200	2970	3850	4840

3. A Distinguisher

$$D_{\mathsf{Goppa}} \geq D_{\mathsf{alternant}} \geq D_{\mathsf{rand}}$$

Table 2: $t_{\min}$ = smallest degree of the Goppa polynomial Γ for which we can not distinguish a binary Goppa code from a random binary linear code when $n = 2^m$.

m	9	10	11	12	13	14	15	16	17	18	19	20	21
$t_{\min}$	8	8	11	16	20	26	34	47	62	85	114	157	213

An explanation for the distinguisher

We have used

$$Y_i Y_i X_i^2 = (Y_i X_i)^2$$

Any identity of the form

$$Y_i X_i^a Y_i X_i^b = Y_i X_i^c Y_i X_i^d$$

with $a, b, c, d \in \{0, 1, \ldots, t-1\}$ such that $a + b = c + d$ would do the same job:

$$Z_{jj'}^{a,b,c,d} \stackrel{\text{def}}{=} Y_j X_j^a Y_{j'} X_{j'}^b + Y_{j'} X_{j'}^a Y_j X_j^b + Y_j X_j^c Y_{j'} X_{j'}^d + Y_{j'} X_{j'}^c Y_j X_j^d$$

$$\sum_{j=k+1}^{n} \sum_{j'>j} p_{i,j} p_{i,j'} Z_{jj'}^{a,b,c,d} = 0$$

Conclusion

- Combinatorial explanation of the distinguisher in the alternant case. Partial combinatorial explanation in the Goppa case.
- A slightly better distinguisher can be obtained by taking the subcode of codewords of even weights.
- Distinguisher $\Rightarrow$ attack ?
- Approach requires $\frac{k}{n}$ very close to 1. Should very high rates be avoided in a McEliece like scheme ?