Algebraic Analysis of McEliece Cryptosystems

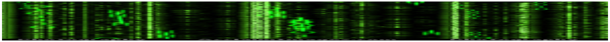
J.-C. Faugère[1] A. Otmani[2,3] L. Perret[1] J.-P. Tillich[2]

SALSA Team-Project – LIP6/UPMC/INRIA Paris-Rocquencourt
jean-charles.faugere@inria.fr, ludovic.perret@lip6.fr

SECRET Team-Project – INRIA Paris-Rocquencourt
ayoub.otmani@inria.fr, jean-pierre.tillich@inria.fr

GREYC – Université de Caen – Ensicaen

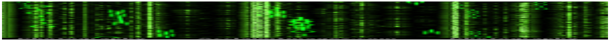
Post-Quantum Cryptography



Known Candidates

- Lattice-based Cryptography
- Multivariate Cryptography
- Code-based Cryptography

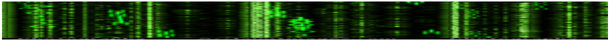
Post-Quantum Cryptography



Known Candidates

- Lattice-based Cryptography
- Multivariate Cryptography
  - Public-key is large, but recent papers seem to mitigate this issue [Petzoldt, Bulygin, Buchmann]
- Code-based Cryptography
  - Public-key is large, but key-reduction techniques [Barreto et al, Berget et al]
  - provable-security

Post-Quantum Cryptography



Known Candidates

- Lattice-based Cryptography
- Multivariate Cryptography
  - Public-key is large, but recent papers seem to mitigate this issue [Petzoldt, Bulygin, Buchmann]
- Code-based Cryptography
  - Public-key is large, ~~but key-reduction techniques [Barreto et al, Berget et al]~~
  - ~~provable-security~~

McEliece's Cryptosystem [R.J. McEliece, 1978]

- One of the *oldest* public-key cryptosystems
  - based on coding theory
- Principle is to *mask a structured code* in such a way that it *looks like random*
  - Trapdoor $= \boldsymbol{H}_t(\boldsymbol{x}, \boldsymbol{y})$ [parity-check matrix of a *Goppa/alternant* code $\boldsymbol{G}_s$]
  - Public key $=$ Random basis $\boldsymbol{G}$ of $\mathrm{Ker}\Big(\boldsymbol{H}_t(\boldsymbol{x}, \boldsymbol{y})\Big) \cap \mathbb{F}_q^n$

Generation of $(\mathtt{pk}, \mathtt{sk})$

1. Choose a generator matrix $\boldsymbol{G}_s$ of a Goppa (or alternant) code $\mathcal{C}_s$ randomly chosen
2. Pick *at random*:
   - $n \times n$ permutation matrix $\boldsymbol{P}$
   - $k \times k$ non-singular matrix $\boldsymbol{S}$
3. Compute $\boldsymbol{G} = \boldsymbol{S} \times \boldsymbol{G}_s \times \boldsymbol{P}$
4. Output

$$\mathtt{pk} = (\boldsymbol{G}, t) \quad \text{and} \quad \mathtt{sk} = (\boldsymbol{S}, \boldsymbol{G}_s, \boldsymbol{P})$$

Encrypt/Decrypt

$\boldsymbol{c} \in \mathbb{F}_2^n \leftarrow \texttt{Encrypt}(\boldsymbol{m} \in \mathbb{F}_2^k)$

1. Draw at random $\boldsymbol{e} \in \mathbb{F}_2^n$ of Hamming weight at most t
2. Output $\boldsymbol{c} = \boldsymbol{m} \times \boldsymbol{G} \oplus \boldsymbol{e}$

$\boldsymbol{m}' \in \mathbb{F}_2^k \leftarrow \texttt{Decrypt}(\boldsymbol{c}' \in \mathbb{F}_2^n)$

1. Let $\gamma_{\boldsymbol{G}_s} : \mathbb{F}_2^n \to \mathbb{F}_2^k$ be a decoding algorithm associated to $\boldsymbol{G}_s$
2. Compute $\boldsymbol{z} = \boldsymbol{c}' \times \boldsymbol{P}^{-1}$ // $\boldsymbol{z} = (\boldsymbol{m} \times \boldsymbol{S} \times \boldsymbol{G}_s) \oplus (\boldsymbol{e} \times \boldsymbol{P}^{-1})$
3. Compute $\boldsymbol{y} = \gamma_{\boldsymbol{G}_s}(\boldsymbol{z})$ // $\boldsymbol{y} = \boldsymbol{m} \times \boldsymbol{S}$
4. Output $\boldsymbol{m}' = \boldsymbol{y} \times \boldsymbol{S}^{-1}$ // $\boldsymbol{m}' = \boldsymbol{m}$

Security of McEliece – Message-recovery

- Related to the difficulty of inverting `Encrypt`:

$$\boldsymbol{c} \rightsquigarrow (\boldsymbol{m}, \boldsymbol{e}) \quad \text{such that} \quad \boldsymbol{c} = \boldsymbol{m} \times \boldsymbol{G} \oplus \boldsymbol{e}.$$

Given (n, k, t) and a random $k \times n$ matrix $\boldsymbol{G}$, we set:

$$\begin{array}{rccl} f_{\boldsymbol{G},t} : & \mathbb{F}_2^k \times \mathcal{B}_n(\mathbf{0}, t) & \longrightarrow & \mathbb{F}_2^n \\ & (\boldsymbol{x}, \boldsymbol{e}) & \longmapsto & \boldsymbol{m} \times \boldsymbol{G} \oplus \boldsymbol{e} \end{array}$$

where $\mathcal{B}_n(\mathbf{0}, t) = \left\{ \boldsymbol{z} \in \mathbb{F}_2^n \ : \ \mathrm{wt}(\boldsymbol{z}) \leq t \right\}$.
Inverting $f_{\boldsymbol{G},t}$ is NP-Hard (Berlekamp - McEliece - van Tilborg '78)
Best algorithms are based on *Information Set Decoding*

- McEliece ('78), Lee - Brickell ('88), Leon ('88), Stern ('93), ...
- *Binary* codes : Canteaut-Chabaud ('98), Sendrier-Finiasz'08, Bernstein - Lange - Peters ('08,'11), May - Meurer - Thomae ('11) ...

Security of McEliece – Key-recovery (I)

- Related to the difficulty of extracting the secret matrices:

$$\mathbf{G} \rightsquigarrow (\mathbf{S}, \mathbf{G}_s, \mathbf{P}) \quad \text{such that} \quad \mathbf{G} = \mathbf{S} \times \mathbf{G}_s \times \mathbf{P}.$$

- Finding the $(\mathbf{S}, \mathbf{P})$ is not hard in practice if $\mathbf{G}_s$ is *known* (SENDRIER '00)
- No real *structural attack* against McEliece's scheme ...

Goppa Code Distinguishing (GD) [COURTOIS, FINIASZ, AND SENDRIER, 2001]

Let $\mathbf{G} = \mathbf{S} \times \mathbf{G}_s \times \mathbf{P}$ be the public matrix of McEliece's scheme.

- GD is the problem of distinguishing $\mathbf{G}$ from a random matrix of the same *type*.

Security of McEliece – Key-recovery (II)

Goppa Code Distinguishing (GD) [CFS'01]

Let $\boldsymbol{G} = \boldsymbol{S} \times \boldsymbol{G_s} \times \boldsymbol{P}$ be the public matrix of McEliece's scheme.

- GD is the problem of distinguishing $\boldsymbol{G}$ from a random matrix of the same *form*.
  - standard assumption for proving the security (Nojima, Imai, Kobara, Morozov, Sendrier, Finaisz, Dallot, Vergnaut, Véron, ...

H. Dinh, C. Moore, and A. Russell.
"The McEliece Cryptosystem Resists Quantum Fourier Sampling Attacks."
Crypto'11.

Outline

Alternant Codes

Consider two fields $\mathbb{F}_q$ and $\mathbb{F}_{q^m}$ with $q = 2^s$ ($s \geq 1$) and $m \geq 1$

- $\boldsymbol{x} = (x_0, \ldots, x_{n-1}) \in \mathbb{F}_{q^m}^n$ such that $x_i \neq x_j$, if $i \neq j$.
- $\boldsymbol{y} = (y_0, \ldots, y_{n-1}) \in \mathbb{F}_{q^m}^n$ with $y_i \neq 0$.

For any $t < n$, we set:

$$\boldsymbol{H}_t(\boldsymbol{x}, \boldsymbol{y}) = \begin{pmatrix} y_0 & y_1 & \cdots & y_{n-1} \\ y_0 x_0 & y_1 x_1 & \cdots & y_{n-1} x_{n-1} \\ \vdots & \vdots & & \vdots \\ y_0 x_0^{t-1} & y_1 x_1^{t-1} & \cdots & y_{n-1} x_{n-1}^{t-1} \end{pmatrix}.$$

Definition

An *alternant* code $\mathcal{A}_t(\boldsymbol{x}, \boldsymbol{y})$ is the *kernel* of $\boldsymbol{H}_t(\boldsymbol{x}, \boldsymbol{y})$ in $\mathbb{F}_q^n$, i.e.

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

Can be efficiently decoded if $\boldsymbol{x}, \boldsymbol{y}$ are *known*.

Algebraic Cryptanalysis of McEliece – (I)

- **What we have**: $\boldsymbol{G} = (g_{i,j})$ is the public matrix
- **What is known:** rows of $\boldsymbol{G}$ belong to the kernel of $\boldsymbol{H}_t(\boldsymbol{x}, \boldsymbol{y})$

$\Rightarrow$ The *secret vectors* $\boldsymbol{x}$ and $\boldsymbol{y}$ satisfy $\boldsymbol{H}_t(\boldsymbol{x}, \boldsymbol{y})\, \boldsymbol{G}^T = \mathbf{0}_{t,k}$, i.e.

$$\begin{pmatrix} Y_0 & Y_1 & \cdots & Y_{n-1} \\ Y_0 X_0 & Y_1 X_1 & \cdots & Y_{n-1} X_{n-1} \\ \vdots & \vdots & & \vdots \\ Y_0 X_0^{t-1} & Y_1 X_1^{t-1} & \cdots & Y_{n-1} X_{n-1}^{t-1} \end{pmatrix} \boldsymbol{G}^T = \mathbf{0}_{t,k}.$$

Algebraic Cryptanalysis of McEliece – (II)

$\mathrm{McE}_{n,k,t}(\boldsymbol{X},\boldsymbol{Y})=$

$$\begin{cases} \vdots \\ g_{i,0}\,Y_0 X_0^j + \ldots + g_{i,n-1}\,Y_{n-1} X_{n-1}^j = 0 \text{ with } \begin{cases} i \in \{0,\ldots,k-1\} \\ j \in \{0,\ldots,t-1\} \end{cases} \\ \vdots \end{cases}$$

- $g_{i,j}$'s are *known* coefficients in $\mathbb{F}_q$ of the public matrix
- k is an integer $\geq n - t\,m$.

[McEliece, 1978]

$q = 2, m = 10, n = 1024, t = 50 \Rightarrow k \geqslant 524$

- Public key has 250Kbits (60-bit security)
- #variables $\approx$ 2048, #equations $\approx$ 26200.

Algebraic Cryptanalysis of McEliece – (II)

$\mathrm{McE}_{n,k,t}(\boldsymbol{X}, \boldsymbol{Y})$=

$$\left\{ \begin{array}{l} \vdots \\ g_{i,0} Y_0 X_0^j + \ldots + g_{i,n-1} Y_{n-1} X_{n-1}^j = 0 \text{ with } \left\{ \begin{array}{l} i \in \{0, \ldots, k-1\} \\ j \in \{0, \ldots, t-1\} \end{array} \right. \\ \vdots \end{array} \right.$$

- $g_{i,j}$'s are *known* coefficients in $\mathbb{F}_q$ of the public matrix
- k is an integer $\geq n - t\, m$.

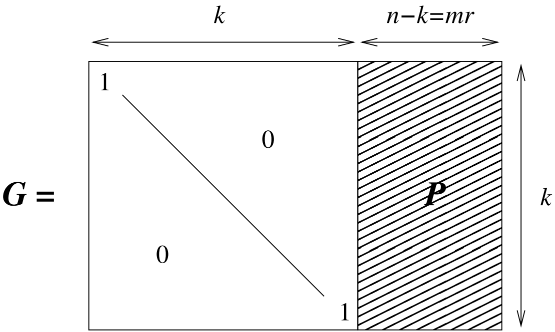
[McEliece, 1978]

$q = 2, m = 10, n = 1024, t = 50 \Rightarrow k \geqslant 524$

- Public key has 250Kbits (60-bit security)
- $\#$variables $\approx$ 2048, $\#$equations $\approx$ 26200.

Outline

Systematic Form of the Public Matrix



- $k = n - m \cdot t$.
- Let $\mathbf{P} = (p_{ij})_{\substack{1 \leq i \leq k \\ k+1 \leq j \leq n}}$ be the sub-matrix of $\boldsymbol{G}$ formed by its last mt columns.

Systematic Form of the System

- Let $\mathbf{P} = (p_{ij})_{\substack{1 \leq i \leq k \\ k+1 \leq j \leq n}}$ be the submatrix of $\boldsymbol{G}$ formed by its last mt columns.

$\mathrm{McE}_{n,k,t}(\boldsymbol{X}, \boldsymbol{Y})$=

$$\begin{cases} \mathbf{Y_i} & = & \sum_{\mathbf{j=k+1}}^{\mathbf{n}} \mathbf{p_{i,j} Y_j}, & \text{for all } i \in \{0, \ldots, k-1\} \\ \mathbf{Y_i X_i} & = & \sum_{\mathbf{j=k+1}}^{\mathbf{n}} \mathbf{p_{i,j} Y_j \cdot X_j}, & \text{for all } i \in \{0, \ldots, k-1\} \\ \mathbf{Y_i X_i^2} & = & \sum_{\mathbf{j=k+1}}^{\mathbf{n}} \mathbf{p_{i,j} Y_j \cdot X_j^2}, & \text{for all } i \in \{0, \ldots, k-1\} \\ & \cdots & & \\ Y_i X_i^{t-1} & = & \sum_{j=k+1}^{n} p_{i,j} Y_j \cdot X_j^{t-1}, & \text{for all } i \in \{0, \ldots, k-1\} \end{cases}$$

[McEliece, 1978]

$q = 2, m = 10, n = 1024, t = 50 \Rightarrow k = 524$

- Public key has 250Kbits (60-bit security)
- $\#$variables $\approx$ 2048, $\#$equations $\approx$ 26200.

Systematic Form of the System

- Let $\mathbf{P} = (p_{ij})_{\substack{1\le i\le k \\ k+1\le j\le n}}$ be the submatrix of $\boldsymbol{G}$ formed by its last mt columns.

$\mathrm{McE}_{n,k,t}(\boldsymbol{X}, \boldsymbol{Y})=$

$$\begin{cases} \mathbf{Y_i} & = & \sum_{\mathbf{j=k+1}}^{\mathbf{n}} \mathbf{p_{i,j} Y_j}, \text{ for all } i \in \{0,\dots,k-1\} \\ \mathbf{Y_i X_i} & = & \sum_{\mathbf{j=k+1}}^{\mathbf{n}} \mathbf{p_{i,j} Y_j \cdot X_j}, \text{ for all } i \in \{0,\dots,k-1\} \\ \mathbf{Y_i X_i^2} & = & \sum_{\mathbf{j=k+1}}^{\mathbf{n}} \mathbf{p_{i,j} Y_j \cdot X_j^2}, \text{ for all } i \in \{0,\dots,k-1\} \\ & \dots & \\ Y_i X_i^{t-1} & = & \sum_{j=k+1}^{n} p_{i,j} Y_j \cdot X_j^{t-1}, \text{ for all } i \in \{0,\dots,k-1\} \end{cases}$$

- Consider the trivial identity $\mathbf{Y_i Y_i X_i^2} = (\mathbf{Y_i X_i})^{\mathbf{2}}$

  i.e. Rows(1)$\times$Rows(3)=Rows(2)^2

Linearization of McEliece

$$
\begin{aligned}
\mathrm{Rows}(1) \times \mathrm{Rows}(3) &= \mathrm{Rows}(2)^2 \\
\left(\sum_{j=k+1}^{n} p_{i,j} Y_j\right) \left(\sum_{j'=k+1}^{n} p_{i,j'} Y_{j'} X_{j'}^2\right) &= \left(\sum_{j=k+1}^{n} p_{i,j} Y_j X_j\right)^2 \\
\left(\sum_{j=k+1}^{n} p_{i,j} Y_j\right) \left(\sum_{j'=k+1}^{n} p_{i,j'} Y_{j'} X_{j'}^2\right) &= \sum_{j=k+1}^{n} p_{i,j}^2 Y_j^2 X_j^2 \quad \text{[Char. 2]}
\end{aligned}
$$

Linearization of McEliece

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

$$\sum_{j=k+1}^{n} p_{i,j}^2 Y_j^2 X_j^2 + \sum_{j=k+1}^{n} \sum_{j' \neq j} p_{i,j} p_{i,j'} Y_j Y_{j'} X_{j'}^2 = \sum_{j=k+1}^{n} p_{i,j}^2 Y_j^2 X_j^2$$

$$\sum_{j=k+1}^{n} \sum_{j' \neq j} p_{i,j} p_{i,j'} Y_j Y_{j'} X_{j'}^2 = 0,\ \forall i \in \{0, \ldots, k-1\},$$

$$\sum_{j=k+1}^{n} \sum_{j' > j}^{n} p_{i,j} p_{i,j'} Y_j Y_{j'} \left(X_j^2 + X_{j'}^2\right) = 0, \forall i \in \{0, \ldots, k-1\}.$$

Linearization of McEliece

$$\begin{aligned}\sum_{j=k+1}^{n}\sum_{j'>j}^{n} p_{i,j}p_{i,j'} Y_j Y_{j'}\left(X_j^2+X_{j'}^2\right) &= 0,\ \text{for all } i\in\{0,\ldots,k-1\},\\ \sum_{j=k+1}^{n}\sum_{j'>j}^{n} p_{i,j}p_{i,j'} Z_{jj'} &= 0,\ \text{for all } i\in\{0,\ldots,k-1\},\end{aligned}$$

with $Z_{jj'} = Y_j Y_{j'}\left(X_j^2+X_{j'}^2\right)$.

- Number of equations k
- Number of variables $\binom{mt}{2}$

Experiments [Binary case ($q = 2$) and $m = 14$]

t	3	4	5	6	7	8	9	10	11	12
N	861	1540	2415	3486	4753	6216	7875	9730	11781	14028
k	16342	16328	16314	16300	16286	16272	16258	16244	16230	16216
D_{random}	0	0	0	0	0	0	0	0	0	0
$D_{\text{alternant}}$	42	126	308	560	882	1274	1848	2520	3290	4158
D_{Goppa}	252	532	980	1554	2254	3080	4158	5390	6776	8316

t	13	14	15	16
N	16471	19110	21945	24976
k	16202	16188	16174	16160
D_{random}	269	2922	5771	8816
$D_{\text{alternant}}$	5124	6188	7350	8816
D_{Goppa}	10010	11858	13860	16016

- $N \stackrel{\text{def}}{=} \binom{mt}{2}$ the number of variables
- D_{random}, dimension of the vector space solution for a random code
- $D_{\text{alternant}}$, dimension of the vector space solution for a random alternant code of degree r
- D_{Goppa}, dimension of the vector space solution for a random Goppa code of degree r.

Experiments [Binary case ($q = 2$) and $m = 14$]

Rank of a Linearized McEliece system using a Goppa code vs
Rank of a Linearized McEliece system using a random code.

Bounds

Table: Smallest order t of a binary Goppa code of length $n = 2^m$ for which our distinguisher does not work.

m	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
$t_{\min}$	8	8	11	16	20	26	34	47	62	85	114	157	213	290	400

M. Finiasz, and N. Sendrier.
"Security Bounds for the Design of Code-Based Cryptosystems."
Asiacrypt'09.

Outline

Cleaning McEliece's Algebraic System

- For $j = 0$, linear equations involving only the variables of $\mathbf{Y}$:

$$\begin{cases} \vdots \\ g_{i,0} Y_0 + \ldots + g_{i,n-1} Y_{n-1} = 0, \quad i \in \{0, \ldots, k-1\}. \\ \vdots \end{cases}$$

- For quasi-cyclic/dyadic alternant codes, we have additional linear equations involving the variables of $\mathbf{Y}$ (resp. $\mathbf{X}$).

T. Berger, P.-L. Cayrel, P. Gaborit, A. Otmani.
"Reducing Key Length of the McEliece Cryptosystem". AFRICACRYPT 2009.

R. Misoczki, P. Barreto.
" Compact McEliece Keys from Goppa Codes". SAC 2009.

Cleaning McEliece's Algebraic System

- For $j = 0$, linear equations involving only the variables of $\mathbf{Y}$:

$$\begin{cases} \vdots \\ g_{i,0} Y_0 + \ldots + g_{i,n-1} Y_{n-1} = 0, \ \ i \in \{0, \ldots, k-1\}. \\ \vdots \end{cases}$$

- For quasi-cyclic/dyadic alternant codes, we have additional linear equations involving the variables of $\mathbf{Y}$ (resp. $\mathbf{X}$).

T. Berger, P.-L. Cayrel, P. Gaborit, A. Otmani.
"Reducing Key Length of the McEliece Cryptosystem".
AFRICACRYPT 2009.

R. Misoczki, P. Barreto.
" Compact McEliece Keys from Goppa Codes". SAC 2009.

BCGO Proposal (Africacrypt'09)

Assumption

Let $n = \ell n_0$ and let β be a *public* element of $\mathbb{F}_{q^m}$ of order ℓ.

- *Secret key.*
  - $(x_0, \ldots, x_{n_0-1})$ with $x_i \in \mathbb{F}_{q^m}$ such that $x_i \neq x_j$ if $i \neq j$
  - $(y_0, \ldots, y_{n_0-1})$ with $y_i \neq 0$ ($y_i \in \mathbb{F}_{q^m}$)
  - $e \in \{0, \ldots, \ell - 1\}$
- *Public key.* A basis $\boldsymbol{G}$ of $\mathrm{Ker}\Big(\boldsymbol{H}_t(\boldsymbol{x}, \boldsymbol{y})\Big) \cap \mathbb{F}_q^n$ with
  - $\boldsymbol{x} = (\overbrace{x_0, \beta x_0 \ldots, \beta^{\ell-1} x_0}^{\ell}, \ldots, \overbrace{x_{n_0-1}, \beta x_{n_0-1}, \ldots, \beta^{\ell-1} x_{n_0-1}}^{\ell})$
  - $\boldsymbol{y} =$
    $(\overbrace{y_0, \beta^e y_0, \ldots, \beta^{e(\ell-1)} y_0}^{\ell}, \ldots, \overbrace{y_{n_0-1}, \beta^e y_{n_0-1}, \ldots, \beta^{e(\ell-1)} y_{n_0-1}}^{\ell})$

BCGO Proposal (Africacrypt'09)

- We have the following linear relations for any $i \in \{0, \ldots, n_0 - 1\}$ and $j \in \{0, \ldots, \ell - 1\}$:

$$\begin{cases} x_{i\ell+j} = \beta^j x_{i\ell} \\ y_{i\ell+j} = \beta^{ej} y_{i\ell} \end{cases}$$

- The system is completely described by n_0 variables Y_i and n_0 variables X_i assuming that e is *known* ($0 \leq e \leq 100$)

MB Proposal (SAC'09)

- The public code is an alternant over $\mathbb{F}_q$ with $q = 2^s$ ($s \geq 1$) where for any $0 \leq j \leq n_0 - 1$ and $0 \leq i, i' \leq \ell - 1$, we have:

$$\begin{cases} y_{j\ell+i} & = & y_{j\ell} \\ x_{j\ell+i} + x_{j\ell} & = & x_i + x_0 \\ x_{j\ell+(i \oplus i')} & = & x_{j\ell+i} + x_{j\ell+i'} + x_{j\ell} \end{cases}$$

- For any $1 \leq i \leq \ell - 1$, if we write the binary decomposition of $i = \sum_{j=0}^{\log_2(\ell-1)} \eta_j 2^j$ then:

$$x_i = x_0 + \sum_{j=0}^{\log_2(\ell-1)} \eta_j (x_{2^j} + x_0).$$

- Hence, the system is described by n_0 variables Y_i and $n_0 + \log_2(\ell)$ variables X_i

Summary

We have equivalent secret-keys.

- some variables can be *fixed*.

Let n_Y (*resp.* n_X) be $\#\mathbf{Y}$ (*resp.* $\#\mathbf{X}$)

- $\mathrm{McE}_{n,k,t}(\boldsymbol{X}, \boldsymbol{Y})$. $n_Y = n - 1$ and $n_X = n - 3$ (one Y_i and three X_i's)
- **BCGO variant.** $n_Y = n_0 - 1$ and $n_X = n_0 - 1$ (one Y_i and one X_i)
- **MB variant.** $n_Y = n_0 - 1$ and $n_X = n_0 - 2 + \log_2(\ell)$ (one Y_i and two X_i's)

[First step – Cleaning.] Reduce the number of variables by removing all the linear equations involving the Y_i's (resp. X_j's)

$\Rightarrow$ Let d be the *remaining* variables in the block $\mathbf{Y}$.

Summary

We have equivalent secret-keys.

- some variables can be *fixed*.

Let n_Y (*resp.* n_X) be $\#\mathbf{Y}$ (*resp.* $\#\mathbf{X}$)

- McE$_{n,k,t}(\boldsymbol{X}, \boldsymbol{Y})$. $n_Y = n - 1$ and $n_X = n - 3$ (one Y_i and three X_i's)
- **BCGO variant.** $n_Y = n_0 - 1$ and $n_X = n_0 - 1$ (one Y_i and one X_i)
- **MB variant.** $n_Y = n_0 - 1$ and $n_X = n_0 - 2 + \log_2(\ell)$ (one Y_i and two X_i's)

[First step – Cleaning.] Reduce the number of variables by removing all the linear equations involving the Y_i's (resp. X_j's)

$\Rightarrow$ Let d be the *remaining* variables in the block $\mathbf{Y}$.

Outline

Solving the Algebraic System

- Naive approach by applying directly a generic Gröbner basis algorithm (Magma)
  - It fails for almost all challenges
  - But, one challenge A_{20} (AfricaCrypt '09) was broken in 24 hours of computation using a non negligible amount of memory
- How to exploit the particular structure of the system ?

Solving the Algebraic System

- Naive approach by applying directly a generic Gröbner basis algorithm (Magma)
  - It fails for almost all challenges
  - But, one challenge A_{20} (AfricaCrypt '09) was broken in 24 hours of computation using a non negligible amount of memory
- How to exploit the particular structure of the system ?

Computing a Gröbner Basis

- Buchberger's algorithm (1965)
- F_4/F_5 (J.-C. Faugère, 1999/2002)

$\Rightarrow$ For a zero-dimensional (i.e. finite number of solutions) system of n variables:

$$\mathcal{O}\left(n^{3 \cdot D_{reg}}\right),$$

D_{reg} being the maximum degree reached during the computation.

- Behavior on random systems of equations
  - D_{reg} is generically equal to $n+1$ (If $\#$eq.$=$ n).
  - $\#\mathcal{S}ol \leq \prod_{i=1}^{n} \mathrm{degree}_i$ (Bezout's bound)

Bi-Homogeneous Structure of $\mathrm{McE}_{n,k,t}(\boldsymbol{X}, \boldsymbol{Y})$

$\mathrm{McE}_{n,k,t}(\boldsymbol{X}, \boldsymbol{Y}) =$

$$\begin{cases} \vdots \\ g_{i,0} Y_0 X_0^j + \ldots + g_{i,n-1} Y_{n-1} X_{n-1}^j = 0 \text{ with } \begin{cases} i \in \{0, \ldots, k-1\} \\ j \in \{0, \ldots, t-1\} \end{cases} \\ \vdots \end{cases}$$

- The only monomials occurring are $Y_i X_i^j$

Definition

$f \in \mathbb{F}_{q^m}[\mathbf{X}, \mathbf{Y}]$ is *bi-homogeneous* of *bi-degree* (d_1, d_2) if:

$$\forall \alpha, \mu \in \mathbb{F}_{q^m},\ f(\alpha \mathbf{X}, \mu \mathbf{Y}) = \alpha^{d_1} \mu^{d_2} f(\mathbf{X}, \mathbf{Y}).$$

f is *bilinear* if it is *bi-homogeneous* of *bi-degree* $(1, 1)$.

Bi-Homogeneous Structure of $\mathrm{McE}_{n,k,t}(\boldsymbol{X}, \boldsymbol{Y})$

$\mathrm{McE}_{n,k,t}(\boldsymbol{X}, \boldsymbol{Y}) =$

$$\begin{cases} \vdots \\ g_{i,0} Y_0 X_0^j + \ldots + g_{i,n-1} Y_{n-1} X_{n-1}^j = 0 \text{ with } \begin{cases} i \in \{0, \ldots, k-1\} \\ j \in \{0, \ldots, t-1\} \end{cases} \\ \vdots \end{cases}$$

- Each block of k equations is *bi-homogeneous* of bi-degree $(1, j)$

Definition

$f \in \mathbb{F}_{q^m}[\mathbf{X}, \mathbf{Y}]$ is *bi-homogeneous* of *bi-degree* (d_1, d_2) if:

$$\forall \alpha, \mu \in \mathbb{F}_{q^m},\ f(\alpha \mathbf{X}, \mu \mathbf{Y}) = \alpha^{d_1} \mu^{d_2} f(\mathbf{X}, \mathbf{Y}).$$

f is *bilinear* if it is *bi-homogeneous* of *bi-degree* $(1, 1)$.

Complexity of Solving Bilinear Systems

J.-C. Faugère, M. Safey El Din, and P.-J. Spaenlehauer. *"Gröbner Bases of Bihomogeneous Ideals Generated by Polynomials of Bidegree (1,1): Algorithms and Complexity"*. *arXiv:1001.4004v1 [cs.SC]*, 2010.

- Dedicated version of F_5 for such systems (avoiding reductions to zeros/specific structure of the matrices)

Complexity of Bilinear System

The degree of regularity of a generic affine bilinear 0-dimensional system over $\mathbb{K}[X, Y]$ is upper bounded by

$$D_{\text{reg}} \leqslant \min(n_X, n_Y) + 1 \quad \text{[vs. } n_X + n_Y + 1 \text{ for a rand. system].}$$

Polynomial time complexity for computing the Gröbner basis if the min is constant.

Complexity of Solving Bilinear Systems

J.-C. Faugère, M. Safey El Din, and P.-J. Spaenlehauer.
"Gröbner Bases of Bihomogeneous Ideals Generated by Polynomials of Bidegree (1,1): Algorithms and Complexity".
arXiv:1001.4004v1 [cs.SC], 2010.

- Dedicated version of F_5 for such systems (avoiding reductions to zeros/specific structure of the matrices)

Complexity of Bilinear System

The degree of regularity of a generic affine bilinear 0-dimensional system over $\mathbb{K}[X, Y]$ is upper bounded by

$$D_{reg} \leqslant \min(n_X, n_Y) + 1 \quad \text{[vs. } n_X + n_Y + 1 \text{ for a rand. system]}.$$

Polynomial time complexity for computing the Gröbner basis if the min is constant.

Complexity of Solving Bilinear Systems

J.-C. Faugère, M. Safey El Din, and P.-J. Spaenlehauer. *"Gröbner Bases of Bihomogeneous Ideals Generated by Polynomials of Bidegree (1,1): Algorithms and Complexity"*. *arXiv:1001.4004v1 [cs.SC]*, 2010.

- Dedicated version of F_5 for such systems (avoiding reductions to zeros/specific structure of the matrices)

Complexity of Bilinear System

The degree of regularity of a generic affine bilinear 0-dimensional system over $\mathbb{K}[X, Y]$ is upper bounded by

$$D_{reg} \leqslant \min(n_X, n_Y) + 1 \text{ [vs. } n_X + n_Y + 1 \text{ for a rand. system]}.$$

Polynomial time complexity for computing the Gröbner basis if the min is constant.

Extracting a Bilinear Subsystem

- [Second step – Extracting a Bilinear Subsystem.] We keep only the exponents of X_i that are powers of 2:

$$\mathrm{biMcE}_{n,k,t}(\boldsymbol{X},\boldsymbol{Y}) \stackrel{\text{def}}{=} \begin{cases} \vdots \\ g_{i,0}\,Y_0X_0^{2^\ell} + \ldots + g_{i,n-1}\,Y_{n-1}X_{n-1}^{2^\ell} = 0 \\ \vdots \end{cases}$$

with $i \in \left\{0,\ldots,k-1\right\}$ and $\ell \in \left\{0,\ldots,\log_2(t-1)\right\}$.

  - The system is *"quasi"* bilinear, precisely bi-homogeneous of bi-degree $(1, 2^\ell)$ ($\mathrm{Char}(\mathbb{F}_q) = 2$)

Solving $\text{biMcE}_{n,k,t}(\boldsymbol{X}, \boldsymbol{Y})$

1. [First step – Cleaning.] Let d be the number of free variables in $\mathbf{Y}$.
2. [Second step – Extracting a Bilinear Subsystem.]

“Naive Approach"

- If d is very small then perform an exhaustive search in $\mathbb{F}_{q^m}$
- Solve the remaining linear system with the X_i's
- Time complexity $\mathcal{O}\left(q^{md}(mn_X)^3\right)$
- Challenge A_{20} (BCGO variant):
  - $q = 2^{10}, m = 2, d = 3 \longrightarrow \geqslant 2^{60}$ (here $2^{15.8}$)

Solving $\mathrm{biMcE}_{n,k,t}(\boldsymbol{X}, \boldsymbol{Y})$

1. [First step – Cleaning.] Let d be the number of free variables in **Y**.
2. [Second step – Extracting a Bilinear Subsystem.]

“Naive Approach"

- If d is very small then perform an exhaustive search in $\mathbb{F}_{q^m}$
- Solve the remaining linear system with the X_i's
- Time complexity $\mathcal{O}\left(q^{md}(mn_X)^3\right)$
- Challenge A_{20} (BCGO variant):
  - $q = 2^{10}, m = 2, d = 3 \longrightarrow \geqslant 2^{60}$ (here $2^{15.8}$)

Complexity of Solving $\mathrm{biMcE}_{n,k,t}(\boldsymbol{X}, \boldsymbol{Y})$

$\mathrm{biMcE}_{n,k,t}(\boldsymbol{X}, \boldsymbol{Y})$

Let d be the number of free variables in **Y**.

- For $\mathrm{biMcE}_{n,k,t}(\boldsymbol{X}, \boldsymbol{Y})$, it holds that $\mathrm{D}_{\mathrm{reg}} \leqslant d + 1$.
- Computing a Gröbner basis of $\mathrm{biMcE}_{n,k,t}(\boldsymbol{X}, \boldsymbol{Y})$ can be done with a tweaked version of F_5 in:

$$\mathcal{O}\left(n_X^{\omega(d+1)}\right),$$

$2 \leq \omega \leq 3$ being the linear algebra constant.

Practical results – BCGO Variant

	q	ℓ	n_0	d	Sec. (log)	n_X	#Eq	Time (Op., M.)	T_{theo}
A_{16}	2^8	51	9	3	80	8	510	0.06 sec ($2^{18.9}$ op, 115 Meg)	2^{17}
B_{16}	2^8	51	10	3	90	9	612	0.03 sec ($2^{17.1}$ op, 116 Meg)	2^{18}
C_{16}	2^8	51	12	3	100	11	816	0.05 sec ($2^{16.2}$ op, 116 Meg)	2^{20}
D_{16}	2^8	51	15	4	120	14	1275	0.02 sec ($2^{14.7}$ op, 113 Meg)	2^{26}
A_{20}	2^{10}	75	6	2	80	5	337	0.05 sec ($2^{15.8}$ op, 115 Meg)	2^{10}
B_{20}	2^{10}	93	6	2	90	5	418	0.05 sec ($2^{17.1}$ op, 115 Meg)	2^{10}
C_{20}	2^{10}	93	8	2	110	7	697	0.02 sec ($2^{14.5}$ op, 115 Meg)	2^{11}
QC_{600}	2^8	255	15	3	600	14	6820	0.08 sec ($2^{16.6}$ op, 116 Meg)	2^{21}

- The solutions always belong to $\mathbb{F}_{q^m}$ with $m = 2$ (BCGO constraint)
- We also proposed the parameter QC_{600} to show the influence of d

Practical Results – MB Variant

	q	d	ℓ	n_0	Sec. (log)	n_X	#Equ	Time (Op., Me.)	T_{theo}
T. 2	2^2	7	64	56	128	59	193, 584	1, 776.3 sec ($2^{34.2}$ op, 360 Meg)	2^{65}
T. 2	2^4	3	64	32	128	36	112, 924	0.50 sec ($2^{22.1}$ op, 118M)	2^{29}
T. 2	2^8	1	64	12	128	16	40, 330	0.03 sec ($2^{16.7}$ op, 35M)	2^8
T. 3	2^8	1	64	10	102	14	32, 264	0.03 sec ($2^{15.9}$ op, 113M.)	2^8
T. 3	2^8	1	128	6	136	11	65, 028	0.02 sec ($2^{15.4}$ op, 113 M.)	2^7
T. 3	2^8	1	256	4	168	10	130, 562	0.11 sec ($2^{19.2}$ op, 113M.)	2^7
T. 5	2^8	1	128	4	80	9	32, 514	0.06 sec ($2^{17.7}$ op, 35M.)	2^6
T. 5	2^8	1	128	5	112	10	48, 771	0.02 sec ($2^{14.5}$ op, 35M.)	2^7
T. 5	2^8	1	128	6	128	11	65, 028	0.01 sec ($2^{16.6}$ op, 35 M.)	2^7
T. 5	2^8	1	256	5	192	11	195, 843	0.05 sec ($2^{17.5}$ op, 35M.)	2^7
T. 5	2^8	1	256	6	256	12	261, 124	0.06 sec ($2^{17.8}$ op, 35M.)	2^7
D_{256}	2^4	3	128	32	256	37	455, 196	7.1 sec ($2^{26.1}$ op, 131M.)	2^{29}
D_{512}	2^8	1	512	6	512	13	1, 046, 532	0.15 sec ($2^{19.7}$ op, 38M.)	2^8

- Binary challenges are not solved (work in progress)
- We proposed the challenges D_{256} and D_{512}

Conclusion

McEliece scheme is a *challenging* public key cryptosystem

- Little is known about key-recovery attacks
- We introduced an algebraic framework for tackling this issue focusing on a bilinear subsystem

This approach gave successful results for variants with compact keys

- The proposed parameters were *too optimistic* (key should be larger)
- An *unbalanced number* of variables does not improve the security

Conclusion

J.-C. Faugère, V. Gauthier, A. Otmani, L. Perret and J-P. Tillich.
“A Distinguisher for High Rate McEliece Cryptosystems". ITW’11.

- Explain the defect of Rank
- Formalize the advantage (prob. of success)

L. Dallot.
“Towards a concrete security proof of Courtois, Finiasz and Sendrier signature scheme." WeWorc’07.

- ALGEBRAIC TECHNIQUES vs QUANTUM ?

  H. Dinh, C. Moore, and A. Russell.
  “The McEliece Cryptosystem Resists Quantum Fourier Sampling Attacks."