EUROCRYPT 2010

29th Annual International Conference on Cryptology – Monaco (France)

Algebraic Cryptanalysis of McEliece Variants with Compact Keys

Jean-Charles Faugère[2], Ayoub Otmani[1,3], Ludovic Perret[2], Jean-Pierre Tillich[3]

[1] GREYC, Université de Caen - Ensicaen

[2] SALSA Project - INRIA (Centre Paris-Rocquencourt) UPMC, Univ Paris 06 - CNRS, UMR 7606, LIP6

[3] SECRET Project - INRIA (Centre Paris-Rocquencourt)

Introduction

▷ **Our contribution**

- Key-recovery attacks against McEliece cryptosystem $\Longleftrightarrow$ Solving a highly structured polynomial system
- The associated systems for two McEliece variants with **very compact** keys proposed by Berger-Cayrel-Gaborit-Otmani (2009) and Misoczki-Barreto (2009) have **few** variables and **many** linear equations
- This leads to a **practical key recovery algebraic attacks** against these two schemes

Introduction

▷ **Our contribution**

- Key-recovery attacks against McEliece cryptosystem $\Longleftrightarrow$ Solving a highly structured polynomial system
- The associated systems for two McEliece variants with **very compact** keys proposed by Berger-Cayrel-Gaborit-Otmani (2009) and Misoczki-Barreto (2009) have **few** variables and **many** linear equations
- This leads to a **practical key recovery algebraic attacks** against these two schemes

▷ An independent work by Gauthier Umana – Leander also proposes an attack **practical for some parameters** (to appear at SCC 2010)

Definitions

▷ $\mathscr{C}$ is a *linear code* over $\mathbb{F}_q$ of length n and dimension k if $\mathscr{C}$ is k-dimensional vector subspace of $\mathbb{F}_q^n$

▷ **Decoding** a code $\mathscr{C}$ consists in solving the **Closest Vector Problem** for the Hamming metric (can be regarded as an analogue of CVP in lattices)

Input. $\mathscr{C}$ is a linear code $\subset \mathbb{F}_q^n$ and $\boldsymbol{y}$ in $\mathbb{F}_q^n$

Output. Find in $\mathscr{C}$ the closest vector to $\boldsymbol{y}$

Algorithmic Issues

▷ Decoding a **random** linear code

- Proved NP-Hard by Berlekamp - McEliece - van Tilborg in '78
- Best practical algorithms are based on *Information Set Decoding*
  - Probabilistic exhaustive search for a codeword inside a ball of radius t
  - Time complexity is $\simeq 2^{\textbf{constant}\, n(1+o(1))}$ (assuming that both t/n and k/n are constant)

Algorithmic Issues

▷ Decoding a **random** linear code

- Proved NP-Hard by Berlekamp - McEliece - van Tilborg in '78
- Best practical algorithms are based on *Information Set Decoding*
  - Probabilistic exhaustive search for a codeword inside a ball of radius t
  - Time complexity is $\simeq 2^{\textsf{constant}\, n(1+o(1))}$ (assuming that both t/n and k/n are constant)

▷ **But structured** codes can be decoded in **polynomial** time...

Alternant Codes

- ▷ Consider two fields $\mathbb{F}_q$ and $\mathbb{F}_{q^m}$ with $q = 2^s$ $(s \geq 1)$ and $m \geq 1$
- ▷ $\boldsymbol{x} = (x_1, \ldots, x_n) \in \mathbb{F}_{q^m}^n$ with $x_i \neq x_j$ if $i \neq j$
- ▷ $\boldsymbol{y} = (y_1, \ldots, y_n) \in \mathbb{F}_{q^m}^n$ with $y_i \neq 0$
- ▷ For any $t < n$, let $\boldsymbol{H}_t(\boldsymbol{x}, \boldsymbol{y}) \stackrel{\text{def}}{=} \begin{pmatrix} y_1 & y_2 & \cdots & y_n \\ y_1 x_1 & y_2 x_2 & \cdots & y_n x_n \\ \vdots & \vdots & & \vdots \\ y_1 x_1^{t-1} & y_2 x_2^{t-1} & \cdots & y_n x_n^{t-1} \end{pmatrix}$

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

$$\boldsymbol{v} \in \mathscr{C} \quad \Longleftrightarrow \quad \boldsymbol{v} \in \mathbb{F}_q^n \ \textit{and} \ \boldsymbol{H}_t(\boldsymbol{x}, \boldsymbol{y})\, \boldsymbol{v}^T = \boldsymbol{0}$$

Proposition. Alternant codes can be decoded in **polynomial time** up to $t/2$ errors as long as $\boldsymbol{x}$ and $\boldsymbol{y}$ are **known**

McEliece Cryptosystem

▷ One of the **oldest** public-key cryptosystems (R.J. McEliece in 1978)

▷ Alternative system based on coding theory

▷ Principle is to **mask a structured code** in such a way that it **looks like random**

- Trapdoor $= \boldsymbol{H}_t(\boldsymbol{x}, \boldsymbol{y})$
- Public key $=$ Random basis $\boldsymbol{G}$ of Ker$\Big(\boldsymbol{H}_t(\boldsymbol{x}, \boldsymbol{y})\Big) \cap \mathbb{F}_q^n$

Algebraic Cryptanalysis of McEliece PKC

▷ What we have: $\boldsymbol{G} = (g_{i,j})$ is the public matrix

▷ What is known: rows of $\boldsymbol{G}$ belong to the kernel of $\boldsymbol{H}_t(\boldsymbol{x}, \boldsymbol{y})$

$\Longrightarrow$ The **secret vectors** $\boldsymbol{x}$ and $\boldsymbol{y}$ have to satisfy $\boldsymbol{H}_t(\boldsymbol{X}, \boldsymbol{Y})\,\boldsymbol{G}^T = \boldsymbol{0}$

$$\begin{pmatrix} Y_1 & Y_2 & \cdots & Y_n \\ Y_1X_1 & Y_2X_2 & \cdots & Y_nX_n \\ \vdots & \vdots & & \vdots \\ Y_1X_1^{t-1} & Y_2X_2^{t-1} & \cdots & Y_nX_n^{t-1} \end{pmatrix} \boldsymbol{G}^T = \boldsymbol{0}$$

Algebraic Cryptanalysis of McEliece PKC

Definition. The *McEliece algebraic system* is the set of equations defined by

$$\mathsf{McE}_{n,k,t}(\boldsymbol{X},\boldsymbol{Y}) \stackrel{\text{def}}{=} \begin{cases} g_{1,0}Y_0 + \cdots + g_{1,n-1}Y_{n-1} = 0 \\ \vdots \\ g_{k,0}Y_0 + \cdots + g_{k,n-1}Y_{n-1} = 0 \\ \vdots \\ g_{i,0}Y_0X_0^j + \cdots + g_{i,n-1}Y_{n-1}X_{n-1}^j = 0 \text{ with } \begin{cases} i \in \{0,\ldots,k-1\} \\ j \in \{0,\ldots,t-1\} \end{cases} \\ \vdots \end{cases}$$

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

Algebraic Cryptanalysis of McEliece PKC

Definition. The *McEliece algebraic system* is the set of equations defined by

$$\mathrm{McE}_{n,k,t}(\boldsymbol{X},\boldsymbol{Y}) \stackrel{\text{def}}{=} \begin{cases} g_{1,0}Y_0 + \cdots + g_{1,n-1}Y_{n-1} = 0 \\ \vdots \\ g_{k,0}Y_0 + \cdots + g_{k,n-1}Y_{n-1} = 0 \\ \vdots \\ g_{i,0}Y_0X_0^j + \cdots + g_{i,n-1}Y_{n-1}X_{n-1}^j = 0 \text{ with } \begin{cases} i \in \{0,\ldots,k-1\} \\ j \in \{0,\ldots,t-1\} \end{cases} \\ \vdots \end{cases}$$

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

Example. McEliece proposed in 1978 $q = 2$, $m = 10$, $n = 1024$, $t = 50 \Rightarrow k \geqslant 524$

$\Rightarrow$ Public key has 250Kbits (60-bit security)

Variants with Compact Keys

▷ McEliece cryptosystem suffers from the key-size problem

▷ Several attempts have been made to solve this problem by taking **structured compact** matrices

- **Quasi-cyclic.** Gaborit 2005 (**insecure**), Baldi-Chiaraluce 2007 (**insecure**) Baldi-Chiaraluce 2008, Berger-Cayrel-Gaborit-Otmani (BCGO) 2009

- **Quasi-dyadic.** Misoczki-Barreto (MB) 2009

BCGO Proposal

Definition. Assume that $n = \ell n_0$ 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 $x_i \in \mathbb{F}_{q^m}$ and $x_i \neq x_j$ 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 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

More formally, we obtain the following linear relations for any $i \in \{0, \ldots, n_0 - 1\}$ and $j \in \{0, \ldots, \ell - 1\}$:

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

Corollary. 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

Proposition. 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:

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

Corollary.

▷ 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

Reducing the Number of Variables

Proposition. Some variables can be **fixed** so that the number of unknowns can be reduced to n_Y (*resp.* n_X) unknowns Y_i (*resp.* X_i) where

- 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)

Solving the Algebraic System

1. 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

Solving the Algebraic System

1. 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

2. A natural approach that exploits the particular structure of the system:
   - Linear equations involving only the variables Y_i
   - **Many quadratic** equations (in $\mathbb{F}_q$) involving $Y_i X_j^{2^l}$ with very **few** unknowns

Extracting a Bilinear Subsystem

▷ Keeping only the exponents of X_i that are powers of 2:

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

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

▷ Reducing the number of variables by removing all the linear equations involving the Y_j's

$\Rightarrow$ Let d be the **remaining** degree of freedom of the Y_i's

Solving $\mathsf{biMcE}_{n,k,t}(\boldsymbol{X}, \boldsymbol{Y})$ – 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 $O\left(q^{md}(mn_X)^3\right)$

Example.

Challenge A_{20} (BCGO variant): $q = 2^{10}, m = 2, d = 3 \longrightarrow \geqslant 2^{60}$

Solving $\mathsf{biMcE}_{n,k,t}(\boldsymbol{X}, \boldsymbol{Y})$ **– 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 $O\left(q^{md}(mn_X)^3\right)$

Example.

Challenge A_{20} (BCGO variant): $q = 2^{10}, m = 2, d = 3 \longrightarrow \geqslant 2^{60} \longrightarrow 2^{15.8}$

Complexity of Gröbner Basis

Proposition. The time complexity of the F_5 algorithm grevlex Gröbner basis for a system of N variables is

$$O\left(N^{3\mathsf{d}_{\mathsf{reg}}}\right)$$

where $\mathsf{d}_{\mathsf{reg}}$ is the **degree of regularity**

Complexity of Gröbner Basis

Proposition. The time complexity of the F_5 algorithm grevlex Gröbner basis for a system of N variables is

$$O\left(N^{3\mathsf{d}_{\mathsf{reg}}}\right)$$

where $\mathsf{d}_{\mathsf{reg}}$ is the **degree of regularity**

Proposition. ([FSS, Theorem 6.1]) For the grevlex ordering, the degree of regularity of a generic affine **bilinear 0-dimensional** system over $\mathbb{K}[X, Y]$ is upper bounded by

$$\mathsf{d}_{\mathsf{reg}} \leqslant \min\left(n_Y, n_X\right) + 1$$

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.

Complexity of Gröbner Basis

Recall that our system has a particular structure

▷ The only monomials occurring are $Y_i X_j^l$

▷ Each block of k equations is **bi-homogeneous** *i.e.* the degrees of the variables of $\boldsymbol{X}$ (resp. $\boldsymbol{Y}$) are the same

Corollary. In all the considered cases,

▷ $\mathsf{d}_{\mathsf{reg}} = d + 1$ and hence the time complexity is roughly $O\left(n_X^{3(d+1)}\right)$

▷ In particular the attack is polynomial when d is a **constant**

Experimental Results

▷ We used a **dedicated** F_5 algorithm that has been implemented in C language in the FGb software for computing the first Gröbner basis

▷ Experimental results have been obtained with several Xeon bi-processor 3.2 Ghz with 16 GBytes of RAM

▷ Instances have been generated using the Magma software (version 2.15)

▷ In practice the most difficult task is to generate the algebraic equations

Practical results – BCGO Variant

Challenge	q	ℓ	n_0	d	Security	Variables	Equations	Time (Operations, Memory)
A_{16}	2^8	51	9	3	80	16	510	0.06 sec ($2^{18.9}$ op, 115 Meg)
B_{16}	2^8	51	10	3	90	18	612	0.03 sec ($2^{17.1}$ op, 116 Meg)
C_{16}	2^8	51	12	3	100	22	816	0.05 sec ($2^{16.2}$ op, 116 Meg)
D_{16}	2^8	51	15	4	120	28	1275	0.02 sec ($2^{14.7}$ op, 113 Meg)
A_{20}	2^{10}	75	6	2	80	10	337	0.05 sec ($2^{15.8}$ op, 115 Meg)
B_{20}	2^{10}	93	6	2	90	10	418	0.05 sec ($2^{17.1}$ op, 115 Meg)
C_{20}	2^{10}	93	8	2	110	14	697	0.02 sec ($2^{14.5}$ op, 115 Meg)
QC_{600}	2^8	255	15	3	600	28	6820	0.08 sec ($2^{16.6}$ op, 116 Meg)

Remark.

▷ 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

Challenge	q	d	ℓ	n_0	Security	Variables	Time (Operations, Memory)
Table 2	2^2	7	64	56	128	115	$1,776.3$ sec ($2^{34.2}$ op, 360 Meg)
Table 2	2^4	3	64	32	128	67	0.50 sec ($2^{22.1}$ op, 118 Meg)
Table 2	2^8	1	64	12	128	27	0.03 sec ($2^{16.7}$ op, 35 Meg)
Table 3	2^8	1	64	10	102	23	0.03 sec ($2^{15.9}$ op, 113 Meg)
Table 3	2^8	1	128	6	136	16	0.02 sec ($2^{15.4}$ op, 113 Meg)
Table 3	2^8	1	256	4	168	13	0.11 sec ($2^{19.2}$ op, 113 Meg)
Table 5	2^8	1	128	4	80	12	0.06 sec ($2^{17.7}$ op, 35 Meg)
Table 5	2^8	1	128	5	112	14	0.02 sec ($2^{14.5}$ op, 35 Meg)
Table 5	2^8	1	128	6	128	16	0.01 sec ($2^{16.6}$ op, 35 Meg)
Table 5	2^8	1	256	5	192	15	0.05 sec ($2^{17.5}$ op, 35 Meg)
Table 5	2^8	1	256	6	256	17	0.06 sec ($2^{17.8}$ op, 35 Meg)
Dya$_{256}$	2^4	3	128	32	256	68	7.1 sec ($2^{26.1}$ op, 131 Meg)
Dya$_{512}$	2^8	1	512	6	512	18	0.15 sec ($2^{19.7}$ op, 38 Meg)

Remark. Binary challenges are not solved (work in progress)

Conclusions

▷ McEliece scheme is a **challenging** public key cryptosystem

- Little is known about key recovery attacks
- We introduced an algebraic framework for tackling this issue
- We focused 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

▷ A variation of this approach gives a way of **distinguishing** a public key from a random matrix for some types of McEliece keys

Jean-Charles Faugère, Ayoub Otmani, Ludovic Perret, Jean-Pierre Tillich, A Distinguisher for High Rate McEliece Cryptosystems, *preprint*.

Open Questions

▷ Sharpen the complexity bounds by taking into account the over-determination of the system

▷ Improve the solving for larger values of d

▷ How far this attack can be pushed to recover the private key of a McEliece cryptosystem?