Cryptanalysis of a McEliece Cryptosystem Based on QC-LDPC Codes

Ayoub Otmani [1]

Ayoub.Otmani@info.unicaen.fr

Léonard Dallot [1]

Leonard.Dallot@info.unicaen.fr

Jean-Pierre Tillich [2]

jean-pierre.tillich@inria.fr

[1] GREYC - Groupe de Recherche en Informatique, Image, Automatique et Instrumentation de Caen (UMR 6072)

[2] Projet Secret, INRIA-Rocquencourt

Journées Codage et Cryptographie. March 17, 2008.

I. Background

Introduction

- **Asymmetric cryptography concepts** introduced by Diffie & Hellman ('76)

- Rivest, Shamir & Adleman invented RSA ('77)
  - **First** asymmetric cryptosystem
  - Widely accepted for practical uses

- But, **alternative** cryptosystems exist ... such as McEliece cryptosystem

McEliece Cryptosystem

- Let $\mathfrak{F}_{n,k,t}$ be a family of **Goppa** codes of length n and dimension k **capable to correct** $\leq t$ errors.
- Cryptosystem described by **three** algorithms:

1. $(PK, SK) \longleftarrow \texttt{Setup}(\mathbf{1}^\lambda)$
2. $\mathbf{c} \in \mathbb{F}_2^n \longleftarrow \texttt{Encrypt}(\mathbf{m} \in \mathbb{F}_2^k)$
3. $\mathbf{m}' \in \mathbb{F}_2^k \longleftarrow \texttt{Decrypt}(\mathbf{c}' \in \mathbb{F}_2^n)$

McEliece.$\texttt{Setup}$

$(PK, SK) \leftarrow \texttt{Setup}(\mathbf{1}^\lambda)$

1. Take n, k, t according to λ
2. *Randomly* choose a *generator matrix* $G' \in \mathfrak{F}_{n,k,t}$
3. *Randomly* pick:
   - $n \times n$ *permutation* matrix P
   - $k \times k$ *invertible* matrix S
4. Set $G = S \times G' \times P$ and $\gamma : \mathbb{F}_2^n \mapsto \mathbb{F}_2^k$ as the decoding algorithm associated with G'
5. Output

$$PK = (G, t) \quad \text{and} \quad SK = (S, P, \gamma)$$

McEliece.$\texttt{Encrypt}$

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

1. Pick a *random* vector $\mathbf{e} \in \mathbb{F}_2^n$ of *weight* $\leq t$
2. Output $\mathbf{c} = \mathbf{m} \times G \oplus \mathbf{e}$

McEliece.`Decrypt`

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

1. Calculate $\mathbf{z} = \mathbf{c}' \times P^{-1}$ $\quad // \ \mathbf{z} = \mathbf{m} \times (S \times G') \oplus (\mathbf{e} \times P^{-1})$
2. Compute $\mathbf{y} = \gamma(\mathbf{z})$ $\quad // \ \mathbf{y} = \mathbf{m} \times S$
3. Output $\mathbf{m}' = \mathbf{y} \times S^{-1}$ $\quad // \ \mathbf{m}' = \mathbf{m}$

McEliece Cryptosystem – Security Assumptions

- **One-Wayness under Chosen Plaintext Attack (OW-CPA)**

  Difficult to invert `Encrypt` (*decoding attack*)

- **Unmasking hardness**

  Difficult to extract secret matrices and a decoding algorithm from the public matrix (*structural attack*)

McEliece Cryptosystem Security – OW-CPA

1. Decoding **random** linear codes is **NP-Hard**

   E. R. Berlekamp, R. J. McEliece, and H. C. A. van Tilborg. **On the intractability of certain coding problems**. *IEEE Transactions on Information Theory*, 24(3):384–386, 1978.

2. **Best practical** algorithms operate **exponentially** with the length and the rate

   A. Canteaut and F. Chabaud. **A new algorithm for finding minimum-weight words in a linear code: Application to McEliece's cryptosystem and to narrow-sense BCH codes of length 511**. *IEEE Transactions on Information Theory*, 44(1):367–378, 1998.

McEliece Cryptosystem – Unmasking Hardness

Two basic attacks

1. Enumerate **all** permutation matrices until a generator matrix of a Goppa code is found

2. Enumerate **all** generator matrices of Goppa codes **until a permutation equivalent matrix** to the public matrix is found

McEliece Cryptosystem – Unmasking Hardness

Security recommendations

- $\mathfrak{F}_{n,k,t}$ and the Symmetric group **must** have huge sizes
- *Problem of code equivalence* solved in practise by *Support Splitting Algorithm*
  - N. Sendrier. **Finding the permutation between equivalent codes: the support splitting algorithm**. IEEE Transactions on Information Theory, vol. 46, no. 4, pages 1193-1203, July 2000.
  - Time complexity **increases** with the dimension of the Hull
  - Codes **should** have a big Hull

Insecure McEliece Cryptosystem Variants

- Reed-Solomon codes

  V.M. Sidelnikov and S.O. Shestakov. **On the insecurity of cryptosystems based on generalized Reed-Solomon codes**. *Discrete Mathematics and Applications*, 1(4):439–444, 1992.

- Concatenated codes

  N. Sendrier. **On the Structure of Randomly Permuted Concatenated Code.** Rapport de recherche de l'INRIA - Rocquencourt. Janvier 1995

- Reed-Muller codes.

  L. Minder and A. Shokrollahi. **Cryptanalysis of the Sidelnikov cryptosystem**. In *Eurocrypt 2007*, volume 4515 of *Lecture Notes in Computer Science*, pages 347–360, Barcelona, Spain, 2007.

Remark.

Original McEliece scheme **is still unbroken**...

McEliece Cryptosystem

- **Three advantages**
  - Fast encryption/decryption algorithms
  - Original scheme still secure
  - Alternative solution to RSA for quantum computers!

- **Main drawback: huge public key**

  For instance, parameters proposed in '78 (now outdated)

  * Goppa codes with $n = 1024$, $k = 524$
  * Private key $\simeq$ 300 Kbits
  * Public key $\simeq$ 500 Kbits

Reducing Key Sizes

1. Sparse matrices

A. Shokrollahi C. Monico, J. Rosenthal. **Using low density parity check codes in the McEliece cryptosystem**. In *IEEE International Symposium on Information Theory (ISIT 2000)*, page 215, Sorrento, Italy, 2000.

2. Quasi-cyclic matrices

P. Gaborit. **Shorter keys for code based cryptography**. In *Proceedings of the 2005 International Workshop on Coding and Cryptography (WCC 2005)*, pages 81–91, Bergen, Norway, March 2005.

3. Sparse quasi-cyclic matrices

M. Baldi, G. F. Chiaraluce. **Cryptanalysis of a new instance of McEliece cryptosystem based on QC-LDPC codes**. In *IEEE International Symposium on Information Theory*, pages 2591–2595, Nice, France, March 2007.

II. Cryptanalysis of a McEliece Cryptosystem Based on Quasi-Cyclic LDPC Codes

Low Density Parity Check Codes

Some facts.

- Invented by Gallager ('68) and rediscovered by Mackay ('98)
- Linear codes defined by very **sparse** parity check matrices
- Iteratively decoded through Belief Propagation algorithm
- For any **cryptographic use**, one has to **hide the sparsity** of matrices

Notation.

$\mathfrak{L}_{n,k,t}$: family of LDPC codes of length n, dimension k and correcting capability of t errors.

Circulant Matrix

Definition.

- M is a *circulant* $p \times p$ matrix if

$$M = \begin{pmatrix} m_0 & m_1 & \cdots & m_{p-1} \\ m_{p-1} & m_0 & \cdots & m_{p-2} \\ \vdots & \vdots & \ddots & \vdots \\ m_1 & m_2 & \cdots & m_0 \end{pmatrix}$$

- *Weight* of M is the weight of $\mathbf{m} = (m_0, \ldots, m_{p-1})$

Notation.

$$M \quad \longmapsto \quad \mathbf{m}(x) = m_0 + m_1 x + \cdots m_{p-1} x^{p-1}$$

Circulant Matrix

Properties. Let M and N be circulant $p \times p$ matrices

- $M + N$ is circulant

$$M + N \longmapsto \mathbf{m}(x) + \mathbf{n}(x)$$

- $M \times N$ is circulant

$$M \times N \longmapsto \mathbf{m}(x) \cdot \mathbf{n}(x) \mod (x^p - 1)$$

- M^T is circulant

$$M^T \longmapsto \mathbf{m}(\frac{1}{x}) \cdot x^p \mod (x^p - 1)$$

- M is **invertible** iff $\mathbf{m}(x)$ is **coprime** with $x^p - 1$

Circulant-by-Block Matrix

Definition. $M = [M_{i,j}]$ is *circulant-by-block* if $M_{i,j}$ is a circulant $p \times p$ matrix

$$M \longmapsto \mathbf{M}(x) = [\mathbf{m}_{i,j}(x)]$$

Properties. Let M and N be circulant-by-block matrices

- $M + N$, $M \times N$, M^T are also circulant-by-block matrices
- M is invertible iff $\det(\mathbf{M})(x)$ is coprime with $(x^p - 1)$
- M^{-1} is a circulant-by-block matrix

Quasi-Cylic Codes

- Let $n = pn_0$ and $r = pr_0$ with p, n_0 and r_0 positive integers
- Let H be an $r \times n$ *parity check matrix* of a code $\mathscr{C}$

Definition.

$\mathscr{C}$ is **quasi-cyclic** if $H = [H_{i,j}]$ with each $H_{i,j}$ is a *circulant* $p \times p$ matrix

$\mathscr{C}$ is a **quasi-cyclic low density parity check** code if each $H_{i,j}$ is *sparse*

McEliece Cryptosystem Based on Quasi-Cyclic LDPC Codes ('07)

Description.

- Assume $r_0 = 1$
- Let $\mathscr{C}$ be a QC-LDPC code defined by

$$H = [H_1 \ \cdots \ H_{n_0}]$$

where H_i is a *sparse circulant* $p \times p$ matrix of *column weight* d_v

- $\mathscr{C}$ is *able to decode* up to t' errors
- H_{n_0} has *full rank* and *dimension* of $\mathscr{C}$ is $k = p(n_0 - 1)$

McEliece Cryptosystem Based on Quasi-Cyclic LDPC Codes

Setup(1^λ)

1. Choose integers s, m such that $m << p$ and $t = t'/m$
2. Randomly pick *invertible* matrix
   – $S = [S_{i,j}]$ where $S_{i,j}$ is *sparse circulant* $p \times p$ matrix of *weight* s
   – $Q = [Q_{i,j}]$ where $Q_{i,j}$ is *sparse circulant* $p \times p$ matrix of *weight* m
3. Calculate a generator matrix G in *row reduced echelon form* from H
4. Compute $G' = S^{-1} \times G \times Q^{-1}$
5. Output $PK = (G', t)$ and $SK = (S, H, Q)$

McEliece Cryptosystem Based on Quasi-Cyclic LDPC Codes

`Encrypt(x)`

1. Randomly choose an error $\mathbf{e} \in \mathbb{F}_2^n$ of *weight* t
2. Calculate $\mathbf{y} = \mathbf{x} \cdot G' \oplus \mathbf{e}$

`Decrypt(y)`

1. Calculate $\mathbf{z} = \mathbf{y} \cdot Q$ $\quad // \; \mathbf{z} = (\mathbf{x} \cdot S^{-1} \times G) \oplus \mathbf{e} \cdot Q$
2. Decode $\mathbf{z}$ into $\mathbf{x}'$ $\quad // \; \mathbf{x}' = \mathbf{x} \cdot S^{-1}$
3. Output $\mathbf{x}' \cdot S$

Remark.

$\mathbf{e}' = \mathbf{e} \cdot Q$ is of weight $\leq mt = t'$

McEliece Cryptosystem Based on Quasi-Cyclic LDPC Codes

Proposed parameters.

- Q is chosen in *diagonal form*

$$Q = \begin{pmatrix} Q_1 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & Q_{n_0} \end{pmatrix}$$

- Q_i's are *invertible*

McEliece Cryptosystem Based on Quasi-Cyclic LDPC Codes

Suggested values.

- $n_0 = 4$, $p = 4032$, $d_v = 13$, $t' = 190$ and $t = 27$
- $s = m = 190/27 = 7$

Key sizes.

- Public Key: 48400 bits
- Secret Key: 1716 bits

Cryptosystem Analysis

Preliminaries.

- Since $H = [H_1 \ \cdots \ H_{n_0}]$ with H_{n_0} *invertible*

$$G = \left(\begin{array}{c|c} & (H_{n_0}^{-1} H_1)^T \\ I_k & \vdots \\ & (H_{n_0}^{-1} H_{n_0-1})^T \end{array} \right)$$

- This implies that k **first columns of public matrix** G' is equal to

$$G'_{\leq k} = S^{-1} \times \begin{pmatrix} Q_1^{-1} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & Q_{n_0-1}^{-1} \end{pmatrix}$$

Cryptosystem Analysis

Or, equivalently by **inverting** $G_{\leq k}$ and **adopting a polynomial approach**

$$\left(\mathbf{G}'_{\leq k}\right)^{-1}(x) = \begin{pmatrix} \mathbf{q}_1(x) \cdot \mathbf{s}_{1,1}(x) & \cdots & \mathbf{q}_1(x) \cdot \mathbf{s}_{1,n_0-1}(x) \\ \vdots & & \vdots \\ \mathbf{q}_i(x) \cdot \mathbf{s}_{i,1}(x) & \cdots & \mathbf{q}_i(x) \cdot \mathbf{s}_{i,n_0-1}(x) \\ \vdots & & \vdots \\ \mathbf{q}_{n_0-1}(x) \cdot \mathbf{s}_{n_0-1,1}(x) & \cdots & \mathbf{q}_{n_0-1}(x) \cdot \mathbf{s}_{n_0-1,n_0-1}(x) \end{pmatrix}$$

where $\mathbf{q}_i(x)$ and $\mathbf{s}_{i,j}(x)$ are **sparse polynomials**

Cryptosystem Analysis

Cryptanalysis principle

Given $\mathbf{g}(x)$ *of degree* $< p$, find $\mathbf{q}(x)$ *and* $\mathbf{s}(x)$ *of weight* $m << p$ such that

$$\mathbf{g}(x) \quad = \quad \mathbf{q}(x) \cdot \mathbf{s}(x) \quad \bmod (x^p - 1)$$

Remark.

With high probability (≥ 0.94), there exists ℓ such that

$$\left(x^\ell \cdot \mathbf{q}(x)\right) \cap \mathbf{g}(x) = x^\ell \cdot \mathbf{q}(x)$$

Cryptanalysis - First Strategy

1. Enumerate all the m-tuples $(e_1, \ldots, e_m)$ of the support of $\mathbf{g}(x)$
2. Calculate $\mathbf{q}(x) = x^{e_1} + \cdots + x^{e_m}$
3. If $\mathbf{q}(x)$ *is coprime with* $x^p - 1$ then
4. Calculate $\mathbf{s} = \mathbf{q}^{-1}(x) \cdot \mathbf{g}(x) \mod (x^p - 1)$
5. If *wt*$(\mathbf{s}) = m$ then
6. Return $\mathbf{q}(x)$ and $\mathbf{s}(x)$
7. end if
8. end if

Cryptanalysis - First Strategy

- **Time complexity.**

$$O\left(\binom{m^2}{m}p^2\right)$$

- **Numerical results.** For $p = 4032$ and $m = 7$, we obtain $2^{50.3}$ operations

- **Probability of success.** $\geq 94\%$

But we can do faster...

Cryptanalysis - Second Strategy

1. For each $1 \leq d \leq p-1$ do
2. $\quad \mathbf{g}_d(x) = x^d \cdot \mathbf{g}(x) \mod (x^p - 1)$
3. $\quad \mathbf{q}(x) = \mathbf{g}_d(x) \cap \mathbf{g}(x)$
4. $\quad$ If *(wt*$(\mathbf{q}) = m$*)* and *(*$\mathbf{q}(x)$ *coprime with* $x^p - 1$*)* then
5. $\qquad \mathbf{s}(x) = \mathbf{q}^{-1}(x) \cdot \mathbf{g}(x) \mod (x^p - 1)$
6. $\qquad$ If *wt*$(\mathbf{s}) = m$ then
7. $\qquad\quad$ Return $\mathbf{q}(x)$ and $\mathbf{s}(x)$
8. $\qquad$ End if
9. $\quad$ End if
10. End for

Cryptanalysis - Second Strategy

- **Time complexity.**

$$O\left(p^3\right)$$

- **Numerical results.** For $p = 4032$, we obtain 2^{36} operations

- **Probability of success.** Difficult to evaluate but experimentally $\simeq 69\%$

Cryptanalysis - Third Strategy

- Recall that each row of $\mathbf{G}^{-1}_{\leq k}(x)$ is of the form

$$\begin{pmatrix} \mathbf{d}_1(x) & \dots & \mathbf{d}_{n_0-1}(x) \end{pmatrix}$$

  with

$$\mathbf{d}_i(x) = \mathbf{q}(x) \cdot \mathbf{s}_i(x) \mod (x^p - 1)$$

- Define $\mathbf{E}_{i,j}(x) = \mathbf{d}_i(x) \cdot \mathbf{d}_j^{-1}(x) \mod (x^p - 1)$
- Note that we also have

$$\mathbf{E}_{i,j}(x) = \mathbf{s}_i(x) \cdot \mathbf{s}_j^{-1}(x) \mod (x^p - 1)$$

Cryptanalysis - Third Strategy

- Let $\mathscr{E}$ be the code defined by the generator matrix

$$\mathbf{E}(x) = \begin{pmatrix} \mathbf{1}(x) & \mathbf{E}_{2,1}(x) & \cdots & \mathbf{E}_{n_0-1,1}(x) \end{pmatrix}$$

- Then $\mathscr{E}$ contains p codewords of low weight $(n_0 - 1)m = 21$ since

$$\mathbf{s}_1(x) \cdot \mathbf{E}(x) = \begin{pmatrix} \mathbf{s}_1(x) & \mathbf{s}_2(x) & \cdots & \mathbf{s}_{n_0-1}(x) \end{pmatrix}$$

- Applying dedicated algorithms like Stern or Canteaut-Chabeaud
- Time complexity is about $2^{32.1}$ with Stern's algorithm

Secret Parity Check Matrix Extraction

- Once secret matrices S and $Q_1, \ldots, Q_{n_0-1}$ are found, calculate matrix

$$\tilde{G} = S \times G' \times \begin{pmatrix} Q_1 & & & \mathbf{0} \\ & \ddots & & \\ & & Q_{n_0-1} & \\ \mathbf{0} & & & I_p \end{pmatrix} = \left(\begin{array}{c|c} & (H_{n_0}^{-1} H_1)^T \times Q_{n_0}^{-1} \\ I_k & \vdots \\ & (H_{n_0}^{-1} H_{n_0-1})^T \times Q_{n_0}^{-1} \end{array} \right)$$

- Note that we **still need to discover** $H_1, \ldots, H_{n_0}$ and Q_{n_0}

Secret Parity Check Matrix Extraction

- Define $A_i = H_i \times H_{n_0}^{-1} \times (Q_{n_0}^{-1})^T$ and $B_{i,j} = A_i \times A_j^{-1}$
- Note that we also have:

$$B_{i,j} = H_i \times H_j^{-1}$$

- Define the code $\mathscr{C}_1$ spanned by the generator matrix G_1

$$G_1 = \begin{pmatrix} I_p & B_{2,1} & \cdots & B_{n_0-1,1} \end{pmatrix}$$

- $\mathscr{C}_1$ contains p codewords of low weight $(n_0 - 1)d_v = 39$ since

$$H_1 \times G_1 = \begin{pmatrix} H_1 & H_2 & \cdots & H_{n_0-1} \end{pmatrix}$$

Secret Parity Check Matrix Extraction

- Time complexity is about 2^{37} with $\textsc{Stern}$'s algorithm
- Final step:
  1. Compute $H_i^{-1} \times A_i = H_{n_0}^{-1} \times (Q_{n_0}^{-1})^T$
  2. Apply strategy 1 or 2 to find H_{n_0} and Q_{n_0}

Conclusion

- **Key reduction** is a **crucial** issue when considering McEliece cryptosystems
- **Hiding structure** is also a **main security** issue
- **Successfully** combining these two aspects represents a **big challenge**