Open access • Book Chapter • DOI:10.1007/978-3-319-03515-4_19

Polynomial Structures in Code-Based Cryptography — Source link

Vlad Dragoi, Pierre-Louis Cayrel, Brice Colombier, Tania Richmond

Institutions: University of Lyon

Published on: 07 Dec 2013 - International Conference on Progress in Cryptology

Topics: Polynomial, McEliece cryptosystem, Timing attack and Cryptography

Polynomial structures in code-based cryptography

Vlad Dragoi, Pierre-Louis Cayrel, Brice Colombier, Tania Richmond

Polynomial structures in code-based cryptography

Vlad Dragoi[1,2], Pierre-Louis Cayrel[1⋆], Brice Colombier[1], and

Tania Richmond[1⋆]

[1] Laboratoire Hubert Curien, UMR CNRS 5516,
University of Lyon, Saint-Etienne, France
pierre.louis.cayrel@univ-st-etienne.fr
tania.richmond@univ-st-etienne.fr
[2] Normandie Univ, France; UR, LITIS, F-76821 Mont-Saint-Aignan, France

Abstract. In this article we discus a probability problem applied in the code based cryptography. It is related to the shape of the polynomials with exactly t different roots. We will show that the structure is very dense and the probability that this type of polynomials has at least one coefficient equal to zero is extremelly low. We treated this issue in our research of natural countermeasures to a timing attack against the polynomial evaluation.

Keywords: *McEliece, Galois field,Monte-Carlo method, the simple roots problem*

Introduction

One of the main threats in modern cryptography is the arrival of the quantum computers, it was shown that cryptosystems based on factorisation of large numbers would be compromised [19]. Therefore, new concepts like hash-based cryptography, code-based cryptography, lattice-based cryptography, and multivariate cryptography were proposed as possible solutions. The new aspects of the post-quantum cryptography are well illustrated in [2].

Even though code-based cryptosystems exist since 1978, being introduced by Robert J. McEliece in [16], they weren't used in real life because of the key length problem. Nowadays, these problems are partially solved as new variants of the classical McEliece using shorter keys, without compromising the security, were proposed in [7,5,6,17] and more recently in [3]. The latest proposal for embedded devices proposed in [13] is based on QC-MDPC codes.

Here we will focus our attention on the last step in the decoding algorithm. If Patterson algorithm [18] or Berlekamp-Massey algorithm [15] is used, the last step is the same : finding the roots of the error locator polynomial. This polynomial has a particular form and it will be detailed in Section 1.

[⋆] Supported in part by NATO's Public Diplomacy Division in the framework of "Science for Peace", SPS Project 984520

McBits [3] is the latest implementation and uses some new algorithms in order to provide a fast constant-time decoding. Other existing implementations like: HyMes [7], CCA2-secure variant of McEliece [8], QD for embedded devices [11], Low-reiter [12], CFS [14], MicroEliece [10] use the mentioned decoding algorithms and manipulate the type of polynomials treated in this paper.

Our contribution

We will provide an answer the following problem :

What is the probability that all the coefficients of a monic polynomial $P(X)$ of degree t with t distinct roots over $\mathbb{F}_{2^m}$ are different from zero ?

Thefinal probability will be bounded by theoretical and experimental results. We will show how this result can be used in the context of side-channel attacks against the McEliece cryptosystem.

As shown, this problem has a direct application in code-based cryptography but it could be also usefull in many other scientific fields e.g. those where error correcting codes are used.

Organization of the paper

In Section 1, we give the required notations and some definitions and properties for the Goppa codes. The Section 2 details the simple roots problem and give the theoretical approach. We provide in Section 3 the experimental results. Section 4 shows how to apply this result and we wonclude in Section 5.

1 Preliminaries

1.1 Notations

We will use the following notations :

- The partial permutations $\mathcal{A}_n^k = n(n-1)\dots(n-k+1)$.
- The Galois field $\mathcal{L} : \mathbb{F}_{2^m} = \{0, 1, \alpha, \alpha^2, \dots, \alpha^{n-2}\}$
- Let $P(x)$ be a monic polynomial of degree t over $\mathcal{L}$ with t distinct roots a_i :

$$P(x) = x^t + S_{t-1}^t x^{t-1} + S_{t-2}^t x^{t-2} + ... + S_2^t x^2 + S_1^t x + S_0^t$$

  where the coefficients $S_i \in \mathbb{F}_q^m$ correspond to :

$$\begin{array}{lll} S_{t-1}^t = \sum\limits_{i=1}^{t} a_i, & S_{t-2}^t = \sum\limits_{\substack{i=1,j=1 \\ i\neq j}}^{t} a_i a_j, & \dots \\ \dots & S_1^t = \sum\limits_{j=1}^{t} \prod\limits_{\substack{i=1 \\ i\neq j}}^{t} a_i, & S_0^t = \prod\limits_{i=1}^{t} a_i. \end{array}$$

- The subset of all roots for a given polynomial $\mathcal{R}_{f(x)} = \{\ a_i \mid f(a_i) = 0\}$.

1.2 Goppa codes

Definition :

The Goppa code $\mathit{\Gamma}(\mathcal{L}, g)$ consists of all vectors $c = (c_0, c_1, ..c_{n-1})$ over $\mathbb{F}_q$ such that $\mathcal{S}_c(x) \equiv 0 \mod g(x)$. Here $g(x)$ is a polynomial over $\mathbb{F}_{2^m}$ and $\mathcal{L} = \{\alpha_0, \alpha_1, .., \alpha_{n-1}\}$ a subset so that $g(\alpha_i) \neq 0$ for all $i = 0 \ldots n-1$. $\mathcal{S}_c(x) = \sum\limits_{i=0}^{n-1} \frac{c_i}{x-\alpha_i}$ is called the syndrome of c and $\mathcal{L}$ the support of the Goppa code.
The syndrome polynomial $\mathcal{S}_c(x)$ satisfies the following property :

$$\mathcal{S}_c(x) = \frac{\omega(x)}{\sigma(x)} \mod g(x)$$

$\sigma(x)$ is called the error locator polynomial : $\sigma(x) = \prod\limits_{i=1}^{t} (x + a_i)$.
$\omega(x) = \sigma'(x)$ for binary Goppa codes.

2 Simple roots problem

Problem : Let $P(x)$ be a monic polynomial of degree t with t distinct roots over $\mathbb{F}_{2^m}$.
What is the probability that all its coefficients are different from zero ?

Proposition 1 : This probability is independent of the primitive generator polynomial $G(x)$ of degree m where $\mathbb{F}_{2^m} = \mathbb{F}_2[x]/G(x)$.

Proof. This is due to:

$$\mathbb{F}_2[x]/G_1(x) \cong \mathbb{F}_2[x]/G_2(x) \cong \mathbb{F}_2[x]/G_3(x) \cong \ldots \mathbb{F}_2[x]/G_{\mathcal{N}}(x)$$

where $G_i(x)$ are primitive polynomials $\forall i \in \{1, 2, .., \mathcal{N}\}$ of degree $= m$. □

2.1 General properties

Let $n = 2^m$.

1. $\boxed{\mathcal{P}\left(S_0^t = 0\right) = \frac{t}{n}}$

   Proof. If $S_0^t = 0$ then $0 \in \mathcal{R}_{P(x)}$. There are t different positions for any possible root. We can choose any of those t positions for zero. □

2. $\boxed{\mathcal{P}(S_1^t = 0 \cap S_0^t = 0) = 0}$

   Proof. If $S_0^t = 0$ then $0 \in \mathcal{R}_{P(x)}$. So $S_1^t = \prod\limits_{i=1}^{t-1} a_i = 0$. It means that zero is a root of order 2 of $P(x)$ and that's impossible. □

3. $\boxed{\mathcal{P}(S_i^t \in \mathbb{F}_{2^m} \cap S_1^t \neq 0 \cap S_0^t = 0) = \mathcal{P}(S_i^{t-2} \in \mathbb{F}_{2^m} \cap a_i \neq 0) \times \mathcal{P}(S_0^t = 0)}$

   Proof. If $S_0^t = 0$ then $0 \in \mathcal{R}_{P(x)}$ This implies that the two following events are equivalent :

   $$\{S_1^t \neq 0 \cap S_0^t = 0\} \Leftrightarrow \{S_0^{t-1} \neq 0 \cap a_t = 0\}.$$

   The only information we obtain with this event is that $a_t = 0$ then :

   $$\mathcal{P}(S_1^t \neq 0 \cap S_0^t = 0) = \mathcal{P}(S_0^t = 0).$$

   □

4. $\boxed{\text{If } S_i^t = 0 \text{ then } S_{i-1}^{t-1} = a_t S_i^{t-1} \quad \forall\, 0 < i < t}$

   Proof. The proof can easily be done by induction.
   Suppose that $S_{t-1}^t = 0$. It means that we can express a_t as :

   $$a_t = \sum_{i=1}^{t-1} a_i \Rightarrow a_t = S_{t-2}^{t-1}.$$

   If $S_{t-2}^t = 0 \Rightarrow a_t \sum_{i=1}^{t-1} a_i = \sum_{i \neq j,1}^{t-1} a_i a_j \Rightarrow a_t S_{t-2}^{t-1} = S_{t-3}^{t-1}$.
   If $S_{t-3}^t = 0 \Rightarrow a_t \sum_{i \neq j,1}^{t-1} a_i a_j = \sum_{i \neq j \neq k,1}^{t-1} a_i a_j a_k \Rightarrow a_t S_{t-3}^{t-1} = S_{t-4}^{t-1}$
   By induction, we obtain $S_1^t = 0 \Rightarrow a_t S_1^{t-1} = S_0^{t-1}$. □

In the following paragraph we will give two bounds for the probability. The lower bound is very close to our experimental results (see Section 3).

2.2 The bounds

We propose a lemma concerning the last coefficient (the sum) and we observe that the probability can be bounded. We consider for $\forall i \geq 3$ the probability $\mathcal{P}(S_{i-1}^i = 0)$. We give a general formula with the following consideration :

Lemma 1 :

$$\textit{Even } i : \mathcal{P}(S_{i-1}^i = 0) = \sum_{k=1}^{\lfloor \frac{i-1}{2} \rfloor} (-1)^{k-1} \tfrac{1}{n+2k-i} + (-1)^{\lfloor \frac{i-1}{2} \rfloor - 1} (\tfrac{1}{n-3} - \tfrac{1}{n-2})$$
$$\textit{Odd } i : \mathcal{P}(S_{i-1}^i = 0) = \sum_{k=1}^{\lfloor \frac{i-1}{2} \rfloor} (-1)^{k-1} \tfrac{1}{n+2k-i} + (-1)^{\lfloor \frac{i-1}{2} \rfloor - 1} (\tfrac{1}{n} - \tfrac{1}{n-1})$$

We will give some simple examples and observe that the general behavior of the sum suits the formula given above. We will use induction in order to prove it.

Main idea :

• Let $i = 3$. The probability associated to this event is $\mathcal{P} = \frac{A^2_{n-1}}{A^3_n} = \frac{1}{n}$. Consider (a_1, a_2, a_3) so that $\forall i \in \{1, 2, 3\}$ $a_i \in \mathcal{R}_{P(x)}$.
The number of all posible combinations is : $\mathcal{A}^3_n = n(n-1)(n-2)$.
The number of good cases is :

$$\#\{((a_1, a_2, a_3) \mid a_1 + a_2 + a_3 = 0\} = \#\{((a_1, a_2, a_3) \mid a_1 + a_2 = a_3\} = \mathcal{A}^2_{(n-1)} = (n-1)(n-2).$$

• Let $i = 4$. The probability is $\mathcal{P} = \frac{A^3_n}{A^4_n} = \frac{1}{n-3}$.
Consider (a_1, a_2, a_3, a_4) so that $\forall i \in \{1, 2, 3, 4\}$ $a_i \in \mathcal{R}_{P(x)}$.
The number of all posible combinations is : $\mathcal{A}^4_n = n(n-1)(n-2)(n-3)$.
The number of good cases is :

$$\#\{((a_1, a_2, a_3, a_4) \mid a_1 + a_2 + a_3 + a_4 = 0\} = \#\{((a_1, a_2, a_3, a_4) \mid a_1 + a_2 + a_3 = a_4\} = \mathcal{A}^3_n = n(n-1)(n-2).$$

Is it possible that for a given (a_1, a_2, a_3, a_4) solution, the choice of a_4 might cause repetitions? We know that a_4 is fixed as $a_4 = a_1 + a_2 + a_3$ and all the elements are different (because $P(X)$ has 4 distincts roots).
Example: If $a_4 = a_1$ then $a_2 = a_3$. But a_2 and a_3 must be different. So it is impossible that $a_4 = a_1$. Therefore we have the exact probability $\mathcal{P} = \frac{1}{n-3}$
• Let $i = 5$
If $a_2 \neq a_3 \neq a_4 \neq a_2$ then the event related to $a_1 + \cdots + a_5 = 0$ has the following form:

$$❶\ \{s = \sum_{i=1}^{5} a_i = 0\} = \{s = 0 \cap a_1 = a_5\} \cup \{s = 0 \cap a_1 \neq a_5\}$$

The event $\{s = 0 \cap a_5 = a_1\}$ was treated in the case $i = 3$. So $\mathcal{P}(\{s = 0 \cap a_5 = a_1\}) = \frac{1}{n}$.

The event $\{s = \sum_{i=1}^{5} a_i = 0\}$ has the following probability:

$$\mathcal{P}(\{s = \sum_{i=1}^{5} a_i = 0\}) = \frac{n(n-1)(n-2)(n-3)}{n(n-1)(n-2)(n-3)(n-3)} = \frac{1}{n-3}$$

Finally we obtain the probabillity $\mathcal{P} = \frac{1}{n-3} - \frac{1}{n}$.

For $i \in 6, 7, 8$ we will only give the final result. The idea and the calculus are the same as for the explained cases.
• Let $i = 6$ the probabillity is : $\mathcal{P} = \frac{1}{n-4} - \frac{1}{n-3}$.
• Let $i = 7$ the probabillity is : $\mathcal{P} = \frac{1}{n-5} - (\frac{1}{n-3} - \frac{1}{n})$.
• Let $i = 8$ the probabillity is : $\mathcal{P} = \frac{1}{n-6} - (\frac{1}{n-4} - \frac{1}{n-3})$.

Proof. By induction :
• For the *even* case : The hypothesis is satisfied for

$i = 4$ as we have $\mathcal{P}(S_3^4 = 0) = \frac{1}{n-3}$.

Suppose that $i = 2p$ and

$$\mathcal{P}(S_{2p-1}^{2p} = 0) = \sum_{k=1}^{\lfloor\frac{2p-1}{2}\rfloor} (-1)^{k-1} \frac{1}{n+2k-2p} + (-1)^{\lfloor\frac{2p-1}{2}\rfloor - 1} (\frac{1}{n-3} - \frac{1}{n-2}).$$

We will search the $\mathcal{P}(S_{2p+1}^{2p+2} = 0)$

As before we distinguish the case where $a_{2p} = a_1$ and the case where $a_{2p} \neq a_1$ (the general case in ❶).

So we have $\mathcal{P}(S_{2p+1}^{2p+2} = 0) + \mathcal{P}(S_{2p-1}^{2p} = 0) = \frac{1}{n-2p}$

We finally obtain :

$$\mathcal{P} = \frac{1}{n-2p} - \left[\sum_{k=1}^{\lfloor\frac{2p-1}{2}\rfloor} (-1)^{k-1} \frac{1}{n+2k-2p} + (-1)^{\lfloor\frac{2p-1}{2}\rfloor - 1} (\frac{1}{n-3} - \frac{1}{n-2})\right] = \sum_{k=1}^{\lfloor\frac{2p+1}{2}\rfloor} (-1)^{k-1} \frac{1}{n+2k-(2p+2)} + (-1)^{\lfloor\frac{2p+1}{2}\rfloor - 1} (\frac{1}{n-3} - \frac{1}{n-2})$$

• For the *odd* case on can easilly use the same proof.

Asymptotically, $\mathcal{P} \approx \frac{1}{n-i+2}$. □

Lemma 2 :

$$\mathcal{P}(S_i^t = 0) \approx \mathcal{P}(S_{t-1}^t = 0) \ \forall i \in \{1, 2, .., t-2\}$$

Proof. Using properties 3 and 4 from 2.1 we get :

$$S_i^t = 0 \Rightarrow S_{i-1}^{t-1} = a_t S_i^{t-1} \quad \forall\ 0 < i < t$$

So we have all the possible choices on the first $t-1$ elements, as for the last one it has to be defined as in the formula above. We get the same number of possible choices for $(a_1, a_2, ..., a_t)$ as in the case $S_{t-1}^t = 0$. □

Proposition 2 : For a given polynomial with t different roots the probability that all coefficients are different from zero can be bouded by the two following quantities :

The two bounds:

$$1 + f(n,t) - \left[\frac{t}{n} + (t-1)\mathsf{ub}\right] \leq \mathcal{P}(\bigcap_{i=0}^{t-1} S_i^t \neq 0) \leq 1 + f(n,t) - \left[\frac{t}{n} + (t-1)\mathsf{lb}\right]$$

Proof From **Lemma 1** we have :

$$\mathcal{P}(S_{t-1}^t = 0) = \frac{1}{n-t+2} - \frac{1}{n-t+4} + \frac{1}{n-t+6} - \frac{1}{n-t+8} + \frac{1}{n-t+10} + \dots$$

So :

$$\mathsf{lb} \leq \mathcal{P}(S_{t-1}^t = 0) \leq \mathsf{ub}$$

where

$$\mathsf{lb} = \frac{1}{n-t+2} - \frac{1}{n-t+4}$$

and

$$\mathsf{ub} = \frac{1}{n-t+2} - \frac{1}{n-t+4} + \frac{1}{n-t+6}$$

Using property 1 from 2.1 we have $\mathcal{P}(S_0^t = 0) = \frac{t}{n}$.
From **Lemma 2** we can approach

$$\mathcal{P}(S_i^t = 0) \approx \mathcal{P}(S_{t-1}^t = 0) \; \forall i \in \{1, 2, .., t-2\}$$

We will be able to approach the sum :

$$\sum_{i=0}^{t-1} \mathcal{P}(S_i^t = 0) \approx \mathcal{P}(S_0^t = 0) + (t-1)\mathcal{P}(S_{t-1}^t = 0)$$

Givind the bounds for the sum it becomes a simple task :

$$\frac{t}{n} + (t-1) \times \mathsf{lb} \leq \sum_{i=0}^{t-1} \mathcal{P}(S_i^t = 0) \leq \frac{t}{n} + (t-1) \times \mathsf{ub}$$

Finally :

$$\mathcal{P}(\bigcap_{i=0}^{t-1} \{S_i^t \neq 0\}) = 1 - \mathcal{P}(\exists i \; S_i^t = 0)$$

Notation : $f(n,t)$ represents the sum of the probabilities associated to all the possible intersections between S_i^t $\forall i$ so that at least two coefficients equal zero.

Example for $t = 3$ *:*

$$\{S_0^3 = 0\} = \bigcup \{S_0^3 = 0, S_1^3 \in \{0, \neq\}, S_2^3 \in \{0, \neq\}\}$$

We have the same relation for S_1^3 and S_2^3. So all the possible combinations where at least two members equal zero will constitute the function.

$$\begin{aligned} f(n,3) = 2 \times \mathcal{P}(S_0^3 = S_1^3 = \quad & S_2^3 = 0) \\ +\mathcal{P}(S_0^3 = S_1^3 = 0, & S_2^3 \neq 0) \\ +\mathcal{P}(S_0^3 = S_2^3 = 0, & S_1^3 \neq 0) \\ +\mathcal{P}(S_1^3 = S_2^3 = 0, & S_0^3 \neq 0) \end{aligned}$$

We use the following relation in order to finalize our proof :

$$\sum_{i=0}^{t-1} \mathcal{P}(S_i^t = 0) = \mathcal{P}(\exists i \ S_i^t = 0) + f(n,t)$$

$$\mathcal{P}(\bigcap_{i=0}^{t-1} \{S_i^t \neq 0\}) = 1 + f(n,t) - \sum_{i=0}^{t-1} \mathcal{P}(S_i^t = 0)$$

So :

$$1 + f(n,t) - \left[\frac{t}{n} + (t-1) \times \mathsf{ub}\right] \leq \mathcal{P}(\bigcap_{i=0}^{t-1} \{S_i^t \neq 0\}) \leq 1 + f(n,t) - \left[\frac{t}{n} + (t-1) \times \mathsf{lb}\right]$$

That sets the two bounds but doesn't allow having a graphic representation since the quantity $f(n,t)$ is unknown. □

One of the ideas was to consider the following result :

$$1 - \left[\frac{t}{n} + (t-1) \times \mathsf{ub}\right] \leq 1 + f(n,t) - \left[\frac{t}{n} + (t-1) \times \mathsf{ub}\right] \leq \mathcal{P}(\bigcap_{i=0}^{t-1} \{S_i^t \neq 0\})$$

We represented in Section 3 the lower bound $1 - \left[\frac{t}{n} + (t-1) \times \mathsf{ub}\right]$ and the experimental values using the Monte Carlo method. As expected the quantity represented by $f(n,t)$ could be neglected in the formula. Therefore we used two following bounds in Section 3 :

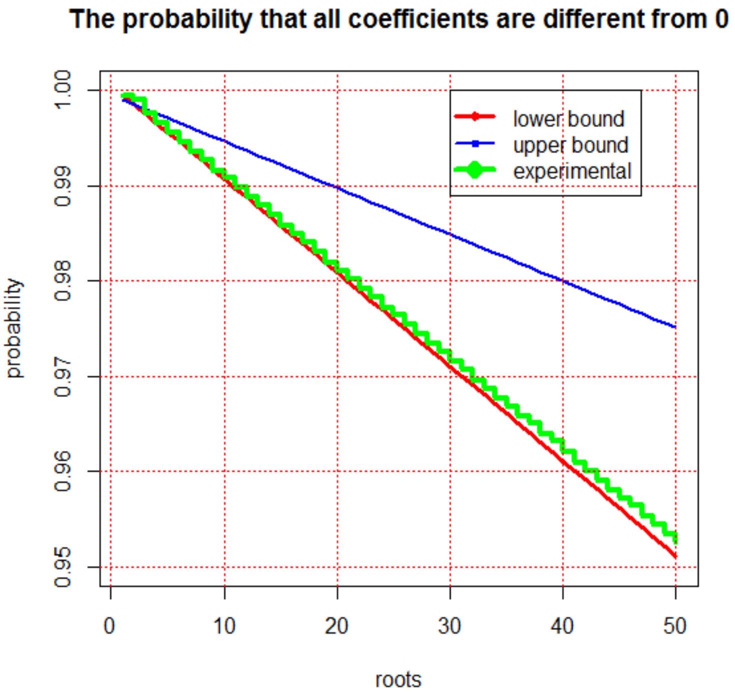
$$\mathsf{LB} = 1 - \left[\frac{t}{n} + (t-1) \times \mathsf{ub}\right] \text{ and } \mathsf{UB} = 1 - \left[\frac{t}{n} + (t-1) \times \mathsf{lb}\right]$$

3 Experiments

Simulations were made using PariGP, a free software used especially for its abillity to generate finite fields in the Galois field theory.

For the experimental approach we used the Monte-Carlo method. It uses the Central Limit Theorem and applied in our case to estimate the number of coefficients equal to zero for a given polynomial. We will detail in the next paragraph the procedure used in order to obtain the results. After that we will give the graphical representation of the simulated variables and the theoretical bounds. We will see that the possible distribution is very close to one of the bounds.

Fig. 1. Experimental and theoretical bounds for $n = 2048$



First of all we simulated for a given number t of roots the corresponding polynomial. Then we counted the number of coefficients that equal zero. We repeated the simulation 3.000.000 times for each t. In our case the Monte Carlo method was applied to the variable : *number of coefficients that equal zero.*

Results : The figure illustrates the importance of the lower bound. Since we are interested in having less coefficients equal to zero the lower bound gives the folowing values :

classical parameters $n = 2048$ and $t \leq 50$ we obtain $\mathcal{P} \geq 0.95$;
128-bit security $[2960; 2288]$ Goppa code ($t = 56$) we obtain $\mathcal{P} \geq 0.9622$;
256-bit security $[6624; 5129]$ Goppa code ($t = 115$) we obtain $\mathcal{P} \geq 0.9651$.

4 Applications

4.1 The McEliece Cryptosystem [16]

KeyGen : The first step is to generate the support $\mathcal{L}$ and the Goppa polynomial $g(x)$. Once this step is achieved, we can build the parity check matrix and bring it into systematic form $\mathsf{pk} = (m, t, R^T, \mathcal{L})$. The permutation $\varPi$ and the Goppa polynomial $g(x)$ form the secret key $\mathsf{sk} = (g(x), \varPi)$.

Encrypt :

- *Input :* message $\mathsf{m} \in \mathbb{F}_2^k$, public key $\mathsf{pk} = (m, t, R^T, \mathcal{L})$
- *Output :* ciphertext $\mathsf{z} \in \mathbb{F}_2^n$

1. Expand public key R^T to $\mathcal{G} = [R^T | \mathcal{I}_k]$;
2. Choose a random n-bit error-vector with $\mathsf{wt}(e) = t$;
3. Encode $\mathsf{z} = \mathsf{m}\mathcal{G} \oplus e$;
4. Return z.

Decrypt :

- *Input :* ciphertext $\mathsf{z} \in \mathbb{F}_2^n$, secret key $\mathsf{sk} = (g(x), \varPi)$
- *Output :* message $\mathsf{m} \in \mathbb{F}_2^k$

1. Find e' using $\mathcal{D}ecode(\mathsf{z}, \mathsf{sk})$
2. $\mathsf{m} \leftarrow$ the first k bits of $\mathsf{z} \oplus e'$
3. Return m.

$\mathcal{D}ecode(.,.)$ is a decoding algorithm used for the Goppa codes.

4.2 Side-channel attacks

The most important side-channel attacks treated in the scientific literature are timing attacks. They operate on the software implementation of the McEliece PKC and can be classified by their goal:

1. Recover the secret message (in [23,1])
2. Recover the secret key, fully or partially (in [22,21,20])

The type of attacks aiming to recover the secret message exploit timing differences between $\deg(\sigma_1) = t$ and $\deg(\sigma_2) = t - 1$. The countermeasures proposed manipulate $\sigma(x)$ so that if $\deg(\sigma) < t$ the designer should either

1. deterministicaly add coefficients so that $\deg(\sigma) = t$ and all coefficients are non zero
2. use coefficents from the non-support so that $\deg(\sigma) = t$ and all coefficients are non zero

Countermeasure Our idea is that the second part of the statement *make sure that all coefficients are non zero* is already verified by 2. So we should only manipulate the degree of σ. Then the probability of having at least one coefficient equal to zero in σ is extremely low.

4.3 CFS signature scheme

In the CFS signature scheme, a small number t is used due to the density of the Goppa codes. It was proven in [9] that the decoding algorithm must be repeated in average $t!$ times. Decoding Goppa Codes for CFS with the recommended values gives the following result :

$$\text{for } n = 2^{16} \text{ and } t \leq 10 \text{ we obtain } \mathcal{P} > 0.999.$$

5 Conclusion

In this article, we have treated the simple roots polynomial problem. We have shown that the structure is such that timing attacks are difficult to be applied, since most of the σ-coefficient are different from 0. The security comes directly from the structure of the Galois field and the form of the error-locator polynomial.

6 Acknowledgements

We would like to thank Alain Couvreur and Michael Bulois for their help and comprehension. We would also like to thank Florent Bernard for his support.

References

1. Roberto Avanzi, Simon Hoerder, Dan Page, and Mike Tunstall. Side-channel attacks on the McEliece and Niederreiter public-key cryptosystems. In *Cryptology ePrint Archive, Report 2010/479*, 2010.
2. Daniel J. Bernstein, Johannes Buchmann, and Erik Dahmen. *Post-Quantum Cryptography,Springer*. 2009.
3. Daniel J. Bernstein, Tung Chou, and Peter Schwabe. McBits: fast constant-time code-based cryptography. 2013.0616.

4. Daniel J. Bernstein, Tanja Lange, and Christiane Peters. Attacking and defending the McEliece cryptosystem. In *PQCrypto 2008, Cincinnati, OH, USA, LNCS 5299, Springer.*
5. Daniel J. Bernstein, Tanja Lange, and Christiane Peters. Wild McEliece. In *Proceedings of Selected Areas Cryptography, SAC 2010, Waterloo, Canada, LNCS,Springer.*
6. Daniel J. Bernstein, Tanja Lange, and Christiane Peters. Wild McEliece incognito. In *PQCrypto 2011, LNCS,Springer.*
7. Bhaskar Biswas and Nicolas Sendrier. McEliece cryptosystem implementation : theory and practice. In *PQCrypto 2008, USA,LNCS, Springer.*
8. Pierre-Louis Cayrel, Gerhard Hoffmann, and Edoardo Persichetti. Efficient implementation of a CCA2-secure variant of McEliece using generalized Srivastava codes. In *Proceedings of PKC 2012, LNCS 7293, Springer-Verlag*, pages 138–155, 2012.
9. Nicolas Courtois, Matthieu Finiasz, and Nicolas Sendrier. How to achieve a McEliece-based digital signature scheme. In *ASIACRYPT 2001, LNCS, Springer.*
10. Thomas Eisenbarth, Stefan Heyse Tim Guneysu, and Christof Paar. Microeliece : McEliece for embedded devices. In *CHES '09, Berlin, Heidelberg, Springer-Verlag.*
11. Stefan Heyse. Implementation of McEliece based on Quasi-dyadic Goppa Codes for Embedded Devices. In *PQCrypto 2011, LNCS 7071, Springer.*
12. Stefan Heyse. Low-reiter : Niederreiter encryption scheme for embedded microcontrollers. In Nicolas Sendrier, editor, *PQCrypto 2010, LNCS, Springer.*
13. Stefan Heyse, Ingo von Maurich, and Tim Guneysu. Smaller Keys for Code-based Cryptography: QC-MDPC McEliece Implementations on Embedded Devices. 2013.
14. Gregory Landais and Nicolas Sendrier. CFS Software Implementation. In *Indocrypt 2012 and Cryptology ePrint Archive, Report 2012/132*, 2012.
15. James L. Massey. Shift-register synthesis and bch decoding. In *Transactions on Information theory, Vol IT-15, No1, January 1969*, pages 122–127, 1969.
16. Robert J. McEliece. A public-key cryptosystem based on algebraic coding theory. In *Jet Propulsion Laboratory DSN Progress Report 42-44*, pages 114–116, 1978.
17. Rafael Misoczki, Jean-Pierre Tillich, Nicolas Sendrier, and Paulo S. L. M. Barreto. Mdpc-McEliece : New McEliece variants from moderate density parity-check codes. In *Cryptology ePrint Archive, Report 2012/409*, 2012.
18. Nicholas J. Patterson. The algebraic decoding of goppa codes. In *IEEE Transactions on Information Theory IT-21*, pages 203–207, 1975.
19. Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. 1994.
20. Abdulhadi Shoufan, Falko Strenzke, H. Gregor Molter, and Marc Stottinger. A Timing Attack against Patterson Algorithm in the McEliece PKC. In *ICISC 2009.*
21. Falko Strenzke. A Timing Attack against the Secret Permutation in the McEliece PKC. In Nicolas Sendrier, editor, *PQCrypto 2010, LNCS, Springer.*
22. Falko Strenzke. Timing attacks against the syndrome inversion in code-based cryptosystems. In *Cryptology ePrint Archive, Report 2011/683*, 2011.
23. Falko Strenzke, Erik Tews, H. Gregor Molter, Raphael Overbeck, and Abdulhadi Shoufan. Side channels in the McEliece pkc. In Johannes Buchmann and Jintai Ding, editors, *PQCrypto 2008, LNCS, Springer.*