A graduate introduction to classic McEliece

Ganyu (Bruce) Xu

November, 2024

1 Introduction

Classic McEliece is an IND-CCA2 key encapsulation mechanism (KEM) submitted to NIST's post-quantum cryptography (PQC) standardization project. Its security is based on the conjectured intractability of the **Syndrome Decoding Problem**.

This document provides an introduction to the mathematics behind classic McEliece with a particular emphasis on the details of binary Goppa code. We assume a graduate level of mathematical maturity, though most of the results can be reasoned about using first or second year undergraduate math.

Classic McEliece is an appealing candidate for PQC because its cryptanalysis has been remarkably stable since its earliest conception in 1978 [McE78] (whereas lattice-based cryptography saw enormous advance in its cryptanalysis in the last decade alone, forcing many latticed-based schemes to revise their parameters). Classic McEliece also has the following advantages:

- Encryption is faster
- Ciphertext is smaller
- There is no decryption failure that can leak information about secret key

On the other hand, Classic McEliece has larger public key size and slower decryption routine.

1.1 A summary of binary Goppa code

Binary Goppa code is a linear error-correcting code. Each instance is parameteried by:

- some base field $K = \mathbb{F}_{2^m}$
- n distinct field elements $\alpha_1, \alpha_2, \ldots, \alpha_n \in K$
- An irreducible polynomial $g \in K[x]$ such that $\deg(g) = t$

1.2 An overview of Classic McEliece

Each instance of Classic McEliece is parameterized by three integers:

- m is the size of the base field $K = \mathbb{F}_{2^m}$
- n is the size of code words: codewords $\subseteq K^n$
- t is the error-correcting capacity of the underlying Goppa code

Key generation. Begin by randomly generating the parameters of a Goppa code instance, which include:

- n distinct field elements $\alpha_1, \alpha_2, \ldots, \alpha_n \in K$
- A degree-t, square-free polynomial $g \in K[x]$ such that $g(\alpha_i) \neq 0$ for $1 \leq i \leq n$

$\alpha_1, \alpha_2, \ldots, \alpha_n$ and g can be used to compute the canonical parity-check matrix $H \in K^{t \times n}$

$$H_{i,j} = \frac{\alpha_j^{i-1}}{g(\alpha_j)} \text{ for } 1 \leq i \leq t, 1 \leq j \leq n$$

H is then transformed into reduced row-echelon form (or systematic form per DJB) $H' = [I_t \mid T]$ for some $T \in K^{t \times (n-t)}$. T is returned as the public key. $\Gamma = (\alpha_1, \alpha_2, \ldots, \alpha_n, g)$ is returned as the secret key.

Encryption. The message space is the subset of $\mathbb{F}_2^n$ whose Hamming weight is exactly t:

$$\mathcal{M} = \{\mathbf{e} \in \mathbb{F}_2^n \mid wt(\mathbf{e}) = t\}$$

To encrypt, compute:

$$\mathbf{y} \leftarrow [I_t \mid T]\mathbf{e}$$

$\mathbf{y} \in K^t$ is returned as the ciphertext.

Decryption. Given $\mathbf{y} \in K^t$ and $\Gamma = (\alpha_1, \alpha_2, \ldots, \alpha_n, g)$, let $\mathbf{r} \leftarrow (y_1, y_2, \ldots, y_t, 0, 0, \ldots, 0) \in \mathbb{F}_2^n$. then feed $(\Gamma, \mathbf{r})$ into some Goppa decoder, which will directly recover $\mathbf{e} \in \mathbb{F}_2^n$ such that $wt(\mathbf{e}) = t$. Return $\mathbf{e}$ as the decryption.

2 Preliminaries

3 Understanding binary Goppa code decoding

3.1 Polynomial interpolation

Theorem 3.1 (Lagrange Interpolation)**.** *Let K be a finite field, $\alpha_1, \alpha_2, \ldots, \alpha_n \in K$ be n distinct field elements, and $r_1, r_2, \ldots, r_n \in K$ be n (possibly non-distinct) elements. There exists a unique degree-$(n-1)$ polynomial f such that $f(\alpha_i) = r_i$:*

$$f = \sum_{i=1}^{n} \left(r_i \prod_{j \neq i} \frac{x - \alpha_j}{\alpha_i - \alpha_j} \right)$$

It's easy to check that the interpolation is correct, and that the degree of f is less than n. If there exists another $g \in K[x]$ with $\deg(g) < n$ that interpolates $(\alpha_i, r_i)_{i=1}^n$, then $f - g$ is a polynomial with n distinct roots. However, $\deg(f - g)$ is less than n since $\deg(f), \deg(g) < n$. Therefore, it must be that $f - g = 0$, which means that $f = g$, thus proving uniqueness.

3.2 Shamir's secret sharing

Lagrange interpolation can be used to build a secret sharing scheme [Sha79].

3.3 Polynomial approximant

Theorem 3.2 (Best approximant theorem)**.** *Let K be some finite field and n, t be non-negative integers such that $2t < n$. Given polynomials $A, B \in K[x]$ such that $\deg(B) < \deg(A)$, then there exists unique pair of polynomials $a, b \in K[x]$ such that $\deg(a) \leq t$, $\deg(b) < \deg(a)$, $\gcd(a, b) = 1$, and $\deg(aB - bA) < n - t$. If $c, d \in K[x]$ is such that $\deg(c) \leq t$ and $\deg(cB - dA) < n - t$, then $(c, d) = \lambda(a, b)$ for some $\lambda \in K[x]$.*

3.4 Interpolation with error

In this section we introduce an algorithm that can recover a degree $(n-2t)$ polynomial given n points with up to t errors.

Let K be some finite field and n, t be non-negative integers such that $2t < n$. Let $f \in K[x]$ be a polynomial with degree $\deg(f) < n - 2t$. Let $\alpha_1, \alpha_2, \ldots, \alpha_n$ be n distinct elements. Denote $\mathbf{c} = (f(\alpha_1), f(\alpha_2), \ldots, f(\alpha_n)) \in K^n$. If $\mathbf{r} \in K^n$ is such that $wt(\mathbf{r} - \mathbf{c}) \leq t$, then the following procedure can recover f using $\alpha_1, \alpha_2, \ldots, \alpha_n$ and $\mathbf{r}$:

1. Let $A = \prod_{i=1}^{n}(x - \alpha_i)$
2. Let B interpolate (α_i, r_i)
3. Compute degree-t approximant (a, b) of (A, B). The error can be corrected if and only if $a \mid A$
4. Compute $f = B - bA/a$. Furthermore, $(B - f)(\alpha_i) \neq 0$ if and only if $a(\alpha_i) = 0$.

Proof. □

3.5 Goppa decoding

3.6 Parity check matrix

4 Implementation details

References

[McE78] Robert J McEliece. A public-key cryptosystem based on algebraic. *Coding Thv*, 4244:114–116, 1978.

[Sha79] Adi Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.