Problems Session 3:

Cryptanalysis and McEliece Cryptosystem

1 McEliece cryptosystem based on GRS codes

1.1 Appetizer

This exercise is widely inspired by [CGGO+13].

Notation 1. *Let k be a non-negative integer. We denote by $\mathbb{F}_q[X]_{\leq k}$ the space of polynomials with coefficients in $\mathbb{F}_q$ of degree less than or equal to k.*

Recall the definition of a Generalized Reed-Solomon code (GRS):

Definition 1 (Generalized Reed-Solomon Code)**.** *Let $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{F}_q^n$ be an n-tuple of pairwise distinct elements of $\mathbb{F}_q$ (in particular it entails $n \leq q$), and let $k \leq n$. Let $\mathbf{z} = (z_1, \ldots, z_n) \in (\mathbb{F}_q^\times)^n$ be an n-tuple of* non zero *elements, not necessarily distinct. The* $\mathrm{GRS}_k(\mathbf{x}, \mathbf{z})$ *code is defined as*

$$\mathrm{GRS}_k(\mathbf{x}, \mathbf{z}) \overset{def}{=} \{(z_1 P(x_1), \ldots, z_n P(x_n)) \mid P \in \mathbb{F}_q[X]_{<k}\}. \tag{1}$$

$\mathbf{x}$ *and* $\mathbf{z}$ *are respectively called the* support *and* multiplier *vectors of* $\mathrm{GRS}_k(\mathbf{x}, \mathbf{z})$.

Recall from Lecture and Problem Session 1 that $GRS_k(\mathbf{x}, \mathbf{z})$ is a code of length n, dimension k and minimum distance $n - k + 1$. Moreover, there exist efficient decoding algorithms up to $\dfrac{n-k}{2}$ errors, which is the largest amount of errors one can hope to uniquely decode.

Q1. Show that $\mathrm{GRS}_k(\mathbf{x}, \mathbf{z})^\perp = \mathrm{GRS}_{n-k}(\mathbf{x}, \mathbf{z}^\perp)$ where $y_i^\perp = \dfrac{1}{z_i \prod_{i \neq j}(x_i - x_j)}$. In particular, the dual of a GRS code of dimension k, is a GRS code of dimension $n - k$, with same support.

Hint: Use Lagrange interpolation.

Due to their unique decoding property, Neiderreiter suggested in [Nie86] to instanciate McEliece cryptosystem with this class of codes, in order to reduce the size of the keys. However, in [SS92] Sidelnikov and Shestakov proved that such an instantiation was insecure. The goal of this exercise is to give another attack on GRS-based McEliece cryptosystem, using a very versatile tool called the *star product* of codes. Although being slower than the historical attack of Sidelnikov and Shestakov, this tool proved itself very useful to design attacks on instanciations of McEliece cryptosystems based on many families of algebraic codes such as Algebraic-Geometry codes (of any genus), which are generalizations of Reed-Solomon codes, Wild Goppa codes over quadratic extensions, some subspace subcodes of GRS codes etc... This tool is also at the core of a very recent approach to cryptanalyze McEliece cryptosystem based on alternant and Goppa codes [CMT23].[1]

1.2 A Distinguisher

Given access to a public GRS code $\mathcal{C}_{pub} = \mathrm{GRS}_k(\mathbf{x}, \mathbf{z})$ (through a generator matrix for instance), the goal is to recover a pair $(\mathbf{x}', \mathbf{z}')$ such that $\mathcal{C}_{pub} = \mathrm{GRS}_k(\mathbf{x}', \mathbf{z}')$ (Note that there exist many secret data $(\mathbf{x}, \mathbf{z})$ which yield the same public GRS code).

Q2. Let $\alpha, \gamma \in \mathbb{F}_q^\times$ and $\mathbf{b} = (b, \dots, b) \in \mathbb{F}_q^n$.

(a) Show that $\mathrm{GRS}_k(\mathbf{x}, \mathbf{z}) = \mathrm{GRS}_k(\alpha\mathbf{x} + \mathbf{b}, \gamma\mathbf{z})$.

(b) Deduce that we can assume without loss of generality that $x_1 = 0$ and $x_2 = 1$.

We introduce the *star-product*, also known as the Schur product (or coordinate-wise product):

Definition 2 ($\star$-product)**.**

- *Let* $\mathbf{a}, \mathbf{b} \in \mathbb{F}_q^n$. *We define the* $\star$*-product of* $\mathbf{a}$ *and* $\mathbf{b}$ *as*

$$\mathbf{a} \star \mathbf{b} \overset{def}{=} (a_1 b_1, \dots, a_n b_n). \tag{2}$$

- *Let* $\mathcal{A}, \mathcal{B} \subset \mathbb{F}_q^n$ *be two linear codes. We define their* $\star$*-product as*

$$\mathcal{A} \star \mathcal{B} \overset{def}{=} \mathrm{Span}\{\mathbf{a} \star \mathbf{b} \mid \mathbf{a} \in \mathcal{A}, \mathbf{b} \in \mathcal{B}\}, \tag{3}$$

Remark 1. *The presence of Span is here to ensure that* $\mathcal{A} \star \mathcal{B}$ *is still a linear code.*

When $\mathcal{A} = \mathcal{B}$, we denote by $\mathcal{A}^2 \overset{\mathrm{def}}{=} \mathcal{A} \star \mathcal{A}$ the *square* of the code $\mathcal{A}$.

[1] There has been a lot of progress in the past few years, even though the parameters of NIST submission Classic McEliece are still out of reach.

Q3. Let $\mathcal{C} \in \mathbb{F}_q^n$ be a linear code of dimension k.

(a) Show that

$$\dim \mathcal{C}^2 \leq \min\left(n, \binom{k+1}{2}\right). \tag{4}$$

(b) Show that the complexity of computing a basis of $\mathcal{C}^2$ given a basis of $\mathcal{C}$ is $O(k^2n^2)$ operations in $\mathbb{F}_q$.

In reality, this inequality is sharp for *random codes*. Indeed, it has been proven in [CCMZ15] that when $\binom{k+1}{2} \leq n$ then $\dim \mathcal{C}^2 = \binom{k+1}{2}$ with overwhelming probability. In particular, the dimension of the square of a random code is *quadratic* in the dimension of the code.

Q4. (a) Show that for $k \leq (n+1)/2$,

$$\mathrm{GRS}_k(\mathbf{x}, \mathbf{z})^2 = \mathrm{GRS}_{2k-1}(\mathbf{x}, \mathbf{z} \star \mathbf{z}). \tag{5}$$

(b) Deduce a way to distinguish between small rate Generalized Reed-Solomon codes and random linear codes using the $\star$-product.

(c) Show that high rate GRS codes (when $2k-1 > n$) are also distinguishable from random codes.

Q5. **(Bonus.)** Can you give a very simple way (not using the $\star$-product) to distinguish between $\mathrm{RS}_k(\mathbf{x})$ (*i.e.*, the multiplier is the all 1 vector) and a random linear code? In particular, for McEliece cryptosystem, using a *non-trivial* multiplier is necessary.

1.3 Defining the Filtration

So far we have found a *distinguisher* between GRS and random linear codes. It can undermine the security, but it is not yet an attack on the cryptosystem. There is still some work to do to recover the secret parameters $(\mathbf{x}, \mathbf{z})$. From now on, we assume that $x_1 = 0, x_2 = 1$ and $k \leq \dfrac{n-1}{2}$.

Defining the Filtration. In order to recover the secret parameters, we will build a *filtration* of codes, *i.e.*, a sequence $(\mathcal{C}_i)$ of codes such that

$$\mathcal{C}_{pub} = \mathcal{C}_0 \supset \mathcal{C}_1 \supset \mathcal{C}_2 \supset \cdots \supset \mathcal{C}_i \supset \dots \tag{6}$$

where $\mathcal{C}_i \star \mathcal{C}_j \subset \mathcal{C}_{i+j}$.

In order to build this filtration, we will need a new operation on codes, namely the *shortening.*

Definition 3 (Shortened Code)**.** *Let $\mathcal{C}$ be a linear code, and $\mathcal{I} \subset \{1, \dots, n\}$ a set of positions. We define the shortening of $\mathcal{C}$ at $\mathcal{I}$ as the code $S_{\mathcal{I}}(\mathcal{C})$:*

$$S_{\mathcal{I}}(\mathcal{C}) \overset{def}{=} \{c \in \mathcal{C} \mid c_i = 0 \quad \forall i \in \mathcal{I}\}. \tag{7}$$

Remark 2. *This definition is slighly different as the one usually used. Indeed, with this definition $S_{\mathcal{I}}(\mathcal{C})$ contains codewords which are 0 on the same coordinates, and one usually delete those entries, yielding a code of length $n - |\mathcal{I}|$. However, the $\star$-product is only well defined for vectors of same length, therefore it is easier to keep those zero components.*

Q6. Given a code $\mathcal{C}$ and a set of positions $\mathcal{I}$, how can we compute a basis of $S_{\mathcal{I}}(\mathcal{C})$?

For $i, j > 0$ and $i + j < k$, we denote by $\mathcal{C}(i, j)$ the subcode of $\mathcal{C}_{pub} = \mathrm{GRS}(\mathbf{x}, \mathbf{z})$ given by the evaluation of polynomials vanishing at 0 with multiplicity at least i and at 1 with multiplicity at least j. We also set $\mathcal{C}(0, 0) \overset{\text{def}}{=} \mathcal{C}_{pub}$.

Q7. (a) Give an interpretation of $\mathcal{C}(1,0), \mathcal{C}(0,1)$ and $\mathcal{C}(1,1)$ as shortenings of $\mathcal{C}_{pub}$. *Hint: Recall that $x_1 = 0$ and $x_2 = 1$.*

(b) Deduce that they can be easily computed.

Q8. Assume $k \leq n/2$, and let i, j be integers such that $1 \leq i \leq k-2$ and $i+j \leq k-2$.

(a) Show that

$$\mathcal{C}(i+1,j) \star \mathcal{C}(i-1,j) = \mathcal{C}(i,j)^2$$

$$\text{and} \tag{8}$$

$$\mathcal{C}(i,j+1) \star \mathcal{C}(i,j-1) = \mathcal{C}(i,j)^2.$$

(b) Deduce an algorithm which takes as inputs generator matrices of $\mathcal{C}(i,j)$ and $\mathcal{C}(i-1,j)$ and recovers a basis of $\mathcal{C}(i+1,j)$ in time $O(k^2n^3 + k^3n^2)$ operations over $\mathbb{F}_q$.

1.4 Finally: the Attack!

For $i \leq k-1$, set $\mathcal{C}_i \stackrel{\text{def}}{=} \mathcal{C}(i,0)$.

Q9. Check that $\mathcal{C}_i$ indeed defines a filtration with the wanted properties, and that each term can actually be computed from previous ones.

Q10. What is the dimension of $\mathcal{C}_{k-1}$? What is the shape of a basis?

Q11. Consider the code $\mathcal{C}(k-2,1)$. What is its dimension? What does a basis look like?

Q12. Show that $\mathbf{x}$ can be easily recovered from a basis of $\mathcal{C}(k-1,0)$ and $\mathcal{C}(k-2,1)$.

Q13. Conclude the attack. What is the overall time complexity?

References

[CCMZ15] Igniacio Cascudo, Ronald Cramer, Diego Mirandola, and Gilles Zémor. “Squares of Random Linear Codes”. In: *IEEE Trans. Inform. Theory* 61.3 (Mar. 2015), pp. 1159–1173. ISSN: 0018-9448 (cit. on p. 3).

[CGGO+13] Alain Couvreur, Philippe Gaborit, Valérie Gautier, Ayoub Otmani, and Jean-Pierre Tillich. "Distinguisher-Based Attacks on Public-Key Cryptosystems Using Reed-Solomon Codes". In: *International Workshop on Coding and Cryptography - WCC 2013*. Bergen, Norway, Apr. 2013, pp. 181–193 (cit. on p. 1).

[CMT23] Alain Couvreur, Rocco Mora, and Jean-Pierre Tillich. "A new approach based on quadratic forms to attack the McEliece cryptosystem". In: *arXiv preprint arXiv:2306.10294* (2023) (cit. on p. 2).

[Nie86] Harald Niederreiter. "Knapsack-type cryptosystems and algebraic coding theory". In: *Problems of Control and Information Theory* 15.2 (1986), pp. 159–166 (cit. on p. 1).

[SS92] Vladimir Michilovich Sidelnikov and S.O. Shestakov. "On the insecurity of cryptosystems based on generalized Reed-Solomon codes". In: *Discrete Math. Appl.* 1.4 (1992), pp. 439–444 (cit. on p. 1).