An analysis of Coggia-Couvreur Attack on Loidreau's Rank-metric public-key encryption scheme in the general case

Pierre Loidreau[1] and Ba-Duc Pham[2]

[1] Univ Rennes, DGA MI, CNRS, IRMAR - UMR 6625, F-35000 Rennes, France
pierre.loidreau@univ-rennes1.fr
[2] Univ Rennes, IRMAR - UMR 6625, F-35000 Rennes, France
ba-duc.pham@univ-rennes1.fr

Abstract. In this paper we show that in the case where the public-key can be distinguished from a random code in Loidreau's encryption scheme, then Coggia-Couvreur attack can be extended to recover an equivalent secret key. This attack can be conducted in polynomial-time if the masking vector space has dimension 3, thus recovering the results of Ghatak.

Keywords: Rank metric codes, Gabidulin codes, code based cryptography, cryptanalysis

Introduction

Since the use of $\mathbb{F}_{q^m}$-linear rank metric permits to design a short public key encryption scheme, one of the directions of code based cryptography consists in instantiating McEliece encryption scheme [1] with codes in rank metric, [2, 3].

Because of the structure of Gabidulin codes, any cryptosystem instantiated with codes containing Gabidulin codes not sufficiently scrambled was attacked [4]. In 2017, Loidreau proposed a scheme based on Gabidulin codes masked with a small dimensional vector space [5]. If the dimension of the vector space is too small, then there exists a very simple polynomial-time distinguishing algorithm.

The question was to know if distinguishing is enough to break. Coggia and Couvreur [6] showed that in the case where the dimension of the masking space is 2, a decryption procedure can be recovered in polynomial-time. More recently, Ghatak [7] presented an extension of the Coggia–Couvreur attack to deal with secret matrices chosen over subspaces of dimension 3.

In this work, we show that this can be extended to any dimension and we can include the previous results. Moreover we are able to prove rigorously under some assumptions the efficiency of the attack.

1 The encryption scheme

1.1 Generalities

Let $\boldsymbol{G}$ a random generator matrix of a Gabidulin code $\mathcal{G}_k(\boldsymbol{g})$. Fix an integer $\lambda \leq m$ and an $\mathbb{F}_q$-vector subspace $\mathcal{V}$ of $\mathbb{F}_{q^m}$ of dimension λ. Let $\boldsymbol{P} \in \boldsymbol{GL}(n, \mathbb{F}_{q^m})$ whose entries are all in $\mathcal{V}$. Then, let

$$\boldsymbol{G}_{\text{pub}} = \boldsymbol{G}\boldsymbol{P}^{-1}$$

- KeyGen: Public key $(\boldsymbol{G}_{\text{pub}}, t)$ where $t = \left\lfloor \frac{n-k}{2\lambda} \right\rfloor$
  Secret key $(\boldsymbol{g}, \boldsymbol{P})$
- Encryption: Given a plaintext $\boldsymbol{m} \in \mathbb{F}_{q^m}^k$, choose $e \in \mathbb{F}_{q^m}^n$ of rank weight t. The ciphertext is:

$$\boldsymbol{c} = \boldsymbol{m}\boldsymbol{G}_{\text{pub}} + \boldsymbol{e}$$

- Decryption:
  - Compute $\boldsymbol{c}\boldsymbol{P} = \boldsymbol{m}\boldsymbol{G} + \boldsymbol{e}\boldsymbol{P}$.
  - Decode in $\mathcal{G}_k(\boldsymbol{g})$ and $\text{rk}(\boldsymbol{e}\boldsymbol{P}) \leq t\lambda \leq \frac{n-k}{2}$

Let us denote by $\mathcal{C}_{\text{pub}}$ the code generated by $\boldsymbol{G}_{\text{pub}}$ and by $\mathcal{C}_{\text{pub}}^{\perp}$, the dual code. Let $\boldsymbol{H}_{\text{pub}}$ be a generator matrix of $\mathcal{C}_{\text{pub}}^{\perp}$. It is immediate that

$$\boldsymbol{H}_{\text{pub}} = \boldsymbol{H}_{\text{sec}}\boldsymbol{P}^T$$

where $\boldsymbol{H}_{\text{sec}}$ is a parity-check matrix of $\mathcal{G}_k(\boldsymbol{g})$.

1.2 Goal of a reconstructing attack and solution set

Our main goal is to design a reconstructing attack from the knowledge of $\mathcal{C}_{\text{pub}}^{\perp}$ and under some particular sets of parameters.

W.l.o.g, one can suppose that $1 \in \mathcal{V}$. Suppose that $\mathcal{V} = \langle 1, \beta_1, \ldots, \beta_{\lambda-1} \rangle_{\mathbb{F}_q}$ for some $\{\beta_i\}_{i=1}^{\lambda-1} \in \mathbb{F}_{q^m} \backslash \mathbb{F}_q$. Therefore, $\boldsymbol{P}^T$ can be decomposed into

$$\boldsymbol{P}^T = \boldsymbol{P}_0 + \sum_{i=1}^{\lambda-1} \beta_i \boldsymbol{P}_i$$

where $\boldsymbol{P}_i$ are $n \times n$ matrices with entries in $\mathbb{F}_q$ not necessarily invertible.

Let $\mathcal{C}_{\text{sec}}^{\perp}$ the dual code of $\mathcal{G}_k(\boldsymbol{g})$. Thus, $\mathcal{C}_{\text{sec}}^{\perp} = \mathcal{G}_{n-k}(\boldsymbol{a})$ for some $\boldsymbol{a} \in \mathbb{F}_{q^m}^n$ with $\text{rk}(\boldsymbol{a}) = n$. We define

$$\boldsymbol{h}_0 = \boldsymbol{a}\boldsymbol{P}_0, \boldsymbol{h}_1 = \boldsymbol{a}\boldsymbol{P}_1, \ldots, \boldsymbol{h}_{\lambda-1} = \boldsymbol{a}\boldsymbol{P}_{\lambda-1}$$

To be convenient, we denote for $i \in \mathbb{Z}, [i] = q^i$

Lemma 1. *The code $\mathcal{C}_{pub}^{\perp}$ is spanned by $\boldsymbol{h}_0^{[i]} + \sum_{j=1}^{\lambda-1} \beta_j \boldsymbol{h}_j^{[i]}$ for $i = 0, \ldots, n-k-1$*

Let us define the so-called solution set of the encryption scheme

Definition 1 (Solution set). *The set $\mathcal{S}$ of all $(\boldsymbol{h}, \vec{\beta}) \in (\mathbb{F}_{q^m}^n)^\lambda \times \mathbb{F}_{q^m}^{\lambda-1}$ such that*

$$\mathcal{C}_{pub}^{\perp} = \left\langle \boldsymbol{h}_0^{[i]} + \sum_{j=1}^{\lambda-1} \beta_j \boldsymbol{h}_j^{[i]},\ i = 0, \ldots, n-k-1 \right\rangle \tag{1}$$

where $\forall j = 0, \ldots, \lambda-1,\ \boldsymbol{h}_j$ has rank n and $\langle 1, \beta_1, \ldots, \beta_{\lambda-1} \rangle_{\mathbb{F}_q}$ has dimension λ is called solution set of the encryption scheme.

It is obvious that finding an element of the solution set $\mathcal{S}$ implies the ability to design a polynomial-time decryption algorithm. What we call a reconstructing attack corresponds to finding an element in $\mathcal{S}$. The solution set $\mathcal{S}$ has the following properties.

Proposition 1. *Let $(\boldsymbol{h}, \vec{\beta}) \in (\mathbb{F}_{q^m}^n)^\lambda \times \mathbb{F}_{q^m}^{\lambda-1}$. Let $\boldsymbol{A} = (a_{j,i})_{j,i=0}^{\lambda-1} \in GL_\lambda(\mathbb{F}_q)$. Let us define the following group action on $(\mathbb{F}_{q^m}^n)^\lambda \times \mathbb{F}_{q^m}^{\lambda-1}$ by $\boldsymbol{A} \cdot (\boldsymbol{h}, \vec{\beta}) = (\boldsymbol{h}', \vec{\beta}')$ where*

$$\begin{cases} \boldsymbol{h}_j = \dfrac{a_{j,0}\boldsymbol{h}_0' + \sum_{i=1}^{\lambda-1} a_{j,i}\boldsymbol{h}_i'}{a_{0,0} + \sum_{i=1}^{\lambda-1} a_{i,0}\beta_i},\ j = 0, \ldots, \lambda-1 \\ \beta_j' = \dfrac{a_{0,j} + \sum_{i=1}^{\lambda-1} a_{i,j}\beta_i}{a_{0,0} + \sum_{i=1}^{\lambda-1} a_{i,0}\beta_i},\ j = 1, \ldots, \lambda-1 \end{cases}$$

1. *Then if $(\boldsymbol{h}, \vec{\beta}) \in \mathcal{S}$ we have $(\boldsymbol{h}', \vec{\beta}') = \boldsymbol{A} \cdot (\boldsymbol{h}, \vec{\beta}) \in \mathcal{S}$.*
2. *Moreover let $\overline{\boldsymbol{A}} = \{\boldsymbol{B} \in GL_\lambda(\mathbb{F}_q) | \exists c \in \mathbb{F}_q^*, \boldsymbol{B}\boldsymbol{A}^{-1} = c\boldsymbol{I}_\lambda\}$. Then, for any $\boldsymbol{B} \in \overline{\boldsymbol{A}}$, and for any $(\boldsymbol{h}, \vec{\beta}) \in (\mathbb{F}_{q^m}^n)^\lambda \times \mathbb{F}_{q^m}^{\lambda-1}$ we have*

$$\boldsymbol{A} \cdot (\boldsymbol{h}, \vec{\beta}) = \boldsymbol{B} \cdot (\boldsymbol{h}, \vec{\beta})$$

2 Attacks on the system

2.1 A distinguishing attack in the general case

If n, k, λ satisfy $k > \frac{(\lambda-1)n}{\lambda} + 1$, then one can distinguish the public-code from a random code in polynomial time by the following theorem.

Theorem 1 ([6], [7]). $\dim_{\mathbb{F}_{q^m}} \left(\mathcal{C}_{pub}^{\perp} + \mathcal{C}_{pub}^{\perp\,[1]} + \cdots + \mathcal{C}_{pub}^{\perp\,[\lambda]} \right) \leq \lambda(n-k) + \lambda$

Now the distinguishing attack comes from this proposition

Proposition 2 ([6] Proposition 2). *If $\mathcal{C}_{rand}$ is a random code of length n and dimension $n-k$, then for a non-negative integer a and a positive $\lambda < n-k$, we have*

$$\mathbb{P}\left(\dim_{\mathbb{F}_{q^m}} \left(\mathcal{C}_{rand} + \mathcal{C}_{rand}^{[1]} + \cdots + \mathcal{C}_{rand}^{[\lambda]} \right) \leq \min(n, (\lambda+1)(n-k)) - a \right) = O(q^{-ma}).$$

Now whenever $k > \frac{(\lambda-1)n}{\lambda} + 1$, the dimension of $\mathcal{C}_{rand} + \mathcal{C}_{rand}^{[1]} + \cdots + \mathcal{C}_{rand}^{[\lambda]}$ is very probably equal to $(\lambda+1)(n-k)$ whereas the dimension of $\mathcal{C}_{\text{pub}}^{\perp} + \mathcal{C}_{\text{pub}}^{\perp\ [1]} + \cdots + \mathcal{C}_{\text{pub}}^{\perp\ [\lambda]}$ is probably equal to $\lambda(n-k+1)$, which is strictly less than $(\lambda+1)(n-k)$.

2.2 Reconstructing attack

We suppose that the public code has rate larger than $(\lambda-1)/\lambda$, so that the distinguisher introduced in Section 2.1 works on it. Although the attack we describe should work heuristically, to have rigorous proofs of work we need the following assumptions, which are not very constraining

(1) There exists an element $(\boldsymbol{h}, \vec{\beta}) \in \mathcal{S}$ such that $\forall i_1, \ldots, i_\lambda \in \{1, \ldots, n-k-1\}$ distinct.
$$\det \begin{bmatrix} 1 & \beta_1^{[i_1]} & \beta_2^{[i_1]} & \ldots & \beta_{\lambda-1}^{[i_1]} \\ 1 & \beta_1^{[i_2]} & \beta_2^{[i_2]} & \ldots & \beta_{\lambda-1}^{[i_2]} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \beta_1^{[i_\lambda]} & \beta_2^{[i_\lambda]} & \ldots & \beta_{\lambda-1}^{[i_\lambda]} \end{bmatrix} \neq 0,$$
(2) $\dim_{\mathbb{F}_{q^m}} \mathcal{C}_{\text{pub}}^{\perp} + \mathcal{C}_{\text{pub}}^{\perp\ [1]} + \mathcal{C}_{\text{pub}}^{\perp\ [2]} + \cdots + \mathcal{C}_{\text{pub}}^{\perp\ [\lambda]} = \lambda(n-k) + \lambda$
(3) There is no $\boldsymbol{A} \in GL_\lambda(\mathbb{F}_q)$ such that
$$\beta_j = \frac{a_{0,j} + \sum_{i=1}^{\lambda-1} a_{i,j}\beta_i}{a_{0,0} + \sum_{i=1}^{\lambda-1} a_{i,0}\beta_i},\ j = 1, \ldots, \lambda-1$$

The main idea of this reconstructing attack is as follows:

Step 1. From $\mathcal{C}_{\text{pub}}^{\perp} = \left\langle \boldsymbol{h}_0^{[i]} + \sum_{j=1}^{\lambda-1} \beta_j \boldsymbol{h}_j^{[i]},\ i = 0, \ldots, n-k-1 \right\rangle$, find one dimensional vector-spaces $\mathcal{A}_i$ for $i = 1, \ldots, n-k-1$, such that any element $(\boldsymbol{h}, \vec{\beta}) \in \mathcal{S}$ satisfies:
$$\mathcal{A}_i = \left\langle \boldsymbol{h}_0 + \sum_{j=1}^{\lambda-1} \beta_j^{[-i]} \boldsymbol{h}_j \right\rangle$$

Step 2. From the linear relation between $\mathcal{A}_i, i = 0, \ldots, n-k-1$, create the system of $\lambda-1$ polynomial equations $\mathcal{P}_s(X)$ such that $\mathcal{P}_s(\vec{\beta}) = 0$. Afterwards, solve this system to find one root $\vec{\beta}'$.

Step 3. Recover $\boldsymbol{h}'$ corresponding to $\vec{\beta}'$ such that $(\boldsymbol{h}', \vec{\beta}') \in \mathcal{S}$ the set of solution of (1)

First step: Recovering one-dimensional vector spaces We now suppose that the three assumptions in section 2.2 are true we have the following theorem:

Theorem 2.

Let $d := n-k-\lambda+1$. Under the assumptions (1), (2), (3), the algorithm (1) returns the 1 *dimensional vector spaces*

$$\mathcal{A}_i = \left\langle \boldsymbol{h}_0 + \sum_{j=1}^{\lambda-1} \beta_j^{[-i]} \boldsymbol{h}_j \right\rangle, \ i = 0, \ldots, n-k-1$$

$\forall\ (\boldsymbol{h}, \vec{\beta}) \in \mathcal{S}$.

Algorithm 1: Recovering 1-dimensional vector spaces

Input: $\mathcal{C}_{\text{pub}}^{\perp}$, $\lambda \leq (n-k)/2$
Output: $\mathcal{A}_i$ for $i = 0, \ldots, n-k-1$

$\mathcal{S}_0 \leftarrow \mathcal{C}_{\text{pub}}^{\perp\ [0]} + \mathcal{C}_{\text{pub}}^{\perp\ [1]} + \cdots + \mathcal{C}_{\text{pub}}^{\perp\ [\lambda-1]}$
$\mathcal{A} \leftarrow \left(\bigcap_{i=0}^{d} \mathcal{S}_0^{[i]} \right)^{[-(n-k-\lambda+1)]}$
$\mathcal{D}_{\lambda-1} \leftarrow \mathcal{A} \cap \mathcal{C}_{\text{pub}}^{\perp\ [2\lambda-2-(n-k)]}$ and $\mathcal{B}_0 \leftarrow \mathcal{A} + \mathcal{D}_{\lambda-1}^{[1-\lambda]}$
$\mathcal{D}_0 \leftarrow \mathcal{B}_0 \cap \mathcal{C}_{\text{pub}}^{\perp\ [-1]}$
for $\ell \in 1, \ldots, \lambda-2$ **do**
    $\mathcal{B}_\ell \leftarrow \mathcal{A} + \sum_{j=0}^{\ell-1} \mathcal{D}_j^{[\ell-j]}$;
    $\mathcal{D}_\ell \leftarrow \mathcal{B}_\ell \cap \mathcal{C}_{\text{pub}}^{\perp\ [-1]}$
$\mathcal{H} \leftarrow \sum_{j=0}^{\lambda-1} \mathcal{C}_j^{[2-j-\lambda]}$
for $i \in 0, \ldots, n-k-1$ **do**
    Return $\mathcal{A}_i \leftarrow \mathcal{H} \cap \mathcal{C}_{\text{pub}}^{\perp\ [-i]}$

Second step: Recovering the vector space
From step 1, we recovered the 1-dimensional vector-spaces

$$\forall i = 0, \ldots, n-k-1, \ \mathcal{A}_i = \left\langle \boldsymbol{h}_0 + \sum_{j=1}^{\lambda-1} \beta_j^{[-i]} \boldsymbol{h}_j \right\rangle$$

The vector spaces $\mathcal{A}_i$ do not depend on $(\boldsymbol{h}, \beta) \in \mathcal{S}$. We introduce the following lemma.

Lemma 2. *For any $\boldsymbol{u}_0 \in \mathcal{A}_0$, and for any set $\mathcal{I} = \{i_1, \ldots, i_\lambda\} \subset \{1, \ldots, n-k-1\}$ of λ distinct elements, there exists a unique λ-tuple $\boldsymbol{u}^{\mathcal{I}} \stackrel{def}{=} (\boldsymbol{u}^{\mathcal{I}}_{i_1}, \boldsymbol{u}^{\mathcal{I}}_{i_2}, \ldots, \boldsymbol{u}^{\mathcal{I}}_{i_\lambda}) \in \bigtimes_{j=1}^{\lambda} \mathcal{A}_{i_j}$ such that $\sum_{i_j \in \mathcal{I}} \boldsymbol{u}^{\mathcal{I}}_{i_j} = \boldsymbol{u}_0$*

A vector $\boldsymbol{u}_0 \in \mathcal{A}_0$ can be written under the form $\boldsymbol{u}_0 = \alpha_{\boldsymbol{h},\beta}(\boldsymbol{h}_0 + \sum_{j=1}^{\lambda} \beta_j \boldsymbol{h}_j)$. From the structure of the solution space $\mathcal{S}$, there exists an $(\boldsymbol{h}, \beta) \in \mathcal{S}$ such that $\alpha_{\boldsymbol{h},\beta} = 1$. It means that we can fix $\boldsymbol{u}_0 := \boldsymbol{h}_0 + \sum_{j=1}^{\lambda} \beta_j \boldsymbol{h}_j$ as a known vector. Let

$$Mat^{\mathcal{I}}(\vec{X}) := \begin{bmatrix} 1 & X_1^{[i_1]} & X_2^{[i_1]} & \cdots & X_{\lambda-1}^{[i_1]} \\ 1 & X_1^{[i_2]} & X_2^{[i_2]} & \cdots & X_{\lambda-1}^{[i_2]} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & X_1^{[i_\lambda]} & X_2^{[i_\lambda]} & \cdots & X_{\lambda-1}^{[i_\lambda]} \end{bmatrix}$$

where $\vec{X} = (X_1, X_2, \ldots, X_{\lambda-1})$ is formed with the unknowns. For any $\mathcal{I} = \{i_1, \ldots, i_\lambda\}$, we denote $\boldsymbol{u}^{\mathcal{I}}_{i_\ell} = k^{\mathcal{I}}_{i_\ell} \left(\boldsymbol{h}_0 + \sum_{j=1}^{\lambda-1} \beta_j^{[-i_\ell]} \boldsymbol{h}_j \right)$. Since $\boldsymbol{h}_i$ are linearly independent over $\mathbb{F}_{q^m}$ and from Lemma 2 we have the following system of equations:

$$(k^{\mathcal{I}}_{i_1}, k^{\mathcal{I}}_{i_2}, \ldots, k^{\mathcal{I}}_{i_\lambda}) Mat^{\mathcal{I}}(\vec{\beta}) = (1, \beta_1, \beta_2, \ldots, \beta_{\lambda-1})$$

in the unknowns $k^{\mathcal{I}}_i$ and β_i. From assumption (1), $\det(Mat^{\mathcal{I}}(\vec{X})) \neq 0$, knowing the β_i's, the solution in $k^{\mathcal{I}}_i$ is unique. We define the multivariate polynomial

$$f^{\mathcal{I}}(\vec{X}) \stackrel{def}{=} \det(Mat^{\mathcal{I}}(\vec{X}))$$

By Cramer's rule, for any $j = 1, \ldots, \lambda$ we have

$$k^{\mathcal{I}}_{i_j} = \frac{f^{-(\mathcal{I} \setminus \{i_j\}) \cup \{0\}}(\vec{\beta})}{f^{-\mathcal{I}}(\vec{\beta})}, \tag{2}$$

where $\vec{\beta} = (\beta_1, \ldots, \beta_\lambda)$.

Let us define $\mathcal{J}_s = (\{1, \ldots, \lambda+1\}) \setminus \{s+1\}$, for all $s = 1, \ldots, \lambda$. From (2), we have

$$\forall s \in \{1, \ldots, \lambda\},\ k_1^{\mathcal{J}_s} = \frac{f^{-(\mathcal{J}_s \setminus \{1\}) \cup \{0\}}(\vec{\beta})}{f^{-\mathcal{J}_s}(\vec{\beta})}$$

Now since we know only the vector space $\mathcal{A}_1$ and not the exact vectors $\boldsymbol{h}_0 + \sum_{j=1}^{\lambda-1} \beta_j^{[-1]} \boldsymbol{h}_j$, we do not know $k_1^{\mathcal{J}_s}$. However, we can compute the quantity $k_1^{\mathcal{J}_\lambda} / k_1^{\mathcal{J}_s}$ for $s \in \{1, \ldots, \lambda-1\}$ thank to algorithm 2 and Lemma 2.

Algorithm 2: Determining quotient $k_1^{\mathcal{J}_\lambda}/k_1^{\mathcal{J}_s}$

Input: $\{\mathcal{A}_i\}_{i=1}^{n-k-1}$, $\{\mathcal{J}_s\}_{s=1}^{\lambda}$ and the vector $\boldsymbol{u}_0 \in \mathcal{A}_0$

Output: $\alpha_s = k_1^{\mathcal{J}_\lambda}/k_1^{\mathcal{J}_s}$ for $s \in \{1, \ldots, \lambda-1\}$

1 For $i = 1, \ldots, n-k-1$, fix $\boldsymbol{u}_i$ arbitrarily in $\mathcal{A}_i$

2 For $s = 1, \ldots, \lambda$, find $a_j^{\mathcal{J}_s} \in \mathbb{F}_{q^m}$ such that of $\sum_{j \in \mathcal{J}_s} a_j^{\mathcal{J}_s} \boldsymbol{u}_j = \boldsymbol{u}_0$

3 Return $\dfrac{a_1^{\mathcal{J}_\lambda}}{a_1^{\mathcal{J}_s}}$, for $s = 1, \ldots, \lambda-1$

Now let us define by $\alpha_s = (k_1^{\mathcal{J}_\lambda}/k_1^{\mathcal{J}_s})^{[\lambda+1]}$, for $s = 1, \ldots, \lambda-1$. To simplify notations, we also define

$$\forall s \in \{1, \ldots, \lambda\} \begin{cases} \mathcal{L}_s = (\lambda+1) - (\mathcal{J}_s \setminus \{1\} \cup \{0\}) \\ \mathcal{M}_s = (\lambda+1) - \mathcal{J}_s \end{cases}$$

We obtain the set of equations

$$\forall s \in \{1, \ldots, \lambda-1\}, \quad f^{\mathcal{L}_\lambda}(\vec{\beta}) f^{\mathcal{M}_s}(\vec{\beta}) - \alpha_s f^{\mathcal{M}_\lambda}(\vec{\beta}) f^{\mathcal{L}_s}(\vec{\beta}) = 0$$

Let

$$\mathcal{F}_s(\vec{X}) \stackrel{def}{=} f^{\mathcal{L}_\lambda}(\vec{X}) f^{\mathcal{M}_s}(\vec{X}) - \alpha_s f^{\mathcal{M}_\lambda}(\vec{X}) f^{\mathcal{L}_s}(\vec{X}) \in \mathbb{F}_{q^m}[\vec{X}].$$

The polynomial $\mathcal{F}_s$ has degree $q^{\lambda+1} + q^\lambda + 2\sum_{j=1}^{\lambda-1} q^j + 1 - q^{\lambda-s}$

This gives us a multivariate polynomial system over $\mathbb{F}_{q^m}$ for which $\vec{\beta}$ is a solution. However, from our hypotheses we can do better and even reduce the degrees of the polynomials. Since $\beta_1, \ldots, \beta_\lambda$ are linearly independent they cannot be roots of linear factors over $\mathbb{F}_q$ of $\mathcal{F}_s$. Therefore we can reduce for all s the polynomial $\mathcal{F}_s(\vec{X})$ by its $\mathbb{F}_q$-linear factors.

Lemma 3. *Let us define*

$$f_0(\vec{X}) = \prod_{a \in \mathbb{F}_q} (X_1 + a) \prod_{i=2}^{\lambda-1} \left(\prod_{a_0, \ldots a_{i-1} \in \mathbb{F}_q} (X_i + \sum_{j=1}^{i-1} a_j X_j + a_0) \right)$$

For any set $\mathcal{I} = \{i_1, \ldots, i_\lambda\}$ of λ distinct elements and $i_1 = \min(\mathcal{I})$, $f^{\mathcal{I}}(\vec{X})$ is divisible by $\left(f_0(\vec{X})\right)^{[i_1]}$. By the construction of $\mathcal{L}_s, \mathcal{M}_s$, we have $f^{\mathcal{L}_\lambda}(\vec{X})$ and $f^{\mathcal{M}_\lambda}(\vec{X})$ is divisible by $\left(f_0(\vec{X})\right)^q$, for all $s \in \{1, \ldots, \lambda-1\}$, $f^{\mathcal{L}_s}(\vec{X})$ and $f^{\mathcal{M}_s}(\vec{X})$ is divisible by $f_0(\vec{X})$. This gives us a new polynomial system for which $\vec{\beta}$ is also a solution, but the degree is reduced.

$$\mathcal{P}_s(\vec{X}) = \frac{\mathcal{F}_s(\vec{X})}{\left(f_0(\vec{X})\right)^{q+1}}$$

Lemma 4. *Let* $\boldsymbol{A} = (a_{i,j})_{i=0,j=0}^{\lambda-1,\lambda-1} \in \boldsymbol{PGL}(\lambda;\mathbb{F}_q)$. *Consider the transformation on* $f^{\mathcal{I}}(\vec{X})$ *defined on* $\vec{X} = (X_1,\ldots,X_{\lambda-1})$ *by*

$$\forall j \in \{1,\ldots,\lambda-1\}, \quad X_j \longmapsto \frac{a_{0,j} + \sum_{i=1}^{\lambda-1} a_{i,j}X_i}{a_{0,0} + \sum_{i=1}^{\lambda-1} a_{i,0}X_i}$$

We denote $D = a_{0,0} + \sum_{i=1}^{\lambda-1} a_{i,0}X_i$ *then the polynomial* $f^{\mathcal{I}}(\vec{X})$ *is transformed into*

$$f^{\mathcal{I}}(\vec{X}) \longmapsto \boldsymbol{A}.f^{\mathcal{I}}(\vec{X}) \stackrel{def}{=} \frac{\Delta_{\boldsymbol{A}}}{D^{\deg(f^{\mathcal{I}})}} f^{\mathcal{I}}(\vec{X})$$

where $\Delta_{\boldsymbol{A}}$ *is the determinant of* $\boldsymbol{A}$. *As a consequence,*

$$\mathcal{P}_s(\vec{X}) \mapsto \frac{1}{D^{q^{\lambda+1}-q^{\lambda-s}}} \mathcal{P}_s(\vec{X})$$

We therefore have

Proposition 3. *If there isn't any common factor between the polynomials* $\mathcal{P}_s(\vec{X})$, *then for any* $(\boldsymbol{h},\vec{\beta}) \in \mathcal{S}$, *the vector* $\vec{\beta} = (\beta_1,\ldots,\beta_{\lambda-1})$ *is a solution to the polynomial system*

$$\forall s = 1,\ldots,\lambda-1, \quad \mathcal{P}_s(\vec{X}) = 0 \tag{3}$$

Proof. If there isn't any common factor between the polynomials $\mathcal{P}_s(\vec{X})$ then the number of roots is at most the product of the total degree of polynomials $\mathcal{P}_s(\vec{X})$, which is $\prod_{j=1}^{\lambda-1}(q^{\lambda+1}-q^j) = |\mathbf{PGL}(\lambda,\mathbb{F}_q)|$ (Bézout bound).

Any element in the orbit of a solution $\vec{\beta}$ under the group action of $\mathbf{PGL}(\lambda,\mathbb{F}_q)$ is again root of the system. From Assumption (3) the orbit of $\vec{\beta}$ under $\mathbf{PGL}(\lambda,\mathbb{F}_q)$ has cardinality $= |\mathbf{PGL}(\lambda,\mathbb{F}_q)|$ which means that the stabiliser of $\vec{\beta}$ with respect to this group action is trivial. In that case any root of the system (3) corresponds to an element of $\mathcal{S}$. □

Final step: We point out the key steps in the Coggia-Couvreur attack for λ as follows. To be convenient, we denote known elements by blue color and unknown elements by red color. Now from a solution $\vec{\beta'} = \beta'_1,\ldots,\beta'_{\lambda-1}$ to (3), we aim at finding the corresponding vector $\vec{\boldsymbol{h'}} = \boldsymbol{h}'_0,\ldots,\boldsymbol{h}'_{\lambda-1} \in (\mathbb{F}_{q^m}^n)^\lambda$ such that $(\vec{\boldsymbol{h'}},\vec{\beta'}) \in \mathcal{S}$.

1. For $\mathcal{I} = \{1,\ldots,\lambda\}$, since $\vec{\beta'}$ is known, $k_i = \dfrac{f^{-(\mathcal{I}\setminus\{i\})\cup\{0\}}(\vec{\beta'})}{f^{-\mathcal{I}}(\vec{\beta'})}, i = 1,\ldots,\lambda$ can be computed. Moreover, from the Lemma 2, there exists a unique λ-tuple $\boldsymbol{u}_{\mathcal{I}} = (\boldsymbol{u}_1,\ldots,\boldsymbol{u}_\lambda) \in \bigtimes_{i=1}^{\lambda} \mathcal{A}_i$ such that $\sum_{i=1}^{\lambda} \boldsymbol{u}_i = \boldsymbol{u}_0$, so we can compute

$\boldsymbol{h}'_0 + \sum_{j=1}^{\lambda-1} \beta_j'^{[-i]} \boldsymbol{h}'_j = \frac{\boldsymbol{u}_i}{k_i}, i = 1, \ldots, \lambda$. Thus,

$$(\boldsymbol{h}'_0, \ldots, \boldsymbol{h}'_{\lambda-1}) \begin{bmatrix} 1 & 1 & \ldots & 1 \\ \beta_1^{[-1]} & \beta_1^{[-2]} & \ldots & \beta_1^{[-\lambda]} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{\lambda-1}^{[-1]} & \beta_{\lambda-1}^{[-2]} & \ldots & \beta_{\lambda-1}^{[-\lambda]} \end{bmatrix} = \left(\frac{\boldsymbol{u}_1}{k_1}, \frac{\boldsymbol{u}_2}{k_2}, \ldots, \frac{\boldsymbol{u}_\lambda}{k_\lambda} \right)$$

It implies to a linear system of λ equation and λ unknowns which are vectors $\boldsymbol{h}'_0, \ldots, \boldsymbol{h}'_{\lambda-1}$ and the determinant of the matrix of coefficients is non-zero.

2. After recovering an alternate key of the form $(\boldsymbol{h}'_0, \ldots, \boldsymbol{h}'_{\lambda-1}, \beta'_1, \ldots, \beta'_{\lambda-1})$, we can compute the dual code $\mathcal{C}_{\text{pub}}^{\perp}$ and hence decrypt the ciphertext.

2.3 Complexity of reconstructing attack

This part shows the complexity of the attack by giving the number of operation in $\mathbb{F}_{q^m}$. Let ω be the exponent of the complexity of linear algebra operations. The Frobenius map costs $O(\log q)$ operations.

Step 1.

- Computation of dual code $\mathcal{C}_{\text{pub}}^{\perp}$ costs $O(n^\omega)$ operations.
- Computation of $\mathcal{C}_{\text{pub}}^{\perp\,[i]}, \forall i = 1, \ldots, n-k+1$ costs $O(n^3 \log q)$ operations.
- Computation $S_j = \sum_{i=j}^{j+\lambda-1} \mathcal{C}_{\text{pub}}^{\perp\,[i]}$ uses Gaussian elimination, so it costs $O(n^\omega)$.

Thus, computation $\bigcap_{i=0}^{n-k-\lambda+1} S_j$ costs $O(n^{\omega+1})$.

Step 2.

- Computation $(u_1^{\mathcal{I}}, \ldots, u_\lambda^{\mathcal{I}})$ represents the resolution of a linear system λ unknowns and n equations, which can be can done by computing QR decomposition. This computation costs $2n\lambda^2 - 2/3\lambda^3$ operations (Section 5.3.3 [8]). This computation should be performed $O(n)$ times, so it costs $O(n^2)$ operations.
- Complexity of finding a root of a system of polynomial equations
  - In case $\lambda = 3$, it includes the complexity of finding a root of a polynomial of degree d by Cantor–Zassenhaus algorithm [9] which costs $\tilde{O}(d^2 m \log q)$ operations in $\mathbb{F}_{q^m}$ and the computation of resultant of bivariate polynomials which can be done in $(d^{3-1/\omega})^{1+o(1)}$ [10] for $d = (q^4 - q)(q^4 - q^2)$ the number of roots.
  - Since the number of roots of the system (3) reaches the Bézout's bound, the complexity of solving this system is polynomial in $d = \prod_{j=1}^{\lambda-1} (q^{\lambda+1} - q^j)$ the number of solutions ([11], [12], [13], [14]).

Step 3. A finite number of linear systems solving costs $O(n^\omega)$.

Summary. For $m = O(n)$, overall cost of the attack is $O(n^3 \log q + n^{\omega+1}) + d^{O(1)}$ for $d = \prod_{j=1}^{\lambda-1} (q^{\lambda+1} - q^j)$ the number of solutions.

Conclusion We generalised the Coggia and Couvreur attack [6] for Loidreau's cryptosystem [5] for any λ and analysed its complexity.

The parameters of (k, n), which $R_{pub} \geq 1 - 1/\lambda$ should be avoided in Loidreau's scheme. In the future, it will be worthwhile to attempt a modification of the attack to work for lower rate codes $R_{pub} < 1 - 1/\lambda$ as well.

References

1. R. J. McEliece, "A Public-Key Cryptosystem Based On Algebraic Coding Theory," *Deep Space Network Progress Report*, vol. 44, pp. 114–116, Jan. 1978.
2. E. M. Gabidulin, A. V. Paramonov, and O. V. Tretjakov, "Ideals over a non-commutative ring and their application in cryptology," in *Advances in Cryptology — EUROCRYPT '91* (D. W. Davies, ed.), (Berlin, Heidelberg), pp. 482–489, Springer Berlin Heidelberg, 1991.
3. P. Gaborit, G. Murat, O. Ruatta, and G. Zemor, "Low Rank Parity Check codes and their application to cryptography," in *The International Workshop on Coding and Cryptography (WCC 13)* (L. Budaghyan, T. Helleseth, and M. G. Parker, eds.), (Bergen, Norway), p. 13 p., Apr 2013. ISBN 978-82-308-2269-2.
4. R. Overbeck, "Structural attacks for public key cryptosystems based on gabidulin codes," *J. Cryptology*, vol. 21, pp. 280–301, 2008.
5. P. Loidreau, "A new rank metric codes based encryption scheme," in *PQCrypto 2017* (T. Lange and T. Takagi, eds.), vol. 10346 of *Lecture Notes in Computer Science*, (Utrecht, Netherlands), pp. 3–17, Springer, June 2017.
6. D. Coggia and A. Couvreur, "On the security of a Loidreau's rank metric code based encryption scheme," in *WCC 2019 - Workshop on Coding Theory and Cryptography*, (Saint Jacut de la mer, France), Mar. 2019.
7. A. Ghatak, "Extending coggia-couvreur attack on loidreau's rank-metric cryptosystem," *Designs, Codes and Cryptography*, vol. 90, pp. 215–238, 2022.
8. G. H. Golub and C. F. Van Loan, *Matrix computations*. JHU press, 2013.
9. A. Bostan, F. Chyzak, M. Giusti, R. Lebreton, G. Lecerf, B. Salvy, and É. Schost, *Algorithmes Efficaces en Calcul Formel*. Palaiseau: Frédéric Chyzak (auto-édit.), Sept. 2017. 686 pages. Imprimé par CreateSpace. Aussi disponible en version électronique.
10. G. Villard, "On computing the resultant of generic bivariate polynomials," in *Proceedings of the 2018 acm international symposium on symbolic and algebraic computation*, pp. 391–398, 2018.
11. Y. N. Lakshman and D. Lazard, "On the complexity of zero-dimensional algebraic systems," in *Effective methods in algebraic geometry*, pp. 217–225, Springer, 1991.
12. J.-C. Faugere, P. Gianni, D. Lazard, and T. Mora, "Efficient computation of zero-dimensional gröbner bases by change of ordering," *Journal of Symbolic Computation*, vol. 16, no. 4, pp. 329–344, 1993.
13. J.-C. Faugère, P. Gaudry, L. Huot, and G. Renault, "Sub-cubic change of ordering for gröbner basis: a probabilistic approach," in *Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation*, pp. 170–177, 2014.
14. J. van der Hoeven and G. Lecerf, "On the complexity exponent of polynomial system solving," *Foundations of Computational Mathematics*, vol. 21, no. 1, pp. 1–57, 2021.