Designing an efficient and secure public-key cryptosystem based on reducible rank codes

Thierry Berger[1] and Pierre Loidreau[2]

[1] LACO, Université de Limoges, France,
Thierry.Berger@unilim.fr
[2] Ecole Nationale Supérieure de Techniques Avancées (ENSTA), France,
Pierre.Loidreau@ensta.fr

Abstract. In this paper we modify the cryptosystem presented in [16] based on the problem of decoding in rank metric. We design a cryptosystem more secure than the original one with a better transmission rate. We show that this system resists to the *message resend* and *reaction* attacks, and can be used with a small public-key (around 10kbits).

1 Introduction

Cryptosystems based on coding theory were first introduced by McEliece in 1978 [13], just a few months after the RSA cryptosystem was published. This system uses as private-key a Goppa code known to have fast polynomial-time decoding algorithms up to its error-correcting capability, and two scrambling non-singular matrices. The public-key is a generator matrix of the scrambled Goppa code.

Despite quite a number of attempts since 1978, the original system remains unbroken. However it is not widely used. Namely, the main problem of cryptosystems basing their security on the problem of decoding in Hamming metric is that the efficiency of general decoding algorithms implies that the size of the public-key has to be huge. Typically several hundred thousands of bits [2]. In 1991 a new public-key cryptosystem was presented by Gabidulin, Paramonov and Tretjakov. Its security relies on the problem of decoding codes in the rank metric [9]. It was a very promising system, since decoding algorithms in rank metric are exponential [14] . Therefore, this would allow a conceiver to choose public-keys of much smaller size. Unfortunately, since Gabidulin codes are strongly structured, Gibson showed that the public-key size had to be increased a lot. Hence it diminishes the practical interest of this system [10, 11].

The last developments in this field are not older than 2003 [16]. A new family of codes decodable in polynomial-time for rank metric was built. These codes are called *reducible rank codes*. This family of codes was used in a Niederreiter type cryptosystem.

In this paper, we design an efficient – in size and speed – public-key cryptosystem based on the family of reducible rank codes. Compared to the original system, we show that we can significantly increase the transmission rate of the system with a simple transformation using properties of the Rijndael S-boxes.

Moreover, we show that this new system can efficiently resist *message resend* attacks as well as *reaction attacks*, together with keeping a small public-key size. To achieve this goal, the designer only needs properties of rank metric and a *good* hash function.

2 Description of the cryptosystem

In a first part, we present rank metric and some of its properties. In a second part, we introduce the problem of of decoding in rank metric. By the state of art this problem is difficult. The complexity of the best decoding algorithms show that it is theoretically possible to use public-keys of much smaller size than for PKCs (Public-Key Cryptosystems) based on problem of decoding in Hamming metric, like McEliece cryptosystem. In a third part, we describe the cryptosystem based on reducible rank codes as it was introduced in [16]. We only present the simplified version of the paper, since it is resistant to any kind of structural attacks as shown in the original paper.

2.1 Rank metric

Rank metric was introduced in 1985 by E.M. Gabidulin [6]. Let $\mathrm{GF}(2^m)$ be the finite field with 2^m elements. Any element α of $\mathrm{GF}(2^m)$ can be written uniquely

$$\alpha = a_1\gamma_1 + \cdots + a_m\gamma_m,$$

where the $a_i \in \mathrm{GF}(2)$ and where $\gamma_1, \ldots, \gamma_m$ is a basis of $\mathrm{GF}(2^m)$ over $\mathrm{GF}(2)$. Any vector $\mathbf{c} = (c_1, \ldots, c_n)$ over $\mathrm{GF}(2^m)$ can be seen as an $m \times n$ matrix, whose ith column is the vector of size m corresponding to the expansion of the element c_i over the chosen basis. The rank of the vector c is by definition the rank of the obtained matrix. In the following it is denoted $\mathrm{Rk}(\mathbf{c} \mid \mathrm{GF}(2))$.

Consider n elements $g_1, \ldots, g_n$ of $\mathrm{GF}(2^m)$ which are linearly independent over $\mathrm{GF}(2)$. Consider the $k \times n$–matrix

$$G = \begin{pmatrix} g_1 & \cdots & g_n \\ g_1^{2^1} & \cdots & g_n^{2^1} \\ g_1^{2^2} & \cdots & g_n^{2^2} \\ \vdots & \ddots & \vdots \\ g_1^{2^{k-1}} & \cdots & g_n^{2^{k-1}} \end{pmatrix}. \tag{1}$$

Let $\mathcal{G}$ be the code generated by G, *i.e.*

$$\mathcal{G} = \{\mathbf{x}G \mid \mathbf{x} \in \mathrm{GF}(2^m)^k\}.$$

The code $\mathcal{G}$ is called *Gabidulin code*. Several polynomial-time algorithms correcting up to $t = \lfloor (n-k)/2 \rfloor$ errors in the rank metric were designed, [6, 7, 21, 17]. Our measurements of complexity will consider that we take the decoding algorithm described in [7], giving a decoding complexity of $\approx t^3 + (2n+m)t + k^2$ multiplications in $\mathrm{GF}(2^m)$.

2.2 Cryptosystems based on rank codes

Since 1978, cryptosystems, identification schemes and even a signature scheme have been designed, the security of which relies on the problem of decoding linear codes in Hamming metric [13, 3]. This problem has been studied for a long time and some results about NP-Completeness have been obtained. The most recent can be found in [24]. In the same way, public-key cryptosystems and identification schemes have been designed on the problem of decoding linear codes in rank metric. Though the problem is believed to be hard, there is no existing proof of NP-Completeness [5, 9]. The problem of decoding in rank metric can be stated as such:

Input: A target vector $\mathbf{y}$ of length n over GF(2^m), a $k \times n$ matrix G of rank k over GF(2^m), and an integer t.
Decoding$(\mathbf{y}, G, t)$
Find, whenever it exists, a vector $\mathbf{x} \in GF(2^m)^k$*, and a vector* $\mathbf{e}$ *where* $Rk(\mathbf{e}|GF(2)) \leq t$ *such that* $\mathbf{y} = \mathbf{x}G + \mathbf{e}$,

The most efficient algorithms that solve this problem can be found in [14]. In this paper, Ourivski and Johannson present two algorithms that, given a linear code C of dimension k over GF(2^m), gives a solution to **Decoding**$(\mathbf{y}, G, t)$ with complexity

- *Minimum rank weight decoding*: $O\left((tm)^3 2^{(t-1)(k+1)}\right)$ binary operations.
- *Basis enumeration*: $O\left((k+t)^3 t^3 2^{(t-1)(m-t)}\right)$ binary operations.

Since these algorithms are strongly exponential in the dimension, minimum distance and extension degree of the codes, the **Decoding** problem in rank metric can be considered as a difficult problem. Therefore it could be suitable for designing PKCs.

Compared to the general decoding algorithms for Hamming metric, the decoding algorithms for rank metric have larger complexity for the same parameters. This implies that we should be able, at least theoretically, to design systems with smaller public-keys for a given security. This would give a nice solution to the major drawback of using cryptosystems based on linear codes that is to know the huge size of the public-key (several hundreds of thousands of bits for McEliece system to be secure against general decoding algorithms, [2]).

Unfortunately, in rank metric, the only families of known codes that are decodable in polynomial-time are constructed from Gabidulin codes. This family is very much structured and, although it is possible to hide the structure in some sense, Gibson showed that the public-key size had to be widely increased to prevent an attacker from breaking completely the system [10, 11].

Different alternatives were proposed to reduce further the public-key size [8, 15] with keeping a sufficient security. The general principle is the following:

- The conceiver chooses a generator matrix G of a code for which he knows a polynomial-time decoding algorithm up to some rank t.

– He scrambles the generator matrix G of the code in some way and then gets the matrix G_{pub}, that he publishes. He uses this matrix to encrypt the message in the same way as for the McEliece cryptosystem.

2.3 System based on reducible rank codes

Among the proposed PKC's based on rank metric, the newest, and maybe most original one uses the so called family of *reducible rank codes*. In the introductory paper a more general approach than ours is presented, [16]. However, since we are mainly interested in the optimality and the efficiency of the system we will only consider reducible rank codes of order 2, without the scrambling matrix. This does not diminish the security of the system.

– Let G be the matrix presented in (1). It generates a Gabidulin code of length n and dimension k. The code corrects $t = \lfloor (n-k)/2 \rfloor$ errors in rank metric in polynomial-time. Let A be a randomly chosen $k \times n$ matrix over GF(2^m). The designer forms the $2k \times 2n$ matrix G_{priv} such that

$$G_{priv} = \begin{pmatrix} G & A \\ 0 & G \end{pmatrix}.$$

– Then he picks up randomly a $2k \times 2k$ non-singular matrix S over GF(2^m) and a $2n \times 2n$ non-singular matrix P with coefficient over the base field GF(2).

The conceiver publishes the $2k \times 2n$-matrix

$$G_{pub} = SG_{priv}P.$$

The encryption–decryption procedure is the following.

– *Encryption:* Alice wants to send the information vector $\mathbf{x}$ of length $2k$ to Bob. She first chooses an error-vector $\mathbf{e}$ over GF(2^m) of length $2n$ and of rank t. Then she computes

$$\mathbf{y} = \mathbf{x}G_{pub} + \mathbf{e},$$

and sends $\mathbf{y}$ to Bob.

– *Decryption:* Since Bob knows the private key, he can compute

$$\mathbf{y}P^{-1} = \mathbf{x}S \begin{pmatrix} G & A \\ 0 & G \end{pmatrix} + \mathbf{e}P^{-1}.$$

For more convenience, let us write $\mathbf{x}' = \mathbf{x}S$, $\mathbf{e}' = \mathbf{e}P^{-1}$, and $\mathbf{y}' = \mathbf{y}P^{-1}$. The vector $\mathbf{x}'$ has length $2k$, thus $\mathbf{x}' = (\mathbf{x}'_1, \mathbf{x}'_2)$ where the $\mathbf{x}'_i$'s are of length k. Similarly we have $\mathbf{y}' = (\mathbf{y}'_1, \mathbf{y}'_2)$ where $\mathbf{y}'_i$ is of length n, and $\mathbf{e}' = (\mathbf{e}'_1, \mathbf{e}'_2)$. Then, Bob obtains the two following equations:

$$\begin{cases} \mathbf{y}'_1 = \mathbf{x}'_1 G + \mathbf{e}'_1, \\ \mathbf{y}'_2 = \mathbf{x}'_2 G + \mathbf{x}'_1 A + \mathbf{e}'_2. \end{cases}$$

The matrix P has coefficients in the base field GF(2). Therefore, multiplying by P^{-1} preserves the rank of the vectors. Since $\mathbf{e}$ has rank less than t, $\mathbf{e}' = \mathbf{e}P^{-1}$ has also rank less than t. As a consequence $\mathbf{e}'_1$ and $\mathbf{e}'_2$ have rank less than t. Hence:

1. By decoding $\mathbf{y}'_1$ in the code generated by G, Bob recovers $(\mathbf{x}'_1, \mathbf{e}'_1)$.
2. Knowing $\mathbf{x}'_1$ and A, he computes $\mathbf{y}_2 - \mathbf{x}'_1 A$. Then he decodes this vector in the code generated by G and recovers $(\mathbf{x}'_2, \mathbf{e}'_2)$.
3. Finally he gets the plaintext $\mathbf{x}$ by multiplying $(\mathbf{x}'_1, \mathbf{x}'_2)$ by S^{-1}.

Because of the small parameters used and because of the efficiency of the decoding algorithms for Gabidulin codes, this procedure is extremely fast.

3 The new system

In the original proposition, the authors use the Niederreiter form of the system, that is using the parity-check matrix. With this method, they can publish only the redundant part of the parity-check matrix, diminishing thus the key-size, which could not be done in the McEliece system without some information from the plaintext leaking out. They proposed a system taken over the field GF(2^{20}) with a key-size of 16000 bits and a transmission rate of 0.55.

But there are two problems:

- The Niederreiter form of the system is not suitable against active attacks as we will mention further in the paper. Namely, a plaintext will always be encrypted into the same ciphertext, which is not the case when one uses the McEliece form, *i.e.* with the generator matrix. Therefore, there is absolutely no semantic security, since by making the difference of two received ciphertexts, an attacker can distinguish whenever the same message has been sent twice.
- The transmission rate is low. It is approximately equal to $(m+2n-t)t/(2m(n-k))$, which is much less than 1.

In the following, we show how to design a system based on reducible rank codes wich satisfies a *kind of* semantic security. We show how to increase significantly the transmission rate by designing a procedure that puts information in the error vector. By continuing our comparison with McEliece cryptosystem, adding information in the error-vector was done by Sendrier in [22], and the idea of rendering the scheme secure against message resend attacks can be found in [23].

In the modification of the system, we consider two things in addition to the standard parameters of the cryptosystem:

1. A hash function h taking as input $m \times 2n$ bits and returning $m \times 2k$ bits.
2. A procedure called $\mathcal{P}$ which takes as arguments a random vector r, an information vector $\widetilde{\mathbf{x}}$, and returns a vector of length $2n$ over GF(2^m) and of rank t. We want our procedure to satisfy three properties:

(a) Be invertible on its image.
(b) The procedures $\mathcal{P}$ and $\mathcal{P}^{-1}$ have to be computable in a time negligible compared to the time of the encryption-decryption of the system.
(c) The procedure $\mathcal{P}$ must diffuse the randomness r *sufficiently* over its output vector.

With these tools, we design an encryption–decryption procedure slightly different from the original one:

- *Encryption:* Alice wants to encrypt the information vectors $\mathbf{x}, \widetilde{\mathbf{x}}$. She computes $\mathbf{y} = (\mathbf{x} + h(\mathcal{P}(r, \widetilde{\mathbf{x}})))G_{pub} + \mathcal{P}(r, \widetilde{\mathbf{x}})$. Finally she sends $\mathbf{y}$ to Bob.
- *Decryption:* Since the vector $\mathcal{P}(r, \widetilde{\mathbf{x}})$ has rank less than t, Bob recovers $\mathcal{P}(r, \widetilde{\mathbf{x}})$ and $\mathbf{x} + h(\mathcal{P}(r, \widetilde{\mathbf{x}}))$. Since $\mathcal{P}$ is invertible, he recovers $\mathbf{x}$ and $\widetilde{\mathbf{x}}$.

The security does not rely exactly on the decoding problem in rank metric. It is more related with the following problem, which can be stated as **Conditional Decoding**.

Input: A target vector $\mathbf{y}$ of length n over GF(2^m), a $k \times n$ matrix G of rank k over GF(2^m), and an integer t.
Conditional Decoding$(\mathbf{y}, G, t)$
Find, whenever it exists, a vector $\mathbf{x}$*, and a vector* $\mathbf{e}$ *where* $Rk(\mathbf{e}|GF(2)) \leq t$ *such that* $\mathbf{y} = (\mathbf{x} + h(\mathbf{e}))G + \mathbf{e}$,

We show that both problem are completely equivalent. This implies that the problem on which the security of our system is based is as difficult as the **Decoding** problem. Namely, suppose that we are given an algorithm $\mathcal{A}$ solving **Conditional Decoding**, *i.e.* $\mathcal{A}$ takes as input an instance $(\mathbf{y}, G, t)$ and returns in polynomial-time:

- *Failure*, if there is no solution to **Conditional Decoding**$(\mathbf{y}, G, t)$.
- If there is no failure it returns a pair $(\mathbf{x}, \mathbf{e})$, such that $\mathbf{y} = (\mathbf{x} + h(\mathbf{e}))G + \mathbf{e}$, and $\mathrm{Rk}(\mathbf{e} \mid \mathrm{GF}(2)) \leq t$.

Let $\mathcal{A}'$ be the following algorithm, taking as input $(\mathbf{y}, G, t)$ and returning:

- *Failure*, if $\mathcal{A}(\mathbf{y}, G, t)$ returns *Failure*.
- The pair $(\mathbf{x} - h(\mathbf{e}), \mathbf{e})$ if $\mathcal{A}(\mathbf{y}, G, t)$ returns $(\mathbf{x}, \mathbf{e})$.

The algorithm $\mathcal{A}'$ solves the **Decoding** problem in polynomial time for the instance $(\mathbf{y}, G, t)$. Namely,

- There is no solution to **Decoding**$(\mathbf{y}, G, t)$ if and only if there is no solution to **Conditional Decoding**$(\mathbf{y}, G, t)$. Indeed, if there were a solution $(\mathbf{c}_0, \mathbf{e}_0)$ to **Conditional Decoding**$(\mathbf{y}, G, t)$, then $(\mathbf{c}_0 + h(\mathbf{e}_0), \mathbf{e}_0)$, would be a solution to **Decoding**$(\mathbf{y}, G, t)$. The converse is easy to show.
- It is immediate to check that if there is a solution $(\mathbf{x}, \mathbf{e})$ to **Conditional Decoding**$(\mathbf{y}, G, t)$ then $(\mathbf{x} - h(\mathbf{e}), \mathbf{e})$ is a solution to **Decoding**$(\mathbf{y}, G, t)$, which is by construction of $\mathcal{A}'$ the value that the algorithm returns.

This show that our problems are both equivalent in terms of complexity.

3.1 The procedure $\mathcal{P}$

Now we design the procedure $\mathcal{P}$. The goal of this procedure is to put information in an error-vector of rank t and of length $2n$. The number of possible error-vectors is equal to the number of vectors rank t, that is

$$\prod_{i=0}^{t-1} \frac{(2^m - 2^i)(2^{2n} - 2^i)}{2^t - 2^i}.$$

Since t is significantly less than $2n$ and m, this quantity can be approximated by $2^{(m+2n)t-t^2+1}$. Thus the maximal amount of information in bits that can be put on this vector is equal to $(m+2n)t-t^2+1$. An efficient encoding procedure to do this was described in [16], but here we are willing to keep r random bits in the error-vector.

- Let D be a binary $t \times t$ matrix of rank t.
- Let E_1 be a $(m-t)t$ matrix, and let E_2 be a $t(2n-t)$ matrix over GF(2).

The matrix

$$E = \begin{pmatrix} D & DE_2 \\ E_1 & E_1E_2 \end{pmatrix} \tag{2}$$

is a binary matrix of rank exactly t, and is thus the expansion of some vector over GF(2^m) of length $2n$ and of rank t. Information can be put in D, E_1 and E_2, as well as randomness.

An important point is to ensure a good diffusion of randomness in the matrix E. For example, suppose that the random positions are located in the matrix D, then the randomness of the error is located in a known subspace of dimension t. This fact could be used in the *message resend attacks* (cf. §3.3).

To avoid this problem, we propose to use the Rijndael S-box, [18–20]: it is an invertible function which takes in input a byte and return a byte in output. This S-box has good diffusion and non-linearity properties. The information bits and random bits must be spread in bytes in such a way that each byte contains at least one random bit. We apply the Rijndael S-box to each byte and then we put these in D, E_1 and E_2.

It is easy to check that recovering the sent message $\mathbf{x}'$ from this matrix is $O(t^3)$ binary operation and is thus negligible in complexity compared to the other procedures. Our procedure $\mathcal{P}$ satisfies thus all the requirements, that we demanded in the previous section.

3.2 Parameters of the cryptosystem

By using the procedure $\mathcal{P}$ previously described, we increase the transmission rate of the system up to

$$\tau = \frac{2mk + (2n+m)t - r - t^2}{2mn} = k/n + \underbrace{\frac{(2n+m-t)t - r}{2mn}}_{\text{additional gain}}$$

The complexity of the encryption is:

- Computing $\mathbf{e} = \mathcal{P}(r, \widetilde{\mathbf{x}})$: Negligible compared to the other procedures since in $O(t^3)$ binary operations.
- Computing $(\mathbf{x} + h(\mathbf{e}))G_{pub}$: $\approx 4nk$ multiplications in GF(2^m).

The complexity of the decryption is:

- Multiplying by P^{-1}: $\approx 2n^2$ binary operations.
- Then one has to compute two decoding steps and a multiplication by A: $\approx 2t^3 + 2(2n+m)t + 2k^2 + nk$ multiplications in GF(2^m).
- Then multiplying by S^{-1}, supposed already precomputed: $4k^2$ multiplications in GF(2^m).

Therefore the overall complexity of the decryption is: $\approx t(2t^2 + 2(2n+m)) + k(6k+n)$ operations in the field GF(2^m).

3.3 Security of the cryptosystem based on rank codes

At the beginning of the section we showed that the problem on which we base the security of our system is equivalent to the problem of decoding in rank metric. However there might be various ways to attack the system.

Structural attacks

They consist in attacking directly the public-key to break the system. In [16], it is discussed the resistance of the private key to any structural attacks. It is shown that it is not possible to break the system by recovering a decoder provided the parameters are well chosen. A security analysis implies moreover that the matrix G of the Gabidulin code can be published without loss of security. Indeed, one can pass from a Gabidulin code to another by changing the bases. Let G_{pub} be written under the form

$$G_{pub} = S \begin{pmatrix} G & A \\ 0 & G \end{pmatrix} P,$$

and let $G' = GU$, where U is non-singular with coefficients over GF(2) be a generator matrix of a Gabidulin code. We we have

$$G_{pub} = S \begin{pmatrix} G' & A \\ 0 & G' \end{pmatrix} P',$$

where

$$P' = \begin{pmatrix} U^{-1} & 0 \\ 0 & U^{-1} \end{pmatrix} P.$$

Therefore, publishing the matrix G of the private key does not give any additional information when one tries to cryptanalyse the system by attacking the public-key.

Decoding attacks

For these attacks, an attacker intercepts a ciphertext and tries to recover the corresponding plaintext. The best known algorithms to achieve this goal were designed by Ourivski and Johansson and have complexity:

- *Minimum rank weight decoding*: $O\left((tm)^3 2^{(t-1)(2k+1)}\right)$ binary operations.
- *Basis enumeration*: $O\left((2k+t)^3 t^3 2^{(t-1)(m-t)}\right)$ binary operations.

Now we only discuss the resistance of the system against active eavesdropping. All basic attacks on our system can be naturally avoided, since there complexity is naturally exponential.

Message resend attack

In Hamming metric, it was presented by Berson in the case of McEliece cryptosystem [1]. It provides two different informations:

- An eavesdropper is able to know whenever the same message is sent twice because the Hamming weight of the difference of the encrypted message is very small.
- This knowledge gives information on the positions of errors, thus considerably diminishing the complexity of the decoding attack.

Suppose that the same message $\mathbf{x}$ is encrypted twice with our system. The attacker gets the two ciphertexts $\mathbf{y}_1 = (\mathbf{x} + h(\mathbf{e}_1))G_{pub} + \mathbf{e}_1$ and $\mathbf{y}_2 = (\mathbf{x} + h(\mathbf{e}_2))G_{pub} + \mathbf{e}_2$, where the $\mathbf{e}_i$'s contain some information and depend on r bits of randomness and on the transformation $\mathcal{P}$.

By computing the difference, he gets $\mathbf{y}_1 - \mathbf{y}_2 = (h(\mathbf{e}_1) - h(\mathbf{e}_2))G + \mathbf{e}_1 - \mathbf{e}_2$. Provided $h(\mathbf{e}_1) \neq h(\mathbf{e}_2)$, there is no way to distinguish this difference with the difference of two random ciphertexts. The probability that such an event occurs depends on the random parameter r. Even if it happens, distinguishing the ciphertexts does not enable one to recover the plaintext easily.

This analysis shows that the hash function introduces a *kind* of semantic security in our system. Another type of active attack that we consider are the reaction attacks.

Reaction attacks

Studying reaction attacks in case of rank metric is not so simple. Indeed, what plays the important role in the error vector is not a bit itself but a full vector space.

Reaction attacks can be described as follows: An attacker eavesdrops the channel and gets a ciphertext $\mathbf{y}$. He modifies a few bits of the ciphertext and then submits the new text to a decryption oracle. If the oracle does not reject the ciphertext it means that it is a valid ciphertext. Thus one obtains some information about the changed bits. This kind of attack is particularly adapted to Hamming metric, [12].

In rank metric, the basic idea would be in the same way to add some error on the ciphertext and then sees if this new message is accepted as a valid ciphertext. We can imagine the following attack:

- He picks up a vector $\mathbf{f}$ and compute $\mathbf{y}' = \mathbf{y} + \mathbf{f}$.
- He submits the new $\mathbf{y}'$ to the decryption oracle and sees its reaction: *Accept* or *Reject*.

By properties of rank metric, the oracle will accept $\mathbf{y}'$ as a valid ciphertext if and only if $\mathbf{f}+\mathbf{e}$ is of rank less than t, that is, since $\mathbf{e} = (e_1, \ldots, e_n)$ has rank t, if and only if every coordinate f_i of $\mathbf{f}$ is a linear combination of the e_j's. We can show that this attack can succeed in approximately $t2^{m-t}$ queries to the decryption oracle. Namely you can obtain a set of t elements forming a basis of the vector space spanned by the error-vector $\mathbf{e}$, and then you can decode and recover the plaintext in polynomial time. However note that in that case, the complexity of reaction attack is not linear but exponential in the size of the extension field. So the question how many queries can do the attacker to the oracle to complete a reasonable attack. We cannot answer to that question here but we see that if this is really problematic it is enough to increase the field size to secure the system.

3.4 Proposition of parameters

We propose two different sets of parameters :
First set: $m = n = 22$, $k = 10$, $t = 6$ and random parameter $r = 80$.

- Size of the public-key : $22 \times 24 \times 20 = 10560$ bits if we consider only the redundant part of the matrix.
- Transmission rate of the system : ≈ 0.78.
- Security against the decoding attacks :
  1. *Minimum rank weight decoding*: $\approx 2^{126}$ binary operations.
  2. *Basis enumeration*: $\approx 2^{100}$ binary operations.

Note that in this case, for an equivalent security the public-key size of McEliece cryptosystem has to be of 700 kbits, that is 70 times larger [2]. The second set of parameters takes into account the necessary resistance to reaction attacks. Therefore we increase the size of the chosen finite field.

Second set: $m = 60$, $n = 20$, $k = 10$, $t = 5$ and random parameter $r = 80$.

- Size of the public-key : $60 \times 20 \times 20 = 24000$ bits if we consider only the redundant part of the matrix.
- Transmission rate of the system : ≈ 0.65.
- Security against the decoding attacks :
  1. *Minimum rank weight decoding*: $\approx 2^{109}$ binary operations.
  2. *Basis enumeration*: $\approx 2^{281}$ binary operations.

This increases notably the size of the public-key but remains 30 times smaller than the public-key size of a secure McEliece cryptosystem. In the parameters we chose to propose a security of 2^{100}. For a security corresponding to 2^{80} binary operations, the key size can be made smaller.

4 Conclusion

From a cryptosystem based on rank metric, we designed a new one. It has the advantages of rank metric that is to know a small public-key size, and it is resistant to different kinds of attacks. Namely, this system provides a kind of semantic security and can be rendered secure against reaction attacks and message resend attacks which are problematic if one uses systems such as McEliece cryptosystem.

References

1. T. A. Berson. Failure of the McEliece public-key cryptosystem under message resend and related-message attack. In *Advances in Cryptology, CRYPTO 1997*, Lecture Notes in Computer Science, pages 213–220, 1997.
2. A. Canteaut and N. Sendrier. Cryptanalysis of the original McEliece cryptosystem. In K. Ohta and D. Pei, editors, *Advances in Cryptology - ASIACRYPT'98*, number 1514 in LNCS, pages 187–199, 1998.
3. N. Courtois, M. Finiasz and N. Sendrier. How to achieve a McEliece-based signature scheme. In Colin Boyd, Editors, *Advances in Cryptology - ASIACRYPT'2001*, number 2248 in LNCS, pages 151–174,2001
4. F. Chabaud and J. Stern. The cryptographic security of the syndrome decoding problem for rank distance codes. In K. Kim and T. Matsumoto, editors, *Advances in Cryptology - ASIACRYPT '96*, volume 1163 of *LNCS*. Springer-Verlag, November 1996.
5. K. Chen. A new identification algorithm. In *Cryptographic policy and algorithms*, volume 1029, pages 244–249. Springer, 1996.
6. E. M. Gabidulin. Theory of codes with maximal rank distance. *Problems of Information Transmission*, 21:1–12, July 1985.
7. E. M. Gabidulin. A fast matrix decoding algorithm for rank-error correcting codes. In G. Cohen, S. Litsyn, A. Lobstein, and G. Zémor, editors, *Algebraic coding*, volume 573 of *LNCS*, pages 126–133. Springer-Verlag, 1991.
8. E. M. Gabidulin and A. V. Ourivski. Modified GPT PKC with right scrambler. In Daniel Augot and Claude Carlet, editors, *Proceedings of the 2nd International workshop on Coding and Cryptography, WCC 2001*, pages 233–242, 2001. ISBN Number : 2-761-1179-3.
9. E .M. Gabidulin, A. V. Paramonov, and O. V. Tretjakov. Ideals over a non-commutative ring and their application in cryptology. *LNCS*, 547:482 – 489, 1991.
10. J. K. Gibson. Severely denting the Gabidulin version of the McEliece public-key cryptosystem. *Designs, Codes and Cryptography*, 6:37–45, 1995.
11. J. K. Gibson. The security of the Gabidulin public-key cryptosystem. In U. Maurer, editor, *EUROCRYPT'96*, pages 212–223, 1996.
12. C. Hall, I. Goldberg, and B. Schneier. Reaction attacks against several public-key cryptosystems. In *Proceedings of the 2nd International Conference on Information and Communication Security, ICICS'99*, number 1726 in LNCS, pages 2–12, 1999.
13. R. J. McEliece. A public-key cryptosystem based on algebraic coding theory. Technical report, Jet Propulsion Lab. DSN Progress Report, 1978.
14. A. Ourivski and T. Johannson. New technique for decoding codes in the rank metric and its cryptography applications. *Problems of Information Transmission*, 38(3):237–246, September 2002.

15. A. V. Ourivski and E. M. Gabidulin. Column scrambler for the GPT cryptosystem. *Discrete Applied Mathematics*, 128(1):207–221, May 2003. Special issue of the second International Workshop on Coding and Cryptography (WCC2001).
16. A. V. Ourivski, E. M. Gabidulin, B. Honary, and B. Ammar. Reducible rank codes and their applications to cryptography. *IEEE Transactions on Information Theory*, 49(12):3289–3293, December 2003.
17. G. Richter and S. Plass. Fast Decoding of Rank-Codes with Rank Errors and Column Erasures In *Proceedings of ISIT 2004*, 2004.
18. J. Daemen, V. Rijmen *The Block Cipher Rijndael*, Smart Card Research and Applications, LNCS 1820, J.-J. Quisquater and B. Schneier, Eds., Springer-Verlag, 2000, pp. 288-296.
19. J. Daemen, V. Rijmen *Rijndael, the advanced encryption standard*, Dr. Dobb's Journal , Vol. 26, No. 3, March 2001, pp. 137–139.
20. `http://www.esat.kuleuven.ac.be/~rijmen/rijndael/rijndaelref.zip`
21. R. M. Roth. Maximum-Rank array codes and their application to crisscross error correction. *IEEE Transactions on Information Theory*, 37(2):328–336, March 1991.
22. N. Sendrier. Efficient generation of binary words of given weight. In C. Boyd, editor, *Cryptography and Coding, 5th IMA Conference, Circencester, UK*, LNCS, pages 184–187, December 1995.
23. H.-M. Sun. Enhancing the security of the McEliece public-key cryptosystem. *Journal of Information Science and Engeneering*, 16:799–812, 2000.
24. A. Vardy. The intractability of computing the minimum distance of a code. *IEEE Transactions on Information Theory*, 43(6):1757–1766, November 1997.