Fifteenth International Workshop on Algebraic and Combinatorial Coding Theory
June 18-24, 2016, Albena, Bulgaria *pp. 215–220*

An evolution of GPT cryptosystem

Pierre Loidreau Pierre.Loidreau@m4x.org
DGA MI and IRMAR, Université de Rennes 1

Abstract. The goal of the paper is to show how to design a Gabidulin based public-key cryptosystem resistant to all Overbeck's like attacks. The main idea consists in taking the coefficients of the right scrambler in a subspace of the coefficients field with sufficiently small dimension. This gives a rank multiplier and scrambles the structure of the code. We propose some parameters.

1 Introduction

In 1991, Gabidulin, Paramonov and Tretjakov presented a Gabidulin-codes based cryptosystem [2]. Unfortunately, the original system and many of its evolutions (such as taking subcodes) were broken.

The main weakness of the systems relies in the fact that Gabidulin codes contain a huge vector space invariant by the action of the Frobenius automorphism ($k-2$, where k is the dimension of the code). Even if one considers the most recent evolutions using a right scrambler the problem remains. Namely, some coefficients of the scrambler are fixed in the base field and this increases the weakness of the system [3, 8].

In this paper we follow this idea of using a right scrambler but benefit from the study of LRPC codes [6]. The idea consists in taking the coefficients of the inverse of the right scrambler in a fixed dimensional secret subspace. When decoding, this provokes a rank multiplication of the error.

In this paper we present the construction, which relies on the existence for rank metric of operators called *rank multipliers*. In the following of the paper we consider codes of length n over $GF(q^m)$.

2 Gabidulin codes, GPT cryptosystem and Overbeck's attacks

We briefly recall the necessary stuff on GPT cryptosystem, and the principle of Overbeck's attacks. A more detailed analysis in English can be found in [4, 5].

2.1 Rank metric and Gabidulin codes

Here we give a non-standard definition of rank metric, but this definition is equivalent to the classical definition of rank metric given in [1].

Definition 1 (Rank metric)**.** *Let* $\mathbf{x} = (x_1, \ldots, x_n) \in GF(q^m)^n$*, and let*

$$\mathcal{X} =< x_1, \ldots, x_n >= \left\{ \sum_{i=1}^{n} \mu_i x_i \mid \mu_i \in GF(q) \right\},$$

be the $GF(q)$*-linear vector space generated by the components of* $\mathbf{x}$*. Then* $Rk(\mathbf{x}) = \dim_q(\mathcal{X})$*.*

Let $n \leq m$ and let $\mathbf{g} = (g_1, \ldots, g_n) \in GF(q^m)$, where the $g_i's$ are linearly independent over $GF(q)$. The code $Gab_k(\mathbf{g})$, is the linear code with generator matrix

$$\mathbf{G} = \begin{pmatrix} g_1 & \cdots & g_n \\ g_1^{[1]} & \cdots & g_n^{[1]} \\ \vdots & \ddots & \vdots \\ g_1^{[k-1]} & \cdots & g_n^{[k-1]} \end{pmatrix} \tag{2.1}$$

This code corrects errors of rank up to $\lfloor (n-k)/2 \rfloor$. There are many different efficient polynomial-time algorithms to perform the decoding. The original one can be found in [1].

2.2 Principle of automorphism based attacks

The principle of GPT cryptosystem, see [2], consists in taking the generator matrix of a Gabidulin code under the form (2.1), scrambling it and then publishing the scrambled form. The most general scrambling has the form

$$\mathbf{G}_{pub} = \mathbf{S}(\mathbf{X} \mid \mathbf{G})\mathbf{P},$$

where $\mathbf{G}$ is a generator matrix of a Gabidulin code $\mathcal{G}$ of length n, and $\mathbf{P}$ the right scrambler. To ensure proper decoding, the right-scrambler $\mathbf{P}$ has to satisfy some properties. In any case it has to be non-singular. Originally $\mathbf{P}$ was taken with coefficients in the base field $GF(q)$. This ascertains that $\mathbf{P}^{-1}$ has coefficients in $GF(q)$, and is therefore an isometry of rank metric.

This is a major weakness since: if we denote by $\mathcal{C}_{pub}$ the code generated by $\mathbf{G}_{pub}$ and by $\mathcal{C}_{pub}^{[i]}$ the code obtained by elevating the codewords of $\mathcal{C}_{pub}$ to the ith power of the Frobenius automorphism, a generator of this code is

$$\mathbf{G}_{pub}^{[i]} = \mathbf{S}^{[i]}(\mathbf{X}^{[i]} \mid \mathbf{G}^{[i]})\mathbf{P},$$

and the dimension of

$$\mathcal{C}_{pub}^{\perp} \cap \left(\mathcal{C}_{pub}^{[1]}\right)^{\perp} \cap \cdots \cap \left(\mathcal{C}_{pub}^{[i]}\right)^{\perp},$$

is at least The dimension of $\mathcal{G}^{\perp} \cup \cdots \cup \left(\mathcal{G}^{[i]}\right)^{\perp}$, that is at least $n-k-i-1$. If the codes were randomly chosen one would expect the dimension to be $\max(n-ik, 0)$. Hence one obtains an efficient distinguisher for GPT cryptosystem. Even worse, if i is sufficiently large, we generally obtain that

$$\mathcal{C}_{pub}^{\perp} \cap \cdots \cap \left(\mathcal{C}_{pub}^{[i]}\right)^{\perp} = \left(\mathcal{G}^{\perp} \cup \cdots \cup \left(\mathcal{G}^{[i]}\right)^{\perp}\right) \mathbf{P}.$$

Provided the codes are non-trivial this enables to recover a decoder.

More elaborate forms of right scrambler have been proposed, for instance in [3, 8]. Unfortunately for the conceivers, the former point remains true that is to know, the public key $\mathbf{G}_{pub}$ can always be rewritten under the form

$$\mathbf{G}_{pub} = \mathbf{S}^*(\mathbf{X}^* \mid \mathbf{G}^*)\mathbf{P}^*, \tag{2.2}$$

where $\mathbf{P}^*$ has coefficients in $GF(q)$, and $\mathbf{G}^*$ a generator matrix for a Gabidulin code of smaller length. This nice result comes from [5].

Once we have (2.2), it is obvious that the public-code contains a subspace invrariant by the action of the Frobenius automorphism.

Now the idea of relaxing optimality on the code by scrambling the columns with a non-isometry of the metric is not new it was done for Hamming metric in the case of GRS codes, [9], by using an almost permutation matrix $\mathbf{P}$ and tolerating few rows and columns to have Hamming weight 2. However this scheme and reparations was broken in [10], by designing a distinguisher on the Hamming weight of the rows of the scrambler.

Rank metric is much more adapted for such transformations as we see in the following.

3 Rank multipliers

The concept of *rank multiplication* can be found in [6]. Consider $\alpha_1, \ldots, \alpha_\lambda \in GF(q^m)$, $GF(q)$-linearly independent elements. Let $\mathcal{V} =< \alpha_1, \ldots, \alpha_\lambda >$ be the $GF(q)$-linear subspace generated by $\alpha_1, \ldots, \alpha_\lambda$. Let $\mathbf{P} \in M_n(\mathcal{V})$, be a $n \times n$-non singular matrix with coefficients taken in $\mathcal{V}$. Then

Proposition 1 (Rank multiplication)**.** *For all* $\mathbf{x} \in GF(q^m)^n$, $Rk(\mathbf{xP}) \leq \lambda Rk(\mathbf{x})$.

Proof. Consider $\mathbf{x} = (x_1, \ldots, x_n) \in GF(q^m)$ of rank r. Let $\mathcal{X} =< x_1, \ldots, x_n >$ be generated by $< y_1, \ldots, y_t >$. Suppose moreover that $\mathcal{V} =< \alpha_1, \ldots, \alpha_\lambda >$, then the components of $\mathbf{xP}$, belong to the vector space $< y_i\alpha_j >_{i,j}$ which has dimension $\leq \lambda t$. □

This property of combining subspaces was first introduced in the design of LRPC codes. These codes are the equivalent of LDPC codes with respect to rank metric. An immediate corollary is

Corollary 1. *Let $\mathcal{C}$ be a $[n,k,d]_r$ code over $GF(q^m)$. Let $\mathcal{V}$ be a λ-dimensional subspsace of $GF(q^m)$ seen as an $GF(q)$-vector space. And let $\mathbf{P} \in M_n(\mathcal{V})$. Then*

$$\mathcal{C}\mathbf{P}^{-1} \stackrel{def}{=} \{\mathbf{c}\mathbf{P}^{-1} \mid \mathbf{c} \in \mathcal{C}\}$$

has rank dimension k and rank distance $d' \geq \lfloor d/\lambda \rfloor$.

Proof. Since $\mathbf{P}$ is invertible $\mathcal{C}$ and $\mathcal{C}\mathbf{P}^{-1}$ have the same dimension. Concerning the minimum distance, suppose that $d' < d/\lambda$. Then let $\mathbf{c} \in \mathcal{C}\mathbf{P}^{-1} \neq \mathbf{0}$ with rank distance d'. By construction $\mathbf{c}\mathbf{P} \in \mathcal{C}$. But from proposition 1, $\mathrm{Rk}(\mathbf{c}\mathbf{P}) \leq d'\lambda < d$, which implies that $\mathbf{c}\mathbf{P} = \mathbf{0}$. Thus $\mathbf{c} = \mathbf{0}$, which contradicts the hypothesis. □

4 Proposition of cryptosystem

In this section we formalize our proposal of cryptosystem, give some security analysis and propose parameters.

4.1 A McEliece like form

The key generation procedure is the following:

- Private key:
  - A Gabidulin code of length n over $GF(q^m)$, dimension k with generator matrix $\mathbf{G}$ under the form (2.1).
  - A non-singular $k \times k$-matrix $\mathbf{S}$ with coefficients in $GF(q^m)$.
  - A λ-dimensional subspace of $GF(q^m)$, denoted by $\mathcal{V}$.
  - A non-singular matrix $\mathbf{P} \in M_n(\mathcal{V})$.
- Public key: $\mathbf{G}_{pub} = \mathbf{S}\mathbf{G}\mathbf{P}^{-1}$.

The encryption and decryption procedure is:

- Encryption of $\mathbf{x} \in GF(q^m)^k$:
  - Choose a random vector $\mathbf{e} \in GF(q^m)^n$ of rank $\lfloor (n-k)/(2\lambda) \rfloor$.
  - Compute $\mathbf{y} = \mathbf{x}\mathbf{G}_{pub} + \mathbf{e}$.
  - Send the encrypted message $\mathbf{y}$ to the receiver.
- Decryption of $\mathbf{y}$:
  - Compute $\mathbf{y}\mathbf{P} = \mathbf{x}\mathbf{S}\mathbf{G} + \mathbf{e}\mathbf{P}$.
  - By construction the rank of $\mathbf{e}\mathbf{P}$ is $\leq \lambda\lfloor (n-k)/(2\lambda) \rfloor \leq \lfloor (n-k)/2 \rfloor$, and therefore can be decoded with $\mathbf{G}$.

– Recover $\mathbf{xS}$ and $\mathbf{eP}$ by decoding and finally get $\mathbf{x}$ by multiplying with $\mathbf{S}^{-1}$.

A Niederreiter form can be obtained similarly to the Niederreiter idea. A remark here is that with this construction it is worth looking at the concept of Trace code or even subfield subcode to scramble the structure.

4.2 How to analyse the system ?

In this section we raise the question of the security. This a two-fold analysis.

1. We analyse the security against decoding attacks. This path is relatively well studied and the most recent results in [7] establish that the average complexity for generic decoding of a code $[n, k]$ over $GF(q^m)$ is at least $m^3 q^{(t-1)\lfloor (k \min(m,n))/n \rfloor}$ operations in $GF(q)$.

2. We provide a basic analysis of the system to show that the commonly employed ideas used to break such systems are inefficient. First it is quite obvious that Overbeck's approach consisting in making use of the Frobenius automorphism cannot be employed. Namely we have

$$\mathbf{G}_{pub}^{[i]} = \mathbf{S}^{[i]}\mathbf{G}^{[i]}\mathbf{P}^{[i]}.$$

Since $\mathcal{V}$ has no reasons to be invariant by the Frobenius, $\mathcal{C}_{pub}$ and $\mathcal{C}_{pub}^{[i]}$ have no reasons to be correlated. So the most straightforward way to attack would be to rewrite the systems originally written for GPT cryptosystem and try to solve it. Provided $\mathbf{P}$ has no particular structure (monomial, cyclic or whatsoever), we propose a lower bound on the complexity of recovering a decoder as being the number of $\lambda - 1$-dimensional $GF(q)$-subspaces of $GF(q^m)$. We choose $\lambda - 1$ comes from the fact that if $\lambda = 1$, *i.e.* $\mathcal{V} =< \alpha >$, for some element $\alpha \in GF(q^m)$, it is obvious that an attack can be achieved in polynomial time. Namely, $\mathbf{P} = (1/\alpha)\mathbf{P}'$ with $\mathbf{P}'$ has coefficients in the base field $GF(q)$. Roughly speaking we lower bound the complexity by $n^3 q^{m(\lambda-1)-(\lambda-1)^2}$.

4.3 Proposition of parameters

We propose parameters for 128 bits security, one in the McEliece setting, the other one in the Niederreiter setting. The complexity estimations are given in terms of binary operations.

1. First option code of length n, dimension k over $GF(2^m)$

   - $m = 96,\ n = 64,\ \lambda = 3,\ t = 4,\ k = 40.$
   - Public-Key size: $\approx 11,5 KBytes$ under systematic form.

- Decoding attack estimation: $\approx 2^{139}$.
- Structural attack estimation: $\approx 2^{206}$.

2. Second option: Niederreiter type

- $m = 64,\ n = 64,\ \lambda = 3,\ t = 8,\ k = 22$.
- Public-Key size: $\approx 7,4 KBytes$ under systematic form.
- Decoding attack estimation: $\approx 2^{130}$.
- Structural attack estimation: $\approx 2^{142}$.

References

[1] E. M. Gabidulin. Theory of codes with maximal rank distance. *Problems of Information Transmission*, 21:1–12, July 1985.

[2] E. M. Gabidulin, A. V. Paramonov and O. V. Tretjakov. Ideals over non-commutative rings and their application in cryptology. *EUROCRYPT'91.*

[3] E. M. Gabidulin, H. Rashwan and B. Honary. On improving security of GPT cryptosystems. *ISIT 2009.*

[4] A. Kshevetskiy. Security of GPT-like public-key cryptosystems based on linear rank codes. *3rd International Workshop on Signal Design and Its Applications in Communications, 2007. IWSDA 2007.*

[5] A. Otmani, H. T. Kalashi and S. Ndjeya. Improved cryptanalysis of rank metric schemes based on Gabidulin codes. http://arxiv.org/abs/1602.08549v1.

[6] P. Gaborit, G. Murat, O. Ruatta and G. Zémor. Low Rank Parity-check codes and their application to cryptography. *International Workshop on Coding and Cryptography, WCC 2013.*

[7] P. Gaborit, O. Ruatta and J. Schrek. On the complexity of rank syndrome decoding problem. *IEEE Trans. on Inf. Theo.*, 62(2), pages 1006–1019.

[8] H. Rashwan, E. M. Gabidulin and B. Honary. Security of the GPT cryptosystem and its applications to cryptography. *Security and Communication Networks*, 4(8):937-946, 2011.

[9] M. Bianchi, F. Chiaraluce, J. Rosenthal and D. Schipani. Enhanced Public Key Security for the McEliece Cryptosystem. *Journal of Cryptology*, 29(1):1-27, 2016.

[10] A. Couvreur, A. Otmani, J.-P. Tillich and V. Gauthier-Umaña. A Polynomial-Time Attack on the BBCRS Scheme. http://arxiv.org/pdf/1501.03736.pdf.