How to mask the structure of error-correcting codes for a cryptographical use

Thierry P. Berger Pierre Loidreau

Abstract

In this paper we show how to strengthen public-key cryptosystems against known attacks, together with the reduction of the public-key. We use properties of subcodes to mask the structure of the codes used by the conceiver of the system. We propose new parameters for the cryptosystems and even a modified Niederreiter cryptosystem in the case of Gabidulin codes, with a public-key size of less than 4 000 bits.

1 Introduction

Among the family of public-key cryptosystems, the ones whose security rely on the difficult problem of decoding a codeword are not much studied. The principle of such systems was introduced in 1978 by McEliece [14]. They form an interesting alternative to cryptosystems whose security is based on problems of number theory. One of their advantages is their speed in the cipher-decipher procedure. However, the main drawback remains the size of the public key which is prohibitive.

In this paper we show that it is possible to strengthen the security of such systems against all known attacks together with reducing the size of the public-key. Namely, the smallest key size attainable for such systems is reached when the conceiver uses optimal codes such as GRS codes for Hamming metric and Gabidulin codes for the rank metric. For such codes it is not possible to scramble enough the structure of the public-key so that it becomes infeasible for the attacker to recover a decoder for the public code. Both attacks of Sidel'nikov–Shestakov [16] and of Gibson [9, 10]. are perfects examples of it.

In this paper we show that it is possible to strengthen the cryptosystems by using "almost optimal" codes, that is by using some well chosen subcodes of optimal codes. Since subcodes can be decoded by the same decoding algorithm as the one for the parent code, by taking subcodes of dimension slightly less than the code's dimension, we obtain a family of codes resistant to known attacks.

In the first part we show how to do it by using the family of GRS codes in the Niederreiter cryptosystem, and we propose secure parameters giving a public-key size significantly less than for the McEliece cryptosystem. In the second part, we build the Niederreiter form of the GPT cryptosystem based on the rank metric and show how to construct an efficient and resistant cryptosystem, with a public-key size that is less than 4 000 bits.

2 McEliece and Niederreiter cryptosystems

The McEliece public key cryptosystem [14] is based on the difficulty of decoding linear codes, that is finding the nearest codeword of a given word. Its dual version, the Niederreiter public key cryptosystem [15] is based on the difficulty of finding a coset leader of a coset of the code, that is finding the smallest with a given syndrome.

Clearly, for a given code, these problems are equivalent. Both use a class $\mathcal{C}$ of codes of length n and dimension k having a fast decoding algorithm for correcting up to t errors.

2.1 Description of the McEliece public key cryptosystem

- Private key:
  - An element C of the class of codes $\mathcal{C}$, with a fast decoding algorithm.
  - An invertible $k \times k$ matrix S.
  - A $n \times n$ permutation matrix P.
- Public key:
  - A matrix $G_{pub} = SGP$, where G is a generator matrix of C. The matrix G_{pub} is a generator matrix of the code C_{pub} equivalent to C under the permutation P.
- Encryption:
  $c \rightarrow y = cG_{pub} + e'$, where c is a message of size k and e' a random word of weight t.
- Decryption:
  - Compute $x = yP^{-1} = cSG + e'P^{-1}$. Note that cSG is a codeword of C and $e = e'P^{-1}$ is a word of weight t.
  - Decode x with the secret algorithm for C. This gives cSG.
  - Compute c from cSG.

The class of codes proposed by McEliece is the class of binary Goppa codes. These codes are subfield subcodes over $GF(2)$ of Generalized Reed-Solomon codes (GRS codes). The secret key is in fact the underlying algebraic structure of GRS codes, and the decoding algorithm is the classical Berlekamp-Massey algorithm for GRS codes.

The original parameters are $n = 1024$, $k \geq 524$, $t = 50$ and a designed distance $d = 101$. With these values, the characteristics of this system are (cf. [2]):

- Size of public key: 67072 bytes.
- Transmission rate: 0.512.
- Number of codes: $\approx 2^{498}$.

There are two classes of attacks against a such system:

- The structural attacks: Except for some weak keys (cf. [12]), there is no know attack recovering the structure of Goppa codes from the public-key.
- The decoding attacks: how to decode a random code without structure? The main idea of such attacks is to recover an information set of symbols without errors. These attacks are briefly described in Section 2.3. However the proposed parameters remain resistant under such attacks.

2.2 Description of the Niederreiter public key cryptosystem

- Private key:
  - An element C of the class of codes $\mathcal{C}$.
  - An invertible $(n-k) \times (n-k)$ matrix S.
  - A $n \times n$ permutation matrix P.

- Public key:
  - A matrix $H_{pub} = SHP$, where H is a parity check matrix of C. The matrix H_{pub} is a parity check matrix of the code C_{pub} equivalent under the permutation P to C.

- Encryption:
  $c \rightarrow s' = cH_{pub}^t$, where c is a message of weight t.

- Decryption:
  - Compute $s = s'S^{-1t} = cP^tH^t$. Set $c' = cP^t$. Note that c' is a word of weight t and s is the syndrome of c' by H.
  - Decode s with the secret algorithm for C. This gives c'.
  - Compute c from $c' = cP$.

In its original paper (cf. [15], Niederreiter proposed as a possibility for $\mathcal{C}$ the class of Generalized Reed-Solomon codes over an extension field $GF(2^m)$. Unfortunately, V.M. Sidel'nikov and S.O. Shestakov [16] gave a polynomial algorithm to recover the structure of GRS codes. We will describe this algorithm in the next section.

However, it is possible to use the Niederreiter cryptosystem with the class of Goppa codes proposed by McEliece. Contrary to McEliece cryptosystem, the matrix H' can be under a systematic form without loss of security.

For the preceding parameters, the characteristics of the Niederreiter system are (cf. [2]):

- Size of public key: 32750 bytes.
- Transmission rate: 0.568.
- Number of codes: $\approx 2^{498}$.

The security of this system against structural and decoding attacks is equivalent to the security of McEliece cryptosystem with same parameters.

One of the great advantage of McEliece or Niederreiter cryptosystems is the work-function of encryption-decryption. For example, using standard parameters (cf. [2]), the number of binary operations performed by the encryption per information bit are respectively 514, 50 and 2402 for the McEliece, Niederreiter and RSA cryptosystems. The number of binary operations performed by the decryption per information bit are respectively 5140, 7863 and 738112.

The main problem of the McEliece or Niederreiter cryptosystems remains the size of the public key: respectively 67072 bytes, 32750 bytes and 256 bytes for the McEliece, Niederreiter and RSA cryptosystems.

2.3 Why construct a cryptosystem with extension fields

We will see in the next section that the structure of GRS code is easy to recover. Hence, the use of this class of codes for $\mathcal{C}$ is not secure. However, the use of codes over an extension field instead of $GF(2)$ is of great interest for many reasons.

The most important property is the fact that, for a given length and a given dimension, the minimum distance is larger than in the binary case.

For example, the parameters of Goppa codes used in the classical McEliece cryptosystem are $[1024, 524, 101]$, i.e. a capacity of correcting $t = 50$ errors. If you use a GRS code, you obtain the parameters $[1024, 524, 501]$, i.e. a capacity of correction $t = 250$.

For a fixed dimension k, the difficulty of decoding increases with the number t of errors used in the cryptosystem.

The best known probabilistic algorithms to correct t errors are based on the search of information set, i.e. k elements of the support without errors (cf. [2]).

The probability of finding such a set is:

$$\binom{n-t}{k} / \binom{n}{k}$$

Moreover, the various improvements of this method in the binary case does not hold in the extension field case. For instance the Lee & Brickell algorithm needs $2^m - 1$ steps at each iteration instead of $2 - 1 = 1$ step in the binary case.

As an example, for a binary code of parameters $[1024, 524, 101]$, the work function of these algorithms is 2^{64}. For a code of parameters $[255, 129, 123]$ over $GF(256)$, the work function is greater than 2^{100}. For this estimation we consider that a multiplication or an addition over $GF(2^m)$ needs m binary operations.

For a fixed level of security, the choice of codes over extension field allows the use of smaller codes. Consequently, the key size as well as the work function of decryption are smaller.

2.4 How to recover the structure of GRS codes.

The main known class of MDS codes with an efficient decoding algorithm is the class of Generalized Reed-Solomon codes. Unfortunately, their algebraic structure is very strong and it is possible to recover it.

In this section, we recall the definition of Generalized Reed-Solomon codes and a method to recover the structure of such a code which is essentially Sidel'nikov and Shestakov's (cf. [16]).

Let K be the finite field $GF(p^m)$ and $\overline{K} = K \cup \{\infty\}$ be a set of coordinates for the projective line.

Definition 1 *Let $\mathcal{L} = (\alpha_0, \ldots, \alpha_{n-1})$ be an n-tuple of distinct elements of $\overline{K}$ and $\mathbf{v} = (v_0, \ldots, v_{n-1})$ be an n-tuple of non-zero elements of K. Let k be an integer less than n. The Generalized Reed-Solomon code $GRS_k(\mathbf{v}, \mathcal{L})$ is the code of length n over K with generator*

matrix

$$M_k(\mathbf{v},\mathcal{L}) = \begin{pmatrix} v_0 & v_1 & \dots & \dots & v_{n-2} & 0 \\ v_0\alpha_0 & v_1\alpha_1 & \dots & \dots & v_{n-2}\alpha_{n-2} & 0 \\ v_0\alpha_0^2 & v_1\alpha_1^2 & \dots & \dots & v_{n-2}\alpha_{n-2}^2 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ v_0\alpha_0^{k-1} & v_1\alpha_1^{k-1} & \dots & \dots & v_{n-2}\alpha_{n-2}^{k-1} & v_{n-1} \end{pmatrix}$$

Where by convenience $\alpha_{n-1} = \infty$.

Such a code is a MDS code (cf [13]). The dual of a GRS code is a GRS code. Moreover, the support of the dual code is $\mathcal{L}$, and its scalars $\mathbf{v}'$ are explicit from $\mathbf{v}$ (cf. [4] Theorem 1).

For a cryptographic use, the public matrix is $G_{pub} = SM_k(\mathbf{v},\mathcal{L})$, where S is a $s \times s$ invertible matrix. The private key is the matrix S and the knowledge of $\mathbf{v}$ and $\mathcal{L}$ which are used for decoding. Note that the choice of an order for the support $\mathcal{L}$ corresponds to the choice of the permutation P in the classical presentation of the McEliece cryptosystem.

An important point is the fact that distinct values of $\mathbf{v}$ and $\mathcal{L}$ give the same GRS code (cf. [4]). More precisely, there is an action of the projective linear group $PGL(2,K)$ on the support $\mathcal{L}$ and the scalar $\mathbf{v}$ which preserves the code $GRS_k(\mathbf{v},\mathcal{L})$. This fact is directly related to the structure of the full automorphism group of GRS codes. For more details, the reader can refer to [4].

For our purpose, we need only two facts:

• First, the group $PGL(2,K)$ is triply-transitive on $\overline{K}$, which implies that it is always possible to fix arbitrary 3 points of the support $\mathcal{L}$ of a GRS code. In the next, we suppose $\alpha_0 = 1$, $\alpha_{k+1} = 0$ and $\alpha_n = \infty$. Moreover, the scalars $\mathbf{v}$ are defined up to a scalar multiplication, and then we can suppose $v_n = (-1)^{k+1}$ without loss of generality.

• The canonical form of the systematic generator matrix of a GRS code is known from $\mathbf{v}$ and $\mathcal{L}$. The following theorem describes this matrix.

Theorem 1 (Theorem 2 of [4].) *Let C be the GRS code $GRS_k(\mathcal{L},\mathbf{v})$. Let $M = (I|B)$ be its systematic generator matrix: I is the $k \times k$ identity matrix and $B = (b_{i,j})$, $i = 1,\dots,k$, $j = k+1,\dots,n$ is a $k \times (n-k)$ matrix . Then $b_{i,j} = \dfrac{d_j}{d_i[\alpha_i,\alpha_j]}$, with*
$[\alpha_i,\alpha_j] = (\alpha_i - \alpha_j)$ *if* $(\alpha_i,\alpha_j) \in K^2$, $[\alpha_i,\infty] = -1$, $[\infty,\alpha_i] = 1$ *if* $\alpha_i \neq \infty$.
$d_i = v_i \prod_{s=1,s\neq i}^{k} [\alpha_s,\alpha_i]$ *for* $i = 1,\dots,k$.
$d_j = v_j \prod_{s=1}^{k} [\alpha_s,\alpha_j]$ *for* $j = k+1,\dots,n$.

An important corollary is the following:

Corollary 1 *For all i, j, u and v such that $1 \leq i,j \leq k$ and $k+1 \leq u,v \leq n$, we have the relation*

$$\frac{b_{i,u} \times b_{j,v}}{b_{j,u} \times b_{i,v}} = \frac{[\alpha_j,\alpha_u] \times [\alpha_i,\alpha_v]}{[\alpha_i,\alpha_u] \times [\alpha_j,\alpha_v]}.$$

Proof : It is just the direct application of relation $b_{i,j} = \dfrac{d_j}{d_i[\alpha_i, \alpha_j]}$. □

Note that the $b_{i,j}$'s are known. Using these relations, if you know 3 values from α_i, α_j, α_u and α_v, we obtain directly the fourth.

Remember that it is always possible to choose three elements of the support and one scalar v_i. Under the hypothesis $\alpha_0 = 1$, $\alpha_{k+1} = 0$, $\alpha_n = \infty$ and $v_n = (-1)^{k+1}$, we obtain the explicit formulas:

- For $i = 1, \ldots, k$: $\quad \alpha_i = \dfrac{b_{i,n} b_{1,k+1}}{b_{1,n} b_{i,k+1}}, \quad d_i = b_{i,n}^{-1}$.
- For $j = k+1, \ldots, n-1$: $\quad \alpha_j = \dfrac{d_2 b_{2,j} \alpha_2 - d_1 b_{1,j} \alpha_1}{d_2 b_{2,j} - d_1 b_{1,j}}, \quad d_j = d_1 b_{1,j} (1 - \alpha_j)$.

Now, we are able to describe the method for recover a generator matrix of the GRS code of the form $M_k(\mathbf{v}, \mathcal{L})$.

1. Construction of the systematic generator matrix of C from the public generator matrix G_{pub}.
2. Compute α_i and d_i for $i = 1, \ldots, k$ from relations $d_i = b_{i,n}^{-1}$ and $\alpha_i = \dfrac{b_{i,n} b_{1,k+1}}{b_{1,n} b_{i,k+1}}$.
3. Compute α_j and d_j for $j = k+1, \ldots, n$ from relations $\alpha_j = \dfrac{d_2 b_{2,j} \alpha_2 - d_1 b_{1,j} \alpha_1}{d_2 b_{2,j} - d_1 b_{1,j}}$ and $d_j = d_1 b_{1,j} (1 - \alpha_j)$.
4. Compute v_i for $i = 1, \ldots, k$ from the relation $d_i = v_i \displaystyle\prod_{s=1, s \neq i}^{k} [\alpha_s, \alpha_i]$ and v_j for $j = k+1, \ldots, n-1$ from the relation $d_j = v_j \displaystyle\prod_{s=1}^{k} [\alpha_s, \alpha_i]$.

The workfunction of this attack is $\mathcal{O}(n^3)$: this corresponds to the construction of the systematic generator matrix. The other operations are linear.

2.5 Masking the GRS structure with subcodes

The main idea for masking the structure of a GRS code GRS_k of parameters $[n, k, d]$ is to use a subcode C of GRS_k of parameters $[n, k-\ell, d']$ with ℓ small and $d' \geq d$. Clearly, the decoding algorithm of GRS_k can be used for C to correct up to $t = \lfloor (d-1)/2 \rfloor$ errors.

The simplest method to construct such a subcode is to add ℓ rows to the parity-check matrix of GRS_k:

- Let $M_d = M_d(\mathbf{v}, \mathcal{L})$ be a generator matrix of $GRS_k^{\perp}$, i.e. a $(d-1) \times n$ parity check matrix of GRS_k.
- A $\ell \times n$-matrix A such that its rows are linearly independent from those of H.

- A $(d-1+\ell)\times(d-1+\ell)$ non singular matrix S over $GF(q^m)$.

Now the parity-check matrix of the subcode C is

$$H_{pub} = S\left(\frac{M_d}{A}\right).$$

2.5.1 Resistance to the Sidel'nikov-Shestakov's attack

Let $M = (I|B)$ be the systematic generator matrix of the GRS code $GRS_k(\mathcal{L}, \mathbf{v})$ given in Theorem 1.

Clearly, up to a permutation of the columns, there exists a generator matrix of C of the form

$$\left(\begin{array}{c|c} I_{k,k} & B \\ \hline 0 & I_{\ell,\ell}\,|\,T \end{array}\right)$$

where $B = (b_{i,j})$ for $i = 1,\ldots,k$ and $j = k+1,\ldots,n$.
$T = (t_{i,j})$, for $i = 1,\ldots,\ell$ and $j = k+\ell+1,\ldots,n$.

Since H_{pub} is known, it is possible to compute the systematic generator matrix of C: $(I_{k+\ell,k+\ell}|R)$, with $R = (r_{i,j})$ for $i = 1,\ldots,k+\ell$ and $j = k+\ell+1,\ldots,n$.

From these remarks, it follows

- For $i = 1,\ldots,k$ and $j = k+\ell+1,\ldots,n$, $\quad r_{i,j} = b_{i,j} - \sum_{s=1}^{\ell} t_{s,j} b_{i,k+s}$.
- For $i = k+1,\ldots,k+\ell$ and $j = k+1,\ldots,n$, $\quad r_{i,j} = t_{i-k,j}$.

Note that the matrix R is known, and then T is known.

In order to develop the Sidel'nikov-Shestakov's attack, we need to recover directly the $b_{i,j}$'s from the $k\times(n-k-\ell)$ equations

$b_{i,j} = r_{i,j} + \sum_{s=1}^{\ell} t_{s,j} b_{i,k+s}$, $i = 1,\ldots,k$ and $j = k+\ell+1,\ldots,n$

and then to solve this with the previous method.

Clearly, this system has $q^{k\ell}$ solutions and only one holds. For usual values of q, k and ℓ this is not feasible (for example, $q = 256$, $n = 255$, $k = 129$ and $\ell = 4$).

Remark 1 *In [5], E. Gabidulin, A. Ourivski and V. Pavlouchkov gave an attack for the case $\ell = 1$. In this attack, they search to recover directly the cross ratios $\dfrac{b_{i,u}\times b_{j,v}}{b_{j,u}\times b_{i,v}} = \dfrac{[\alpha_j,\alpha_u]\times[\alpha_i,\alpha_v]}{[\alpha_i,\alpha_u]\times[\alpha_j,\alpha_v]}$ instead of the values of $b_{i,j}$'s. Unfortunately, this attack does not work for $\ell \geq 2$.*

Choice of the matrix A

For some particular values of A it is possible that the matrix H_{pub} corresponds to the parity check matrix of another GRS code of parameters $[n, k-\ell, d+\ell]$. Note that this possibility is very improbable for a random choice of A. However it is possible to avoid this problem easily: if the subcode has a minimal distance d' such that $d' = d$, it cannot be a MDS code, and then it cannot be a GRS code.

More precisely, the public parity-check matrix is $H_{pub} = S\left(\frac{M_d}{A}\right)$.

We can suppose without loss of generality that

$$A = \begin{pmatrix} 0 & \dots & 0 & a_{1,d} & \dots & a_{1,n} \\ \vdots & \vdots & \ddots & \vdots & & \\ 0 & \dots & 0 & a_{\ell,d} & \dots & a_{\ell,n} \end{pmatrix}.$$

Suppose now what we add one more zero column to A, i.e.

$$A = \begin{pmatrix} 0 & \dots & 0 & a_{1,d+1} & \dots & a_{1,n} \\ \vdots & \vdots & \ddots & \vdots & & \\ 0 & \dots & 0 & a_{\ell,d+1} & \dots & a_{\ell,n} \end{pmatrix}.$$

Proposition 1 *Under the previous hypothesis, the minimum distance of the code C having A as parity-check matrix is $d' = d$*

Proof : Let GRS_k be the generalized Red Solomon codes of parity-check matrix M_d. Let d' be the minimum distance of C. Since C is a subcode of GRS_k, $d' \geq d$.

Let $c \in GRS_k$ be a codeword of weight d and support $\{1, \dots, d\}$, $c = (c_1, \dots, c_d, 0, \dots, 0)$. Clearly $cH_{pub}^t = 0$, c is a codeword of C and $d' = d$. □

2.5.2 Example of parameters

We propose the following parameters: the alphabet is $K = GF(2^8)$, the parameters of the GRS code is $[255, 133, 123]$. We choose $\ell = 4$ and then the parameters of our public code is then $[255, 129, 123]$. The correction capacity is $t = 61$ errors.

The work-function of the decoding attacks is greater than 2^{100}.

The structural attack needs approximatively 2^{2000} Sidel'nikov & Shestakov attacks!

With these parameters, the characteristics of the McEliece cryptosystem are

- Size of public key: 32895 bytes.
- Transmission rate: 0.506.

Whose of the Niederreiter cryptosystem are

- Size of public key: 16254 bytes.
- Transmission rate: 0.680.

3 Cryptosystems based on the rank metric

GPT cryptosystem originally presented in [8] is a McEliece type system. Its security relies on the difficulty of decoding linear codes with respect to the rank metric introduced in [6] by Gabidulin. In the rank metric, the decoding attacks consisting in finding information sets of symbols without errors do not work. They are replaced by attacks denoted Basis enumeration attacks, which can be prevented even with a small key size.

However, despite the use of some distortion matrix, the Gabidulin codes used in the conception of the system are too much structured and an attacker can easily recover a polynomial-time decoder for the public code, thus breaking the system by some structural attack, see [9, 10].

In this section we present briefly the properties of the rank metric, and of the Gabidulin codes, which play the same role as GRS codes in the classical McEliece system. Then we present the GPT system, and the known attacks. Finally, we construct a full Niederreiter type cryptosystem, and we show that it is easy to control the structure of the public key by choosing some subcode of a Gabidulin code, so that the system resists all known attacks. Thus we preserve the small size of the public-key, and the efficiency of the system.

3.1 Rank metric and Gabidulin codes

Consider any finite field $GF(q)$. Given a vector

$$a = (a_1, a_2, \ldots, a_n) \in GF(q^m)^n,$$

the rank weight of a is by definition the rank of the $m \times n$-matrix over $GF(q)$ formed by extending every coordinate a_i on a basis of $GF(q^m)/GF(q)$. The construction is independent of the chosen basis.

The rank weight being a norm, it also defines a metric. With the distance related to the metric, we define the minimum rank distance of a linear code, in the same way as the classical minimum distance for a code in the Hamming metric.

Definition 2 *Let C be a linear code over $GF(q^m)$. The minimum rank distance of C is d where*

$$d = \mathrm{Min}_{c \in C^*}(\mathrm{Rk}(c)).$$

Given any matrix over $GF(q^m)$ we also define the rank of a matrix over $GF(q)$, and the minimum rank distance of a code

Definition 3 *Let X be a $k \times n$ matrix with coefficients in $GF(q^m)$. The rank of X over $GF(q)$ is equal to the maximum number of columns of X that are linearly independent over $GF(q)$.*

In 1985 Gabidulin, [6] published a family of codes which are optimal for the rank metric. Namely, they reach the "Singleton bound" for the rank metric.

One first chooses n elements $g_1, g_2, \ldots, g_n \in GF(q^m)^n$, which are linearly independent over q. Thus the matrix

$$G = \begin{pmatrix} g_1 & g_2 & \cdots & g_n \\ g_1^q & g_2^q & \cdots & g_n^q \\ \vdots & \vdots & \ddots & \vdots \\ g_1^{q^{k-1}} & g_2^{q^{k-1}} & \cdots & g_n^{q^{k-1}} \end{pmatrix}. \tag{1}$$

is of rank k. Let C be the linear code of length n and of dimension k generated by G.

Proposition 2 (see [6])

- *The linear code with generating matrix G reaches the Singleton bound for the rank metric. That is, let d be the minimum rank distance of C, we have*

$$d - 1 = n - k.$$

- *There exists a polynomial-time algorithm decoding up to $t = (d-1)/2$ errors. One step decoding has complexity $t(t^2 + 2n)$ multiplications in $GF(q^m)$.*

Definition 4 *The code C with generating matrix (1) is called Gabidulin code of dimension k, and of generating vector $g = (g_1, g_2, \ldots, g_n)$.*

3.2 Description of the GPT cryptosystem

The family of Gabidulin codes is optimal for the rank metric, and has a polynomial-time decoding algorithm [6]. Thus it is suitable to build McEliece like cryptosystems. We consider a finite field $GF(q^m)$. Usually the examples are taken with fields of characteristic 2, but they can be generalized to any characteristic without loss of generalities.

3.2.1 Construction of the cryptosystem

The conceiver of the system chooses as private key:

- A $k \times n$-generating matrix of a Gabidulin code under the form $G = \left(g_j^{q^i}\right)_{i=0,j=1}^{k-1,n}$.
- A $k \times k$-non singular matrix S with coefficients in $GF(q^m)$.
- A $k \times n$ matrix X of rank t_1 over $GF(q)$ – see definition 3. X is denoted distortion matrix and the integer t_1 is denoted *distortion parameter*.

The matrices (G, S, X) form the private key of the cryptosystem. The conceiver publishes:

$$(G_{pub} = SG + X,\ t_1)\,.$$

The encryption-decryption procedure is very similar to the encryption-decryption procedure in the McEliece system. Namely,

- Let c be a vector of length k. The sender computes and sends

$$y = cG_{pub} + e,$$

  where e is a random error-vector of length n and of rank $t - t_1$ over $GF(q)$.

- The receiver gets

$$y = cG_{pub} + e = cSG + (cX + e).$$

  By property of the rank weight, $\mathrm{Rk}(cX + e) \leq \mathrm{Rk}(cX) + \mathrm{Rk}(e)$. Since X is of rank t_1 over $GF(q)$, then the vector cX is of rank less than t_1. Since the rank weight of e is equal to $t - t_1$, we obtain

$$\mathrm{Rk}(cX + e) \leq t.$$

  Therefore, by applying the polynomial-time algorithm decoding t errors in the Gabidulin code, the receiver recovers cS. Then c by multiplying by S^{-1}, he gets c.

3.2.2 Bases enumeration attack

Contrary to McEliece system which relies on the difficulty of decoding in the Hamming metric, GPT system does not at all suffer from decoding attacks as described in [1, 11]. Indeed, since the rank of a word is the crucial thing, the Hamming weight of error-words e can be as high as possible, provided their rank is not greater than $t-t_1$. Since the most efficient decoding attacks need to pick up randomly a set of k non corrupted positions of the support of the intercepted word, it can easily be avoided in our case.

The only possibility to decrypt an intercepted word is called bases enumeration attack.

Let H_{pub} be a parity-check matrix of the code with generating matrix G_{pub}. Given the ciphertext y, the best way to cryptanalyze the system is first to evaluate the syndrome $s = H_{pub}^T y$ of the ciphertext y and second to find the unique vector e of length n and of rank $t - t_1$ such that

$$s = eH_{pub}^T. \tag{2}$$

Since e is of rank $t - t_1$, there exists a vector $E = (E_1, \ldots, E_{t-t_1}) \in GF(q^m)^{t-t_1}$ of maximum rank over $GF(q)$, and a $(t - t_1) \times n$-matrix S over $GF(q)$ of rank $(t - t_1)$ such that

$$e = ES. \tag{3}$$

Thus, by combining both equations, we have

$$s = ESH_{pub}^T, \tag{4}$$

for some 2-uple (E, S). Moreover, it is easily seen that whenever the 2-uple (E_0, S_0) is a solution of (4), then any 2-uple $(E_0U, U^{-1}S_0)$ is also a solution, where U runs for non-singular $(t - t_1) \times (t - t_1)$-matrix over $GF(q)$. The attacks consists in picking randomly a vector E and try to solve (4). The average number of tries to do is thus equal to

$$\left[\begin{array}{c} m \\ t-t_1 \end{array} \right] = \frac{(q^m-1)(q^m-q)\cdots(q^m-q^{t-t_1-1})}{(q^{t-t_1}-1)(q^{t-t_1}-q)\cdots(q^{t-t_1}-q^{t-t_1-1})}.$$

Since the numerator is greater than $q^{mt}(1-(t-t_1)/q^m)$, and since the denominator is less than $q^{(t-t_1)^2}$, we have

$$\left[\begin{array}{c} m \\ t-t_1 \end{array} \right] > q^{(m-t+t_1)(t-t_1)} \left(1 - \frac{t-t_1}{q^{m+1-t+t_1}}\right).$$

For practical use, $m > 2t$, the second term is negligible and the number of tries is lower bounded by $q^{(m-t+t_1)(t-t_1)}$. Since for each try one has to solve the linear system (4), the complexity of the attack is $\mathcal{O}\left(n(t-t_1)^3 q^{(m-t+t_1)(t-t_1)}\right)$ multiplications in $GF(q^m)$.

3.2.3 Structural attacks

Above in section 2.4 we saw that there existed an efficient structural attack recovering a decoder for the public code whenever the secret code is a generalized Reed-Solomon code. The reason is that the structure of GRS codes cannot be hidden in an efficient way. For the same reason, it can be proved that the distortion matrix of a GPT system plays a crucial role in its strength.

Given the public key $(G_{pub},\ t_1)$ of a GPT cryptosystem, the goal of a structural attack is to recover a decoder for the public code. Practically this means that one has to find three matrices G', S', X' such that

$$G_{pub} = S'G' + X',$$

where

- G' is a generating matrix under the form given by (1),
- S' is a non-singular matrix,
- X' is a distortion matrix of rank t_1 over $GF(q)$.

In 1995 and 1996, Gibson presented two complementary structural attacks against GPT cryptosystem, see [9, 10]. Let us denote by r the natural rank of the distortion matrix X. Note that necessarily t_1 the rank of X over $GF(q)$ is such that $t_1 \geq r$.

- *First attack*: the average complexity is greater than q^{mr} multiplications in $GF(q^m)$.

  The complexity q^{mr} of the first attack comes from the fact that any $k \times n$-matrix X of rank r over $GF(q^m)$ can be written $X = AB$, where A is a $k \times r$-matrix of full rank over $GF(q^m)$ and B is a $r \times n$-matrix of full rank over $GF(q^m)$.

- *Second attack*: the average complexity is greater than $k^3 + \max(0, t_1 - 2r)q^{\max(0, t_1-2r)(k+2)}$ multiplications in $GF(q^m)$.

  The complexity of the second attack partly derives from the fact that any $k \times n$-matrix X of rank t_1 over $GF(q)$ and of rank r over $GF(q^m)$ can be written

  $$X = A'B',$$

  where A' is a $k \times t_1$ matrix over $GF(q^m)$ of rank r and where B is a $t_1 \times n$ matrix over $GF(q)$ of rank t_1.

Both attacks are complementary since if r increases the first attack becomes more difficult whereas the second attack gets easier. Thus to prevent any attack, one should choose t_1 high and r high such that $t_1 - 2r$ remains high.

The ideal would be to find parameters such that the complexities of both structural attacks are roughly equivalent and are much greater than the complexity of the decoding attack, that is try to find m, k, t, t_1, r positive integers, such that

$$(t_1 - 2r)(k + 2) = mr \implies t_1 = \frac{m - 2k - 4}{k + 2} r,$$

Since $t_1 - 2r$ must be high enough to avoid the second attack we must have at least $(m - 2k - 4)/(k + 2) > 2$. Thus if one takes $k = 10$, then the extension of the code to choose is at least $m = 47$. Thus $t = 19$, and if we take $t_1 = 17$ and $r = 3$. These parameters could be a good choice. However there are two main problems

- the transmission rate of the system is very bad : $k/m = 0.213$.

Parameters	Size of the public key,	Transmission rate	Decoding	Gibson 1	Gibson 2
$m = 20,\ k = 11,\ t = 4$ $t_1 = 1,\ r = 1$	4 400 bits	0.55	2^{51}	2^{64}	2^{10}
$m = 46,\ k = 23,\ t = 12,$ $t_1 = 10,\ r = 2$	48 668 bits	0.5	2^{88}	2^{92}	2^{150}
$m = 47,\ k = 10,\ t = 19,$ $t_1 = 17,\ r = 3$	22 090 bits	0.22	2^{94}	2^{141}	2^{132}

Table 1: Lower bound on the average complexity of known attacks

- the extension degree of the field increases. For example, if we consider fields of characteristic 2, the cost of one multiplication in $GF(2^m)$ cannot be made smaller in software implementation than $m\text{Log } m$ binary operations, provided the field is too big to be stored on any media.

In Table 1 we sum up the results of the complexity of the different attacks for various parameters.

For practical interest, the smaller the better, the public-key size should be the smallest possible by keeping a enough security against the known attacks.

By choosing carefully the parameters, one can assure the security of the system, however the size of the public-key is in this case prohibitive. Namely for the second example in table 1, $m = n = 46,\ k = 23$ the size of the key is 48 668 bits.

Even if the size of the public-key is ten times smaller than that of the McEliece system, it is not satisfactory, since it remains big, and the calculus have to be computed into fields that are becoming very large.

3.3 A Niederreiter-type system

The idea of modifying the GPT system so that it resists the Gibson attacks without increasing too much the size of the public-key is not new and is part of the work accomplished by Gabidulin and Ourivski and can be found in [7]. However they still use the McEliece form of the cryptosystem by publishing a generating matrix of some scrambled subcode of a Gabidulin code.

Our approach is to show that a simple Niederreiter type cryptosystem can be constructed, based on the property of the rank metric. To hide the structure of the chosen Gabidulin codes, we use the general idea of the first section, that is we take a subcode of the Gabidulin code by adding some extra lines to the parity-check matrix. This approach has the advantage to be simple. Moreover, by choosing carefully the lines, we show that the system can efficiently resist to all known attacks, still keeping a low public-key size. Moreover the distortion matrix is unnecessary.

3.3.1 Presentation of the system

The conceiver of the system chooses as private key:

- A $(d-1)\times n$-parity check matrix $H=\left(h_i^{q^j}\right)_{i=1,j=0}^{n,d-2}$ of a Gabidulin code C over $GF(q^m)$. Hence C has minimum rank distance d, and error-correcting capability $t=\lfloor (d-1)/2\rfloor$.
- A $\ell\times n$-matrix A with coefficients in $GF(q^m)$ whose properties will be discussed below. A is the matrix which enabling to control the security of the system, against all attacks. Its parameters are thus crucial.
- A $(d-1+\ell)\times(d-1+\ell)$-non singular matrix S over $GF(q^m)$.

Let

$$H_{pub}=S\left(\frac{H}{A}\right).$$

H_{pub} is a parity-check matrix of a subcode of C whose structure is controlled by the matrix A.
The conceiver publishes (H_{pub},t).
The encryption-decryption procedure is the following:

- Let c be the message of length n and of rank t, that has to be encrypted. The sender computes and sends
$$s=cH_{pub}^T.$$
- The receiver gets
$$s=\left(cH^T\mid cA^T\right)S^T,$$
and need only to compute $s(S^T)^{-1}=\left(cH^T\mid cA^T\right)$. Then he decodes the syndrome cH^T thanks to the polynomial-time decoding algorithm of the Gabidulin code C of parity-check matrix H.

For the complexity of the system we have:

- *Complexity of the encryption*: $\ell(d-1)$ multiplications in $GF(q^m)$.
- *Complexity of the decryption*: we have one decoding in the code C and a multiplication by a $(2t+\ell)\times(2t+\ell)$ matrix over $GF(q^m)$. Thus it is equal to $t^3+nt+(2t+\ell)^2$ multiplications in $GF(q^m)$.

Proposition 3 *The transmission rate of the system is equal to*

$$\tau=\frac{Log_qN_{m,t}+Log_qN_{n,t}-Log_qN_{t,t}}{m(d-1+\ell)},\tag{5}$$

where

$$N_{i,j}=\prod_{s=0}^{j-1}(q^i-q^s),\quad j<i.$$

Proof : By definition the transmission rate τ of the system is equal to the logarithm in base q of the number of transmissible words, divided by the logarithm in base q of the number of possible words, that is

$$\tau=\frac{\mathrm{Log}_qN}{\mathrm{Log}_qq^{m(2t+\ell)}},$$

where N denotes the number of words over $GF(q^m)$ of length n and of rank t. By definition, this number is exactly equal to the number of matrices of size $m \times n$ over $GF(q)$ and of rank t, that is

$$N = \left[\begin{array}{c} m \\ t \end{array} \right] N_{n,t} = \frac{N_{m,t}}{N_{t,t}} N_{n,t}.$$

□

Provided m and n are significantly larger than t, the transmission rate can be approximated by

$$\tau \approx \frac{t(m+n-t)}{m(d-1+\ell)}.$$

3.3.2 Encoding procedure for the messages

The transmission rate of the system is given by (5), provided every word of length n and of rank t corresponds to a message. Messages consist usually in continuous blocks of bits. In this part we consider fields of characteristic 2, and describe a procedure enabling to encode messages into matrices of size $m \times n$ and of rank t, corresponding thus to a decodable syndrome. A necessary condition being that the encoding-decoding procedure is faster than the encryption-decryption procedure.

For the binary case the procedure is the following. We encode blocks of $\ell = t(m+n-2t)$ bits into words of length n, of weight t over $GF(2^m)$. Let f be a received word of $t(m+n-2t)$ bits. First the block f is splitted into two blocks f_1 and f_2 respectively of length $t(n-t)$ and $(m-t)t$. Then

1. $f_1 = \left(f_{1,1}, f_{1,2}, \ldots, f_{1,t(n-t)}\right)$ is viewed as a $t \times (n-t)$-matrix F_1 where

$$F_1 = \begin{pmatrix} f_{1,1} & f_{1,2} & \cdots & f_{1,n-t} \\ f_{1,n-t+1} & f_{1,n-t+2} & \cdots & f_{1,2(n-t)} \\ \vdots & \vdots & \ddots & \vdots \\ f_{1,(t-1)(n-t)+1} & f_{1,(t-1)(n-t)+2} & \cdots & f_{1,t(n-t)} \end{pmatrix}.$$

2. $f_2 = \left(f_{2,1}, f_{2,2}, \ldots, f_{2,(m-t)t}\right)$ is viewed as a $(m-t) \times t$-matrix F_2 where

$$F_2 = \begin{pmatrix} f_{2,1} & f_{2,2} & \cdots & f_{2,t} \\ f_{2,t+1} & f_{2,n-t+2} & \cdots & f_{2,2t} \\ \vdots & \vdots & \ddots & \vdots \\ f_{2,(m-t-1)t+1} & f_{2,(m-t-1)t+2} & \cdots & f_{2,(m-t)t} \end{pmatrix}.$$

3. The encoding matrix is thus the $m \times n$-binary matrix of rank t:

$$F = \left(\begin{array}{c|c} I_{t\times t} & F_1 \\ -- & -- \\ F_2 & F_2F_1 \end{array} \right). \tag{6}$$

The complexity of the encoding-decoding procedure is:

- *Encoding* : The cost is equal to the multiplication of the matrices F_2 and F_1, that is on average
$$\frac{mnt-(m+n)t^2+t^3}{2} \text{ binary multiplications}$$
Recall that the encryption costs $n(d-1+\ell)$ multiplications over $GF(2^m)$. Experimentally, the size of $m \geq 32$ prevents from putting the whole field in memory. Thus underestimating the cost of a multiplication by $m\text{Log}_2 m$ seems legitimate. Thus the complexity of the encryption is roughly, with $d-1=2t$
$$mn(2t+\ell)\text{Log}_2 m \text{ binary operations.}$$
With $m=32$, and $t=\ell=4$, the encoding procedure is more than 100 times faster than the encryption procedure.

- *Decoding* : It consists in reading both matrices F_1 and F_2. It is linear in complexity. Thus the decoding is negligible compared to the decryption procedure.

In that case the practical transmission rate of the system is equal to
$$\tau_p = \frac{t(m+n-2t)}{m(2t+\ell)}.$$

Hence the loss between the theoretical and the practical transmission rate is approximately equal to
$$\tau - \tau_p \approx \frac{t}{m(2t+\ell)}.$$

3.3.3 Controlling the security by using subcodes

This new system is based on the theory of Gabidulin codes. Such cryptosystems can be attacked with the bases enumeration attack and by using the Gibson attacks. Since we require that the cryptosystem be strong against such attacks, this provides us conditions on the matrix A.

The scheme can be made resistant against bases enumeration attacks even for small parameters. Namely, the only condition required is on the parameter d of the system, that has to be large enough.

However, we want it to be resistant to Gibson attacks. The first step is to decompose the public-key into the sum of a parity-check matrix of a Gabidulin code and of some distortion matrix, that is
$$H_{pub} = SH + \underbrace{SY}_{X},$$
where with $[i]=q^i$,
$$H = \begin{pmatrix} h_1 & \cdots & h_n \\ h_1^{[1]} & \cdots & h_n^{[1]} \\ \vdots & \cdots & \vdots \\ h_1^{[d+\ell-2]} & \cdots & h_n^{[d+\ell-2]} \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \\ a_{11}-h_1^{[d-1]} & \cdots & a_{1n}-h_n^{[d-1]} \\ \vdots & \ddots & \vdots \\ a_{\ell 1}-h_1^{[d+\ell-2]} & \cdots & a_{\ell n}-h_n^{[d+\ell-2]} \end{pmatrix}.$$

Under this form we can evaluate the complexity of the attacks described in section 3.2.3 against this system. We saw that the complexity of the Gibson attacks depended on the rank t_1 of the matrix X over $GF(q)$ and on its rank r over $GF(q^m)$. Namely, if $k = d - 1 + \ell$ we have:

- first Gibson attack: $\mathcal{O}\left((n-k)k^3q^{rm}\right)$ multiplications in $GF(q^m)$,
- second Gibson attack: $\mathcal{O}\left((k+t+2)(t_1-2r)q^{(t_1-2r)(k+2)}\right)$ multiplications in $GF(q^m)$.

The goal of the conceiver is to choose A such that the rank of X over $GF(q)$ is high and its rank over $GF(q^m)$ is small. Thus the parameters of X depend only on the matrix Y, which itself depends on the matrix A.

Regarding the complexity of the different attacks, the ideal is first to have the parameter t_1 the highest possible, that is $t_1 = n$. Recall that by definition of the distortion parameter, we have

$$t_1 = \mathrm{Max}_{c \in GF(q^m)^k}(\mathrm{Rk}(cX)).$$

It is very easy to pick up matrices A such that

$$\mathrm{Rk}(X|GF(q)) = m, \quad \mathrm{Rk}(X|GF(q^m)) = \ell.$$

One way to proceed is the following:

1. Choose the first line of A such that the rank over $GF(q)$ of the vector

   $$\mathbf{y}_1 = (a_{11} - h_1^{[d-1]}, a_{12} - h_2^{[d-1]}, \ldots, a_{1n} - h_n^{[d-1]}),$$

   is equal to n. One way to proceed just choose randomly the vector (a_{1j}) until $\mathbf{y}_1$ is of rank n. Namely, the number of vectors of length n in $GF(q^m)$ and of rank n over $GF(q)$ is equal to

   $$N_{m,n} = (q^m - 1)(q^m - q)\cdots(q^m - q^{n-1}).$$

   Thus the probability that a random chosen vector of length n in $GF(q^m)$ is of rank n exactly is equal to

   $$P = \frac{N_{m,n}}{q^{mn}} = \left(1 - \frac{1}{q^m}\right)\left(1 - \frac{1}{q^{m-1}}\right)\cdots\left(1 - \frac{1}{q^{m+1-n}}\right).$$

   With $q \geq 2$, and by choosing $m = n$, we have :

   $$P = 1 - \frac{1}{q} - \frac{1}{q^2} + o\left(\frac{1}{q^4}\right),$$

   That is P is greater than $1/4$. Hence it takes at most 4 tries on average to get a vector of rank n.

2. Choose the $\ell - 1$ remaining lines of A such that Y of maximum rank ℓ.

Now this proposition gives what we need to evaluate the complexity of Gibson attacks against the modified Niederreiter system.

Parameters	Size of the key	Transmission rate	Bases enumeration	Gibson 1	Gibson 2
$m = n = 25,\ t = 4,\ \ell = 4$	3 850 bits	0.56	2^{92}	2^{111}	2^{212}
$m = n = 25,\ t = 5,\ \ell = 5$	3 750 bits	0.53	2^{136}	2^{137}	2^{233}
$m = n = 32,\ t = 4,\ \ell = 4$	7 680 bits	0.58	2^{123}	2^{138}	2^{393}

Table 2: Complexity of the different attacks against the modified Niederreiter cryptosystem

Proposition 4 *Let the $\ell \times n$-matrix A be of rank ℓ over $GF(q^m)$ and of rank n over $GF(q)$. The $(d-1+\ell) \times n$-distortion matrix X is of rank ℓ over $GF(q^m)$ and of rank n over $GF(q)$*

Proof : We have $X = SY$, where

$$Y = \begin{pmatrix} 0 \\ A \end{pmatrix}.$$

The rank of Y is thus equal to ℓ. Moreover by definition 3 the fact that a line vector of Y – a line vector of A – is of rank n is enough to prove that Y is of rank n over $GF(q)$. Thus Y can be written as $Y = Y_0 B$, where Y_0 is a $(d-1+\ell) \times n$ matrix of rank ℓ, and B is a $n \times n$-non singular matrix over $GF(q)$. Hence

$$X = SY_0 B,$$

where SY_0 is of rank ℓ. Thus X has rank ℓ over $GF(q^m)$ and rank n over $GF(q)$ □

Proposition 5 *For the modified Niederreiter cryptosystem, with $d = 2t$, the complexity of the attacks is the following.*

- *Bases enumeration : $\mathcal{O}\left(nt^3 q^{t(m-t)}\right)$ multiplications in $GF(2^m)$.*
- *First Gibson attack : $\mathcal{O}((2t+\ell)^3 q^{\ell m})$ multiplications in $GF(2^m)$.*
- *Second Gibson attack : $\mathcal{O}\left((3t+\ell+2)(n-2\ell)q^{(n-2\ell)(2t+\ell)}\right)$ multiplications in $GF(2^m)$.*

The proof derives directly from the replacement of d by $2t$, r by ℓ and t_1 by n in the formulas giving the complexity of the different attacks.

3.4 Examples

In Table 2, we give the different parameters of the modified Niederreiter system for some practical parameters. The size of the public-key is given for the reduced form of the public matrix. First note that the size of the public key is not really a criterion for evaluating the security of the system against known attacks. Namely the second line provides the best choice. The public-key is very small, and the system is extremely secure against every type of attacks.

References

[1] A. Canteaut and F. Chabaud. A new algorithm for finding minimum-weight words in a linear code: Application to McEliece's cryptosystem and to narrow-sense BCH codes of length 511. *IEEE Transactions on Information Theory*, 44(1):367–378, January 1998.

[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] 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.

[4] A. Dür. The automorphism group of Reed–Solomon codes. *Journal of Combinatorial Theory, series A*, 4(1), 1987.

[5] A. Ourivski E. Gabidulin and V. Pavlouchkov. On the modified niederreiter cryptosystem. preprint.

[6] E. M. Gabidulin. Theory of codes with maximal rank distance. *Problemy Peredachi Informatsii*, 21:1–12, July 1985. in russian.

[7] E. M. Gabidulin and A. V. Ourivski. Improved GPT public-key cryptosystems. In P. Farrell, M. Darnell, and B. Honary, editors, *Coding Communications and Broadcasting*, pages 73–102, 2000.

[8] E .M. Gabidulin, A. V. Paramonov, and O. V. Tretjakov. Ideals over a non-commutative ring and their application in cryptology. *LNCS*, 573:482 – 489, 1991.

[9] J. K. Gibson. Severely denting the Gabidulin version of the McEliece public-key cryptosystem. *Designs, Codes and Cryptography*, 6:37–45, 1995.

[10] J. K. Gibson. The security of the Gabidulin public-key cryptosystem. In U. Maurer, editor, *EUROCRYPT'96*, pages 212–223, 1996.

[11] P. J. Lee and E. F. Brickell. An observation on the security of McEliece's public-key cryptosystem. In C. G. Günter, editor, *Advances in Cryptology - EUROCRYPT'88*, volume 330 of *LNCS*, pages 275–280. Springer-Verlag, 1988.

[12] P. Loidreau and N. Sendrier. Weak keys in McEliece public-key cryptosystem. *IEEE Transactions on Information Theory*, 47(3), March 2001.

[13] F. J. MacWilliams and N. J. A. Sloane. *The Theory of Error–Correcting Codes*. North Holland, 1977.

[14] R. J. McEliece. A public-key cryptosystem based on algebraic coding theory. Technical report, Jet Propulsion Lab. DSN Progress Report, 1978.

[15] H. Niederreiter. Knapsack-type cryptosystems and algebraic coding theory. *Problems of Control and Information Theory*, 15(2):159 – 166, 1986.

[16] V. M. Sidel'nikov and S. O. Shestakov. On cryptosystems based on generalized Reed-Solomon codes. *Discrete Mathematics*, 4(3):57–63, 1992. en russe.