RankSynd a PRNG Based on Rank Metric

Philippe Gaborit, Adrien Hauteville, Jean-Pierre Tillich

RankSynd a PRNG Based on Rank Metric

Philippe Gaborit[1], Adrien Hauteville[1,2], and Jean-Pierre Tillich[2]

[1] Université de Limoges, XLIM-DMI, 123, Av. Albert Thomas, 87060 Limoges, Cedex, France
[2] Inria, Domaine de Voluceau, BP 105, Le Chesnay 78153, France

Abstract. In this paper, we consider a pseudo-random generator based on the difficulty of the syndrome decoding problem for rank metric codes. We also study the resistance of this problem against a quantum computer. Our results show that with rank metric it is possible to obtain fast PRNG with small public data, without considering additional structure for public matrices like quasi-cyclicity for Hamming distance.

1 Introduction

Pseudo-random number generators (PRNG) are an essential tool in cryptography. They can be used for one-time cryptography or to generate random keys for cryptosystems. A long series of articles have demonstrated that the existence of a PRNG is equivalent to the existence of one-way functions [29,22,19]. Basically, a one-way function is a function which is easy to compute but hard to invert.

There are two types of PRNG in cryptography. The first one is based on block cipher schemes, like AES for instance, used in OFB mode. This gives in general very fast random generators. The second type includes PRNG proven to be secure by reduction to a hard problem. The problems considered can be based on classical problems from cryptography, like factorization or discrete logarithm, [6,5] or they may be based on linear algebra, like coding theory [11] or lattices [1] or multivariate quadratic systems [2].

Recent works [13,26] have proven that PRNG based on the syndrome decoding (SD) problem could be almost as fast as PRNG based on AES. However the PRNG based on the SD problem have to store huge matrices. This problem can be solved with the use of quasi-cyclic codes but there is currently no proof of the hardness of the SD problem for quasi-cyclic codes. Moreover recent quantum attacks on special ideal lattices [10], clearly raise the issue of the security of quasi-cyclic structures for lattices and codes, even if a straight generalization of this quantum attack from cyclic structures to quasi-cyclic structures seems currently out of reach.

Code-based cryptography has been studied for many years, since the proposal of the McEliece cryptosystem [25]. This type of cryptography relies on the difficulty of the SD problem for Hamming distance, which is proven NP-hard [3]. Besides this particular metric, other metrics may be interesting for cryptographic purposes. For instance, the rank metric leads to SD problems whose complexity grows very fast with the size of parameters. In particular, recent advances in this field have shown that the problem of decoding general codes in rank metric is hard [15]. Moreover the best known attacks have an exponential complexity with a quadratic term in the exponent. In practice it means that it is possible to obtain cryptosystems with keysizes of only a few thousand bits and without additional structure such as cyclicity (or quasi-cyclicity). This is particularly interesting since it avoids relying on the hardness of structured problems whose security is less known than the security of general instances.

In this paper we study the case of a PRNG based on general instances of the Rank Syndrome Decoding problem. We build a PRNG based on the rank metric which has both a reasonable data size (a few thousand bits), which is reasonably fast and which is asymptotically better than PRNG based on the Hamming metric without cyclic structure. It is possible to optimize separately each of these aspects, like the size in constrained environments such as chip cards. We prove that breaking our PRNG is not easier than breaking the Fischer-Stern PRNG[11]. We also study how a quantum computer can be used to speed up the best known combinatorial attacks on the rank syndrome decoding problem. In the last section, we give parameters for our system, against classical and quantum attacks.

2 Generalities on the rank metric

First, let us define the central notion of this paper, namely matrix codes

Definition 1 (matrix code). *A matrix code $\mathcal{C}$ of length $m \times n$ over $\mathbb{F}_q$ is a subspace of the vector space of matrices of size $m \times n$ with entries in $\mathbb{F}_q$. If $\mathcal{C}$ is of dimension K, we say that $\mathcal{C}$ is an $[m \times n, K]_q$ matrix code, or simply an $[m \times n, K]$ code if there is no ambiguity.*

The difference between an $[m \times n, K]$ matrix code and a code of length mn and dimension K is that it allows to define another metric given by $d(A, B) \stackrel{\text{def}}{=} \text{Rank}(A - B)$. The weight of a word $\boldsymbol{c}$ is equal to $w_R(\boldsymbol{c}) \stackrel{\text{def}}{=} d(c, 0)$. Linear codes over an extension field $\mathbb{F}_{q^m}$ give in a natural way matrix codes, and they have in this case a very compact representation which allows to decrease key sizes.

Definition 2 (matrix code associated to an $\mathbb{F}_{q^m}$-linear code). *Let $\mathcal{C}$ be an $[n,k]$ linear code over $\mathbb{F}_{q^m}$. Each word $\boldsymbol{c}$ of $\mathcal{C}$ can be associated to an $m \times n$ matrix over $\mathbb{F}_q$ by representing each coordinate $\boldsymbol{c}_i$ by a column vector $(c_{i1}, \ldots, c_{im})^T$ where $\boldsymbol{c}_i = \sum_{j=1}^{m} c_{ij}\beta_j$ with $\beta_1, \ldots, \beta_m$ being an arbitrary basis of $\mathbb{F}_{q^m}$ viewed as a vector space over $\mathbb{F}_q$ and $c_{ij} \in \mathbb{F}_q$. In other words the c_{ij}'s are the coordinates of $\boldsymbol{c}_i$ in this basis. The matrix code associated to $\mathcal{C}$ is of type $[m \times n, km]_q$.*

By definition, the weight of a word $\boldsymbol{c} \in \mathcal{C}$ is the rank of its associated matrix. It does not depend on the choice of the basis. Such matrix codes have a more compact representation than generic matrix codes. Indeed an $[n,k]$ $\mathbb{F}_{q^m}$-linear code can be described by a systematic parity-check matrix over $\mathbb{F}_{q^m}$, which requires $k(n-k)m \lceil \log q \rceil$ bits, whereas a representation of an $[m \times n, km]_q$ matrix code requires in general $km(mn-km) \lceil \log q \rceil = k(n-k)m^2 \lceil \log q \rceil$ bits. In other words we can reduce the size of the representation of such codes by a factor m if we consider the subclass of matrix codes obtained from $\mathbb{F}_{q^m}$-linear codes.

There is also a notion of Gilbert-Varshamov distance for the rank metric. For the Hamming metric, the Gilbert Varshamov distance for $[n,k]_q$ codes corresponds to the "typical" minimum distance of such codes. It is given by the smallest t for which $|B_t^{\mathrm{H}}| \geq q^{n-k}$ where B^{H} is the ball of radius t centered around 0 for the Hamming metric. The Gilbert-Varshamov distance for $[m \times n, km]_q$ matrix codes in the rank metric is given by the smallest t for which

$$|B_t^{\mathrm{R}}| \geq q^{m(n-k)}$$

where B^{R} is the ball of radius t centered around 0 for the rank metric (in other words it is the set of $m \times n$ matrices over $\mathbb{F}_q$ of rank $\leq t$). It is readily checked that (see [24])

$$|B_t^{\mathrm{R}}| \approx q^{t(m+n-t)}$$

which gives $d_{GV} \approx \frac{m+n-\sqrt{(m+n)^2-4m(n-k)}}{2}$.

3 Cryptography based on rank metric

3.1 A difficult problem

Similarly to the syndrome decoding problem for the Hamming metric we can define the rank syndrome decoding (RSD) problem.

Problem 1 (Rank Syndrome Decoding). Let $\mathcal{C}$ be an $[n,k]$ $\mathbb{F}_{q^m}$-linear code, w an integer and $s \in \mathbb{F}_{q^m}^{n-k}$. Let $\boldsymbol{H}$ be a parity-check matrix of $\mathcal{C}$. The problem is to find a word $\boldsymbol{e} \in \mathbb{F}_{q^m}^n$ such that

$$\begin{cases} \boldsymbol{H}e^T &= s \\ w_R(\boldsymbol{e}) = w \end{cases}$$

Recently it was proven in [15] that this problem had a probabilistic reduction to the Syndrome Decoding problem for the Hamming distance which is known to be NP-complete. This substantiates claims on the hardness of this problem.

3.2 Complexity of practical attacks

The complexity of practical attacks grows quickly with the size of parameters, there is a structural reason for this: for the Hamming distance a key notion in the attacks is counting the number of words of length n and support size t, which corresponds to the notion of Newton binomial coefficient $\binom{n}{t}$, whose value is exponential in n for a fixed ratio t/n, since $\log_2 \binom{n}{t} = nh(t/n)(1+o(1))$ where $h(x) \stackrel{\text{def}}{=} -x\log_2 x - (1-x)\log_2(1-x)$. In the case of the rank metric, counting the number of possible supports of size w for a matrix code associated to an $\mathbb{F}_{q^m}$-linear code of length n corresponds to counting the number of subspaces of dimension w in $\mathbb{F}_{q^m}$. This is given by the Gaussian binomial coefficient $\begin{bmatrix} m \\ r \end{bmatrix}_q$. In this case $\log_q \begin{bmatrix} m \\ r \end{bmatrix}_q = w(m-w)(1+0(1))$. Again this number behaves exponentially but the exponent is quadratic. This is of course to be compared to the "real" length of the matrix code which is also quadratic: $m \times n$.

The approaches that have been tried to solve this problem fall into two categories:

- **combinatorial approach**: this approach gives the best results for small values of q (typically $q = 2$) and for large values of n and k. When q becomes large, they become less efficient however. The first non-trivial combinatorial algorithm for the RSD problem was proposed in 1996 (see [8]), then in 2002 Ourivski and Johannson [27] improved it. However for both of the algorithms suggested in [27] the exponent of the complexity does not involve n. Recently these two algorithms were generalized in [14] by Gaborit et *al.* with a complexity in $\mathcal{O}\big((n-k)^3 m^3 q^{(w-1)\left\lceil \frac{(k+1)m}{n} \right\rceil}\big)$. Notice that the exponent involves now n and when $n > m$ the exponent becomes better than the one in [27].

- **algebraic approach**: the particular nature of rank metric makes it a natural field for algebraic system solving by Groebner bases. The

complexity of these algorithms is largely independent of the value of q and in some cases may also be largely independent from m. These attacks are usually the most efficient ones when q becomes large. There exist different types of algebraic modeling for the rank metric decoding problem. The algebraic modeling proposed by Levy and Perret [23] in 2006 considers a quadratic system over $\mathbb{F}_q$ by taking as unknowns the support E of the error and the error coordinates regarding E. There are also other ways of performing the algebraic modeling: the Kernel attack [9,17], the Kipnis-Shamir modeling [21] or the minor approach (see [28] for the most recent results on this topic). The last one uses the fact that the determinant of minors of size greater than w is zero to derive algebraic equations of degree $w+1$. All of these proposed algorithms can be applied to the RSD problem but they are based on an algebraic modeling in the base field $\mathbb{F}_q$ so that the number of unknowns is always quadratic in n (for $m = \Theta(n)$ and $w = \Theta(n)$), so that the general complexity for solving these algebraic equations with Groebner basis techniques is exponential in $\mathcal{O}(n^2)$.

More recently, a new algebraic modeling based on a annulator approach was proposed by Gaborit et *al.* in [14]. It yields multivariate sparse equations of degree q^{r+1} but on the extension field $\mathbb{F}_{q^m}$ rather than on the base field $\mathbb{F}_q$ and results in a drastic reduction of the number of unknowns. The latter attack is based on the notion of q-polynomial and is particularly efficient when w is small. Moreover all these attacks can be declined in a hybrid approach where some unknowns are guessed but asymptotically they are less efficient than other approaches.

Overall, all the known attacks for solving the RSD problem in the case where $m = \mathcal{O}(n), w = \mathcal{O}(n)$ have a complexity in $2^{\mathcal{O}(n^2)}$. Moreover because of the behavior of the Gaussian binomial coefficient and because of the number of unknowns for algebraic solving, it seems delicate to do better.

4 One-way functions based on rank metric

We use here the hardness of the RSD problem to build a family of one-way functions based on this problem. Let us start by recalling the definition of a strongly one-way function (see [12, Definition 1]):

Definition 3. *A collection of functions* $\{f_n : E_n \to \mathbb{F}_2^{k_n}\}$ *is called strongly one way if :*

- *there exists a polynomial-time algorithm which computes* $f_n(x)$ *for all* $x \in E_n$

– *for every probabilistic polynomial-time algorithm A, for all $c > 0$ and for sufficiently large n, $Prob\big(A(f_n(x)) \in f_n^{-1}(f_n(x))\big) < \dfrac{1}{n^c}$*

We will consider the following family :
$E_{n,k} = \{(\boldsymbol{H}, \boldsymbol{y}) : \boldsymbol{H} \in \mathbb{F}_{q^n}^{(n-k)\times n}, \boldsymbol{y} \in \mathbb{F}_{q^n}^n, w_R(\boldsymbol{y}) = w_n\}$

$$\begin{aligned} f : \ E_{n,k} &\to \mathbb{F}_{q^n}^{(n-k)\times(n+1)} \\ (\boldsymbol{H}, \boldsymbol{y}) &\mapsto (\boldsymbol{H}, \boldsymbol{H}\boldsymbol{y}^T) \end{aligned}$$

We take $m = n$ so that the first algorithm of [14] does not improve the complexity of [27]. These functions should be strongly one-way if we choose $w_n \approx d_{GV}(n,k)$ which corresponds to the range where there is basically in general a unique preimage.

5 A PRNG based on rank metric codes

5.1 Description of the generator

Now that we have a family of one-way functions based on a hard problem, our goal is to use them to build a PRNG which will inherit of that hardness. We begin by letting $k = Rn$ and $w = \omega n$ for some constant R and ω. The security and the complexity of computing the pseudo-random sequence associated to this generator will then be expressed as a function of n, with R and ω as parameters.

First it is necessary to expand the size of the input, so that the number of syndromes becomes larger than the number of words of weight w_n. By definition, these two numbers are equal when $w = d_{GV}$ so that we can choose $\omega < \frac{d_{GV}}{n}$. The size of the input is $n(n-k)n\lceil \log q \rceil = n^3(1-R)\lceil \log q \rceil$ for $\boldsymbol{H}$ plus $w_n(2n-w_n)\lceil \log q \rceil = n^2(2\omega-\omega^2)\lceil \log q \rceil$ for $\boldsymbol{y}$ and the size of the output is $n^3(1-R)\lceil \log q \rceil + n^2(1-R)\lceil \log q \rceil$. So the function f_n expands the size of the input by $n^2(1-R-2\omega+\omega^2)\lceil \log q \rceil = \mathcal{O}\big(n^2\big)$ bits. To compute $f_n(\boldsymbol{H}, \boldsymbol{y})$ one has to perform a product matrix-vector in a field of degree n, which costs $\mathcal{O}\big(n^3\big)$ operations in $\mathbb{F}_q$.

Secondly we need an algorithm which computes a word $\boldsymbol{y} \in \mathbb{F}_{q^n}^n$ of weight ωn with $n^2(2\omega-\omega^2)\lceil \log q \rceil$ bits. This can be done very easily. According to Definition 2, $\boldsymbol{y}$ can be seen as an $n \times n$ matrix M over $\mathbb{F}_q$ of rank ωn. Let $\beta = (\beta_1, \ldots, \beta_{\omega n})$ be a basis of the subspace generated by the rows of M. We can represent β by a matrix $B \in \mathbb{F}_q^{\omega n \times n}$. There exists a unique matrix $A \in \mathbb{F}_q^{nn}$ such that $M = AB$. In order to ensure the unicity of this representation, we need to take B in its echelon form B_{ech}, then $M = A'B_{ech}$ for some matrix A'. Unfortunately, it is not so easy to

enumerate all the echelon matrices efficiently. To avoid this problem, we only generate words with a certain form, as it is done for SYND [13].

Definition 4 (Regular Rank Words). *A word* $\boldsymbol{y} \in \mathbb{F}_{q^n}^n$ *of weight* r *is said regular if its associated matrix* $M \in \mathbb{F}_q^{n\times n}$ *is of the form*

$$M = A \begin{pmatrix} 1 & & \\ & \ddots & C \\ & & 1 \end{pmatrix}$$

with $A \in \mathbb{F}_q^{n\times r}$ *and* $C \in \mathbb{F}_q^{r\times(n-r)}$

The probability that a word of weight r is regular is equal to the probability that a $r \times r$ matrix over $\mathbb{F}_q$ is invertible. This probability is greater than a constant $c > 0$ for all r and q. Thus it is not harder to solve the RSD problem in the general case than to solve the RSD problem by restraining it to the regular words, since if a polynomial algorithm could solve the RSD problem in the case of regular words then it would also give an algorithm solving the RSD problem with a probability divided by a constant, hence the RSD problem with regular words remains hard.

Algorithm 1: Expansion Algorithm

Input: $n^2(2\omega - \omega^2)\lceil \log q \rceil$ bits
Output: $\boldsymbol{y} \in \mathbb{F}_{q^n}^n, w_R(\boldsymbol{y}) = \omega n$
Data: A basis $(\beta_1, \ldots, \beta_n)$ of $\mathbb{F}_{q^n}/\mathbb{F}_q$
begin
  compute $x \in \mathbb{F}_q^{n^2(2\omega-\omega^2)}$ with the input bits;
  compute $A \in \mathbb{F}_q^{n\times\omega n}$ with the first ωn^2 coordinates of x;
  compute $B \in \mathbb{F}_q^{\omega n\times(n-\omega n)}$ with the last coordinates of x;
  $B \leftarrow (I_{\omega n}|B)$ `/* this is the concatenation of two matrices */`;
  $M \leftarrow AB$;
  $\boldsymbol{y} \leftarrow (\beta_1, \ldots, \beta_n)M$;
  return $\boldsymbol{y}$;

The most expensive step of this algorithm is the matrix product which takes ωn^3 operations in $\mathbb{F}_q$, so its overall complexity is $\mathcal{O}(n^3)$.

With these two functions, we can construct an iterative version of the generator which can compute as many bits as we want.

Algorithm 2: Our Pseudo-Random Generator

Input: a vector $\boldsymbol{x} \in \mathbb{F}_q^K$ where K is the security parameter
Output: N pseudo-random bits
Data: a random matrix in systematic form $\boldsymbol{H} \in \mathbb{F}_{q^n}^{(1-R)n \times n}$, an initialization vector $\boldsymbol{v} \in \mathbb{F}_q^{n^2(2\omega-\omega^2)-K}$
begin
  $\boldsymbol{y} \leftarrow \text{Expansion}(\boldsymbol{x} \| \boldsymbol{v})$;
  repeat
    $s \leftarrow \boldsymbol{H}\boldsymbol{y}^T$;
    split s into two strings of bits s_1 and s_2, with s_1 of length $n^2(2\omega - \omega^2)\lceil \log q \rceil$;
    output s_2;
    $\boldsymbol{y} \leftarrow \text{Expansion}(s_1)$;
  until *the number of bits generated* $> N$;

5.2 Security of the generator

We recall that a distribution is pseudo-random if it is polynomial-time indistinguishable from a truly random distribution. If our generator were not pseudo-random, then there would exist a distinguisher D_R which distinguishes a sequence produced by our generator from a truly random sequence with a non-negligible advantage. We can use this distinguisher to build another distinguisher for the Fischer-Stern generator [11]. That generator is proven pseudo-random if syndrome decoding in the Hamming metric is hard [3]. It takes as input a parity-check matrix $M \in \mathbb{F}_2^{k \times n}$ of a random code and a vector $\boldsymbol{x} \in \mathbb{F}_2^n$ of Hamming weight d, with d smaller than the Gilbert-Varshamov bound (in the Hamming metric) of the code and outputs $(M, M\boldsymbol{x}^T)$.

We need a method to embed an $\mathbb{F}_q$-linear code into an $\mathbb{F}_{q^m}$-linear code. We use the same technique as in [15].

Definition 5. *Let $m \geqslant n$ and $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n) \in \mathbb{F}_{q^m}^n$. We define the embedding of $\mathbb{F}_q^n$ into $\mathbb{F}_{q^m}^n$ by :*

$$\begin{aligned} \psi_{\boldsymbol{\alpha}} : \quad \mathbb{F}_q^n &\to \mathbb{F}_{q^m}^n \\ (x_1, \ldots, x_n) &\mapsto (\alpha_1 x_1, \ldots, \alpha_n x_n) \end{aligned} \tag{1}$$

For every $\mathbb{F}_q$-linear code $\mathcal{C}$, we denote by $\mathcal{C}(\mathcal{C}, \boldsymbol{\alpha})$ the $\mathbb{F}_{q^m}$-linear code generated by the set $\psi_{\boldsymbol{\alpha}}(\mathcal{C})$.

Our distinguisher works as follow :

- it takes as input $M \in \mathbb{F}_2^{(n-k)\times n}$ and $s \in \mathbb{F}_2^{n-k}$.
- it chooses a vector $\boldsymbol{\alpha} \in \mathbb{F}_{2^m}^n$ at random until the coordinates of $\boldsymbol{\alpha}$ are $\mathbb{F}_2$-linearly independent.
- it gives to D_R the input $(\psi_{\boldsymbol{\alpha}}(M), s)$.
- it returns the same value as D_R.

If (M, s) is an output of the Fisher-Stern generator, then there exists an $\boldsymbol{x}$ such that $s = M\boldsymbol{x}^T$ and $w_H(\boldsymbol{x}) = d$. Hence $s = \psi_{\boldsymbol{\alpha}}(M)\psi_{\boldsymbol{\beta}}(\boldsymbol{x})^T$ with $\boldsymbol{\beta} = \boldsymbol{\alpha}^{-1} = (\alpha_1^{-1}, \ldots, \alpha_n^{-1})$.

Let $\mathcal{C}$ be the code of parity-check matrix M. Since $\mathcal{C}$ is a random code, its Hamming minimum distance d is on the Gilbert-Varshamov bound, so $d \approx d_{GV}$.

Note that $w_H(\psi_\beta(\boldsymbol{x})) = d$. According to Theorem 8 of [15], if we choose $m > 8n$, the probability that the rank minimum distance d_R of $\mathcal{C}(\mathcal{C}, \boldsymbol{\alpha})$ is different from d decreases exponentially with n. According to Lemma 7 of [15], the rank weight of $\psi_\beta(\boldsymbol{x})$ satisfies $w_R(\psi_\beta(\boldsymbol{x})) = w_H(\boldsymbol{x}) = d$. This implies that the distinguisher D_R accepts (M, s) with a non-negligible advantage.

If (M, s) is purely random, D_R sees only a random distribution and accepts the inputs with probability $1/2$.

Thus the existence of a distinguisher for our generator implies the existence of a distinguisher for the Fisher-Stern generator, which contradicts Theorem 2 of [12]. This implies that our generator is pseudo-random.

6 Quantum attacks

In this section we evaluate the complexity of solving the rank (metric) syndrome decoding problem with a quantum computer. We will use for that a slight generalization of Grover's quantum search algorithm [16,18] given in [7] what we will use in the following form. We will use the NAND circuit model as in [4], which consists in a directed acyclic graph where each node has two incoming edges and computes the NAND of its predecessors.

Theorem 1. *[7] Let f be a Boolean function $f : \{0,1\}^b \to \{0,1\}$ that is computable by a NAND circuit of size S. Let p be the proportion of roots of the Boolean function*

$$p \stackrel{def}{=} \frac{\#\{x \in \{0,1\}^b : f(x) = 0\}}{2^b}.$$

Then there is a quantum algorithm based on iterating a quantum circuit $\mathcal{O}\big(\frac{1}{\sqrt{p}}\big)$ *many times that outputs with probability at least* $\frac{1}{2}$ *one of the roots of the Boolean function. The size of this circuit is* $\mathcal{O}\big(S\big)$.

Basically this tool gives a quadratic speed-up when compared to a classical algorithm. Contrarily to what happens for the Hamming metric [4], where using this tool does not yield a quadratic speed-up over the best classical decoding algorithms, the situation is here much clearer : we can divide the exponential complexity of the best algorithms by two. The point is that the algorithms of [14] and [20] can be viewed as looking for a linear subspace which has the right property, where linear spaces with appropriate parameters are drawn uniformly at random and this property can be checked in polynomial time. The exponential complexity of these algorithms is basically given by $\mathcal{O}\big(\frac{1}{p}\big)$ where p is the fraction of linear spaces that have this property. More precisley we have

$$\frac{1}{p} = \mathcal{O}\big(q^{(w-1)(k+1)}\big)$$

for $m > n$, see [20]) and

$$\frac{1}{p} = \mathcal{O}\big(q^{(w-1)\lfloor\frac{(k+1)m}{n}\rfloor}\big)$$

when $m \leq n$, see [14]. Checking whether the linear space has the right property can be done by
(i) solving a linear system with $(n-k-1)m$ equations and with about as many unknowns over $\mathbb{F}_q$,
(ii) checking whether a matrix over $\mathbb{F}_q$ of size $r \times r'$ is of rank equal to w where $(r, r') = (m - \lceil\frac{(k+1)m}{n}\rceil, n)$ in the case $m \leq n$ and $(r, r') = (n-k-1, m)$ in the case $m > n$.

If we view q as a fixed quantity, there is a classical NAND circuit of size $\mathcal{O}\big((n-k)^3m^3\big)$ that realizes these operations. In other words, by using Theorem 1 we obtain

Proposition 1. *For fixed* q*, there is a quantum circuit with* $\mathcal{O}\big((n-k)^3m^3\big)$ *gates that solves the rank metric syndrome decoding problem in time* $\mathcal{O}\big((n-k)^3m^3\big)q^{(w-1)(k+1)/2}$ *when* $m > n$ *and in time* $\mathcal{O}\big((n-k)^3m^3q^{(w-1)\left\lceil\frac{(k+1)m}{n}\right\rceil/2}\big)$ *when* $m \leq n$.

7 Performances and examples of parameters

7.1 Asymptotic behaviour

Consider the case of a random parity check matrix without any structure, $n = m$, $q = 2$ and the case where w is on the rank Gilbert-Varsahmov bound (which is equal in this case to $n(1 - \sqrt{k/n})$). In that case the cost of a syndrome computation is in $\mathcal{O}(n^3)$ operations in the base field $\mathbb{F}_2$ and the number of random bits that are obtained is in $\mathcal{O}(n^2)$, hence the number of operations in the base field $\mathbb{F}_2$ per bit is $\mathcal{O}(n)$. Now since the complexity of the best attacks is in $2^{\mathcal{O}(n^2)}$, for a given security parameter λ, we obtain $n = \mathcal{O}(\sqrt{\lambda})$ and the cost of the protocol is $\mathcal{O}(\sqrt{\lambda})$ per bit, when for other Hamming based approach the cost is in $\mathcal{O}(\lambda)$ in the case of random parity check matrices without additional structure.

Concerning the amount of data, there is also an asymptotic improvement when compared to the Hamming metric. The data for this protocol is given by the matrix $\boldsymbol{H}$ of the code. If it is written in systematic form, $\boldsymbol{H}$ is described by its $k(n-k)$ coefficients in $\mathbb{F}_{2^n}$, so $k(n-k)n$ bits, hence the size of the data N is in $\mathcal{O}(n^3) = \mathcal{O}(\lambda^{3/2})$ whereas the size of the data is in $\mathcal{O}(\lambda^2)$ for the Hamming metric with random matrices [11].

7.2 Parameters

We propose two sets of parameters. In the first table, we optimize the size of the required data, that is to say the matrix of the code and the initialization vector. These parameters are useful in constrained environments where the lack of memory is the main issue. The drawback of small data size is that the performance of the generator is very low.

n	n-k	$d_{GV}(n,k)$	w	security	data size	key size	cycles/bytes
31	18	11	10	128	7646 bits	128	273
41	25	16	12	192	17048 bits	192	144
47	30	19	15	256	24899 bits	256	183
61	39	25	23	512	54103 bits	512	977

In the second table, we optimize the speed of the generator at the cost of data size (but still with small matrix sizes compared to SYND and XSYND). We obtain speeds comparable to those of SYND [13] and we are less efficient than XSYND [26]. But we can always increase the parameters to improve the speed of the generator. Moreover our parameters are chosen at random and have no cyclic structure. In practice as for SYND and

XSYND our results are slower than optimized versions of AES in counter mode, but at the price of increasing the size of the matrix it is possible to do better: the parameters we propose in this second table show how it is possible to optimize the speed at the cost of a larger matrix size. Note that the parameters we propose here are only examples and that it should be possible to have faster speed.

n	n-k	$d_{GV}(n,k)$	w	security	data size	key size	cycles/bytes
43	36	24	14	128	14038 bits	128	48
61	50	35	17	192	35143 bits	192	51
83	73	54	25	256	63859 bits	256	51
127	115	87	42	512	183652 bits	512	76

All the lengths n are prime, although there is no evidence that a specific attack against composite length would exist. As quantum attacks divide the exponent of the complexity of the attack by two, our two last parameters are quantum resilient.

We made a non-optimized implementation of our scheme with the MPFQ library, which showed that the theoretical complexity we give, fitted with what we obtained in practice. Moreover the main operation of our system, the syndrome computation (a matrix-vector product) is highly parallelizable, which can be used to further improve the performances.

8 Conclusion

In this paper we give the first PRNG based on rank metric. The security if system relies on the hardness of solving general instances of the RSD problem, which permits to obtain small size of keys without considering additional structure like cyclicity or quasi-cyclicity. We give results and parameters which show that our system is a good trade-off between speed and data size when compared to other code-based PRNG in a context of PRNG provably as secure as known difficult problems. We also study the improvement of the complexity of the best known combinatorial attacks a quantum computer may bring. We give parameters both resistant to the best known classical and quantum attacks.

Acknowledgment

Jean-Pierre Tillich acknowledges the support of the Commission of the European Communities through the Horizon 2020 program under project number 645622 PQCRYPTO.

References

1. Abhishek Banerjee, Chris Peikert, and Alon Rosen. Pseudorandom functions and lattices. In *Advances in Cryptology–EUROCRYPT 2012*, pages 719–737. Springer, 2012.
2. Côme Berbain, Henri Gilbert, and Jacques Patarin. Quad: A practical stream cipher with provable security. In *Advances in Cryptology-EUROCRYPT 2006*, pages 109–128. Springer, 2006.
3. Elwyn Berlekamp, Robert McEliece, and Henk van Tilborg. On the inherent intractability of certain coding problems. *IEEE Trans. Inform. Theory*, 24(3):384–386, May 1978.
4. Daniel J. Bernstein. Grover vs. McEliece. In Nicolas Sendrier, editor, *Post-Quantum Cryptography 2010*, volume 6061 of *Lecture Notes in Comput. Sci.*, pages 73–80. Springer, 2010.
5. Lenore Blum, Manuel Blum, and Mike Shub. A simple unpredictable pseudo-random number generator. *SIAM Journal on computing*, 15(2):364–383, 1986.
6. Manuel Blum and Silvio Micali. How to generate cryptographically strong sequences of pseudorandom bits. *SIAM journal on Computing*, 13(4):850–864, 1984.
7. M. Boyer, G. Brassard, P. Høyer, and A. Tapp. Tight bounds on quantum searching. *Fortsch. Phys.*, 46:493, 1998.
8. Florent Chabaud and Jacques Stern. The cryptographic security of the syndrome decoding problem for rank distance codes. In *Advances in Cryptology - ASIACRYPT 1996*, volume 1163 of *Lecture Notes in Comput. Sci.*, pages 368–381, Kyongju, Korea, November 1996. Springer.
9. Nicolas Courtois. Efficient zero-knowledge authentication based on a linear algebra problem MinRank. In *Advances in Cryptology - ASIACRYPT 2001*, volume 2248 of *Lecture Notes in Comput. Sci.*, pages 402–421, Gold Coast, Australia, 2001. Springer.
10. Ronald Cramer, Léo Ducas, Chris Peikert, and Oded Regev. Recovering short generators of principal ideals in cyclotomic rings. Cryptology ePrint Archive, Report 2015/313, 2015. http://eprint.iacr.org/.
11. A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In A.M. Odlyzko, editor, *Advances in Cryptology - CRYPTO '86*, volume 263 of *Lecture Notes in Comput. Sci.*, pages 186–194. Springer, 1987.
12. Jean-Bernard Fischer and Jacques Stern. An efficient pseudo-random generator provably as secure as syndrome decoding. In Ueli Maurer, editor, *Advances in Cryptology - EUROCRYPT'96*, volume 1070 of *Lecture Notes in Comput. Sci.*, pages 245–255. Springer, 1996.
13. Philippe Gaborit, Cédric Lauradoux, and Nicolas Sendrier. SYND: a fast code-based stream cipher with a security reduction. In *Proc. IEEE Int. Symposium Inf. Theory - ISIT*, pages 186–190, Nice, France, June 2007.
14. Philippe Gaborit, Olivier Ruatta, and Julien Schrek. On the complexity of the rank syndrome decoding problem. *CoRR*, abs/1301.1026, 2013.
15. Philippe Gaborit and Gilles Zémor. On the hardness of the decoding and the minimum distance problems for rank codes. *CoRR*, abs/1404.3482, 2014.
16. Keith Gibson. The security of the Gabidulin public key cryptosystem. In Ueli Maurer, editor, *Advances in Cryptology - EUROCRYPT '96*, volume 1070 of *Lecture Notes in Comput. Sci.*, pages 212–223. Springer, 1996.

17. Louis Goubin and Nicolas Courtois. Cryptanalysis of the TTM cryptosystem. In Tatsuaki Okamoto, editor, *Advances in Cryptology - ASIACRYPT 2000*, volume 1976 of *Lecture Notes in Comput. Sci.*, pages 44–57. Springer, 2000.
18. L. K. Grover. Quantum mechanics helps in searching for a needle in a haystack. *Phys. Rev. Lett.*, 79:325, 1997.
19. Johan Håstad, Russell Impagliazzo, Leonid A Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999.
20. Adrien Hauteville and Jean-Pierre Tillich. New algorithms for decoding in the rank metric and an attack on the LRPC cryptosystem, 2015. abs/1504.05431.
21. Aviad Kipnis and Adi Shamir. Cryptanalysis of the HFE public key cryptosystem by relinearization. In *Advances in Cryptology - CRYPTO'99*, volume 1666 of *Lecture Notes in Comput. Sci.*, pages 19–30, Santa Barbara, California, USA, August 1999. Springer.
22. Leonid A Levin. One way functions and pseudorandom generators. *Combinatorica*, 7(4):357–363, 1987.
23. Françoise Lévy-dit Vehel and Ludovic Perret. Algebraic decoding of codes in rank metric. In *proceedings of YACC06*, Porquerolles, France, June 2006. available on `http://grim.univ-tln.fr/YACC06/abstracts-yacc06.pdf`.
24. Rudolf Lidl and Harald Niederreiter. *Finite fields*, volume 20 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, Cambridge, second edition, 1997. With a foreword by P. M. Cohn.
25. Robert J. McEliece. *A Public-Key System Based on Algebraic Coding Theory*, pages 114–116. Jet Propulsion Lab, 1978. DSN Progress Report 44.
26. Mohammed Meziani, Gerhard Hoffmann, and Pierre-Louis Cayrel. Improving the Performance of the SYND Stream Cipher. In *Progress in Cryptology - AFRICACRYPT 2012*, volume 7374 of *Lecture Notes in Comput. Sci.*, pages 99–116. Springer, 2012.
27. Alexei V. Ourivski and Thomas Johansson. New technique for decoding codes in the rank metric and its cryptography applications. *Problems of Information Transmission*, 38(3):237–246, 2002.
28. Pierre-Jean Spaenlenhauer. *Résolution de systèmes multi-homogènes et determinantiels*. PhD thesis, Univ. Pierre et Marie Curie- Paris 6, October 2012.
29. Andrew C Yao. Theory and application of trapdoor functions. In *Foundations of Computer Science, 1982. SFCS'08. 23rd Annual Symposium on*, pages 80–91. IEEE, 1982.