A new structural attack for GPT and variants
(preprint)

R. Overbeck

GK Electronic Commerce,
TU-Darmstadt,
Department of Computer Science,
Cryptography and Computer Algebra Group.
overbeck@cdc.informatik.tu-darmstadt.de

Abstract. In this paper we look at the Gabidulin version of the McEliece cryptosystem (GPT) and its variants. We propose a new polynomial time attack on the private key, which is applicable to all variants proposed so far, breaking some of them completely.

Keywords: GPT, Gabidulin codes, code based cryptography, public key cryptography.

1 Introduction

The security of cryptosystems based on error correcting codes is connected to the hardness of the general decoding problem. In 1991 Gabidulin, Paramonov and Tretjakov proposed a variant of the McEliece scheme (GPT) [4] using *rank distance* codes instead of hamming distance codes. Smaller public-key sizes have been proposed for GPT than for the original McEliece cryptosystem, as general decoding algorithms are much slower for for the rank metric than for the hamming-metric.

Gibson developed two structural attacks for the GPT cryptosystem (see e.g. [3] and [5]) and proved the parameter sets proposed in [4] and [3] to be insecure. A drawback of Gibson's attacks is, that they are very slow if $t \geq k$ or if the secret key was carefully chosen in the case that $t < k$. Several variants of GPT have been proposed in order to avoid Gibson's attacks (see e.g. [1] and [7]).

In this paper we build a structural attack on the Niederreiter variant of GPT [1] for popular parameter sets. This new attack may be extended to a polynomial time attack on the GPT cryptosystem and all other variants proposed so far.

The paper is structured as follows: First we give a short introduction to rank distance codes and the Niederreiter variant of the GPT cryptosystem. Then we attack this variant and finally extend our attack to the original GPT cryptosystem.

2 Rank distance codes

Rank distance codes were presented by Gabidulin in 1985 They are linear codes over the finite field $\mathbb{F}_{q^m}$ for q (power of a) prime and $m \in \mathbb{N}$. As their name suggests they use a special concept of distance.

Definition 1. *Let $x = (x_1, \cdots, x_n) \in \mathbb{F}_{q^m}^n$ and $b_1, \cdots, b_m$ a basis of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$. We can write $x_i = \sum_{j=1}^m x_{ij} b_j$ for each $i = 1, \cdots, n$ with $x_{ij} \in \mathbb{F}_q$. The rank norm $\| \cdot \|_r$ is defined as follows:*

$$\|x\|_r := \operatorname{rank}\left((x_{ij})_{1 \le i \le n,\ 1 \le j \le m} \right) .$$

The rank norm of a vector $x \in \mathbb{F}_{q^m}^n$ is uniquely determined (independent of the choice of basis) and induces a metric, called *rank distance.*

Definition 2. *An (n, k)-code $\mathcal{C}$ over a finite field $\mathbb{F}$ is a k-dimensional subvectorspace of the vector space $\mathbb{F}^n$. We call the code $\mathcal{C}$ an (n, k, d) rank distance code if $d = \min_{x,y \in \mathcal{C}} \|x - y\|_r$. The matrix $C \in \mathbb{F}^{k \times n}$ is a generator matrix for the (n, k) code $\mathcal{C}$ over $\mathbb{F}$, if the rows of C span $\mathcal{C}$ over $\mathbb{F}$. The matrix $H \in \mathbb{F}^{n \times (n-k)}$ is called check matrix for the code $\mathcal{C}$ if it is the right kernel of C. The code generated by $H^\top$ is called dual code of $\mathcal{C}$ and denoted by $\mathcal{C}^\perp$.*

In [6] Ourivski and Johansson presented an algorithm which solves the general decoding problem in $\mathcal{O}\left((m\frac{d-1}{2})^3 q^{(d-3)(k+1)/2}\right)$ operations over $\mathbb{F}_q$ for (n, k, d) rank distance codes over $\mathbb{F}_{q^m}$. A special class of rank distance codes are the *Gabidulin codes* for which an efficient decoding algorithm exists [3]. We will define these codes by their generator matrix.

Definition 3. *Let $g \in \mathbb{F}_{q^m}^n$ be a vector s.t. the components g_i, $i = 1, \cdots, n$ are linearly independent over $\mathbb{F}_q$. This implies that $n \le m$. The (n, k, d) Gabidulin code $\mathcal{G}$ is the rank distance code with generator matrix*

$$G = \begin{pmatrix} g_1 & g_2 & \cdots & g_n \\ g_1^q & g_2^q & \cdots & g_n^q \\ \vdots & & \ddots & \vdots \\ g_1^{q^{k-1}} & g_2^{q^{k-1}} & \cdots & g_n^{q^{k-1}} \end{pmatrix} \in \mathbb{F}_{q^m}^{k \times n}. \tag{1}$$

An (n, k) Gabidulin code $\mathcal{G}$ corrects $\left\lfloor \frac{n-k}{2} \right\rfloor$ errors and has a minimum distance of $d = n - k + 1$. The dual code of an (n, k) Gabidulin code is a $(n, n - k)$ Gabidulin code (see [3]). The vector g is said to be the *generator vector* of the Gabidulin code $\mathcal{G}$. A decoding algorithm based on the “right Euclidian division algorithm” runs in $\mathcal{O}\left(d \log_2^2 d + dn\right)$ operations over $\mathbb{F}_{q^m}$ for (n, k, d) Gabidulin codes [3].

Throughout this paper we will use the following notation. We write $\mathcal{G} = \langle G \rangle$ if the linear (n, k)-code $\mathcal{G}$ over the field $\mathbb{F}$ has the generator matrix G. We will identify $x \in \mathbb{F}^n$ with $(x_1, \cdots, x_n), x_i \in \mathbb{F}$ for $i = 1, \cdots, n$. For any (ordered) subset $\{j_1, \cdots j_m\} = J \subseteq \{1, \cdots n\}$ we denote the vector $(x_{j_1}, \cdots, x_{j_m}) \in \mathbb{F}^m$ with x_J. Similary, we denote by $M_{\cdot J}$ the submatrix of a $k \times n$ matrix M consisting of the columns corresponding to the indizes of J and $M_{J' \cdot} = \left(\left(M^\top\right)_{\cdot J'}\right)^\top$ for any (ordered) subset J' of $\{1, \cdots, k\}$. Block matrices will be given in brackets.

3 The Niederreiter variant of GPT

In this section, we briefly introduce the Niederreiter variant of the GPT cryptosystem presented in [1].

- **System Parameters:** $q, n \leq m$, k, l, where $l < k$.
- **Key Generation:** First generate the following matrices over $\mathbb{F}_{q^m}$:

  G: $k \times n$ generator matrix of an (n, k, d) Gabidulin code $\mathcal{G}$ over $\mathbb{F}_{q^m}$.
  S: $(n - k + l) \times (n - k + l)$ random non-singular matrix
  A: $l \times n$ random matrix with rank l over $\mathbb{F}_{q^m}$ and rank n over $\mathbb{F}_q$.
  Then, compute $e = \frac{n-k}{2}$ and the $k \times n$ matrix

  $$(H')^\top = S \begin{bmatrix} G^\perp \\ A \end{bmatrix} .$$

  Further let $\mathcal{D}_\mathcal{G}$ be an efficient syndrome decoding algorithm for $\mathcal{G}$.
- **Public Key:** (H', e)
- **Private Key:** $(\mathcal{D}_\mathcal{G}, S, A)$ or (G, S, A) where G is of the form in (1).
- **Encryption:** To encode a plaintext $x \in \mathbb{F}_{q^m}^n$ of rank norm less then e, compute the ciphertext c as follows:

  $$c = xH' .$$

- **Decryption:** To decode a ciphertext c apply the decoding algorithm $\mathcal{D}_\mathcal{G}$ for $\mathcal{G}$ to the syndrome build from the first $n - k$ columns of $s = S^{-1}c^\top$.

In all examples and figures we will choose $n = m$ and $q = 2$. Figure 3.1 shows public key sizes and approximate workfactors (operations over $\mathbb{F}_q$) for the fastest general decoding attack. Parameters were taken from [2]. Note that the matrix H' describes a $\mathbb{F}_{q^m}$-linear subcode of $\mathcal{G}$, i.e. the intersection of $\mathcal{G}$ with the left kernel of $A^\top$.

Parameters			Size Public	WF general
m	k	l	Key (Bytes)	decoding
25	17	4	487	2^{71}
25	15	5	469	2^{82}
32	24	4	960	2^{93}

Fig. 3.1. Parameter sets for the Niederreiter GPT

4 Attacking the Niederreiter variant of GPT

In this section we show how to break the Niederreiter version of GPT with the aid of either one of Gibson's attacks or of an attack on GPT without distortion matrix. To attack this GPT variant we don't consider the dual code of the code described by the public key (as it was done in [2]), but the code itself.

Theorem 1. *Let H' be a public check matrix of an instance of the Niederreiter variant of GPT with parameters q, m, n, k and l, where the private matrix S was generated at random with no more conditions than being non-singular. If $k-l>1$ and $n-k-2 \geq \lceil l/(k-l-1) \rceil$, then we may recover the private key corresponding to H' in $\mathcal{O}\left(n^3\right)$ operations over $\mathbb{F}_{q^m}$ with high probability.*

Proof. Let G be a generator matrix of an (n,k) Gabidulin code over $\mathbb{F}_{q^m}$ with $n \leq m$ and generating vector $g_1, \cdots, g_n$. Let $\bar{S} \in \mathbb{F}_{q^m}^{(k-l)\times k}$ be a matrix of full rank over $\mathbb{F}_{q^m}$. Then $\left((H')^\top = (G')^\perp, e = (n-k)/2\right)$ with $G' = \bar{S}G$ is a public key of an instance of the Niederreiter variant of the GPT cryptosystem.

Given the public key we choose $f \in \mathbb{N}$ with $n-k-2 \geq f \geq \lceil l/(k-l-1) \rceil$. (For the parameter sets proposed e.g. in [2], the choice of $f=1$ will be sufficient.) Let $\bar{G}$ be the generator matrix of an $(n, k+f)$ Gabidulin code over $\mathbb{F}_{q^m}$ with generator vector $g_1, \cdots, g_n$. Then we have

$$G' = \left[\bar{S}|0|\cdots|0\right]\bar{G}.$$

For a matrix M let $M^{[j]}$ denote the result of rising every element of M to the power of j. It is easy to see that

$$\dot{G} := \begin{bmatrix} G' \\ (G')^{[q]} \\ \vdots \\ (G')^{\left[q^f\right]} \end{bmatrix} \text{ may be written as } \begin{bmatrix} \bar{S} & 0 & \cdots & 0 \\ 0 & \bar{S}^{[q]} & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \bar{S}^{\left[q^f\right]} \end{bmatrix} \bar{G}. \qquad (2)$$

Further, $\dot{G}$ is a generator matrix of $\langle \bar{G} \rangle$ with high probability. Employing e.g. the algorithm described in [5], we may recover the generator vector g and thus $G^\perp$ in $\mathcal{O}\left((k+f)^3\right)$ Operations over $\mathbb{F}_{q^m}$.

We still have to recover S and A from H' and $G^\perp$. In order to do so, we guess a set L_1 of l rows of H', s.t. $S_{L_1L_2}$ with $L_2 = \{n-k+1, \cdots, n-k+l\}$ is invertible. For a random guess, $S_{L_1L_2}$ is invertible with high probability, if S was generated at random with no more conditions than being non-singular. We may assume that $L_1 = L_2$ and thus

$$(H')^\top = \left[\begin{array}{c} S' \begin{bmatrix} G^\perp \\ A \end{bmatrix} \\ \hline A \end{array}\right]$$

for some $S' \in \mathbb{F}_{q^m}^{(n-k)\times(n-k+l)}$. Knowing A, $G^\perp$ and H' we can solve this system of $(n-k)\cdot n$ equations with $(n-k)\cdot(n-k+l)$ variables, to recover S'. This may be done in $\mathcal{O}\left((n-k)^3\right)$ operations.

Figure 4.2 shows modified parameter sets for which the presented attack does not work. Parameters should be chosen, taking into account the attack proposed in [2]. Further, it is evident that $k - l$ should not be too small.

Parameters			Size Public	WF general
m	k	l	Key (Bytes)	decoding
32	24	20	448	2^{93}
64	52	47	2360	2^{288}

Fig. 4.2. Modified parameter sets for the Niederreiter GPT

5 Attacking the original GPT cryptosystem

In this section we want to show, how to extend our attack to the GPT cryptosystem. This extension will also be applicable to "GPT with column scrambler" and the variant presented in [7].

5.1 The case where $t \ll k$

The public generator matrix of an instance of the GPT cryptosystem may be described as

$$G' = S \left[X \middle| G \right] T \in \mathbb{F}_{q^m}^{k \times n}$$

for a special $k \times t$ matrix X, S non-singular, G generator matrix of an $(n-t, k)$ Gabidulin code and T an non-singular matrix over $\mathbb{F}_q$ [7]. The code generated by G' has a check matrix of the form

$$\left(H'\right)^\top = \begin{bmatrix} 0 & G^\perp \\ \mathrm{Id}_t & A \end{bmatrix} \cdot \left(T^{-1}\right)^\top \tag{3}$$

for some matrix A and the t-dimensional identity matrix Id_t. An attacker could guess a set N_1 of $n-t$ rows of H' s.t. $\left(T^{-1}\right)_{N_1 N_2}$ with $N_2 = \{t+1, \cdots, n\}$ is invertible. Because of the special structure of $G^\perp$ we may assume without loss of generality that $\left(T^{-1}\right)_{N_1 N_2}$ is the identity matrix. The matrix $H'_{N_1 \cdot}$ corresponds to an instance of the Niederreiter version of GPT as long as $k - t > 1$, which is fulfilled for most parameter sets proposed (compare [5], [2] and [7]). As $k-t > 1$ implies that $n-k-t-2 > \lceil t/\left(k-t-1\right) \rceil$, the attack described in the previous section may be employed to recover $G^\perp$ and A. This reveals the private key of the GPT cryptosystem.

5.2 Generalization of the attack

The attack described above is limited to parameter sets, where $k - t > 1$. Using the approach described in the section 4 we want to augment k and lower the dimension of the dual code in a preprocessing step. Afterwards we apply our attack to recover the private key.

In order to augment k we take a row of the public generator matrix G' and apply the frobenius automorphism to its entries. Let G'_i be the chosen row and suggest, that $S_{ik} \neq 0$, which is fulfilled with high probability for a random choice. Then for $f \leq n - t - k$ the matrix build from the rows of G' and $(G'_{i\cdot})^{[q^j]}$ for $j = 1, \cdots, f$ is the public generator matrix of an GPT instance with larger dimension of the Gabidulin code. If we can choose f in a way s.t. $k + f - t > 1$, then we can apply our attack on the GPT cryptosystem, to recover G. In a second step, we would be able to compute an alternative private key.

Still, it might be possible to choose the GPT parameter t larger than $n/2$ to avoid the attacks presented so far. (This is not possible for the original GPT cryptosystem, but for some of its variants.) Let $\bar{G}$ be the generator matrix of the $(n, k + f)$ Gabidulin code with the same generator vector as G and $f \leq n-k-t$. Let $\dot{G}$ the matrix build from G' as described in equation (2). Then $\dot{G}$ has a check matrix of the form

$$(H')^{\top} = \begin{bmatrix} 0 & (\bar{G})^{\perp} \\ A_1 & A_2 \end{bmatrix} \cdot \left(T^{-1}\right)^{\top},$$

where A_1 is a $l \times t$ matrix with $l \leq t$. With a little bit of luck (i.e. if $k+f-l > 1$ and $n - t - k - f - 2 \geq \lceil l/\left(k + f - l - 1\right) \rceil$), we now may continue as above and compute G and thus an alternative private key. In almost all experiments we were able to choose f in a way, such that l decreased to 0, especially for popular parameter sets. If $l = 0$, the check matrix reveals enough information to determine the set N_1 mentioned above and to recover an alternatrive column scrambler.

6 Conclusion

The attacks proposed in this paper are far from being deterministic, but succeed with good probability. We conclude that the original GPT cryptosystem from [4] may not be considered to be secure. Hopefully our results may be extended to the GPT cryptosystem using reducible rank codes (compare [7]). However, after several attacks on the GPT cryptosystem and its variants, it seems to be difficult to name secure parameter sets for the GPT cryptosystem.

References

1. T. Berger and P. Loidreau. Security of the niederreiter form of the GPT public-key cryptosystem. In *IEEE International Symposium on Information Theory, Lausanne, Suisse.* IEEE, July 2002.

2. T.P. Berger and P. Loidreau. How to mask the structure of codes for a cryptographic use. to appear.
3. E.M. Gabidulin. On public-key cryptosystems based on linear codes. In *Proc of 4th IMA Conference on Cryptography and Coding 1993*, Codes and Ciphers. IMA Press, 1995.
4. E.M. Gabidulin, A.V. Paramonov, and O.V. Tretjakov. Ideals over a non-commutative ring and their applications to cryptography. In *Proc. Eurocrypt '91*, volume 547 of *LNCS*. Springer Verlag, 1991.
5. K. Gibson. The security of the Gabidulin public key cryptosystem. In *Proc. of Eurocrypt'96*, volume 1070 of *LNCS*, pages 212–223. Springer Verlag, 1996.
6. A.V. Ourivski and T. Johansson. New technique for decoding codes in the rank metric and its cryptography applications. *Problems of Information Transmission*, 38, No. 3:237–246, 2002.
7. R. Overbeck. Extending Gibson's attacks on the GPT cryptosystem. In *Proc. of WCC 2005*, 2005. to appear.