Optimal Distortion Parameter for the GPT Public-Key Cryptosystem

Maximilien Gadouleau
Department of Electrical and Computer Engineering
Lehigh University
Bethlehem, PA 18015
E-mail: maximilien@lehigh.edu

Zhiyuan Yan
Department of Electrical and Computer Engineering
Lehigh University
Bethlehem, PA 18015
E-mail: yan@lehigh.edu

***Abstract*— The Gabidulin-Paramonov-Trejtakov (GPT) public-key cryptosystems, based on Gabidulin codes, seem to have some advantages than McEliece's public-key cryptosystems using Goppa codes. In this paper we define the optimal distortion parameter for the GPT cryptosystem to be the distortion parameter that maximizes the work load of the best (structural or decoding) attack. The work factors achieved by the GPT cryptosystem using the optimal distortion parameter thus give the guaranteed level of security against any attack. We also show that under reasonable assumptions, the optimal distortion parameter always exists and is unique. Furthermore, we propose an algorithm that computes the optimal distortion parameter with low complexity.**

I. Introduction

After the concept of public-key cryptosystems was introduced, McEliece [6] proposed a public-key system based on error-correcting codes. Even though it is not as popular as, for example, number-theory oriented cryptosystems, it has several advantages. First, its security is based on the problem of general decoding of linear codes, which has been proved to be a NP-hard problem. Secondly, the encryption and decryption schemes are based on binary matrix operations, which implies faster operations and smaller computational costs. Finally, McEliece's cryptosystem using Goppa codes has resisted cryptanalysis for more than 25 years, and therefore remains an important alternative to the number-theory based cryptosystems.

Gabidulin, Paramonov, and Trejtakov [2] proposed a new public-key cryptosystem which uses a new class of codes proposed by Gabidulin [3] based on the rank metric. The new cryptosystem, referred to as the GPT cryptosystem henceforth, uses the rank metric instead of the Hamming metric. In addition to sharing the all the advantages of McEliece's cryptosystem mentioned above, the GPT cryptosystem has two extra advantages: The GPT cryptosystem seems to be much more robust against the decoding attack than McEliece's cryptosystem; the key size of the GPT cryptosystem is much smaller than that of McEliece's cryptosystem.

As for the McEliece's cryptosystem, there are two types of attacks against the GPT cryptosystem: structural and decoding. The best structural and decoding attacks of which we are aware were proposed by Gibson [4] and by Chabaud and Stern [1] respectively. The robustness of the GPT scheme against structural attacks improves when the *distortion parameter*, denoted as t_1, increases. In contrast, the work factor of the decoding attack decreases with t_1. Hence, the value of t_1 should be carefully chosen.

In this paper, we define the *optimal distortion parameter* to be the distortion parameter that maximizes the work load of the best (structural or decoding) attack. Hence, the optimal distortion parameter maximizes the work factor against either kind of attack. Under reasonable assumptions, we further show that the optimal distortion parameter always exists, is unique, and has some other useful properties. Using these properties, we propose an algorithm that computes the optimal distortion parameter with very low complexity with respect to the work factors of the attacks proposed in [4] and [1], the best structural and decoding attack of which we are aware. We remark that although our algorithm depend on the work factors given in [4] and [1], we conjecture that the approach used herein to design the algorithm can be extended to other work factors.

The rest of the paper is organized as follows. In Section II, we give a brief review of the concepts of rank metric and Gabidulin codes. Section II also summarizes the GPT cryptosystem and the work factors of the attacks. In Section III, we define the optimal distortion parameter and illustrates how to compute it. Finally, some concluding remarks are given in Section IV.

II. Preliminaries

In this section, we give a brief review of the main concepts of rank metric and Gabidulin codes. We then summarizes how the GPT cryptosystem works and give the work factors of the attacks against the GPT system.

A. Rank metric

Consider a n-dimensional vector space over the finite field $GF(q^m)$. Let us call $\gamma_1, \gamma_2, \ldots, \gamma_m$ a basis of $GF(q^m)$ over $GF(q)$. Let $\mathbf{a} = (a_1, a_2, \ldots, a_n)$ be a vector of length n over $GF(q^m)$. Note that every element a_j of $GF(q^m)$ can be written as $a_j = \sum_{i=1}^{m} a_{ij}\gamma_i$, where the coefficients $a_{ij} \in GF(q)$ for $i = 1, 2, \ldots, m$. Hence, a_j can be expanded to an m-dimensional column vector $(a_{1j}, a_{2j}, \ldots, a_{mj})^T$ with respect to the basis $\gamma_1, \gamma_2, \ldots, \gamma_m$. Let $\mathbf{A}$ be the matrix with m rows and n columns obtained by expanding all the coordinates

of $\mathbf{a}$ as described above of the development of the coordinates a_i of $\mathbf{a}$. $\mathbf{A}$ is given by:

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix}$$

where $a_j = \sum_{i=1}^{m} a_{ij}\gamma_i$. We can then define the rank norm and rank metric over $GF(q^m)^n$ based on this expansion as follows:

Definition 1 (Rank norm): The *rank norm* of the word $\mathbf{a}$, denoted as $rk(\mathbf{a})$, is defined to be the rank of the matrix $\mathbf{A}$, i.e. $rk(\mathbf{a}) \stackrel{\text{def}}{=} rank(\mathbf{A})$.

Definition 2 (Rank metric): Let $\mathbf{a}$ and $\mathbf{b}$ be two words of length n over $GF(q^m)$, d_{rk} defined by $d_{rk}(\mathbf{a}, \mathbf{b}) \stackrel{\text{def}}{=} rk(\mathbf{a}-\mathbf{b})$ is a distance over $GF(q^m)^n$. It is called the *rank distance* and defines *the rank metric*.

Hence, the *minimum rank distance of a code* d_{rk} is simply the minimum rank distance over all possible pairs of distinct codewords. It can be shown that $d_{\text{rk}} \leq d_{\text{H}}$, the minimum Hamming distance of the same code. The minimum rank distance is also bounded by the following bound.

Theorem 1 (Singleton bound for rank distance codes): The minimum rank distance d_{rk} of a code must satisfy

$$d_{\text{rk}} \leq \frac{m}{n}(n-k)+1. \tag{1}$$

We notice, that, for $n \leq m$, the bound given by the (1) is less tight than the classic Singleton bound $d_{\text{H}} \leq n-k+1$. In contrast, the Singleton bound for rank distance codes is tighter when $n > m$.

B. Gabidulin codes

To simplify the notations, we denote q^i as $[i]$ henceforth.

Definition 3 (Gabidulin codes): Let $\mathbf{g} = (g_1, g_2, \dots, g_n)$ be elements of $GF(q^m)$ linearly independent over $GF(q)$. This implies that $n \leq m$. Then the code which has the following generator matrix

$$\mathbf{G} = \begin{pmatrix} g_1 & g_2 & \dots & g_n \\ g_1^{[1]} & g_2^{[1]} & \dots & g_n^{[1]} \\ \vdots & \vdots & \ddots & \vdots \\ g_1^{[k-1]} & g_2^{[k-1]} & \dots & g_n^{[k-1]} \end{pmatrix}$$

is called *Gabidulin code* $Gab(\mathbf{g}, k)$, of dimension k, generated by $\mathbf{g} = (g_1, g_2, \dots, g_n)$.

Gabidulin codes, as defined above, have minimum rank distance $d_{\text{rk}} = n-k+1$. Hence, they are optimal in the sense that they maximize the minimum rank distance in the case $n \leq m$. Gabidulin [3] also proposed a decoding algorithm in the rank metric for Gabidulin codes. This decoding algorithm is similar to the Berlekamp-Massey algorithm for the Reed-Solomon codes and takes polynomial time. The GPT cryptosystem is based on the existence of this efficient decoding algorithm for the Gabidulin codes.

C. The GPT cryptosystem

The GPT system is quite similar to McEliece's cryptosystem. Instead of using codes which have a decoding algorithm in the Hamming metric, the GPT cryptosystem uses the Gabidulin codes which have an efficient decoding algorithm for the rank metric. Unfortunately, these codes are very structured, so it is necessary to break their structure by using a distortion matrix, which is described below. The following are the main components of the system:

Private key: the triple $(\mathbf{G}, \mathbf{S}, \mathbf{X})$, where

- $\mathbf{G}$ is a $(k \times n)$ generator matrix of a Gabidulin code over $GF(q^m)$, of length $n \leq m$, with correction capability $t = \lfloor (d_{\text{rk}}-1)/2 \rfloor$. $\mathbf{G}$ is given by $\mathbf{G} = (g_j^{[i]})_{i=0,j=1}^{k-1,n}$.
- $\mathbf{S}$ is an invertible $(k \times k)$ random matrix over $GF(q^m)$.
- $\mathbf{X}$ is a $(k \times n)$ matrix over $GF(q^m)$ such that

$$rk(\mathbf{cX}) \leq t_1, \forall\, \mathbf{c} = (c_1, c_2, \dots, c_k) \in GF(q^m)^k,$$

where $t_1 < t$ is called the *distortion parameter*. $\mathbf{X}$ is called the *distortion matrix* with distortion parameter t_1

Public key:

1) $\mathbf{G}' = \mathbf{SG} + \mathbf{X}$.
2) The distortion parameter t_1.

Encryption scheme:

Let $\mathbf{c}$ be a word of length k. The emitter computes and sends the length-n word $\mathbf{y} = \mathbf{cG}' + \mathbf{e}$, where $\mathbf{e}$ is a random error vector with length n and rank less or equal to $(t - t_1)$.

Decryption scheme:

Since $\mathbf{y} = \mathbf{cSG} + (\mathbf{cX} + \mathbf{e})$, the word $(\mathbf{cX} + \mathbf{e})$ has a rank less or equal to t. Using the decoding algorithm of Gabidulin codes, the receiver retrieves the word $\mathbf{c}' = \mathbf{cS}$, and computes $\mathbf{c} = \mathbf{c}'\mathbf{S}^{-1}$.

D. Attacks against the cryptosystem

As for McEliece's cryptosystem, two types of attacks—decoding and structural attacks—against the GPT cryptosystem have been proposed. In [1], a general rank-error decoding algorithm is introduced, and from this we can derive a decoding attack. Its work factor is

$$W_D \sim O\left((n(t-t_1)+m)^3 q^{(m-t+t_1)(t-t_1-1)}\right).$$

In [4], Gibson proposed a structural attack against the GPT cryptosystem that uses some heuristics. Loidreau [5] eliminated all the heuristics and found the following work factor for the same attack:

$$W_S \sim O\left((n-2t)^3 q^{t_1(n-t_1)}\right).$$

III. Optimal Distortion Parameter

A. Definition and Properties

The work factors of the attacks depend on all four parameters—m, n, t_1, and t—of the GPT cryptosystem. m, n, and t depend on the Gabidulin secret matrix $\mathbf{G}$, and are usually given. However, although t_1 determines the robustness against the attacks, how best to choose t_1, to our knowledge, were not described in the literature. Thus, we assume that the degree

of extension m of the field and the block length n and t of the Gabidulin code are fixed, and try to determine the best value for t_1. Our goal is, given the triple (m, n, t), to find the optimal value of t_1, denoted as t_o, such that the system has the most robustness against both kinds of attacks. This means that t_o has to maximize the minimum work factors over all possible attacks. Let us define

$$W_m(t_1) = \min\left(W_D(t_1), W_S(t_1)\right)$$

where $W_D(t_1)$ and $W_S(t_1)$ denote the work factors of the best decoding and structural attacks respectively. It can be shown that t_o must belong to the set $T = \{1, 2, \ldots, t-1\}$. This leads us to the following definition. The *optimal distortion parameter* against the structural and decoding attacks is defined to be

$$t_o = \arg\max_{t_1 \in T} \left(W_m(t_1)\right). \tag{2}$$

Intuitively, if t_1 increases, the structure of the private code is better hidden by the distortion matrix and hence the work factor of any reasonable structural attack increases. In contrast, when t_1 increases, the rank of the error vector $t - t_1$ decreases and a decoding attack will have a smaller work factor. Hence, in the discussions below, we assume that the work factors of the decoding and structural attacks are monotonically decreasing and increasing functions with t_1 respectively. Using this assumption, we can show that

Proposition 1: If $W_S(t_1)$ and $W_D(t_1)$ are increasing and decreasing functions of t_1 respectively, then the optimal distortion parameter always exists and is unique.

It is in general difficult to come up with a closed formula for t_o. Our approach will be to treat t_1 as a continuous variable belonging to the interval $I = [1, t-1]$. With a slight abuse of notation, both $W_D(t_1)$ and $W_S(t_1)$ are now continuous functions on I, and so is $W_m(t_1)$. The uniqueness of the optimal value is still preserved. We can denote the unique value in I that maximizes W_m as t_1^*. That is,

$$t_1^* = \arg\max_{t_1 \in I} \left(W_m(t_1)\right). \tag{3}$$

Using our assumption about the behavior of the work factors of the attacks, we can show that the relation between t_o and t_1^* is given by

Proposition 2: If $W_S(t_1)$ and $W_D(t_1)$ are increasing and decreasing functions of t_1 respectively, then $t_o = \lfloor t_1^* \rfloor$ or $t_o = \lfloor t_1^* \rfloor + 1$.

By Proposition 2, if we can find t_1^*, we only need to evaluate $W_m(\lfloor t_1^* \rfloor)$ and $W_m(\lfloor t_1^* \rfloor + 1)$ and to compare them to determine t_o. More generally, we have

Proposition 3: If $W_S(t_1)$ and $W_D(t_1)$ are increasing and decreasing functions of t_1 respectively, then t_o must be an integer between $\lfloor l \rfloor$ and $\lceil u \rceil$ if $l \leq t_1^* \leq u$.

Thus, we only need to evaluate $W_m(t_1)$ from $\lfloor l \rfloor$ and $\lceil u \rceil$ and pick t_o.

Once t_o is found, we define $\text{maxmin}(m, n, t) \stackrel{\text{def}}{=} W_m(t_o)$. Given a triple (m, n, t), $\text{maxmin}(m, n, t)$ represents the work factor guaranteed by the GPT cryptosystem regardless of the chosen attack.

B. Finding the Optimal Distortion Parameter

In this section, we use the work factors of the attacks in [1] and [4], and propose an algorithm that finds t_o with low complexity. We remark that the attacks in [1] and [4] are the best decoding and structural attacks of which we are aware. However, we believe the approach used below can be extended to other similar attacks as well.

In the sequel, we consider only the dominant term of the work factors of the attacks in [1] and [4]. This simplification should have no effect on t_o in most practical cases. Hence, the work factor of the decoding attack in [1] is given by

$$W_D(t_1) = c_D \left[n(t - t_1) + m\right]^3 \cdot q^{(m-t+t_1)(t-t_1-1)}.$$

Similarly, the work factor of the structural attack in [4] is given by $W_S(t_1) = c_S(n-2t)^3 \cdot q^{t_1(n-t_1)}$. Note that c_D and c_S are both constants.

We will first consider the case where $\frac{c_S}{c_D} = 1$, and then extend our approach to the more general case. It is easy to verify that $W_D(t_1)$ is monotonically decreasing and $W_S(t_1)$ is monotonically increasing. Moreover, one can check that $W_D(1) > W_S(1)$ and $W_D(t-1) < W_S(t-1)$. Hence, there is a unique cross-over point between the two functions. This crossover occurs exactly at t_1^*. It is however unnecessary to find a closed formula for t_1^* since, in the end, we only care about the integers surrounding t_1^*. Due to Proposition 3, it suffices to bound t_1^* tight enough so that only a few choices are left for t_o.

Since t_1^* is given by the cross-over point, we have

$$W_D(t_1^*) = W_S(t_1^*).$$

Now let us assume that the characteristic q of the field is equal to 2. All the logarithms will be taken in base 2 henceforth. Taking $\log_2$ on both sides and assuming that $\log_2\left(\frac{c_S}{c_D}\right) = 0$, we get

$$t_1^*(-m - n + 2t - 1) + f(t_1^*) + mt - m + t - t^2 = 0 \tag{4}$$

where $f(t_1) = 3\log_2\left(\frac{n(t-t_1)+m}{n-2t}\right)$. Rearranging Eq. (4), we have

$$\begin{aligned} t_1^* &= \underbrace{\frac{-t^2 + t(m+1) - m}{m+n+1-2t}}_{\bar{t}_1} + \underbrace{\frac{f(t_1^*)}{m+n+1-2t}}_{g(t_1^*)} \\ &= \bar{t}_1 + g(t_1^*) = \bar{g}(t_1^*) \end{aligned} \tag{5}$$

where $\bar{g}(t_1) \stackrel{\text{def}}{=} \bar{t}_1 + g(t_1)$. In other words, t_1^* is a fixed point of $\bar{g}(t_1)$ and it can be shown that the fixed point is unique.

Now, let us bound the value of t_1^*. We know that g is a positive function over the interval I, so we have

$$\bar{t}_1 \leq t_1^* \leq t - 1.$$

But $\bar{g}(t_1)$ is a decreasing function in t_1, so

$$\bar{g}(t-1) \leq t_1^* \leq \bar{g}(\bar{t}_1). \tag{6}$$

Using the fact that $\bar{g}(t_1)$ is a decreasing function, we can in fact define two series u_i and l_i by applying the function $\bar{g}$ iteratively:

$$\begin{aligned} u_0 &= t-1 \qquad & l_0 &= \bar{t}_1 \\ u_{i+1} &= \bar{g}(l_i) \qquad & l_{i+1} &= \bar{g}(u_i). \end{aligned}$$

It can be shown that the series u_i and l_i give upper and lower bounds of t_1^* respectively, since $l_i \leq t_1^* \leq u_i$ for $i = 0, 1, \cdots$. In fact, it can be shown that the series u_i and l_i both converge to t_1^*. However, we will show in the next subsection that the convergence is not necessary and that the bound given by Equation (6) is sufficient.

C. Speed of convergence

Eventually, we are interested in finding not t_1^*, but t_o. Hence, it is sufficient to get bounds that are tight enough for us to test few integers to determine which one is t_o by Proposition 3. If we can bound, after a iterations, t_1^* between two numbers l_a and u_a which differ by less than 1 but have different floors, then we have 3 possibilities for t_o: $\lfloor l_a \rfloor$, $\lceil l_a \rceil = \lfloor u_a \rfloor$, or $\lceil u_a \rceil$. If the two bounds have the same floor, we only have two possibilities for t_o: $\lfloor l_a \rfloor$ or $\lceil l_a \rceil$. Hence, the iterative procedure can stop when $(u_j - l_j < 1)$.

We are interested in the behavior of the difference between the bounds given by Equation (6), which is

$$\begin{aligned} h(t,n) &\stackrel{\text{def}}{=} \bar{g}(\bar{t}_1) - \bar{g}(t-1) \\ &= \frac{3}{m+n+1-2t} \log_2 \left(\frac{n(t-\bar{t}_1)+m}{n+m} \right). \end{aligned}$$

Proposition 4: The difference between the bounds given by Equation (6) always satisfies the equation

$$\begin{aligned} h(t,n) &\leq h\left(\frac{m}{2}-1, m\right) \\ &\leq \frac{3}{m+3}\left[\log_2(m+8)-3\right]. \end{aligned} \tag{7}$$

Since the upper bound for $h(t,n)$ in Eq. (7) achieves a maximum of 0.2727 at $m = 8$, we have the following

Corollary 1: The bounds given by Equation (6) satisfy the stopping condition.

Hence, the stop condition is met after *only one* iteration. For reasonable values of (m, n, t), the difference between the two bounds after one iteration is often small so that both bounds have the same floor, and hence we only have to check two integers instead of three.

The above discussions are for the case where $\frac{c_S}{c_D} = 1$. This assumption is also justified when the order of the work factor is much more important than the constant coefficient in the work factor. Hence, the distortion parameter obtained under the assumption that $\frac{c_S}{c_D} = 1$ is optimized with respect to the orders of the work factors for the two attacks. When it is necessary to take the constant coefficients in the work factors into account of the optimal distortion parameter, $\frac{c_S}{c_D}$ has to be accounted for. When c_S and c_D are far apart, $\bar{t}_1$ in Eq. (6) is offset by $-\log_2\left(\frac{c_S}{c_D}\right)$. That is,

$$\bar{t}_1 = \frac{-t^2 + t(m+1) - m - \log_2\left(\frac{c_S}{c_D}\right)}{m+n+1-2t}.$$

If $\bar{t}_1 \geq t-1$, it is clear that $t_o = t-1$. Otherwise, we redefine $l_0 = \max\{1, \bar{t}_1\}$, and the algorithm proposed above is still applicable.

D. Example

In Table I, we show the optimal distortion parameter t_o and the corresponding $\text{maxmin}(m,n,t)$ for $(m,n) = (32,26)$ and varying t, obtained by one iteration of our algorithm. We assumed that $\frac{c_S}{c_D} = 1$. We note that in the example, t_o is selected between two possibilities in all the cases.

t	u_0	l_0	u_1	l_1	t_o	$\log_2$(maxmin)
2	1	0.4918	0.5665	0.5505	1	18.0000
3	2	0.9831	1.0793	1.0491	1	45.1013
4	3	1.4737	1.5912	1.5482	2	49.7549
5	4	1.9636	2.1029	2.0477	2	73.3783
6	5	2.4528	2.6149	2.5478	3	79.0000
7	6	2.9412	3.1276	3.0488	3	99.5098
8	7	3.4286	3.6415	3.5510	4	105.9658
9	8	3.9149	4.1573	4.0548	4	123.4221
10	9	4.4000	4.6758	4.5610	5	130.7549
11	10	4.8837	5.1983	5.0706	5	144.9658
12	11	5.3659	5.7268	5.5854	6	153.4221
13	12	5.8462	6.2648	6.1088	6	163.7549
14	13	6.3243	6.8203	6.6486	7	174.0000
15	14	6.8000	7.4173	7.2286	7	178.0000

TABLE I
THE OPTIMAL DISTORTION PARAMETER FOR VARYING t AND $(m,n) = (32,32)$

IV. CONCLUSIONS

In this paper, the optimal distortion parameter is defined to be the one that maximizes the work factor of the best (structural or decoding) attack. That is, the GPT cryptosystem with the optimal distortion parameter achieves the largest work factor regardless the type of attack. We also show that the optimal distortion parameter has useful properties under reasonable assumptions. Finally, We propose an algorithm that finds the optimal parameter with low complexity.

REFERENCES

[1] F. Chabaud, J. Stern, "The cryptographic security of the syndrome decoding problem for rank distance codes", *Advances in Cryptology: ASIACRYPT '96*, volume 1163 of LNCS, pages 368–381. Springer-Verlag, 1996.

[2] E. M. Gabidulin, A. V. Paramonov, O. V. Trejtakov, "Ideals over a non-commutative ring and their application in cryptology". *LNCS*, vol. 573, 1991, pp. 482-489.

[3] E. M. Gabidulin, "Theory of codes with maximum rank distance," Problems on Information Transmission, vol. 21, num. 1, pp. 1-12, January 1985.

[4] J. K. Gibson, "The security of the Gabidulin public-key cryptosystem," in EUROCRYPT'96, pp. 212-223. 1996

[5] P. Loidreau, "Étude et Optimisation de Cryptosystèmes Clé Publique Fondés sur la Théorie des Codes Correcteurs," Ph. D. Dissertation, École Polytechnique, May 2001. In French.

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