Security of the GPT-Type Cryptosystems

Maximilien Gadouleau
Department of Electrical and Computer Engineering
Lehigh University
Bethlehem, PA 18015 USA

Zhiyuan Yan
Department of Electrical and Computer Engineering
Lehigh University
Bethlehem, PA 18015 USA

Abstract— **The Gabidulin-Paramonov-Tretjakov (GPT) public-key cryptosystem and the GPT system with column scrambler, both based on Gabidulin codes, seem to have some advantages over McEliece's public-key cryptosystems using Goppa codes. Since the parameters of the GPT-type systems affect the work factors of attacks in different fashions and determine the effectiveness of these attacks in some cases, we evaluate the security and study the choice of parameters of the GPT-type cryptosystems with respect to some well-known structural and decoding attacks *jointly*. The performances of the GPT system with column scrambler and the GPT system are compared with that of McEliece's system.**

I. Introduction

After the concept of public-key cryptosystems (PKCs) was introduced, McEliece [8] proposed a public-key system based on Goppa codes. Even though it is not as popular as, for example, number-theory based PKCs, it has several advantages. Firstly, its security is based on the problem of general decoding of linear codes, which is an NP-complete problem. Secondly, the encryption and decryption schemes are based on binary matrix operations, which implies faster encryption and decryption 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 PKCs. Two types of attacks — structural and decoding attacks — have been proposed against McEliece's cryptosystem. The former type attempts to recover a private key based on the given public key, while the latter type tries to recover only the plaintext for a given ciphertext. The main disadvantage of McEliece's cryptosystem is its large key size, which is necessary to achieve sufficient security level against decoding attacks.

Gabidulin, Paramonov, and Tretjakov [4] proposed a new public-key cryptosystem which uses a new class of codes proposed by Gabidulin [2] based on the rank metric. This cryptosystem, referred to as the GPT cryptosystem henceforth, is based on the rank metric instead of the Hamming metric. In addition to sharing all the advantages of McEliece's cryptosystem mentioned above, the GPT cryptosystem seems to be much more robust against decoding attacks than McEliece's cryptosystem and hence the key size of the GPT cryptosystem is much smaller than that of McEliece's cryptosystem. The primary threat to the security of the GPT system seems to be structural attacks (see, for example, [6], [7]). To strengthen the security, variants of the GPT cryptosystem were proposed (see, for example, [3], [9]). In [9], the security is improved by first appending some extra columns to the public generator matrix and then mixing the columns with a column scrambler (see [9] for details). This variant of the GPT cryptosystem, referred to as the GPT with column scrambler (GPTCS) PKC henceforth, has a smaller transmission rate than the GPT cryptosystem due to the extra columns.

Both structural ([6], [7], [9], [11]) and decoding attacks [10] have been proposed against the GPT-type PKCs. The attacks introduced in [10] are the most efficient decoding attacks to our knowledge. The structural attacks proposed in [6], [7], [9], [11] are considered in this paper. The work factors of all these attacks depend on the parameters of the GPT-type PKCs. The parameters affect the work factors of these attacks in different fashions [5] and determine whether an attack is effective. For example, it is shown in [9] that when the parameters of the GPTCS system are carefully chosen, the attacks proposed by Gibson are not directly applicable [9]. Also, the work factor of one of Overbeck's attacks is highly sensitive to the quantity $\hat{s}$ [11], which depends on the parameters of the GPTCS system. Thus, for some instances of the GPT-type PKCs, one type of attacks may be very efficient while others may be ineffective. Thus, to study the overall security of the GPT-type PKCs, it is necessary to evaluate all relevant attacks *jointly*. Since the distortion parameter of the GPT cryptosystem affects the work factors of decoding and structural attacks in different manners, the optimal distortion parameter of the GPT cryptosystem with respect to a given set of attacks is defined as the distortion parameter that maximizes the work factor of the best attack of the given set of attacks in [5]. It thus guarantees the highest security level possible for the considered cryptosystem against the given attacks.

In this paper, we evaluate the overall security of the GPT-type PKCs against the attacks in [6], [7], [9]–[11] jointly. Our results suggest that the GPT-type PKCs are secure for reasonable public key sizes and transmission rates, and that they still have smaller key sizes than McEliece's cryptosystem. We also study how to choose the parameters so as to maximize the overall security of the GPT-type PKCs against a set of attacks. This extends the concept of optimal distortion parameter in [5] to other parameters of the GPT-type cryptosystem.

The rest of the paper is organized as follows. Section II gives a brief review of the rank metric, Gabidulin codes, the GPTCS PKC and some attacks against it. In Section III, we evaluate the overall security of the GPT and GPTCS systems with respect

1-4244-0504-1/06/$20.00 ©2006 IEEE

to the given attacks, and compare the performances of the GPT and GPTCS systems with those of McEliece's system. Some concluding remarks are given in Section IV.

II. Preliminaries

In this section, we give a brief review of the main concepts of the rank metric and Gabidulin codes. We then summarize the key points of the GPT and GPTCS systems and review the attacks against the GPT-type PKCs.

A. The rank metric and Gabidulin codes

Consider $\mathbf{a} = (a_1, a_2, \ldots, a_n) \in \mathrm{GF}(q^N)^n$, an n-dimensional vector space over $\mathrm{GF}(q^N)$[1]. Assume $\{\gamma_1, \gamma_2, \ldots, \gamma_N\}$ is a basis set of $\mathrm{GF}(q^N)$ over $\mathrm{GF}(q)$, then a_j can be expanded to an N-dimensional column vector $(a_{1j}, a_{2j}, \ldots, a_{Nj})^T$ with respect to the basis set where $a_j = \sum_{i=1}^{N} a_{ij}\gamma_i$ and $a_{ij} \in \mathrm{GF}(q)$ for $i = 1, 2, \ldots, N$. Let $\mathbf{A}$ denote the $N \times n$ matrix obtained by expanding all the coordinates of $\mathbf{a}$ as described above, i.e.,

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N1} & a_{N2} & \cdots & a_{Nn} \end{pmatrix},$$

where $a_j = \sum_{i=1}^{N} a_{ij}\gamma_i$. The *rank norm* (over $\mathrm{GF}(q)$) of $\mathbf{a}$, denoted as $rk(\mathbf{a}|\mathrm{GF}(q))$, is defined to be the rank of $\mathbf{A}$ over $\mathrm{GF}(q)$, i.e., $rk(\mathbf{a}|\mathrm{GF}(q)) \overset{\text{def}}{=} rank(\mathbf{A})$. Accordingly, $\forall\, \mathbf{a}, \mathbf{b} \in \mathrm{GF}(q^N)^n$, $d_{\text{r}}(\mathbf{a}, \mathbf{b}) \overset{\text{def}}{=} rk(\mathbf{a} - \mathbf{b}|\mathrm{GF}(q))$ is shown to be a metric over $\mathrm{GF}(q^N)^n$, referred to as *the rank metric* henceforth. Hence, the *minimum rank distance* d_{r} of a code is simply the minimum rank distance over all possible pairs of distinct codewords. Clearly, a code with a minimum rank distance d_{r} can correct errors with rank up to $t = \lfloor (d_{\text{r}} - 1)/2 \rfloor$. It can be shown that $d_{\text{r}} \le d_{\text{H}}$, the minimum Hamming distance of the same code. Due to the Singleton bound of linear block codes, d_{r} of an (n, k) linear block code over $\mathrm{GF}(q^N)$ thus satisfies

$$d_{\text{r}} \le n - k + 1. \tag{1}$$

When $n \le N$, the bound in (1) is achieved with equality by the family of Gabidulin codes [2]. To simplify notations, we denote q^i as $[i]$ henceforth. Let $\{g_0, g_1, \ldots, g_{n-1}\}$ be n elements ($n \le N$) of $\mathrm{GF}(q^N)$ that are linearly independent over $\mathrm{GF}(q)$. The code generated by

$$\mathbf{G} = \begin{pmatrix} g_0 & g_1 & \cdots & g_{n-1} \\ g_0^{[1]} & g_1^{[1]} & \cdots & g_{n-1}^{[1]} \\ \vdots & \vdots & \ddots & \vdots \\ g_0^{[k-1]} & g_1^{[k-1]} & \cdots & g_{n-1}^{[k-1]} \end{pmatrix}$$

is called a *Gabidulin code* generated by $\mathbf{g} = (g_0, g_1, \ldots, g_{n-1})$. This code is shown to be an (n, k) code over $\mathrm{GF}(q^N)$ with $d_{\text{r}} = n - k + 1$. Gabidulin [2] also proposed a polynomial-time decoding algorithm in the rank metric for Gabidulin codes, which is similar to the Berlekamp-Massey algorithm for Reed-Solomon codes.

B. The GPTCS cryptosystem

The GPTCS cryptosystem uses Gabidulin codes which have an efficient decoding algorithm in the rank metric. Since Gabidulin codes are very structured, it is necessary to break their structure by using a distortion matrix. In addition, some extra columns are appended to the generator matrix of the secret code, and the resulting generate matrix is then scrambled using a column scrambler.

The *private key* of the GPTCS system is $(\mathbf{G}, \mathbf{S}, \mathbf{X}_1, \mathbf{X}_2, \mathbf{P})$, where

- $\mathbf{G}$ is a $k \times n$ generator matrix of a Gabidulin code of length $n \le N$ over $\mathrm{GF}(q^N)$, with error-correction capability $t = \lfloor (d_{\text{r}} - 1)/2 \rfloor = \lfloor (n - k)/2 \rfloor$,
- $\mathbf{S}$ is an invertible $k \times k$ random matrix over $\mathrm{GF}(q^N)$,
- $\mathbf{X}_1$ is a $k \times m$ matrix over $\mathrm{GF}(q^N)$,
- $\mathbf{X}_2$ is a $k \times n$ matrix over $\mathrm{GF}(q^N)$ with $rk(\mathbf{X}_2|\mathrm{GF}(q)) = t_1$ and $rk(\mathbf{X}_2|\mathrm{GF}(q^N)) = s$,
- $\mathbf{P}$ is an invertible $(n+m) \times (n+m)$ random matrix over $\mathrm{GF}(q)$.

Note that $\mathbf{S}$ and $\mathbf{P}$ are called the row and column scramblers respectively. Details about how to properly choose $\mathbf{X}_1$ and $\mathbf{X}_2$ are given in [9]. The *public key* is the distortion parameter t_1 and the $k \times (m + n)$ matrix $\mathbf{G}' = \mathbf{S}([\mathbf{O} \,|\, \mathbf{G}] + [\mathbf{X}_1 \,|\, \mathbf{X}_2])\mathbf{P}$, where $\mathbf{O}$ is a $k \times m$ matrix of zeros.

Let $\mathbf{c}$ be a plaintext of length k, and its corresponding ciphertext is $\mathbf{y} = \mathbf{c}\mathbf{G}' + \mathbf{e}$ with length $n + m$, where $\mathbf{e}$ is a random error vector with length $n + m$ and rank $\le (t - t_1)$. The decryption of the ciphertext $\mathbf{y}$ proceeds as follows. The receiver computes $\mathbf{y}\mathbf{P}^{-1} = [\mathbf{c}\mathbf{S}\mathbf{X}_1 \,|\, \mathbf{c}\mathbf{S}\mathbf{G} + \mathbf{c}\mathbf{S}\mathbf{X}_2] + \mathbf{e}\mathbf{P}^{-1}$. Note the sum of $\mathbf{c}\mathbf{S}\mathbf{X}_2$ and the last n columns of $\mathbf{e}\mathbf{P}^{-1}$ has a rank $\le t$. Using the decoding algorithm of Gabidulin codes for the last n columns of $\mathbf{y}\mathbf{P}^{-1}$, the receiver first retrieves $\mathbf{c}' = \mathbf{c}\mathbf{S}$, and then computes $\mathbf{c} = \mathbf{c}'\mathbf{S}^{-1}$.

The GPT cryptosystem in [3] can be viewed as a special case of the GPTCS system with $m = 0$ and $\mathbf{P}$ being the identity matrix.

C. Attacks against the GPT and GPTCS systems

Both decoding and structural attacks have been proposed against the GPT and GPTCS systems. The two decoding attacks introduced in [10] are applicable to both the GPT and GPTCS systems. Since these two attacks are the most efficient decoding attacks to our knowledge, they are the only decoding attacks considered in this paper. For the GPTCS PKC, the work factors for the two attacks are on the order $W_{D1}(N, n, m, t, t_1) \sim O(N^3(t - t_1)^3 q^{(n+m-2t+1)(t-t_1-1)})$ and $W_{D2}(N, n, m, t, t_1) \sim O((n + m - t - t_1)^3(t - t_1)^3 q^{(t-t_1-1)(N-t+t_1)})$, respectively.

Four different structural attacks against the GPT and GPTCS systems are considered in this paper. The structural attacks in [6] and [7] are proposed for the GPT cryptosystem, but can be applied to the GPTCS system by selecting n columns of the public generator matrix. In this paper, we use the work factors

[1] The notations used in this paper are mostly consistent with those in [9].

given in [9] for the two attacks in [6] and [7] respectively. The work factor of the former attack, which is efficient only when s is small, is given by $W_{G1}(N,n,s) \sim O(n^3 q^{sN})$. The latter attack is more efficient when s takes greater values [9] and has a work factor $W_{G2}(k,t_1,f) \sim O(k^3 + (k+t_1+2)fq^{f(k+2)})$, where $s \leq t_1$ and $f = \max\{0, t_1 - 2s\}$. Note that the expected value of f is $\max\{0, t_1 - s\}$ under some conditions [9], but we use $\max\{0, t_1 - 2s\}$ for f, which gives smaller work factors. The two structural attacks in [9], [11] respectively are proposed for the GPTCS system and they are not considered for the GPT system. In order to simplify the notations, let us introduce some induced parameters for the GPTCS system: $r = \left\lfloor \frac{n+m}{t_1+m} \right\rfloor$, $v = \left\lfloor \frac{m}{r} \right\rfloor$, $u = 2t_1 - 2t + m - v + 1$. We remark that when $u > 0$, Gibson's attacks are ineffective against the GPTCS system and the attack in [9] can be used instead and has a work factor of $W_{OG}(N,n,u,t) \sim O(t^3 q^{u(N+n-u)})$. When $u \leq 0$, the attack in [9] should not be considered. In [11, Theorem 6.2], it is shown that the GPTCS system with parameters N, n, m, k, t_1, and s is equivalent to an instance of the GPT PKC with parameters $\hat{n} = n + m$, $\hat{N} = \lceil (n+m)/N \rceil \cdot N$, $\hat{k} = k$, $\hat{t}_1 = t_1 + m$, $\hat{s} \leq \min\{k, s+m\}$. Thus, both Gibson's attacks can be extended to the GPTCS system. In most cases, only the extension of the second Gibson's attack leads to an efficient attack. This attack, referred to as Overbeck's attack henceforth, has a work factor of $W_{G2}(\hat{k}, \hat{t}_1, \hat{f})$, where $\hat{f} \overset{\text{def}}{=} \max\{0, \hat{t}_1 - 2\hat{s}, \hat{t}_1 + 1 - k\}$ [11]. Note that the work factor of Overbeck's attack depends on the exact value of $\hat{s}$, which depends on the instance of the GPTCS system.

III. Security of the GPT and GPTCS PKCs

A. Optimal parameters

The work factors of the attacks against the GPT and GPTCS PKCs depend on their parameters: N, n, m, t, t_1, s, and possibly induced parameters. These parameters affect the work factors of various attacks in different fashions [5]. For instance, the work factors for decoding attacks usually *decrease* with t_1, whereas the work factors for structural attacks *increase* with t_1. Furthermore, the effectiveness of these attacks mentioned above depends on the choice of parameters. For example, the work factor of the attack in [11] is highly sensitive to the quantity $\hat{f}$, which depends on the parameters of the GPTCS system: Overbeck's attack has only polynomial complexity when $\hat{f} = 0$, but is ineffective if $\hat{f}$ takes a small value such as 3 or 4. Thus, it is not enough to consider the attacks *individually*. Instead, the attacks must be considered *jointly* in order to determine the security of the GPT or GPTCS system. This joint evaluation will also offer insights about how to choose the parameters for the GPT or GPTCS systems so that it is secure against *all* the attacks considered. Let P denote the set of parameters of an instance of the GPT (or GPTCS) system, then the work factor of an attack a is given by $W_a(P)$. Given a set $\mathcal{A}$ of potential attacks for the PKC, the security level of the instance determined by P with respect to the set of attacks, denoted by $W(\mathcal{A}, P)$, is given by $W(\mathcal{A}, P) = \min_{a \in \mathcal{A}} W_a(P)$. Let $\mathcal{P}$ denote the space of permissible parameter sets according to specific requirements on aspects (block length, public key size, transmission rate, etc.) of the PKC other than its security level, then the *optimal work factor* of the GPT (or GPTCS) system with respect to $\mathcal{A}$, denoted by $W(\mathcal{A}, \mathcal{P})$, is given by

$$W(\mathcal{A}, \mathcal{P}) = \max_{P \in \mathcal{P}} W(\mathcal{A}, P). \tag{2}$$

Note that $W(\mathcal{A}, \mathcal{P})$ represents a guaranteed level of security against all the attacks in $\mathcal{A}$. Given a required security level against the attacks in $\mathcal{A}$, the parameters can be optimized with respect to other criteria using $W(\mathcal{A}, P)$. Among the parameters of the GPTCS (or GPT) system, N, n, m, and t affect not only the security level but also other aspects such as public key size and transmission rate, whereas t_1 and s affect only the security levels. Hence, N, n, m, and t may be predetermined and in this case the maximization in (2) is mainly with respect to the parameters t_1 and s. That is,

$$W(\mathcal{A}, \mathcal{P}) = \max_{t_1, s} \left[\min_{a \in \mathcal{A}} W_a(P) \right]. \tag{3}$$

This is an extension of our work in [5], where the work factors of one structural and one decoding attacks are treated as functions of t_1 and the above approach is used to optimize the distortion parameter t_1.

For the GPT cryptosystem, $\mathcal{A}$ consists of the two decoding attacks introduced in [10] and the two structural attacks in [6], [7]. For the GPTCS cryptosystem, $\mathcal{A}$ consists of the two decoding attacks introduced in [10] and the structural attacks in [6], [7], [9], [11].

B. Choices of parameters

We would like to comment on the impact of various attacks on the choice of parameters. First, note that the exponent in $W_{D1}(N,n,m,t,t_1)$, $(t - t_1 - 1)(n+m-2t+1) = -2t^2 + (n+m+2t_1+3)t - (n+m+1)(t_1+1)$, is a quadratic expression in t with maximum reached when $t = \frac{n+m+2t_1+3}{4}$. It is interesting to observe that the security of the GPTCS (GPT) system actually decreases with t when $t > \frac{n+m+2t_1+3}{4}$. Also, observe that the decoding attacks in [10] have small work factors when t_1 is close to t. Indeed, if $t = t_1 + 1$, then $W_{D1}(N,n,m,t,t_1)$ and $W_{D2}(N,n,m,t,t_1)$ are polynomial; furthermore, if $t = t_1 + 2$, then $W_{D1}(N,n,m,t,t_1) = (2N)^3 q^{n+m-2t+1}$, which typically results in a low security level. To ensure security against the decoding attacks in [10], it seems that t should satisfy

$$t \geq t_1 + 3. \tag{4}$$

Note that Gibson's attacks [6], [7] are applicable to the GPTCS system when $u = 2t_1 + m - v + 1 - 2t > 0$. Using (4), we have

$$u \leq m - v - 5. \tag{5}$$

The right hand side of (5) is positive only if $m \geq 6$ since v is nonnegative. Therefore, to ensure that Gibson's attacks are not applicable to the GPTCS system, in practice we need $m \geq 6$. Thus, to thwart Gibson's attacks against the GPTCS system, it is necessary to either ensure $u > 0$ or select large parameters so that Gibson's attacks have large work factors even if applicable. Furthermore, if Gibson's attacks are applicable and $s = 1$, then the work factor $W_{G1}(N, n, s)$ is roughly $n^3 q^N$, which is small unless both N and n are large. Thus if Gibson's attacks are applicable, $s \geq 2$ is necessary in order to achieve a sufficient security level.

The advantage of the GPTCS system is that parameters can be chosen so that $u > 0$, rendering Gibson's attacks not directly applicable. However, Overbeck's attack in [11] first translates the GPTCS system into a GPT system and then applies Gibson's second attack. Thus Overbeck's attack is applicable to the GPTCS system regardless the value of u. The impact of this attack is complicated since its work factor depends on $\hat{s}$, whose value depends on the instance of the GPTCS system. In this subsection and Section III-C, we assume that $\hat{s}$ always takes its upper bound, i.e., $\hat{s} = \min\{k, s+m\}$. Since both the parameter $\hat{f}$ and hence the work factor of Overbeck's attack decreases with $\hat{s}$, our assumption results in a lower bound of the work factor for Overbeck's attack against the GPTCS system. In Section III-D, we suggest an approach to evaluate the value of $\hat{s}$ for a given instance of the GPTCS system.

We shall not consider the case where $k < s+m$, as it leads to a low transmission rate. Thus, $\hat{s} = s + m$. The work factor $W_{G2}(\hat{k}, \hat{t}_1, \hat{f})$ of Gibson's second attack strongly depends on $\hat{f} = \max\{0, \hat{t}_1 + 1 - k, \hat{t}_1 - 2\hat{s}\}$. Recall that $\hat{f} \geq 3$ is needed to achieve a sufficient security level. It is easy to see that $\hat{f} \geq 3$ implies either $\hat{t}_1 - k + 1 \geq 3$, which implies a low transmission rate, or

$$t_1 \geq m + 2s + 3. \quad (6)$$

Combining (6) with $t \geq t_1 + 3$ and $s \geq 2$, we obtain $t \geq m + 10$. Clearly, this implies that m must take low values in order to guarantee a satisfactory security level.

C. Overall security of the GPT and GPTCS cryptosystems

Assuming $\hat{s} = \min\{k, s+m\}$, Figure 1 shows the optimal work factors in (3) for the GPTCS system with $N = n = 46$ and variable m and t. We observe that the optimal work factor of the GPTCS decreases with m for reasonable values of t in Figure 1. This is because in this case $\hat{s} = s + m$ and the parameter $\hat{f}$ and the work factor $W_{G2}(\hat{k}, \hat{t}_1, \hat{f})$ both decrease with m. Recall that the GPT system is merely the GPTCS system with $m = 0$. Thus, if $\hat{s} = s + m$, the GPTCS system is not as secure as the GPT system.

For the GPT system, on the other hand, we do not assume that s takes its largest value. Hence the parameter f can be chosen to be positive, and the work factor of the attack in [7] remains large. The optimal work factors in (3) for the GPT system with $N = 46$ and variable n and t are shown in Figure 2.

Thus, when $\hat{s} = s + m$ Overbeck's attack significantly weakens the security of the GPTCS system and longer codes are necessary to ensure sufficient security level.

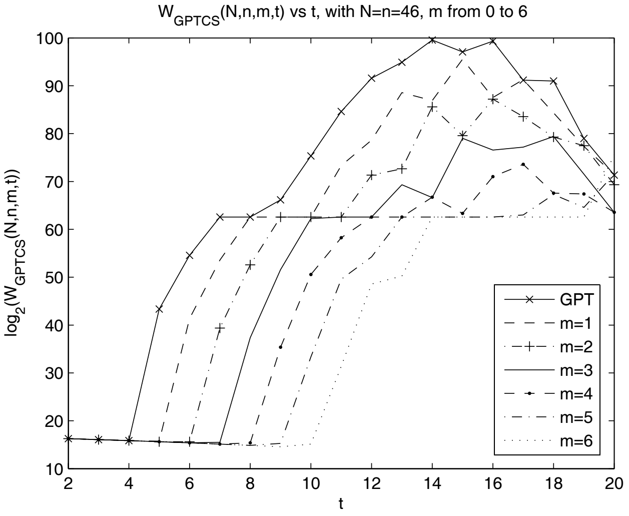

Fig. 1. The optimal work factor $W_{\text{GPTCS}}(N, n, m, t)$ as a function of t, $n = N = 46$, and m varies from 0 to 6

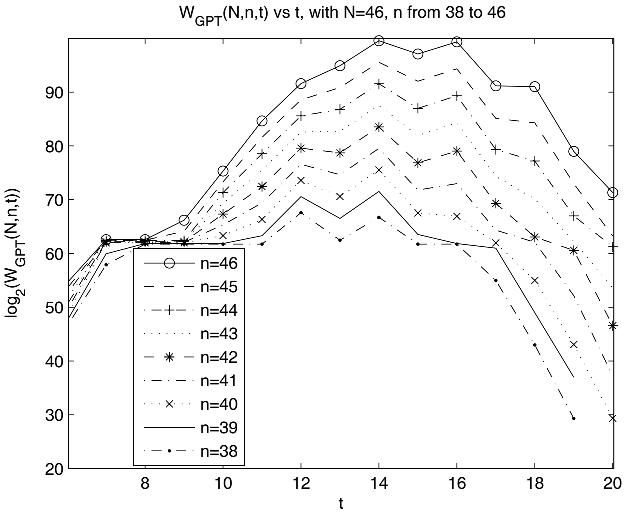

Fig. 2. The optimal work factor $W_{\text{GPT}}(N, n, t)$ as a function of t, $N = 46$, and n varies from 38 to 46

D. Evaluation of the $\hat{s}$ parameter

Above, we evaluated the security of the GPTCS and GPT systems using the worst-case scenario: $\hat{s} = s + m$. Under this assumption, Overbeck's attack considerably damages the security of the GPTCS system. The issue now is how to evaluate the security of an instance of the GPTCS system against Overbeck's attack if $\hat{s}$ for the given instance of the GPTCS system does not equal $s + m$.

Suppose $\hat{s}$ can take any value between 1 and $s + m$, we can still determine whether the instance of the GPTCS system is secure or not without knowing the exact value of $\hat{s}$. Note that the possible values for $\hat{s}$ fall into two sets, one leading to very large work factors for Overbeck's attack while the other leading to small work factors. Take the parameter set introduced in [9] for example. If $P_a = \{n = N = 32, m = 8, t = 6, t_1 = 3, \text{ and } s \in \{1, 2, 3\}\}$ is chosen, the work factor of Overbeck's attack is either $\geq 2^{72}$ for $\hat{s} \leq 4$ (regardless the value of s) or $\leq 2^{28}$ for $\hat{s} > 4$. A higher transmission rate can be achieved using another parameter set, $P_b = \{n = N = 32, m = 8, t = 4, t_1 = 1, s = 1\}$. For this parameter set, the work factor of Overbeck's attack is either $\geq 2^{84}$ for $\hat{s} \leq 3$ or $\leq 2^{32}$ for $\hat{s} > 3$. This wide gap in security makes it easy to test whether the instance of the GPTCS system is secure against Overbeck's attack. Given any instance of the GPTCS system, when Overbeck's attack is applied, two scenarios may occur: either the instance of the GPTCS system is cracked rather quickly, or the attack fails after a long time. If the latter occurs, $\hat{s}$ for the given instance probably falls into the first set above and leads to a large work factor for Overbeck's attack.

E. Performance comparison

Now we can compare the transmission rates, public key sizes, and optimal work factors of the GPT and GPTCS systems with those of McEliece's system. The most efficient attack against McEliece's system is that in [1] to our knowledge. Setting $n = 1024$ and using the optimal parameter $t = 41$ and $k = 614$ for McEliece's system given in [1], the parameter set leads to a public key size of roughly $nk = 614$ kbits and a work factor of 2^{66} [1]. For the GPT system, we choose the parameter set $P_c = \{N = n = 46, t = 9, t_1 = 6, s = 2\}$ so that it has roughly the same transmission rate 0.6 as the instance of McEliece's system. For the GPTCS system, the parameter set P_a, which leads to a transmission rate of 0.5, was suggested in [9]. To achieve the same rate as the parameters for McEliece's and the GPT systems, we shall use the parameter set P_b instead. We also assume the instance satisfies $\hat{s} \leq 3$.

The performances of the three instances of McEliece's system, the GPT, and the GPTCS system respectively are compared in Table I. With the same transmission rate and approximately the same security level, the public key size of the GPTCS system is smaller than that of the GPT system, both significantly smaller than that of McEliece's system. However, the advantage is smaller than suggested in [4]. The advantage of the GPTCS system over the GPT system is that if t_1 is small and $u > 0$, then both the decoding attacks in [10] and Gibson's attacks are thwarted. In this case, the structural attack in [9] will have large work factors as well. The parameters in both P_a and P_b illustrate this advantage.

IV. Conclusion and future work

Considering the attacks proposed in [6], [7], [9]–[11] jointly, the overall security of the GPT and GPTCS cryptosystems have been evaluated and the choices of parameters have been investigated. The parameter sets obtained in this study result in guaranteed security levels for the GPT and GPTCS systems against the given attacks. Under some assumptions, the attacks in [11] significantly weaken the security of the GPTCS system. Otherwise, the attacks do not affect the overall security of the GPTCS significantly. Finally, the GPT and GPTCS systems still outperform McEliece's system, but the advantage in key size is much smaller than suggested in [4].

System	Parameters						rate	key size	WF
	N	n	m	t	t_1	s			
McEliece	-	1024	-	41	-	-	0.6	614	2^{66}
GPT	46	46	0	9	6	2	0.6	57.8	2^{66}
GPTCS	32	32	8	4	1	1	0.6	30	2^{69}

TABLE I

The proposed sets of parameter values for McEliece's, the GPT, and the GPTCS PKCs with their according transmission rates, public key sizes (in Kbits), and security levels

Another structural attack against the GPTCS system was recently proposed in [12], and the new attack was claimed to have a probabilistic polynomial complexity. That is, with some probability, the attack in [12] succeeds with only polynomial complexity. Clearly, this new attack impacts the security of the GPTCS and GPT systems and the choice of their parameters. How to thwart this new attack is the topic of our future work.

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 Trans. Inform. Theory*, vol. 44, no. 1, pp. 367–378, January 1998.

[2] E.M. Gabidulin, "Theory of codes with maximum rank distance," *Problems on Information Transmission*, vol. 21, no. 1, pp. 1–12, January 1985.

[3] E.M. Gabidulin, "On public-key cryptosystems based on linear codes," *Proceedings of the 4th IMA Conference on Cryptography and Coding 1993*, Codes and Ciphers, pp.17–31, 1995.

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

[5] M. Gadouleau and Z. Yan, "Optimal distortion parameter for the GPT public-key cryptosystem," *Proceedings of the 2005 IEEE Sarnoff Symposium*, pp. 130–133, Princeton NJ, April 2005.

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

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

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

[9] A.V. Ourivski and E.M. Gabidulin, "Column scrambler for the GPT cryptosystem," *Discrete Applied Mathematics*, vol. 128, pp. 207–221, 2003.

[10] A.V. Ourivski and T. Johansson, "New technique for decoding codes in the rank metric and its cryptography applications," *Problems on Information Transmission*, vol. 38, no. 3, pp. 237–246, 2002.

[11] R. Overbeck, "Extending Gibson's attacks on the GPT cryptosystem," *Proceedings of the 2005 International Workshop on Coding and Cryptography*, Bergen Norway, March 2005.

[12] R. Overbeck, "A new structural attack for GPT and variants," *Proceedings of Mycrypt 2005*, LNCS, vol. 3715, pp. 50–63, 2005.