

Cryptanalysis of a System Based on Twisted Reed–Solomon Codes

Julien Lavauzelle* Julian Renner†

March 23, 2020

Abstract

Twisted Reed–Solomon (TRS) codes are a family of codes that contains a large number of maximum distance separable codes that are non-equivalent to Reed–Solomon codes. TRS codes were recently proposed as an alternative to Goppa codes for the McEliece code-based cryptosystem, resulting in a potential reduction of key sizes. The use of TRS codes in the McEliece cryptosystem has been motivated by the fact that a large subfamily of TRS codes is resilient to a direct use of known algebraic key-recovery methods.

In this paper, an efficient key-recovery attack on the TRS variant that was used in the McEliece cryptosystem is presented. The algorithm exploits a new approach based on recovering the structure of a well-chosen subfield subcode of the public code. It is proved that the attack always succeeds and breaks the system for all practical parameters in $O(n^4)$ field operations. A software implementation of the algorithm retrieves a valid private key from the public key within a few minutes, for parameters claiming a security level of 128 bits. The success of the attack also indicates that, contrary to common beliefs, subfield subcodes of the public code need to be precisely analyzed when proposing a McEliece-type code-based cryptosystem. Finally, the paper discusses an attempt to repair the scheme and a modification of the attack aiming at Gabidulin–Paramonov–Tretjakov cryptosystems based on twisted Gabidulin codes.

1 Introduction

In the last years, cryptosystems relying on the hardness of decoding in a generic code have gained a lot of attention due to their potential resistance against quantum computer attacks. The first code-based cryptosystem was proposed by McEliece already in 1978 [McE78]. Its hardness is based on the assumption that a random generator matrix of a random binary Goppa code is hard to distinguish from the generator matrix of a random code. To this day, the principle behind the McEliece system still plays a significant role in the design of code-based cryptography. In particular, four out of the six code-based proposals in round 2 of the National Institute of Standards and Technology (NIST) post-quantum cryptography standardization process are based on McEliece’s principle.

Compared to other post-quantum-secure public-key encryption schemes, *e.g.* some lattice-based cryptosystems, the main drawback of the McEliece cryptosystem lies in the size of its public key. To overcome this drawback, other families of codes have been proposed to replace Goppa codes, but most of them can be subjected to algebraic attacks.

*Université de Rennes, CNRS, IRMAR – UMR 6625, France. Email: julien.lavauzelle@univ-rennes1.fr

†Institute for Communications Engineering, Technical University of Munich (TUM), Germany. Email: julian.renner@tum.de

For instance, generalized Reed–Solomon (GRS) codes were proposed in 1986 by Niederreiter [Nie86], but Sidelnikov and Shestakov mounted a very efficient attack to recover an alternative secret key [SS92]. Wieschebrink proved that also random subcodes of GRS codes — proposed in [BL05] — cannot be used due to their vulnerability to the *code squaring* attack [Wie10]. Further instances and cryptanalyses of algebraic code-based schemes can be found in [Sid94, MS07, BCGO09, FOP⁺16, JM96, CCP17]. One should emphasize that many recent attacks are largely based on previously known methods. For example, some instances of the RLCE scheme [Wan16] were broken by Couvreur, Lequesne and Tillich by a sophisticated analysis of the squares of *puncturings and shortenings* of the public code [CLT19].

One recent alternative class of codes for the McEliece cryptosystem emerged from twisted Reed–Solomon (TRS) codes [BPR17]. In particular, Beelen *et al.* analyzed the structural properties of a very specific subfamily of TRS codes [BBPR18]. They proved that this subfamily is disjoint from the class of GRS codes; thus the attack by Sidelnikov and Shestakov [SS92] cannot be applied to their system. Further, they showed that shortenings of these codes up to two positions have maximal Schur square dimension [Puc18], meaning that the proposed system is impervious to a direct application of the attack presented by Couvreur *et al.* in [CGGU⁺14]. Additionally, the authors gave evidence that their system is not vulnerable to straight-forward applications of methods introduced by Wieschebrink in [Wie06, Wie10].

The intention of the authors of [BBPR18] was to exploit the optimal error-correction capability of TRS codes to reduce the length of the public code, and accordingly the size of the public key. In [BBPR18], an explicit subfamily of TRS codes was proposed, providing a reduction of the public key up to a factor of 7.4 compared to binary Goppa codes, for a claimed security level of 128 bits.

In this paper, we present an efficient key-recovery attack on this cryptosystem based on TRS codes. As analyzed by the authors of [BBPR18], the direct application of previously known structural attacks does not work. Instead, we recover the structure of a well-chosen *subfield subcode* \mathcal{S} of the public TRS code \mathcal{T} . We give a characterization of the structure of this subfield subcode, as a subspace of low codimension contained in a classical Reed–Solomon code \mathcal{R} . We then prove that the Wieschebrink squaring method *always* succeeds when applied to the subfield subcode \mathcal{S} , and this enables us to retrieve an algebraic description of \mathcal{R} . By analyzing equivalent representations of TRS codes, we finally deduce an algebraic description of the public code \mathcal{T} . The application of the squaring method to the subfield subcode is a non-trivial modification of Wieschebrink’s attack.

To the best of our knowledge, our attack is the first of its kind to exploit structural weaknesses of *subfield subcodes* of the public code. On the contrary, the restriction to a subfield is usually considered as an operation that breaks the structure of an algebraic code and therefore makes it suitable for cryptography as attested by the attack-resilience of Goppa codes despite being subfield subcodes of Reed–Solomon codes. Our approach of attacking the subfield subcode instead of the original code might also be applicable to other classes of codes used in code-based cryptography.

We show that for all practical parameters proposed by the designers, our algorithm recovers a valid private key from the public key in $O(n^4)$ operations over the underlying field, where n denotes the code length. The attack is implemented in the computer-algebra system SageMath [The19] and is made public. Although the implementation is not optimized, it determines a valid private key in approximately two minutes for the parameters proposed in [BBPR18].

The paper is structured as follows. In Section 2, we introduce the notation, and state the definition as well as important structural properties of TRS codes. In Section 3, we present the key generation, encryption and decryption algorithm and the parameters proposed in [BBPR18]. In Section 4, we derive a structural attack on the scheme and we precisely analyze its complexity. Additionally, in Section 5, we discuss a potential fix of the cryptosystem, as well as an extension of the attack to the rank metric setting [PRW18]. Conclusions are given in Section 6.

2 Preliminaries

2.1 Notation

Let \mathbb{F}_q denote the finite field of order q , where q is a prime power. Vectors in \mathbb{F}_q^n are row vectors, and we use $\mathbb{F}_q^{m \times n}$ to represent the set of $m \times n$ matrices over \mathbb{F}_q . For $i \in \{1, \dots, m\}$ and $j \in \{1, \dots, n\}$, the (i, j) -th entry of $\mathbf{A} \in \mathbb{F}_q^{m \times n}$ is denoted by $A_{i,j}$. The set of invertible matrices of size m over \mathbb{F}_q is denoted by $\text{GL}_m(\mathbb{F}_q)$.

Let us fix a finite field extension \mathbb{F}/\mathbb{F}_q . The \mathbb{F} -vector space generated by a subset $S \subset \mathbb{F}_q^n$ is denoted by $\text{Span}_{\mathbb{F}}(S)$. By convention, we also represent the \mathbb{F} -vector space spanned by the rows of $\mathbf{A} \in \mathbb{F}_q^{m \times n}$ by $\text{Span}_{\mathbb{F}}(\mathbf{A})$.

A linear code $\mathcal{C} \subseteq \mathbb{F}_q^n$ with parameters $[n, k, d]$ is an \mathbb{F}_q -vector space of \mathbb{F}_q^n of dimension k , where d is the minimum Hamming weight $w_{\text{H}}(\mathbf{c}) := |\{i \in \{1, \dots, n\}, c_i \neq 0\}|$ of a non-zero codeword $\mathbf{c} \in \mathcal{C}$. A generator matrix of \mathcal{C} is a matrix $\mathbf{G} \in \mathbb{F}_q^{k \times n}$ such that $\mathcal{C} = \text{Span}_{\mathbb{F}_q}(\mathbf{G})$.

Given $\mathbf{a} \in \mathbb{F}_q^n$ and $\mathbf{b} \in \mathbb{F}_q^n$, their componentwise product is defined as $\mathbf{a} \star \mathbf{b} := (a_1 b_1, \dots, a_n b_n) \in \mathbb{F}_q^n$. Further, we define the Schur-square (or Hadamard-square) of a linear code $\mathcal{C} \subseteq \mathbb{F}_q^n$ as

$$\mathcal{C}^{(\star 2)} := \text{Span}_{\mathbb{F}_q}(\{\mathbf{a} \star \mathbf{b} : \mathbf{a}, \mathbf{b} \in \mathcal{C}\}).$$

Let $\mathbb{F}_q[X]$ denote the set of univariate polynomials over \mathbb{F}_q . For a fixed evaluation vector $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_n) \in \mathbb{F}_q^n$, we define the evaluation map

$$\begin{aligned} \text{ev}_{\boldsymbol{\alpha}} : \mathbb{F}_q[X] &\rightarrow \mathbb{F}_q^n \\ f &\mapsto (f(\alpha_1), f(\alpha_2), \dots, f(\alpha_n)). \end{aligned}$$

Finally, if $\mathcal{I}, \mathcal{J} \subset \mathbb{N}$ are two finite subsets of integers, then we define their sumset

$$\mathcal{I} + \mathcal{J} := \{a + b : a \in \mathcal{I}, b \in \mathcal{J}\} \subseteq \mathbb{N}.$$

2.2 Twisted Reed–Solomon codes

Before introducing TRS codes, let us first recall the definition of (classical) Reed–Solomon codes.

Definition 1 (Reed–Solomon code). *Let the entries of $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_n) \in \mathbb{F}_q^n$ be pairwise distinct, and fix $1 \leq k \leq n$. The Reed–Solomon (RS) code of length n and dimension k is defined by*

$$\text{RS}_{k,n}[\boldsymbol{\alpha}] := \{\text{ev}_{\boldsymbol{\alpha}}(f) : f \in \mathbb{F}_q[X], \deg f \leq k - 1\} \subseteq \mathbb{F}_q^n.$$

The entries of $\boldsymbol{\alpha}$ are called locators of the Reed–Solomon code $\text{RS}_{k,n}[\boldsymbol{\alpha}]$.

RS codes are maximum distance separable (MDS) codes, i.e., they reach the Singleton bound $d \leq n - k + 1$. They also admit the use of efficient decoding algorithms for an error of weight up to the unique decoding radius $\lfloor \frac{n-k}{2} \rfloor$.

TRS codes were recently constructed as a generalization of RS codes [BPR17]. Let us first define a specific subspace of polynomials. Let $\ell \geq 1$, and $n \geq k \geq 1$. Given a vector $\mathbf{h} \in \{0, \dots, k-1\}^\ell$ of pairwise distinct increasing *hooks*, a vector $\mathbf{t} \in \{1, \dots, n-k\}^\ell$ of pairwise distinct *twists*, and a vector of field coefficients $\boldsymbol{\eta} \in (\mathbb{F}_q \setminus \{0\})^\ell$, the set of $[\mathbf{t}, \mathbf{h}, \boldsymbol{\eta}]$ -twisted polynomials is

$$\mathcal{P}_{k,n}[\mathbf{t}, \mathbf{h}, \boldsymbol{\eta}] := \left\{ \sum_{i=0}^{k-1} f_i X^i + \sum_{j=1}^{\ell} \eta_j f_{h_j} X^{k-1+t_j} : f_i \in \mathbb{F}_q \right\} \subseteq \mathbb{F}_q[X].$$

Definition 2 (Twisted Reed–Solomon code, [BPR17]). *Let the entries of $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_n) \in \mathbb{F}_q^n$ be pairwise distinct, and fix $1 \leq k \leq n$. Let $\mathbf{t}, \mathbf{h}, \boldsymbol{\eta}$ be defined as above. The $[\mathbf{t}, \mathbf{h}, \boldsymbol{\eta}]$ -twisted Reed–Solomon (TRS) code of length n , dimension k and locators $\boldsymbol{\alpha}$ is defined by*

$$\text{TRS}_{k,n}[\boldsymbol{\alpha}, \mathbf{t}, \mathbf{h}, \boldsymbol{\eta}] := \{\text{ev}_{\boldsymbol{\alpha}}(f) : f \in \mathcal{P}_{k,n}[\mathbf{t}, \mathbf{h}, \boldsymbol{\eta}]\}.$$

According to Definition 2, a generator matrix of $\text{TRS}_{k,n}[\boldsymbol{\alpha}, \mathbf{t}, \mathbf{h}, \boldsymbol{\eta}]$ is given by

$$\mathbf{G}_{\boldsymbol{\alpha}, \mathbf{t}, \mathbf{h}, \boldsymbol{\eta}} := \begin{pmatrix} 1 \\ \alpha^1 \\ \vdots \\ \alpha^{h_1-1} \\ \alpha^{h_1} + \eta_1 \alpha^{k-1+t_1} \\ \alpha^{h_1+1} \\ \vdots \\ \alpha^{h_\ell-1} \\ \alpha^{h_\ell} + \eta_\ell \alpha^{k-1+t_\ell} \\ \alpha^{h_\ell+1} \\ \vdots \\ \alpha^{k-1} \end{pmatrix},$$

where $\boldsymbol{\alpha}^i := (\alpha_1^i, \dots, \alpha_n^i)$ for $i = 1, \dots, k-1$.

In [BBPR18], the authors show that the construction of TRS codes according to Definition 2 does not necessarily lead to MDS codes. However, they provide a method to obtain a subfamily of MDS TRS codes, cf. Theorem 1.

Theorem 1 (Explicit MDS TRS codes [BBPR18]). *Let q_0 be a prime power, and $1 = s_0 < \dots < s_\ell \in \mathbb{Z}_{>0}$ be non-negative integers such that $\mathbb{F}_{q_0}^{s_0} \subset \mathbb{F}_{q_0}^{s_1} \subset \dots \subset \mathbb{F}_{q_0}^{s_\ell} = \mathbb{F}_q$ is a chain of subfields. Fix $k < n \leq q_0$ and the entries of $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_n) \in \mathbb{F}_{q_0}^n$ as pairwise distinct locators. Finally, let \mathbf{t}, \mathbf{h} and $\boldsymbol{\eta}$ be chosen as in Definition 2, such that $\eta_i \in \mathbb{F}_{q_0}^{s_i} \setminus \mathbb{F}_{q_0}^{s_{i-1}}$ for $i = 1, \dots, \ell$. Then $\text{TRS}_{k,n}[\boldsymbol{\alpha}, \mathbf{t}, \mathbf{h}, \boldsymbol{\eta}]$ is MDS.*

A decoding algorithm for TRS codes is also proposed in [BBPR18]. Given a corrupted codeword $\mathbf{r} = \mathbf{c} + \mathbf{e} \in \mathbb{F}_q^n$, where $\mathbf{c} \in \text{TRS}_{k,n}[\boldsymbol{\alpha}, \mathbf{t}, \mathbf{h}, \boldsymbol{\eta}]$, the strategy is to guess ℓ elements $g_1, \dots, g_\ell \in \mathbb{F}_q$ and then to decode $\mathbf{r} - \text{ev}_{\boldsymbol{\alpha}}(\sum_{i=1}^{\ell} g_i \eta_i X^{t_i+k-1})$ in the Reed–Solomon code $\text{RS}_{k,n}[\boldsymbol{\alpha}]$. This approach succeeds if $g_i = f_{h_i}$ and thus admits a worst case complexity in

$O(q^\ell n \log^2 n \log \log n)$. Notice that for the explicit family presented in Theorem 1, we have $q = \Omega(q_0^{2^\ell})$, hence this decoding algorithm is only practical for a tiny number of twists.

The following lemma shows that TRS codes are invariant under specific transformations of their parameters. This property is a key element for the cryptanalysis of the system, and could be of independent interest.

Lemma 2. *Let α , \mathbf{t} , \mathbf{h} and $\boldsymbol{\eta}$ be defined as in Definition 2. Then for any $a \in \mathbb{F}_q \setminus \{0\}$,*

$$\text{TRS}_{k,n}[\alpha, \mathbf{t}, \mathbf{h}, \boldsymbol{\eta}] = \text{TRS}_{k,n}[\hat{\alpha}, \mathbf{t}, \mathbf{h}, \hat{\boldsymbol{\eta}}],$$

where $\hat{\alpha} = a\alpha$ and $\hat{\boldsymbol{\eta}} = (\hat{\eta}_1, \dots, \hat{\eta}_\ell)$ with $\hat{\eta}_i = \eta_i a^{-(k-1+t_i-h_i)}$, $1 \leq i \leq \ell$.

Proof. Let $\text{ev}_{\hat{\alpha}}(f) \in \text{TRS}_{k,n}[\hat{\alpha}, \mathbf{t}, \mathbf{h}, \hat{\boldsymbol{\eta}}]$, where $f(X) = \sum_{i=0}^{k-1} f_i X^i + \sum_{j=1}^{\ell} \hat{\eta}_j f_{h_j} X^{k-1+t_j}$. We have

$$f(aX) = \sum_{i=0}^{k-1} (f_i a^i) X^i + \sum_{j=1}^{\ell} (\hat{\eta}_j a^{k-1+t_j-h_j}) (f_{h_j} a^{h_j}) X^{k-1+t_j} = g(X),$$

where $g(X) \in \mathcal{P}_{k,n}[\mathbf{t}, \mathbf{h}, \boldsymbol{\eta}]$. Hence by definition $\text{ev}_{\hat{\alpha}}(f) \in \text{TRS}_{k,n}[\alpha, \mathbf{t}, \mathbf{h}, \boldsymbol{\eta}]$, and it follows that $\text{TRS}_{k,n}[\hat{\alpha}, \mathbf{t}, \mathbf{h}, \hat{\boldsymbol{\eta}}] \subseteq \text{TRS}_{k,n}[\alpha, \mathbf{t}, \mathbf{h}, \boldsymbol{\eta}]$. The proof on the converse inclusion is similar since a is non-zero. \square \square

3 The variant of the McEliece cryptosystem using TRS codes

3.1 Definition of the cryptosystem

Setup. Fix a prime power q_0 , and integers $k < n \leq q_0 - 1$ with $2\sqrt{n} + 6 < k \leq \frac{n}{2} - 2$. Also fix $\ell \in \mathbb{Z}_{>0}$ satisfying

$$\frac{n+1}{k-\sqrt{n}} < \ell + 2 < \min \left\{ k + 3; \frac{2n}{k}; \sqrt{n} - 2 \right\}.$$

Further, set $q_i := q_{i-1}^2 = q_0^{2^i}$ for $i = 1, \dots, \ell$, such that $\mathbb{F}_{q_0} \subset \mathbb{F}_{q_1} \subset \dots \subset \mathbb{F}_{q_\ell} = \mathbb{F}_q$ is a chain of subfields. Finally, set $t_i = (i+1)(r-2) - k + 2$ and $h_i = r - 1 + i$ for $i = 1, \dots, \ell$, where $r := \lceil \frac{n+1}{\ell+2} \rceil + 2$.

Integers q_0 , n , k , ℓ , and tuples \mathbf{t} , \mathbf{h} satisfying the above conditions are referred to as *valid parameters* of the cryptosystem [BBPR18]. They are public parameters.

Key generation. Given valid parameters q_0 , n , k , ℓ , \mathbf{t} and \mathbf{h} , a pair of public/private keys is generated as follows.

1. Choose $\alpha \in \mathbb{F}_{q_0}^n$ at random such that the entries of α are pairwise distinct.
2. Choose $\boldsymbol{\eta} \in \mathbb{F}_q^\ell$ at random such that $\eta_i \in \mathbb{F}_{q_i} \setminus \mathbb{F}_{q_{i-1}}$ for $i = 1, \dots, \ell$.
3. Choose $\mathbf{S} \in \text{GL}_k(\mathbb{F}_q)$ at random.
4. Output the public key $\mathbf{G}_{\text{pub}} = \mathbf{S}\mathbf{G}_{\alpha, \mathbf{t}, \mathbf{h}, \boldsymbol{\eta}} \in \mathbb{F}_q^{k \times n}$, where $\mathbf{G}_{\alpha, \mathbf{t}, \mathbf{h}, \boldsymbol{\eta}}$ is the generator matrix of $\text{TRS}_{k,n}[\alpha, \mathbf{t}, \mathbf{h}, \boldsymbol{\eta}]$ described in Section 2.2.

The private key consists of $(\mathbf{S}, \alpha, \boldsymbol{\eta})$ and the public key is \mathbf{G}_{pub} .

Encryption. Given a plaintext $\mathbf{m} \in \mathbb{F}_q^k$ and the public key \mathbf{G}_{pub} , the ciphertext is generated as follows.

1. Choose $\mathbf{e} \in \mathbb{F}_q^n$ at random with Hamming weight $w_{\text{H}}(\mathbf{e}) = \lfloor \frac{n-k}{2} \rfloor$.
2. Output the ciphertext

$$\mathbf{y} := \mathbf{m}\mathbf{G}_{\text{pub}} + \mathbf{e} \in \mathbb{F}_q^n.$$

Decryption. Given a ciphertext $\mathbf{y} \in \mathbb{F}_q^n$ and the private key $(\mathbf{S}, \boldsymbol{\alpha}, \boldsymbol{\eta})$, the decryption algorithm can be described as follows.

1. Decode \mathbf{y} to $\tilde{\mathbf{m}} = \mathbf{m}\mathbf{S} \in \mathbb{F}_q^k$ using the decoding algorithm of $\text{TRS}_{k,n}[\boldsymbol{\alpha}, \mathbf{t}, \mathbf{h}, \boldsymbol{\eta}]$ given in [BBPR18].
2. Output the plaintext $\mathbf{m} = \tilde{\mathbf{m}}\mathbf{S}^{-1}$.

Proposed Parameters. The designers of the system proposed the parameters listed in Table 1 [BBPR18]. Recall that the public code is defined over the field $\mathbb{F}_q = \mathbb{F}_{q_0^{2^\ell}}$.

q_0	n	k	ℓ	\mathbf{t}	\mathbf{h}
256	255	117	1	(57)	(88)

Table 1: Parameters proposed in [BBPR18] for a claimed security ≥ 100 bits.

There are two main reasons for choosing a small number of twists. On the one hand, the complexity of the decoding algorithm proposed in [BBPR18] is in $O(q_0^{2^\ell} n \log^2 n \log \log n)$ and thus increases doubly exponentially with the number of twists. On the other hand, the number of elements in the largest field \mathbb{F}_q also scales exponentially with the number of twists, which impacts the key sizes.

3.2 Resistance to some known key-recovery algebraic attacks

As mentioned in Section 1, Beelen *et al.* showed that some existing attacks cannot be *directly* mounted on their system [BBPR18]. Let us recall these attacks and explain why they are ineffective.

Sidelnikov–Shestakov attack. In [SS92], Sidelnikov and Shestakov presented an attack on a variant of the McEliece cryptosystem using GRS codes. The attack uses two key facts: first, for MDS codes it is easy to find minimal-weight codewords with a given support, by running a simple Gaussian elimination; second, the ratio between two minimal-weight codewords of a GRS code, whose supports differ in only two coordinates, gives a rational function of degree one. Using these properties, the recovery of an alternate public key (*i.e.* an algebraic description of the public code as a GRS code) reduces to solving linear systems of equations involving the coefficients of the rational functions and the parameters of the GRS code. Formally, the result of Sidelnikov and Shestakov [SS92] can be summarized as follows.

Theorem 3 (Sidelnikov–Shestakov [SS92]). *Let $\text{RS}_{k,n}[\boldsymbol{\alpha}]$ be a Reed–Solomon code with locators $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_n) \in \mathbb{F}_{q_0}^n$. Given any generator matrix of $\text{RS}_{k,n}[\boldsymbol{\alpha}]$, there exists an algorithm which determines in time $O(n^4)$ a vector $\boldsymbol{\alpha}' \in \mathbb{F}_{q_0}^n$ such that*

$$\text{RS}_{k,n}[\boldsymbol{\alpha}] = \text{RS}_{k,n}[\boldsymbol{\alpha}'].$$

In particular, it holds that $\alpha' = a\alpha + b\mathbf{1} := (a\alpha_1 + b, \dots, a\alpha_n + b)$ with $a \in \mathbb{F}_{q_0} \setminus \{0\}$ and $b \in \mathbb{F}_{q_0}$.

However for TRS codes, the ratio of two minimal-weight codewords with close support is a high degree rational function involving many coefficients. This property prevents a direct use of Sidelnikov–Shestakov’s attack.

Wieschebrink attack. In order to attack a variant of McEliece cryptosystem using random subcodes of GRS codes, Wieschebrink considered the following structural properties. Let \mathcal{C} be a random subcode of dimension $k - m$ of a GRS code of dimension k , with m small compared to k . With high probability, the Schur square $\mathcal{C}^{(*2)}$ is a GRS code of dimension $\min\{n, 2k - 1\}$. If $k < n/2$, a Sidelnikov–Shestakov attack can be applied to recover the secret parameters. Otherwise, one can shorten the public code to fulfill the latter condition, since a shortened RS code is again a RS code.

As proved by the designers of the cryptosystem, Wieschebrink’s idea cannot be directly applied to TRS codes, due to a smart choice of parameters: the Schur square of the public code has dimension n , and shortening techniques seem unappropriate since the family of TRS codes is not stable under this operation. We will see in the following section that restricting TRS codes to subfields however leaks the algebraic structure of the public code.

4 An efficient key-recovery attack using subfield subcodes

This section presents an efficient key-recovery algorithm for the cryptosystem with the parameters proposed in [BBPR18]. The algorithm first determines a linear transformation of the secret locators α by exploiting structural properties of the *subfield subcode* of the public code. Then, the algorithm finds the coefficients of the twist monomials by Lagrange interpolation. The algorithm finally outputs $(\hat{S}, \hat{\alpha}, \hat{\eta})$ such that $\hat{S}G_{\hat{\alpha}, t, h, \hat{\eta}} = G_{\text{pub}}$. As shown in Lemma 2, $(\hat{S}, \hat{\alpha}, \hat{\eta})$ is a valid private key that can be used in the decryption algorithm (see Section 3.1).

4.1 Key-recovery algorithm

4.1.1 First step: recovery of an affine transformation of the secret locators

Let us consider the \mathbb{F}_{q_0} -subfield subcode of the code \mathcal{C}_{pub} spanned by the public generator matrix G_{pub} . We first state a technical lemma.

Lemma 4. *Let the entries of $\alpha = (\alpha_1, \dots, \alpha_n) \in \mathbb{F}_{q_0}^n$ be pairwise distinct. Further, let $P \in \mathbb{F}_q[X]$ where \mathbb{F}_q is an extension of \mathbb{F}_{q_0} , such that $\deg(P) < n$. Then, $\text{ev}_{\alpha}(P) \in \mathbb{F}_{q_0}^n$ if and only if $P \in \mathbb{F}_{q_0}[X]$.*

Proof. Let $c = \text{ev}_{\alpha}(P)$ and assume that $c \in \mathbb{F}_{q_0}^n$. Since $\alpha \in \mathbb{F}_{q_0}^n$ and $n \leq q_0$, there exists a polynomial $Q \in \mathbb{F}_{q_0}[X]$ of degree $\leq n$ such that $c = \text{ev}_{\alpha}(Q)$. Moreover, ev_{α} is injective over the \mathbb{F}_q -subspace of polynomials of degree $< q_0$, hence $P = Q$. The converse is straightforward. \square \square

Let us now define $\mathcal{I} := \{0, 1, \dots, k - 1\} \setminus \{h_1, \dots, h_{\ell}\}$ as the set of exponents of monomials which do not support twists¹. For valid parameters, $\mathcal{I} = \{0, 1, \dots, r - 1\} \cup \{r + \ell, \dots, k - 1\}$ since $h_i = r - 1 + i$ for each $1 \leq i \leq \ell$. We can now prove the following characterization of subfield subcodes of TRS codes with valid parameters.

¹Since the parameters k and h_1, \dots, h_{ℓ} are public, an attacker knows the set \mathcal{I} .

Proposition 5. *Let $\text{TRS}_{k,n}[\boldsymbol{\alpha}, \mathbf{t}, \mathbf{h}, \boldsymbol{\eta}]$ be chosen with valid parameters, as described in Section 3. Define $\mathcal{I} = \{0, 1, \dots, k-1\} \setminus \{h_1, \dots, h_\ell\}$ as above. Then,*

$$\text{TRS}_{k,n}[\boldsymbol{\alpha}, \mathbf{t}, \mathbf{h}, \boldsymbol{\eta}] \cap \mathbb{F}_{q_0}^n = \text{Span}_{\mathbb{F}_{q_0}}(\{\text{ev}_{\boldsymbol{\alpha}}(X^i), i \in \mathcal{I}\}).$$

Proof. Let us denote $\mathcal{S} = \text{Span}_{\mathbb{F}_{q_0}}(\{\text{ev}_{\boldsymbol{\alpha}}(X^i), i \in \mathcal{I}\})$ and $\mathcal{C}_{\text{pub}} = \text{TRS}_{k,n}[\boldsymbol{\alpha}, \mathbf{t}, \mathbf{h}, \boldsymbol{\eta}]$. First, it is clear that $\mathcal{S} \subseteq \mathcal{C}_{\text{pub}} \cap \mathbb{F}_{q_0}^n$. Indeed, for $i \in \mathcal{I}$ we have $\text{ev}_{\boldsymbol{\alpha}}(X^i) \in \mathcal{C}_{\text{pub}}$, and since $\boldsymbol{\alpha}$ is a vector over \mathbb{F}_{q_0} , it yields that $\text{ev}_{\boldsymbol{\alpha}}(X^i) \in \mathbb{F}_{q_0}^n$.

Conversely, let $\mathbf{c} = \text{ev}_{\boldsymbol{\alpha}}(f) \in \mathcal{C}_{\text{pub}} \cap \mathbb{F}_{q_0}^n$, where $f \in \mathcal{P}_{k,n}[\mathbf{t}, \mathbf{h}, \boldsymbol{\eta}]$. Lemma 4 ensures that $f \in \mathbb{F}_{q_0}[X]$, since $\deg(f) < n$ for valid parameters. It remains to notice that $\mathbb{F}_{q_0}[X] \cap \mathcal{P}_{k,n}[\mathbf{t}, \mathbf{h}, \boldsymbol{\eta}] = \text{Span}_{\mathbb{F}_{q_0}}(\{X^i, i \in \mathcal{I}\})$. \square \square

We observe by Proposition 5 that the subfield subcode $\mathcal{C}_{\text{sub}} := \text{TRS}_{k,n}[\boldsymbol{\alpha}, \mathbf{t}, \mathbf{h}, \boldsymbol{\eta}] \cap \mathbb{F}_{q_0}^n$ is a proper non-MDS subcode of the RS code $\text{RS}_{k,n}[\boldsymbol{\alpha}]$. Thus, one cannot directly use a Sidelnikov–Shestakov attack [SS92] on \mathcal{C}_{sub} . In 2006, Wieschebrink mounted an attack on cryptosystems based on *random* subcodes of RS codes [Wie10]. The author’s idea is that, with very high probability over the chosen subcode \mathcal{C}' , the square code $\mathcal{C}'^{(\star 2)}$ is a RS code. A Sidelnikov–Shestakov attack can then be used on $\mathcal{C}'^{(\star 2)}$ to recover the private parameters.

In the following, we prove that for most valid parameters defined in [BBPR18], and for *all* practical ones, the square code $\mathcal{C}_{\text{sub}}^{(\star 2)}$ is a RS code subject to a Sidelnikov–Shestakov attack.

Proposition 6. *Let $q_0, n, k, \ell, \mathbf{t}$ and \mathbf{h} be valid parameters, and assume that $\ell \leq \frac{1}{2}(\sqrt{n} - 3)$. Let $\mathcal{C}_{\text{sub}} = \text{TRS}_{k,n}[\boldsymbol{\alpha}, \mathbf{t}, \mathbf{h}, \boldsymbol{\eta}] \cap \mathbb{F}_{q_0}^n$. Then,*

$$(\mathcal{C}_{\text{sub}})^{(\star 2)} = \text{RS}_{2k-1,n}[\boldsymbol{\alpha}].$$

Proof. We use the notation and the results of Proposition 5. This yields

$$\begin{aligned} (\mathcal{C}_{\text{sub}})^{(\star 2)} &= \text{Span}_{\mathbb{F}_{q_0}}(\{\text{ev}_{\boldsymbol{\alpha}}(X^i) \star \text{ev}_{\boldsymbol{\alpha}}(X^j) : (i, j) \in \mathcal{I}\}) \\ &= \text{Span}_{\mathbb{F}_{q_0}}(\{\text{ev}_{\boldsymbol{\alpha}}(X^i) : i \in \mathcal{I} + \mathcal{I}\}). \end{aligned}$$

As a consequence, the theorem holds if $\mathcal{I} + \mathcal{I} = \{0, \dots, 2k-2\}$.

Notice that for valid parameters, we have $2k-1 \leq n-3$ and $\mathcal{I} = \mathcal{I}_1 \cup \mathcal{I}_2$, where $\mathcal{I}_1 = \{0, \dots, r-1\}$, $\mathcal{I}_2 = \{r+\ell, \dots, k-1\}$ and $r = \lceil \frac{n+1}{\ell+2} \rceil + 2$. We have $\{0\} + \mathcal{I} = \{0, \dots, r-1\}$, $\mathcal{I}_1 + \mathcal{I}_2 = \{r+\ell, \dots, k+r-2\}$ and $\{k-1\} + \mathcal{I}_2 = \{k+r+\ell-1, \dots, 2k-2\}$, hence it is clear that $\mathcal{I} + \mathcal{I}$ contains the subset

$$\{0, \dots, r-1\} \cup \{r+\ell, \dots, k+r-2\} \cup \{k+r+\ell-1, \dots, 2k-2\}.$$

Moreover one can easily check that if $\ell \leq r-1$, then $\{r, \dots, r+\ell-1\} \subseteq \mathcal{I}_1 + \mathcal{I}_1$. The condition $\ell \leq r-1$ is always satisfied with valid parameters since $\ell < \sqrt{n}-3$ and $r > \sqrt{n}$. Finally, the assumption $\ell \leq \frac{1}{2}(\sqrt{n}-3)$ leads us to $\ell \leq \frac{k-r-1}{2}$ using constraints on valid parameters. This easily yields $\{k+r-1, \dots, k+r+\ell-2\} \subseteq \mathcal{I}_2 + \mathcal{I}_2$. \square \square

Remark 7. *In practice, the assumption $\ell \leq \frac{1}{2}(\sqrt{n}-3)$ is not restrictive, since the decryption algorithm is effective only if $\ell \ll \log n$.*

For valid parameters, we have $2k-1 \leq n-3$, hence we can apply a Sidelnikov–Shestakov attack to the code $\mathcal{C}_{\text{sub}}^{(\star 2)} \subseteq \mathbb{F}_{q_0}^n$. This algorithm outputs a vector of locators $\boldsymbol{\alpha}' \in \mathbb{F}_{q_0}^n$ which is an affine transformation of the secret locators $\boldsymbol{\alpha}$ (see Theorem 3). Formally, $\boldsymbol{\alpha}' = a\boldsymbol{\alpha} + b\mathbf{1}$ for some $a \in \mathbb{F}_{q_0} \setminus \{0\}$ and $b \in \mathbb{F}_{q_0}$, where $\mathbf{1} := (1, \dots, 1) \in \mathbb{F}_{q_0}^n$.

4.1.2 Second step: from an affine to a linear transformation of the secret locators

Lemma 2 only ensures that $\text{TRS}_{k,n}[\boldsymbol{\alpha}, \mathbf{t}, \mathbf{h}, \boldsymbol{\eta}] = \text{TRS}_{k,n}[\hat{\boldsymbol{\alpha}}, \mathbf{t}, \mathbf{h}, \hat{\boldsymbol{\eta}}]$ if $\hat{\boldsymbol{\alpha}} = a\boldsymbol{\alpha}$ for a non-zero $a \in \mathbb{F}_{q_0}$. Therefore, given $\boldsymbol{\alpha}' = a\boldsymbol{\alpha} + b\mathbf{1}$, the search of a valid $b \in \mathbb{F}_{q_0}$ such that $\boldsymbol{\alpha}' - b\mathbf{1} = a\boldsymbol{\alpha}$ remains. Since q_0 is rather small, this search can be proceeded exhaustively as follows. Given $\boldsymbol{\alpha}'$ and $b \in \mathbb{F}_{q_0}$, one first computes the code

$$\mathcal{A}_b := \text{Span}_{\mathbb{F}_{q_0}} (\{\text{ev}_{\boldsymbol{\alpha}' - b\mathbf{1}}(X^i) : i \in \mathcal{I}\}).$$

If $\mathcal{A}_b \subseteq \mathcal{C}_{\text{pub}}$ holds, then we have found a valid b and hence a valid $\hat{\boldsymbol{\alpha}} = \boldsymbol{\alpha}' - b\mathbf{1}$. Notice that each individual test $\mathcal{A}_b \subseteq \mathcal{C}_{\text{pub}}$ can be performed in time $O(n^3)$.

4.1.3 Third step: recovery of a valid pair $(\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\eta}})$

The previous steps provide a tuple $\hat{\boldsymbol{\alpha}} \in \mathbb{F}_{q_0}^n$ which can be used as a vector of locators for the public TRS code. In order to determine a vector $\hat{\boldsymbol{\eta}} \in \mathbb{F}_q^n$ such that $\text{TRS}_{k,n}[\boldsymbol{\alpha}, \mathbf{t}, \mathbf{h}, \boldsymbol{\eta}] = \text{TRS}_{k,n}[\hat{\boldsymbol{\alpha}}, \mathbf{t}, \mathbf{h}, \hat{\boldsymbol{\eta}}]$, we use the following result.

Lemma 8. *Let $1 \leq \ell$, and $P(X) = \sum_{i=0}^{k-1} u_i X^i + \sum_{j=1}^{\ell} \eta_j u_{h_j} X^{k-1+t_j} \in \mathcal{P}_{k,n}[\mathbf{t}, \mathbf{h}, \boldsymbol{\eta}]$ such that $u_{h_j} \neq 0$. Denote by \hat{p}_{h_j} and \hat{p}_{k-1+t_j} the coefficients of the monomials X^{h_j} and X^{k-1+t_j} in $\hat{P}(X) = P(a^{-1}X)$. Then, we have*

$$\hat{\eta}_j = \eta_j a^{-(k-1+t_j-h_j)} = \frac{\hat{p}_{k-1+t_j}}{\hat{p}_{h_j}}.$$

Proof. This is clear from the following simple computation

$$\hat{P}(X) := P(a^{-1}X) = \sum_{i=0}^{k-1} u_i a^{-i} X^i + \sum_{j=1}^{\ell} \eta_j u_{h_j} a^{-(k-1+t_j)} X^{k-1+t_j}.$$

□

□

Hence, a vector of coefficients $\hat{\boldsymbol{\eta}}$ such that $\text{TRS}_{k,n}[\boldsymbol{\alpha}, \mathbf{t}, \mathbf{h}, \boldsymbol{\eta}] = \text{TRS}_{k,n}[\hat{\boldsymbol{\alpha}}, \mathbf{t}, \mathbf{h}, \hat{\boldsymbol{\eta}}]$ can be computed as follows. Pick at random a codeword $\mathbf{c} = \text{ev}_{\boldsymbol{\alpha}}(P) \in \mathcal{C}_{\text{pub}} = \text{TRS}_{k,n}[\boldsymbol{\alpha}, \mathbf{t}, \mathbf{h}, \boldsymbol{\eta}]$. Then, interpolate $\mathbf{c} = \text{ev}_{\hat{\boldsymbol{\alpha}}}(\hat{P})$ as a polynomial evaluated over the vector of locators $\hat{\boldsymbol{\alpha}}$. Notice that we have $\hat{P}(X) = P(a^{-1}X)$, thus for every non-zero coefficient u_{h_j} of P , we obtain the coefficient $\hat{\eta}_j$ due to Lemma 8.

It remains to be observed that, if a codeword \mathbf{c} is picked uniformly at random in \mathcal{C}_{pub} , the probability that $u_j = 0$ is roughly $1/q$. Since $\ell \ll q$, a random \mathbf{c} leads to the recovery of the whole vector $\hat{\boldsymbol{\eta}}$ with high probability. Note that this procedure can be derandomized by iteratively taking each row the public matrix \mathbf{G}_{pub} .

4.1.4 Final step: recovery of an alternative private key $(\hat{\mathbf{S}}, \hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\eta}})$

After determining $\hat{\boldsymbol{\alpha}}$ and $\hat{\boldsymbol{\eta}}$, one can easily compute a matrix $\hat{\mathbf{S}}$ such that $\hat{\mathbf{S}}\mathbf{G}_{\hat{\boldsymbol{\alpha}}, \mathbf{t}, \mathbf{h}, \hat{\boldsymbol{\eta}}} = \mathbf{G}_{\text{pub}}$. Then, $(\hat{\mathbf{S}}, \hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\eta}})$ can be used in the proposed decryption algorithm as a valid (alternative) private key to retrieve any secret plaintext \mathbf{m} .

Algorithm 1 Key-recovery attack

Input: \mathbf{G}_{pub} **Output:** $\hat{\mathbf{S}}, \hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\eta}}$

Step 1: recovery of some locators

- 1: $\mathbf{G}_{\text{sub}} \leftarrow \text{SubfieldSubcode}(\mathbf{G}_{\text{pub}}) \in \mathbb{F}_{q_0}^{(k-\ell) \times n}$
- 2: $\mathbf{G}_{\text{sq}} \leftarrow \text{Square}(\mathbf{G}_{\text{sub}}) \in \mathbb{F}_{q_0}^{(2k-1) \times n}$
- 3: $\boldsymbol{\alpha}' \leftarrow \text{SidelShest}(\mathbf{G}_{\text{sq}}) \in \mathbb{F}_{q_0}^n$

Step 2: exhaustive search for b

- 4: **for all** $b \in \mathbb{F}_{q_0}$ **do**
- 5: $\hat{\boldsymbol{\alpha}} \leftarrow (\alpha'_1 - b, \dots, \alpha'_n - b) \in \mathbb{F}_{q_0}^n$
- 6: $\mathbf{G}' \leftarrow \text{GenSub}(\hat{\boldsymbol{\alpha}}) \in \mathbb{F}_{q_0}^{(k-\ell) \times n}$
- 7: **if** $\mathbf{G}'(\mathbf{G}_{\text{sub}}^\perp)^\top = \mathbf{0}$ **then**
- 8: **break**

Step 3: recovery of $\hat{\boldsymbol{\eta}}$

- 9: $J \leftarrow \{1, \dots, \ell\}$
- 10: **for all** row \mathbf{r}_i of \mathbf{G}_{pub} **do**
- 11: $P(X) \leftarrow \text{Interpolate}(\boldsymbol{\alpha}', \mathbf{r}_i) \in \mathbb{F}_q^n$
- 12: **for all** $j \in J$ **do**
- 13: **if** $p_{h_j} \neq 0$ **then**
- 14: $\hat{\eta}_j \leftarrow \frac{p_{k-1+t_j}}{p_{h_j}} \in \mathbb{F}_q$
- 15: $J \leftarrow J \setminus \{j\}$
- 16: **if** $J = \emptyset$ **then**
- 17: **break**

Step 4: recovery of $\hat{\mathbf{S}}$

- 18: $\hat{\mathbf{G}}_{\text{TRS}} \leftarrow \text{GTRS}(\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\eta}}) \in \mathbb{F}_q^{k \times n}$
 - 19: $\hat{\mathbf{S}} \leftarrow \hat{\mathbf{G}}_{\text{TRS}} \setminus \mathbf{G}_{\text{pub}} \in \mathbb{F}_q^{k \times k}$
 - 20: **return** $\hat{\mathbf{S}}, \hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\eta}}$
-

4.2 Analysis of the attack

A summary of the attack is given in Algorithm 1. Let us explain the notation we use there. Given a matrix $\mathbf{A} \in \mathbb{F}_q^{k \times n}$, its transpose is represented by \mathbf{A}^\top , and \mathbf{A}^\perp is a matrix whose rows form a basis of the right kernel of \mathbf{A} . The reduced row echelon form of \mathbf{A} is denoted by $\text{rref}(\mathbf{A})$. Moreover, if $\mathbf{A} \in \mathbb{F}_q^{k \times n}$ and $\mathbf{B} \in \mathbb{F}_q^{k \times n}$ have the same rowspace, then $\mathbf{D} = \mathbf{A} \setminus \mathbf{B}$ denotes any solution to $\mathbf{D}\mathbf{A} = \mathbf{B}$. Finally, in Table 2 we describe functions involved in Algorithm 1.

Theorem 9. *Given any generator matrix \mathbf{G}_{pub} of a TRS code $\mathcal{C}_{\text{pub}} = \text{TRS}_{k,n}[\boldsymbol{\alpha}, \mathbf{t}, \mathbf{h}, \boldsymbol{\eta}] \subseteq \mathbb{F}_q^n$, Algorithm 1 retrieves a tuple $(\hat{\mathbf{S}}, \hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\eta}})$ such that the matrix $\hat{\mathbf{S}}\mathbf{G}_{\hat{\boldsymbol{\alpha}}, \mathbf{t}, \mathbf{h}, \hat{\boldsymbol{\eta}}}$ generates \mathcal{C}_{pub} in $O(\max\{q_0, 2^\ell, n\} \cdot n^3)$ operations over \mathbb{F}_q .*

Proof. The correctness of Algorithm 1 was proved in Section 4.1. Let us now provide details about the complexity of Algorithm 1.

Function	Description
SubfieldSubcode	maps a generator matrix of \mathcal{C}_{pub} to a generator matrix of the subfield subcode of \mathcal{C}_{pub}
Square	maps a generator matrix of \mathcal{C}_{sub} to a generator matrix of the code $\mathcal{C}_{\text{sub}}^{(*2)}$
Interpolate	maps vectors $(\mathbf{a}, \mathbf{b}) \in (\mathbb{F}_q^n)^2$ to $P(X)$ of degree $< n$ such that $P(a_i) = b_i$ for $i = 1, \dots, n$
GenSub	maps a vector $\mathbf{a} = (a_1, \dots, a_n) \in \mathbb{F}_{q_0}^n$ to a matrix $\mathbf{A} \in \mathbb{F}_{q_0}^{(k-\ell) \times n}$ whose rows are (a_1^j, \dots, a_n^j) for each $j \in \mathcal{I} = \{0, \dots, k-1\} \setminus \{h_1, \dots, h_\ell\}$.
SidelShest	implements a Sidelnikov–Shestakov attack, which takes a generator matrix \mathbf{G} of a RS code as input, and returns a vector of locators $\hat{\alpha}'$ describing the code
GTRS	maps the vectors $\hat{\alpha}$ and $\hat{\eta}$ to the generator matrix $\mathbf{G}_{\hat{\alpha}, t, h, \hat{\eta}}$ of the corresponding TRS code

Table 2: List of functions used in Algorithm 1.

- Line 1: The computation of $\mathbf{G}_{\text{sub}} \in \mathbb{F}_{q_0}^{(k-\ell) \times n}$ requires $O(n^2(k+n)) \subseteq O(n^3)$ operations in \mathbb{F}_q and $O(n^2(2^\ell(n-k) + n)) \subseteq O(2^\ell n^3)$ operations in \mathbb{F}_{q_0} .
- Line 2: The computation of $\mathbf{G}_{\text{sq}} \in \mathbb{F}_{q_0}^{(2k-1) \times n}$ can be performed in time $O(n^4)$. Informally, one needs to find a basis of the space generated by the set $\{\mathbf{g}_{i,j} := (\mathbf{G}_{\text{sub}})_i \star (\mathbf{G}_{\text{sub}})_j, 1 \leq i, j \leq \dim \mathcal{C}_{\text{sub}}\}$. This basis can be built iteratively; updating the basis with a new element costs $O(n^3)$ operations in \mathbb{F}_{q_0} and must be done $O(n)$ times, and rejecting candidates costs $O(n^2)$ operations in \mathbb{F}_{q_0} and must be done $O(n^2)$ times.
- Line 3: Applying the SidelShest function on $\mathbf{G}_{\text{sq}} \in \mathbb{F}_{q_0}^{(2k-1) \times n}$ requires $O((2k-2)^4 + (2k-2)n) \subseteq O(n^4)$ operations in \mathbb{F}_{q_0} [SS92].
- Line 4 to Line 8: The computation of $\hat{\alpha} \in \mathbb{F}_{q_0}^n$ requires $O(n)$ operations in \mathbb{F}_{q_0} ; building $\mathbf{G}' \in \mathbb{F}_{q_0}^{(k-\ell) \times n}$ needs $O((k-\ell)n)$ operations in \mathbb{F}_{q_0} ; matrix multiplication $\mathbf{G}'(\mathbf{G}_{\text{sub}}^\perp)^\top$ needs $O((k-\ell)(n-k+\ell)n) \subseteq O(n^3)$ operations in \mathbb{F}_{q_0} ($\mathbf{G}_{\text{sub}}^\perp$ was already computed in Line 1). In the worst case, the previous sequences of computations have to be performed q_0 times. Hence these steps require $O(q_0 n^3)$ operations in \mathbb{F}_{q_0} .
- Line 10 to Line 17: In the worst case, $\ell \cdot k$ interpolations have to be performed, requiring $O(\ell k n^2) \subseteq O(\ell n^3)$ operations in \mathbb{F}_q .
- Line 18: Computation of $\hat{\mathbf{G}}_{\text{TRS}} \in \mathbb{F}_q^{k \times n}$ needs $O(kn) \subseteq O(n^2)$ operations in \mathbb{F}_q .
- Line 19: Computation of $\hat{\mathbf{S}} \in \mathbb{F}_q^{k \times k}$ by a reduction to row echelon form of the matrix $\begin{pmatrix} \hat{\mathbf{G}}_{\text{TRS}}^\top & \mathbf{G}_{\text{pub}}^\top \end{pmatrix} \in \mathbb{F}_q^{n \times 2k}$ needs $O(n^2(2k)) \subseteq O(n^3)$ operations in \mathbb{F}_q . \square

\square

In practice, ℓ and $q_0 = q^{1/2^\ell}$ must be chosen to be small (for instance, $\ell = 1$ and $q_0 = n + 1 = 2^8$ were proposed in [BBPR18]) in order to obtain an efficient decryption algorithm and keys of moderate size. Hence, for practical parameters Algorithm 1 has a complexity in $O(n^4)$ and thus recovers a valid private key in polynomial time.

q_0	n	k	ℓ	$w_H(\mathbf{e})$	Claimed security level	Runtime of Algorithm 1
2^8	255	117	1	83	128 bits	133 seconds
2^8	255	117	2	83	128 bits	141 seconds
2^9	511	200	3	192	196 bits	2260 seconds
2^9	511	170	3	217	256 bits	1532 seconds

Table 3: Experimental results obtained by averaging several runtimes of Algorithm 1 on an Intel(R) Core(TM) i7-7600U CPU @ 2.80GHz. The first row refers to parameters proposed by the designers of the system. Remaining security levels were computed according to formulae given in [BBPR18].

Our attack is implemented in the computer algebra system SageMath v8.7 [The19] and is available at https://bitbucket.org/julianrenner/trs_attack. Although the implementation is not optimized, it recovers a valid private key within a few minutes for the proposed parameters, see Table 3.

5 Discussion and open questions

5.1 Repairing the cryptosystem?

After a notification of this attack, the authors of [BBPR18] described a possible fix of the system, in which a modified version of the generator matrix is made public. The idea is to multiply the generator matrix \mathbf{G}_{pub} from the right by a diagonal matrix with non-zero entries $\mathbf{y} = (y_1, \dots, y_n) \in (\mathbb{F}_q \setminus \{0\})^n$, such that the \mathbb{F}_{q_0} -subfield subcode of the vector space spanned by the rows of \mathbf{G}_{pub} is not contained in a RS code. This clearly prevents a direct application of our attack.

Nevertheless, we would like to point out that this possible repair might not fix the inherent weaknesses of the cryptosystem. In fact, the subfield subcode of a GRS code $\mathbf{y} \star \text{RS}_{k,n}[\alpha]$ is a so-called *alternant code* $\text{Alt}_{k',n}[\alpha, \mathbf{y}] \subseteq \mathbb{F}_{q_0}^n$, which also admits an algebraic description. As a consequence, it seems very plausible that the security of the proposed repaired cryptosystem can be reduced to the security of a McEliece-like cryptosystem using the subfield subcode $\text{Alt}_{k',n}[\alpha, \mathbf{y}]$.

One can then notice that the parameters proposed by the authors of [BBPR18] are far below those considered as secure for alternant codes. For instance, BIG QUAKE [BBB⁺17] and Classic McEliece [Dan17] (both are unbroken candidates for the NIST standardization call on post-quantum cryptography) use alternant codes with a length and dimension of several thousands, while in the proposed parameters for the TRS codes, we have $n = 255$ and $k = 117$ with a field size $q_0 = 2^8$. Algebraic attacks as developed in [FOPT10, FOP⁺16] should then be considered as a potential threat. One should also mention the recent attack on the cryptosystem DAGS [BBB⁺18] based on alternant codes, performed by Barelli and Couvreur [BC18]. Informally, the authors of [BC18] manage to derive an alternant code with much smaller parameters from the public code, allowing the last step of the key recovery algorithm — which is exponential in the involved parameters — to remain feasible due to the small size of the derived alternant code.

Finally and most crucially, one can question the possible benefit to consider codes whose security might not be better than those based on alternant codes (for which cryptosystems

have been designed and studied), but which suffer from larger key sizes and much less efficient decoding algorithms.

5.2 On the rank metric version of the cryptosystem

In [PRW18] a modified version of the previous system was proposed, based on a subfamily of twisted Gabidulin codes. The idea is to consider a variant of the GPT cryptosystem [GPT91], where twisted Gabidulin codes are used instead of (subcodes of) Gabidulin codes. Although we do not claim to have a proper attack on the system, let us show some potential weaknesses that could be analyzed in a future work.

The GPT cryptosystem can be viewed as an analogue of the McEliece cryptosystem, using rank metric codes instead of Hamming metric codes. We refer to [Ove07] for more details about rank metric codes and variants of the GPT cryptosystem. Let us give a short overview of the latter.

Let $\mathbb{F}_p \subset \mathbb{F}_q$ and $\Gamma \subseteq \{\mathcal{C} \subseteq \mathbb{F}_q^{n-t}, \dim \mathcal{C} = k\}$ be a family of rank metric codes. The GPT cryptosystem works as follows:

- *Key generation:* Alice generates a secret generator matrix $\mathbf{G} \in \mathbb{F}_q^{n-t}$ for a code \mathcal{C} randomly chosen in Γ . Then she computes a public key $\mathbf{G}_{\text{pub}} = \mathbf{S}[\mathbf{X}|\mathbf{G}]\mathbf{P}$, where the matrices $\mathbf{S} \in \text{GL}_k(\mathbb{F}_q)$, $\mathbf{X} \in \mathbb{F}_q^{k \times t}$ of rank $s \leq t$, and $\mathbf{P} \in \text{GL}_n(\mathbb{F}_p)$ are chosen uniformly at random and kept secret.
- *Encryption:* given a plaintext $\mathbf{m} \in \mathbb{F}_q^k$, Bob computes the ciphertext $\mathbf{y} = \mathbf{m}\mathbf{G}_{\text{pub}} + \mathbf{e}$, where $\mathbf{e} \in \mathbb{F}_q^n$ is a random error with small rank over \mathbb{F}_p (the rank of the error is such that \mathbf{e} can be decoded in \mathcal{C}).
- *Decryption:* Alice decodes the last $n - t$ coordinates of $\mathbf{y}\mathbf{P}^{-1}$ in the code \mathcal{C} and retrieves \mathbf{m} .

In most variants of the GPT cryptosystem, Γ is a (sub-)family of Gabidulin codes [Gab85]

$$\mathcal{G}_{k,n-t}[\boldsymbol{\alpha}] = \text{Span}_{\mathbb{F}_q} \{ \text{ev}_{\boldsymbol{\alpha}}(X^{[i]}), i = 0, \dots, k-1 \},$$

where $\boldsymbol{\alpha} \in \mathbb{F}_q^{n-t}$ are \mathbb{F}_p -linearly independent, and $X^{[i]} := X^{p^i}$. Polynomials with monomials only of the form $X^{[i]}$ are called p -polynomials, or linearized polynomials. In [PRW18], the authors proposed to define Γ as the subfamily of twisted Gabidulin codes

$$\mathcal{G}_{n-t,k}[\boldsymbol{\alpha}, \mathbf{t}, \mathbf{h}, \boldsymbol{\eta}] = \left\{ \text{ev}_{\boldsymbol{\alpha}}(f) : f \in \left\{ \sum_{i=0}^{k-1} f_i X^{[i]} + \sum_{j=1}^{\ell} \eta_j f_{h_j} X^{[k-1+t_j]} : f_i \in \mathbb{F}_q \right\} \right\},$$

where η_i are chosen in the chain of subfields $\mathbb{F}_{q_0} \subset \mathbb{F}_{q_1} \subset \dots \subset \mathbb{F}_{q_\ell} = \mathbb{F}_q$, and $(\alpha_1, \dots, \alpha_{n-t}) \in \mathbb{F}_{q_0}^{n-t}$ are \mathbb{F}_p -linearly independent, similar to the case of TRS codes.

Our claim is that the code \mathcal{C}_{pub} generated by \mathbf{G}_{pub} also admits structured subfield subcodes which could be used to attack the system. Indeed, one can prove that the last $n - t$ coordinates of $(\mathcal{C}_{\text{pub}} \cap \mathbb{F}_{q_0}^n) \mathbf{P}^{-1}$ form a subcode of the Gabidulin code $\mathcal{G}_{k,n-t}[\boldsymbol{\alpha}] \subseteq \mathbb{F}_{q_0}^{n-t}$ of rather small codimension. Applying variants of Overbeck's attacks — e.g. in [Ove05] — might lead to the recovery of a linear transformation of $\boldsymbol{\alpha}$ and thus a structural attack on the public key close to the one presented in this paper.

For a code $\mathcal{A} \subseteq \mathbb{F}_q^n$ and $f \geq 0$, let $\mathcal{A}^{[1]} := \{(a_1^{[f]}, \dots, a_n^{[f]})\}$, $\mathbf{a} \in \mathcal{A}$, and

$$\Lambda_f(\mathcal{A}) := \mathcal{A} + \mathcal{A}^{[1]} + \dots + \mathcal{A}^{[f]}.$$

In fact, we observe in simulations that for $f = n - k - t - 1$, if $\Lambda_f(\mathcal{C}_{\text{pub}} \cap \mathbb{F}_{q_0}^n)$ has dimension $n - 1$, then one recovers an \mathbb{F}_p -linear transformation $\hat{\alpha}$ of α , as well as a full-rank matrix $\hat{P} \in \mathbb{F}_p^{n \times n}$, by applying [Ove07, Algorithm 3.5.1] to a generator matrix of $\mathcal{C}_{\text{pub}} \cap \mathbb{F}_{q_0}^n$. Then, the coefficients $\hat{\eta}$ are determined by interpolation of the last $n - t$ positions of the rows of $\mathbf{G}_{\text{pub}} \hat{P}^{-1}$ with p -polynomials of p -degree smaller than n , similar to Section 4.1.3. Finally, one chooses \hat{S} such that

$$\hat{S}\hat{G} = (\mathbf{G}_{\text{pub}} \hat{P}^{-1})_{[t+1:n]},$$

where subscript $[t+1 : n]$ refers to the last $n - t$ positions of $\mathbf{G}_{\text{pub}} \hat{P}^{-1}$ and \hat{G} is a generator matrix of $\mathcal{G}_{n-t,k}[\hat{\alpha}, t, \mathbf{h}, \hat{\eta}]$. Clearly, $(\hat{S}, \hat{\alpha}, \hat{\eta}, \hat{P})$ is then a valid private key.

Further simulations show that if \mathbf{X} has full \mathbb{F}_q -rank and t is small, then the code $\Lambda_f(\mathcal{C}_{\text{pub}} \cap \mathbb{F}_{q_0}^n)$ has a dimension $n - 1$ with high probability. However, if t is large or if \mathbf{X} has \mathbb{F}_q -rank smaller than t , then $\Lambda_f(\mathcal{C}_{\text{pub}} \cap \mathbb{F}_{q_0}^n)$ has dimension smaller than $n - 2$ and this straightforward approach fails.

Since a precise analysis of the potential weakness of system proposed in [PRW18] is out of the scope of this paper, we leave it as an open problem for future research.

6 Conclusion

This paper presents an efficient key-recovery attack on the McEliece cryptosystem based on a subfamily of TRS codes. The attack does not contradict the structural properties presented in [BBPR18], but recovers the structure of a *subfield subcode* of the public TRS code, which enables us to determine a description of the supercode. This attack retrieves a valid private key from the public key for all practical parameters in $O(n^4)$ field operations. This is formally proved, and confirmed by experimental results: one retrieves a valid private key for a claimed security level of 128 bits within a few minutes. In addition, the security of the system after an attempt to repair it is discussed, as well as potential ways to adapt our attack to the rank metric variant of the considered system.

The subfield subcode approach presented in this paper is unique, in the sense that a widespread idea considers the restriction of codes to subfields as a way to break their structure. However, our cryptanalysis proves that subfield subcodes — as well as punctured codes and shortened codes — must also be taken into account when trying to assert the security of McEliece-like cryptosystems.

Acknowledgements

This work was done while the second author was visiting the Institut de Recherche Mathématique de Rennes (IRMAR), Université de Rennes 1, France.

The first author is funded by the French *Direction Générale l'Armement*, through the *Pôle d'excellence cyber*.

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 801434).

We would like to thank Antonia Wachter-Zeh (TUM) for fruitful discussions and Oliver De Candido (TUM) for his comments that helped to improve the manuscript. We would further like to thank the authors of the proposed cryptosystem [BBPR18] for validating our attack and pointing out a possible repair of the system with respect to our attack.

References

- [BBB⁺17] Magali Bardet, Élise Barelli, Olivier Blazy, Rodolfo C. Torres, Alain Couvreur, Philippe Gaborit, Ayoub Otmani, Nicolas Sendrier, and Jean-Pierre Tillich. BIG QUAKE BInary Goppa QUAsi-cyclic Key Encapsulation. <https://bigquake.inria.fr>, 2017.
- [BBB⁺18] Gustavo Banegas, Paulo S. L. M. Barreto, Brice O. Boidje, Pierre-Louis Cayrel, Gilbert N. Dione, Kris Gaj, Cheikh T. Gueye, Richard Haeussler, Jean B. Klamti, Ousmane Ndiaye, Duc T. Nguyen, Edoardo Persichetti, and Jefferson E. Ricardini. DAGS: Key Encapsulation Using Dyadic GS Codes. *J. Mathematical Cryptology*, 12(4):221–239, 2018.
- [BBPR18] Peter Beelen, Martin Bossert, Sven Puchinger, and Johan Rosenkilde né Nielsen. Structural Properties of Twisted Reed–Solomon Codes with Applications to Code-Based Cryptography. In *IEEE Int. Symp. Inf. Theory (ISIT)*, 2018.
- [BC18] Élise Barelli and Alain Couvreur. An Efficient Structural Attack on NIST Submission DAGS. In Thomas Peyrin and Steven D. Galbraith, editors, *Advances in Cryptology - ASIACRYPT*, volume 11272, pages 93–118. Springer, 2018.
- [BCGO09] Thierry P. Berger, Pierre-Louis Cayrel, Philippe Gaborit, and Ayoub Otmani. Reducing Key Length of the McEliece Cryptosystem. In Bart Preneel, editor, *Progress in Cryptology - AFRICACRYPT*, volume 5580, pages 77–97. Springer, 2009.
- [BL05] Thierry P. Berger and Pierre Loidreau. How to Mask the Structure of Codes for a Cryptographic Use. *Designs, Codes and Cryptogr.*, 35(1):63–79, Apr 2005.
- [BPR17] Peter Beelen, Sven Puchinger, and Johan Rosenkilde né Nielsen. Twisted Reed–Solomon Codes. In *IEEE Int. Symp. Inf. Theory (ISIT)*, 2017.
- [CCP17] Alain Couvreur, Irene M. Corbella, and Ruud Pellikaan. Cryptanalysis of McEliece Cryptosystem Based on Algebraic Geometry Codes and Their Subcodes. *IEEE Trans. Information Theory*, 63(8):5404–5418, 2017.
- [CGGU⁺14] Alain Couvreur, Philippe Gaborit, Valérie Gauthier-Umaña, Ayoub Otmani, and Jean-Pierre Tillich. Distinguisher-Based Attacks on Public-Key Cryptosystems Using Reed–Solomon Codes. *Designs, Codes and Cryptogr.*, 73(2):641–666, Nov 2014.
- [CLT19] Alain Couvreur, Matthieu Lequesne, and Jean-Pierre Tillich. Recovering short secret keys of RLCE in polynomial time. In Jintai Ding and Rainer Steinwandt, editors, *Post-Quantum Cryptography - 10th International Conference, PQCrypto 2019, Chongqing, China, May 8-10, 2019 Revised Selected Papers*, volume 11505 of *Lecture Notes in Computer Science*, pages 133–152. Springer, 2019.
- [Dan17] Daniel J. Bernstein and Tung Chou and Tanja Lange and Ingo von Maurich and Rafael Misoczki and Ruben Niederhagen and Edoardo Persichetti and

- Christiane Peters and Peter Schwabe and Nicolas Sendrier and Jakub Szefer and Wen Wang. Classic McEliece. <https://classic.mceliece.org>, 2017.
- [FOP⁺16] Jean-Charles Faugère, Ayoub Otmani, Ludovic Perret, Frédéric de Portzamparc, and Jean-Pierre Tillich. Structural Cryptanalysis of McEliece Schemes with Compact Keys. *Des. Codes Cryptogr.*, 79(1):87–112, 2016.
- [FOPT10] Jean-Charles Faugère, Ayoub Otmani, Ludovic Perret, and Jean-Pierre Tillich. Algebraic Cryptanalysis of McEliece Variants with Compact Keys. In Henri Gilbert, editor, *Advances in Cryptology - EUROCRYPT 2010*, volume 6110, pages 279–298. Springer, 2010.
- [Gab85] Ernst M. Gabidulin. Theory of Codes with Maximum Rank Distance. *Probl. Inf. Transm.*, 21(1):3–16, 1985.
- [GPT91] Ernst M. Gabidulin, A.V. Paramonov, and O.V. Tretjakov. Ideals over a Non-Commutative Ring and Their Application in Cryptology. In *Workshop Theory and Appl. Cryptogr. Techn.*, pages 482–489. Springer, 1991.
- [JM96] Heeralal Janwa and Oscar Moreno. McEliece Public Key Cryptosystems Using Algebraic-Geometric Codes. *Des. Codes Cryptogr.*, 8(3):293–307, 1996.
- [McE78] Robert J. McEliece. A Public-Key Cryptosystem Based on Algebraic Coding Theory. *Jet Propulsion Laboratory DSN Progress Report*, 42–44:114–116, 1978.
- [MS07] Lorenz Minder and Amin Shokrollahi. Cryptanalysis of the Sidelnikov Cryptosystem. In *Advances in Cryptology - EUROCRYPT 2007*, volume 4515, pages 347–360. Springer, 2007.
- [Nie86] Harald Niederreiter. Knapsack type cryptosystems and algebraic coding theory. *Probl. Control Inf. Theory*, 15, 01 1986.
- [Ove05] Raphael Overbeck. A New Structural Attack for GPT and Variants. *LNCS: MYCRYPT*, 3715:50–63, 2005.
- [Ove07] Raphael Overbeck. *Public Key Cryptography Based on Coding Theory*. PhD thesis, Darmstadt University of Technology, Germany, 2007.
- [PRW18] Sven Puchinger, Julian Renner, and Antonia Wachter-Zeh. Twisted Gabidulin Codes in the GPT Cryptosystem. In *Int. Workshop Alg. Combin. Coding Theory (ACCT)*, 2018.
- [Puc18] Sven Puchinger. *Construction and Decoding of Evaluation Codes in Hamming and Rank Metric*. PhD thesis, Ulm University, Germany, 2018.
- [Sid94] M. V. Sidelnikov. Public-key Cryptosystem Based on Binary Reed-Muller Codes. *Discrete Math. Appl.*, 4:191–208, 01 1994.
- [SS92] M. V. Sidelnikov and O. S. Shestakov. On Insecurity of Cryptosystems Based on Generalized Reed-Solomon Codes. *Discrete Math. Appl.*, 2:439–444, 01 1992.
- [The19] The Sage Developers. *SageMath, the Sage Mathematics Software System*, 2019. <https://www.sagemath.org>.

- [Wan16] Yongge Wang. Quantum resistant random linear code based public key encryption scheme RLCE. In *IEEE International Symposium on Information Theory, ISIT 2016, Barcelona, Spain, July 10-15, 2016*, pages 2519–2523. IEEE, 2016.
- [Wie06] Christian Wieschebrink. An Attack on a Modified Niederreiter Encryption Scheme. In *Public Key Cryptography - PKC 2006*, pages 14–26, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.
- [Wie10] Christian Wieschebrink. Cryptanalysis of the Niederreiter Public Key Scheme Based on GRS Subcodes. In Nicolas Sendrier, editor, *Post-Quantum Cryptography*, pages 61–72, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.