# Cryptanalysis of a System Based on Twisted Reed–Solomon Codes

Julien Lavauzelle* Julian Renner†

April 26, 2019

**Abstract**

It was recently proved that twisted Reed–Solomon codes represent a family of codes which contain a large amount of MDS codes, non-equivalent to Reed–Solomon codes. As a consequence, they were proposed as an alternative to Goppa codes for the McEliece cryptosystem, resulting to a potential reduction of key sizes.

In this paper, an efficient key-recovery attack is given on this variant of the McEliece cryptosystem. The algorithm is based on the recovery of the structure of subfield subcodes of twisted Reed–Solomon codes, and it always succeeds. Its correctness is proved, and it is shown that the attack breaks the system for all practical parameters in $O(n^4)$ field operations. A practical implementation is also provided and retrieves a valid private key from the public key within just a few minutes, for parameters claiming a security level of 128 bits.

We also discuss a potential repair of the scheme and an application of the attack to GPT cryptosystems using twisted Gabidulin codes.

## 1 Introduction

In the last years, systems based on the hardness of decoding in a generic code have gained large attention since they are potentially resistant to quantum computer attacks. The first code-based cryptosystem was proposed by McEliece in 1978 [16]. It is based on binary Goppa codes and is still considered to be secure.

The main drawback of the original McEliece system is its large public key. To overcome this drawback, many code classes have been proposed to replace Goppa codes, but most of them were subject to algebraic attacks. For instance, generalised Reed–Solomon (GRS) codes were proposed in 1986 by Niederreiter [18], but Sidelnikov and Shestakov mounted a very efficient attack to recover an alternative secret key [24]. Wieschebrink proved that also random subcodes of GRS codes — proposed in [7] — cannot be used due to their vulnerability against the *code squaring* attack [27]. Further instances and cryptanalyses of algebraic code-based schemes can be found in [6, 8, 11, 15, 17, 23].

One of the recent alternative classes emerged from twisted Reed–Solomon codes [5]. Beelen *et al.* analysed the structural properties of a specific subfamily of twisted Reed–Solomon codes in [4]. In their work, they proved that none of the codes they consider is a generalised Reed–Solomon code and thus the attack by Sidelnikov and Shestakov [24] cannot be applied to their system. Further, they showed that shortenings of these codes

*Institut de Recherche Mathématique de Rennes (IRMAR), Université de Rennes 1, France. Email: julien.lavauzelle@univ-rennes1.fr

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

up to two positions have maximal Schur square dimension [21], meaning that the proposed system is impervious to the attack presented by Couvreur *et al.* in [9]. Additionally, the authors conjecture that their proposed system is not vulnerable to the algorithms proposed by Wieschebrink in [26, 27]. As a result of the mentioned structural properties, specific subfamilies of twisted Reed–Solomon codes seem to be interesting for code-based cryptography. In [4], the authors propose an explicit subfamily of twisted Reed–Solomin codes and sets of parameters that provide 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 mount an attack on the twisted Reed–Solomon code-based cryptosystem given in [4]. Since it does not seem straightforward to directly retrieve the structure of the proposed codes, our idea is to first recover the structure of the *subfield subcodes* of twisted Reed–Solomon codes, which then in turn reveal the structure of the supercodes. We show that for all practical parameters, 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. We implemented the attack in the computer-algebra system SageMath [25] and although the implementation is not optimized, it determines a valid private key for the parameters proposed by the designers in approximately two minutes (a link to the mentioned implementation is provided in the paper). Additionally, we discuss a potential application of the proposed attack to the rank-metric version of the considered system [22].

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

# 2 Preliminaries

## 2.1 Notation

Let $q$ be a power of a prime and let $\mathbb{F}_q$ denote the finite field of order $q$. We use $\mathbb{F}_q^{m\times n}$ to denote the set of $m \times n$ matrices over $\mathbb{F}_q$ and $\mathbb{F}_q^n = \mathbb{F}_q^{1\times n}$ for the set of row vectors of length $n$ over $\mathbb{F}_q$. Rows and columns of $m \times n$-matrices are indexed by $1 \leq i \leq m$ and $1 \leq j \leq n$, where $A_{i,j}$ is the element in the $i$-th row and $j$-th column of the matrix $\boldsymbol{A} \in \mathbb{F}_q^{m\times n}$.

For a field extension $\mathbb{F}_q \subseteq \mathbb{F}$, the $\mathbb{F}$-row space of a matrix $\boldsymbol{A} \in \mathbb{F}_q^{m\times n}$ is the $\mathbb{F}$-vector space spanned by its rows, i.e.,

$$\mathcal{R}_{\mathbb{F}}\big(\boldsymbol{A}\big) := \left\{ \sum_{i=1}^{m} a_i \left(A_{i,1}, \ldots, A_{i,n}\right) \,:\, a_i \in \mathbb{F} \right\} \subseteq \mathbb{F}^n.$$

We denote the component-wise product of $\boldsymbol{a} \in \mathbb{F}_q^n$ and $\boldsymbol{b} \in \mathbb{F}_q^n$ by

$$\boldsymbol{a} \star \boldsymbol{b} := (a_1 b_1, \ldots, a_n b_n) \in \mathbb{F}_q^n.$$

Further, given a linear code $\mathcal{C} \subseteq \mathbb{F}_q^n$, we define its square as

$$\mathcal{C}^2 := \mathcal{R}_{\mathbb{F}_q}\big(\{\boldsymbol{a} \star \boldsymbol{b} : \boldsymbol{a}, \boldsymbol{b} \in \mathcal{C}\}\big)\,.$$

The set of all univariate polynomials over a field $\mathbb{F}$ is denoted by $\mathbb{F}[x]$. Let us now fix some $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n) \in \mathbb{F}_q^n$. We define the evaluation map $\mathrm{ev}_{\boldsymbol{\alpha}}$ as

$$\begin{aligned} \mathrm{ev}_{\boldsymbol{\alpha}} : \mathbb{F}_q[x] &\to \mathbb{F}_q^n \\ f &\mapsto (f(\alpha_1), f(\alpha_2), \ldots, f(\alpha_n)). \end{aligned}$$

Finally, if $\mathcal{I}$ and $\mathcal{J}$ are two finite subsets of integers, then we define

$$\mathcal{I} \oplus \mathcal{J} := \{a + b \,:\, a \in \mathcal{I},\ b \in \mathcal{J}\}.$$

## 2.2 Twisted Reed–Solomon Codes

**Definition 1** (Reed–Solomon Code)**.** *Let $n, k \in \mathbb{N}$ with $k \leq n \leq q$, the elements $\alpha_1, \ldots, \alpha_n \in \mathbb{F}_q$ be distinct and $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)$. The* Reed–Solomon (RS) code *of length $n$ and dimension $k$ is defined by*

$$\mathcal{C}_{\boldsymbol{\alpha}}[n, k]_{\mathbb{F}_q} := \left\{ \mathrm{ev}_{\boldsymbol{\alpha}}(f) \,:\, f \in \left\{ \sum_{i=0}^{k-1} f_i x^i : f_i \in \mathbb{F}_q \right\} \right\}.$$

*The entries in $\boldsymbol{\alpha}$ are called locators of the RS code.*

Reed–Solomon codes are maximum-distance separable (MDS) codes, i.e., they reach the so-called Singleton bound $d \leq n-k+1$, where $d$ denotes the (Hamming) minimum distance of the code. Twisted Reed–Solomon codes were recently proposed as a generalisation of Reed–Solomon codes.

**Definition 2** (Twisted Reed–Solomon Code, [5])**.** *Let $n, k, \ell \in \mathbb{N}$ with $k < n$ and $\ell \leq n-k$. Further, denote the hook vector by $\boldsymbol{h} \in \{0, \ldots, k-1\}^\ell$ with distinct $h_i$, the twist vector by $\boldsymbol{t} \in \{1, \ldots, n-k\}^\ell$ with distinct $t_i$, and $\boldsymbol{\eta} \in (\mathbb{F}_q \setminus \{0\})^\ell$. The set of twisted polynomials over $\mathbb{F}_q$ is defined by*

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

*Let $\alpha_1, \ldots, \alpha_n \in \mathbb{F}_q$ be distinct and $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)$. The* $[\boldsymbol{\alpha}, \boldsymbol{t}, \boldsymbol{h}, \boldsymbol{\eta}]$-twisted Reed–Solomon (RS) code *of length $n$ and dimension $k$ is defined by*

$$\mathcal{C}_{\boldsymbol{\alpha},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}[k, n] := \left\{ \mathrm{ev}_{\boldsymbol{\alpha}}(f) \,:\, f \in \mathcal{P}^{n,k}_{\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}} \right\}.$$

*The elements $\alpha_1, \ldots, \alpha_n$ are called locators of the twisted RS code.*

According to Definition 2, a generator matrix of $\mathcal{C}_{\boldsymbol{\alpha},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}[k,n]$ with $h_1 < h_2 < \cdots < h_\ell$ is given by

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

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

In [4], the authors show that by constructing a twisted RS code according to Definition 2, one does not necessarily obtain an MDS code. However, they provide a method to obtain twisted RS codes that are MDS, cf. Theorem 1.

**Theorem 1** (Explicit MDS twisted RS codes [4])**.** *Let $q_0$ be a prime power, and $1 = s_0 < \ldots < s_\ell \in \mathbb{Z}_{>0}$ be non-negative integers such that $\mathbb{F}_{q_0^{s_0}} \subset \mathbb{F}_{q_0^{s_1}} \subset \ldots \subset \mathbb{F}_{q_0^{s_\ell}} = \mathbb{F}_q$ is a chain of subfields. Let $k < n \leq q_0$, the elements $\alpha_1, \ldots, \alpha_n \in \mathbb{F}_{q_0}$ be distinct, and let $\boldsymbol{t}$, $\boldsymbol{h}$ and $\boldsymbol{\eta}$ be chosen as in Definition 2 and such that $\eta_i \in \mathbb{F}_{q_0^{s_i}} \setminus \mathbb{F}_{q_0^{s_{i-1}}}$ for $i = 1, \ldots, \ell$. Then $\mathcal{C}_{\boldsymbol{\alpha},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}[k,n]$ is MDS.*

A decoding algorithm for twisted RS codes is also proposed in [4]. Given a corrupted codeword $\boldsymbol{r}$, the strategy is to guess $\ell$ elements $g_1, \ldots, g_\ell \in \mathbb{F}_q$ and then decode $\boldsymbol{r} - \mathrm{ev}_{\boldsymbol{\alpha}}(\sum_{i=1}^{\ell} g_i \eta_i X^{t_i+k-1})$ in the Reed–Solomon code $\mathcal{C}_{\boldsymbol{\alpha}}[n,k]_{\mathbb{F}_q}$. This approach succeeds if $g_i = f_{h_i}$ and thus, has a worst case complexity of $O(q^\ell n \log^2 n \log\log n)$. Notice that $q = \Omega(q_0^{2^\ell})$, and thus this decoding algorithm is only practical for a tiny number of twists.

In the following lemma, we show a property of twisted RS codes that is important for the attack proposed in this paper.

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

$$\mathcal{C}_{\boldsymbol{\alpha},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}[k,n] = \mathcal{C}_{\hat{\boldsymbol{\alpha}},\boldsymbol{t},\boldsymbol{h},\hat{\boldsymbol{\eta}}}[k,n],$$

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

*Proof.* Let $\mathrm{ev}_{\hat{\boldsymbol{\alpha}}}(f) \in \mathcal{C}_{\hat{\boldsymbol{\alpha}},\boldsymbol{t},\boldsymbol{h},\hat{\boldsymbol{\eta}}}[k,n]$, 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 $g(x) \in \mathcal{P}^{n,k}_{\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}$. Hence by definition $\mathrm{ev}_{\hat{\boldsymbol{\alpha}}}(f) \in \mathcal{C}_{\boldsymbol{\alpha},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}[k,n]$, and it follows that $\mathcal{C}_{\boldsymbol{\alpha},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}[k,n] \subseteq \mathcal{C}_{\hat{\boldsymbol{\alpha}},\boldsymbol{t},\boldsymbol{h},\hat{\boldsymbol{\eta}}}[k,n]$. The proof on the converse inclusion is similar since $a$ is non-zero. □

# 3 The Twisted RS Code Based McEliece Cryptosystem

In this section we describe the system proposed in [4].

## 3.1 Setup

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

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

Further, set $q_i := q_{i-1}^2 = q_0^{2^i}$ for $i = 1, \ldots, \ell$, such that

$$\mathbb{F}_{q_0} \subset \mathbb{F}_{q_1} \subset \ldots \subset \mathbb{F}_{q_\ell} = \mathbb{F}_q$$

is a chain of subfields. Finally, set $t_i = (i+1)(r-2) - k + 2$ and $h_i = r - 1 + i$ for $i = 1, \ldots, \ell$, where $r := \lceil \frac{n+1}{\ell+2} \rceil + 2$.

Integers $q_0$, $n$, $k$, $\ell$, and vectors $\boldsymbol{t}$, $\boldsymbol{h}$ satisfying the above conditions are referred to as *valid parameters* of the cryptosystem [4]. They are public parameters of the cryptosystem.

## 3.2 Key Generation

Given valid parameters $q_0$, $n$, $k$, $\ell$, $\boldsymbol{t}$ and $\boldsymbol{h}$:

1. Choose $\boldsymbol{\alpha} \in \mathbb{F}_{q_0}^n$ at random such that the entries of $\boldsymbol{\alpha}$ are 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 $1 \leq i \leq \ell$.
3. Choose $\boldsymbol{S} \in \mathbb{F}_q^{k \times k}$ at random and full rank.
4. Compute the public key $\boldsymbol{G}_{\text{pub}} = \boldsymbol{S}\boldsymbol{G}_{\boldsymbol{\alpha},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}} \in \mathbb{F}_q^{k \times n}$, where $\boldsymbol{G}_{\boldsymbol{\alpha},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}$ is the generator matrix of $\mathcal{C}_{\boldsymbol{\alpha},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}[k,n]$ described in Section 2.2.

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

## 3.3 Encryption

Given a plaintext $\boldsymbol{m} \in \mathbb{F}_q^k$ and a public key $\boldsymbol{G}_{\text{pub}}$:

1. Choose $\boldsymbol{e} \in \mathbb{F}_q^n$ at random with Hamming weight $w_{\text{H}}(\boldsymbol{e}) = \lfloor \frac{n-k}{2} \rfloor$.
2. Compute the ciphertext
$$\boldsymbol{y} = \boldsymbol{m}\boldsymbol{G}_{\text{pub}} + \boldsymbol{e} \in \mathbb{F}_q^n.$$

## 3.4 Decryption

Given a ciphertext $\boldsymbol{y} \in \mathbb{F}_q^n$ and the private key $(\boldsymbol{S}, \boldsymbol{\alpha}, \boldsymbol{\eta})$:

1. Decode $\boldsymbol{y}$ in $\mathcal{C}_{\boldsymbol{\alpha},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}[k,n]$ to $\tilde{\boldsymbol{m}} = \boldsymbol{m}\boldsymbol{S} \in \mathbb{F}_q^k$ using the decoding algorithm given in [4].
2. Compute the plaintext $\boldsymbol{m} = \tilde{\boldsymbol{m}}\boldsymbol{S}^{-1}$.

### 3.5 Proposed Parameters

In [4], the parameters $n = 255$, $k = 117$, $\ell = 1$ and $q_0 = 2^8$ are proposed for a security level $\geq 100$ bits. There are two main reasons for choosing a small number of twists. On the one hand, the proposed decoding algorithm has a complexity of $q^\ell = q_0^{\ell 2^\ell}$ times $O(n \log^2 n \log\log n)$ and thus increases doubly exponentially with the number of twists. On the other hand, the field size and thus the key sizes also scale exponentially as the number of twists.

## 4 An Efficient Key-Recovery Attack Using Subfield Subcodes

In this section, we propose an efficient key-recovery algorithm for the cryptosystem and parameters proposed in [4]. The algorithm first determines a linear transformation of the secret locators $\boldsymbol{\alpha}$ 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 $(\hat{\boldsymbol{S}}, \hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\eta}})$ such that $\hat{\boldsymbol{S}}\boldsymbol{G}_{\hat{\boldsymbol{\alpha}},\boldsymbol{t},\boldsymbol{h},\hat{\boldsymbol{\eta}}} = \boldsymbol{G}_{\text{pub}}$. As shown in Section 2.2, $(\hat{\boldsymbol{S}}, \hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\eta}})$ is a valid private key that can be used in the decryption algorithm (Section 3.4).

### 4.1 Derivation of the 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 $\mathcal{C}_{\text{pub}}$ spanned by the public generator matrix $\boldsymbol{G}_{\text{pub}}$. We first state a technical lemma.

**Lemma 3.** *Let $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n) \in \mathbb{F}_{q_0}^n$ with distinct $\alpha_i$, and $P \in \mathbb{F}_q[x]$ where $\mathbb{F}_q$ is an extension of $\mathbb{F}_{q_0}$. Assume that $\deg(P) < n$. Then,*

$$\text{ev}_{\boldsymbol{\alpha}}(P) \in \mathbb{F}_{q_0}^n \iff P \in \mathbb{F}_{q_0}[x]\,.$$

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

Let us now define $\mathcal{I} := \{0, 1, \ldots, k-1\} \setminus \{h_1, \ldots, h_\ell\}$ as the set of exponents of monomials which are not twisted.

**Theorem 4.** *Let $\boldsymbol{G}_{\text{pub}}$ be chosen as described in Section 3 and $\mathcal{C}_{\text{pub}} = \mathcal{R}_{\mathbb{F}_q}(\boldsymbol{G}_{\text{pub}})$. Then,*

$$\mathcal{C}_{\text{pub}} \cap \mathbb{F}_{q_0}^n = \{\text{ev}_{\boldsymbol{\alpha}}(f) \,:\, f \in \mathcal{F}\},$$

*where*

$$\mathcal{F} := \left\{ \sum_{i \in \mathcal{I}} f_i x^i : f_i \in \mathbb{F}_{q_0} \right\} \subseteq \mathbb{F}_{q_0}[x].$$

*Proof.* First, it is clear that $\{\text{ev}_{\boldsymbol{\alpha}}(f) \,:\, f \in \mathcal{F}\} \subseteq \mathcal{C}_{\text{pub}} \cap \mathbb{F}_{q_0}^n$. Indeed, we obviously have $\text{ev}_{\boldsymbol{\alpha}}(f) \in \mathcal{C}_{\boldsymbol{\alpha},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}[k, n] = \mathcal{C}_{\text{pub}}$ for every $f \in \mathcal{F}$, and since $\boldsymbol{\alpha}$ is a vector over $\mathbb{F}_{q_0}$, we also get $\text{ev}_{\boldsymbol{\alpha}}(f) \in \mathbb{F}_{q_0}^n$.

Let us now prove that $\mathcal{C}_{\text{pub}} \cap \mathbb{F}_{q_0}^n \subseteq \{\text{ev}_{\boldsymbol{\alpha}}(f) \,:\, f \in \mathcal{F}\}$. Let $\boldsymbol{c} = \text{ev}_{\boldsymbol{\alpha}}(f) \in \mathcal{C}_{\text{pub}} \cap \mathbb{F}_{q_0}^n$, where $f \in \mathcal{P}_{\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}^{n,k}$. Since $\deg(f) < n$, Lemma 3 implies $f \in \mathbb{F}_{q_0}[x]$. It remains to notice that $\mathcal{F} = \mathbb{F}_{q_0}[x] \cap \mathcal{P}_{\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}^{n,k}$.

□

We observe by Theorem 4 that the subfield subcode $\mathcal{C}_{\text{sub}} := \mathcal{C}_{\text{pub}} \cap \mathbb{F}_{q_0}^n$ of the public code is a strict subcode of a Reed–Solomon code, since the evaluated polynomials do not have monomials of degree $h_1, \ldots, h_\ell$. Thus, one cannot directly use the Sidelnikov–Shestakov attack [24] on $\mathcal{C}_{\text{sub}}$. In 2006, Wieschebrink mounted an attack on cryptosystems based on random subcodes of Reed–Solomon codes [27]. The author's idea is that, with very high probability over the chosen subcode $\mathcal{C}'$, the square code $\mathcal{C}'^2$ is a Reed–Solomon code. Sidelnikov–Shestakov attack can then be used on $\mathcal{C}'^2$ to recover the private parameters.

In the following, we prove that for most valid parameters of [4], and for *every practical ones*, the square code $\mathcal{C}_{\text{sub}}^2$ is a Reed–Solomon code subject to Sidelnikov–Shestakov attack.

**Theorem 5.** *Let $q_0$, $n$, $k$, $\ell$, $\boldsymbol{t}$ and $\boldsymbol{h}$ be valid parameters, and assume that $\ell \leq \frac{1}{2}(\sqrt{n}-3)$. Let $\mathcal{C}_{\text{sub}} = \mathcal{C}_{\boldsymbol{\alpha},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}[k,n] \cap \mathbb{F}_{q_0}^n$. Then,*

$$\mathcal{C}_{\text{sub}}^2 = \mathcal{C}_{\boldsymbol{\alpha}}[2k-1, n]_{\mathbb{F}_{q_0}}.$$

*Proof.* We use the notation of Theorem 4. Notice that for valid parameters, we have $2k-1 \leq n-3$ and $\mathcal{I} = \{0, \ldots, r-1\} \cup \{r+\ell, \ldots, k-1\}$ where $r = \lceil \frac{n+1}{\ell+2} \rceil + 2$. Theorem 4 implies that

$$\mathcal{C}_{\text{sub}}^2 = \mathcal{R}_{\mathbb{F}_{q_0}}\big(\{\text{ev}_{\boldsymbol{\alpha}}(h) \,:\, h \in \mathcal{L}\}\big)\,,$$

where $\mathcal{L} = \{g_1 g_2 \,:\, g_1, g_2 \in \mathcal{F}\}$ and $\mathcal{F} = \{\sum_{i \in \mathcal{I}} f_i x^i \,:\, f_i \in \mathbb{F}_{q_0}\}$. As a consequence, the claimed result holds if and only if $\mathcal{I} \oplus \mathcal{I} = \{0, \ldots, 2k-2\}$.

It is clear that $\mathcal{I} \oplus \mathcal{I}$ contains the subset

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

On the one hand, one can easily check that if $\ell \leq r-1$, then $\{r, \ldots, r+\ell-1\} \subset \mathcal{I} \oplus \mathcal{I}$. Moreover, $\ell \leq r-1$ is always fulfilled by valid parameters since $\ell < \sqrt{n}-3$ and $r > \sqrt{n}+3$. On the other hand, if we assume $\ell \leq \frac{1}{2}(\sqrt{n}-3)$, then we can prove that $\ell \leq \frac{k-r}{2}$, which a sufficient condition for having $\{k+r-1, \ldots, k+r+\ell-2\} \subset \mathcal{I} \oplus \mathcal{I}$. □

**Theorem 6.** *Let $\mathcal{C}_{\boldsymbol{\alpha}}[n,k]_{\mathbb{F}_{q_0}}$ be a Reed–Solomon code with locators $\alpha_1, \ldots, \alpha_n \in \mathbb{F}_{q_0}$. Given any generator matrix of $\mathcal{C}_{\boldsymbol{\alpha}}[n,k]_{\mathbb{F}_{q_0}}$, the algorithm given by Sidelnikov and Shestakov [24] determines, in time $O(n^4)$, a vector $\boldsymbol{\alpha}' \in \mathbb{F}_{q_0}^n$ such that*

$$\mathcal{C}_{\boldsymbol{\alpha}}[n,k]_{\mathbb{F}_{q_0}} = \mathcal{C}_{\boldsymbol{\alpha}'}[n,k]_{\mathbb{F}_{q_0}}.$$

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

*Proof.* See [24]. □

It follows that by applying the Sidelnikov–Shestakov algorithm to $\mathcal{C}_{\text{sub}}^2$, we obtain a vector $\boldsymbol{\alpha}' \in \mathbb{F}_{q_0}$ which is an affine transformation of the secret locators, i.e., $\boldsymbol{\alpha}' = a\boldsymbol{\alpha} + b\mathbf{1}$ for some $a \in \mathbb{F}_{q_0} \setminus \{0\}$ and $b \in \mathbb{F}_{q_0}$.

#### 4.1.2 Second Step: Recovery of a Linear Transformation of the Secret Locators

Lemma 2 only ensures that $\mathcal{C}_{\boldsymbol{\alpha},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}[k,n] = \mathcal{C}_{\hat{\boldsymbol{\alpha}},\boldsymbol{t},\boldsymbol{h},\hat{\boldsymbol{\eta}}}[k,n]$ if $\hat{\boldsymbol{\alpha}} = a\boldsymbol{\alpha}$ for some non-zero $a \in \mathbb{F}_{q_0}$. Therefore, given $\boldsymbol{\alpha}' = a\boldsymbol{\alpha} + b\mathbf{1}$, it remains to search exhaustively for $b$ such that

$\boldsymbol{\alpha}' - b\mathbf{1} = a\boldsymbol{\alpha}$. This exhaustive search can be proceeded as follows: given $\boldsymbol{\alpha}'$ and $b \in \mathbb{F}_{q_0}$, compute the code

$$\mathcal{A}_b := \mathcal{R}_{\mathbb{F}_q}\big(\{\mathrm{ev}_{\boldsymbol{\alpha}'-b\mathbf{1}}(x^i) : i \in \mathcal{I}\}\big)\,.$$

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

### 4.1.3 Third Step: Recovery of a Valid Pair $(\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\eta}})$

Previous steps provide a tuple $\hat{\boldsymbol{\alpha}} \in \mathbb{F}_{q_0}^n$ which can be used as locators for the twisted RS code. To determine a vector $\hat{\boldsymbol{\eta}}$ such that $\mathcal{C}_{\boldsymbol{\alpha},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}[k,n] = \mathcal{C}_{\hat{\boldsymbol{\alpha}},\boldsymbol{t},\boldsymbol{h},\hat{\boldsymbol{\eta}}}[k,n]$, we use the following Lemma 7.

**Lemma 7.** *Let $\boldsymbol{G}_{\mathrm{pub}} = \boldsymbol{S}\boldsymbol{G}_{\boldsymbol{\alpha},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}$ be chosen as described in Section 3, $\hat{\boldsymbol{\alpha}} = a\boldsymbol{\alpha}$ for some $a \in \mathbb{F}_{q_0} \setminus \{0\}$ and $g_i(x)$ denote the unique polynomial that interpolates the pairs $(\hat{\boldsymbol{\alpha}}_1, \boldsymbol{G}_{\mathrm{pub}_{i,1}}), \ldots, (\hat{\boldsymbol{\alpha}}_n, \boldsymbol{G}_{\mathrm{pub}_{i,n}})$. Further, let $I_1, \ldots, I_\ell \in \{1, \ldots, k\}$ be such that $S_{I_j,h_j+1} \neq 0$ and*

$$\hat{\eta}_j = \frac{g_{I_j,k+t_j}}{g_{I_j,h_j+1}}, \quad j = 1, \ldots, \ell,$$

*where $g_{I_j,1}, \ldots, g_{I_j,n}$ are the coefficients of $g_{I_j}(x)$. Then, $\mathcal{C}_{\boldsymbol{\alpha},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}[k,n] = \mathcal{C}_{\hat{\boldsymbol{\alpha}},\boldsymbol{t},\boldsymbol{h},\hat{\boldsymbol{\eta}}}[k,n]$.*

*Proof.* By definition,

$$\begin{aligned}
\boldsymbol{G}_{\mathrm{pub}_{i,j}} &= \sum_{s=1}^{k} S_{i,s}\boldsymbol{G}_{\boldsymbol{\alpha},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}_{s,j}} \\
&= \sum_{s=1}^{k} S_{i,s}\alpha_j^{s-1} + \sum_{u=1}^{\ell} S_{i,h_u+1}\eta_u\alpha_j^{k-1+t_u} \\
&= \sum_{s=1}^{k} S_{i,s}a^{-s+1}\hat{\alpha}_j^{s-1} + \sum_{u=1}^{\ell} S_{i,h_u+1}\eta_u a^{-(k-1+t_u)}\hat{\alpha}_j^{k-1+t_u}.
\end{aligned}$$

By interpolating $(\hat{\boldsymbol{\alpha}}_1, \boldsymbol{G}_{\mathrm{pub}_{i,1}}), \ldots, (\hat{\boldsymbol{\alpha}}_n, \boldsymbol{G}_{\mathrm{pub}_{i,n}})$, one obtains a unique polynomial $g_i(x)$ with coefficients

$$g_{i,s} = \begin{cases} S_{i,s}a^{-s+1} & \text{if } s \in \{1, \ldots, k\} \\ S_{i,h_u+1}\eta_u a^{-(k-1+t_u)} & \text{if } s = k + t_u,\ u = 1 \ldots, \ell \\ 0 & \text{otherwise.} \end{cases}$$

If $S_{i,h_u+1} \neq 0$, then we get

$$\hat{\eta}_u = \eta_u a^{-(k-1+t_u-h_u)} = \frac{g_{i,k+t_u}}{g_{i,h_u+1}}.$$

□

### 4.1.4 Final Step: Recovery of an Alternative Private Key $(\hat{\boldsymbol{S}}, \hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\eta}})$

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

**Algorithm 1** Key-Recovery Attack

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

$\boldsymbol{G}_{\text{sub}} \leftarrow \texttt{SubfieldSubcode}(\boldsymbol{G}_{\text{pub}}) \in \mathbb{F}_{q_0}^{(k-\ell)\times n}$
$\boldsymbol{G}_{\text{sq}} \leftarrow \texttt{Square}(\boldsymbol{G}_{\text{sub}}) \in \mathbb{F}_{q_0}^{(2k-1)\times n}$
$\boldsymbol{\alpha}' \leftarrow \texttt{SidelShest}(\boldsymbol{G}_{\text{sq}}) \in \mathbb{F}_{q_0}^{n}$
$i \leftarrow 1 \in \mathbb{N}$
**do**
    $b \leftarrow \beta_i \in \mathbb{F}_{q_0}$
    $\hat{\boldsymbol{\alpha}} \leftarrow (\alpha'_1 - b, \ldots, \alpha'_n - b) \in \mathbb{F}_{q_0}^{n}$
    $\boldsymbol{G}' \leftarrow \texttt{GenSub}(\hat{\boldsymbol{\alpha}}) \in \mathbb{F}_{q_0}^{(k-\ell)\times n}$
    $i \leftarrow i + 1 \in \mathbb{N}$
**while** $\boldsymbol{G}'(\boldsymbol{G}_{\text{sub}}^{\perp})^{\top} \neq \boldsymbol{0}$
**for all** $j$ in $\{1, \ldots, \ell\}$ **do**
    $i \leftarrow 1 \in \mathbb{N}$
    **do**
        $\boldsymbol{g} \leftarrow \texttt{Interpolate}\big(\boldsymbol{\alpha}', (\boldsymbol{G}_{\text{pub}_{i,1}}, \ldots, \boldsymbol{G}_{\text{pub}_{i,n}})\big) \in \mathbb{F}_{q}^{n}$
        $i \leftarrow i + 1 \in \mathbb{N}$
    **while** $g_{h_j+1} = 0$
    $\hat{\eta}_j = \frac{g_{k-1+t_j}}{g_{h_j+1}} \in \mathbb{F}_q$
$\hat{\boldsymbol{G}}_{\text{TRS}} \leftarrow \texttt{GTRS}(\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\eta}}) \in \mathbb{F}_{q}^{k\times n}$
$\hat{\boldsymbol{S}} \leftarrow \hat{\boldsymbol{G}}_{\text{TRS}} \backslash \boldsymbol{G}_{\text{pub}} \in \mathbb{F}_{q}^{k\times k}$
**return** $\hat{\boldsymbol{S}}$, $\hat{\boldsymbol{\alpha}}$, $\hat{\boldsymbol{\eta}}$

### 4.2 Performance Analysis of the Attack

A pseudo algorithm describing the attack is given in Algorithm 1. Let us explain the notation we use there. We arbitrarily order $\mathbb{F}_{q_0} = \{\beta_1, \ldots, \beta_{q_0}\}$. By $\boldsymbol{A}^{\top}$ we denote the transpose of the matrix $\boldsymbol{A}$ and by $\boldsymbol{A}^{\perp}$ a matrix whose rows form a basis of the right kernel of $\boldsymbol{A}$. The reduced row echelon form of $\boldsymbol{A}$ is denoted by $\texttt{rref}(\boldsymbol{A})$. The function $\texttt{SubfieldSubcode} : \mathbb{F}_{q}^{k\times n} \to \mathbb{F}_{q_0}^{(k-\ell)\times n}$ maps a generator matrix of $\mathcal{C}_{\text{pub}}$ to a generator matrix of the subfield subcode of $\mathcal{C}_{\text{pub}}$, i.e., $\mathcal{R}_{\mathbb{F}_{q_0}}\big(\texttt{SubfieldSubcode}(\boldsymbol{G}_{\text{pub}})\big) = \mathcal{C}_{\text{pub}} \cap \mathbb{F}_{q_0}$. The function $\texttt{Square} : \mathbb{F}_{q_0}^{(k-\ell)\times n} \to \mathbb{F}_{q_0}^{(2k-1)\times n}$ maps a generator matrix of $\mathcal{C}_{\text{sub}}$ to a generator matrix of the code $\mathcal{C}_{\text{sub}}^2$. The interpolation function is defined as $\texttt{Interpolate} : \mathbb{F}_{q_0}^{n} \times \mathbb{F}_{q}^{n} \to$

$\mathbb{F}_q^n, (\boldsymbol{a}, \boldsymbol{b}) \mapsto \boldsymbol{g}$ such that $\sum_{j=1}^{n} g_j a_i^{j-1} = b_i$ for $i = 1, \ldots, n$. We define the function

$$\texttt{GenSub} : \mathbb{F}_{q_0}^n \to \mathbb{F}_{q_0}^{(k-\ell)\times n},$$
$$(a_1, \ldots, a_n) \mapsto \begin{pmatrix} 1 & \ldots & 1 \\ a_1 & \ldots & a_n \\ \vdots & \ddots & \vdots \\ a_1^{h_1-1} & \ldots & a_n^{h_1-1} \\ a_1^{h_1+1} & \ldots & a_n^{h_1+1} \\ \vdots & \ddots & \vdots \\ a_1^{h_\ell-1} & \ldots & a_n^{h_\ell-1} \\ a_1^{h_\ell+1} & \ldots & a_n^{h_\ell+1} \\ \vdots & \ddots & \vdots \\ a_1^{k-1} & \ldots & a_n^{k-1} \end{pmatrix}$$

and the function implementing Sidelnikov–Shestakov attack as $\texttt{SidelShest} : \mathbb{F}_{q_0}^{k\times n} \to \mathbb{F}_{q_0}^n$ such that if $\boldsymbol{G}$ is a generator matrix of a Reed–Solomon code $\mathcal{C}_{\boldsymbol{\alpha}}[n,k]_{\mathbb{F}_{q_0}}$, then

$$\mathcal{R}_{\mathbb{F}_{q_0}}(\boldsymbol{G}) = \mathcal{R}_{\mathbb{F}_{q_0}}\big(\texttt{GenSub}(\texttt{SidelShest}(\boldsymbol{G}))\big)\,.$$

The function $\texttt{GTRS} : \mathbb{F}_{q_0}^n \times \mathbb{F}_q^\ell \to \mathbb{F}_q^{k\times n}$ maps the vectors $\hat{\boldsymbol{\alpha}}$ and $\hat{\boldsymbol{\eta}}$ to the corresponding twisted RS generator matrix, i.e., $\texttt{GTRS}(\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\eta}}) = \boldsymbol{G}_{\hat{\boldsymbol{\alpha}},\boldsymbol{t},\boldsymbol{h},\hat{\boldsymbol{\eta}}}$. Further, if $\boldsymbol{A} \in \mathbb{F}_q^{k\times n}$ and $\boldsymbol{B} \in \mathbb{F}_q^{k\times n}$ have the same rowspace, then $\boldsymbol{D} = \boldsymbol{A}\backslash\boldsymbol{B}$ is a solution to $\boldsymbol{DA} = \boldsymbol{B}$.

Below we provide details on the complexity of the steps in Algorithm 1.

- Line 1: Computation of $\boldsymbol{G}_{\text{sub}} \in \mathbb{F}_{q_0}^{(k-\ell)\times n}$ 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: Computation of $\boldsymbol{G}_{\text{sq}} \in \mathbb{F}_{q_0}^{(2k-1)\times n}$ can be performed in time $O(n^4)$. Informally, one needs to find basis of the space generated by the family $\{\boldsymbol{g}_{i,j} := \boldsymbol{G}_{\text{sub}i} \star \boldsymbol{G}_{\text{sub}j}, 1 \le i, j \le \dim \mathcal{C}_{\text{sub}}\}$. 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 $\texttt{SidelShest}$ function on $\boldsymbol{G}_{\text{sq}} \in \mathbb{F}_{q_0}^{(2k-1)\times n}$ needs $O((2k-2)^4 + (2k-2)n) \subseteq O(n^4)$ operations in $\mathbb{F}_{q_0}$ [24].

- Line 4 to Line 10: In the worst case, the following computations have to be performed $q_0$ times. Computation of $\hat{\boldsymbol{\alpha}} \in \mathbb{F}_{q_0}^n$ needs $O(n)$ operations in $\mathbb{F}_{q_0}$, building $\boldsymbol{G}' \in \mathbb{F}_{q_0}^{(k-\ell)\times n}$ needs $O((k-\ell)n)$ operations in $\mathbb{F}_{q_0}$ and matrix multiplication of $\boldsymbol{G}'(\boldsymbol{G}_{\text{sub}}^\perp)^\top$ needs $O((k-\ell)(n-k+\ell)n) \subseteq O(n^3)$ operations in $\mathbb{F}_{q_0}$ ($\boldsymbol{G}_{\text{sub}}^\perp$ was already computed in Line 1). In total $O(q_0 n^3)$ operations in $\mathbb{F}_{q_0}$ are required.

- Line 11 to Line 17: In the worst case, $\ell \cdot k$ Lagrange interpolations have to be performed, which needs in total $O(\ell k n^2) \subseteq O(n^4)$ operations in $\mathbb{F}_q$.

- Line 18: Computation of $\hat{\boldsymbol{G}}_{\text{TRS}} \in \mathbb{F}_q^{k\times n}$ needs $O(kn) \subset O(n^2)$ operations in $\mathbb{F}_q$.

- Line 19: Computation of $\hat{\boldsymbol{S}} \in \mathbb{F}_q^{k\times k}$ by transformation of $\begin{pmatrix} \hat{\boldsymbol{G}}_{\text{TRS}}^\top & \boldsymbol{G}_{\text{pub}}^\top \end{pmatrix} \in \mathbb{F}_q^{n\times 2k}$ in reduced row echelon form needs $O(n^2(2k)) \subseteq O(n^3)$ operations in $\mathbb{F}_q$.

| $q_0$ | $n$ | $k$ | $l$ | $w_{\mathrm{H}}(\boldsymbol{e})$ | 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 1: Experimental results obtained by averaging several runtimes of Algorithm 1 on an Intel(R) Core(TM) i7-7600U CPU @ 2.80GHz. The line annotated with a star refers to parameters proposed by the designers of the system. The remaining security levels were computed according to formulae given in [4].

In practice, $\ell$ and $q_0 = q^{1/2^\ell}$ have to be chosen small (for instance, $\ell = 1$ and $q_0 = n + 1 = 2^8$ were proposed in [4]) for decryption efficiency and key size reduction. Hence, Algorithm 1 has a complexity in $O(n^4)$ and thus recovers a valid private key in polynomial time.

We implemented our attack in the computer-algebra system SageMath v8.7 [25], and we make it available online under `https://bitbucket.org/julianrenner/trs_attack`. Although our implementation is not optimized, we were able to obtain a valid private key for the proposed parameters within a few minutes, cf. Table 1.

# 5 Discussion and Open Questions

## 5.1 Repairing the Cryptosystem?

We notified the authors about our attack, and they validated the weaknesses of the cryptosystem as it is presented above. They also described a possible fix, where a modified version of the generator matrix is made public. The idea is to multiply the generator matrix $\boldsymbol{G}_{\mathrm{pub}}$ on the right by a diagonal matrix with non-zero entries $\boldsymbol{y} = (y_1, \ldots, y_n) \in (\mathbb{F}_q \setminus \{0\})^n$, such that the $\mathbb{F}_{q_0}$-subfield subcode of the vector space spanned by the rows of $\boldsymbol{G}_{\mathrm{pub}}$ is not contained in a Reed–Solomon 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 generalised Reed–Solomon code $\boldsymbol{y} \star \mathcal{C}_{\boldsymbol{\alpha}}$ is a so-called *alternant code* $\mathrm{Alt}(\boldsymbol{\alpha}, \boldsymbol{y}) \subseteq \mathbb{F}_{q_0}^n$ which also admit 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 $\mathrm{Alt}(\boldsymbol{\alpha}, \boldsymbol{y})$.

One can then notice that the parameters proposed by the authors are way below those considered as secure for alternant codes. For instance, BIG QUAKE [2] and Classic McEliece [10] (both are unbroken candidates for the NIST standardisation call on post-quantum cryptography) use alternant codes of length and dimension several thousands, while in the proposed parameters for twisted Reed–Solomon codes, we have $n = 255$ and $k = 117$ with a field size $q_0 = 2^8$. Algebraic attacks as developed in [11, 12] should then considered as potential threat. One can also mention the recent attack on the alternant code-based cryptosystem DAGS [1] performed by Barelli and Couvreur [3]. Informally, the authors manage to derive from the public code an alternant code with much smaller parameters, and the last step of the key recovery algorithm — which is exponential in the involved parameters — remains doable due to the small size of the derived alternant code.

Finally, a crucial point is that one can wonder about the possible benefit to consider codes whose security might be not 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 [22] was proposed a modified version of the previous system, based on a subfamily of twisted Gabidulin codes. The idea is to consider a variant of the GPT cryptosystem [14], 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 which could be analysed in a future work.

### 5.2.1 A Short Description of the System

The GPT cryptosystem can be viewed as an analogue of the McEliece cryptosystem, using rank-metric codes instead of codes in the Hamming metric. We refer to [20] for more details about rank-metric codes and variants of the GPT cryptosystem. Let us give a short overview of the latter.

Let $\mathbb{F}_p \subset \mathbb{F}_{q_0}$ 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 $\boldsymbol{G} \in \mathbb{F}_q^{n-t}$ for a code $\mathcal{C}$ randomly chosen in $\Gamma$. Then she computes a public key $\boldsymbol{G}_{\text{pub}} = \boldsymbol{S}[\boldsymbol{X}|\boldsymbol{G}]\boldsymbol{P}$, where matrices $\boldsymbol{S} \in \mathbb{F}_q^{k\times k}$ of full-rank, $\boldsymbol{X} \in \mathbb{F}_q^{k\times t}$ of rank $s \leq t$, and $\boldsymbol{P} \in \mathbb{F}_p^{n\times n}$ of full-rank are chosen randomly and kept secret.
- *Encryption:* given a plaintext $\boldsymbol{m} \in \mathbb{F}_q^k$, Bob computes the ciphertext $\boldsymbol{y} = \boldsymbol{m}\boldsymbol{G}_{\text{pub}} + \boldsymbol{e}$, where $\boldsymbol{e} \in \mathbb{F}_q^n$ is a random error with small rank over $\mathbb{F}_p$ (such that it can be decoded in $\mathcal{C}$).
- *Decryption:* Alice decodes the last $n-t$ coordinates of $\boldsymbol{y}\boldsymbol{P}^{-1}$ in the code $\mathcal{C}$ and retrieves $\boldsymbol{m}$.

In most variants of the GPT cryptosystem, $\Gamma$ is a (sub-)family of Gabidulin codes $\mathcal{G}_{\boldsymbol{\alpha}}[n-t,k]_{\mathbb{F}_q} = \{\text{ev}_{\boldsymbol{\alpha}}(f) \,:\, f \in \{\sum_{i=0}^{k-1} f_i x^{[i]} : f_i \in \mathbb{F}_q\}\}$, where $x^{[i]} := x^{p^i}$, firstly defined in [13]. In [22], the authors proposed to define $\Gamma$ as the subfamily of twisted Gabidulin codes

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

where $\eta_i$ are chosen in the chain of subfields $\mathbb{F}_{q_0} \subset \mathbb{F}_{q_1} \subset \ldots \subset \mathbb{F}_{q_\ell} = \mathbb{F}_q$, and $(\alpha_1, \ldots, \alpha_{n-t}) \in \mathbb{F}_{q_0}^{n-t}$ are $\mathbb{F}_p$-linearly independent, similarly to the case of twisted Reed–Solomon codes.

### 5.2.2 Potential Weakness

Our claim is that the code $\mathcal{C}_{\text{pub}}$ generated by $\boldsymbol{G}_{\text{pub}}$ also admit structured subfield subcodes which could be used to attack the system. Indeed, one can prove that the last $n-t$ coordinates of $(\mathcal{C}_{\text{pub}} \cap \mathbb{F}_{q_0}^n)\boldsymbol{P}^{-1}$ form a subcode of the Gabidulin code $\mathcal{G}_{\boldsymbol{\alpha}}[n-t,k]_{\mathbb{F}_{q_0}} \subseteq \mathbb{F}_{q_0}^{n-t}$ of rather small codimension. Applying variants of Overbeck's attacks — e.g. in [19] — 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.

In fact, we observe in simulations that if $\lambda_f(\mathcal{C}_{\text{pub}} \cap \mathbb{F}_{q_0}^n)$ has dimension $n-1$, where $f = n-k-t-1$ and

$$\lambda_f(\mathcal{C}) := \mathcal{R}_{\mathbb{F}_{q_0}}\left(\begin{pmatrix} \boldsymbol{G} \\ \boldsymbol{G}^{[1]} \\ \vdots \\ \boldsymbol{G}^{[f]} \end{pmatrix}\right)$$

for $\boldsymbol{G}$ being a generator matrix of $\mathcal{C}$, one recovers an $\mathbb{F}_p$-linear transformation $\hat{\boldsymbol{\alpha}}$ of $\boldsymbol{\alpha}$, as well as a full-rank matrix $\hat{\boldsymbol{P}} \in \mathbb{F}_p^{n\times n}$, by simply applying the algorithm shown in [20, Algorithm 3.5.1] to a generator matrix of $\mathcal{C}_{\text{pub}} \cap \mathbb{F}_{q_0}^n$. Then, the coefficients $\hat{\boldsymbol{\eta}}$ are determined by interpolating the last $n-t$ positions of $\boldsymbol{G}_{\text{pub}}\hat{\boldsymbol{P}}^{-1}$ with $p$-polynomials of $p$-degree smaller than $n$, similar to Section 4.1.3. Finally, one chooses $\hat{\boldsymbol{S}}$ such that

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

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

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

Since a precise analysis of the potential weakness of system proposed in [22] is out of the scope of this paper, we leave it as an open problem for future research.

# 6 Conclusion

In this paper, we have presented an efficient key-recovery attack on the McEliece cryptosystem based on a subfamily of twisted Reed–Solomon codes. The attack does not contradict the structural properties presented in [4], but recovers the structure of the *subfield subcode* of the used twisted Reed–Solomon code, which then in turn enables us to determine a description of the supercode.

We have proven that the attack retrieves a valid private key from the public key for all practical parameters in $O(n^4)$ field operations. This is confirmed by experimental results which indicate that one is able to retrieve a valid private key for a claimed security level of 128 bits within a few minutes by running a non-optimized SageMath implementation of the proposed algorithm on a general purpose processor. In addition, we have discussed the security of an attempt to repair the system and potential ways to adapt our attack to the rank-metric variant of the considered system.

Although we have shown that a variant of the McEliece cryptosystem based on the subfamily of twisted Reed–Solomon codes proposed in [4] is not secure, this does not imply that *any* subfamily of twisted Reed–Solomon codes is not suitable for code-based cryptography. In fact, twisted Reed–Solomon codes represent a very large family of codes, and it requires further research to determine if there could exist other subfamilies that can be used for the design of cryptosystem.


# 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 for fruitful discussions. We would further like to thank the authors of the proposed cryptosystem [4] for validating our attack and pointing out a possible repair of the system with respect to our attack.

## References

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] Peter Beelen, Sven Puchinger, and Johan Rosenkilde né Nielsen. Twisted Reed–Solomon Codes. In *IEEE Int. Symp. Inf. Theory (ISIT)*, 2017.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] Ernst M. Gabidulin. Theory of Codes with Maximum Rank Distance. *Probl. Inf. Transm.*, 21(1):3–16, 1985.

[14] 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.

[15] Heeralal Janwa and Oscar Moreno. McEliece Public Key Cryptosystems Using Algebraic-Geometric Codes. *Des. Codes Cryptogr.*, 8(3):293–307, 1996.

[16] Robert J. McEliece. A Public-Key Cryptosystem Based on Algebraic Coding Theory. *Coding Thv*, 4244:114–116, 1978.

[17] Lorenz Minder and Amin Shokrollahi. Cryptanalysis of the Sidelnikov Cryptosystem. In *Advances in Cryptology - EUROCRYPT 2007*, volume 4515, pages 347–360. Springer, 2007.

[18] Harald Niederreiter. Knapsack type cryptosystems and algebraic coding theory. *Probl. Control Inf. Theory*, 15, 01 1986.

[19] Raphael Overbeck. A New Structural Attack for GPT and Variants. *LNCS: MYCRYPT*, 3715:50–63, 2005.

[20] Raphael Overbeck. *Public Key Cryptography Based on Coding Theory.* PhD thesis, Darmstadt University of Technology, Germany, 2007.

[21] Sven Puchinger. *Construction and Decoding of Evaluation Codes in Hamming and Rank Metric.* PhD thesis, Ulm University, Germany, 2018.

[22] Sven Puchinger, Julian Renner, and Antonia Wachter-Zeh. Twisted Gabidulin Codes in the GPT Cryptosystem. In *Int. Workshop Alg. Combin. Coding Theory (ACCT)*, 2018.

[23] M. V. Sidelnikov. Public-key Cryptosystem Based on Binary Reed-Muller Codes. *Discrete Math. Appl.*, 4:191–208, 01 1994.

[24] 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.

[25] The Sage Developers. *SageMath, the Sage Mathematics Software System*, 2019. `https://www.sagemath.org`.

[26] 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.

[27] 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.