# Recovering Short Secret Keys of RLCE in Polynomial Time

Alain Couvreur[1], Matthieu Lequesne[2,3](✉), and Jean-Pierre Tillich[3]

[1] Inria and LIX, CNRS UMR 7161 École polytechnique,
91128 Palaiseau Cedex, France
alain.couvreur@lix.polytechnique.fr
[2] Sorbonne Université, UPMC Univ Paris 06, Paris, France
[3] Inria, Paris, France
{matthieu.lequesne,jean-pierre.tillich}@inria.fr

**Abstract.** We present a key recovery attack against Y. Wang's Random Linear Code Encryption (RLCE) scheme recently submitted to the NIST call for post-quantum cryptography. The public key of this code based encryption scheme is a generator matrix of a generalised Reed Solomon code whose columns are mixed in a certain manner with purely random columns. In this paper, we show that it is possible to recover the underlying structure when there are not enough random columns. The attack reposes on a distinguisher on the dimension of the square code. This process allows to recover the secret key for all the short key parameters proposed by the author in $O(n^5)$ operations. Our analysis explains also why RLCE long keys stay out of reach of our attack.

**Keywords:** Code based cryptography · McEliece scheme · RLCE · Distinguisher · Key recovery attack · Generalised Reed Solomon codes · Schur product of codes

## 1 Introduction

The McEliece encryption scheme dates back to the late 70's [14] and lies among the possible post-quantum alternatives to number theory based schemes using integer factorisation or discrete logarithm. However, the main drawback of McEliece's original scheme is the large size of its keys. Indeed, the classic instantiation of McEliece using binary Goppa codes requires public keys of several hundreds of kilobytes to assert a security of 128 bits. For example, the recent NIST submission *Classic McEliece* [4] proposes public keys of 1.1 to 1.3 megabytes to assert 256 bits security (with a classical computer).

To reduce the size of the keys, two general trends appear in the literature : the first one consists in considering codes with a non trivial automorphism group, the second one in using codes with a higher decoding capacity for encryption. In the last decade, the second trend led to many proposals involving generalised Reed Solomon (GRS) codes, which are well-known to have a large minimum distance

J. Ding and R. Steinwandt (Eds.): PQCrypto 2019, LNCS 11505, pp. 133–152, 2019.
https://doi.org/10.1007/978-3-030-25510-7_8

together with efficient decoding algorithms correcting up to half the minimum distance. On the other hand, the raw use of GRS codes has been proved to be insecure by Sidelnikov and Shestakov [15]. Subsequently, some variations have been proposed as a counter-measure of Sidelnikov and Shestakov's attack. Berger and Loidreau [3] suggested to replace a GRS code by a random subcode of small codimension, Wieschebrink [18] proposed to join random columns in a generator matrix of a GRS code and Baldi *et al.* [1] suggested to mask the structure of the code by right multiplying a generator matrix of a GRS code by the sum of a low rank matrix and a sparse matrix. It turns out that all of these proposals have been subject to efficient polynomial time attacks [8,11,19].

A more recent proposal by Yongge Wang [16] suggests another way of hiding the structure of GRS codes. The outline of Wang's construction is the following: start from a $k \times n$ generator matrix of a GRS code of length $n$ and dimension $k$ over a field $\mathbb{F}_q$, add $w$ additional random columns to the matrix, and mix the columns in a particular manner. The design of this scheme is detailed in Sect. 3.1. This approach entails a significant expansion of the public key size but may resist above-mentioned attacks such as distinguisher and filtration attacks [8,10]. This public key encryption primitive is the core of Wang's recent NIST submission "RLCE-KEM" [17].

*Our Contribution:* In the present article we give a polynomial time key recovery attack against RLCE. For an $[n, k]$ code with $w$ additional random columns, our attack breaks the system in $O(wk^2n^2)$ operations, when $w < n - k$. This allows us to break half the parameter sets proposed in [17].

## 2 Notation and Prerequisites

### 2.1 Generalised Reed Solomon Codes

**Notation 1.** *Let $q$ be a power of prime and $k$ a positive integer. We denote by $\mathbb{F}_q[X]_{<k}$ the vector space of polynomials over $\mathbb{F}_q$ whose degree is strictly bounded from above by $k$.*

**Definition 1 (Generalised Reed Solomon codes).** *Let $\boldsymbol{x} \in \mathbb{F}_q^n$ be a vector whose entries are pairwise distinct and $\boldsymbol{y} \in \mathbb{F}_q^n$ be a vector whose entries are all nonzero. The* generalised Reed Solomon (GRS) code with support $\boldsymbol{x}$ and multiplier $\boldsymbol{y}$ of dimension $k$ *is defined as*

$$\boldsymbol{GRS}_k(\boldsymbol{x}, \boldsymbol{y}) \stackrel{def}{=} \{(y_1 f(x_1), \ldots, y_n f(x_n)) \mid f \in \mathbb{F}_q[x]_{<k}\}.$$

### 2.2 Schur Product of Codes and Square Codes Distinguisher

**Notation 2.** *The component-wise product of two vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ in $\mathbb{F}_q^n$ is denoted by : $\boldsymbol{a} \star \boldsymbol{b} \stackrel{def}{=} (a_1 b_1, \ldots, a_n b_n)$. This definition extends to the product of codes where the* Schur product *of two codes $\mathscr{A}$ and $\mathscr{B} \subseteq \mathbb{F}_q^n$ is defined as*

$$\mathscr{A} \star \mathscr{B} \stackrel{def}{=} \mathbf{Span}_{\mathbb{F}_q} \{\boldsymbol{a} \star \boldsymbol{b} \mid \boldsymbol{a} \in \mathscr{A},\ \boldsymbol{b} \in \mathscr{B}\}.$$

*In particular, $\mathscr{A}^{\star 2}$ denotes the* square code *of a code $\mathscr{A}$: $\mathscr{A}^{\star 2} \overset{def}{=} \mathscr{A} \star \mathscr{A}$.*

We recall the following result on the generic behaviour of random codes with respect to this operation.

**Proposition 1** *(*[6, Theorem 2.3]*, informal). For a linear code $\mathscr{R}$ chosen at random over $\mathbb{F}_q$ of dimension $k$ and length $n$, the dimension of $\mathscr{R}^{\star 2}$ is typically* $\min(n, \binom{k+1}{2})$*.*

This provides a distinguisher between random codes and algebraically structured codes such as generalised Reed Solomon codes [8,19], Reed Muller codes [7], polar codes [2] some Goppa codes [10,12] or algebraic geometry codes [9]. For instance, in the case of GRS codes, we have the following result.

**Proposition 2.** *Let $n, k, \boldsymbol{x}, \boldsymbol{y}$ be as in Definition 1. Then,*

$$(\boldsymbol{GRS}_k(\boldsymbol{x}, \boldsymbol{y}))^{\star 2} = \boldsymbol{GRS}_{2k-1}(\boldsymbol{x}, \boldsymbol{y} \star \boldsymbol{y}).$$

*In particular, if $k < n/2$, then* $\dim (\boldsymbol{GRS}_k(\boldsymbol{x}, \boldsymbol{y}))^{\star 2} = 2k - 1$.

Thus, compared to a random code $\mathscr{R}$ whose square has a dimension quadratic in $\dim \mathscr{R}$, the square of a GRS code $\mathscr{C}$ has a dimension which is linear in $\dim \mathscr{C}$. This criterion allows to distinguish GRS codes of appropriate dimension from random codes.

### 2.3 Punctured and Shortened Codes

The notions of *puncturing* and *shortening* are classical ways to build new codes from existing ones. These constructions will be useful for the attack. We recall here their definition. For a codeword $\boldsymbol{c} \in \mathbb{F}_q^n$, we denote $(c_1, \ldots, c_n)$ its entries.

**Definition 2 (punctured and restricted codes).** *Let $\mathscr{C} \subseteq \mathbb{F}_q^n$ and $\mathcal{L} \subseteq [\![1, n]\!]$. The* puncturing of $\mathscr{C}$ at $\mathcal{L}$ *is defined as the code*

$$\mathcal{P}_{\mathcal{L}}(\mathscr{C}) \overset{def}{=} \{(c_i)_{i \in [\![1,n]\!] \setminus \mathcal{L}} \text{ s.t. } \boldsymbol{c} \in \mathscr{C}\}.$$

*The* restriction *of $\mathscr{C}$ to $\mathcal{L}$ is defined as the code $\mathcal{R}_{\mathcal{L}}(\mathscr{C}) \overset{def}{=} \mathcal{P}_{[\![1,n]\!] \setminus \mathcal{L}}(\mathscr{C})$.*

**Definition 3 (shortened code).** *Let $\mathscr{C} \subseteq \mathbb{F}_q^n$ and $\mathcal{L} \subseteq [\![1, n]\!]$. The* shortening of $\mathscr{C}$ at $\mathcal{L}$ *is defined as the code*

$$\mathcal{S}_{\mathcal{L}}(\mathscr{C}) \overset{def}{=} \mathcal{P}_{\mathcal{L}}(\{\boldsymbol{c} \in \mathscr{C} \text{ s.t. } \forall i \in \mathcal{L},\ c_i = 0\}).$$

Shortening a code is equivalent to puncturing the dual code, as explained by the following proposition, whose proof can be found in [13, Theorem 1.5.7].

**Proposition 3.** *Let $\mathscr{C}$ be a linear code over $\mathbb{F}_q^n$ and $\mathcal{L} \subseteq [\![1, n]\!]$. Then,*

$$\mathcal{S}_{\mathcal{L}}\left(\mathscr{C}^{\perp}\right) = (\mathcal{P}_{\mathcal{L}}(\mathscr{C}))^{\perp} \text{ and } (\mathcal{S}_{\mathcal{L}}(\mathscr{C}))^{\perp} = \mathcal{P}_{\mathcal{L}}\left(\mathscr{C}^{\perp}\right),$$

*where $\mathscr{A}^{\perp}$ denotes the dual of the code $\mathscr{A}$.*

**Notation 3.** *Throughout the document, the indexes of the columns (or positions of the codewords) will always refer to the indexes in the original code, although the code has been punctured or shortened. For instance, consider a code $\mathscr{C}$ of length 5 where every word $\boldsymbol{c} \in \mathscr{C}$ is indexed $\boldsymbol{c} = (c_1, c_2, c_3, c_4, c_5)$. If we puncture $\mathscr{C}$ in $\{1, 3\}$, a codeword $\boldsymbol{c}' \in \mathcal{P}_{\{1,3\}}(\mathscr{C})$ will be indexed $(c'_2, c'_4, c'_5)$ and not $(c'_1, c'_2, c'_3)$.*

# 3 The RLCE Scheme

## 3.1 Presentation of the Scheme

The RLCE encryption scheme is a code based cryptosystem, inspired by the McEliece scheme. It has been introduced by Wang in [16] and a proposal called "RLCE-KEM" has recently been submitted as a response for the NIST's call for post-quantum cryptosystems [17].

For a message $\boldsymbol{m} \in \mathbb{F}_q^k$, the cipher text is $\boldsymbol{c} = \boldsymbol{m}\boldsymbol{G} + \boldsymbol{e}$ where $\boldsymbol{e} \in \mathbb{F}_q^{n+w}$ is a random error vector of small weight $t$ and $\boldsymbol{G} \in \mathbb{F}_q^{k \times (n+w)}$ is a generator matrix defined as follows, for given parameters $n, k$ and $w$.

1. Let $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{F}_q^n$ be respectively a support and a multiplier (as in Definition 1).
2. Let $\boldsymbol{G}_0$ denote a random $k \times n$ generator matrix of the generalised Reed Solomon code $\mathbf{GRS}_k(\boldsymbol{x}, \boldsymbol{y})$ of length $n$ and dimension $k$. Denote by $g_1, \dots, g_n$ the columns of $\boldsymbol{G}_0$.
3. Let $r_1, \dots, r_w$ be column vectors chosen uniformly at random in $\mathbb{F}_q^k$. Denote by $\boldsymbol{G}_1$ the matrix obtained by inserting the random columns between GRS columns at the end of $\boldsymbol{G}_0$ as follows:
$$\boldsymbol{G}_1 \stackrel{\text{def}}{=} [g_1, \dots, g_{n-w}, g_{n-w+1}, r_1, \dots, g_n, r_w] \in \mathbb{F}_q^{k \times (n+w)}.$$
4. Let $\boldsymbol{A}_1, \dots, \boldsymbol{A}_w$ be $2 \times 2$ matrices chosen uniformly at random in $\mathbf{GL}_2(\mathbb{F}_q)$. Let $\boldsymbol{A}$ be the block–diagonal non singular matrix
$$\boldsymbol{A} \stackrel{\text{def}}{=} \begin{pmatrix} \boldsymbol{I}_{n-w} & & & (0) \\ & \boldsymbol{A}_1 & & \\ & & \ddots & \\ (0) & & & \boldsymbol{A}_w \end{pmatrix} \in \mathbb{F}_q^{(n+w)\times(n+w)}.$$
5. Let $\pi \in \mathfrak{S}_{n+w}$ be a randomly chosen permutation of $[\![1, n+w]\!]$ and $\boldsymbol{P}$ the corresponding $(n+w) \times (n+w)$ permutation matrix.
6. The public key is the matrix $\boldsymbol{G} \stackrel{\text{def}}{=} \boldsymbol{G}_1\boldsymbol{A}\boldsymbol{P}$ and the private key is $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{A}, \boldsymbol{P})$.

*Remark 1.* This presentation of the scheme is not exactly the same as in the original specifications of RLCE [17]. It is however equivalent. Indeed, the differences with the original scheme are listed below.

1. The original specifications of RLCE propose as a public key a matrix

$$\boldsymbol{G} = \boldsymbol{S}\boldsymbol{G}_1\boldsymbol{A}\boldsymbol{P},$$

where $\boldsymbol{S}$ is a $k \times k$ non singular matrix. But, since we chose $\boldsymbol{G}_0$ to be a *random* generator matrix of the GRS code to which we included random columns, left multiplication by a random nonsingular matrix does not change the probability distribution of the public keys.
2. In [17], the matrix $\boldsymbol{G}_0$ is called $\boldsymbol{G}_s$ and is a generator matrix of a GRS code but its columns are permuted using a permutation matrix $\boldsymbol{P}_1$ before including random columns. Actually, if we chose *arbitrary* supports and multipliers, applying a permutation on the columns does not change the probability distribution of the public keys.

### 3.2 Suggested Sets of Parameters

In [17] the author proposes 2 groups of 3 sets of parameters. The first group (referred to as *odd ID* parameters) corresponds to parameters such that $w \in [0.6(n-k), 0.7(n-k)]$, whereas in the second group (*even ID* parameters) the parameters satisfy $w = n - k$. The parameters of these two groups are listed in Tables 1 and 2.

The attack of the present paper recovers in polynomial time any secret key when parameters lie in the first group.

**Table 1.** Set of parameters for the first group: $w \in [0.6(n-k), 0.7(n-k)]$.

| Security level (bits) | Name in [17] | $n$ | $k$ | $t$ | $w$ | $q$ | Public key size (kB) |
|---|---|---|---|---|---|---|---|
| 128 | ID 1 | 532 | 376 | 78 | 96 | $2^{10}$ | 118 |
| 192 | ID 3 | 846 | 618 | 114 | 144 | $2^{10}$ | 287 |
| 256 | ID 5 | 1160 | 700 | 230 | 311 | $2^{11}$ | 742 |

**Table 2.** Set of parameters for the second group: $w = n - k$.

| Security level (bits) | Name in [17] | $n$ | $k$ | $t$ | $w$ | $q$ | Public key size (kB) |
|---|---|---|---|---|---|---|---|
| 128 | ID 0 | 630 | 470 | 80 | 160 | $2^{10}$ | 188 |
| 192 | ID 2 | 1000 | 764 | 118 | 236 | $2^{10}$ | 450 |
| 256 | ID 4 | 1360 | 800 | 280 | 560 | $2^{11}$ | 1232 |

## 4 Distinguishing by Shortening and Squaring

We will show here that it is possible to distinguish some public keys from random codes by computing the square of some shortening of the public code. More precisely, here is our main result.

**Theorem 4.** *Let $\mathscr{C}$ be a code over $\mathbb{F}_q$ of length $n+w$ and dimension $k$ with generator matrix $\boldsymbol{G}$ which is the public key of an RLCE scheme that is based on a GRS code of length $n$ and dimension $k$. Let $\mathcal{L} \subset [\![1, n+w]\!]$. Then,*

$$\dim\left(\mathcal{S}_{\mathcal{L}}\left(\mathscr{C}\right)\right)^{\star 2} \leqslant \min(n+w-|\mathcal{L}|,\ 2(k+w-|\mathcal{L}|)-1).$$

*Remark 2.* Actually, according to computer experiments, the inequality established in Theorem 4 seems to be an equality with a probability close to 1 when we are not in the degenerate case described in Sect. 6.7. See Remark 4 for further details.

To prove Theorem 4 we can assume that $\boldsymbol{P}$ is the identity matrix. This is because of the following lemma.

**Lemma 1.** *For any permutation $\sigma$ of the code positions $[\![1, n+w]\!]$ we have*

$$\dim\left(\mathcal{S}_{\mathcal{L}}\left(\mathscr{C}\right)\right)^{\star 2} = \dim\left(\mathcal{S}_{\mathcal{L}^\sigma}\left(\mathscr{C}^\sigma\right)\right)^{\star 2},$$

*where $\mathscr{C}^\sigma$ is the set of codewords in $\mathscr{C}$ permuted by $\sigma$, that is $\mathscr{C}^\sigma = \{\boldsymbol{c}^\sigma : \boldsymbol{c} \in \mathscr{C}\}$ where $\boldsymbol{c}^\sigma \overset{def}{=} (c_{\sigma(i)})_{i \in [\![1,n+w]\!]}$ and $\mathcal{L}^\sigma \overset{def}{=} \{\sigma(i) : i \in \mathcal{L}\}$.*

Therefore, for the analysis of the distinguisher, we can make the following assumption which we will use several times the rest of the section, especially to simplify the notation. The general case will follow by using Lemma 1.

**Assumption 5.** *The permutation matrix $\boldsymbol{P}$ is the identity matrix.*

### 4.1 Analysis of the Different Kinds of Columns

**Notation and Terminology.** Before proving the result, let us introduce some notation and terminology. The set of positions $[\![1, n+w]\!]$ splits in a natural way into four sets, whose definitions are given in the sequel

$$[\![1, n+w]\!] = \mathcal{I}^1_{\text{GRS}} \cup \mathcal{I}^2_{\text{GRS}} \cup \mathcal{I}_{\text{R}} \cup \mathcal{I}_{\text{PR}}. \tag{1}$$

**Definition 4.** *The set of* GRS positions of the first kind, *denoted $\mathcal{I}^1_{\text{GRS}}$, corresponds to GRS columns which have not been associated to a random column. This set has cardinality $n-w$ and is given by*

$$\mathcal{I}^1_{\text{GRS}} \overset{def}{=} \{i \in [\![1, n+w]\!] \mid \pi^{-1}(i) \leqslant n-w\}. \tag{2}$$

*Under Assumption 5, this becomes: $\mathcal{I}^1_{\text{GRS}} \overset{def}{=} [\![1, n-w]\!]$.*

This set is called this way, because at a position $i \in \mathcal{I}^1_{\text{GRS}}$, any codeword $\boldsymbol{v} \in \mathscr{C}$ has an entry of the form

$$v_i = y_i f(x_i). \tag{3}$$

As we will see later, there might be other code positions that are of this form.

**Definition 5.** *The set of* twin positions, *denoted* $\mathcal{I}_{\mathrm{T}}$*, corresponds to columns that result in a mix of a random column and a GRS one. This set has cardinality* $2w$ *and is equal to:*

$$\mathcal{I}_{\mathrm{T}} \overset{def}{=} \{i \in [\![1, n+w]\!] \mid \pi^{-1}(i) > n-w\}.$$

*Under Assumption* 5*, this becomes:* $\mathcal{I}_{\mathrm{T}} \overset{def}{=} [\![n-w+1, n+w]\!]$.

The set $\mathcal{I}_{\mathrm{T}}$ can be divided in several subsets as follows.

**Definition 6.** *Each position in* $\mathcal{I}_{\mathrm{T}}$ *has a unique corresponding twin position which is the position of the column with which it was mixed. For all* $s \in [\![1, w]\!]$*,* $\pi(n-w+2s-1)$ *and* $\pi(n-w+2s)$ *are twin positions. Under Assumption* 5*, the positions* $n-w+2s-1$ *and* $n-w+2s$ *are twins for all* $s$ *in* $[\![1, w]\!]$*.*

For convenience, we introduce the following notation.

**Notation 6.** *The twin of a position* $i \in \mathcal{I}_{\mathrm{T}}$ *is denoted by* $\tau(i)$*.*

To any twin pair $\{i, \tau(i)\} = \{\pi(n-w+2s-1), \pi(n-w+2s)\}$ with $s \in \{1, \ldots, w\}$ is associated a unique linear form $\psi_s : \mathbb{F}_q[x]_{<k} \to \mathbb{F}_q$ and a non-singular matrix $\boldsymbol{A}_s$ such that for any codeword $\boldsymbol{v} \in \mathscr{C}$, we have

$$\begin{aligned} v_i &= a_s y_j f(x_j) + c_s \psi_s(f) \\ v_{\tau(i)} &= b_s y_j f(x_j) + d_s \psi_s(f), \end{aligned} \tag{4}$$

where $j = n-w+s$ and

$$\begin{pmatrix} a_s & b_s \\ c_s & d_s \end{pmatrix} = \boldsymbol{A}_s. \tag{5}$$

The linear form $\psi_s$ is the form whose evaluations provides the random column added on the right of the $(n-w+s)$-th column during the construction process of $\boldsymbol{G}$ (see Sect. 3.1, Step 3). From (4), we see that we may obtain more GRS positions: indeed $v_i = a_s y_j f(x_j)$ if $c_s = 0$ or $v_{\tau(i)} = b_s y_j f(x_j)$ if $d_s = 0$. On the other hand if $c_s d_s \neq 0$ the twin pairs are *correlated* in the sense that they behave in a non-trivial way after shortening: Lemma 3 shows that if one shortens the code in such a position its twin becomes a GRS position. We therefore call such a twin pair a *pseudo-random* twin pair and the set of pseudo-random twin pairs forms what we call the set of *pseudo-random* positions.

**Definition 7.** *The set of* pseudo-random positions *(PR in short), denoted* $\mathcal{I}_{\mathrm{PR}}$*, is given by*

$$\mathcal{I}_{\mathrm{PR}} \overset{def}{=} \bigcup_{s \in [\![1,w]\!] \ s.t. \ c_s d_s \neq 0} \{\pi(n-w+2s-1), \pi(n-w+2s)\}. \tag{6}$$

*Under Assumption* 5*, this becomes:*

$$\mathcal{I}_{\mathrm{PR}} = \bigcup_{s \in [\![1,w]\!] \ s.t. \ c_s d_s \neq 0} \{n-w+2s-1, n-w+2s\}. \tag{7}$$

If $c_s d_s = 0$, then a twin pair splits into a GRS position of the second kind and a random position. The GRS position of the second kind is $\pi(n-w+2s-1)$ if $c_s = 0$ or $\pi(n-w+2s)$ if $d_s = 0$ ($c_s$ and $d_s$ can not both be equal to 0 since $\boldsymbol{A}_s$ is invertible).

**Definition 8.** *The set* GRS positions of the second kind*, denoted* $\mathcal{I}^2_{\mathrm{GRS}}$*, is defined as*

$$\mathcal{I}^2_{\mathrm{GRS}} \overset{def}{=} \{\pi(n-w+2s-1) \,|\, c_s = 0\} \cup \{\pi(n-w+2s) \,|\, d_s = 0\}. \tag{8}$$

*Under Assumption 5, this becomes:*

$$\mathcal{I}^2_{\mathrm{GRS}} = \{n-w+2s-1 \,|\, c_s = 0\} \cup \{n-w+2s \,|\, d_s = 0\}. \tag{9}$$

**Definition 9.** *The set of* random positions*, denoted* $\mathcal{I}_{\mathrm{R}}$*, is defined as*

$$\mathcal{I}_{\mathrm{R}} \overset{def}{=} \{\pi(n-w+2s-1) \,|\, d_s = 0\} \cup \{\pi(n-w+2s) \,|\, c_s = 0\}. \tag{10}$$

*Under Assumption 5, this becomes:*

$$\mathcal{I}_{\mathrm{R}} = \{n-w+2s-1 \,|\, d_s = 0\} \cup \{n-w+2s \,|\, c_s = 0\}. \tag{11}$$

We also define the *GRS positions* to be the GRS positions of the first or the second kind.

**Definition 10.** *The set of* GRS positions*, denoted* $\mathcal{I}_{\mathrm{GRS}}$*, is defined as*

$$\mathcal{I}_{\mathrm{GRS}} \overset{def}{=} \mathcal{I}^1_{\mathrm{GRS}} \cup \mathcal{I}^2_{\mathrm{GRS}}. \tag{12}$$

We finish this subsection with a lemma.

**Lemma 2.** $|\mathcal{I}^2_{\mathrm{GRS}}| = |\mathcal{I}_{\mathrm{R}}|$ *and* $|\mathcal{I}_{\mathrm{PR}}| = 2(w - |\mathcal{I}_{\mathrm{R}}|)$.

*Proof.* Using (7), (9) and (11) we see that, under Assumption 5,

$$[\![n-w+1, n+w]\!] = \mathcal{I}_{\mathrm{PR}} \cup \mathcal{I}^2_{\mathrm{GRS}} \cup \mathcal{I}_{\mathrm{R}} \tag{13}$$

and the above union is disjoint. Next, there is a one-to-one correspondence relating $\mathcal{I}^2_{\mathrm{GRS}}$ and $\mathcal{I}_{\mathrm{R}}$. Indeed, still under Assumption 5, if $c_s = 0$ for some $s \in [\![1, w]\!]$, then $n-w+2s-1 \in \mathcal{I}^2_{\mathrm{GRS}}$ and $n-w+2s \in \mathcal{I}_{\mathrm{R}}$ and conversely if $d_s = 0$. This proves that $|\mathcal{I}^2_{\mathrm{GRS}}| = |\mathcal{I}_{\mathrm{R}}|$, which, together with (13) yields the result. □

### 4.2 Intermediate Results

Before proceeding to the proof of Theorem 4, let us state and prove some intermediate results. We will start by Lemmas 3 and 4, that will be useful to prove Proposition 4 on the structure of shortened RLCE codes, by induction on the number of shortened positions. This proposition will be the core of the proof of Theorem 4. Then, we will prove a general result on modified GRS codes with additional random columns.

**Two Useful Lemmas.** The first lemma explains that, after shortening a PR position, its twin will behave like a GRS position. This is actually a crucial lemma that explains why PR columns in $\boldsymbol{G}$ do not really behave like random columns after shortening the code at the corresponding position.

**Lemma 3.** *Let $i$ be a PR position and $\mathcal{L}$ a set of positions that neither contains $i$ nor $\tau(i)$. Let $\mathscr{C}' \overset{def}{=} \mathcal{S}_{\mathcal{L}}\left(\mathscr{C}\right)$. The position $\tau(i)$* behaves like a GRS position *in the code $\mathcal{S}_{\{i\}}\left(\mathscr{C}'\right)$. That is, the $\tau(i)$-th column of a generator matrix of $\mathcal{S}_{\{i\}}\left(\mathscr{C}'\right)$ has entries of the form*

$$\tilde{y}_j f(x_j)$$

*for some $j$ in $[\![n-w+1, n]\!]$ and $\tilde{y}_j$ in $\mathbb{F}_q$.*

*Proof.* Let us assume that $i = n-w+2s-1$ for some $s \in \{1, \ldots, w\}$. The case $i = n-w+2s$ can be proved in a similar way. At position $i$, for any $\boldsymbol{c} \in \mathscr{C}'$, from (4), we have

$$c_i = ay_j f(x_j) + c\psi_s(f),$$

where $j = n-w+s$. By shortening, we restrict our space of polynomials to the subspace of polynomials in $\mathbb{F}_q[x]_{<k}$ satisfying $c_i = 0$. Since $i$ is a PR position, $c \neq 0$ and therefore

$$\psi_s(f) = -c^{-1}ay_j f(x_j).$$

Therefore, at the twin position $\tau(i) = n-w+2s$ and for any $\boldsymbol{c} \in \mathcal{S}_{\{i\}}\left(\mathscr{C}'\right)$, we have

$$\begin{aligned} c_{\tau(i)} &= by_j f(x_j) + d\psi_j(f) \\ &= y_j(b - dac^{-1})f(x_j). \end{aligned}$$

□

*Remark 3.* This lemma does not hold for a random position, since the proof requires that $c \neq 0$. It is precisely because of this that we have to make a distinction between twin pairs, *i.e.* pairs for which the associated matrix $\boldsymbol{A}_s$ is such that $c_s d_s \neq 0$ and pairs for which it is not the case.

This lemma allows us to get some insight on the structure of the shortened code $\mathcal{S}_{\mathcal{L}}\left(\mathscr{C}\right)$. Before giving the relevant statement let us first recall the following result.

**Lemma 4.** *Consider a linear code $\mathscr{A}$ over $\mathbb{F}_q$ whose restriction to a subset $\mathcal{L}$ is a subcode of a $k$-dimensional GRS code. Let $i$ be an element of $\mathcal{L}$. Then the restriction of $\mathcal{S}_{\{i\}}\left(\mathscr{A}\right)$ to $\mathcal{L} \setminus \{i\}$ is a subcode of a $(k-1)$-dimensional GRS code.*

*Proof.* By definition, the restriction $\mathscr{A}'$ to $\mathcal{L}$ is a code of the form

$$\mathscr{A}' \overset{\text{def}}{=} \left\{ \left(y_j f(x_j)\right)_{j \in \mathcal{L}} : f \in L \right\},$$

where the $y_j$'s are nonzero elements of $\mathbb{F}_q$, the $x_j$'s are distinct elements of $\mathbb{F}_q$ and $L$ is a subspace of $\mathbb{F}_q[X]_{<k}$. Clearly the restriction $\mathscr{A}''$ of $\mathcal{S}_{\{i\}}\left(\mathscr{A}\right)$ to $\mathcal{L}\setminus\{i\}$ can be written as

$$\mathscr{A}'' = \left\{\left(y_j f(x_j)\right)_{j\in\mathcal{L}\setminus\{i\}} : f \in L, f(x_i) = 0\right\}.$$

The polynomials $f(X)$ in $L$ such that $f(x_i) = 0$ can be written as $f(X) = (X - x_i)g(X)$ where $\deg g = \deg f - 1$ and $g$ ranges in this case over a subspace $L'$ of polynomials of degree $< k - 1$. We can therefore write

$$\mathscr{A}'' = \left\{\left(y_j(x_j - x_i)g(x_j)\right)_{j\in\mathcal{L}\setminus\{i\}} : g \in L'\right\}.$$

This implies our lemma. □

**The Key Proposition.** Using Lemmas 3 and 4, we can prove the following result by induction. This result is the key proposition for proving Theorem 4.

**Proposition 4.** *Let $\mathcal{L}$ be a subset of $[\![1, n + w]\!]$ and let $\mathcal{L}_0, \mathcal{L}_1, \mathcal{L}_2$ be subsets of $\mathcal{L}$ defined as*

- *$\mathcal{L}_0$ the set of GRS positions (see (2), (8) and (12) for a definition) of $\mathcal{L}$:*

$$\mathcal{L}_0 \overset{def}{=} \mathcal{L} \cap \mathcal{I}_{\mathrm{GRS}};$$

- *$\mathcal{L}_1$ the set of PR positions (see (6)) of $\mathcal{L}$ that do not have their twin in $\mathcal{L}$:*

$$\mathcal{L}_1 \overset{def}{=} \{i \in \mathcal{L} \cap \mathcal{I}_{\mathrm{PR}} \mid \tau(i) \notin \mathcal{L}\};$$

- *$\mathcal{L}_2$ the set of PR positions of $\mathcal{L}$ whose twin position is also included in $\mathcal{L}$:*

$$\mathcal{L}_2 \overset{def}{=} \{i \in \mathcal{L} \cap \mathcal{I}_{\mathrm{PR}} \mid \tau(i) \in \mathcal{L}\}.$$

*Let $\mathscr{C}'$ be the restriction of $\mathcal{S}_{\mathcal{L}}\left(\mathscr{C}\right)$ to $(\mathcal{I}_{\mathrm{GRS}} \setminus \mathcal{L}_0) \cup \tau(\mathcal{L}_1)$. Then, $\mathscr{C}'$ is a subcode of a GRS code of length $|\mathcal{I}_{\mathrm{GRS}}| - |\mathcal{L}_0| + |\mathcal{L}_1|$ and dimension $k - |\mathcal{L}_0| - \frac{|\mathcal{L}_2|}{2}$.*

*Proof.* Let us prove by induction on $\ell = |\mathcal{L}|$ that $\mathscr{C}'$ is a subcode of a GRS code of length $|\mathcal{I}_{\mathrm{GRS}}| - |\mathcal{L}_0| + |\mathcal{L}_1|$ and dimension $k - |\mathcal{L}_0| - \frac{|\mathcal{L}_2|}{2}$.

This statement is clearly true if $\ell = 0$, *i.e.* if $\mathcal{L}$ is the empty set. Assume that the result is true for all $\mathcal{L}$ up to some size $\ell \geqslant 0$. Consider now a set $\mathcal{L}$ of size $\ell + 1$. We can write $\mathcal{L} = \mathcal{L}' \cup \{i\}$ where $\mathcal{L}'$ is of size $\ell$.

Let $\mathcal{L}_0, \mathcal{L}_1, \mathcal{L}_2$ be subsets of $\mathcal{L}$ as defined in the statement and $\mathcal{L}'_0, \mathcal{L}'_1, \mathcal{L}'_2$ be the subsets of $\mathcal{L}'$ obtained by replacing in the statement $\mathcal{L}$ by $\mathcal{L}'$. There are now several cases to consider for $i$.

**Case 1:** $i \in \mathcal{L}_0$. In this case, $\mathcal{L}_0 = \mathcal{L}'_0 \cup \{i\}$, $\mathcal{L}_1 = \mathcal{L}'_1$ and $\mathcal{L}_2 = \mathcal{L}'_2$.
We can apply Lemma 4 with $\mathscr{A} = \mathcal{S}_{\mathcal{L}'}\left(\mathscr{C}\right)$ because by the induction hypothesis, its restriction to $\mathcal{L}'' \overset{\text{def}}{=} (\mathcal{I}_{\mathrm{GRS}} \setminus \mathcal{L}'_0) \cup \tau(\mathcal{L}'_1)$ is a subcode of a GRS code of length $|\mathcal{I}_{\mathrm{GRS}}| - |\mathcal{L}'_0| + |\mathcal{L}'_1|$ and dimension $k - |\mathcal{L}'_0| - \frac{|\mathcal{L}'_2|}{2}$.
Therefore the restriction of the shortened code $\mathcal{S}_{\mathcal{L}}\left(\mathscr{C}\right) = \mathcal{S}_{\{i\}}\left(\mathscr{A}\right)$ to $\mathcal{L}'' \setminus \{i\} = (\mathcal{I}_{\mathrm{GRS}} \setminus \mathcal{L}_0) \cup \tau(\mathcal{L}_1)$ is a subcode of a GRS code of length $|\mathcal{I}_{\mathrm{GRS}}| - |\mathcal{L}_0| + |\mathcal{L}_1|$ and dimension $k - |\mathcal{L}'_0| - \frac{|\mathcal{L}'_2|}{2} - 1 = k - |\mathcal{L}_0| - \frac{|\mathcal{L}_2|}{2}$.

**Case 2:** $i \in \mathcal{L}_1$. In this case, $\mathcal{L}_0 = \mathcal{L}'_0, \mathcal{L}_1 = \mathcal{L}'_1 \cup \{i\}$ and $\mathcal{L}_2 = \mathcal{L}'_2$. This implies that $\mathcal{L}'$ does not contain $i$ nor $\tau(i)$.

We can therefore apply Lemma 3 with $\mathscr{C}' = \mathcal{S}_{\mathcal{L}'}(\mathscr{C})$. Lemma 3 states that the position $\tau(i)$ behaves like a GRS position in $\mathcal{S}_{\{i\}}(\mathscr{C}') = \mathcal{S}_{\mathcal{L}}(\mathscr{C})$. By induction hypothesis, the restriction of the code $\mathscr{C}'$ to $(\mathcal{I}_{\mathrm{GRS}} \setminus \mathcal{L}'_0) \cup \tau(\mathcal{L}'_1)$ is a subcode of a GRS code of length $|\mathcal{I}_{\mathrm{GRS}}| - |\mathcal{L}'_0| + |\mathcal{L}'_1|$ and dimension $k - |\mathcal{L}'_0| - \frac{|\mathcal{L}'_2|}{2} = k - |\mathcal{L}_0| - \frac{|\mathcal{L}_2|}{2}$.

Therefore the restriction of $\mathcal{S}_{\{i\}}(\mathscr{C}') = \mathcal{S}_{\mathcal{L}}(\mathscr{C})$ to $(\mathcal{I}_{\mathrm{GRS}} \setminus \mathcal{L}_0) \cup \tau(\mathcal{L}_1) = (\mathcal{I}_{\mathrm{GRS}} \setminus \mathcal{L}'_0) \cup \tau(\mathcal{L}'_1) \cup \{\tau(i)\}$ is a subcode of a GRS code of dimension $k - |\mathcal{L}_0| - \frac{|\mathcal{L}_2|}{2}$ and length $|\mathcal{I}_{\mathrm{GRS}}| - |\mathcal{L}'_0| + |\mathcal{L}'_1| + 1 = |\mathcal{I}_{\mathrm{GRS}}| - |\mathcal{L}_0| + |\mathcal{L}_1|$.

**Case 3:** $i \in \mathcal{L}_2$. In this case, $\mathcal{L}_0 = \mathcal{L}'_0, \mathcal{L}_1 = \mathcal{L}'_1 \setminus \{\tau(i)\}$ and $\mathcal{L}_2 = \mathcal{L}'_2 \cup \{i, \tau(i)\}$. In fact, this case can only happen if $\ell \geqslant 1$ and we will rather consider the induction with respect to the set $\mathcal{L}'' = \mathcal{L} \setminus \{i, \tau(i)\}$ of size $\ell - 1$ and the sets $\mathcal{L}''_0, \mathcal{L}''_1, \mathcal{L}''_2$ such that $\mathcal{L}''_0 = \mathcal{L}_0, \mathcal{L}''_1 = \mathcal{L}_1, \mathcal{L}''_2 = \mathcal{L}_2 \setminus \{i, \tau(i)\}$.

By induction hypothesis on $\mathcal{L}''$, the restriction of $\mathscr{C}'' \stackrel{\text{def}}{=} \mathcal{S}_{\mathcal{L}''}(\mathscr{C})$ to $(\mathcal{I}_{\mathrm{GRS}} \setminus \mathcal{L}''_0) \cup \tau(\mathcal{L}''_1)$ is a subcode of a GRS code of length $|\mathcal{I}_{\mathrm{GRS}}| - |\mathcal{L}''_0| + |\mathcal{L}''_1| = |\mathcal{I}_{\mathrm{GRS}}| - |\mathcal{L}_0| + |\mathcal{L}_1|$ and dimension $k - |\mathcal{L}''_0| - \frac{|\mathcal{L}''_2|}{2} = k - |\mathcal{L}_0| - \frac{|\mathcal{L}_2|}{2} + 1$.

Following Assumption 5, we can write without loss of generality that $i = n - w + 2s - 1$ for some $s \in \{1, \ldots, w\}$. The case $i = n - w + 2s$ can be proved in a similar way.

Denote $\boldsymbol{A}_s = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ the non-singular matrix and $j = n - w + s$. For any $\boldsymbol{c} \in \mathscr{C}'$, at positions $i$ and $\tau(i)$ we have

$$\begin{aligned} c_i &= a y_j f(x_j) + c\psi_s(f), \\ c_{\tau(i)} &= b y_j f(x_j) + d\psi_s(f). \end{aligned}$$

Shortening $\mathscr{C}''$ at $\{i, \tau(i)\}$ has the effect of requiring to consider only the polynomials $f$ for which $f(x_j) = \psi_s(f) = 0$. Therefore the restriction of $\mathcal{S}_{\{i,\tau(i)\}}(\mathscr{C}'') = \mathcal{S}_{\mathcal{L}}(\mathscr{C})$ at $(\mathcal{I}_{\mathrm{GRS}} \setminus \mathcal{L}''_0) \cup \tau(\mathcal{L}''_1)$ is a subcode of a GRS code of length $|\mathcal{I}_{\mathrm{GRS}}| - |\mathcal{L}_0| + |\mathcal{L}_1|$ and dimension $k - |\mathcal{L}_0| - \frac{|\mathcal{L}_2|}{2} + 1 - 1 = k - |\mathcal{L}_0| - \frac{|\mathcal{L}_2|}{2}$.

**Case 4:** $i \in \mathcal{I}_{\mathrm{R}}$. In this case $\mathcal{L}_0 = \mathcal{L}'_0, \mathcal{L}_1 = \mathcal{L}'_1$ and $\mathcal{L}_2 = \mathcal{L}'_2$. Using the induction hypothesis yields directly that $\mathscr{A} = \mathcal{S}_{\mathcal{L}'}(\mathscr{C})$ is a subcode of a GRS code of length $|\mathcal{I}_{\mathrm{GRS}}| - |\mathcal{L}'_0| + |\mathcal{L}'_1| = |\mathcal{I}_{\mathrm{GRS}}| - |\mathcal{L}_0| + |\mathcal{L}_1|$ and dimension $k - |\mathcal{L}'_0| - \frac{|\mathcal{L}'_2|}{2} = k - |\mathcal{L}_0| - \frac{|\mathcal{L}_2|}{2}$. This is also clearly the case for $\mathcal{S}_{\mathcal{L}}(\mathscr{C}) = \mathcal{S}_{\{i\}}(\mathscr{A})$.

This proves that the induction hypothesis also holds for $|\mathcal{L}| = \ell + 1$ and finishes the proof of the proposition. □

**A General Result on Modified GRS Codes.** Finally, we need a very general result concerning modified GRS codes where some arbitrary columns have been joined to the generator matrix. A very similar lemma is already proved in [8, Lemma 9]. Its proof is repeated below for convenience and in order to provide further details about the equality case.

**Lemma 5.** *Consider a linear code $\mathscr{A}$ over $\mathbb{F}_q$ with generator matrix of the form $\boldsymbol{G} = \left(\boldsymbol{G}_{\mathrm{SCGRS}}\ \boldsymbol{G}_{\mathrm{rand}}\right) \boldsymbol{P}$ of size $k \times (n+r)$ where $\boldsymbol{G}_{\mathrm{SCGRS}}$ is a $k \times n$ generator matrix of a subcode of a GRS code of dimension $k_{\mathrm{GRS}}$ over $\mathbb{F}_q$, $\boldsymbol{G}_{\mathrm{rand}}$ is an arbitrary matrix in $\mathbb{F}_q^{k\times r}$ and $\boldsymbol{P}$ is the permutation matrix of an arbitrary permutation $\sigma \in \mathfrak{S}_{n+r}$. We have*

$$\dim \mathscr{A}^{\star 2} \leqslant 2k_{\mathrm{GRS}} - 1 + r.$$

*Moreover, if the equality holds, then for every $i \in [\![n+1, n+w]\!]$ we have:*

$$\dim \mathcal{P}_{\{\sigma(i)\}}\left(\mathscr{A}^{\star 2}\right) = \dim \mathscr{A}^{\star 2} - 1.$$

*Proof.* Without loss of generality, we may assume that $\boldsymbol{P}$ is the identity matrix since the dimension of the square code is invariant by permuting the code positions (see Lemma 1). Let $\mathscr{B}$ be the code with generator matrix $\left(\boldsymbol{G}_{\mathrm{SCGRS}}\ \boldsymbol{0}_{k\times r}\right)$, where $\boldsymbol{0}_{k\times r}$ is the zero matrix of size $k \times r$. We also define the code $\mathscr{B}'$ generated by the generator matrix $\left(\boldsymbol{0}_{k\times n}\ \boldsymbol{G}_{\mathrm{rand}}\right)$. We obviously have

$$\mathscr{A} \subseteq \mathscr{B} + \mathscr{B}'.$$

Therefore

$$\begin{aligned}(\mathscr{A})^{\star 2} &\subseteq (\mathscr{B} + \mathscr{B}')^{\star 2}\\ &\subseteq \mathscr{B}^{\star 2} + (\mathscr{B}')^{\star 2} + \mathscr{B} \star \mathscr{B}'\\ &\subseteq \mathscr{B}^{\star 2} + (\mathscr{B}')^{\star 2},\end{aligned}$$

where the last inclusion comes from the fact that $\mathscr{B} \star \mathscr{B}'$ is the zero subspace since $\mathscr{B}$ and $\mathscr{B}'$ have disjoint supports. The code $\mathscr{B}^{\star 2}$ has dimension $\leqslant 2k_{\mathrm{GRS}} - 1$ whereas $\dim (\mathscr{B}')^{\star 2} \leqslant r$.

Next, if $\dim \mathscr{A}^{\star 2} = 2k_{\mathrm{GRS}} - 1 + r$, then

$$\mathscr{A}^{\star 2} = \mathscr{B}^{\star 2} \oplus (\mathscr{B}')^{\star 2} \quad \text{and} \quad \dim(\mathscr{B}')^{\star 2} = r.$$

Since $\mathscr{B}'$ has length $r$, this means that $(\mathscr{B}')^{\star 2} = \mathbb{F}_q^r$ and hence, any word of weight 1 supported by the $r$ rightmost positions is contained in $\mathscr{A}^{\star 2}$. Therefore, puncturing this position will decrease the dimension. □

### 4.3 Proof of Theorem 4

*Proof.* By using Proposition 4, we know that the restriction of $\mathcal{S}_{\mathcal{L}}(\mathscr{C})$ to $(\mathcal{I}_{\mathrm{GRS}} \setminus \mathcal{L}_0) \cup \tau(\mathcal{L}_1)$ is a subcode of a GRS code of length $|\mathcal{I}_{\mathrm{GRS}}| - |\mathcal{L}_0| + |\mathcal{L}_1| = n - w + |\mathcal{I}^2_{\mathrm{GRS}}| - |\mathcal{L}_0| + |\mathcal{L}_1|$ and dimension $k_{\mathrm{GRS}} \overset{\text{def}}{=} k - |\mathcal{L}_0| - \frac{|\mathcal{L}_2|}{2}$, where:

- $\mathcal{L}_0 \overset{\text{def}}{=} \mathcal{I}_{\mathrm{GRS}} \cap \mathcal{L}$;
- $\mathcal{L}_1$ is the set of PR positions of $\mathcal{L}$ that do not have their twin in $\mathcal{L}$;
- $\mathcal{L}_2$ is the union of all twin PR positions that are both included in $\mathcal{L}$.

We also denote by $\mathcal{L}_3$ the set $\mathcal{I}_{\rm R} \cap \mathcal{L}$. We can then apply Lemma 5 to $\mathcal{S}_{\mathcal{L}}(\mathscr{C})$ and derive from it the following upper bound:

$$\dim(\mathcal{S}_{\mathcal{L}}(\mathscr{C}))^{\star 2} \leqslant 2k_{\rm GRS} - 1 + |\mathcal{I}_{\rm PR} \setminus (\mathcal{L} \cup \tau(\mathcal{L}_1))| + |\mathcal{I}_{\rm R} \setminus \mathcal{L}_3|. \tag{14}$$

Next, using Lemma 2, we get

$$\begin{aligned}
&\dim(\mathcal{S}_{\mathcal{L}}(\mathscr{C}))^{\star 2} \\
&\quad\leqslant 2\left(k - |\mathcal{L}_0| - \frac{|\mathcal{L}_2|}{2}\right) - 1 + 2(w - |\mathcal{I}_{\rm R}|) - 2|\mathcal{L}_1| - |\mathcal{L}_2| + |\mathcal{I}_{\rm R}| - |\mathcal{L}_3| \\
&\quad\leqslant 2(k + w - |\mathcal{L}_0| - |\mathcal{L}_1| - |\mathcal{L}_2| - |\mathcal{L}_3|) - 1 + (|\mathcal{L}_3| - |\mathcal{I}_{\rm R}|) \qquad (15)\\
&\quad\leqslant 2(k + w - |\mathcal{L}|) - 1. \qquad (16)
\end{aligned}$$

The other upper bound on $\dim(\mathcal{S}_{\mathcal{L}}(\mathscr{C}))^{\star 2}$ which is $\dim(\mathcal{S}_{\mathcal{L}}(\mathscr{C}))^{\star 2} \leqslant n + w - |\mathcal{L}|$ follows from the fact that the dimension of this code is bounded by its length. Putting both bounds together yields the theorem. □

*Remark 4.* We ran the following simulations using ID 1 parameters (see Table 1): for three hundred random independent public keys, we computed $\dim(\mathcal{S}_{\mathcal{L}}(\mathscr{C}))^{\star 2}$ for $|\mathcal{L}|$ ranging over $[\![\ell_{\min}, \ell_{\max}]\!]$, as defined in (21). For more than 99% of the cases, inequality (14) is an equality. In particular, this means that the inequality of Theorem 4 is almost always an equality whenever $\mathcal{I}_{\rm R}$ is the empty set, *i.e.* when we are not in the degenerate case defined in Sect. 6.7.

## 5 Reaching the Range of the Distinguisher

For this distinguisher to work we need to shorten the code enough so that its square does not fill in the ambient space, but not too much since the square of the shortened code should have a dimension strictly less than the typical dimension of the square of a random code given by Proposition 1. Namely, we need to have:

$$\dim(\mathcal{S}_{\mathcal{L}}(\mathscr{C}))^{\star 2} < \binom{k+1-|\mathcal{L}|}{2} \quad \text{and} \quad \dim(\mathcal{S}_{\mathcal{L}}(\mathscr{C}))^{\star 2} < n + w - |\mathcal{L}|. \tag{17}$$

Thanks to Theorem 4, we know that (17) is satisfied as soon as

$$2(k+w-|\mathcal{L}|)-1 < \binom{k+1-|\mathcal{L}|}{2} \quad \text{and} \quad 2(k+w-|\mathcal{L}|)-1 < n+w-|\mathcal{L}|. \tag{18}$$

We will now find the values $|\mathcal{L}|$ for which the inequalities of (18) are satisfied.

*First Inequality.* In order to determine when the first inequality of (18) is verified, let us denote

$$k' \overset{\text{def}}{=} k - |\mathcal{L}|.$$

Inequality (18) becomes $4k' - 2 + 4w < k'^2 + k'$, or equivalently $k'^2 - 3k' - 4w + 2 > 0$, which after a resolution leads to $k' > \frac{3+\sqrt{16w+1}}{2}\cdot$

Hence, we have:

$$|\mathcal{L}| < k - \frac{3+\sqrt{16w+1}}{2}\cdot \tag{19}$$

*Second Inequality.* The second inequality of (18) is equivalent to

$$|\mathcal{L}| \geqslant w + 2k - n. \tag{20}$$

*Conditions to Verify Both Inequalities.* Putting inequalities (19) and (20) together gives that $|\mathcal{L}|$ should satisfy

$$w + 2k - n \leqslant |\mathcal{L}| < k - \frac{3 + \sqrt{16w+1}}{2}.$$

We can therefore find an appropriate $\mathcal{L}$ if and only if

$$w + 2k - n < k - \frac{3 + \sqrt{16w+1}}{2},$$

which is equivalent to

$$n - k > w + \frac{3 + \sqrt{16w+1}}{2} = w + O(\sqrt{w}).$$

In other words, the distinguisher works up to values of $w$ that are close to the second choice $n - k = w$. From now on, we set

$$\ell_{\min} \overset{\text{def}}{=} w + 2k - n \qquad \text{and} \qquad \ell_{\max} \overset{\text{def}}{=} \left\lceil k - \frac{3 + \sqrt{16w+1}}{2} - 1 \right\rceil. \tag{21}$$

*Practical Results.* We have run experiments using Magma [5] and Sage. For the parameters of Table 1, here are the intervals of possible values of $|\mathcal{L}|$ so that the code $\mathcal{S}_{\mathcal{L}}\left(\mathscr{C}\right)^{\star 2}$ has a non generic dimension:

- ID 1: $n = 532, k = 376, w = 96$, $|\mathcal{L}| \in [\![316, 354]\!]$;
- ID 3: $n = 846, k = 618, w = 144$, $|\mathcal{L}| \in [\![534, 592]\!]$;
- ID 5: $n = 1160, k = 700, w = 311$, $|\mathcal{L}| \in [\![551, 663]\!]$.

The interval always coincides with the theoretical interval $[\![\ell_{\min}, \ell_{\max}]\!]$.

## 6 The Attack

In this section we will show how to find an equivalent private key $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{A}, \boldsymbol{P})$ defining the same code.

We assume that all the matrices $\boldsymbol{A}_s = \begin{pmatrix} a_s & b_s \\ c_s & d_s \end{pmatrix}$ appearing in the definition of the scheme in Subsect. 3.1 are such that $c_s d_s \neq 0$. We explain in Sect. 6.7 how to deal with the special case $c_s d_s = 0$. Note that this corresponds to a case where $\mathcal{I}_{\mathrm{R}} = \emptyset$ and $\mathcal{I}_{\mathrm{GRS}}^2 = \emptyset$.

*Remark 5.* In the present section where we the goal is to recover the permutation, we no longer work under Assumption 5.

### 6.1 Outline of the Attack

In summary, the attack works as follows.

1. Compute the interval $[\![\ell_{\min}, \ell_{\max}]\!]$ of the distinguisher and choose $\ell$ in the middle of the distinguisher interval. Ensure $\ell < \ell_{\max}$.
2. For several sets of indices $\mathcal{L} \subseteq [\![1, n+w]\!]$ such that $|\mathcal{L}| = \ell$, compute $\mathcal{S}_{\mathcal{L}}(\mathscr{C})$ and identify pairs of twin positions contained in $[\![1, n+w]\!]$. Repeat this process until identifying all pairs of twin positions, as detailed in Sect. 6.2.
3. Puncture the twin positions in order to get a GRS code and recover its structure using the Sidelnikov Shestakov attack [15].
4. For each pair of twin positions, recover the corresponding 2 × 2 non-singular matrix $A_i$, as explained in Sect. 6.6.
5. Finish to recover the structure of the underlying GRS code.

### 6.2 Identifying Pairs of Twin Positions

Let $\mathcal{L} \subseteq [\![1, n+w]\!]$ be such that both $|\mathcal{L}|$ and $|\mathcal{L}|+1$ are contained in the distinguisher interval. We compare the dimension of $(\mathcal{S}_{\mathcal{L}}(\mathscr{C}))^{\star 2}$ with the dimension of $\left(\mathcal{P}_{\{i\}}(\mathcal{S}_{\mathcal{L}}(\mathscr{C}))\right)^{\star 2}$ for all positions $i$ in $[\![1, n+w]\!] \setminus \mathcal{L}$.

- If $i \in \mathcal{I}_{\mathrm{GRS}}$ (see (2), (8) and (12) for the definition), puncturing does not affect the dimension of the square code:

$$\dim (\mathcal{S}_{\mathcal{L}}(\mathscr{C}))^{\star 2} = \dim \left(\mathcal{P}_{\{i\}}(\mathcal{S}_{\mathcal{L}}(\mathscr{C}))\right)^{\star 2}.$$

- If $i \in \mathcal{I}_{\mathrm{PR}}$ (see (6) for a definition) and $\tau(i) \in \mathcal{L}$, then according to Lemma 3, the position $i$ is "derandomised" in $\mathcal{S}_{\mathcal{L}}(\mathscr{C})$ and hence behaves like a GRS position in the shortened code. Therefore, very similarly to the previous case, the dimension does not change.
- If $i \in \mathcal{I}_{\mathrm{PR}}$ and $\tau(i) \notin \mathcal{L}$, in $\mathcal{S}_{\mathcal{L}}(\mathscr{C})$, the two corresponding columns behave like random ones. Assuming that the inequality of Theorem 4 is an equality, which almost always holds when no pair of twin positions is degenerate (see Sect. 6.7 and Remark 4), then, according to Lemma 5, puncturing $\mathcal{S}_{\mathcal{L}}(\mathscr{C})^{\star 2}$ at $i$ (resp. $\tau(i)$) reduces its dimension. Therefore,

$$\dim \left(\mathcal{P}_{\{i\}}(\mathcal{S}_{\mathcal{L}}(\mathscr{C}))\right)^{\star 2} = \dim \left(\mathcal{P}_{\{\tau(i)\}}(\mathcal{S}_{\mathcal{L}}(\mathscr{C}))\right)^{\star 2} = \dim (\mathcal{S}_{\mathcal{L}}(\mathscr{C}))^{\star 2} - 1.$$

  If some pair of twin positions is degenerate, the non-degenerate ones can be identified in the same way.

This provides a way to identify any position in $[\![1, n+w]\!] \setminus \mathcal{L}$ having a twin which also lies in $[\![1, n+w]\!] \setminus \mathcal{L}$: by searching zero columns in a parity-check matrix of $\mathcal{S}_{\mathcal{L}}(\mathscr{C})^{\star 2}$, we obtain the set $\mathcal{T}_{\mathcal{L}} \subset [\![1, n+w]\!] \setminus \mathcal{L}$ of even cardinality of all the positions having their twin in $[\![1, n+w]\!] \setminus \mathcal{L}$:

$$\mathcal{T}_{\mathcal{L}} \stackrel{\text{def}}{=} \bigcup_{\{i,\tau(i)\} \subseteq [\![1, n+w]\!] \setminus \mathcal{L}} \{i, \tau(i)\}.$$

Once these positions are identified, we can associate each such position to its twin. This can be done as follows. Take $i \in \mathcal{T}_{\mathcal{L}}$ and consider the code $\mathcal{S}_{\mathcal{L}\cup\{i\}}(\mathscr{C})$. The column corresponding to the twin position $\tau(i)$ has been derandomised and hence will not give a zero column in a parity-check matrix of $\left(\mathcal{S}_{\mathcal{L}\cup\{i\}}(\mathscr{C})\right)^{\star 2}$, so puncturing the corresponding column will not affect the dimension.

This process can be iterated by using various shortening sets $\mathcal{L}$ until obtaining $w$ pairs of twin positions. It is readily seen that considering $O(1)$ such sets is enough to recover all pairs with very large probability.

### 6.3 Recovering the Remainder of the Code

As soon as all the pairs of twin positions are identified, consider the code $\mathcal{P}_{\mathcal{I}_{\mathrm{PR}}}(\mathscr{C})$ punctured at $\mathcal{I}_{\mathrm{PR}}$. Since the randomised positions have been punctured this code is nothing but a GRS code and, applying the Sidelnikov Shestakov attack [15], we recover a pair $\boldsymbol{a}, \boldsymbol{b}$ such that $\mathcal{P}_{\mathcal{I}_{\mathrm{PR}}}(\mathscr{C}) = \mathbf{GRS}_k(\boldsymbol{a}, \boldsymbol{b})$.

### 6.4 Joining a Pair of Twin Positions : The Code $\mathscr{C}^{(i)}$

To recover the remaining part of the code we will consider iteratively the pairs of twin positions. We recall that $\mathcal{I}_{\mathrm{PR}}$ corresponds to the set of positions having a twin. Let $\{i, \tau(i)\}$ be a pair of twin positions and consider the code

$$\mathscr{C}^{(i)} \stackrel{\mathrm{def}}{=} \mathcal{P}_{[\![1,n]\!]\setminus(\mathcal{I}_{\mathrm{GRS}}\cup\{i,\tau(i)\})}(\mathscr{C}).$$

In this code, any position is GRS but positions $i$ and $\tau(i)$. Hence, for any codeword $\boldsymbol{c} \in \mathscr{C}^{(i)}$ we have:

$$\begin{aligned} c_i &= a y_j f(x_j) + c\psi_j(f) \\ c_{\tau(i)} &= b y_j f(x_j) + d\psi_j(f) \end{aligned} \tag{22}$$

for some integer $j \in [\![n-w+1, n]\!]$, where $\psi_j$ and $\boldsymbol{A} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ are defined as in (4) and (5).

Note that we do not need to recover exactly $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{A}, \boldsymbol{P})$. We need to recover a 4-tuple $(\boldsymbol{x}', \boldsymbol{y}', \boldsymbol{A}', \boldsymbol{P}')$ which describes the same code. Thus, without loss of generality, after possibly replacing $a$ by $ay_j$ and $b$ by $by_j$, one can suppose that $y_j = 1$. Moreover, after possibly replacing $\psi_j$ by $d\psi_j$, one can suppose that $d = 1$. Recall that in this section we suppose that $cd \neq 0$.

Thanks to these simplifying choices, (22) becomes

$$\begin{aligned} c_i &= a f(x_j) + c\psi_j(f) \\ c_{\tau(i)} &= b f(x_j) + \psi_j(f). \end{aligned}$$

### 6.5 Shortening $\mathscr{C}^{(i)}$ at the Last Position to Recover $x_j$

If we shorten $\mathscr{C}^{(i)}$ at the $\tau(i)$-th position, according to Lemma 3, it will "derandomise" the $i$-th position (it implies $\psi_j(f) = -bf(x_j)$) and any $\boldsymbol{c} \in \mathcal{S}_{\{\tau(i)\}}\left(\mathscr{C}^{(i)}\right)$ verifies

$$c_i = (a - bc)f(x_j).$$

Since the support $x_j$ and multiplier $y_j$ are known at all the positions of $\mathscr{C}^{(i)}$ but the two PR ones, for any codeword $\boldsymbol{c} \in \mathcal{S}_{\{\tau(i)\}}\left(\mathscr{C}^{(i)}\right)$, one can find the polynomial $f \in \mathbb{F}_q[x]_{<k}$ whose evaluation provides $\boldsymbol{c}$. Therefore, by collecting a basis of codewords in $\mathcal{S}_{\{\tau(i)\}}\left(\mathscr{C}^{(i)}\right)$ and the corresponding polynomials, we can recover the values of $x_j$ and $a - bc$.

### 6.6 Recovering the 2 × 2 Matrix

Once we have $x_j$ we need to recover the matrix

$$\boldsymbol{A} = \begin{pmatrix} a & b \\ c & 1 \end{pmatrix}.$$

Note that, its determinant $\det \boldsymbol{A} = a - bc$ has already been obtained in the previous section. First, one can guess $b$ as follows. Let $\boldsymbol{G}^{(i)}$ be a generator matrix of $\mathscr{C}^{(i)}$. As in the previous section, by interpolation, one can compute the polynomials $f_1, \dots, f_k$ whose evaluations provide the rows of $\boldsymbol{G}^{(i)}$. Consider the column vector

$$\boldsymbol{v} \stackrel{\text{def}}{=} \begin{pmatrix} f_1(x_j) \\ \vdots \\ f_k(x_j) \end{pmatrix}$$

and denote by $\boldsymbol{v}_i$ and $\boldsymbol{v}_{\tau(i)}$ the columns of $\boldsymbol{G}^{(i)}$ corresponding to positions $c_i$ and $c_{\tau(i)}$:

$$\boldsymbol{v}_i = \begin{pmatrix} af_1(x_j) + c\psi_j(f_1) \\ \vdots \\ af_k(x_j) + c\psi_j(f_k) \end{pmatrix} \quad \text{and} \quad \boldsymbol{v}_{\tau(i)} = \begin{pmatrix} bf_1(x_j) + \psi_j(f_1) \\ \vdots \\ bf_k(x_j) + \psi_j(f_k) \end{pmatrix}.$$

Next, search $\lambda \in \mathbb{F}_q$ such that $\boldsymbol{v}_i - \lambda \boldsymbol{v}_{\tau(i)}$ is collinear to $\boldsymbol{v}$. This relation of collinearity can be expressed in terms of cancellation of some 2 × 2 determinants which are polynomials of degree 1 in $\lambda$. Their common root is nothing but $c$.

Finally, we can find the pair $(a, b)$ by searching the pairs $(\lambda, \mu)$ such that

(i) $\lambda - c\mu = \det \boldsymbol{A}$;
(ii) $\boldsymbol{v}_i - \lambda \boldsymbol{v}$ and $\boldsymbol{v}_{\tau(i)} - \mu \boldsymbol{v}$ are collinear.

Here the relation of collinearity will be expressed as the cancellation of 2 × 2 determinants which are linear combinations of $\lambda, \mu$ and $\lambda\mu$ and elementary elimination process provides us with the value of the pair $(a, b)$.

### 6.7 How to Treat the Case of Degenerate Twin Positions?

Recall that a pair of twin positions $i, \tau(i)$ is such that any codeword $\boldsymbol{c} \in \mathscr{C}$ has $i$-th and $\tau(i)$-th entries of the form:

$$\boldsymbol{c}_i = ay_j f(x_j) + b\psi_j(f) \qquad \boldsymbol{c}_{\tau(i)} = cy_j f(x_j) + d\psi_j(f).$$

This pair is said to be *degenerate* if either $b$ or $d$ is zero. In such a situation, some of the steps of the attack cannot be applied. In what follows, we explain how this rather rare issue can be addressed.

If either $b$ or $d$ is zero, then one of the positions is actually a pure GRS position while the other one is PR but the process explained in the article does not manage to associate the two twin columns.

Suppose without loss of generality that $b = 0$. In the first part if the attack, when we collect pairs of twin positions, the position $\tau(i)$ will be identified as PR with no twin sister *a priori*. To find its twin sister, we can proceed as follows. For any GRS position $j$ replace the $j$-th column $\boldsymbol{v}_j$ of a generator matrix $\boldsymbol{G}$ of $\mathscr{C}$ by an arbitrary linear combination of $\boldsymbol{v}_j$ and the $\tau(i)$-th column, this will "pseudo–randomise" this column and if the $j$-th column is the twin of the $\tau(i)$-th one, this will be detected by the process of shortening, squaring and searching zero columns in the parity check matrix.

## 7 Complexity of the Attack

The most expensive part of the attack is the step consisting in identifying pairs of twin positions. Recall that, from [8], the computation of the square of a code of length $n$ and dimension $k$ costs $O(k^2n^2)$ operations in $\mathbb{F}_q$. We need to compute the square of a code $O(w)$ times, because there are $w$ pairs of twin positions. Hence this step has a total complexity of $O(wn^2k^2)$ operations in $\mathbb{F}_q$. Note that the actual dimension of the shortened codes is significantly less than $k$ and hence the previous estimate is overestimated.

The cost of the Sidelnikov Shestakov attack is that of a Gaussian elimination, namely $O(nk^2)$ operations in $\mathbb{F}_q$ which is negligible compared to the previous step. The cost of the final part is also negligible compared to the computation of the squares of shortened codes. This provides an overall complexity in $O(wn^2k^2)$ operations in $\mathbb{F}_q$.

## Conclusion

We presented a polynomial time key-recovery attack based on a square code distinguisher against the public key encryption scheme RLCE. This attack allows us to break all the so-called *odd ID* parameters suggested in [17]. Namely, the attack breaks the parameter sets for which the number $w$ of random columns was strictly less than $n - k$. Our analysis suggests that, for this kind of distinguisher by squaring shortenings of the code, the case $w = n - k$ is the critical one. The *even ID* parameters of [17], for which the relation $w = n-k$ always holds, remain out of the reach of our attack.

**Acknowledgements.** The authors are supported by French *Agence nationale de la* recherche grants ANR-15-CE39-0013 *Manta*, ANR-17-CE39-0007 *CBCrypt* and by the Commission of the European Communities through the Horizon 2020 program under project number 645622 (PQCRYPTO). Computer aided calculations have been performed using softwares Sage and Magma [5].

## References

1. Baldi, M., Bianchi, M., Chiaraluce, F., Rosenthal, J., Schipani, D.: Enhanced public key security for the McEliece cryptosystem. J. Cryptol. **29**(1), 1–27 (2016). https://doi.org/10.1007/s00145-014-9187-8
2. Bardet, M., Chaulet, J., Dragoi, V., Otmani, A., Tillich, J.-P.: Cryptanalysis of the McEliece public key cryptosystem based on polar codes. In: Takagi, T. (ed.) PQCrypto 2016. LNCS, vol. 9606, pp. 118–143. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-29360-8_9
3. Berger, T.P., Loidreau, P.: Security of the Niederreiter form of the GPT public-key cryptosystem. In: Proceedings IEEE International Symposium on Information Theory - ISIT 2002, p. 267. IEEE, June 2002
4. Bernstein, D.J., et al.: Classic McEliece: conservative code-based cryptography, November 2017. https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/round-1/submissions/Classic_McEliece.zip, first round submission to the NIST post-quantum cryptography call
5. Bosma, W., Cannon, J., Playoust, C.: The Magma algebra system I: the user language. J. Symbolic Comput. **24**(3/4), 235–265 (1997)
6. Cascudo, I., Cramer, R., Mirandola, D., Zémor, G.: Squares of random linear codes. IEEE Trans. Inform. Theory **61**(3), 1159–1173 (2015)
7. Chizhov, I.V., Borodin, M.A.: Effective attack on the McEliece cryptosystem based on Reed-Muller codes. Discrete Math. Appl. **24**(5), 273–280 (2014)
8. Couvreur, A., Gaborit, P., Gauthier-Umaña, V., Otmani, A., Tillich, J.P.: Distinguisher-based attacks on public-key cryptosystems using Reed-Solomon codes. Des. Codes Cryptogr. **73**(2), 641–666 (2014). https://doi.org/10.1007/s10623-014-9967-z
9. Couvreur, A., Márquez-Corbella, I., Pellikaan, R.: Cryptanalysis of McEliece cryptosystem based on algebraic geometry codes and their subcodes. IEEE Trans. Inform. Theory **63**(8), 5404–5418 (2017)
10. Couvreur, A., Otmani, A., Tillich, J.P.: Polynomial time attack on wild McEliece over quadratic extensions. IEEE Trans. Inform. Theory **63**(1), 404–427 (2017)
11. Couvreur, A., Otmani, A., Tillich, J.-P., Gauthier–Umaña, V.: A polynomial-time attack on the BBCRS scheme. In: Katz, J. (ed.) PKC 2015. LNCS, vol. 9020, pp. 175–193. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46447-2_8
12. Faugère, J.C., Gauthier, V., Otmani, A., Perret, L., Tillich, J.P.: A distinguisher for high rate McEliece cryptosystems. IEEE Trans. Inform. Theory **59**(10), 6830–6844 (2013)
13. Huffman, W.C., Pless, V.: Fundamentals of Error-Correcting Codes. Cambridge University Press, Cambridge (2003). https://doi.org/10.1017/CBO9780511807077
14. McEliece, R.J.: A Public-Key System Based on Algebraic Coding Theory, pp. 114–116. Jet Propulsion Lab (1978). DSN Progress Report 44
15. Sidelnikov, V.M., Shestakov, S.: On the insecurity of cryptosystems based on generalized Reed-Solomon codes. Discrete Math. Appl. **1**(4), 439–444 (1992)

16. Wang, Y.: Quantum resistant random linear code based public key encryption scheme RLCE. In: Proceedings of the IEEE International Symposium on Information Theory - ISIT 2016, pp. 2519–2523. IEEE, Barcelona, July 2016. https://doi.org/10.1109/ISIT.2016.7541753
17. Wang, Y.: RLCE-KEM (2017). http://quantumca.org, first round submission to the NIST post-quantum cryptography call
18. Wieschebrink, C.: Two NP-complete problems in coding theory with an application in code based cryptography. In: Proceedings IEEE International Symposium Information Theory - ISIT, pp. 1733–1737 (2006)
19. Wieschebrink, C.: Cryptanalysis of the Niederreiter public key scheme based on GRS subcodes. In: Sendrier, N. (ed.) PQCrypto 2010. LNCS, vol. 6061, pp. 61–72. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-12929-2_5