Distinguisher-Based Attacks on Public-Key Cryptosystems Using Reed-Solomon Codes

A. Couvreur[*], P. Gaborit[†], V. Gautier[‡], A. Otmani[§] and J.-P. Tillich[¶]

April 14, 2013

Abstract

The purpose of this paper is to demonstrate that a distinguisher of Reed-Solomon codes based on the square code construction leads to the cryptanalysis of several cryptosystems relying on them. These schemes are respectively (i) a homomorphic encryption scheme proposed by Bogdanov and Lee; (ii) a variation of the McEliece cryptosystem proposed by Baldi et *al.* which firstly uses Reed-Solomon codes instead of Goppa codes and secondly, adds a rank 1 matrix to the permutation matrix; (iii) Wieschebrink's variant of the McEliece cryptosystem which consists in concatenating a few random columns to a generator matrix of a secretly chosen generalized Reed-Solomon code. XXX signaler l'attaque sur les GRSXXXX

1 Reed-Solomon Codes and the Square Code Construction

We recall in this section a few relevant results and definitions from coding theory and bring in the fundamental notion which is used in both attacks, namely the square code construction. Generalized Reed-Solomon codes (GRS in short) form a special case of codes with a very powerful low complexity decoding algorithm. It will be convenient to use the definition of these codes as *evaluation codes*

Definition 1 (Generalized Reed-Solomon code)**.** Let k and n be integers such that $1 \leqslant k < n \leqslant q$ where q is a power of a prime number. The generalized Reed-Solomon code $\mathbf{GRS}_k(\boldsymbol{x}, \boldsymbol{y})$ of dimension k is associated to a pair $(\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{F}_q^n \times \mathbb{F}_q^n$ where $\boldsymbol{x}$ is an n-tuple of distinct elements of $\mathbb{F}_q$ and the entries y_i are arbitrary nonzero elements in $\mathbb{F}_q$. $\mathbf{GRS}_k(\boldsymbol{x}, \boldsymbol{y})$ is defined as:

$$\mathbf{GRS}_k(\boldsymbol{x}, \boldsymbol{y}) \stackrel{\text{def}}{=} \Big\{ (y_1 p(x_1), \ldots, y_n p(x_n)) : p \in \mathbb{F}_q[X], \deg p < k \Big\}.$$

[*]GRACE Project, INRIA Saclay & LIX, CNRS UMR 7161 - École Polytechnique, 91120 Palaiseau Cedex, France. alain.couvreur@lix.polytechnique.fr

[†]XLIM, CNRS UMR 7252 - Université de Limoges, 123 avenue Albert Thomas, 87060 Limoges Cedex, France. philippe.gaborit@unilim.fr

[‡]Normandie Univ, France; UNICAEN, GREYC, F-14050 Caen, France; CNRS, UMR 6072, F-14032 Caen, France. valerie.gauthier01@unicaen.fr

[§]Normandie Univ, France; UR, LITIS, F-76821 Mont-Saint-Aignan, France. ayoub.otmani@univ-rouen.fr

[¶]SECRET Project - INRIA Rocquencourt, 78153 Le Chesnay Cedex, France. jean-pierre.tillich@inria.fr

Remark 1. Reed-Solomon codes correspond to the case where $y_i = 1$ for all i.

The first work that suggested to use GRS code in a public-key cryptosystem scheme was [Nie86]. But Sidelnikov and Shestakov discovered in [SS92] that this scheme is insecure. They namely showed that for any GRS code it is possible to recover in polynomial time a couple $(\boldsymbol{x}, \boldsymbol{y})$ which defines it. This is all that is needed to decode efficiently such codes and is therefore enough to break the Niederreiter cryptosystem suggested in [Nie86] or any McEliece type cryptosystem [McE78] that uses GRS codes instead of binary Goppa codes.

Definition 2 (Componentwise products)**.** Given two vectors $\boldsymbol{a} = (a_1, \ldots, a_n)$ and $\boldsymbol{b} = (b_1, \ldots, b_n) \in \mathbb{F}_q^n$, we denote by $\boldsymbol{a} \star \boldsymbol{b}$ the componentwise product

$$\boldsymbol{a} \star \boldsymbol{b} \stackrel{\text{def}}{=} (a_1 b_1, \ldots, a_n b_n)$$

Definition 3 (Product of codes & square code)**.** Let $\mathscr{A}$ and $\mathscr{B}$ be two codes of length n. The *star product code* denoted by $\mathscr{A} \star \mathscr{B}$ of $\mathscr{A}$ and $\mathscr{B}$ is *the vector space spanned by all products* $\boldsymbol{a} \star \boldsymbol{b}$ where $\boldsymbol{a}$ and $\boldsymbol{b}$ range over $\mathscr{A}$ and $\mathscr{B}$ respectively. When $\mathscr{B} = \mathscr{A}$ then $\mathscr{A} \star \mathscr{A}$ is called the *square code* of $\mathscr{A}$ and is rather denoted by $\mathscr{A}^2$.

It is clear that $\mathscr{A} \star \mathscr{B}$ is also generated by the $\boldsymbol{a}_i \star \boldsymbol{b}_j$'s where the $\boldsymbol{a}_i$'s and the $\boldsymbol{b}_j$'s form a basis of $\mathscr{A}$ and $\mathscr{B}$ respectively. Therefore, we have the following result.

Proposition 4. *Let $\mathscr{A}$ and $\mathscr{B}$ be two codes of length n, then*

1. $\dim(\mathscr{A} \star \mathscr{B}) \leqslant \dim(\mathscr{A}) \dim(\mathscr{B})$
2. $\dim(\mathscr{A}^2) \leqslant \binom{\dim(\mathscr{A}) + 1}{2}$.

The importance of the square code construction will become clear when we compare the dimensions of square codes obtained through a *structured* code and random code and one major question is to know what one should expect. The following Proposition 5 shows that when applied to GRS codes, the dimension of the square code is roughly twice as large as the dimension of the underlying code.

Proposition 5. $\mathbf{GRS}_k(\boldsymbol{x}, \boldsymbol{y})^2 = \mathbf{GRS}_{2k-1}(\boldsymbol{x}, \boldsymbol{y} \star \boldsymbol{y})$.

Proof. This follows immediately from the definition of a GRS code as an evaluation code since the star product of two elements $\boldsymbol{c} = (y_1 p(x_1), \ldots, y_n p(x_n))$ and $\boldsymbol{c}' = (y_1 q(x_1), \ldots, y_n q(x_n))$ of $\mathbf{GRS}_k(\boldsymbol{x}, \boldsymbol{y})$ where p and q are two polynomials of degree at most $k-1$ is of the form

$$\boldsymbol{c} \star \boldsymbol{c}' = \big(y_1^2 p(x_1) q(x_2), \ldots, y_n^2 p(x_n) q(x_n)\big) = \big(y_1^2 r(x_1), \ldots, y_n^2 r(x_n)\big)$$

where r is a polynomial of degree $\leqslant 2k-2$. Conversely, any element of the form $\big(y_1^2 r(x_1), \ldots, y_n^2 r(x_n)\big)$ where r is a polynomial of degree less than or equal to $2k-2$ is a linear combination of star products of two elements of $\mathbf{GRS}_k(\boldsymbol{x}, \boldsymbol{y})$. □

This proposition shows that the square code is only of dimension $2k-1$ when $2k-1 \leqslant n$. This property can also be used in the case $2k-1 > n$. To see this, consider the dual of the Reed-Solomon code itself a Reed-Solomon code [MS86, Theorem 4, p.304]

Proposition 6. $\mathbf{GRS}_k(\boldsymbol{x}, \boldsymbol{y})^\perp = \mathbf{GRS}_{n-k}(\boldsymbol{x}, \boldsymbol{y}')$ *where the length of* $\mathbf{GRS}_k(\boldsymbol{x}, \boldsymbol{y})$ *is n and $\boldsymbol{y}'$ is a certain element of $\mathbb{F}_q^n$ depending only on $\boldsymbol{x}$ and $\boldsymbol{y}$.*

This result is clearly different what would obtain if random linear codes are taken. Indeed, we expect that the square code when applied to a random linear code of dimension k should be a code of dimension of order $\min\left\{\binom{k+1}{2}, n\right\}$. Actually it can be shown by the proof technique of [FGO+11] the following result (see also [MCP12]).

Proposition 7 ([FGO+11])**.** *Let k and n be non-negative integers such that $k = o(n^{1/2})$ and consider a random $(n-k) \times (n-k)$ matrix $\boldsymbol{R}$ where each entry is independently and uniformly drawn from $\mathbb{F}_q$. Let $\mathscr{R}$ be the linear code defined by the generator matrix $(\boldsymbol{I}_k \mid \boldsymbol{R})$ where $\boldsymbol{I}_k$ is the $k \times k$ identity matrix.*

For any ε such that $0 < \varepsilon < 1$ and any $\alpha > 0$, we have as k tends to $+\infty$:

$$\mathsf{Prob}\left(\dim\left(\mathscr{R}^2\right) \leqslant \binom{k+1}{2}\left(1 - \alpha k^{-\varepsilon}\right)\right) = o(1)$$

Therefore $\mathbf{GRS}_k(\boldsymbol{x}, \boldsymbol{y})$ can be distinguished from a random linear code of the same dimension by computing the dimension of the associated square codes. This phenomenon was already observed in [FGO+11] for q-ary alternant codes (in particular Goppa codes) at very high rates. Let us note that even when $2k - 1 > n$ it is still possible to distinguish GRS codes from random codes by focusing on $\left(\mathbf{GRS}_k(\boldsymbol{x}, \boldsymbol{y})^{\perp}\right)^2$. We have in this case:

$$\left(\mathbf{GRS}_k(\boldsymbol{x}, \boldsymbol{y})^{\perp}\right)^2 = \mathbf{GRS}_{n-k}(\boldsymbol{x}, \boldsymbol{y}')^2 = \mathbf{GRS}_{2n-2k-1}(\boldsymbol{x}, \boldsymbol{y}' \star \boldsymbol{y}')$$

which is a code of dimension $2n - 2k - 1$.

The star product of codes has been used for the first time by Wieschebrink to cryptanalyze a McEliece-like scheme [BL05] based on subcodes of Reed-Solomon codes [Wie10]. The use of the star product is nevertheless different in [Wie10] from the way we use it here. In Wieschebrink's paper, the star product is used to identify for a certain subcode $\mathscr{C}$ of a GRS code $\mathbf{GRS}_k(\boldsymbol{x}, \boldsymbol{y})$ a possible pair $(\boldsymbol{x}, \boldsymbol{y})$. This is achieved by computing $\mathscr{C}^2$ which turns out to be $\mathbf{GRS}_k(\boldsymbol{x}, \boldsymbol{y})^2 = \mathbf{GRS}_{2k-1}(\boldsymbol{x}, \boldsymbol{y} \star \boldsymbol{y})$. The Sidelnikov and Shestakov algorithm is then used on $\mathscr{C}^2$ to recover a possible $(\boldsymbol{x}, \boldsymbol{y} \star \boldsymbol{y})$ pair to describe $\mathscr{C}^2$ as a GRS code, and hence, a pair $(\boldsymbol{x}, \boldsymbol{y})$ is deduced for which $\mathscr{C} \subset \mathbf{GRS}_k(\boldsymbol{x}, \boldsymbol{y})$.

In this work, we use directly the fact that the square code of a somehow GRS code has an abnormally small dimension. When applied on several public-key encryption schemes [Nie86, Wie06, BBC+11, BL11], we achieve different goals but it always results in an efficient key-recovery attack. For instance, computing the dimensions of the square of various:

- subcodes of the public code permits to detect random columns in the generator matrix of the public code of Wieschebrink's cryptosystem [Wie06] (Section 2);
- punctured versions of the public code in the Bogdanov-Lee cryptosystem [BL11] enables to retrieve the Reed-Solomon part of the public code (Section 3).

In the case of the scheme [BBC+11], it is possible to identify a certain subcode that is both included in a GRS code and the public code (Section 4). In the case of a McEliece-like cryptosystem based on a GRS code [Nie86], it enables to get a full filtration by means of GRS subcodes, so that the structure of the public code as a GRS code is recovered (Section 5).

2 Wieschebrink's Scheme

In [Wie06] Wieschebrink suggests a variant of the McEliece cryptosystem based on GRS codes whose purpose was to resist to the Sidelnikov–Shestakov attack. The idea of this proposal is to use the generator matrix of a GRS code in which a small number of randomly chosen columns are inserted. More precisely, let $\boldsymbol{G}$ be a generator matrix of a GRS code of length n and dimension k defined over $\mathbb{F}_q$. Let $C_1, \ldots, C_r$ be r column vectors in $\mathbb{F}_q^k$ drawn uniformly at random and let $\boldsymbol{G}'$ be the matrix obtained by concatenating $\boldsymbol{G}$ and the columns $C_1, \ldots, C_r$. Choose $\boldsymbol{S}$ to be a $k \times k$ random invertible matrix and let $\boldsymbol{Q}$ be a an $(n+r) \times (n+r)$ permutation matrix. The public key of the scheme is

$$\boldsymbol{G_{pub}} \stackrel{\text{def}}{=} \boldsymbol{S}^{-1}\boldsymbol{G}'\boldsymbol{Q}^{-1}.$$

This cryptosystem can be cryptanalyzed if a description of the GRS code can be recovered from $\boldsymbol{G_{pub}}$. We give here a way to break this scheme in polynomial time which relies on two ingredients. The first one is given by

Lemma 8. *Let $\boldsymbol{G}'$ be a $k \times (n+r)$–matrix obtained by inserting r random columns in a generator matrix of an $[n,k]$ GRS code $\mathscr{C}$. Let $\mathscr{C}'$ be the corresponding code. Assume that $k < n/2$, then*

$$2k - 1 \leqslant \dim \mathscr{C}'^2 \leqslant 2k - 1 + r.$$

Proof. The first inequality comes from the fact that puncturing $\mathscr{C}'^2$ at the r positions corresponding to the added random columns yields the code $\mathscr{C}^2$ which is the square of an $[n,k]$ GRS code and hence an $[n, 2k-1]$ GRS code. To prove the upper bound, let $\mathscr{D}$ be the code with generator matrix $\boldsymbol{G}''$ obtained from $\boldsymbol{G}'$ by replacing the C_i's columns by all-zero columns and let $\mathscr{D}'$ be the code with generator matrix $\boldsymbol{G}'''$ obtained by replacing in $\boldsymbol{G}'$ all columns which are not the C_i's by zero columns. Since $\boldsymbol{G}' = \boldsymbol{G}'' + \boldsymbol{G}'''$ we have

$$\mathscr{C}' \subset \mathscr{D} + \mathscr{D}'. \tag{1}$$

Therefore

$$\begin{aligned} \mathscr{C}'^2 &\subset \left(\mathscr{D} + \mathscr{D}'\right)^2 \\ &\subset \mathscr{D}^2 + \mathscr{D}'^2 + \mathscr{D} \star \mathscr{D}' \\ &\subset \mathscr{D}^2 + \mathscr{D}'^2 \end{aligned}$$

where the last inclusion comes from the fact that $\mathscr{D} \star \mathscr{D}'$ is the zero subspace since $\mathscr{D}$ and $\mathscr{D}'$ have disjoint supports. The right-hand side inequality follows immediately from this, since $\dim \mathscr{D}^2 = 2k-1$ and $\dim \mathscr{D}'^2 \leqslant r$. □

Actually the right-hand inequality of Lemma 8 is sharp and with very high probability we observe that if $2k - r - 1 < n$ then

$$\dim \mathscr{C}'^2 = 2k - 1 + r.$$

This will be useful to detect the positions which correspond to the C_i's. We call such positions the *random positions* whereas the other positions are referred to as the *GRS positions*. We use in this case a shortening trick which relies upon the following well known fact.

Fact. Shortening a GRS code of parameters $[n,k]$ in $\ell \leqslant k$ positions gives a GRS code with parameters $[n-\ell, k-\ell]$.

An attack easily follows from these facts. First of all, let us consider the case when $2k-1+r \leqslant n$, then consider $\mathscr{C}'_i$ which is the shortened $\mathscr{C}'$ code at position i. Two cases can occur

– *i belongs to the random positions*, then we expect that the dimension of $\mathscr{C}'_i$ is given by

$$\dim \mathscr{C}'^2_i = 2k - 2 + r.$$

since $\mathscr{C}'_i$ is nothing but a k-dimensional GRS code with $r-1$ random columns inserted in its generator matrix.

– *i belongs to the GRS positions*, then $\mathscr{C}'_i$ is a $k-1$-dimensional GRS code with r random columns added to its generator matrix and we expect that

$$\dim \mathscr{C}'^2_i = 2k - 3 + r.$$

This gives a straightforward way to distinguish between the random positions and the GRS positions.

Consider now the case where $2k-1+r > n$. The point is to shorten $\mathscr{C}'$ in a positions in order to be able to apply again the same principle. Here a is chosen such that $a < k$ and $2(k-a)-1+r < n-a \implies a > 2k-1+r-n$. Notice that these conditions on a can be met as soon as $k > 2k+r-n \implies n > k+r$, which always holds true. Among these a positions, a_0 of them are random positions and $a_1 \stackrel{\text{def}}{=} a - a_0$ are GRS positions. This yields a GRS code of parameters $[n-a_1, k-a_1]$ to which $r-a_0$ random positions have been added (or more precisely this yields a code with generator matrix given by the generator matrix of a GRS code of size $(k-a_1)\times(n-a_1)$ with $r-a_0$ random columns added to it). Denote by $\mathscr{C}'_a$ this shortened code. Using the previous results, we get that with high probability,

$$\dim \mathscr{C}'^2_a = 2(k-a_1) - 1 + r - a_0$$

By this manner we get the value of $2a_1 + a_0$ and since $a = a_1 + a_0$ is already known we can deduce the values of a_0 and a_1. To identify which positions of $\mathscr{C}'_a$ are random positions and which ones are GRS positions we just use the previous approach by shortening $\mathscr{C}'_a$ in an additional position and checking whether or not the dimension decreases by one or two. This approach has been implemented in Magma and leads to identify easily all the random columns for the parameters suggested in [Wie06]. After identifying the random columns in the public generator matrix, it just remains to puncture the public code at these positions and to apply the Sidelnikov-Shestakov attack to completely break the scheme proposed in [Wie06].

3 Bogdanov-Lee Homomorphic Cryptosystem

3.1 Description of the Scheme

The cryptosystem proposed by Bogdanov and Lee in [BL11] is a public-key homomorphic encryption scheme based on linear codes. It encrypts a plaintext m from $\mathbb{F}_q$ into a ciphertext $\boldsymbol{c}$ that belongs to $\mathbb{F}_q^n$ where n is a given integer satisfying $n < q$. The key generation requires two non-negative integer ℓ, k such that $3\ell < n$ and $\ell < k$ together with a subset $L \subset \{1, \ldots, n\}$ of cardinality 3ℓ. A set of n distinct elements $x_1, \ldots, x_n$ from $\mathbb{F}_q^\times$ are generated at random. They serve to construct a $k \times n$ matrix $\boldsymbol{G}$ whose i-th column $\boldsymbol{G}_i^T$ $(1 \leqslant i \leqslant n)$ is defined by

$$\boldsymbol{G}_i^T \stackrel{\text{def}}{=} \begin{cases} (x_i, x_i^2, \ldots, x_i^\ell, 0, \ldots, 0) & \text{if } i \in L \\ \\ (x_i, x_i^2, \ldots, x_i^\ell, x_i^{\ell+1}, \ldots, x_i^k) & \text{if } i \notin L \end{cases},$$

where the symbol ^{T} stands for the transpose. The cryptosystem is defined as follows:

1. **Secret key.** $(L, \boldsymbol{G})$.
2. **Public key.** $\boldsymbol{P} \stackrel{\text{def}}{=} \boldsymbol{SG}$ where $\boldsymbol{S}$ is a $k \times k$ random invertible matrix over $\mathbb{F}_q$.
3. **Encryption.** The ciphertext $\boldsymbol{c} \in \mathbb{F}_q^n$ corresponding to $m \in \mathbb{F}_q$ is obtained as $\boldsymbol{c} \stackrel{\text{def}}{=} \boldsymbol{xP}+m\boldsymbol{1}+\boldsymbol{e}$ where $\boldsymbol{1} \in \mathbb{F}_q^n$ is the all-ones row vector, $\boldsymbol{x}$ is picked uniformly at random in $\mathbb{F}_q^k$ and $\boldsymbol{e}$ in $\mathbb{F}_q^n$ by choosing its components according to a certain distribution $\tilde{\eta}$.
4. **Decryption.** Solve the following linear system with unknowns $\boldsymbol{y} \stackrel{\text{def}}{=} (y_1, \ldots, y_n) \in \mathbb{F}_q^n$:

$$\boldsymbol{G}\boldsymbol{y}^T = 0, \ \sum_{i \in L} y_i = 1 \text{ and } y_i = 0 \text{ for all } i \notin L. \tag{2}$$

The plaintext is then $m = \sum_{i=1}^{n} y_i c_i$.

Let us explain here why the decryption algorithm outputs the correct plaintext when ℓ and n are chosen such that the entry e_i at position i of the error vector is zero when $i \in L$. If this property on $\boldsymbol{e}$ holds, notice that the linear system (2) has 3ℓ unknowns and $\ell + 1$ equations and since it is by construction of rank $\ell + 1$, it always admits at least one solution. Then observe that

$$\begin{aligned}
\sum_{i=1}^{n} y_i c_i &= (\boldsymbol{xP} + m\boldsymbol{1} + \boldsymbol{e})\boldsymbol{y}^T \\
&= (\boldsymbol{xP} + m\boldsymbol{1})\boldsymbol{y}^T \quad (\text{since } e_i = 0 \text{ if } i \in L \text{ and } y_i = 0 \text{ if } i \notin L) \\
&= \boldsymbol{xSG}\boldsymbol{y}^T + m\sum_{i=1}^{n} y_i \\
&= m \quad (\text{since } \boldsymbol{G}\boldsymbol{y}^T = 0 \text{ and } \sum_{i=1}^{n} y_i = 1).
\end{aligned}$$

The decryption algorithm will output the correct plaintext when ℓ and n are chosen such that the entry e_i at position i of the error vector is zero when $i \in L$. The distribution η which is used to draw at random the coordinates of $\boldsymbol{e}$ is chosen such that this property holds with very large probability. More precisely, the parameters k, q, ℓ and the noise distribution $\tilde{\eta}$ are chosen such that $q = \Omega\left(2^{n^\alpha}\right)$, $k = \Theta\left(n^{1-\alpha/8}\right)$, $\ell = \Theta\left(n^{\alpha/4}\right)$ and the noise distribution $\tilde{\eta}$ is the q-ary symmetric channel with noise rate[1] $\eta = \Theta\left(1/n^{1-\alpha/4}\right)$ where α is a in $(0, \frac{1}{4}]$. For further details see [BL11, §2.3]. It is readily checked that the probability that $e_i \neq 0$ for $i \in L$ is vanishing as n goes to infinity since it is upper-bounded by $\eta\ell = \Theta\left(\frac{n^{\alpha/4}}{n^{1-\alpha/4}}\right) = \Theta\left(n^{-1+\alpha/2}\right) = o(1)$.

3.2 An Efficient Key-Recovery Attack

The attack consists in first recovering the secret set L and from here, one finds directly a suitable vector $\boldsymbol{y}$ by solving the system

$$\boldsymbol{P}\boldsymbol{y}^T = 0, \ \sum_{i \in L} y_i = 1, \ y_i = 0 \text{ for all } i \notin L. \tag{3}$$

[1] It means that $\mathsf{Prob}(e_i = 0) = 1 - \eta$ and $\mathsf{Prob}(e_i = x) = \frac{\eta}{q-1}$ for any x in $\mathbb{F}_q$ different from zero.

Indeed, requiring that $\boldsymbol{P}\boldsymbol{y}^T = 0$ is equivalent to $\boldsymbol{S}\boldsymbol{G}\boldsymbol{y}^T = 0$ and since $\boldsymbol{S}$ is invertible this is equivalent to the equation $\boldsymbol{G}\boldsymbol{y}^T = 0$. Therefore System (3) is equivalent to the "secret" system (2). An attacker may therefore recover m without even knowing $\boldsymbol{G}$ just by outputting $\sum_i y_i c_i$ for any solution $\boldsymbol{y}$ of (3). In the following subsection, we will explain how L can be recovered from $\boldsymbol{P}$ in polynomial time.

Our attack which recovers L relies heavily on the fact that the public matrix may be viewed as a the generator matrix of a code $\mathscr{C}$ which is quite close to a generalized Reed-Solomon code (or to a Reed-Solomon code if a row consisting only of 1's is added to it). Notice that any punctured version of the code has also this property (a punctured code consists in keeping only a fixed subset of positions in a codeword). More precisely, let us introduce

Definition 9. For any $I \subset \{1, \ldots, n\}$ of cardinality $|I|$, the restriction of a code $\mathscr{A}$ of length n is the subset of $\mathbb{F}_q^{|I|}$ defined as $\mathscr{A}_I \stackrel{\text{def}}{=} \left\{ \boldsymbol{v} \in \mathbb{F}_q^{|I|} \mid \exists \boldsymbol{a} \in \mathscr{A}, \boldsymbol{v} = (a_i)_{i \in I} \right\}$.

The results about the unusual dimension of the square of a Reed-Solomon codes which are given in Section 1 prompt us to study the dimension of the square code $\mathscr{C}^2$ or more generally the dimension of $\mathscr{C}_I^2$. When I contains no positions in L, then $\mathscr{C}_I$ is nothing but a generalized Reed-Solomon code and we expect a dimension of $2k-1$ when $|I|$ is larger than $2k-1$. On the other hand, when there are positions in I which also belong to L we expect the dimension to become bigger and the dimension of $\mathscr{C}^2$ to behave as an increasing function of $|I \cap L|$. This is exactly what happens as shown in the proposition below.

Proposition 10. *Let I be a subset of $\{1, \ldots, n\}$ and set $J \stackrel{def}{=} I \cap L$. If the cardinality of I and J satisfy $|J| \leqslant \ell - 1$ and $|I| - |J| \geqslant 2k$ then*

$$\dim(\mathscr{C}_I^2) = 2k - 1 + |J| \,. \tag{4}$$

Proof. Set $a \stackrel{\text{def}}{=} |I| - |J|$ and $b \stackrel{\text{def}}{=} |I|$. After a suitable permutation of the support and the indexes of the x_j's, the code $\mathscr{C}_I$ has a generator matrix of the form

$$\begin{pmatrix} x_1 & x_2 & \cdots & x_a & x_{a+1} & \cdots & x_b \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ x_1^\ell & x_2^\ell & \cdots & x_a^\ell & x_{a+1}^\ell & \cdots & x_b^\ell \\ \\ x_1^{\ell+1} & x_2^{\ell+1} & \cdots & x_a^{\ell+1} & & & \\ \vdots & \vdots & & \vdots & & (0) & \\ x_1^k & x_2^k & \cdots & x_a^k & & & \end{pmatrix}$$

We define the maps

$$\Phi_I : \left\{ \begin{array}{ccc} \mathbb{F}_q[x] & \to & \mathbb{F}_q^b \\ P & \mapsto & (P(x_1), \ldots, P(x_b)) \end{array} \right. , \quad \Phi_{I\setminus J} : \left\{ \begin{array}{ccc} \mathbb{F}_q[x] & \to & \mathbb{F}_q^b \\ P & \mapsto & (P(x_1), \ldots, P(x_a), 0 \ldots, 0) \end{array} \right. .$$

We have the two following obvious lemmas.

Lemma 11. *Both maps Φ_I and $\Phi_{I\setminus J}$ are linear. In addition, their restrictions to the vector space $< x^2, \ldots, x^{2k} >$ are injective.*

Proof. It is sufficient to prove that the restriction of $\Phi_{I\setminus J}$ is injective. It is an elementary consequence of polynomial interpolation, since $a = |I| - |J|$ is assumed to be be larger than $2k$. □

Lemma 12. *For all $P, Q \in \mathbb{F}_q[x]$, we have*

$$\begin{aligned} \Phi_I(P) \star \Phi_I(Q) &= \Phi_I(PQ) && (5)\\ \Phi_{I\setminus J}(P) \star \Phi_{I\setminus J}(Q) &= \Phi_{I\setminus J}(PQ) && (6)\\ \Phi_I(P) \star \Phi_{I\setminus J}(Q) &= \Phi_{I\setminus J}(PQ) && (7) \end{aligned}$$

Clearly, we have

$$\mathscr{C}_I = \Phi_I\left(< x, \ldots, x^{\ell} >\right) \oplus \Phi_{I\setminus J}\left(< x^{\ell+1}, \ldots, x^k >\right). \quad (8)$$

Using (5), (6) and (7), we get

$$\begin{aligned} \mathscr{C}_I^2 &= \Phi_I\left(< x, \ldots, x^{\ell} >\right)^2 + \Phi_{I\setminus J}\left(< x^{\ell+1}, \ldots, x^k >\right)^2 \\ &\qquad + \Phi_I\left(< x, \ldots, x^{\ell} >\right) \star \Phi_{I\setminus J}\left(< x^{\ell+1}, \ldots, x^k >\right) \\ &= \Phi_I\left(< x^2, \ldots, x^{2\ell} >\right) + \Phi_{I\setminus J}\left(< x^{2\ell+2}, \ldots, x^{2k} >\right) + \Phi_{I\setminus J}\left(< x^{\ell+2}, \ldots, x^{k+\ell} >\right) \\ &= \Phi_I\left(< x^2, \ldots, x^{2\ell} >\right) + \Phi_{I\setminus J}\left(< x^{2\ell+2}, \ldots, x^{2k} > + < x^{\ell+2}, \ldots, x^{k+\ell} >\right) \end{aligned}$$

Since, by assumption, $\ell < k$, we have

$$< x^{\ell+2}, \ldots, x^{k+\ell} > + < x^{2\ell+2}, \ldots, x^{2k} > = < x^{\ell+2}, \ldots, x^{2k} >$$

Therefore,

$$\mathscr{C}_I^2 = \Phi_I\left(< x^2, \ldots, x^{2\ell} >\right) + \Phi_{I\setminus J}\left(< x^{\ell+2}, \ldots, x^{2k} >\right). \quad (9)$$

Lemma 11 entails

$$\dim \Phi_I\left(< x^2, \ldots, x^{2\ell} >\right) = 2\ell - 1, \quad \text{and} \quad \dim \Phi_{I\setminus J}\left(< x^{\ell+2}, \ldots, x^{2k} >\right) = 2k - \ell - 1. \quad (10)$$

To conclude the proof, we need to compute the dimension of the intersection of these spaces. For this purpose, set

$$R(x) \stackrel{\text{def}}{=} \prod_{j=a+1}^{b} (x - x_j).$$

An element of $\Phi_I\left(< x^2, \ldots, x^{2\ell} >\right) \cap \Phi_{I\setminus J}\left(< x^{\ell+2}, \ldots, x^{2k} >\right)$ is an element of $\Phi_I\left(< x^2, \ldots, x^{2\ell} >\right)$ which vanishes on the $|J| = b-a$ last positions: it is an element of $\Phi_I\left(< x^2 R(x), \ldots, x^{2\ell-|J|} R(x) >\right)$. Thus,

$$\begin{aligned} &\Phi_I\left(< x^2, \ldots, x^{2\ell} >\right) \cap \Phi_{I\setminus J}\left(< x^{\ell+2}, \ldots, x^{2k} >\right) \\ &\qquad = \Phi_I\left(< x^2 R, \ldots, x^{2\ell-|J|} R >\right) \cap \Phi_{I\setminus J}\left(< x^{\ell+2}, \ldots, x^{2k} >\right) \\ &\qquad = \Phi_{I\setminus J}\left(< x^2 R, \ldots, x^{2\ell-|J|} R >\right) \cap \Phi_{I\setminus J}\left(< x^{\ell+2}, \ldots, x^{2k} >\right) \\ &\qquad = \Phi_{I\setminus J}\left(< x^2 R, \ldots, x^{2\ell-|J|} R > \cap < x^{\ell+2}, \ldots, x^{2k} >\right). \end{aligned}$$

The last equality is also a consequence of Lemma 11: the direct image of an intersection by an injective map is the intersection of the direct images.

Since all the x_i's are nonzero, the polynomials $x^{\ell+2}$ and R are prime to each other, this yields

$$< x^2R, \ldots, x^{2\ell-|J|}R > \cap < x^{\ell+2}, \ldots, x^{2k} > = < x^{\ell+2}R, \ldots, x^{2\ell-|J|}R > .$$

Therefore,

$$\Phi_I\left(< x^2, \ldots, x^{2\ell} >\right) \cap \Phi_{I\setminus J}\left(< x^{\ell+2}, \ldots, x^{2k} >\right) = \Phi_{I\setminus J}\left(< x^{\ell+2}R(x), \ldots, x^{2\ell-|J|}R(x) >\right) \quad (11)$$

and this last space has dimension $\ell - |J| - 1$. Finally, combining (9), (10) and (11), we get

$$\dim \mathscr{C}_I^2 = (2k - \ell - 1) + (2\ell - 1) - (\ell - |J| - 1) = 2k - |J| - 1.$$

□

An attacker can exploit this proposition to mount a distinguisher that recognizes whether a given position belongs to the secret set L. At first a set I which satisfies with high probability the assumptions of Proposition 10 is randomly chosen. Take for instance $|I| = 3k$. Then $k_I \stackrel{\text{def}}{=} \dim(\mathscr{C}_I^2)$ is computed. Next, one element x is removed from I to get a new set I' and $k_{I'} = \dim(\mathscr{C}_{I'}^2)$ is computed. The only two possible cases are either $x \notin L$ then $k_{I'} = k_I$ or $x \in L$ and then $k_{I'} = k_I - 1$. By repeating this procedure, the whole set $J = I \cap L$ is easily recovered. The next step now is to find all the elements of L that are not in I. One solution is to exchange one element in $I \setminus J$ by another element in $\{1, \ldots, n\} \setminus I$ and compare the values of k_I. If it increases, it means that the new element belongs to L. At the end of this procedure the set L is totally recovered. This probabilistic algorithm is obviously of polynomial time complexity and breaks completely the homomorphic scheme suggested in [BL11].

4 The BBCRS Cryptosystem

4.1 Description of the Scheme

The cryptosystem proposed by Baldi et *al.* in [BBC+11] is a variant of McEliece's cryptosystem [McE78] which replaces the permutation matrix used to hide the secret generator matrix by one of the form $\boldsymbol{\Pi} + \boldsymbol{R}$ where $\boldsymbol{\Pi}$ is a permutation matrix and $\boldsymbol{R}$ is a rank-one matrix. From the authors' point of view, this new kind of transformation would allow to use families of codes that were shown insecure in the original McEliece's cryptosystem. In particular, it would become possible to use GRS codes in this new framework. The scheme can be summarized as follows.

Secret key.

- $\boldsymbol{G_{sec}}$ is a generator matrix of a GRS code of length n and dimension k over $\mathbb{F}_q$,
- $\boldsymbol{Q} \stackrel{\text{def}}{=} \boldsymbol{\Pi} + \boldsymbol{R}$ where $\boldsymbol{\Pi}$ is an $n \times n$ permutation matrix;
- $\boldsymbol{R}$ is a rank-one matrix over $\mathbb{F}_q$ such that $\boldsymbol{Q}$ is invertible. In other words there exist $\boldsymbol{\alpha} \stackrel{\text{def}}{=} (\alpha_1, \ldots, \alpha_n)$ and $\boldsymbol{\beta} \stackrel{\text{def}}{=} (\beta_1, \ldots, \beta_n)$ in $\mathbb{F}_q^n$ such that $\boldsymbol{R} \stackrel{\text{def}}{=} \boldsymbol{\alpha}^T\boldsymbol{\beta}$.
- $\boldsymbol{S}$ is a $k \times k$ random invertible matrix over $\mathbb{F}_q$.

Public key. $\boldsymbol{G_{pub}} \stackrel{\text{def}}{=} \boldsymbol{S}^{-1}\boldsymbol{G_{sec}}\boldsymbol{Q}^{-1}$.

Encryption. The ciphertext $\boldsymbol{c} \in \mathbb{F}_q^n$ of a plaintext $\boldsymbol{m} \in \mathbb{F}_q^k$ is obtained by drawing at random $\boldsymbol{e}$ in $\mathbb{F}_q^n$ of weight less than or equal to $\frac{n-k}{2}$ and computing $\boldsymbol{c} \stackrel{\text{def}}{=} \boldsymbol{m}\boldsymbol{G_{pub}} + \boldsymbol{e}$.

Decryption. It consists in performing the three following steps:

1. Guessing the value of $\boldsymbol{eR}$;
2. Calculating $\boldsymbol{c}' \stackrel{\text{def}}{=} \boldsymbol{cQ} - \boldsymbol{eR} = \boldsymbol{mS}^{-1}\boldsymbol{G_{sec}} + \boldsymbol{eQ} - \boldsymbol{eR} = \boldsymbol{mS}^{-1}\boldsymbol{G_{sec}} + \boldsymbol{e\Pi}$ and using the decoding algorithm of the GRS code to recover $\boldsymbol{mS}^{-1}$ from the knowledge of $\boldsymbol{c}'$;
3. Multiplying the result of the decoding by $\boldsymbol{S}$ to recover $\boldsymbol{m}$.

The first step of the decryption, that is guessing the value $\boldsymbol{eR}$, boils down to trying q elements (in the worst case) since $\boldsymbol{eR} = \boldsymbol{e\alpha}^T\boldsymbol{\beta} = \gamma\boldsymbol{\beta}$ where γ is an element of $\mathbb{F}_q$.

4.2 Key-Recovery Attack When $2k + 2 < n$

We define $\mathscr{C}_{\text{sec}}$ and $\mathscr{C}_{\text{pub}}$ to be the codes generated by the matrices $\boldsymbol{G_{sec}}$ and $\boldsymbol{G_{pub}}$ respectively. We denote by n the length of these codes and by k their dimension. We assume in this subsection that

$$2k + 2 < n \tag{12}$$

The case of rates larger than $1/2$ will be treated in Subsection 4.3.

As explained in Subsection 4.1, $\mathscr{C}_{\text{sec}}$ is a GRS code. It will be convenient to bring in the code

$$\mathscr{C} \stackrel{\text{def}}{=} \mathscr{C}_{\text{sec}}\boldsymbol{\Pi}^{-1}. \tag{13}$$

This code $\mathscr{C}$, being a permutation of a GRS code, is itself a GRS code. So there are elements $\boldsymbol{x}$ and $\boldsymbol{y}$ in $\mathbb{F}_q^n$ such that $\mathscr{C} = \mathbf{GRS}_k(\boldsymbol{x}, \boldsymbol{y})$. There is a simple relation between $\mathscr{C}_{\text{pub}}$ and $\mathscr{C}$ as explained by Lemma 13 below.

First, notice that, since $\boldsymbol{R}$ has rank 1, then so does $\boldsymbol{R\Pi}^{-1}$. Hence there exist $\boldsymbol{a}$ and $\boldsymbol{b}$ in $\mathbb{F}_q^n$ such that:

$$\boldsymbol{R\Pi}^{-1} = \boldsymbol{b}^T\boldsymbol{a}. \tag{14}$$

Lemma 13. *Let $\boldsymbol{\lambda} \stackrel{def}{=} -\frac{1}{1+\boldsymbol{a}\cdot\boldsymbol{b}}\boldsymbol{b}$ For any $\boldsymbol{c}$ in $\mathscr{C}_{pub}$ there exists $\boldsymbol{p}$ in $\mathscr{C}$ such that:*

$$\boldsymbol{c} = \boldsymbol{p} + (\boldsymbol{p}\cdot\boldsymbol{\lambda})\boldsymbol{a}. \tag{15}$$

Remark 2. Notice that the definition of $\boldsymbol{\lambda}$ makes sense if and only of $\boldsymbol{a}\cdot\boldsymbol{b} \neq -1$. This actually holds since $\boldsymbol{Q}$ is assumed to be invertible (See Lemmas 22 and 23 in Appendix B).

Proof of Lemma 13. Let $\boldsymbol{c}$ be an element of $\mathscr{C}_{\text{pub}}$. Since

$$\mathscr{C}_{\text{sec}} = \mathscr{C}_{\text{pub}}\boldsymbol{Q} = \mathscr{C}_{\text{pub}}(\boldsymbol{\Pi} + \boldsymbol{R}) = \mathscr{C}_{\text{pub}}(\boldsymbol{I} + \boldsymbol{R\Pi}^{-1})\boldsymbol{\Pi} = \mathscr{C}_{\text{pub}}\boldsymbol{P\Pi},$$

we obtain $\mathscr{C}_{\text{sec}}\boldsymbol{\Pi}^{-1} = \mathscr{C}_{\text{pub}}\boldsymbol{P}$ and therefore

$$\mathscr{C}_{\text{pub}} = (\mathscr{C}_{\text{sec}}\boldsymbol{\Pi}^{-1})\boldsymbol{P}^{-1} = \mathscr{C}\boldsymbol{P}^{-1}.$$

From this, we obtain that there exists $\boldsymbol{p}$ in $\mathscr{C}$ such that $\boldsymbol{c} = \boldsymbol{pP}^{-1}$. Thus, from Lemma 23,

$$\boldsymbol{c} = \boldsymbol{p}\left(\boldsymbol{I} - \frac{1}{1+\boldsymbol{a}\cdot\boldsymbol{b}}\boldsymbol{b}^T\boldsymbol{a}\right) = \boldsymbol{p} - \frac{\boldsymbol{b}\cdot\boldsymbol{p}}{1+\boldsymbol{a}\cdot\boldsymbol{b}}\boldsymbol{a} = \boldsymbol{p} + (\boldsymbol{\lambda}\cdot\boldsymbol{p})\boldsymbol{a}.$$

□

From now on we make the assumption that

$$\boldsymbol{\lambda} \notin \mathscr{C}^{\perp} \text{ and } \boldsymbol{a} \notin \mathscr{C}. \tag{16}$$

If this is not the case then $\mathscr{C}_{\text{pub}} = \mathscr{C} = \mathbf{GRS}_k(\boldsymbol{x}, \boldsymbol{y})$ and there is straightforward attack by applying the Sidelnikov and Shestakov algorithm [SS92]. It finds $(\boldsymbol{x}', \boldsymbol{y}')$ that expresses $\mathscr{C}_{\text{pub}}$ as $\mathbf{GRS}_k(\boldsymbol{x}', \boldsymbol{y}')$. Our attack relies on identifying a code of dimension $k-1$ that is both a subcode of $\mathscr{C}_{\text{pub}}$ and the GRS code $\mathscr{C}$. It consists more precisely of codewords $\boldsymbol{p} + (\boldsymbol{p} \cdot \boldsymbol{\lambda})\boldsymbol{a}$ with $\boldsymbol{p}$ in $\mathscr{C}$ such that $\boldsymbol{p} \cdot \boldsymbol{\lambda} = 0$. This particular code which is denoted by $\mathscr{C}_{\boldsymbol{\lambda}^{\perp}}$ is therefore:

$$\mathscr{C}_{\boldsymbol{\lambda}^{\perp}} \stackrel{\text{def}}{=} \mathscr{C} \cap < \boldsymbol{\lambda} >^{\perp} \tag{17}$$

where $< \boldsymbol{\lambda} >$ denotes the vector space spanned by $\boldsymbol{\lambda}$. It is a subspace of $\mathscr{C}_{\text{pub}}$ of codimension 1 if Assumption (16) holds. Here is an inclusion diagram for the involved codes.

$$\begin{array}{ccc} \mathscr{C}_{\text{pub}} & & \mathscr{C} \\ & \searrow^{\text{Codim1}} \quad \swarrow_{\text{Codim1}} & \\ & \mathscr{C}_{\boldsymbol{\lambda}^{\perp}} & \end{array} \tag{18}$$

Summary of the attack. Before describing it in depth, let us give the main steps of the attack.

Step 1. Compute a basis of $\mathscr{C}_{\boldsymbol{\lambda}^{\perp}}$ using distinguisher-based methods. See § 4.2.1 for further details.

Step 2. Use Wieschebrink's method [Wie10], which asserts that: $\mathscr{C}^2_{\boldsymbol{\lambda}^{\perp}} = \mathscr{C}^2$ to recover the structure of $\mathscr{C}^2$ and then that of $\mathscr{C}$. See § 4.2.2.

Step 3. Compute a pair $(\boldsymbol{a}_0, \boldsymbol{\lambda}_0)$ called a *valid pair* (Definition 17), which will have similar properties than the pair $(\boldsymbol{a}, \lambda)$ (see (14) and Lemma 13 for the definitions of $\boldsymbol{a}$ and $\boldsymbol{\lambda}$). See § 4.2.3.

Step 4. Thanks to the valid pair, one can decrypt any ciphered message. See § 4.2.4.

4.2.1 First step: computing a basis of $\mathscr{C}_{\boldsymbol{\lambda}^{\perp}}$

The inclusion relations described in the diagram (18) strongly suggest that $\mathscr{C}^2_{\text{pub}}$ should have an unusual low dimension since $\mathscr{C}^2$ has dimension $2k-1$ by Proposition 5. More exactly we have the following result.

Proposition 14. *The square code of $\mathscr{C}_{pub}$ satisfies*

(1) $\mathscr{C}^2_{pub} \subset \mathscr{C}^2 + \mathscr{C} \star \boldsymbol{a} + < \boldsymbol{a} \star \boldsymbol{a} >$;

(2) $\dim\left(\mathscr{C}^2_{pub}\right) \leqslant 3k-1$.

Proof. Assertion (1) follows immediately from Lemma 13. As for the proof of (2), let $\boldsymbol{c}$ and $\boldsymbol{c}'$ be two elements in $\mathscr{C}_{\text{pub}}$. By applying Lemma 13 to them we know that there exist two elements $\boldsymbol{p}$ and $\boldsymbol{p}'$ in $\mathscr{C}$ such that

$$\begin{aligned} \boldsymbol{c} &= \boldsymbol{p} + (\boldsymbol{\lambda} \cdot \boldsymbol{p})\boldsymbol{a} \\ \boldsymbol{c}' &= \boldsymbol{p}' + (\boldsymbol{\lambda} \cdot \boldsymbol{p}')\boldsymbol{a}. \end{aligned}$$

This implies that

$$\begin{aligned} \boldsymbol{c} \star \boldsymbol{c}' &= (\boldsymbol{p} + (\boldsymbol{\lambda} \cdot \boldsymbol{p})\boldsymbol{a}) \star (\boldsymbol{p}' + (\boldsymbol{\lambda} \cdot \boldsymbol{p}')\boldsymbol{a}) \\ &= \boldsymbol{p} \star \boldsymbol{p}' + ((\boldsymbol{\lambda} \cdot \boldsymbol{p})\boldsymbol{p}' + (\boldsymbol{\lambda} \cdot \boldsymbol{p}')\boldsymbol{p}) \star \boldsymbol{a} + (\boldsymbol{\lambda} \cdot \boldsymbol{p})(\boldsymbol{\lambda} \cdot \boldsymbol{p}')\boldsymbol{a} \star \boldsymbol{a} \end{aligned} \tag{19}$$

Consider the symmetric bilinear map

$$\Phi : \left\{ \begin{array}{rcl} \mathscr{C} \times \mathscr{C} & \to & \mathscr{C}_{\text{pub}} \\ (\boldsymbol{p}, \boldsymbol{p}') & \mapsto & (\boldsymbol{\lambda} \cdot \boldsymbol{p})\boldsymbol{p}' + (\boldsymbol{\lambda} \cdot \boldsymbol{p}')\boldsymbol{p} + (\boldsymbol{\lambda} \cdot \boldsymbol{p})(\boldsymbol{\lambda} \cdot \boldsymbol{p}')\boldsymbol{a} \end{array} \right.$$

Relation (19) can be reformulated as

$$\boldsymbol{c} \star \boldsymbol{c}' = \boldsymbol{p} \star \boldsymbol{p}' + \Phi(\boldsymbol{p}, \boldsymbol{p}') \star a. \tag{20}$$

Set $\mathscr{K} \stackrel{\text{def}}{=} < \Phi(\boldsymbol{p}, \boldsymbol{p}') \mid (\boldsymbol{p}, \boldsymbol{p}') \in \mathscr{C} \times \mathscr{C} >$. Equation (20) entails

$$\mathscr{C}_{\text{pub}}^2 \subseteq \mathscr{C}^2 + \mathscr{K} \star < a > . \tag{21}$$

Since $\mathscr{C}$ is a GRS code of dimension k, from Proposition 5, we know that $\dim \mathscr{C}^2 = 2k - 1$. Hence, to prove the result, there remains to prove that $\dim \mathscr{K} \leqslant k$. To prove that, let $(\boldsymbol{p}_1, \ldots, \boldsymbol{p}_{k-1}, \boldsymbol{p}_k)$ be a basis of $\mathscr{C}$, such that $(\boldsymbol{p}_1, \ldots, \boldsymbol{p}_{k-1})$ is a basis of $\mathscr{C}_{\boldsymbol{\lambda}^\perp}$. First, observe that, since Φ is bilinear and symmetric, we have

$$\mathscr{K} = < \Phi(\boldsymbol{p}_i, \boldsymbol{p}_j) \mid 1 \leqslant i \leqslant j \leqslant k > . \tag{22}$$

Second, notice that $\Phi(\boldsymbol{p}, \boldsymbol{p}') = 0$ as soon as both $\boldsymbol{p}$ and $\boldsymbol{p}' \in \mathscr{C}_{\boldsymbol{\lambda}^\perp}$. Hence (22) reduces to

$$\mathscr{K} = < \Phi(\boldsymbol{p}_i, \boldsymbol{p}_k) \mid 1 \leqslant i \leqslant k > . \tag{23}$$

Therefore $\dim \mathscr{K} \leqslant k$. □

Experimentally it has been observed that the upper-bound is sharp. Indeed, the dimension of $\mathscr{C}_{\text{pub}}^2$ has always been found to be equal to $3k - 1$ in all our experiments when choosing randomly the codes and $\boldsymbol{Q}$.

The second observation is that when a basis $\boldsymbol{g}_1, \ldots, \boldsymbol{g}_k$ of $\mathscr{C}_{\text{pub}}$ is chosen together with l other random elements $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_l \in \mathscr{C}_{\text{pub}}$, then we may expect that the dimension of the vector space generated by all products $\boldsymbol{z}_i \star \boldsymbol{g}_j$ with i in $\{1, \ldots, l\}$ and j in $\{1, \ldots, k\}$ is the dimension of the full space $\mathscr{C}_{\text{pub}}^2$ when $l \geqslant 3$. This is indeed the case when $l \geqslant 4$ but it is not true for $l = 3$ since we have the following result.

Proposition 15. *Let $\mathscr{B}$ be the space spanned by $\left\{\boldsymbol{z}_i \star \boldsymbol{g}_j \mid 1 \leqslant i \leqslant 3,\ 1 \leqslant j \leqslant k\right\}$, then* $\dim(\mathscr{B}) \leqslant 3k - 3$.

Proof. This follows immediately from the fact that we can express $\boldsymbol{z}_i$ in terms of the $\boldsymbol{g}_j$'s, say

$$\boldsymbol{z}_i = \sum_{1 \leqslant j \leqslant k} a_{ij} \boldsymbol{g}_j.$$

We observe now that we have the following three relations between the $\boldsymbol{z}_i \star \boldsymbol{g}_j$'s:

$$\sum_{1 \leqslant j \leqslant k} a_{2j} \boldsymbol{z}_1 \star \boldsymbol{g}_j - \sum_{1 \leqslant j \leqslant k} a_{1j} \boldsymbol{z}_2 \star \boldsymbol{g}_j = 0 \tag{24}$$

$$\sum_{1 \leqslant j \leqslant k} a_{3j} \boldsymbol{z}_1 \star \boldsymbol{g}_j - \sum_{1 \leqslant j \leqslant k} a_{1j} \boldsymbol{z}_3 \star \boldsymbol{g}_j = 0 \tag{25}$$

$$\sum_{1 \leqslant j \leqslant k} a_{2j} \boldsymbol{z}_3 \star \boldsymbol{g}_j - \sum_{1 \leqslant j \leqslant k} a_{3j} \boldsymbol{z}_2 \star \boldsymbol{g}_j = 0 \tag{26}$$

(24) can be verified as follows

$$\sum_{1\leqslant j\leqslant k} a_{2j}\boldsymbol{z}_1\star\boldsymbol{g}_j - \sum_{1\leqslant j\leqslant k} a_{1j}\boldsymbol{z}_2\star\boldsymbol{g}_j = \boldsymbol{z}_1\star\boldsymbol{z}_2 - \boldsymbol{z}_2\star\boldsymbol{z}_1 = 0.$$

The two remaining identities can be proved in a similar fashion. It remains to prove that the three obtained identities relating the $\boldsymbol{z}_i\star\boldsymbol{g}_j$'s are independent under some conditions on the $\boldsymbol{z}_i$'s. Actually, these relations are independent if and only if the $\boldsymbol{z}_i$'s generate a space of dimension larger than or equal to 2. Indeed, sort the $\boldsymbol{z}_1\star\boldsymbol{g}_j$'s as $\boldsymbol{z}_1\star\boldsymbol{g}_1,\ldots,\boldsymbol{z}_1\star\boldsymbol{g}_k,\boldsymbol{z}_2\star\boldsymbol{g}_1,\ldots,\boldsymbol{z}_2\star\boldsymbol{g}_k,\boldsymbol{z}_3\star\boldsymbol{g}_1,\ldots,\boldsymbol{z}_3\star\boldsymbol{g}_k$. Then the system defined by Equations (24) to (26) is defined by the $3\times 3k$ matrix

$$A := \begin{pmatrix} a_{21} & \cdots & a_{2k} & -a_{11} & \cdots & -a_{1k} & 0 & \cdots & 0 \\ a_{31} & \cdots & a_{3k} & 0 & \cdots & 0 & -a_{11} & \cdots & -a_{1k} \\ 0 & \cdots & 0 & -a_{31} & \cdots & -a_{3k} & a_{21} & \cdots & a_{2k} \end{pmatrix}.$$

Then, A has rank strictly less than 3 if there exists a vector $\boldsymbol{u}=(u_1,u_2,u_3)$ such that $\boldsymbol{u}A=0$ which is equivalent to the system

$$\begin{cases} u_1\boldsymbol{z}_2 + u_2\boldsymbol{z}_3 &= 0 \\ -u_1\boldsymbol{z}_1 - u_3\boldsymbol{z}_3 &= 0 \\ -u_2\boldsymbol{z}_1 + u_3\boldsymbol{z}_2 &= 0 \end{cases}$$

and such a system has a nonzero solution $\boldsymbol{u}=(u_1,u_2,u_3)$ if and only if the $\boldsymbol{z}_i$'s are pairwise collinear i.e. generate a subspace of dimension lower than or equal to 1. □

Experimentally, it turns out that almost always this upper-bound is tight and the dimension is generally $3k-3$. But if we assume now that $\boldsymbol{z}_1$, $\boldsymbol{z}_2$, $\boldsymbol{z}_3$ all belong to $\mathscr{C}_{\boldsymbol{\lambda}^\perp}$, which happens with probability $\frac{1}{q^3}$ since $\mathscr{C}_{\boldsymbol{\lambda}^\perp}$ is a subspace of $\mathscr{C}_{\text{pub}}$ of codimension 1 (at least when (16) holds), then the vectors $\boldsymbol{z}_i\star\boldsymbol{g}_j$ generate a subspace with a much smaller dimension.

Proposition 16. *If $\boldsymbol{z}_i$ is in $\mathscr{C}_{\boldsymbol{\lambda}^\perp}$ for i in $\{1,2,3\}$ then for all j in $\{1,\ldots,k\}$:*

$$\boldsymbol{z}_i\star\boldsymbol{g}_j \;\subset\; \mathscr{C}^2 \;+\; <\boldsymbol{z}_1\star\boldsymbol{a}> \;+\; <\boldsymbol{z}_2\star\boldsymbol{a}> \;+\; <\boldsymbol{z}_3\star\boldsymbol{a}> \tag{27}$$

and if $\mathscr{B}$ is the linear code spanned by $\left\{\boldsymbol{z}_i\star\boldsymbol{g}_j \mid 1\leqslant i\leqslant 3 \text{ and } 1\leqslant j\leqslant k\right\}$ then

$$\dim(\mathscr{B}) \leqslant 2k+2. \tag{28}$$

Proof. Assume that the $\boldsymbol{z}_i$'s all belong to $\mathscr{C}_{\boldsymbol{\lambda}^\perp}$. For every $\boldsymbol{g}_j$ there exists $\boldsymbol{p}_j$ in $\mathscr{C}$ such that $\boldsymbol{g}_j=\boldsymbol{p}_j+\boldsymbol{\lambda}\cdot\boldsymbol{p}_j\boldsymbol{a}$. We obtain now

$$\begin{aligned} \boldsymbol{z}_i\star\boldsymbol{g}_j &= \boldsymbol{z}_i\star(\boldsymbol{p}_j+(\boldsymbol{\lambda}\cdot\boldsymbol{p}_j)\boldsymbol{a}) \\ &= \boldsymbol{z}_i\star\boldsymbol{p}_j+(\boldsymbol{\lambda}\cdot\boldsymbol{p}_j)\boldsymbol{z}_i\star\boldsymbol{a} \\ &\in \mathscr{C}^2+<\boldsymbol{z}_1\star\boldsymbol{a}>+<\boldsymbol{z}_2\star\boldsymbol{a}>+<\boldsymbol{z}_3\star\boldsymbol{a}>. \end{aligned} \tag{29}$$

This proves the first part of the proposition, the second part follows immediately from the first part since it implies that the dimension of the vector space generated by the $\boldsymbol{z}_i\star\boldsymbol{g}_j$'s is upperbounded by the sum of the dimension of $\mathscr{C}^2$ (that is $2k-1$) and the dimension of the vector space spanned by the $\boldsymbol{z}_i\star\boldsymbol{a}$'s (which is at most 3). □

Algorithm 1 Recovering $\mathscr{C}_{\lambda^\perp}$.

Input: A basis $\{\boldsymbol{g}_1, \ldots, \boldsymbol{g}_k\}$ of $\mathscr{C}_{\text{pub}}$.
Output : A basis $\mathcal{L}$ of $\mathscr{C}_{\boldsymbol{\lambda}^\perp}$.

repeat
  for $1 \leqslant i \leqslant 3$ **do**
    Randomly choose $\boldsymbol{z}_i$ in $\mathscr{C}_{\text{pub}}$
  end for
  $\mathscr{B} \leftarrow \ < \left\{ \boldsymbol{z}_i \star \boldsymbol{g}_j \mid 1 \leqslant i \leqslant 3 \text{ and } 1 \leqslant j \leqslant k \right\} >$
until $\dim(\mathscr{B}) \leqslant 2k+2$ and $\dim\left(< \boldsymbol{z}_1, \boldsymbol{z}_2, \boldsymbol{z}_3 >\right) = 3$
$\mathcal{L} \leftarrow \{\boldsymbol{z}_1, \boldsymbol{z}_2, \boldsymbol{z}_3\}$
$s \leftarrow 4$
while $s \leqslant k-1$ **do**
  repeat
    Randomly choose $\boldsymbol{z}_s$ in $\mathscr{C}_{\text{pub}}$
    $\mathscr{T} \leftarrow \ < \left\{ \boldsymbol{z}_i \star \boldsymbol{g}_j \mid i \in \{1, 2, s\} \text{ and } 1 \leqslant j \leqslant k \right\} >$
  until $\dim(\mathscr{T}) \leqslant 2k+2$ $\dim\left(< \mathcal{L} \cup \{\boldsymbol{z}_s\} >\right) = s$
  $\mathcal{L} \leftarrow \mathcal{L} \cup \{\boldsymbol{z}_s\}$
  $s \leftarrow s+1$
end while$\mathcal{L}$;

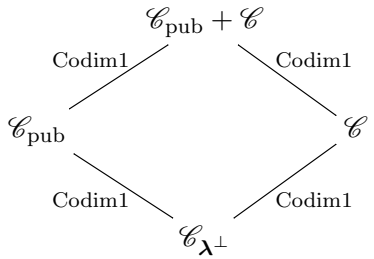
The upper-bound given in (28) on the dimension follows immediately from (27). This leads to Algorithm 1 which computes a basis of $\mathscr{C}_{\boldsymbol{\lambda}^\perp}$. It is essential that the condition in (12) holds in order to distinguish the case when the dimension is less than or equal to $2k+2$ from higher dimensions. The first phase of the attack, namely finding a suitable triple $\boldsymbol{z}_1, \boldsymbol{z}_2, \boldsymbol{z}_3$ runs in expected time $O\left(k^3 q^3\right)$ because each test in the **repeat** loop 4.2.1 has a chance of $\frac{1}{q^3}$ to succeed. Indeed, $\mathscr{C}_{\boldsymbol{\lambda}^\perp}$ is of codimension 1 in $\mathscr{C}_{\text{pub}}$ and therefore a fraction $\frac{1}{q}$ of elements of $\mathscr{C}_{\text{pub}}$ belongs to $\mathscr{C}_{\boldsymbol{\lambda}^\perp}$. Once $\boldsymbol{z}_1, \boldsymbol{z}_2, \boldsymbol{z}_3$ are found, getting any other element of $\mathscr{C}_{\boldsymbol{\lambda}^\perp}$ is easy. Indeed, take a random element $\boldsymbol{z} \in \mathscr{C}_{\text{pub}}$ and use the same test to check whether the triple $\boldsymbol{z}_1, \boldsymbol{z}_2, \boldsymbol{z}$ is in $\mathscr{C}_{\boldsymbol{\lambda}^\perp}$. Since $\boldsymbol{z}_1, \boldsymbol{z}_2 \in \mathscr{C}_{\boldsymbol{\lambda}^\perp}$ the probability of success is $\frac{1}{q}$ and hence $\boldsymbol{z}$ can be found in $O(q)$ tests. The whole algorithm runs in expected time $O\left(k^3 q^3\right) + O\left(k^4 q\right) = O\left(k^3 q^3\right)$ since $k = O(q)$ and the first phase of the attack is dominant in the complexity.

4.2.2 Second step: recovering the structure of $\mathscr{C}$

Once $\mathscr{C}_{\boldsymbol{\lambda}^\perp}$ is recovered, it still remains to recover the secret code and $\boldsymbol{a}$. The problem at hand can be formulated like this: we know a very large subcode, namely $\mathscr{C}_{\boldsymbol{\lambda}^\perp}$, of a GRS code that we want to recover. This is exactly the problem which was solved in [Wie10]. In our case this amounts to compute $\mathscr{C}^2_{\boldsymbol{\lambda}^\perp}$ which turns out to be equal to $\mathbf{GRS}_{2k-1}(\boldsymbol{x}, \boldsymbol{y} \star \boldsymbol{y})$ (see [MCMMP11, MCMMP12] for more details). It suffices to use the Sidelnikov and Shestakov algorithm [SS92] or the algorithm described in Section 5 to compute a pair $(\boldsymbol{x}, \boldsymbol{y} \star \boldsymbol{y})$ describing $\mathscr{C}^2_{\boldsymbol{\lambda}^\perp}$ as a GRS code. From this, we deduce a pair $(\boldsymbol{x}, \boldsymbol{y})$ defining the secret code $\mathscr{C}$ as a GRS code.

4.2.3 Deriving $\boldsymbol{a}$ and $\boldsymbol{\lambda}$ from $\mathscr{C}$ and $\mathscr{C}_{\boldsymbol{\lambda}^\perp}$

At this step of the attack let us summarize what has been done. We have been able to compute the codes $\mathscr{C}$ and $\mathscr{C}_{\boldsymbol{\lambda}^\perp}$ defined in (13) and (17) respectively. We recall the inclusion diagram.



In addition, we know that the code $\mathscr{C}$ and $\mathscr{C}_{\text{pub}}$ are related by the map

$$\psi_{\boldsymbol{a},\boldsymbol{\lambda}} : \left\{ \begin{array}{ccc} \mathscr{C} & \to & \mathscr{C}_{\text{pub}} \\ \boldsymbol{p} & \mapsto & \boldsymbol{p} + (\boldsymbol{p}\cdot\boldsymbol{\lambda})\boldsymbol{a} \end{array} \right. . \tag{30}$$

To finish the attack, we need to find a pair $(\boldsymbol{a}_0, \boldsymbol{\lambda}_0) \in \mathbb{F}_q^n \times \mathbb{F}_q^n$ such that the map $\psi_{\boldsymbol{a}_0,\boldsymbol{\lambda}_0}$ induces an isomorphism from $\mathscr{C}$ to $\mathscr{C}_{\text{pub}}$. This motivates the following definition.

Definition 17. A pair $(\boldsymbol{a}_0, \boldsymbol{\lambda}_0) \in \mathbb{F}_q^n \times \mathbb{F}_q^n$ is said to be a *valid pair* is and only if

(a) $\boldsymbol{a}_0 \cdot \boldsymbol{\lambda}_0 \neq -1$;

(b) $\psi_{\boldsymbol{a}_0,\boldsymbol{\lambda}_0}(\mathscr{C}) \subseteq \mathscr{C}_{\text{pub}}$.

Remark 3. From Corollary 24 (Appendix B), Condition (a) asserts that $\psi_{\boldsymbol{a}_0,\boldsymbol{\lambda}_0}$ is an isomorphism. Thus,

$$\forall \boldsymbol{p} \in \mathscr{C}_{\text{pub}},\ \exists \boldsymbol{p}' \in \mathscr{C},\ \text{such that} \boldsymbol{p} = \boldsymbol{p}' + (\boldsymbol{p}'\cdot\boldsymbol{\lambda}_0)\boldsymbol{a}_0.$$

Moreover, if (a) holds then the inclusion in (b) is an equality since both codes have the same dimension.

First, we choose $\boldsymbol{u} \in \mathscr{C} \setminus \mathscr{C}_{\boldsymbol{\lambda}^\perp}$ and $\boldsymbol{v} \in \mathscr{C}_{\text{pub}} \setminus \mathscr{C}_{\boldsymbol{\lambda}^\perp}$. Since $\mathscr{C}_{\boldsymbol{\lambda}^\perp}$ has codimension 1 in $\mathscr{C}$, we have

$$\mathscr{C} = \mathscr{C}_{\boldsymbol{\lambda}^\perp} \oplus < \boldsymbol{u} > \quad \text{and} \quad \mathscr{C}_{\text{pub}} = \mathscr{C}_{\boldsymbol{\lambda}^\perp} \oplus \boldsymbol{v}. \tag{31}$$

A valid pair $(\boldsymbol{a}_0, \boldsymbol{\lambda}_0)$ can be found easily using the two following elementary lemmas.

Lemma 18. *For all* $\boldsymbol{\lambda}_0 \in \mathscr{C}_{\boldsymbol{\lambda}^\perp}^\perp \setminus (\mathscr{C}^\perp \cup \mathscr{C}_{pub}^\perp)$*, we have*

$$\boldsymbol{\lambda}_0 \cdot \boldsymbol{u} \neq 0 \quad \textit{and} \quad \boldsymbol{\lambda}_0 \cdot \boldsymbol{v} \neq 0.$$

Proof. Assume that $\boldsymbol{\lambda}_0 \cdot \boldsymbol{u} = 0$. Then, $\boldsymbol{\lambda}_0 \in \mathscr{C}_{\boldsymbol{\lambda}^\perp}^\perp \cap < \boldsymbol{u} >^\perp = (\mathscr{C}_{\boldsymbol{\lambda}^\perp} + < \boldsymbol{u} >)^\perp$. Hence, from (31), we would have $\boldsymbol{\lambda}_0 \in \mathscr{C}^\perp$ which yields a contradiction. The other non-equality is proved by the very same manner. □

Lemma 19. *For all* $\boldsymbol{\lambda}_0 \in \mathscr{C}_{\boldsymbol{\lambda}^\perp}^\perp$ *and for all* $\boldsymbol{x} \in \mathbb{F}_q^n$*, we have*

$$\psi_{\boldsymbol{\lambda}_0,\boldsymbol{x}}(\mathscr{C}) \subset \mathscr{C}_{pub} \iff \psi_{\boldsymbol{\lambda}_0,\boldsymbol{x}}(\boldsymbol{u}) \in \mathscr{C}_{pub}.$$

Proof. Since $\boldsymbol{u} \in \mathscr{C}$, the implication ($\Longleftarrow$) is obvious. Conversely, assume that $\psi_{\boldsymbol{\lambda}_0,\boldsymbol{x}}(\boldsymbol{u}) \in \mathscr{C}_{\text{pub}}$. Then, from (31), to show the result there remains to show that $\psi_{\boldsymbol{\lambda}_0,\boldsymbol{x}}(\mathscr{C}_{\boldsymbol{\lambda}^\perp}) \subset \mathscr{C}_{\text{pub}}$. But, since $\boldsymbol{\lambda}_0 \in \mathscr{C}_{\boldsymbol{\lambda}^\perp}^\perp$, then for all $\boldsymbol{p} \in \mathscr{C}_{\boldsymbol{\lambda}^\perp}$, we have

$$\psi_{\boldsymbol{\lambda}_0,\boldsymbol{x}}(\boldsymbol{p}) = \boldsymbol{p} + (\boldsymbol{\lambda}_0 \cdot \boldsymbol{p})\boldsymbol{x} = \boldsymbol{p}.$$

Thus, $\psi_{\boldsymbol{\lambda}_0,\boldsymbol{x}}(\mathscr{C}_{\boldsymbol{\lambda}^\perp}) = \mathscr{C}_{\boldsymbol{\lambda}^\perp} \subset \mathscr{C}_{\text{pub}}$. □

Procedure to recover a valid pair. Before starting, recall that we fixed vectors $\boldsymbol{u} \in \mathscr{C} \setminus \mathscr{C}_{\boldsymbol{\lambda}^\perp}$ and $\boldsymbol{v} \in \mathscr{C}_{\text{pub}} \setminus \mathscr{C}_{\boldsymbol{\lambda}^\perp}$ so that (31) holds.

Step 1. Choose $\boldsymbol{\lambda}_0 \in \mathscr{C}_{\boldsymbol{\lambda}^\perp}^\perp \setminus (\mathscr{C}^\perp \cup \mathscr{C}_{\text{pub}}^\perp)$ at random. Notice that the set $\mathscr{C}_{\boldsymbol{\lambda}^\perp}^\perp \setminus (\mathscr{C}^\perp \cup \mathscr{C}_{\text{pub}}^\perp)$ is nonempty since both $\mathscr{C}^\perp$ and $\mathscr{C}_{\text{pub}}^\perp$ have codimension 1 in $\mathscr{C}_{\boldsymbol{\lambda}^\perp}^\perp$ and even over a finite field, no vector space of dimension $\geqslant 1$ is a union of two vector subspaces of codimension 1.

Step 2. Set

$$\boldsymbol{a}_0 := \frac{1}{\boldsymbol{\lambda}_0 \cdot \boldsymbol{u}} (\boldsymbol{v} - \boldsymbol{u}).$$

It is well–defined thanks to Lemma 18.

We claim that the pair $(\boldsymbol{a}_0, \boldsymbol{\lambda}_0)$ is valid. Indeed, we have

$$\boldsymbol{a}_0 \cdot \boldsymbol{\lambda}_0 = \frac{\boldsymbol{\lambda}_0 \cdot \boldsymbol{v}}{\boldsymbol{\lambda}_0 \cdot \boldsymbol{u}} - 1.$$

Moreover, $\boldsymbol{\lambda}_0 \cdot \boldsymbol{v} \neq 0$ thanks to Lemma 18, and hence $\boldsymbol{a}_0 \cdot \boldsymbol{\lambda}_0 \neq -1$. Thus, the pair satisfies Condition (a) of Definition 17.

To show that Condition (b) is satisfied too, Lemma 19 asserts that we only need to prove that $\psi_{\boldsymbol{a}_0, \boldsymbol{\lambda}_0}(\boldsymbol{u}) \in \mathscr{C}_{\text{pub}}$ which is true since an elementary computation yields

$$\psi_{\boldsymbol{a}_0, \boldsymbol{\lambda}_0}(\boldsymbol{u}) = \boldsymbol{v}$$

which is in $\mathscr{C}_{\text{pub}}$ by construction.

4.2.4 Final step: decryption of any ciphertext

We have found a valid pair (Definition 17) $(\boldsymbol{a}_0, \boldsymbol{\lambda}_0)$. We want to decode the vector $\boldsymbol{z} \stackrel{\text{def}}{=} \boldsymbol{c} + \boldsymbol{e}$ where $\boldsymbol{e}$ is an error of a certain Hamming weight which can be corrected by the decoding algorithm chosen for $\mathscr{C}$ and $\boldsymbol{c}$ is an element of the public code. From Remark 3 page 15, we know that there exists $\boldsymbol{p}$ in $\mathscr{C}$ such that

$$\boldsymbol{c} = \boldsymbol{p} + (\boldsymbol{\lambda}_0 \cdot \boldsymbol{p})\boldsymbol{a}_0. \tag{32}$$

We compute $\boldsymbol{z}(\alpha) \stackrel{\text{def}}{=} \boldsymbol{z} + \alpha \boldsymbol{a}_0$ for all elements α in $\mathbb{F}_q$. One of these elements α is equal to $-\boldsymbol{\lambda}_0 \cdot \boldsymbol{p}$ and we obtain $\boldsymbol{z}(\alpha) = \boldsymbol{p} + \boldsymbol{e}$ in this case. Decoding $\boldsymbol{z}(\alpha)$ in $\mathscr{C}$ will reveal $\boldsymbol{p}$ and this gives $\boldsymbol{c}$ by using Equation (32).

4.3 Using duality when rates are larger than $\frac{1}{2}$

The codes suggested in [BBC+11, §5.1.1,§5.1.2] are all of rate significantly larger than $\frac{1}{2}$, for instance Example 1 p.15 suggests a GRS code of length 255, dimension 195 over $\mathbb{F}_{256}$, whereas Example 2. p.15 suggests a GRS code of length 511, dimension 395 over $\mathbb{F}_{512}$. The attack suggested in the previous subsection only applies to rates smaller than $\frac{1}{2}$. There is a simple way to adapt the previous attack for this case by considering the dual $\mathscr{C}_{\text{pub}}^\perp$ of the public code. Note that by Proposition 6, there exists $\boldsymbol{y}'$ in $\mathbb{F}_q^n$ for which we have $\mathscr{C}^\perp = \mathbf{GRS}_{n-k}(\boldsymbol{x}, \boldsymbol{y}')$. Moreover, $\mathscr{C}_{\text{pub}}^\perp$ displays a similar structure as $\mathscr{C}_{\text{pub}}$.

Lemma 20. *For any $\boldsymbol{c}$ from $\mathscr{C}_{pub}^\perp$ there exists an element $\boldsymbol{p}$ in $\mathscr{C}^\perp$ such that:*

$$\boldsymbol{c} = \boldsymbol{p} + (\boldsymbol{p} \cdot \boldsymbol{a})\boldsymbol{b}. \tag{33}$$

Proof. The key to Lemma 20 is the fact that the dual of $\mathscr{C}_{\rm pub}$ is equal to $\mathscr{C}^{\perp}\boldsymbol{P}^T$. Indeed $\mathscr{C}_{\rm pub} = \mathscr{C}\boldsymbol{P}^{-1}$ and therefore for any element $\boldsymbol{c}$ of $\mathscr{C}_{\rm pub}$ there exists an element $\boldsymbol{p}$ of $\mathscr{C}$ such that $\boldsymbol{c} = \boldsymbol{p}\boldsymbol{P}^{-1}$. Observe now that every element $\boldsymbol{c}'$ in $\mathscr{C}_{\rm pub}^{\perp}$ satisfies

$$0 = \boldsymbol{c}\cdot\boldsymbol{c}' = \boldsymbol{p}\boldsymbol{P}^{-1}\cdot\boldsymbol{c}' = \boldsymbol{p}\cdot\boldsymbol{c}'\left(\boldsymbol{P}^{-1}\right)^T.$$

Therefore $\mathscr{C}_{\rm pub}^{\perp} = \mathscr{C}^{\perp}\boldsymbol{P}^T$. This discussion implies that there exists an element $\boldsymbol{p}'$ in $\mathscr{C}^{\perp}$ such that:

$$\boldsymbol{c}' = \boldsymbol{p}'\boldsymbol{P}^T = \boldsymbol{p}'\left(\boldsymbol{I} + \boldsymbol{b}^T\boldsymbol{a}\right)^T = \boldsymbol{p}' + \boldsymbol{p}'\boldsymbol{a}^T\boldsymbol{b} = \boldsymbol{p}' + (\boldsymbol{p}'\cdot\boldsymbol{a})\boldsymbol{b}.$$

□

It implies that the whole approach of the previous subsection can be carried out over $\mathscr{C}_{\rm pub}^{\perp}$. It allows to recover the secret code $\mathscr{C}^{\perp}$ and therefore also $\mathscr{C}$. This attack needs that $2(n-k)+2 < n$, that is $2k > n+2$. In summary, there is an attack as soon as k is outside a narrow interval around $n/2$ which is $[\frac{n-2}{2}, \frac{n+2}{2}]$. We have implemented this attack on magma for $n = 127$, $q = 2^7$, $k = 30$ and the average running time over 50 attacks was about 9 hours.

5 McEliece Variants Based on GRS

Let $\mathscr{C}$ be a GRS code $\mathscr{C} \stackrel{\rm def}{=} \mathbf{GRS}_k(\boldsymbol{a}, \boldsymbol{b})$. Assume that it has dimension $k \leqslant n/2$ (if not, then one can work with the dual code).

First assume that the two first positions, *i.e.* the two first entries if $\boldsymbol{a}$ are 0 and 1. Such an assumption makes sense since every GRS code is permutation equivalent to a code satisfying this condition. This is a consequence of the 3–transitivity of the action of $\mathbf{PGL}(2, \mathbb{F}_q)$ on the points of the projective line.

Notation 1. *For all i, j such that $i > 0$, $j > 0$ and $i+j \leqslant k-1$, we denote by $\mathscr{C}(i,j)$ the subcode of $\mathscr{C}$ given by the evaluation of polynomials vanishing at 0 (i.e. the first position by assumption) with multiplicity at least i and at 1 (i.e. the second position) with multiplicity at least j, i.e. multiples of $x^i(x-1)^j$. For convenience sake, we set $\mathscr{C}(0,0) \stackrel{def}{=} \mathscr{C}$.*

The main step of our attack is to compute some codes among $\mathscr{C}(i,j)$. Notice that these codes are also GRS codes.

5.1 Computing some subcodes

Clearly, the computation of a generator matrix of $\mathscr{C}(0,1), \mathscr{C}(1,0)$ and $\mathscr{C}(1,1)$ is easy since it reduces to Gauss–elimination.

The main tool of our attack is the following result.

Theorem 21. *Assume that $k \leqslant n/2$. For all $1 \leqslant i \leqslant k-2$ and all j such that $i+j \leqslant k-2$, we have*

$$\mathscr{C}(i+1,j) \star \mathscr{C}(i-1,j) = \mathscr{C}(i,j)^2 \quad \textit{and} \quad \mathscr{C}(i,j+1) \star \mathscr{C}(i,j-1) = \mathscr{C}(i,j)^2.$$

Proof. We prove the first identity, the second is obtained easily by symmetry. Set,

$$\begin{aligned} V_{i,j} &\stackrel{\rm def}{=} x^i(x-1)^j\mathbb{F}_q[x]_{<k-i-j} \\ V_{i-1,j} &\stackrel{\rm def}{=} x^{i-1}(x-1)^j\mathbb{F}_q[x]_{<k-1-i-j} \\ V_{i+1,j} &\stackrel{\rm def}{=} x^{i+1}(x-1)^j\mathbb{F}_q[x]_{<k+1-i-j}. \end{aligned}$$

These spaces have respective dimensions $k-i-j$, $k-i-j-1$ and $k-i-j+1$ and are related to our GRS codes by

$$\begin{aligned}\mathscr{C}(i,j) &\stackrel{\text{def}}{=} <(P(\boldsymbol{a})\star\boldsymbol{b}) \mid P\in V_{i,j}>\\ \mathscr{C}(i+1,j) &\stackrel{\text{def}}{=} <(P(\boldsymbol{a})\star\boldsymbol{b}) \mid P\in V_{i+1,j}>\\ \mathscr{C}(i-1,j) &\stackrel{\text{def}}{=} <(P(\boldsymbol{a})\star\boldsymbol{b}) \mid P\in V_{i-1,j}>.\end{aligned}$$

Clearly, we have

$$V_{i,j}^2 = x^{2i}(x-1)^{2j}\mathbb{F}_q[x]_{<2k-2i-2j-1}$$

and it is also simple to check that

$$V_{i-1,j}\star V_{i+1,j} = x^{2i}(x-1)^{2j}\mathbb{F}_q[x]_{<2k-2i-2j-1}.$$

This yields the result. □

Thus, from the previous result, as long as $\mathscr{C}(i,j)^2 \neq \mathbb{F}_q^n$, which holds for $k \leqslant n/2$, given generator matrices of $\mathscr{C}(i,j)$ and $\mathscr{C}(i-1,j)$, one can recover a basis of $\mathscr{C}(i+1,j)$ by solving a simple linear system. Indeed, deciding whether an element $c\in\mathscr{C}(i,j)$ is actually in $\mathscr{C}(i+1,j)$ reduces to solve

$$c\star\mathscr{C}(i-1,j)\subseteq\mathscr{C}(i,j)^2.$$

Consequently, because of our knowledge of $\mathscr{C}=\mathscr{C}(0,0),\mathscr{C}(0,1),\mathscr{C}(1,0)$ and $\mathscr{C}(1,1)$, we are able to compute recursively all the $\mathscr{C}(i,j)$'s.

5.2 Description of the attack

The attack summarises as follows. We assume that the dimension of the GRS code is less than $n/2$, if not one can apply the attack on its dual.

Step 1. Compute a basis of $\mathscr{C}(k-1,0)$, *i.e.* compute a nonzero vector c of this 1–dimensional space. The corresponding vector comes from the evaluation of a polynomial of the form λx^{k-1} for some $\lambda\in\mathbb{F}_q^\times$. More precisely, we get the vector $\lambda(\boldsymbol{a}^k\star\boldsymbol{b})$. Then, compute a basis of $\mathscr{C}(k-2,1)$. The corresponding vector c' is of the form $\mu\boldsymbol{a}^{k-2}\star(\boldsymbol{a}-\mathbf{1})$ for $\mu\in\mathbb{F}_q^\times$ and where $\mathbf{1}:=(1,\ldots,1)$.

Step 2. The vectors c and c' have no zero position but the two first ones. Thus, after puncturing at the two first positions the quotient c'/c makes sense and corresponds to the evaluation of the fraction $\nu(x-1)/x$ for some $\nu\in\mathbb{F}_q^\times$ (i.e. is $(\boldsymbol{a}-\mathbf{1})/\boldsymbol{a}$, which makes sense after a suitable puncturing).

It is worth noting that compared to the vectors c and c', the vector c'/c corresponds to the exact evaluation of $\nu(x-1)/x$ at some elements of $\mathbb{F}_q\setminus\{0,1\}$ since the entries of $\boldsymbol{b}$, are cancelled by the quotient.

Step 3. Up to now, we only made two arbitrary choices by fixing the position of 0 and 1. Because of the 3–transitivity of $\mathbf{PGL}(2,\mathbb{F}_q)$, one can make a third arbitrary choice. Thus, without loss of generality, one can assume that $\nu=1$. Now, notice that the map $x\mapsto(x-1)/x$ is a bijection from $\mathbb{F}_q\setminus\{0,1\}$ to itself with reciprocal map $y\mapsto 1/(1-y)$.

Thus, by applying the map $y\mapsto 1/(1-y)$ to the entries of the vector c'/c we get the corresponding positions, *i.e.* the vector $\boldsymbol{a}$.

Step 4. Now, comparing the vector c with the vector $\boldsymbol{a}^k$, we get $\boldsymbol{b}$ up to multiplication by an element $\alpha \in \mathbb{F}_q^\times$, which does not matter since $\mathbf{GRS}_k(\boldsymbol{a}, \boldsymbol{b}) = \mathbf{GRS}_k(\boldsymbol{a}, \alpha\boldsymbol{b})$ for $\alpha \in \mathbb{F}_q^\times$.

Conclusion

XXX ajouter un mot sur l'attaque sur les GRS et sur Wieschebrink XXXX The homomorphic scheme suggested in [BL11] actually leads in a natural way to choose codes for which the square product is of unusually small dimension (see Appendix A for more details). This sheds some light on why considerations of this kind might lead to an attack. It is worthwhile mentioning that replacing Reed-Solomon codes by Reed-Muller ones for instance in this scheme does not seem to prevent this kind of attack.

Both attacks we presented here against [BL11, BBC$^+$11] may be viewed as trying to identify, through square code dimension considerations, certain subcodes or punctured codes of the public codes of the schemes. In the case of Bogdanov-Lee's scheme, this was for identifying the punctured codes with a certain number of elements of L in their support. In the Baldi et *al.* case, this was for identifying codewords in a subcode of codimension 1. Reed-Solomon codes are particularly prone to this kind of attack because of the very low dimension of their square code.

The approach we developped here seems to have other applications to cryptanalysis. For instance, it is not too difficult to use it for finding another way of breaking a McEliece type cryptosystem based on generalized Reed-Solomon (the Sidelnikov-Shestakov attack [SS92]) which would start by trying to identify the subcode $\mathbf{GRS}_{k-1}(\boldsymbol{x}, \boldsymbol{y})$ of the generalized Reed-Solomon code $\mathbf{GRS}_k(\boldsymbol{x}, \boldsymbol{y})$. It might also be applied to other codes such as for instance Reed-Muller codes [Sid94]. The square code of these codes have also an abnormal dimension. Finally, the most challenging task would be to attack the McEliece cryptosystem with similar tools (at least for a range of parameters) since duals of Goppa codes also have, in a limited way, square codes with low dimensions.[2]

References

[BBC$^+$11] M. Baldi, M. Bianchi, F. Chiaraluce, J. Rosenthal, and D. Schipani. Enhanced public key security for the McEliece cryptosystem. Submitted, 2011. ArXiv:1108.2462v3.

[BL05] T. P. Berger and P. Loidreau. How to mask the structure of codes for a cryptographic use. *Designs Codes and Cryptography*, 35(1):63–79, 2005.

[BL11] A. Bogdanov and C.H. Lee. Homomorphic encryption from codes. See http://arxiv.org/abs/1111.4301. This paper was accepted for publication in the proceedings of the 44th ACM Symposium on Theory of Computing (STOC). The authors withdrew their paper after they learned that their scheme was threatened, 2011.

[FGO$^+$11] J.-C. Faugère, V. Gauthier, A. Otmani, L. Perret, and J.-P. Tillich. A distinguisher for high rate McEliece cryptosystems. In *Proceedings of the Information Theory Workshop 2011, ITW 2011*, pages 282–286, Paraty, Brasil, 2011.

[McE78] R. J. McEliece. *A Public-Key System Based on Algebraic Coding Theory*, pages 114–116. Jet Propulsion Lab, 1978. DSN Progress Report 44.

[2] See [MCP12] which contains much more examples of codes with this kind of behavior

[MCMMP11] I. Márquez-Corbella, E. Martínez-Moro, and R. Pellikaan. The non-gap sequence of a subcode of a generalized Reed–Solomon code. In M. Finiasz N. Sendrier, P. Charpin and A. Otmani, editors, *Proceedings of the 7-th International Workshop on Coding and Cryptography WCC 2011*, pages 183–193, April 2011.

[MCMMP12] Irene Márquez-Corbella, Edgar Martínez-Moro, and Ruud Pellikaan. The non-gap sequence of a subcode of a generalized Reed–Solomon code. *Designs, Codes and Cryptography*, pages 1–17, 2012.

[MCP12] I. Márquez-Corbella and R. Pellikaan. Error-correcting pairs for a public-key cryptosystem. preprint, 2012.

[MS86] F. J. MacWilliams and N. J. A. Sloane. *The Theory of Error-Correcting Codes*. North–Holland, Amsterdam, fifth edition, 1986.

[Nie86] H. Niederreiter. Knapsack-type cryptosystems and algebraic coding theory. *Problems Control Inform. Theory*, 15(2):159–166, 1986.

[Sid94] V.M. Sidelnikov. A public-key cryptosytem based on Reed-Muller codes. *Discrete Mathematics and Applications*, 4(3):191–207, 1994.

[SS92] V.M. Sidelnikov and S.O. Shestakov. On the insecurity of cryptosystems based on generalized Reed-Solomon codes. *Discrete Mathematics and Applications*, 1(4):439–444, 1992.

[Wie06] C. Wieschebrink. Two NP-complete problems in coding theory with an application in code based cryptography. In *Information Theory, 2006 IEEE International Symposium on*, pages 1733 –1737, july 2006.

[Wie10] C. Wieschebrink. Cryptanalysis of the Niederreiter Public Key Scheme Based on GRS Subcodes. In N. Sendrier, editor, *Post-Quantum Cryptography, Third International Workshop, PQCrypto 2010*, volume 6061 of *Lecture Notes in Computer Science*, pages 61–72, Darmstadt, Germany, May 2010. Springer.

A On the Inherent Existence of Low-Dimension Square Codes for the Bogdanov and Lee Homomorphic Scheme

The purpose of this section is to explain why the homomorphic scheme of [BL11] leads in a natural way to define codes whose square code has an abnormal dimension. The point of [BL11] is to define a code which is homomorphic for addition over $\mathbb{F}_q$ (all linear codes do the job here) but also protohomorphic for the multiplication over $\mathbb{F}_q$ [BL11, Claim 3.5]. This property holds for their scheme, because there is a solution $\boldsymbol{y}$ of (2) which satisfies for two ciphertexts $\boldsymbol{c}$ and $\boldsymbol{c}'$ in $\mathbb{F}_q^n$ corresponding respectively to the plaintexts m and m' in $\mathbb{F}_q$:

$$\boldsymbol{y} \cdot (\boldsymbol{c} \star \boldsymbol{c}') = mm' \tag{34}$$

Recall that $\boldsymbol{c}$ and $\boldsymbol{c}'$ are given by

$$\begin{aligned} \boldsymbol{c} &= \boldsymbol{x}\boldsymbol{P} + m\boldsymbol{1} + \boldsymbol{e} && (35)\\ \boldsymbol{c}' &= \boldsymbol{x}'\boldsymbol{P} + m'\boldsymbol{1} + \boldsymbol{e}' && (36) \end{aligned}$$

where $\boldsymbol{e}$ and $\boldsymbol{e}'$ are error vectors whose support does not intersect L. We also know that $\boldsymbol{y}$ satisfies:

(i) $\boldsymbol{G}\boldsymbol{y}^T = 0$;

(ii) $\sum_{i=1}^n y_i = 1$;

(iii) $y_i = 0$ if $i \notin L$ with $\boldsymbol{P}$ and $\boldsymbol{G}$ related by a multiplication of an invertible matrix $\boldsymbol{S}$, i.e. $\boldsymbol{P} = \boldsymbol{S}\boldsymbol{G}$.

We deduce from this

$$\begin{aligned} \boldsymbol{y}(\boldsymbol{c} \star \boldsymbol{c}')^T &= \boldsymbol{y}\left((\boldsymbol{x}\boldsymbol{P} + m\boldsymbol{1} + \boldsymbol{e}) \star (\boldsymbol{x}'\boldsymbol{P} + m'\boldsymbol{1} + \boldsymbol{e}')\right)^T \\ &= \boldsymbol{y}\left(\boldsymbol{P}^T\boldsymbol{x}^T \star \boldsymbol{P}^T\boldsymbol{x}'^T + \boldsymbol{P}^T\boldsymbol{x}^T \star m'\boldsymbol{1}^T + \boldsymbol{P}^T\boldsymbol{x}'^T \star m\boldsymbol{1}^T + m\boldsymbol{1}^T \star m'\boldsymbol{1}^T\right) \\ &\quad + \boldsymbol{y}\left(\boldsymbol{e}^T \star (\boldsymbol{P}^T\boldsymbol{x}'^T + m'\boldsymbol{1}^T + \boldsymbol{e}'^T)\right) + \boldsymbol{y}\left((\boldsymbol{P}^T\boldsymbol{x}^T + m\boldsymbol{1}^T) \star \boldsymbol{e}'^T\right) \end{aligned}$$

The terms $\boldsymbol{y}\left(\boldsymbol{e}^T \star (\boldsymbol{P}^T\boldsymbol{x}'^T + m'\boldsymbol{1}^T + \boldsymbol{e}'^T)\right)$ and $\boldsymbol{y}\left((\boldsymbol{P}^T\boldsymbol{x}^T + m\boldsymbol{1}^T) \star \boldsymbol{e}'^T\right)$ are equal to zero because the support of $\boldsymbol{y}$ is contained in L and $\boldsymbol{e}^T \star (\boldsymbol{P}^T\boldsymbol{x}'^T + m'\boldsymbol{1}^T + \boldsymbol{e}'^T)$, $(\boldsymbol{P}^T\boldsymbol{x}^T + m\boldsymbol{1}^T) \star \boldsymbol{e}'^T$ have their support outside L. The terms $\boldsymbol{y}(\boldsymbol{P}^T\boldsymbol{x}^T \star m'\boldsymbol{1}^T) = m'\boldsymbol{y}\boldsymbol{G}^T\boldsymbol{S}^T\boldsymbol{x}^T$ and $\boldsymbol{y}(\boldsymbol{P}^T\boldsymbol{x}'^T \star m\boldsymbol{1}^T) = m\boldsymbol{y}\boldsymbol{G}^T\boldsymbol{S}^T\boldsymbol{x}'^T$ are equal to 0 from Condition (i) on $\boldsymbol{y}$ given above. Therefore in order to ensure (34) we need that

$$\boldsymbol{y}\left(\boldsymbol{P}^T\boldsymbol{x}^T \star \boldsymbol{P}^T\boldsymbol{x}'^T\right) = 0. \tag{37}$$

has a non zero solution whose support is contained in L. Let $\mathscr{C}$ be the code with generating matrix $\boldsymbol{P}$, that is the set of elements of the form $\boldsymbol{P}\boldsymbol{x}$. Notice that the set of solutions of (37) is precisely the dual of $\mathscr{C}^2$. This implies that $\mathscr{C}^2$ should not be the whole space $\mathbb{F}_q^n$. This is quite unusual as explained in Section 1 when the dimension k of $\mathscr{C}$ satisfies $k \gg n^{1/2}$. Furthermore, since we are interested in solutions of (37) whose support is contained in L we actually need that the dual of $\mathscr{C}_L^2$ is non empty which is even more abnormal since $\mathscr{C}_L$ is a code of length 3ℓ and dimension ℓ. In other words, the Bogdanov and Lee homomorphic scheme leads in a natural way to choose codes $\mathscr{C}$ which have a non-random behavior with respect to the dimension of the square product.

B Proof of Lemma 13

Recall that $\boldsymbol{R}$ has rank 1, then so does $\boldsymbol{R}\boldsymbol{\Pi}^{-1}$ and there exist $\boldsymbol{a}$ and $\boldsymbol{b}$ in $\mathbb{F}_q^n$ such that $\boldsymbol{R}\boldsymbol{\Pi}^{-1} = \boldsymbol{b}^T\boldsymbol{a}$. Set

$$\boldsymbol{P} \stackrel{\text{def}}{=} \boldsymbol{I} + \boldsymbol{R}\boldsymbol{\Pi}^{-1} = \boldsymbol{I} + \boldsymbol{b}^T\boldsymbol{a}.$$

We need the following lemmas

Lemma 22. *The matrix $\boldsymbol{Q}$ is invertible if and only if $\boldsymbol{P}$ is.*

Proof. We have $\boldsymbol{Q} = \boldsymbol{\Pi} + \boldsymbol{R} = (\boldsymbol{I} + \boldsymbol{R}\boldsymbol{\Pi}^{-1})\boldsymbol{\Pi} = \boldsymbol{P}\boldsymbol{\Pi}$, which yields the proof. □

Lemma 23. *The matrix $\boldsymbol{P}$ is invertible if and only if $\boldsymbol{a}\cdot\boldsymbol{b} \neq -1$. In addition, if it is invertible, then*

$$\boldsymbol{P}^{-1} = \boldsymbol{I} - \frac{1}{1+\boldsymbol{a}\cdot\boldsymbol{b}}\boldsymbol{b}^T\boldsymbol{a}.$$

Proof. First, assume that $\boldsymbol{a}\cdot\boldsymbol{b} \neq -1$. Then,

$$\begin{aligned}
\boldsymbol{P}\left(\boldsymbol{I} - \frac{1}{1+\boldsymbol{a}\cdot\boldsymbol{b}}\boldsymbol{b}^T\boldsymbol{a}\right) &= \left(\boldsymbol{I} + \boldsymbol{b}^T\boldsymbol{a}\right)\left(\boldsymbol{I} - \frac{1}{1+\boldsymbol{a}\cdot\boldsymbol{b}}\boldsymbol{b}^T\boldsymbol{a}\right) \\
&= \boldsymbol{I} + \left(1 - \frac{1}{1+\boldsymbol{a}\cdot\boldsymbol{b}}\right)\boldsymbol{b}^T\boldsymbol{a} - \frac{1}{1+\boldsymbol{a}\cdot\boldsymbol{b}}\boldsymbol{b}^T\boldsymbol{a}\boldsymbol{b}^T\boldsymbol{a} \\
&= \boldsymbol{I} + \frac{\boldsymbol{a}\cdot\boldsymbol{b}}{1+\boldsymbol{a}\cdot\boldsymbol{b}}\boldsymbol{b}^T\boldsymbol{a} - \frac{\boldsymbol{a}\cdot\boldsymbol{b}}{1+\boldsymbol{a}\cdot\boldsymbol{b}}\boldsymbol{b}^T\boldsymbol{a} \\
&= \boldsymbol{I}.
\end{aligned}$$

To conclude the "only if" part of the proof, we prove that $\det(\boldsymbol{P}) = \boldsymbol{a}\cdot\boldsymbol{b} + 1$. If either $\boldsymbol{a}$ or $\boldsymbol{b}$ is zero, then it is obvious. Thus assume the vectors are nonzero. Up to a re-ordering of the entries of the vectors (i.e. up to conjugation by a permutation matrix) one can assume that the first entry a_1 of $\boldsymbol{a}$ is nonzero. In addition, up to rescaling $\boldsymbol{a}$ by $a_1^{-1}\boldsymbol{a}$ and $\boldsymbol{b}$ by $a_1\boldsymbol{b}$, which has no influence on $\boldsymbol{P}$, one can assume that $a_1 = 1$. Hence the matrix $\boldsymbol{P}$ is

$$\boldsymbol{P} = \begin{pmatrix} b_1+1 & b_2 & \cdots & b_n \\ a_2b_1 & a_2b_2+1 & \cdots & a_2b_n \\ \vdots & & \ddots & \vdots \\ a_nb_1 & a_nb_2 & \cdots & a_nb_n+1 \end{pmatrix}$$

Consider the determinant of this matrix and apply first the operations $Row_i \leftarrow Row_i - a_iRow_1$ for $i = 2,\ldots,n$ and then the operations $Row_1 \leftarrow Row_1 - b_iRow_i$ for $i = 2,\ldots,n$. This yields

$$\text{first, } \det(\boldsymbol{P}) = \begin{vmatrix} b_1+1 & b_2 & \cdots & b_n \\ -a_2 & 1 & & (0) \\ \vdots & & \ddots & \\ -a_n & (0) & & 1 \end{vmatrix}, \text{ then, } \det(\boldsymbol{P}) = \begin{vmatrix} \boldsymbol{a}\cdot\boldsymbol{b}+1 & 0 & \cdots & 0 \\ -a_2 & 1 & & (0) \\ \vdots & & \ddots & \\ -a_n & (0) & & 1 \end{vmatrix} = \boldsymbol{a}\cdot\boldsymbol{b}+1.$$

□

Corollary 24. *Given $\boldsymbol{u},\boldsymbol{v} \in \mathbb{F}_q^n$ the map $\boldsymbol{p} \mapsto \boldsymbol{p} + (\boldsymbol{u}\cdot\boldsymbol{p})\boldsymbol{v}$ is an automorphism of $\mathbb{F}_q^n$ if and only if $\boldsymbol{u}\cdot\boldsymbol{v} \neq -1$.*