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

Alain Couvreur[1], Philippe Gaborit[2], Valérie Gauthier[3], Ayoub Otmani[4], and Jean-Pierre Tillich[5]

[1] GRACE Project, INRIA Saclay & LIX, CNRS UMR 7161 - École Polytechnique, 91120 Palaiseau Cedex, France.
alain.couvreur@lix.polytechnique.fr
[2] XLIM, CNRS UMR 7252 - Université de Limoges, 123 avenue Albert Thomas, 87060 Limoges Cedex, France.
philippe.gaborit@unilim.fr
[3] Departamento de Matemáticas, Universidad de los Andes, Bogotá Colombia ve.gauthier@uniandes.edu.co,
[4] Normandie Univ, France; UR, LITIS, F-76821 Mont-Saint-Aignan, France. ayoub.otmani@univ-rouen.fr
[5] SECRET Project - INRIA Rocquencourt, 78153 Le Chesnay Cedex, France. jean-pierre.tillich@inria.fr

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.

1 Introduction

The first cryptographic scheme using generalized Reed-Solomon codes was proposed in 1986 by Niederreiter [Nie86] but it was shown to be insecure in [SS92]. The attack recovers the underlying Reed-Solomon code allowing the decoding of any encrypted data. However during the past years there were several attempts to repair this scheme. The first one was proposed by Wieschebrink [Wie06] and consists in choosing a generator matrix of a generalized Reed-Solomon code and adding to it a few random columns. It was advocated that this modification avoids the Sidelnikov-Shestakov attack [SS92]. The second one is another variant of McEliece's cryptosystem [McE78] proposed in [BBC$^+$11] which uses this time a generator matrix of a generalized Reed-Solomon but hides its structure differently than in the McEliece cryptosystem: instead of multiplying by a permutation matrix, the generator matrix is multiplied by a matrix whose inverse is of the form $\boldsymbol{\Pi} + \boldsymbol{R}$ where $\boldsymbol{\Pi}$ is a permutation matrix and $\boldsymbol{R}$ is a rank 1 matrix. The key point of this modification is that the public code obtained with this method is not anymore a generalized Reed-Solomon code and this seems to thwart the Sidelnikov and Shestakov attack completely. More recently, some of the nice algebraic properties of the Reed-Solomon codes were also used to devise the first public-key homomorphic encryption scheme [BL11] based on coding theory.

Contrarily to the Niederreiter's proposal [Nie86] based on generalized Reed-Solomon codes, the original McEliece cryptosystem [McE78] which uses Goppa codes, has withstood many key-recovery attacks and after more than thirty years now, it still belongs to the very few unbroken public-key cryptosystems. No significant breakthrough has been observed with respect to the problem of recovering the private key. For instance, the weak keys found in [Gib91,LS01] can be easily avoided. This fact has led to claim that the generator matrix of a binary Goppa code does not disclose any visible structure that an attacker could exploit. This is strengthened by the fact that Goppa codes share many characteristics with random codes.

However, in [FGO$^+$11], an algorithm that manages to distinguish between a random code and a Goppa code has been introduced. This work, without undermining the security of [McE78], prompts to wonder whether it would be possible to devise an attack based on such a distinguisher. It turns out [MCP12] that the distinguisher in [FGO$^+$11] has an equivalent but simpler description in terms of the component-wise product of codes. This notion was first put forward in coding theory to unify many different algebraic decoding algorithms [Pel92,Köt92]. Recently, it was used in [MCMMP11a,MCMMP12a] to study the security of cryptosystems based on Algebraic-Geometric codes. Powers of codes are also studied in the context of secure multi-party computation (see for example [CCCX09,CCX11]). This distinguisher is even more powerful in the case of Reed-Solomon codes than for Goppa codes because, whereas for Goppa codes it is only successful for rates close to 1, it can distinguish Reed-Solomon codes of any rate from random codes.

In the specific case of [BL11], the underlying public code is a modified Reed-Solomon code obtained from the insertion of a zero submatrix in the Vandermonde generating matrix defining it and in this case, the aforementioned distinguisher leads to an attack. More exactly, we present a key-recovery attack on the Bogdanov-Lee homomorphic scheme based on the version of our distinguisher presented in [MCP12]. Our attack runs in polynomial time and is efficient: it only amounts to calculate the ranks of certain matrices derived from the public key. In [BL11] the columns that define the zero submatrix are kept secret and form a set L. We give here a distinguisher that detects if one or several columns belong to L or not. It is constructed by considering the code generated by component-wise products of codewords of the public code (the so-called "square code"). This operation is applied to punctured versions of this square code obtained by picking a subset I of the whole set indexing the columns. It turns out that the dimension of the punctured square code is directly related to the cardinality of the intersection of I with L. This gives a way to recover the full set L allowing the decryption of any ciphertext.

To demonstrate further the power of this approach, we propose another cryptanalysis against the variant of McEliece's cryptosystem [McE78] proposed in [BBC$^+$11]. As explained above, the public code obtained with this method is not anymore a generalized Reed-Solomon code. On the other hand, it contains a very large secret generalized Reed-Solomon code. We present an attack that is based on a distinguisher which is able to identify elements of this secret code. This distinguisher is again derived from considerations about the dimension of component-wise products of codes. Once this secret code is obtained, it is then possible to completely recover the initial generalized Reed-Solomon code by using the square-code construction as in [Wie10]. We are then able to decode any ciphertext.

Finally, we also cryptanalyze the first variant of the McEliece's cryptosystem based on Reed-Solomon codes [Wie06]. We show here how a refinement of our distinguisher permits to recover the random columns added to the generator matrix of the generalized Reed-Solomon code. Once these column positions are recovered, the Sidelnikov and Shestakov attack can be used on the non-random part of the generator matrix to completely break the scheme.

It should also be pointed out that the properties of Reed-Solomon codes with respect to the component-wise product of codes have already been used to cryptanalyze a McEliece-like scheme [BL05] based on subcodes of Reed-Solomon codes [Wie10]. The use of this product is nevertheless different in [Wie10] from the way we use it here. Note also that our attack is not an adaptation of the Sidelnikov and Shestakov approach [SS92]. Our approach is completely new: it illustrates how a distinguisher that detects an abnormal behavior can be used to recover a private key.

2 Reed-Solomon Codes and the Square Code

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. 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 (Reed-Solomon code and 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 tuple $(\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{F}_q^n \times \mathbb{F}_q^n$ where $\boldsymbol{x}$ is an n-tuple of distinct elements of $\mathbb{F}_q$ and the entries y_i are arbitrary non zero elements in $\mathbb{F}_q$. It is defined as $\mathbf{GRS}_k(\boldsymbol{x}, \boldsymbol{y}) \overset{def}{=} \{(y_1 p(x_1), \dots, y_n p(x_n)) : p \in \mathbb{F}_q[X], \deg p < k\}$. Reed-Solomon codes correspond to the case where $y_i = 1, \forall i$.*

It has been suggested to use them in a public-key cryptosystem for the first time in [Nie86] but it was discovered that this scheme is insecure in [SS92]. Sidelnikov and Shestakov namely showed that it is possible to recover in polynomial time for any GRS code a tuple $(\boldsymbol{x}, \boldsymbol{y})$ which defines it. This is all what is needed to decode efficiently such codes and is therefore enough to break the Niederreiter cryptosystem suggested in [Nie86] or a McEliece type cryptosystem [McE78] when GRS codes are used instead of Goppa codes.

A GRS code displays a quite peculiar property with respect to the component-wise product which is denoted by $\boldsymbol{a} \star \boldsymbol{b}$ for two vectors $\boldsymbol{a} = (a_1, \dots, a_n)$ and $\boldsymbol{b} = (b_1, \dots, b_n)$ and which is defined by $\boldsymbol{a} \star \boldsymbol{b} \overset{\text{def}}{=} (a_1 b_1, \dots, a_n b_n)$. This can be seen by bringing in the following definition

Definition 2 (Star 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 $\mathscr{B} = \mathscr{A}$, $< \mathscr{A} \star \mathscr{A} >$ is called the* square code *of $\mathscr{A}$ and is 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

Proposition 1.

$$\dim(< \mathscr{A} \star \mathscr{B} >) \leq \dim(\mathscr{A}) \dim(\mathscr{B}).$$

We expect that the square code when applied to a random linear code 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] that with probability going to 1 as k tends to infinity, the square code is of dimension $\min\left\{\binom{k+1}{2}(1 + o(1)), n\right\}$ when k is of the form $k = o(n^{1/2})$. On the other hand, GRS codes behave in a completely different way

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

This follows immediately from the definition of a GRS code as an evaluation code since the star product of two elements $\boldsymbol{c} = (y_1 p(x_1), \dots, y_n p(x_n))$ and $\boldsymbol{c}' = (y_1 q(x_1), \dots, y_n q(x_n))$ 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

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

where r is a polynomial of degree $\leq 2k-2$. Conversely, any element of the form $(y_1^2 r(x_1), \dots, y_n^2 r(x_n))$ 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 when $2k-1 \leq n$, the square code is only of dimension $2k-1$, which is abnormally small. 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 3.

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

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 on* $\boldsymbol{x}$ *and* $\boldsymbol{y}$*.*

Therefore when $2k-1 > n$ a Reed-Solomon code $\mathbf{GRS}_k(\boldsymbol{x},\boldsymbol{y})$ can also be distinguished from a random linear code of the same dimension by computing the dimension of $< \left(\mathbf{GRS}_k(\boldsymbol{x},\boldsymbol{y})^{\perp}\right)^2 >$. We have in this case $< \left(\mathbf{GRS}_k(\boldsymbol{x},\boldsymbol{y})^{\perp}\right)^2 >=< \mathbf{GRS}_{n-k}(\boldsymbol{x},\boldsymbol{y}')^2 >=< \mathbf{GRS}_{2n-2k-1}(\boldsymbol{x},\boldsymbol{y}' \star \boldsymbol{y}') >$ and we obtain 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 GRScode $\mathbf{GRS}_k(\boldsymbol{x},\boldsymbol{y})$ a possible pair $(\boldsymbol{x},\boldsymbol{y})$. This is achieved by computing $< \mathscr{C}^2 >$ which in the case which is considered turns out to be equal to $< \mathbf{GRS}_k(\boldsymbol{x},\boldsymbol{y})^2 >$ which is equal to $\mathbf{GRS}_{2k-1}(\boldsymbol{x},\boldsymbol{y} \star \boldsymbol{y})$. 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. From this, a possible $(\boldsymbol{x},\boldsymbol{y})$ pair for which $\mathscr{C} \subset \mathbf{GRS}_k(\boldsymbol{x},\boldsymbol{y})$ is deduced. Whereas in our case we really use directly that the square code of a GRS code has abnormally small dimension by computing for instance the dimensions of

- the square of various subcodes of the public code to detect random columns in the generator matrix of the public code: this is basically the attack on Wieschebrink's cryptosystem presented in Section 5;
- the square of various punctured versions of the public code in the Bogdanov and Lee case in order to retrieve the columns which correspond to the Reed-Solomon part;
- or identifying a certain subcode of the public code by this means in the Baldi et al. case.

3 The Bogdanov-Lee Homomorphic Cryptosystem

3.1 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. The key generation requires a non-negative integer ℓ such that $3\ell < n$ and a subset L of $\{1,\dots,n\}$ of cardinality 3ℓ. A set of n distinct elements $x_1,\dots,x_n$ from $\mathbb{F}_q$ 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 $\boldsymbol{G}_i^T \overset{\text{def}}{=} \begin{cases} (x_i, x_i^2, \dots, x_i^{\ell}, 0, \dots, 0) & \text{if } i \in L \\ \\ (x_i, x_i^2, \dots, x_i^{\ell}, x_i^{\ell+1}, \dots, x_i^k) & \text{if } i \notin L \end{cases}$ where the symbol ^{T} stands for the transpose. The cryptosystem is defined as follows.

1. **Secret key.** $(L, \boldsymbol{G})$.
2. **Public key.** $\boldsymbol{P} \overset{\text{def}}{=} \boldsymbol{S}\boldsymbol{G}$ 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 $\boldsymbol{c} \overset{\text{def}}{=} \boldsymbol{x}\boldsymbol{P} + m\mathbf{1} + \boldsymbol{e}$ where $\mathbf{1} \in \mathbb{F}_q^n$ 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.** The linear system (1) is solved for $\boldsymbol{y} \overset{\text{def}}{=} (y_1, \dots, y_n) \in \mathbb{F}_q^n$:

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

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

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[6] $\eta = \Theta\left(1/n^{1-\alpha/4}\right)$ where α is a in $(0, \frac{1}{4}]$ (for more 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 Attack on the Bogdanov-Lee Scheme

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

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

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 (2) is equivalent to the "secret" system (1). 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 (2). In the following subsection, we will explain how L can be recovered from $\boldsymbol{P}$ in polynomial time.

Recovering L. 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 3. *For any $I \subset \{1, \dots, n\}$ of cardinality $|I|$, the restriction of a code $\mathscr{A}$ of length n is the subset of $\mathbb{F}_q^{|I|}$ defined as $\mathscr{A}_I \overset{def}{=} \left\{\boldsymbol{v} \in \mathbb{F}_q^{|I|} \mid \exists \boldsymbol{a} \in \mathscr{A}, \boldsymbol{v} = (a_i)_{i \in I}\right\}$.*

The results about the unusual dimension of the square of a Reed-Solomon codes which are given in Section 2 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.

[6] 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.

Proposition 4. *Let I be a subset of $\{1,\dots,n\}$ and set $J \overset{\text{def}}{=} I \cap L$. If the cardinality of I and J satisfy $|J| \leqslant \ell - 1$ and $|I| - |J| \geqslant 2k$ then*

$$\dim(< \mathscr{C}_I^2 >) = 2k - 1 + |J| \,. \tag{3}$$

The proof of this proposition can be found in [GOT12a, Appendix A]. 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 4 is randomly chosen. Take for instance $|I| = 3k$. Then $d_I \overset{\text{def}}{=} \dim(< \mathscr{C}_I^2 >)$ is computed. Next, one element x is removed from I to get a new set I' and $d_{I'} = \dim(< \mathscr{C}_{I'}^2 >)$ is computed. The only two possible cases are either $x \notin L$ then $d_{I'} = d_I$ or $x \in L$ and then $d_{I'} = d_I - 1$. 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 $\{1,\dots,n\} \setminus I$ and compare the values of d_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 Baldi et *al.* Variant of McEliece's Cryptosystem

4.1 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 $\mathbb{F}_q$,
- $\boldsymbol{Q} \overset{\text{def}}{=} \boldsymbol{\Pi} + \boldsymbol{R}$ 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 $\boldsymbol{\alpha} \overset{\text{def}}{=} (\alpha_1,\dots,\alpha_n)$ and $\boldsymbol{\beta} \overset{\text{def}}{=} (\beta_1,\dots,\beta_n)$ in $\mathbb{F}_q^n$ such that $\boldsymbol{R} \overset{\text{def}}{=} \boldsymbol{\alpha}^T\boldsymbol{\beta}$.
- $\boldsymbol{S}$ is a $k \times k$ random invertible matrix over $\mathbb{F}_q$.

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

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 $\boldsymbol{c} \overset{\text{def}}{=} \boldsymbol{m}\boldsymbol{G_{pub}} + \boldsymbol{e}$.

Decryption. It consists in performing the three following steps:

1. Guessing the value of $\boldsymbol{eR}$;
2. Calculating $\boldsymbol{c}' \overset{\text{def}}{=} \boldsymbol{cQ} - \boldsymbol{eR} = \boldsymbol{m}\boldsymbol{S}^{-1}\boldsymbol{G_{sec}} + \boldsymbol{eQ} - \boldsymbol{eR} = \boldsymbol{m}\boldsymbol{S}^{-1}\boldsymbol{G_{sec}} + \boldsymbol{e\Pi}$ and using the decoding algorithm of the GRS code to recover $\boldsymbol{m}\boldsymbol{S}^{-1}$ 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 $\boldsymbol{eR} = \boldsymbol{e}\boldsymbol{\alpha}^T\boldsymbol{\beta} = \gamma\boldsymbol{\beta}$ where γ is an element of $\mathbb{F}_q$.

4.2 Attack on the Baldi et *al.* Cryptosystem when $2k + 2 < n$

We define $\mathscr{C}_{sec}$ and $\mathscr{C}_{pub}$ 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

$$2k + 2 < n \tag{4}$$

As explained in Subsection 4.1, $\mathscr{C}_{sec}$ is a GRS code. It will be convenient to bring in the code $\mathscr{C} \overset{\text{def}}{=} \mathscr{C}_{\text{sec}}\boldsymbol{\Pi}^{-1}$. From [GOT12b, Lemma 3, Appendix A], the matrix $\boldsymbol{R}\boldsymbol{\Pi}^{-1}$ is also of rank one. Hence there exist $\boldsymbol{a}$ and $\boldsymbol{b}$ in $\mathbb{F}_q^n$ such that:

$$\boldsymbol{R}\boldsymbol{\Pi}^{-1} = \boldsymbol{b}^T\boldsymbol{a}. \tag{5}$$

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 the following lemma.

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

$$\boldsymbol{c} = \boldsymbol{p} + (\boldsymbol{p}\cdot\boldsymbol{\lambda})\boldsymbol{a}. \tag{6}$$

The proof of this lemma is given in [GOT12b, Lemma 2, §4.2]. From now on we make the assumption that

$$\boldsymbol{\lambda} \notin \mathscr{C}^{\perp}. \tag{7}$$

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:

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

where $< \boldsymbol{\lambda} >$ denotes the vector space spanned by $\boldsymbol{\lambda}$. It is a subspace of $\mathscr{C}_{\text{pub}}$ of codimension 1 if $\boldsymbol{\lambda} \notin \mathscr{C}^{\perp}$. This strongly suggests that $< \mathscr{C}_{\text{pub}}^2 >$ should have an unusual low dimension since $< \mathscr{C}^2 >$ has dimension $2k - 1$ by Proposition 2. More exactly we have here:

Proposition 5.
1. $< \mathscr{C}_{pub}^2 > \;\subset\; < \mathscr{C}^2 > \;+\; \mathscr{C}\star\boldsymbol{a} \;+\; < \boldsymbol{a}\star\boldsymbol{a} >$
2. $\dim\left(< \mathscr{C}_{pub}^2 >\right) \leqslant 3k - 1$

The first fact follows immediately from Lemma 1 and the proof of this proposition is given in [GOT12b, Appendix A] Experimentally it has been observed that the upper-bound on the dimension 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 $\boldsymbol{g}_1, \dots, \boldsymbol{g}_k$ of $\mathscr{C}_{\text{pub}}$ is chosen and l other random elements $\boldsymbol{z}_1, \dots, \boldsymbol{z}_l$, then we may expect that the dimension of the vector space generated by all products $\boldsymbol{z}_i \star g_j$ with i in $\{1, \dots, l\}$ and j in $\{1, \dots, k\}$ 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.

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

```
Input: A basis $\{g_1, \dots, g_k\}$ of $\mathscr{C}_{\mathrm{pub}}$.
Output : A basis $\mathcal{L}$ of $\mathscr{C}_{\boldsymbol{\lambda}^\perp}$.
repeat
    for $1 \leqslant i \leqslant 3$ do
        Randomly choose $z_i$ in $\mathscr{C}_{\mathrm{pub}}$
    end for
    $\mathscr{B} \leftarrow \ < \{ z_i \star g_j \mid 1 \leqslant i \leqslant 3 \text{ and } 1 \leqslant j \leqslant k \} >$
until $\dim(\mathscr{B}) \leqslant 2k+2$ and $\dim(< z_1, z_2, z_3 >) = 3$
$\mathcal{L} \leftarrow \{z_1, z_2, z_3\}$
$s \leftarrow 4$
while $s \leqslant k-1$ do
    repeat
        Randomly choose $z_s$ in $\mathscr{C}_{\mathrm{pub}}$
        $\mathscr{T} \leftarrow \ < \{ z_i \star g_j \mid i \in \{1,2,s\} \text{ and } 1 \leqslant j \leqslant k \} >$
    until $\dim(\mathscr{T}) \leqslant 2k+2$ and $\dim(< \mathcal{L} \cup \{z_s\} >) = s$
    $\mathcal{L} \leftarrow \mathcal{L} \cup \{z_s\}$
    $s \leftarrow s+1$
end while
return $\mathcal{L}$;
```

Proposition 6. *Let $\mathscr{B}$ be the space spanned by $\{ z_i \star g_j \mid 1 \leqslant i \leqslant 3,\ 1 \leqslant j \leqslant k \}$, then* $\dim(\mathscr{B}) \leqslant 3k-3$.

An explanation of this phenomenon is given in [GOT12b, Appendix A]. 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 z_1, z_2, 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 $\mathscr{C}_{\mathrm{pub}}$ of codimension 1 (at least when $\boldsymbol{\lambda} \notin \mathscr{C}^\perp$), then the vectors $z_i \star \boldsymbol{g}_j$ generate a subspace with a much smaller dimension.

Proposition 7. *If z_i is in $\mathscr{C}_{\boldsymbol{\lambda}^\perp}$ for i in $\{1,2,3\}$ then for all j in $\{1,\dots,k\}$:*

$$z_i \star \boldsymbol{g}_j \ \subset \ <\mathscr{C}^2> \ + \ <z_1 \star \boldsymbol{a}> \ + \ <z_2 \star \boldsymbol{a}> \ + \ <z_3 \star \boldsymbol{a}> \tag{8}$$

and if $\mathscr{B}$ is the linear code spanned by $\{ z_i \star g_j \mid 1 \leqslant i \leqslant 3$ and $1 \leqslant j \leqslant k \}$ then

$$\dim(\mathscr{B}) \leqslant 2k+2. \tag{9}$$

The proof of this proposition is straightforward and is given in [GOT12b, Appendix A]. The upper-bound given in (9) on the dimension follows immediately from (8). This leads to Algorithm 1 which computes a basis of $\mathscr{C}_{\boldsymbol{\lambda}^\perp}$. It is essential that the condition in (4) 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 z_1, z_2, z_3 runs in expected time $O(k^3q^3)$ because each test in the **repeat** loop 1 has a chance of $\frac{1}{q^3}$ to succeed. Indeed, $\mathscr{C}_{\boldsymbol{\lambda}^\perp}$ is of codimension 1 in $\mathscr{C}_{\mathrm{pub}}$ and therefore a fraction $\frac{1}{q}$ of elements of $\mathscr{C}_{\mathrm{pub}}$ belongs to $\mathscr{C}_{\boldsymbol{\lambda}^\perp}$. Once z_1, z_2, z_3 are found, getting any other element of $\mathscr{C}_{\boldsymbol{\lambda}^\perp}$ is easy. Indeed, take a random element $z \in \mathscr{C}_{\mathrm{pub}}$ and use the same test to check whether the triple z_1, z_2, z is in $\mathscr{C}_{\boldsymbol{\lambda}^\perp}$. Since $z_1, z_2 \in \mathscr{C}_{\boldsymbol{\lambda}^\perp}$ the probability of success is $\frac{1}{q}$ and hence z can be found in $O(q)$ tests. The whole algorithm runs in expected time $O(k^3q^3) + O(k^4q) = O(k^3q^3)$ since $k = O(q)$ and the first phase of the attack is dominant in the complexity. 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 [MCMMP11b,MCMMP12b] for more details). It suffices to use the Sidelnikov and Shestakov algorithm [SS92] 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. The final phase, that is, recovering a possible $(\boldsymbol{\lambda}, \boldsymbol{a})$ pair and using it to decode the public code $\mathscr{C}_{\text{pub}}$, is detailed in [GOT12b, Appendix B].

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

$$\boldsymbol{c} = \boldsymbol{p} + (\boldsymbol{p} \cdot \boldsymbol{a})\boldsymbol{b}. \tag{10}$$

The proof of this lemma is given in [GOT12b, Appendix A]. It implies that the whole approach of the previous subsection can be carried out over $\mathscr{C}_{\text{pub}}^\perp$. 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 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 $C_1, \dots, C_r$ 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 $C_1, \dots, C_r$. 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

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

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

Lemma 1 *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*

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

The proof of this statement is given in Appendix B. Actually the right-hand inequality of Lemma 1 is sharp and with very high probability we observe that

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

A discussion which explains this behavior is given in Appendix B. When $2k-1+r \geqslant n$, $< \mathscr{C}'^2 >$ is typically the whole ambient space $\mathbb{F}_q^n$. This will be useless 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 1. *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

  $$\dim < \mathscr{C}'(i)^2 >= 2k - 2 + r.$$

  since $\mathcal{C}'(i)$ is nothing but a k-dimensional GRS code with $r - 1$ random columns added to its generator matrix.
- *i belongs to the GRS positions*, then $\mathcal{C}'(i)$ is a $k - 1$-dimensional GRS code with r random columns added to its generator matrix and we expect that

  $$\dim < \mathscr{C}'(i)^2 >= 2k - 3 + r.$$

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 $k > 2k + r - n \implies n > k + r$. Among these a positions, a_0 of them are random positions and $a_1 \overset{\text{def}}{=} a - a_0$ 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,

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

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].

6 Conclusion

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. 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.[7]

Acknowledgement: We thank the anonymous reviewer for a careful reading of this submission which helped us to improve its editorial quality.

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

A Correctness of the Bogdanov-Lee decoding procedure

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 (1) 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

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

B Proof of Lemma 1

Let us prove Lemma 1.

Proof. The first inequality comes from the fact that puncturing $\mathcal{C}'^2$ at the r positions corresponding to the added random columns yields the code $\mathcal{C}^2$ 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

$$\mathscr{C}' \subset \mathscr{D} + \mathscr{D}'. \tag{11}$$

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 $\dim < \mathscr{D}'^2 >\leq r$.

References

[BBC$^+$11] M. Baldi, M. Bianchi, F. Chiaraluce, J. Rosenthal, and D. Schipani. Enhanced public key security for the McEliece cryptosystem. submitted, 2011. See arxiv:1108.2462v2[cs.IT]. After learning that their scheme was attacked by this paper, the authors changed their scheme accordingly and the new version is now arxiv:1108.2462v3[cs.IT].

[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.

[CCCX09] I. Cascudo, H. Chen, R. Cramer, and C. Xing. Asymptotically Good Ideal Linear Secret Sharing with Strong Multiplication over Any Fixed Finite Field. In S. Halevi, editor, *Advances in Cryptology - CRYPTO 2009*, volume 5677 of *Lecture Notes in Computer Science*, pages 466–486. Springer Berlin / Heidelberg, 2009.

[CCX11] I. Cascudo, R. Cramer, and C. Xing. The Torsion-Limit for Algebraic Function Fields and Its Application to Arithmetic Secret Sharing. In P. Rogaway, editor, *Advances in Cryptology CRYPTO 2011*, volume 6841 of *Lecture Notes in Computer Science*, pages 685–705. Springer Berlin / Heidelberg, 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.

[Gib91] J. Gibson. Equivalent Goppa codes and trapdoors to McEliece's public key cryptosystem. In Donald Davies, editor, *Advances in Cryptology – EUROCRYPT 91*, volume 547 of *Lecture Notes in Computer Science*, pages 517–521. Springer Berlin / Heidelberg, 1991.

[GOT12a] V. Gauthier, A. Otmani, and J.-P. Tillich. A distinguisher-based attack of a homomorphic encryption scheme relying on Reed-Solomon codes, 2012. http://arxiv.org/abs/1203.6686.

[GOT12b] V. Gauthier, A. Otmani, and J.-P. Tillich. A distinguisher-based attack on a variant of McEliece's cryptosystem based on Reed-Solomon codes, 2012. http://arxiv.org/abs/1204.6459.

[Köt92] R. Kötter. A unified description of an error locating procedure for linear codes. In *Proc. Algebraic and Combinatorial Coding Theory*, pages 113–117, Voneshta Voda, 1992.

[LS01] P. Loidreau and N. Sendrier. Weak keys in the McEliece public-key cryptosystem. *IEEE Transactions on Information Theory*, 47(3):1207–1211, 2001.

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

[MCMMP11a] I. Márquez-Corbella, E. Martínez-Moro, and R. Pellikaan. Evaluation of public-key cryptosystems based on algebraic geometry codes. In J. Borges and M. Villanueva, editors, *Proceedings of the Third International Castle Meeting on Coding Theory and Applications*, pages 199–204, Barcelona, Spain, September 11-15 2011.

[MCMMP11b] 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.

[MCMMP12a] I. Márquez-Corbella, E. Martínez-Moro, and R. Pellikaan. On the unique representation of very strong algebraic geometry codes. *Designs, Codes and Cryptography*, pages 1–16, 2012.

[MCMMP12b] Irene Márquez-Corbella, Edgar Martínez-Moro, and Ruud Pellikaan. The non-gap sequence of a subcode of a generalized reedsolomon 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.

[Pel92] R. Pellikaan. On decoding by error location and dependent sets of error positions. *Discrete Mathematics*, 106-107:368–381, 1992.

[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.