An Attack on High Rate McEliece Cryptosystems Using Generalized Reed Solomon Codes with Weight 2 Mask

Julia Lieb, Abhinaba Mazumder, Michael Schaller

July 27, 2026

Abstract

Due to the insecurity of McEliece cryptosystems instantiated with Generalized Reed-Solomon codes, there have been several proposals of McEliece type systems that replace the permutation matrix by a matrix M with larger row and column weight. In many of them, the secret key is still a GRS code. There have been successful attacks on some of those schemes with row and column weight between 1 and $1 + R$, where R is the rate of the code. The case of weight two and larger has been left open in these works. Subsequently, several authors proposed schemes with weight exactly two and with even higher weight. We provide distinguishers for the public codes appearing in these cryptosystems in the high rate regime. In addition, we give a framework to turn a good enough distinguisher into a key-recovery attack. In the case where the matrix M has row and column weight 2, we can successfully attack the scheme in the high rate regime using a cube code distinguisher.

1 Introduction

After Sidelnikov and Shestakov [13] broke the McEliece cryptosystem instantiated with generalized Reed-Solomon (GRS) codes [10], there have been proposals to modify the cryptosystem in order to still use GRS codes for the private key [1, 2, 3, 14]. The modifications are mainly based on using as a mask a matrix with higher row and column weight instead of a permutation matrix in the McEliece cryptosystem with GRS codes. The schemes [2, 3] which use row and column weight between 1 and $1 + R$ where R is the rate of the code have been successfully attacked in [6, 7]. The case of row and column weight equal to 2, which is used in [14, 4, 1], has been left open in [6, 7]. A similar scheme with higher weight was proposed in [8]. We provide a framework that shows that having access to a certain distinguisher, we are able to attack the McEliece cryptosystem with GRS codes also when a mask with row and column weight two is used as proposed in [14]. Under certain heuristic assumptions, we obtain that the cube code can serve as such a distinguisher for high enough code rates. Using shortening, one can improve the range of the distinguisher to lower rates. However, extending it to all rates remains an open problem. In the high rate regime without shortening where $n > 2k^2 - 4k + 4$, where k is the dimension of the dual code, we are able to turn the distinguisher into a key-recovery attack.

The distinguisher only uses column weight equal to 2, while key recovery becomes a bit more complicated if only the column weight is 2 without the row weight also being 2. For this reason, we only claim an attack on [14] in the high rate regime. However, it seems likely that it is possible

to adapt the attack to the case of only column weight equal to 2 and one can certainly recover some information about the private key of [1, 4] in the high rate regime.

The basic idea for the attack in the high rate regime is similar to the ideas in [6, 7] in the sense that we use a distinguisher in order to recover information about the private key. We are given a public code of the form

$$\mathrm{GRS}_k(P, \mu)M$$

where the evaluation points and column multipliers of the GRS code are secret and M is a secret invertible matrix with row and column weight 2. In order to recover the mask M, we multiply a generator matrix of the public code $\mathrm{GRS}_k(P, \mu)M$ on the right with a matrix N that corresponds to an elementary column operation adding the multiple of one column to another column. In this way we get the code

$$\mathrm{GRS}_k(P, \mu)MN.$$

We apply a distinguisher to the code $\mathrm{GRS}_k(P, \mu)MN$ to find among the matrices N those for which the number of non-zero entries in MN does not exceed the number of non-zero entries in M. Since all columns of M are of weight 2, this corresponds to finding cancellation patterns when performing column operations on M. This information about the mask M allows us to recover all of M (or rather a matrix which is in some sense equivalent) so that we can apply the Sidelnikov-Shestakov [13] attack or a filtration attack [6] to recover a GRS code with which we can decode the encrypted message.

The paper is organized as follows. In Section 2 we recall the necessary background from coding theory. In Section 3 we describe the McEliece cryptosystem with GRS codes and weight 2 mask. In Section 4 we show that for high enough rate the cube code provides a distinguisher for the public codes in the cryptosystem from random codes. This gives evidence that we can use the cube code as the distinguisher for cancellation as described in the introduction. We also show an improvement on the distinguisher using shortening. Section 5 is at the heart of the paper. First we describe the framework showing how a distinguisher as described in the introduction leads to a key-recovery attack. Then we show how we can instantiate it in the high rate regime with the cube code. We conclude this section with the experiments of our attack. In Section 6 we describe several interesting open problems.

2 Basics of Linear Codes

Definition 1. Let $v = (v_1, \ldots, v_n), w = (w_1, \ldots, w_n) \in \mathbb{F}_q^n$. We define the **standard inner product** between v and w as

$$\langle v, w \rangle := \sum_{i=1}^{n} v_i w_i.$$

Definition 2. Let $k \leq n$ be positive integers and let C be an $[n, k]_q$ linear code over $\mathbb{F}_q$, i.e., C is a k-dimensional subspace of $\mathbb{F}_q^n$. Then, the linear $[n, n-k]_q$ code

$$C^{\perp} := \{x \in \mathbb{F}_q^n \mid \langle c, x \rangle = 0 \ \forall \ c \in C\}$$

is called the **dual code** of C.

Definition 3. Let $k \leq n \leq q$ be positive integers. Let $P = (P_1, \ldots, P_n) \in \mathbb{F}_q^n$ with $P_i \neq P_j$, for all $i \neq j \in \{1, \ldots, n\}$. Let $\mu = (\mu_1, \ldots, \mu_n) \in \mathbb{F}_q^n$ with $\mu_i \neq 0$ for all $i \in \{1, \ldots, n\}$. Then,

$$\mathrm{GRS}_k(P, \mu) = \left\{ (\mu_1 f(P_1), \ldots, \mu_n f(P_n)) \;\middle|\; f \in \mathbb{F}_q[x],\ \deg(f) < k \right\}$$

is called a **Generalized Reed-Solomon (GRS)** code of length n and dimension k over $\mathbb{F}_q$.

Lemma 4 ([11], Prop. 5.1.26)**.** *Let $k \leq n \leq q$ be positive integers. Then,*

$$\mathrm{GRS}_k(P, \mu)^{\perp} = \mathrm{GRS}_{n-k}(P, \tilde{\mu}),$$

where

$$\tilde{\mu}_i = \mu_i^{-1} \prod_{\substack{j=1 \\ j \neq i}}^{n} (P_i - P_j)^{-1}.$$

In the following, we introduce power codes. The special case of square codes has become prominent as a distinguisher in GRS code based McEliece type cryptosystems; see e.g. [6, 7].

Definition 5. Let $v = (v_1, \ldots, v_n), w = (w_1, \ldots, w_n) \in \mathbb{F}_q^n$. The **Schur product** of v and w is defined as

$$v * w := (v_1 w_1, \ldots, v_n w_n).$$

Let $C_1, C_2 \subset \mathbb{F}_q^n$ be two linear codes. The **Schur product** of C_1 and C_2 is defined as

$$C_1 * C_2 := \langle \{ c_1 * c_2 \mid c_1 \in C_1,\ c_2 \in C_2 \} \rangle \subset \mathbb{F}_q^n.$$

For a linear code $C \subset \mathbb{F}_q^n$, we call $C * C$ the **square code** of C and denote it by $C^{(2)}$. More generally, we call $C^{(w)} := \underbrace{C * \cdots * C}_{w \text{ times}}$ the **w-th order power code of** C. In particular, we call $C^{(3)}$ the **cube code** of C.

The reason why the square code can serve as a distinguisher for GRS code based McEliece cryptosystems where the mask has row and column weight at most $1 + R$ where R is the code rate, is that the dimension of the square code of a GRS code is smaller than the dimension of the square code of a random code. It is easy to see that

$$\dim(GRS_k(P, \mu)^{(2)}) = \min \{2k - 1, n\}.$$

The following theorem provides the dimension of the square code of a random code.

Theorem 6 ([5, Theorem 2.3])**.** *For a random linear code $C \subset \mathbb{F}_q^n$ of dimension k, one has with high probability that*

$$\dim(C^{(2)}) = \min \left\{ \binom{k+1}{2}, n \right\}.$$

In this paper, we will employ the cube code as a distinguisher for GRS codes with a weight two mask. It is conjectured that the dimension of the cube code of a random code of dimension k and length n is with high probability equal to

$$\min \left\{ \binom{k+2}{3}, n \right\}.$$

To the best of our knowledge, a proof of this result is still missing.

In Section 4, we will show that the dimension of the cube code of a weight two masked GRS code of dimension k and length n is upper bounded by

$$\min\{n, 2k^2 - 4k + 4\}.$$

Later in the paper we will also need the following lemma about power codes of sums of codes, which is easy to see.

Lemma 7. *Let $m, w, n \in \mathbb{N}$ and let $C_1, \ldots, C_m \subset \mathbb{F}_q^n$ be linear codes. Then,*

$$\left(\sum_{i=1}^{m} C_i\right)^{(w)} \subseteq \sum_{i_1,\ldots,i_w \in \{1,\ldots,m\}} C_{i_1} * \cdots * C_{i_w}$$

Definition 8. Let C be an $[n,k]_q$ linear code. Let $J \subseteq \{1,\ldots,n\}$. We define the **punctured code** of C in the coordinates of J to be

$$\mathrm{Pct}_J(C) := \{(c_i)_{i \notin J} \mid c \in C\}.$$

Definition 9. Let C be an $[n,k]_q$ linear code and let $J \subseteq \{1,\ldots,n\}$. We define the code

$$C(J) := \{c \in C : c_i = 0 \text{ for all } i \in J\}. \tag{1}$$

We define the **shortened code** of C in the coordinates of J to be

$$\mathrm{Sh}_J(C) := \mathrm{Pct}_J(C(J)).$$

3 Description of the McEliece Cryptosystem with GRS Codes and Weight Two Mask

In the following, we describe the setup of the weight two masked McEliece cryptosystem with GRS codes [14]. Note that we describe the codes here in a slightly different manner using duals and transposes to make the exposition simpler. Both representations can be derived and computed in a simple manner from each other using Lemma 4.

The private key consists of the following data:

- A secret GRS code $C_{sec} = \mathrm{GRS}_k(P, \mu)^\perp$ over $\mathbb{F}_q$ of length n, dimension $n-k$, the dual of a GRS code with vector of evaluation points P and vector of column multipliers μ with an efficient decoding algorithm and generator matrix $G_{sec} \in \mathbb{F}_q^{(n-k)\times n}$.
- An $(n-k) \times (n-k)$ change of basis matrix S over $\mathbb{F}_q$.
- An invertible $n \times n$ matrix M over $\mathbb{F}_q$ of row and column weight two, i.e., a matrix in which each row and column has exactly two non-zero entries.

The public key then consists of the matrix $G_{pub} = SG_{sec}(M^T)^{-1}$.

Next, we explain encryption and decryption. Let $t_{GRS} = \lfloor \frac{k}{2} \rfloor$ be the error correction capability of the GRS code C_{sec}. We set

$$t = \left\lfloor \frac{t_{GRS}}{2} \right\rfloor.$$

Now, in order to encrypt, the sender who wants to send message $m \in \mathbb{F}_q^{n-k}$ sends

$$c = mG_{pub} + e$$

where e is a random vector of weight t.

The receiver of the ciphertext c decrypts as follows. She multiplies c by M^T to get $cM^T = mSG_{sec} + eM^T$ with the weight of eM^T being at most t_{GRS}. Therefore, she can decode using the GRS decoder for C_{sec} to get mS and multiplying that by S^{-1} yields the original message m.

To switch between a weight two masked code and its dual, we use the following well-known lemma.

Lemma 10. *Let C be an $[n, n-k]_q$-code. Let $D = C(M^T)^{-1}$ for an $n \times n$ invertible matrix M. Then we have*

$$D^\perp = C^\perp M.$$

Proof. Let $v^\perp = c^\perp M$ for $c^\perp \in C^\perp$ and let $v = c(M^T)^{-1}$ for $c \in C$. Then we have

$$v^\perp v^T = c^\perp M((M^T)^{-1})^T c^T = c^\perp c^T = 0.$$

Hence $v^\perp \in D^\perp$ which yields the inclusion $C^\perp M \subseteq D^\perp$. Since the dimensions of $D^\perp$ and $C^\perp$ are the same, equality follows. □

Applying the previous lemma to $C = \mathrm{GRS}_k(P, \mu)^\perp$ and D the code generated by G_{pub} one obtains

$$D^\perp = \mathrm{GRS}_k(P, \mu)M.$$

Here, M is a matrix where all rows and columns have weight exactly two. From now on we will only work with this code $\mathrm{GRS}_k(P, \mu)M$. If we can recover a GRS code and a weight two mask from this code, we can also recover the information for the duals.

In the following, for $v = (v_1, \ldots, v_n) \in \mathbb{F}_q^n$, we define

$$\mathrm{diag}(v) = \begin{pmatrix} v_1 & & 0 \\ & \ddots & \\ 0 & & v_n \end{pmatrix}.$$

With this, we can write

$$M = \Pi_1(\mathrm{diag}(\alpha) + \Pi \cdot \mathrm{diag}(\beta)) \tag{2}$$

with $n \times n$ permutation matrices Π_1, Π and $\alpha = (\alpha_1, \ldots, \alpha_n), \beta = (\beta_1, \ldots, \beta_n) \in (\mathbb{F}_q^*)^n$, where the permutation corresponding to the permutation matrix Π has no fixed points. By abuse of notation, we write Π for both the permutation matrix and the corresponding permutation. Due to the equation

$$\mathrm{GRS}_k(P, \mu)M = \mathrm{GRS}_k(P, \mu)A^{-1}AM \tag{3}$$

which holds for every monomial matrix A, we can work with the equivalent GRS code $\hat{C} = \mathrm{GRS}_k(P,\mu)A^{-1}$ and the corresponding weight two mask $\hat{M} = AM$ instead. In the following, we will use this several times for different monomial matrices A. By abuse of notation, we will call the resulting codes and matrices again $\hat{C}$ and $\hat{M}$ respectively. It will be clear from the context what $\hat{C}$ and $\hat{M}$ will be.

First, we use (3) with $A = \Pi_1^{-1}$ to obtain

$$\hat{M} = \mathrm{diag}(\alpha) + \Pi \cdot \mathrm{diag}(\beta). \tag{4}$$

Let $f \in \mathbb{F}_q[x]_{<k}$ represent a codeword $(\mu_1 f(P_1), \ldots, \mu_n f(P_n))$ of $\mathrm{GRS}_k(P,\mu)$. By abuse of notation we set

$$(\mu_1 f(P_1), \ldots, \mu_n f(P_n)) := (\mu_1 f(P_1), \ldots, \mu_n f(P_n))\Pi_1 \in \mathrm{GRS}_k(P,\mu)\Pi_1 = \hat{C}.$$

We apply (3) a second time, now with $A = \mathrm{diag}(\mu)$ to obtain that

$$\hat{C} = \mathrm{GRS}_k(P,\mu)\Pi_1 \,\mathrm{diag}(\mu)^{-1}$$

is a Reed-Solomon code with evaluation points $P_1, \ldots, P_n$ reordered according to Π_1. By abuse of notation we call them $P_1, \ldots, P_n$ again. Moreover, we get

$$\hat{M} = \mathrm{diag}(\mu * \alpha) + \mathrm{diag}(\mu) \cdot \Pi \cdot \mathrm{diag}(\beta).$$

In order to be able to work with a representation for the codewords of the public code that is as easy as possible, we consider in the following the equivalent code

$$\mathrm{GRS}_k(P,\mu)M \,\mathrm{diag}(\mu * \alpha)^{-1} = \hat{C}\hat{M} \,\mathrm{diag}(\mu * \alpha)^{-1},$$

whose codewords are of the form

$$(f(P_1) + \nu_1 f(P_{\Pi(1)}), \ldots, f(P_n) + \nu_n f(P_{\Pi(n)})) \tag{5}$$

for some $\nu \in (\mathbb{F}_q^*)^n$. In this way, we can describe codewords in $\mathrm{GRS}_k(P,\mu)M \,\mathrm{diag}(\mu * \alpha)^{-1}$ as evaluations of polynomials $f(x_1) + yf(x_2) \in \mathbb{F}_q[x_1, x_2, y]$. Note that for any invertible diagonal matrix $D \in \mathbb{F}_q^{n\times n}$, one has

$$(C \cdot D)^{(3)} = C^{(3)} \cdot D^3,$$

i.e. the cube codes of $\mathrm{GRS}_k(P,\mu)M \,\mathrm{diag}(\mu * \alpha)^{-1}$ and $\mathrm{GRS}_k(P,\mu)M$ have the same dimension.

Remark 11. We use the representation in terms of an equivalent code only to simplify theoretical arguments. For the actual attack we will never have to compute this equivalent code.

This will be used in the next section to show that the cube code of such codes has smaller dimension than the cube code of a random code with the same parameters if the dimension k is sufficiently small.

4 Cube Code Distinguisher

In order to employ the cube code as a distinguisher for the public code in the McEliece cryptosystem with GRS codes and weight two mask, we use the model of the public code as a certain

evaluation code as described at the end of the previous section and we show that such evaluation codes can be distinguished using cube codes or higher-order power codes.

Recall that we consider a public code of the form

$$\mathrm{GRS}_k(P,\mu)M$$

where M is a matrix with row and column weight 2. We define the following space of polynomials in three variables:

$$S_k = \{f(x_1) + yf(x_2) \in \mathbb{F}_q[x_1, x_2, y] : f \in \mathbb{F}_q[x]_{<k}\}. \tag{6}$$

Note that we take the same polynomial f in both terms. S_k is a k-dimensional vector space over $\mathbb{F}_q$ with basis

$$\{1 + y, x_1 + yx_2, \ldots, x_1^{k-1} + yx_2^{k-1}\}. \tag{7}$$

Evaluating the polynomials in S_k at the n points

$$(P_1, P_{\Pi(1)}, \nu_1), \ldots, (P_n, P_{\Pi(n)}, \nu_n) \in \mathbb{F}_q^3$$

gives exactly the code $\mathrm{GRS}_k(P,\mu)M\,\mathrm{diag}(\mu * \alpha)^{-1}$ as described above; see (5). In order to show that we can use the cube code as distinguisher for such codes, we give upper bounds on the dimension of the space $S_k^{(3)}$ over $\mathbb{F}_q$, which is the linear space generated by three products in S_k. Using the basis for S_k given in (7), we consider the following products of three arbitrary basis elements:

$$\begin{aligned}
&(x_1^i + yx_2^i)(x_1^j + yx_2^j)(x_1^\ell + yx_2^\ell) \\
&= (x_1^{i+j} + yx_1^i x_2^j + yx_1^j x_2^i + y^2 x_2^{i+j})(x_1^\ell + yx_2^\ell) \\
&= x_1^{i+j+\ell} + yx_1^{j+\ell} x_2^i + yx_1^{i+\ell} x_2^j + yx_1^{i+j} x_2^\ell \\
&+ y^2 x_1^i x_2^{j+\ell} + y^2 x_1^j x_2^{i+\ell} + y^2 x_1^\ell x_2^{i+j} + y^3 x_2^{i+j+\ell}
\end{aligned} \tag{8}$$

with $0 \le i, j, \ell < k$. Since these polynomials generate the space $S_k^{(3)}$, we will use the correspondence of these polynomials to triples (i, j, ℓ) later.

In the following we show an upper bound for the dimension of $S_k^{(3)}$ that is equal to the actual dimension we observed in all of our experiments, for $S_k^{(3)}$ as well as for the cube code of the public code; see Remark 15. Since the proof is a bit technical we will illustrate parts of the proof by means of an example and we encourage the reader to jump to Example 13 when something is unclear.

Theorem 12. *The dimension of $S_k^{(3)}$ over $\mathbb{F}_q$ is upper bounded by $2k^2 - 4k + 4$ for $S_k = \{f(x_1) + yf(x_2) \in \mathbb{F}_q[x_1, x_2, y] : f \in \mathbb{F}_q[x]_{<k}\}$ and $k \ge 2$.*

Proof. We identify the generators from equation (8) of $S_k^{(3)}$ with triples of the form (i, j, ℓ) where $0 \le \ell \le j \le i < k$ and order them (in non-decreasing order) according to their value of $i + j + \ell$. Setting $s = i + j + \ell$ in (8), we obtain that the set of monomials appearing in any of the elements of $S_k^{(3)}$ is a subset of

$$\bigcup_{s=0}^{3k-3} \{x_1^s, yx_1^s, yx_1^{s-1}x_2, yx_1^{s-2}x_2^2, \ldots, yx_2^s, y^2x_1^s, y^2x_1^{s-1}x_2, \ldots, y^2x_2^s, y^3x_2^s\}.$$

The dimension of the space $S_k^{(3)}$ is equal to the rank of the matrix $\mathcal{R}$ whose columns correspond to these monomials and whose rows correspond to the generators from equation (8) of $S_k^{(3)}$, where the entry in row (i,j,ℓ) and column $y^{s_0}x_1^{s_1}x_2^{s_2}$ is equal to the coefficient of the monomial $y^{s_0}x_1^{s_1}x_2^{s_2}$ in the polynomial corresponding to (i,j,ℓ).

The matrix $\mathcal{R}$ has only entries in $\{0,1,2,3\}$. The rank of $\mathcal{R}$ over $\mathbb{F}_q$ is upper bounded by the rank of $\mathcal{R}$ considered as a matrix over $\mathbb{Z}$. Therefore, we consider it as a matrix over $\mathbb{Z}$.

Ordering rows and columns according to s, we have that $\mathcal{R}$ is a block diagonal matrix and hence, its rank is equal to the sum of the ranks of these $3k-2$ blocks. Note that, for all $s \in \{0,\ldots,3k-3\}$, the column of block s corresponding to x_1^s is the all-one column. The same is true for the column corresponding to $y^3x_2^s$, which we therefore can remove without changing the rank of $\mathcal{R}$. Also note that the column corresponding to $y^2x_1^{s_1}x_2^{s_2}$ is identical to the column corresponding to $yx_1^{s_2}x_2^{s_1}$ and hence, the rank of $\mathcal{R}$ does not change if we only consider the columns corresponding to

$$\bigcup_{s=0}^{3k-3} \{x_1^s, yx_1^s, yx_1^{s-1}x_2, yx_1^{s-2}x_2^2, \ldots, yx_2^s\}.$$

We first count the number of non-zero columns of each block and afterwards we will show that the rank for each block is upper bounded by its number of non-zero columns minus 2.

For $s \leq k-1$, the corresponding block has $s+2$ non-zero columns corresponding to the monomials $x_1^s, yx_1^s, yx_1^{s-1}x_2, \ldots, yx_2^s$.

For $k-1 < s \leq 2(k-1)$, the corresponding block has $k+1$ non-zero columns corresponding to the monomials $x_1^s, yx_1^s, yx_1^{s-1}x_2, \ldots, yx_1^{s-(k-1)}x_2^{k-1}$ since we deleted the other identical non-zero columns.

For $2(k-1) < s \leq 3(k-1)$, the block has $3k-3-s+2$ non-zero columns corresponding to the monomials $x_1^s, yx_1^{2(k-1)}x_2^{s-2(k-1)}, \ldots, yx_1^{s-(k-1)}x_2^{k-1}$ since we deleted the other identical non-zero columns.

Note that the first column of each block is equal to the all-one column and the sum of the entries in each row of each block is equal to 4 since we get 8 terms when multiplying three brackets with two terms each in equation (8) and we already showed that we can delete half of the columns because each column appears twice. Then, multiplying the first column with 3 and subtracting all the other columns in each block gives the all-zero column. This implies that the columns of each block are linearly dependent and any column can be deleted without changing the rank.

For $s = 0$ and $s = 3k-3$, the block consists only of one row (and two non-zero columns), corresponding to the polynomial where (i,j,ℓ) is equal to $(0,0,0)$ and $(k-1,k-1,k-1)$, respectively, and has rank one.

In the following, we assume $0 < s < 3k-3$, which also implies that we have at least 3 non-zero columns per block, and show that in each of these blocks we can eliminate also a second column without changing the rank of the block.

Let $s = 3z+d$ with $d \in \{0,1,2\}$. The polynomials of the block correspond to $(i,j,\ell) = (z+r_1, z+r_2, z+r_3)$ with $k-z > r_1 \geq r_2 \geq r_3 \geq -z \in \mathbb{Z}$, such that $r_1+r_2+r_3 = d$. We choose to eliminate the column corresponding to the monomial $yx_1^{s-z}x_2^z$ at first (as described above).

We remark here that from equation (8) we get that the coefficient of $yx_1^{i+j}x_2^\ell$ is equal to the maximum number of i,j,ℓ that coincide, i.e., it is 1 if $\ell < j < i$, it is 2 if $\ell = j, j < i$ or $i = j, \ell < j$ and it is 3 if $i = j = \ell$. Equivalently, this can be expressed in terms of r_1, r_2, r_3, where we have a 1 if $r_3 < r_2 < r_1$, a 2 if $r_3 = r_2, r_2 < r_1$ or $r_2 = r_1, r_3 < r_2$ and a 3 if $r_3 = r_2 = r_1$.

Now, we multiply the column corresponding to $yx_1^{s-z-r}x_2^{z+r}$ with r and sum up the results. We claim that for the row $(i,j,\ell) = (z+r_1, z+r_2, z+r_3)$ this sum is equal to $r_1+r_2+r_3$. This can be seen as follows. If $r_3 < r_2 < r_1$, then the only non-zero entries in this row are in the columns corresponding to the monomials $x_1^s, yx_1^{s-z-r_1}x_2^{z+r_1}, yx_1^{s-z-r_2}x_2^{z+r_2}, yx_1^{s-z-r_3}x_2^{z+r_3}$ and these non-zero entries are all equal to 1. Multiplying these entries equal to 1 with the values described above, i.e., with r_1, r_2, r_3, respectively, gives $r_1+r_2+r_3$ (note that we do not use the column for x_1^s). The other cases for r_1, r_2, r_3 are dealt with similarly. In this way we obtain a column all of whose entries are equal to $r_1+r_2+r_3 = d$. Therefore, the result is d-times the first column (i.e., the column corresponding to x_1^s). Hence, due to this dependency, we can eliminate a second column in each block.

Therefore, the overall rank of $S_k^{(3)}$ is upper bounded by

$$\begin{aligned}&1+\sum_{s=1}^{k-1} s + (k-1)^2 + \sum_{s=2k-1}^{3k-4} (3k-3-s) + 1 = 2 + (k-1)k + 2\sum_{s=1}^{k-2} s \\ &= 2k^2 - 4k + 4,\end{aligned}$$

as we claimed. □

Example 13. Let $k = 6$, i.e., $s \in \{0, \ldots, 15\}$. We consider first what happens for $s = 2$. The polynomials to be considered here are the ones corresponding to $(1,1,0)$ and $(2,0,0)$. The monomials appearing in these polynomials, i.e., the monomials corresponding to non-zero columns in the block for $s = 2$, are $x_1^2, yx_1^2, yx_1x_2, yx_2^2, y^2x_1^2, y^2x_1x_2, y^2x_2^2, y^3x_2^2$ and the corresponding block/submatrix inside the matrix $\mathcal{R}$ is

	x_1^2	yx_1^2	yx_1x_2	yx_2^2	$y^2x_1^2$	$y^2x_1x_2$	$y^2x_2^2$	$y^3x_2^2$
$(1,1,0)$	1	1	2	0	0	2	1	1
$(2,0,0)$	1	2	0	1	1	0	2	1

After eliminating the second half of columns (since they are all identical to one column from the first half) and observing that $z = 0$ in this case, we obtain

	x_1^2	$yx_1^2x_2^z$	$yx_1x_2^{z+1}$	yx_2^{z+2}
$(1,1,0)$	1	1	2	0
$(2,0,0)$	1	2	0	1

Multiplying the first column by 3 and subtracting the other columns lets us eliminate the second column. Moreover, we observe that the third column times 1 plus the fourth column times 2 gives two times the first column. Therefore, the rank of this block is two, which in this very simple case of course could have been seen much easier considering rows instead of columns, but we use it to illustrate the general case, in which column operations are easier since the same kind of column operations works for each block.

Consider now in addition the case $s = 7$. The corresponding block is

	x_1^7	yx_1^7	$yx_1^6x_2$	$yx_1^5x_2^2$	$yx_1^4x_2^3$	$yx_1^3x_2^4$	$yx_1^2x_2^5$	$yx_1x_2^6$	yx_2^7
$(3,2,2)$	1	0	0	2	1	0	0	0	0
$(3,3,1)$	1	0	1	0	2	0	0	0	0
$(4,2,1)$	1	0	1	1	0	1	0	0	0
$(4,3,0)$	1	1	0	0	1	1	0	0	0
$(5,1,1)$	1	0	2	0	0	0	1	0	0
$(5,2,0)$	1	1	0	1	0	0	1	0	0
	x_1^7	$yx_1^7x_2^{z-2}$	$yx_1^6x_2^{z-1}$	$yx_1^5x_2^z$	$yx_1^4x_2^{z+1}$	$yx_1^3x_2^{z+2}$	$yx_1^2x_2^{z+3}$	$yx_1x_2^{z+4}$	yx_2^{z+5}

where we already deleted the second half of (identical) columns. Since $s > k-1$ in this case, the last two columns are zero and can be removed. Eliminating afterwards two more columns with the column operations as described above with $d = r_1 + r_2 + r_3 = 1$, gives that the rank of this block is equal to 5. Hence, for this block the rank is smaller than its number of rows. The existence of blocks with this property is the reason why the dimension of the cube code here is smaller than what is conjectured for the random case.

Corollary 14. *The dimension of the cube code of the public code* $\mathrm{GRS}_k(P,\mu)M$ *is upper bounded by* $\min\{n, 2k^2-4k+4\}$.

Remark 15. We verified with the computer that for $2 \leq k < 60$ the dimension of $S_k^{(3)}$ is actually equal to $2k^2-4k+4$ over the fields $\mathbb{Q}, \mathbb{F}_2, \mathbb{F}_3, \mathbb{F}_4, \mathbb{F}_5, \mathbb{F}_7, \mathbb{F}_9, \mathbb{F}_{81}, \mathbb{F}_{97}, \mathbb{F}_{257}, \mathbb{F}_{625}, \mathbb{F}_{10007}$. We did not find a field where this was not the case. Moreover, in all our experiments for the attack and in additional experiments for $k < 60$, the actual dimension of the cube code of the public code was equal to $\min\{n, 2k^2-4k+4\}$.

If instead of this best possible upper bound from Corollary 14 one is satisfied with a quadratic upper bound with worse constants, there are considerably shorter ways to prove such a bound without using Theorem 12. For example one can use that the public code is a subcode of the sum $C_1 + C_2$ of two GRS codes C_1, C_2 of dimension k each, as well as Lemma 7 in the form

$$(C_1+C_2)^{(3)} \subseteq C_1^{(3)} + C_1^{(2)} * C_2 + C_1 * C_2^{(2)} + C_2^{(3)},$$

where the dimension of the latter is upper bounded by $(3k-2)+(2k-1)k+k(2k-1)+(3k-2) = 4k^2+4k-4$. We can also use this to get bounds on power codes of higher weight masked GRS codes.

Lemma 16. *Let* M *be an* $n \times n$ *matrix whose column weight is bounded from above by* w. *Then*

$$\dim((\mathrm{GRS}_k(P,\mu)M)^{w+1}) = O(k^w).$$

Proof. By definition of M we have that $\mathrm{GRS}_k(P,\mu)M$ is contained in a sum of w many GRS codes (here and only here we also allow column multipliers to be zero since this does not change the bounds on the dimension of powers of GRS codes). Let us denote them by $\mathrm{GRS}_k(Q_1,\mu_1), \ldots, \mathrm{GRS}_k(Q_w,\mu_w)$. Then, Lemma 7 gives

$$\left(\sum_{i=1}^{w} \mathrm{GRS}_k(Q_i,\mu_i)\right)^{(w+1)}$$
$$\subseteq \sum_{i_1,\ldots,i_{w+1}\in\{1,\ldots,w\}} \mathrm{GRS}_k(Q_{i_1},\mu_{i_1}) * \ldots * \mathrm{GRS}_k(Q_{i_{w+1}},\mu_{i_{w+1}}).$$

Note that for each summand at least two of $i_1, \ldots, i_{w+1}$ are the same. Using in addition

$$\dim \mathrm{GRS}_k(Q_{i_j}, \mu_{i_j})^{(2)} \leq 2k - 1 = O(k),$$

the result follows. □

4.1 The Effect of Shortening

In this section we discuss the effect of shortening and how to use it to extend the range of the distinguisher.

Recall from equation (1) that for a code C and a subset $J \subseteq \{1, \ldots, n\}$ we consider the code

$$C(J) = \{c \in C : c_i = 0 \text{ for all } i \in J\}.$$

In addition recall that $\mathrm{Sh}_J(C) = \mathrm{Pct}_J(C(J))$. We assume that the codes never degenerate and that the code $C(J)$ has dimension $k - |J|$ where k is the dimension of C. Moreover, for ease of notation we assume using equations (3) and (4) that M is equal to $\hat{M}$ of the form

$$\hat{M} = \mathrm{diag}(\alpha) + \Pi \cdot \mathrm{diag}(\beta).$$

First consider a single position at which the code is shortened. For simplicity assume we shorten the code $\mathrm{GRS}_k(P, \mu)M$ in the first position. Recall from equation (5) that we can go to an equivalent code with codewords c such that $c_1 = f(P_1) + \nu_1 f(P_{\Pi(1)})$.

Note that if $f(P_1) = 0 = f(P_{\Pi(1)})$, then also $c_1 = 0$. Therefore, the $(k-1)$-dimensional code $(\mathrm{GRS}_k(P, \mu)M)(\{1\})$ contains the code $\mathrm{GRS}_k(P, \mu)(\{1, \Pi(1)\})M$ as a $(k-2)$-dimensional subcode.

More generally, let J be the set of positions in which we shorten the code $\mathrm{GRS}_k(P, \mu)M$, i.e., we consider the code $(\mathrm{GRS}_k(P, \mu)M)(J)$. Then, we have

$$\mathrm{GRS}_k(P, \mu)(\{J \cup \Pi(J)\})M \subseteq (\mathrm{GRS}_k(P, \mu)M)(J). \tag{9}$$

Note that the right hand side has dimension $k - |J|$ and the left hand side has dimension $k - |J \cup \Pi(J)|$, for which we know

$$|J| \leq |J \cup \Pi(J)| \leq 2|J|.$$

For illustration we start with a special case. If $|J| = |J \cup \Pi(J)|$, then the two codes in equation (9) are the same, i.e.,

$$\mathrm{GRS}_k(P, \mu)(J)M = (\mathrm{GRS}_k(P, \mu)M)(J).$$

Lemma 17. *If $J = \Pi(J)$, then*

$$\mathrm{Sh}_J(\mathrm{GRS}_k(P, \mu)M) = \mathrm{Sh}_J(\mathrm{GRS}_k(P, \mu))M_J$$

where M_J is a matrix with row and column weight two that is obtained from M by deleting all rows and columns with index in J.

Proof. Without loss of generality and for ease of notation assume that $J = \{1, \ldots, |J|\}$. Since $J = \Pi(J)$, the weight two mask M is of the form $\begin{pmatrix} * & 0_{|J| \times (n-|J|)} \\ 0_{(n-|J|) \times |J|} & M_J \end{pmatrix}$. Denote by G a

generator matrix of $\mathrm{GRS}_k(P,\mu)(J)$ and by G_J the generator matrix of $\mathrm{Sh}_J(\mathrm{GRS}_k(P,\mu))$ which satisfies

$$G = \begin{pmatrix} 0_{(k-|J|)\times|J|} & G_J \end{pmatrix}.$$

We obtain,

$$\begin{aligned} GM &= \begin{pmatrix} 0_{(k-|J|)\times|J|} & G_J \end{pmatrix} \cdot \begin{pmatrix} * & 0_{|J|\times(n-|J|)} \\ 0_{(n-|J|)\times|J|} & M_J \end{pmatrix} \\ &= \begin{pmatrix} 0_{(k-|J|)\times|J|} & G_J M_J \end{pmatrix}. \end{aligned}$$

Since $\mathrm{Sh}_J(\mathrm{GRS}_k(P,\mu)M)$ is obtained by puncturing $GRS_k(P,\mu)(J)M$ in the positions in J, the result follows. □

Recall the following well-known lemma.

Lemma 18. *The code*

$$\mathrm{Sh}_J(\mathrm{GRS}_k(P,\mu))$$

is a GRS code of length $n-|J|$ and dimension $k-|J|$.

Thus, in the setting of Lemma 17, the shortening in positions J gives another weight 2 masked code.

In general, when $|J \cup \Pi(J)| = |J| + c$ with $0 \leq c \leq |J|$ we have that the codimension in (9) is c. This gives at least an upper bound for the dimension of the cube code of $(\mathrm{GRS}_k(P,\mu)M)(J)$. Namely, let

$$(\mathrm{GRS}_k(P,\mu)M)(J) = \mathrm{GRS}_k(P,\mu)(\{J \cup \Pi(J)\})M + \mathcal{E} \tag{10}$$

where $\mathcal{E}$ is a c-dimensional complement. Then, we have

$$\begin{aligned} &((\mathrm{GRS}_k(P,\mu)M)(J))^{(3)} \subseteq \\ &(\mathrm{GRS}_k(P,\mu)(\{J \cup \Pi(J)\})M)^{(3)} + (\mathrm{GRS}_k(P,\mu)(\{J \cup \Pi(J)\})M)^{(2)} * \mathcal{E} \\ &+ (\mathrm{GRS}_k(P,\mu)(\{J \cup \Pi(J)\})M) * \mathcal{E}^{(2)} + \mathcal{E}^{(3)}. \end{aligned} \tag{11}$$

Remark 19. In our experiments we observed that the first two summands have non-trivial intersection. This means that also the bounds we derive below are not tight. Moreover, the bound we derive for $(\mathrm{GRS}_k(P,\mu)(\{J \cup \Pi(J)\})M)^{(3)}$ to get the overall bound is also not tight. In our experiments we observed that one can shorten in way more positions than the bound below suggests.

In the experiments it even seems as if asymptotically the quotient between the number of positions that we can shorten in and n goes to 1. However this does not mean that the distinguisher works for all rates (not even for constant rates) because the growth in the number of positions in which we can shorten is too slow for that.

Lemma 20. *We have for the cube code of $(\mathrm{GRS}_k(P,\mu)M)(J)$ the following bound:*

$$\begin{aligned} &\dim((\mathrm{GRS}_k(P,\mu)M)(J))^{(3)} \\ &\leq 2k^2 - 4k + 4 - (4k(|J|+c) - 2(|J|+c)^2 - 4(|J|+c)) + \binom{k+1}{2}c + k\binom{c+1}{2} + \binom{c+2}{3}. \end{aligned}$$

Proof. By Lemma 18 we know that $\mathrm{GRS}_k(P,\mu)(\{J\cup\Pi(J)\})$ is a GRS code of dimension $k-|J|-c$ and length $n-|J|-c$ with zero columns appended. In particular, the code $\mathrm{GRS}_k(P,\mu)(\{J\cup\Pi(J)\})M$ is equivalent to an evaluation code of $S^{(3)}_{k-|J|-c}$ with zero columns appended and thus, we can bound the dimension using Theorem 12 as follows:

$$\begin{aligned}&\dim(\mathrm{GRS}_k(P,\mu)(\{J\cup\Pi(J)\})M)^{(3)} \leq 2(k-|J|-c)^2-4(k-|J|-c)+4\\ &=2(k^2+(|J|+c)^2-2k(|J|+c))-4k+4(|J|+c)+4\\ &=2k^2-4k+4-(4k(|J|+c)-2(|J|+c)^2-4(|J|+c)).\end{aligned}$$

Putting everything together we get

$$\begin{aligned}&\dim((\mathrm{GRS}_k(P,\mu)M)(J))^{(3)}\\ &\leq 2k^2-4k+4-(4k(|J|+c)-2(|J|+c)^2-4(|J|+c))+\binom{k+1}{2}\cdot c+k\cdot\binom{c+1}{2}+\binom{c+2}{3},\end{aligned}$$

as claimed. □

If this expression is smaller than $\min\left\{n-|J|,\binom{k-|J|+2}{3}\right\}$, we can distinguish using the cube code. Note that in the special case $c=0$, we get just the usual bound $2(k-|J|)^2-4(k-|J|)+4$ that we would also get from Lemma 17 together with Theorem 12.

5 From Distinguishing to Key Recovery

5.1 Basic Idea

Assume that we consider a certain class of $[n,k]_q$ codes that we can distinguish from random codes with the help of a certain distinguisher. We assume that also monomial transformations of such codes are distinguishable. Let C be such a code and let $M\in\mathbb{F}_q^{n\times n}$ be an invertible matrix with average column or row weight $\delta\geq 1$. Now, if we start with $\delta=1$ and increase it, we expect the code CM to get closer to random with respect to our distinguisher and our distinguisher to degrade. Ideally, the probability of distinguishing does not change until it falls abruptly to $\frac{1}{2}$, but the information our distinguisher gives us comes closer to the random value.

Let us give an example. Assume we can distinguish using the square code. Ideally, the dimension of the square code increases when we increase δ, but the probability that CM is distinguishable does not substantially fall until we hit the point where the dimension of the square code of CM is the dimension of the square code of a random code. At this point it falls abruptly to $\frac{1}{2}$.

In our setting we actually start with a code CM with M having row weight $\delta>1$. If we have a corresponding distinguisher, we can try to use it to recover M by considering CMN for some very sparse N. Depending on the form of N we might or might not have cancellation in the product MN, i.e., the number of non-zero entries in MN might or might not be smaller than or equal to the number of non-zero entries in M. Then, we use our distinguisher to distinguish between the cases, i.e., different matrices N, with or without cancellation and in that way we get information about M.

In the next subsection, we give concrete choices for the matrix N as above for the weight two masked McEliece cryptosystem with GRS codes. Afterwards we use the information obtained

from the distinguisher to recover a matrix that is in some sense equivalent to M, from which we can then recover the whole secret key. The actual attack will only work for $n > 2k^2 - 4k + 4$.

The basic idea of using cancellations when doing column operations for key recovery was already observed in [7, Proposition 5] for $\delta < 1 + R$ (where R is the rate) using the square code.

However, there the authors use in addition puncturing to detect whether a column has weight 1 or 2. After they have detected this, they turn columns of weight 2 into columns of weight 1 using cancellations in the columns. Note that puncturing is a tool that does not seem to help in our situation since all our columns have weight 2.

5.2 Finding Cancellations

As described in Section 3, we are given a public code $\mathrm{GRS}_k(P, \mu)M$ where M is an invertible matrix with row and column weight 2. We multiply on the right by a matrix of the form $I_n + \gamma E_{i,j}$ for $\gamma \in \mathbb{F}_q^*, 1 \leq i \neq j \leq n$, where $E_{i,j} \in \mathbb{F}_q^{n \times n}$ is the matrix which has a one in position i, j and zeros elsewhere. We get the code

$$\mathrm{GRS}_k(P, \mu)M(I_n + \gamma E_{i,j}).$$

We consider now the matrix

$$M(I_n + \gamma E_{i,j}).$$

This matrix has the same columns as M except for the j-th column. If we let M_i denote the i-th column of M, then the j-th column of $M(I_n + \gamma E_{i,j})$ is given by

$$M_j + \gamma M_i.$$

Now there are several possibilities of what can happen:

- If M_i and M_j have no common non-zero position, then $M_j + \gamma M_i$ has weight 4.
- If M_i and M_j have one common non-zero position, then generically $M_j + \gamma M_i$ has weight 3, but for a unique value of γ, it has weight 2.
- If both non-zero positions in M_i and M_j coincide, then $M_j + \gamma M_i$ has generically weight 2 and it has weight 1 for two values of γ.

If we can somehow distinguish between several of these cases using the code $\mathrm{GRS}_k(P, \mu)M(I_n + \gamma E_{i,j})$, we can recover certain information about the matrix M.

We saw that the j-th column of $M(I_n + \gamma E_{i,j})$ can have weight $1, 2, 3, 4$. The ideal situation would be if our distinguisher would allow us to distinguish between all these 4 cases.

Remark 21. In our experiments using the cube code as a distinguisher this was not the case. However, in the regime $n > 2k^2 - 4k + 4$, it allowed us to distinguish between the two cases of at most 2 and weight greater than 2. The experiments gave us

$$\begin{aligned} &\dim(\mathrm{GRS}_k(P, \mu)M(I_n + \gamma E_{i,j}))^{(3)} \\ &= \begin{cases} 2k^2 - 4k + 4 & \text{if } M_j + \gamma M_i \text{ has weight at most 2,} \\ 2k^2 - 4k + 5 & \text{if } M_j + \gamma M_i \text{ has weight greater than 2.} \end{cases} \end{aligned} \tag{12}$$

We will make the assumption that these are the cases we can distinguish from now on.

Therefore, we can see three different cases with our distinguisher by going through all values of $\gamma \in \mathbb{F}_q^*$:

1. There is no row in which the entries of the columns M_i and M_j are both non-zero.
2. There is exactly one row where the columns M_i and M_j both have a non-zero entry and we know the $\gamma \in \mathbb{F}_q^*$ such that $M_j + \gamma M_i$ is zero in that row.
3. The columns M_i and M_j have the non-zero entries in the same two rows.

We detect Case 1. when the distinguisher gives us that for all values of $\gamma \in \mathbb{F}_q^*$ the weight of column $M_j + \gamma M_i$ is at least 3. We detect Case 2. if the distinguisher tells us that for $q-2$ values of γ the column $M_j + \gamma M_i$ has weight at least 3 and for a unique value of γ the column $M_j + \gamma M_i$ has weight at most 2. This value is the γ mentioned in Case 2. Finally, we see that we are in Case 3. if for all values of γ the column $M_j + \gamma M_i$ has weight at most 2.

5.3 From Cancellation to Key Recovery

In this section, we will show how to use the previous distinction to recover a weight two mask $\hat{M}$ and a GRS code $\hat{C}$ such that $\hat{C}\hat{M} = GRS_k(P, \mu)M$.

Lemma 22. *Let M be a weight two matrix, i.e., a matrix where all rows and columns are of weight 2, and let $\mathcal{G}$ be the bipartite graph with adjacency matrix M. Then, $\mathcal{G}$ can be decomposed into cycles.*

Proof. Choose an arbitrary vertex v_1 in $\mathcal{G}$ and one of the two edges incident with v_1 in $\mathcal{G}$, denoted by e, and walk on a maximal path in $\mathcal{G}$ starting from v_1 along e such that all the vertices in the path are distinct. Since all vertices in $\mathcal{G}$ have degree 2 and the number of vertices is finite, such a path is unique. We denote this path by $[v_1, \ldots, v_\ell]$. Let v_ℓ be the last vertex on this path. Since the path cannot be extended, we know that the vertices adjacent to v_ℓ are $v_{\ell-1}$ and v_i for some $i \in [1, \ell-2]$. But if $i > 1$, then v_i would be adjacent to v_ℓ, v_{i-1}, v_{i+1}, a contradiction. Thus, v_ℓ is adjacent to v_1. Therefore, each vertex of $\mathcal{G}$ is contained in a unique cycle and $\mathcal{G}$ can be decomposed into cycles. □

Note that the cycle decomposition of the graph $\mathcal{G}$ as in the previous lemma corresponds to the cycle decomposition of the permutation Π in (2) and the submatrix of M corresponding to a cycle of length 2ℓ is up to row and column permutations equal to

$$\begin{pmatrix} a_1 & a_2 & 0 & \cdots & \cdots & 0 \\ 0 & a_3 & a_4 & 0 & \cdots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & a_{2\ell-3} & a_{2\ell-2} \\ a_{2\ell} & 0 & \cdots & \cdots & 0 & a_{2\ell-1} \end{pmatrix} \in \mathbb{F}_q^{\ell \times \ell} \tag{13}$$

with $a_i \in \mathbb{F}_q^*$ for $i \in \{1, \ldots, 2\ell\}$.

We will use the distinction into Cases 1., 2., 3. from the previous subsection to recover the cycle structure of the weight two mask. Note that with Case 2. and Case 3. of the previous

subsection we can actually find all the cycles. In Case 2. we even know that if we have an entry a in column M_j, then in column M_i in the same row there is the entry $-\frac{a}{\gamma}$. Since in Case 3., which corresponds to a 4-cycle in the corresponding graph, we cannot distinguish whether or not cancellations occur, i.e., we cannot distinguish whether $M_j + \gamma M_i$ has weight 1 or 2, we cannot recover the values of the non-zero entries corresponding to 4-cycles in this way. These values will be recovered in a later step, which we will explain in the next subsection.

First, we apply again equation (3), where we now choose the monomial matrix A such that $\hat{M}$ is a weight two mask that is ordered according to the cycles and is such that for each cycle the first non-zero entry in the first row of the cycle is a 1 and when moving along the cycle each second non-zero entry is a 1.

We illustrate this with an example.

Example 23. Consider, the weight two mask

$$\begin{pmatrix} a & 0 & 0 & 0 & 0 & b \\ 0 & 0 & 0 & c & d & 0 \\ e & 0 & f & 0 & 0 & 0 \\ 0 & 0 & g & 0 & 0 & h \\ 0 & i & 0 & j & 0 & 0 \\ 0 & k & 0 & 0 & l & 0 \end{pmatrix}.$$

Then, $\hat{M}$ is of the form

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & \frac{b}{a} \\ 0 & 0 & \frac{g}{h} & 0 & 0 & 1 \\ \frac{e}{f} & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & \frac{j}{i} & 0 & 0 \\ 0 & 0 & 0 & 1 & \frac{d}{c} & 0 \\ 0 & \frac{k}{l} & 0 & 0 & 1 & 0 \end{pmatrix}$$

where the first and last three rows each correspond to a 6-cycle. The values γ together with the columns i, j in Equation (12) that lead to cancellation are in this example

$$\left(-\frac{b}{a}, 1, 6\right), \left(-\frac{g}{h}, 6, 3\right), \left(-\frac{e}{f}, 3, 1\right), \left(-\frac{j}{i}, 2, 4\right), \left(-\frac{d}{c}, 4, 5\right), \left(-\frac{k}{l}, 5, 2\right).$$

So we can assume that we know half of the non-zero entries of $\hat{M}$ already because we fixed them to be 1. But Case 2. of the previous section gives us the quotient between two non-zero entries in the same row. In this case we also get the other value.

Therefore, we recovered all of the mask $\hat{M}$ except the non-zero values corresponding to 4-cycles. As mentioned above we will treat these values in the next subsection. The previous steps are summarized in Algorithm 1. In the case that no 4-cycles appear the attack is essentially finished.

Algorithm 1 FindCancellations($C_{pub}, \mathcal{D}_{cube}$)

Require: Public $[n,k]_q$ code C_{pub}, cube code distinguisher $\mathcal{D}_{cube}$
Ensure: Set of disjoint cycles W, partial multiplier matrix $\Gamma \in \mathbb{F}_q^{n\times n}$

$W \leftarrow \emptyset$, $\Gamma \leftarrow 0_{n\times n}$, Visited $\leftarrow \{\text{False}\}_{i=0}^{n-1}$
for $start \leftarrow 0$ to $n-1$ **do**
    if Visited[$start$] **then continue**
    end if
    $w \leftarrow (start)$, Visited[$start$] $\leftarrow$ True
    while true do
        $curr \leftarrow$ last(w), $prev \leftarrow$ second-to-last(w) (if $|w| \geq 2$ else null)
        for $j \leftarrow 0$ to $n-1$ **do**
            if $j == curr$ **or** $j == prev$ **or** (Visited[j] **and** $j \neq start$) **then**
                continue ▷ Skip self, immediate backtrack, and visited columns
            end if
            $\Gamma_{valid} \leftarrow \{\gamma \in \mathbb{F}_q^* \mid \mathcal{D}_{cube}(C_{pub} \cdot (I_n + \gamma E_{curr,j})) = \text{True}\}$
            if $|\Gamma_{valid}| == q-1$ **then** ▷ All γ cancel, implying a 4-cycle
                $w \leftarrow w \parallel (j)$, Visited[$j$] $\leftarrow$ True, $W \leftarrow W \cup \{w\}$
                break while
            else if $|\Gamma_{valid}| == 1$ **then**
                $\Gamma_{curr,j} \leftarrow$ sole element of Γ_{valid}
                if $j == start$ **then**
                    $W \leftarrow W \cup \{w\}$
                    break while
                else
                    $w \leftarrow w \parallel (j)$, Visited[$j$] $\leftarrow$ True
                    break for
                end if
            end if
        end for
    end while
end for
return (W, Γ)

5.4 Treating Four Cycles in the Weight Two Mask

In this subsection we detail how to treat 4-cycles. First we give a very simple attack using known techniques. Then we detail a heuristic approach which is more in spirit with the rest of the paper. Finally, we give a partial treatment of the probability of 4-cycles.

5.4.1 Basic Idea and Simple Attack on the 4-cycles

We saw in Lemma 22 that a weight two mask can be decomposed into cycles and we already assumed that the rows of $\hat{M}$ are ordered according to these cycles using (3). To also order the columns of $\hat{M}$ according to its cycle decomposition with a block diagonal structure, we multiply the public code on the right with a permutation matrix $\hat{\Pi}$, such that we end up with an equivalent

code of the form

$$\mathrm{GRS}_k(P,\mu)M\hat{\Pi} = \hat{C}\hat{M}\hat{\Pi}.$$

This means that $\hat{M}\hat{\Pi}$ is in block diagonal form where each cycle forms a block. Let

$$\hat{M}\hat{\Pi} = \begin{pmatrix} \tilde{M} & 0 & 0 & \cdots & 0 \\ 0 & \tilde{M}_1 & 0 & \cdots & 0 \\ 0 & 0 & \tilde{M}_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \tilde{M}_r \end{pmatrix}$$

with $\tilde{M}$ consisting of all cycles of length larger than 4 and the 4-cycles $\tilde{M}_1, \ldots, \tilde{M}_r$. Note that we have already found $\tilde{M}$. Then, multiplying the code $GRS_k(P,\mu)M\hat{\Pi}$ with

$$\begin{pmatrix} \tilde{M}^{-1} & 0 & 0 & \cdots & 0 \\ 0 & I_2 & 0 & \cdots & 0 \\ 0 & 0 & I_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & I_2 \end{pmatrix} \tag{14}$$

gives the code

$$\hat{C} \cdot \begin{pmatrix} I_{n-2r} & 0 & 0 & \cdots & 0 \\ 0 & \tilde{M}_1 & 0 & \cdots & 0 \\ 0 & 0 & \tilde{M}_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \tilde{M}_r \end{pmatrix} \tag{15}$$

with $\hat{C} = GRS_k(P,\mu) \cdot A^{-1}$ for some monomial matrix A. The inverse of the matrix in (14) is constructed in Algorithm 2.

Algorithm 2 PARTIALMASK($W_{>4}, W_4, \Gamma$)

Require: Set of cycles $W_{>4}$ ($|w| > 2$), Set of 4-cycles W_4 ($|w| == 2$), Multiplier matrix $\Gamma \in \mathbb{F}_q^{n \times n}$

Ensure: Partial block-diagonal mask $\hat{M}_{partial} \in \mathbb{F}_q^{n \times n}$ (the inverse of the matrix in (14)), Permutation matrix $\hat{\Pi} \in \mathbb{F}_q^{n \times n}$

$\Delta \leftarrow ()$
for $w \in W_{>4}$ **do**
    $\Delta \leftarrow \Delta \parallel w$
end for
for $w \in W_4$ **do**
    $\Delta \leftarrow \Delta \parallel w$
end for
$\hat{\Pi} \leftarrow 0_{n \times n}$
for $i \leftarrow 0$ to $n - 1$ **do**
    $\hat{\Pi}_{i,\Delta[i]} \leftarrow 1$
end for
$\hat{M}_{partial} \leftarrow I_n$
$idx \leftarrow 0$
for $w \in W_{>4}$ **do**
    $L \leftarrow |w|$
    for $i \leftarrow 0$ to $L - 1$ **do**
        $curr \leftarrow idx + i$
        $next \leftarrow idx + ((i + 1) \bmod L)$
        $\hat{M}_{partial}[curr, next] \leftarrow -\Gamma_{w[i],w[(i+1) \bmod L]}$
    end for
    $idx \leftarrow idx + L$
end for
return $(\hat{M}_{partial}, \hat{\Pi})$

At this point we have several ways to finish the attack (assuming not everything is a 4-cycle and we have enough non 4-cycles). One is to run the Sidelnikov-Shestakov or a filtration attack for all possible values of the 4-cycles. This gives a complexity of $(q-1)^2$ times the complexity of Sidelnikov-Shestakov for each 4-cycle. Another possibility would be to run the attack from [7, Subsection "Case of remaining degree-2 positions"] to recover $\tilde{M}_1, \ldots, \tilde{M}_r$ (and the GRS code). However, we did not implement this attack. With $\hat{\Pi}$ we can then recover all of the mask. In Section 5.4.2 we will also detail a third method given in Algorithm 3, which is more in the spirit of the rest of the paper. But first, let us give an overview of the whole attack.

In summary, we go the following route, where arrows denote which code we are able to obtain

from the preceding code:

$$\begin{array}{c} \mathrm{GRS}_k(P,\mu)M \\ \downarrow \\ \mathrm{GRS}_k(P,\mu)M\hat{\Pi} \\ \downarrow \\ \mathrm{GRS}_k(P,\mu)A^{-1} \begin{pmatrix} I_{n-2r} & 0 & 0 & \cdots & 0 \\ 0 & \tilde{M}_1 & 0 & \cdots & 0 \\ 0 & 0 & \tilde{M}_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \tilde{M}_r \end{pmatrix} \\ \downarrow \\ \mathrm{GRS}_k(P,\mu)A^{-1}. \end{array}$$

The permutation $\hat{\Pi}$ we find using Algorithms 1 and 2. For the second step, we also use our attack and for the last any of the three possibilities of Sidelnikov-Shestakov for each of the 4-cycles, the attack from [7] or the attack we detail next in Section 5.4.2.

Remark 24. So far our attack is only applicable in the regime where $n > 2k^2 - 4k + 4$. Using shortening, it seems reasonable to expect to be able to extend the attack to more parameters. This is, however, not straightforward. Remark 21 no longer holds true in this setting. One would need a more intricate handling of the situation, for example, doing several column operations at once instead of only one.

5.4.2 A New Heuristic Way of Treating 4-cycles

This more heuristic, but simpler, attack is as follows. We multiply each unknown 4-cycle $\left(\begin{smallmatrix} 1 & a \\ b & 1 \end{smallmatrix}\right)$ by a matrix of the form $\left(\begin{smallmatrix} 1 & \kappa \\ \lambda & 1 \end{smallmatrix}\right)$. We are interested in the case where the product has exactly 2 non-zero entries, and the second matrix $\left(\begin{smallmatrix} 1 & \kappa \\ \lambda & 1 \end{smallmatrix}\right)$ is invertible. The second matrix is invertible if and only if $\lambda \neq \frac{1}{\kappa}$. Note that we have

$$\begin{pmatrix} 1 & a \\ b & 1 \end{pmatrix} \begin{pmatrix} 1 & \kappa \\ \lambda & 1 \end{pmatrix} = \begin{pmatrix} 1+a\lambda & \kappa+a \\ b+\lambda & b\kappa+1 \end{pmatrix} =: \Lambda.$$

Now, we want to study how zero entries in the resulting matrix Λ can arise. Note that the product of matrices also has to be invertible. Assume that a diagonal entry of Λ is zero, without loss of generality,

$$1 + a\lambda = 0$$

or equivalently $\lambda = -\frac{1}{a}$. In this case, the off-diagonal entries have to be non-zero in order for the product Λ to be invertible. Therefore, if we want exactly 2 non-zero entries in Λ, we also need to have $b\kappa + 1 = 0$ and thus $\kappa = -\frac{1}{b}$. Now, assume an off-diagonal entry of Λ is zero. By invertibility, again, the diagonal entries are non-zero. To get exactly two non-zero entries in Λ we have $\kappa = -a$ and $\lambda = -b$.

This implies that if we can detect which values for κ and λ lead to the case of exactly 2 non-zero elements in Λ, we can recover a, b up to order.

In Algorithm 3, for each $i \in \{1, \ldots, r\}$, we multiply the code

$$C' := \mathrm{GRS}_k(P, \mu) A^{-1} \begin{pmatrix} I_{n-2r} & 0 & 0 & \cdots & 0 \\ 0 & \tilde{M}_1 & 0 & \cdots & 0 \\ 0 & 0 & \tilde{M}_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \tilde{M}_r \end{pmatrix}$$

with a block diagonal matrix of the form

$$\begin{pmatrix} I_{n-2(r-i+1)} & & 0 & & 0 \\ & 1 & \kappa_i & & \\ 0 & \lambda_i & 1 & & 0 \\ & & 0 & & \\ 0 & & & & I_{2(r-i)} \end{pmatrix}$$

to obtain the code C_i and apply the square-code as a distinguisher to C_i in order to find the values for κ_i and λ_i such that $\tilde{M}_i \cdot \begin{pmatrix} 1 & \kappa_i \\ \lambda_i & 1 \end{pmatrix}$ has exactly two non-zero entries.

Remark 25. The heuristic we used for the square code was the following:

$$\dim(C_i)^{(2)} = \begin{cases} \dim(C')^{(2)} - 1 & \text{if there are exactly two non-zero entries in } \tilde{M}_i \cdot \begin{pmatrix} 1 & \kappa_i \\ \lambda_i & 1 \end{pmatrix}, \\ \dim(C')^{(2)} & \text{otherwise.} \end{cases}$$

This always worked in all our experiments from Section 5.6.

Remark 26. To decrease the complexity of the square code distinguisher, one could, in addition, puncture C_i in positions that get multiplied with an identity matrix in equation (15) as long as the length of the resulting code is still large enough to be square code distinguishable.

Algorithm 3 CompleteMask($C_{base}, W_4, \hat{M}_{partial}, \mathcal{D}_{sq}$)

Require: Base $[n,k]_q$ code $C_{base} := C_{pub} \cdot \hat{\Pi} \cdot \hat{M}_{partial}^{-1}$, W_4, Partial mask $\hat{M}_{partial} \in \mathbb{F}_q^{n \times n}$, Square distinguisher $\mathcal{D}_{sq}$

Ensure: Complete mask $\hat{M}_{\hat{\Pi}} := \hat{M}\hat{\Pi} \in \mathbb{F}_q^{n \times n}$

$\hat{M}_{\hat{\Pi}} \leftarrow \hat{M}_{partial}$
$idx \leftarrow n - 2 \cdot |W_4|$
for $w \in W_4$ **do**
    $resolved \leftarrow$ False
    for $\kappa \in \mathbb{F}_q^*$ **do**
        for $\lambda \in \mathbb{F}_q^* \setminus \{\kappa^{-1}\}$ **do**
            $T \leftarrow I_n + \kappa E_{idx,idx+1} + \lambda E_{idx+1,idx}$
            if $\mathcal{D}_{sq}(C_{base} \cdot T) ==$ True **then**
                $\hat{M}_{\hat{\Pi}}[idx, idx+1] \leftarrow -\kappa$
                $\hat{M}_{\hat{\Pi}}[idx+1, idx] \leftarrow -\lambda$
                $resolved \leftarrow$ True
                break for
            end if
        end for
        if *resolved* **then break for**
        end if
    end for
    $idx \leftarrow idx + 2$
end for
return $\hat{M}_{\hat{\Pi}}$

Note that by reordering rows and multiplying with a non-zero scalar, it does not matter if we consider the mask with 4-cycle

$$\begin{pmatrix} 1 & a \\ b & 1 \end{pmatrix}$$

or the mask with 4-cycle

$$\begin{pmatrix} 1 & \frac{1}{b} \\ \frac{1}{a} & 1 \end{pmatrix}.$$

Therefore, both solutions that we can find for κ and λ work.

Using this, the overall algorithm is given in the following.

Algorithm 4 KeyRecovery$(C_{pub}, \mathcal{D}_{cube}, \mathcal{D}_{sq})$

Require: Public $[n,k]_q$ code C_{pub}, Distinguishers $\mathcal{D}_{cube}, \mathcal{D}_{sq}$
Ensure: Secret generator matrix $\hat{G}_{sec}$, Equivalent mask $\hat{M} \in \mathbb{F}_q^{n\times n}$

$(W, \Gamma) \leftarrow$ FindCancellations$(C_{pub}, \mathcal{D}_{cube})$
$W_{>4} \leftarrow \{w \in W \mid |w| > 2\}$
$W_4 \leftarrow \{w \in W \mid |w| == 2\}$
$(\hat{M}_{partial}, \hat{\Pi}) \leftarrow$ PartialMask$(W_{>4}, W_4, \Gamma)$
$C_{base} \leftarrow C_{pub} \cdot \hat{\Pi} \cdot \hat{M}_{partial}^{-1}$
$\hat{M}_{\hat{\Pi}} \leftarrow$ CompleteMask$(C_{base}, W_4, \hat{M}_{partial}, \mathcal{D}_{sq})$
$C_{GRS} \leftarrow C_{pub} \cdot \hat{\Pi} \cdot \hat{M}_{\hat{\Pi}}^{-1}$
$\hat{G}_{sec} \leftarrow$ SidelnikovShestakov(C_{GRS})
$\hat{M} \leftarrow \hat{M}_{\hat{\Pi}} \cdot \hat{\Pi}^T$
return $(\hat{G}_{sec}, \hat{M})$

5.4.3 Probabilities of 4-cycles

In our experiments we had some 4-cycles, but not many. A more thorough understanding of the probability of 4-cycles would be beneficial. A first step is given by the following lemma.

Lemma 27. *If the bipartite graph with adjacency matrix $M \in \mathbb{F}_q^{n\times n}$ can be decomposed into N cycles of lengths $2\ell_1, \dots 2\ell_N$ with $\ell_1 + \ldots + \ell_N = n$, then the probability that M is invertible is $\left(1 - \frac{1}{q-1}\right)^N$.*

Proof. Assume that M is in block diagonal form where the blocks correspond to the cycles of its cycle decomposition according to Lemma 22 and Equation (13). The determinant of the matrix in (13) representing a 2ℓ-cycle is equal to

$$\prod_{i=1,\ i \text{ odd}}^{2\ell} a_i + (-1)^{\ell+1} \prod_{i=1,\ i \text{ even}}^{2\ell} a_i \tag{16}$$

with $a_i \neq 0$ for $i = 1, \dots, 2\ell$. There are $(q-1)^{2\ell-1}$ possibilities for $a_1, \dots, a_{2\ell}$ such that (16) is equal to zero, since we can choose all a_i except one arbitrarily from $\mathbb{F}_q^*$ and then the last of the a_i is fixed. Hence, the probability that a matrix corresponding to a 2ℓ-cycle has full rank is equal to $1 - \frac{1}{q-1}$. Therefore, if the bipartite graph corresponding to M can be decomposed into N cycles, the probability that M has full rank is equal to $\left(1 - \frac{1}{q-1}\right)^N$. □

An application of Bayes' rule gives the probability of 4-cycles. However we have not found good bounds yet.

5.5 Complexity

All matrices in our algorithms have at most n rows and columns. Therefore, matrix multiplication and computing the rank can be done in $O(n^3)$.

The complexity of the attack in Algorithm 4 can be computed as follows. Algorithm 4 employs the other algorithms and some matrix multiplications. The main part of the complexity for

reasonable values of q comes from Algorithm 1. Here, we have two for-loops of length n. We also have a while-loop, but this only skips certain steps in the first for-loop. We also need to use $q-1$ many times the cube code distinguisher. For the cube code we need to compute the rank, what can be done in $O(n^3)$. Therefore, everything in Algorithm 1 can be done with $O(n^5 q)$ operations (technically over $\mathbb{F}_q$, so the bit complexity is slightly higher). The complexity of Algorithm 2 is $O(n)$. Assuming that the number of 4-cycles is constant, the complexity of Algorithm 3 is q^2 times the complexity of computing the rank, i.e., in total $O(q^2 n^3)$. The matrix multiplications are in $O(n^3)$ and the Sidelnikov-Shestakov attack is in $O(k^3 + k^2 n)$, which in our case is $O(n^2)$ and thus, the overall complexity of Algorithm 4 is $O(qn^5 + q^2 n^3)$.

This complexity is similar to the complexities of [6, 7].

It would be good to reduce the complexity in order to be able to run more experiments since in our setting qn^5 is already of the order of k^{12}.

5.6 Experiments

We ran the attack successfully on the following sets of parameters for (n, k, q) (10 experiments each).

$$(60, 6, 61), (60, 6, 64), (85, 7, 89), (90, 7, 121), (110, 8, 113), (112, 8, 125),$$
$$(150, 9, 151), (180, 10, 181), (220, 11, 223), (255, 12, 257), (300, 13, 307).$$

A toy implementation can be found here: `https://git.math.uzh.ch/abmazu/an-attack-on-high-rate-mceliece-cryptosystems-using-generalized-reed-solomon-codes-with-weight-2-mask`. For the larger parameter sets we parallelized the algorithm.

6 Outlook and Open Problems

We divide this section into heuristics which we have used, but not proven for our distinguisher and attack on the one hand and open questions about more general attacks on the other hand.

6.1 Unproven Assumptions

We give a list of the assumptions that underlie our attacks but are not proven. It might be worthwhile to try to prove all of them.

- The dimension of the cube code of a random code of dimension k and length n is with high probability equal to
$$\min\left\{\binom{k+2}{3}, n\right\}.$$
- The dimension of $S_k^{(3)}$ over $\mathbb{F}_q$ is equal to $2k^2 - 4k + 4$.
- The dimension of $(\mathrm{GRS}_k(P, \mu)M)^{(3)}$ is $2k^2 - 4k + 4$. In particular the evaluation of $S_k^{(3)}$ at $(P_1, P_{\Pi(1)}, \nu_1), \ldots, (P_n, P_{\Pi(n)}, \nu_n) \in \mathbb{F}_q^3$ is injective.
- Equation (12) holds.
- The probability of having many 4-cycles is small.
- The heuristic Algorithm 3 always works.

6.2 Other Distinguishers

It is worth emphasizing again that we give a general framework that turns a good enough distinguisher into a key-recovery attack and for high rate use the cube code as distinguisher. This immediately leads to the question if we can use other distinguishers here.

A first question is whether Betti numbers [12] can help to extend the attack to smaller rates. We guess that the answer will be a little, but not much. The reason is as follows. Let C be a code of regularity 2, i.e., with $C^{(2)} = \mathbb{F}_q^n$. Then in [12] it was shown that all Betti numbers except for $\beta_{1,3}$ are determined by $I_2(C)$, the kernel of the evaluation map of quadratic forms. So far, we have no evidence that there are special relations in $I_2(C)$ or that in this case the Betti numbers behave differently from random. In this work we have dealt with the case of codes of regularity ≥ 4, i.e., $C^{(3)} \neq \mathbb{F}_q^n$. We think that the distinguisher and attack can be extended to codes of regularity 3, i.e., codes with $C^{(2)} \neq \mathbb{F}_q^n$, but $C^{(3)} = \mathbb{F}_q^n$ using $\beta_{1,4}, \beta_{2,3}$.

Also other distinguishers such as the higher-order vanishing distinguisher [9] do not seem to be directly applicable.

6.3 Open Problems

Here we mention some open problems. The first questions concern the range of applicability of our attack.

1. The distinguisher can be improved using shortening. Is this also true for the attack? See Remark 24.

2. Explain Remark 19 and find the exact asymptotics of the number of positions in which we can shorten and the asymptotics of the distinguisher using shortening.

3. Even with shortening the distinguisher applies only to a limited range of parameters. Are there other ways to attack in the lower rate regime? For example, is it possible to find subcodes $\mathrm{GRS}_{k'}(P, \mu)M$ where $k' < k$? In this case we could use the high rate attack on $\mathrm{GRS}_{k'}(P, \mu)M$.

4. Related to the previous point: Can we use the cycle structure and Lemma 17 to extend the range of the attack? Assume we have shortened in a certain number of positions J and for simplicity assume that the set $J \cup \Pi(J)$ lies within a single cycle. Further assume that after shortening in J we are in the distinguishable regime. Then it seems highly likely that shortening in further positions we can detect using equations (10) and (11) whether the new positions are in $\Pi(J)$ or not. In this way one could obtain certain information about the cycle structure of the matrix M even if one would be far from recovering all of M.

5. Can we use other distinguishers as mentioned in the previous subsection?

6. In this work we only attacked [14] in the high rate regime. Is it possible to also attack schemes which have similar assumptions such as [1, 4, 8], in the high rate regime?

7. Can we also attack higher weight masked schemes with average weight $\delta > 2$? We know that in the high rate regime a sufficiently large power code will yield a distinguisher by Lemma 16 and if our heuristics about cancellation still hold, we can use it in the same way as before to recover information about the mask M. However, actual key recovery will be more complicated because the structure of M is less straightforward.

8. In the BBCRS scheme [3] the authors used as a mask the sum of a weight two matrix and a matrix of rank 1. Can we also break such a scheme in the high rate regime?

9. In [4] the authors also propose to use AG codes with a weight 2 mask. It seems very likely that the distinguisher and possibly also the attack will apply for general AG codes in the high rate regime.

While some of the open problems seem relatively straightforward and some very hard we encourage the community to investigate all of them to have a more thorough understanding of the security of these problems.

Acknowledgements

This work has been supported in part by the Swiss National Science Foundation under SNSF grant number 212865, and by the German research foundation, project number 513811367.

References

[1] Paulo Almeida, Miguel Beltrá, and Diego Napp. "A convolutional variant of the Niederreiter cryptosystem with GRS codes". In: *2024 IEEE International Symposium on Information Theory (ISIT)*. IEEE. 2024, pp. 1818–1823.

[2] Marco Baldi, Marco Bianchi, Franco Chiaraluce, Joachim Rosenthal, and Davide Schipani. "A variant of the McEliece cryptosystem with increased public key security". In: *WCC 2011-Workshop on coding and cryptography*. 2011, pp. 173–182.

[3] Marco Baldi, Marco Bianchi, Franco Chiaraluce, Joachim Rosenthal, and Davide Schipani. "Enhanced public key security for the McEliece cryptosystem". In: *Journal of Cryptology* 29.1 (2016), pp. 1–27.

[4] Jessalyn Bolkema, Heide Gluesing-Luerssen, Christine A Kelley, Kristin E Lauter, Beth Malmskog, and Joachim Rosenthal. "Variations of the McEliece cryptosystem". In: *Algebraic Geometry for Coding Theory and Cryptography: IPAM, Los Angeles, CA, February 2016*. Springer, 2017, pp. 129–150.

[5] Ignacio Cascudo, Ronald Cramer, Diego Mirandola, and Gilles Zémor. "Squares of random linear codes". In: *IEEE Transactions on Information Theory* 61.3 (2015), pp. 1159–1173.

[6] Alain Couvreur, Philippe Gaborit, Valérie Gauthier-Umaña, Ayoub Otmani, and Jean-Pierre Tillich. "Distinguisher-based attacks on public-key cryptosystems using Reed-Solomon codes". In: *Designs, Codes and Cryptography* 73.2 (2014), pp. 641–666.

[7] Alain Couvreur, Ayoub Otmani, Jean-Pierre Tillich, and Valérie Gauthier-Umaña. "A polynomial-time attack on the BBCRS scheme". In: *IACR International Workshop on Public Key Cryptography*. Springer. 2015, pp. 175–193.

[8] Victor Dyseryn, Danilo Francati, and Daniele Venturi. *Public-Key Pseudorandom Codes from Distorted McEliece Assumptions*. Cryptology ePrint Archive, Paper 2026/1184. 2026. URL: `https://eprint.iacr.org/2026/1184`.

[9] Tobias Hemmert and Andreas Wiemers. *Distinguishing Goppa codes using higher-order vanishing*. Cryptology ePrint Archive, Paper 2025/1661. 2025. URL: https://eprint.iacr.org/2025/1661.

[10] Harald Niederreiter. "Knapsack-type cryptosystems and algebraic coding theory". In: *Prob. Contr. Inform. Theory* 15.2 (1986), pp. 157–166.

[11] Ruud Pellikaan, Xin-Wen Wu, Stanislav Bulygin, and Relinde Jurrius. *Codes, cryptology and curves with computer algebra*. Vol. 1. Cambridge University Press, 2017.

[12] Hugues Randriambololona. "The syzygy Distinguisher". In: *Advances in Cryptology - EUROCRYPT 2025 - 44th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Madrid, Spain, May 4-8, 2025, Proceedings, Part VI*. Ed. by Serge Fehr and Pierre-Alain Fouque. Lecture Notes in Computer Science. Springer, 2025, pp. 324–354. DOI: 10.1007/978-3-031-91095-1_12. URL: https://doi.org/10.1007/978-3-031-91095-1_12.

[13] V. M. Sidelnikov and S. O. Shestakov. "On insecurity of cryptosystems based on generalized Reed-Solomon codes". In: *Discrete Mathematics and Applications* 2.4 (1992), pp. 439–444. DOI: doi:10.1515/dma.1992.2.4.439.

[14] Violetta Weger. "A code-based cryptosystem using GRS codes". Master Thesis. University of Zurich, 2016.