Square Code Attack on a Modified Sidelnikov Cryptosystem

Ayoub Otmani[1(✉)] and Hervé Talé Kalachi[2]

[1] LITIS, University of Rouen, 76821 Mont-Saint-Aignan, France
ayoub.otmani@univ-rouen.fr

[2] Department of Mathematics, University of Yaounde 1, ERAL, Cameroon
hervekalachi@gmail.com

Abstract. This paper presents a cryptanalysis of a modified version of the Sidelnikov cryptosystem which is based on Reed-Muller codes. This modified scheme consists in inserting random columns in the secret generating matrix or parity check matrix. The cryptanalysis relies on the computation of the squares of the public code. The particular nature of Reed-Muller which are defined by means of multivariate binary polynomials, permits to predicate the value of dimension of the square codes and then to fully recover in polynomial time the secret positions of the random columns. Our work shows that the insertion of random columns in the Sidelnikov scheme does not bring any security improvement.

Keywords: Sidelnikov scheme · Component-wise product · Cryptanalysis · Distinguisher

1 Introduction

Contrary to the cryptosystems based on number theory, the safety of cryptosystems based on error correcting codes appear to be resistant to the emergence of quantum computers [22]. Its other advantage is that the encryption and decryption are very fast, about five times faster for encryption, and 10 to 100 times faster for decryption compared to RSA cryptosystem. The most important representative of this cryptography is the McEliece cryptosystem [17] which is also one of the oldest public key cryptosystems. Its security is based on two problems: the difficulty of decoding a random linear code and the difficulty of recovering a decoding algorithm from a public matrix representation of a binary Goppa code. The second assumption was reformulated in a more formal way by stating there is no polynomial-time algorithm that distinguishes between a random matrix and a generating matrix of a binary Goppa code [4,21].

Although efficient, the main drawback of this scheme is the enormous size of the public key. During these last years, several authors have proposed to consider more structured codes. The common idea is to focus on codes equipped with a non-trivial permutation group.[1] This is the case for example of Misoczki and

[1] The permutation group of a code is the set of permutations leaving globally invariant the code.

S. El Hajji et al. (Eds.): C2SI 2015, LNCS 9084, pp. 173–183, 2015.
DOI: 10.1007/978-3-319-18681-8_14

Barreto [19] who proposed quasi-dyadic Goppa codes. Their worked followed Gaborit's idea to use quasi-cyclic BCH codes [11] and Berger, Cayrel, Gaborit and Otmani's paper [1] which used quasi-cyclic alternant codes. The algebraic attack given in [10] succeeds in breaking most of the parameters of [1,19]. It makes use of the fact that the underlying codes which are alternant codes come with an algebraic structure. It allows a cryptanalysis consisting in setting up a polynomial system and then solving it with Gröbner bases techniques. In the very specific the case of [1,19], the quasi-cyclic and quasi-dyadic structures allow a huge reduction of the number of variables. Recently, the attack was further improved for against [19] by exploiting more efficiently the underlying Goppa structure [8,9].

The apparition of algebraic attacks [10], although it does not undermine the security of the McEliece scheme, shows however the importance of finding a better hiding of the structure of the codes. A possible solution would be to change the description of the scheme by inserting some randomness. Probably, the first attempt towards this objective, is Berger-Loidreau's paper [2]. The authors suggest to add random rows to the description of the codes. They applied this to Niederreiter encryption scheme [20] instantiated with generalised Reed-Solomon codes. The goal is to come up with a protection against Sidelnikov and Shestakov [24]. But Wieshebrink's paper shows that component-wise product of codes [27] enables to break Berger-Loidreau's proposal.

Another simple example would be to insert random columns in the secret matrix. Several authors [25,14] have indeed proposed this technique to avoid structural attacks on similar versions of the McEliece cryptosystem. This kind modification was proposed for the first time by Wieschebrink in [25]. Its primary goal was to avoid the Sidelnikov-Shestakov attack [24] on the McEliece cryptosystem using generalized Reed-Solomon codes. Although this proposal had effectively avoided the original attack, recent studies have shown that in that case of generalized Reed-Solomon codes, the random columns can be found through considerations of the dimensions of component-wise product of codes [12,13,5]. This operation turns out to be a powerful tool. Thanks to [16], it has also helped in understanding the distinguisher of Goppa code derived in [6,7] which challenged the validity of the Goppa code indistinguishability assumption introduced in [4,21]. The paper [16] proves that the distinguisher in [6,7] has an equivalent but simpler description in terms of the component-wise product of codes. This distinguisher is even more powerful in the case of Reed-Solomon codes than for Goppa codes. Indeed, whereas for Goppa codes it is only successful for rates close to 1 [6,7], it can distinguish Reed-Solomon codes of any rate from random codes.

This paper develops a cryptanalysis of the modified version given in [14] of the Sidelnikov encryption scheme [23] which is a McEliece-type public key encryption scheme [17] based on Reed-Muller codes. The idea of [14] is to add random columns to prevent sub-exponential time key-recovery attacks of [18,3]. But, like Reed-Solomon codes, Reed-Muller codes are evaluation codes and because of this, they can be distinguished from random codes. These two families of codes share very similar properties which facilitates the recovering of the random columns. Our key-recovery attack is divided into two steps. The first one is an adaptation to Reed-Muller codes of the attacks presented in [12,13,5] in order to find the secret random columns. This is achieved in $O(n^5)$ operations in the binary field where n is the block length of the codes. The second step applies [18,3] to recover the secret permutation that hides the structure of the Reed-Muller codes. The rest of the paper is devoted to the description of the first step of the attack.

2 Preliminary Facts

We present in this section definitions and properties from coding theory we need in the paper.

Let $\mathbb{F}_q$ be the finite field of q elements, n and k be two non-zero integers such that $k \geqslant n$. A *linear code* of length n and dimension k over $\mathbb{F}_q$ is a linear subspace $\mathscr{C}$ of $\mathbb{F}_q^n$ of dimension k over $\mathbb{F}_q$. A *generating matrix* of $\mathscr{C}$ is a $k \times n$ matrix whose rows form a basis of $\mathscr{C}$. The *dual* of $\mathscr{C}$, generally denoted by $\mathscr{C}^\perp$, is the set of vectors $\boldsymbol{v} \in \mathbb{F}_q^n$ such that for all $\boldsymbol{c} \in \mathscr{C}$ the inner product $\boldsymbol{c} \cdot \boldsymbol{v} \overset{\text{def}}{=} \sum_i c_i v_i = 0$. A generating matrix for $\mathscr{C}^\perp$ is also called a *parity-check matrix.*

Definition 1 (Generalised Reed-Solomon). *Let $\boldsymbol{x} = (x_1, \ldots, x_n)$ where x_i are distinct elements of $\mathbb{F}_{q^m}^n$ and let $\boldsymbol{y}$ be the vector $(y_1, \ldots, y_n)$ where y_i are non-zero elements of $\mathbb{F}_{q^m}$. The generalised Reed-Solomon code (GRS) of length n and dimension k over $\mathbb{F}_{q^m}$ is given by:*

$$\mathbf{GRS}_k(\boldsymbol{x}, \boldsymbol{y}) \overset{def}{=} \left\{ \Big((y_1 f(x_1), \ldots, y_n f(x_n) \Big) \;:\; f \in \mathbb{F}_{q^m}[X],\ \deg(f) < k \right\}$$

Definition 2 (Component-Wise Product). *Given two vectors $\boldsymbol{a} = (a_1, \ldots, a_n)$ and $\boldsymbol{b} = (b_1, \ldots, b_n)$ in $\mathbb{F}^n$ where $\mathbb{F}$ is field, we denote by $\boldsymbol{a} \star \boldsymbol{b}$ the component-wise product:*

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

Definition 3 (Product of codes). *Let $\mathscr{A}$ and $\mathscr{B}$ be two linear 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 $a \star b$ where a and b range over A and B respectively.*

When $\mathscr{B} = \mathscr{A}$ then $\mathscr{A} \star \mathscr{A}$ is called the square code of $\mathscr{A}$ and is rather denoted by $\mathscr{A}^2$.

The importance of the square code construction becomes clear when we compare the dimensions of a code $\mathscr{A}$ with the dimension of its square code $\mathscr{A}^2$ and one major question is to know what one should expect. This comparison has already been made in [12,13,5] in the case of generalized Reed-Solomon codes which allowed to mount efficient attacks on several different schemes based on generalised Reed-Solomon codes [26,12,13,5]. The results of this paper are based on these comparisons in the case of Reed-Muller codes.

We recall here important facts about the product of codes.

Proposition 1. *For any linear subspaces $F \subseteq E$ and $G \subseteq E$:*

$$\dim F \star G \leqslant \dim F \dim G - \binom{\dim F \cap G}{2}. \tag{1}$$

Proof. Assume $d \overset{\text{def}}{=} \dim F \cap G$ and let $\mathcal{B} = \{b_1, \dots, b_d\}$ be a basis of $F \cap G$. We complete $\mathcal{B}$ with vectors $\mathcal{F} = \{f_1, \dots, f_t\}$ so that $\mathcal{B} \cup \mathcal{F}$ is a basis of F. We do the same for G by completing $\mathcal{B}$ with $\mathcal{G} = \{g_1, \dots, g_m\}$ so that $\mathcal{B} \cup \mathcal{G}$ is a basis of G. A generating set of $F \star G$ is the union of the four sets $\{b_i \star b_j : 1 \leqslant i \leqslant j \leqslant d\}$, $\{b_i \star f_j : 1 \leqslant i \leqslant d, 1 \leqslant j \leqslant t\}$, $\{b_i \star g_j : 1 \leqslant j \leqslant d, 1 \leqslant j \leqslant m\}$ and $\{f_i \star g_j : 1 \leqslant j \leqslant t, 1 \leqslant j \leqslant m\}$. The proof is terminated by observing the equality:

$$dt + dm + tm + \binom{d+1}{2} = (t+d)(d+m) - \frac{1}{2}d(d-1).$$

Corollary 1. *For any linear subspace $F \subseteq E$:*

$$\dim F \star E \leqslant \dim F \dim E - \binom{\dim F}{2}.$$

In particular $\dim E^2 \leqslant \binom{\dim E + 1}{2}$.

3 Code-Based Public-Key Encryption Schemes

3.1 McEliece Encryption Scheme

In this section we give the basic notion about the McEliece [17] and Niederreiter [20] cryptosystems . Let $\mathcal{G}$ be a family of (n,k)-linear codes over $\mathbb{F}_q$ for which a polynomial-time algorithm to decode t-error is available. The general version of the McEliece cryptosystem is described as follows but McEliece proposed to use binary Goppa codes.

Key Generation

1. Let $\boldsymbol{G}' \in \mathscr{M}_{k \times n}(\mathbb{F}_q)$, be a generating matrix of a t-error correcting code $\mathscr{C}' \in \mathcal{G}$
2. Pick an $n \times n$ permutation matrix $\boldsymbol{P}$ and a $k \times k$ invertible matrix $\boldsymbol{S}$ at random over $\mathbb{F}_q$.
3. Compute $\boldsymbol{G} = \boldsymbol{S}^{-1}\boldsymbol{G}'\boldsymbol{P}^{-1}$ which is another generating matrix.

The public key is $(\boldsymbol{G}, t)$ and the private key is $(\boldsymbol{S}, \boldsymbol{G}, \boldsymbol{P})$.

Encryption. To encrypt the message $\boldsymbol{m} \in \mathbb{F}_q^k$, one randomly generates $\boldsymbol{e} \in \mathbb{F}_q^n$ of Hamming weight $\leqslant t$. The ciphertext is then the vector $\boldsymbol{c} = \boldsymbol{m}\boldsymbol{G} + \boldsymbol{e}$.

Decryption. The vector $\boldsymbol{c}\boldsymbol{P}^{-1}$ is at a distance at most t of $\mathscr{C}$. The decoding algorithm thus allows to find the vector $\boldsymbol{y} \overset{\text{def}}{=} \boldsymbol{m}\boldsymbol{S}^{-1}$. The plaintext is deduced by computing $\boldsymbol{y}\boldsymbol{S}$.

3.2 Niederreiter Encryption Scheme

A version of the McEliece cryptosystem that uses the parity-check matrix instead of the generating matrix has been proposed by Niederreiter [20], and has been proved to be completely equivalent in term of security. The Niederreiter cryptosystem is generally describes as follows. In the following, the transpose of matrix is denoted by ^{T}.

Key Generation

1. Let $\boldsymbol{H}' \in \mathscr{M}_{(n-k)\times n}(\mathbb{F}_q)$, be a parity check matrix of a t-error correcting code $\mathscr{C}' \in \mathcal{G}$
2. Pick at random an $n \times n$ permutation matrix $\boldsymbol{P}$ and a $(n-k)\times(n-k)$ non singular matrix $\boldsymbol{S}$ over $\mathbb{F}_q$.
3. Compute $\boldsymbol{H} = \boldsymbol{S}^{-1}\boldsymbol{H}'\boldsymbol{P}^{-1}$.

The public key is $(\boldsymbol{H}, t)$ and the private key is $(\boldsymbol{S}, \boldsymbol{H}', \boldsymbol{P})$.

Encryption. For a message $\boldsymbol{m} \in \mathbb{F}_q^n$ of Hamming weight $\leqslant t$. The cipher text is given by $\boldsymbol{c} = \boldsymbol{H}\boldsymbol{m}^T$.

Decryption. Since $\boldsymbol{c} = \boldsymbol{S}^{-1}\boldsymbol{H}'\boldsymbol{P}^{-1}\boldsymbol{m}^T = \boldsymbol{S}^{-1}\boldsymbol{H}'\left(\boldsymbol{m}\boldsymbol{P}\right)^T$ and $\boldsymbol{m}\boldsymbol{P}$ is a word of weight less than or equal to t, the receiver decodes $\boldsymbol{S}\boldsymbol{c}$ to get the word $\boldsymbol{y}$. The associated plaintext is then $\boldsymbol{y}\boldsymbol{P}$.

4 Wieschebrink's Masking Technique

Here we present a masking technique first developed in [25] and then proposed several times with different families of codes. It consists in inserting random columns in the secret matrix. This technique can be used both in the McEliece cryptosystem and the Niederreiter version.

4.1 Modified McEliece Scheme

Key Generation

1. Choose three integers n_0, k, ℓ with $\ell \ll n$ and set $n \overset{\text{def}}{=} n_0 + \ell$. Pick a random a generating matrix $\boldsymbol{G}_0$ of an (n_0, k)-code $\mathscr{C}$ that is able to decode t errors.
2. Pick randomly a $k \times \ell$ matrix $\boldsymbol{R}$, a $k \times k$ invertible matrix $\boldsymbol{S}$ over $\mathbb{F}_q$ and an $n \times n$ permutation matrix $\boldsymbol{P}$.
3. Set $\boldsymbol{G}' = (\boldsymbol{G}_0 \mid \boldsymbol{R})$ and compute $\boldsymbol{G} = \boldsymbol{S}^{-1}\boldsymbol{G}'\boldsymbol{P}^{-1}$.

The public key is $(\boldsymbol{G}, t)$ and the private key is $(\boldsymbol{S}, \boldsymbol{P}, \boldsymbol{G}')$.

Encryption. To encrypt a plaintext $\boldsymbol{m} \in \mathbb{F}_q^k$, one randomly generates $\boldsymbol{e} \in \mathbb{F}_q^n$ of weight $\leqslant t$ and computes the ciphertext $\boldsymbol{c} = \boldsymbol{mG} + \boldsymbol{e}$.

Decryption. To decrypt $\boldsymbol{c}$, one computes $\boldsymbol{y} = \boldsymbol{cP}$ and let $\boldsymbol{y}'$ be the n_1 first columns of $\boldsymbol{y}$. The vector $\boldsymbol{y}'$ is located within distance t from $\mathscr{C}$. The decoding of $\boldsymbol{y}'$ provides the plaintext.

4.2 Modified Niederreiter Scheme

Here one can apply the same principle as in the case of McEliece cryptosystem, but here the insertion of random columns is done in the parity check matrix.

Key Generation

1. Choose three integers n_0, k, t, ℓ with $\ell \ll n$ and set $n \overset{\text{def}}{=} n_0 + \ell$. Pick a random parity-check matrix $\boldsymbol{H}_0$ of an (n_0, k)-code $\mathscr{C}$ that is able to decode t errors.
2. Pick randomly an $(n_0 - k) \times \ell$ matrix $\boldsymbol{R}$ and a $(n_0 - k) \times (n_0 - k)$ non singular matrix $\boldsymbol{S}$ over $\mathbb{F}_q$, and an $n \times n$ permutation matrix $\boldsymbol{P}$.
3. Set $\boldsymbol{H}' = (\boldsymbol{H}_0 \mid \boldsymbol{R})$ and compute $\boldsymbol{H} = \boldsymbol{S}^{-1}\boldsymbol{H}'\boldsymbol{P}^{-1}$

The public key is $(\boldsymbol{H}, t)$ and the private key is $(\boldsymbol{S}, \boldsymbol{H}', \boldsymbol{P})$.

Encryption. For a plaintext $\boldsymbol{m} \in \mathbb{F}_q^n$ of Hamming weight $\leqslant t$, the corresponding ciphertext is given by $\boldsymbol{c} = \boldsymbol{H}\boldsymbol{m}^T$.

Decryption. Let $\mathsf{dec}(\cdot)$ be the decoding algorithm of $\mathscr{C}$. The symbol $\perp$ stands for a decoding failure[2]. The decryption of a ciphertext $\boldsymbol{c}$ is described in Algorithm 1.

Algorithm 1. Decryption of Niederreiter scheme with Wieschebrink's masking.

```
$\boldsymbol{u} = \perp$
for all $\boldsymbol{z} \in \mathbb{F}_q^\ell$ do
  $\boldsymbol{y} = \mathsf{dec}\left(\boldsymbol{Sc} + \boldsymbol{Rz}^T\right)$
  if $\boldsymbol{y} \neq \perp$ then
    $\boldsymbol{u} = (\boldsymbol{y}, \boldsymbol{z})\boldsymbol{P}$
    return $\boldsymbol{u}$
  end if
end for
return $\boldsymbol{u}$
```

Note that it is possible for the word $\boldsymbol{u}$ to be different from the transmitted message $\boldsymbol{m}$. But an analysis of the meaning of the received message can eliminate

[2] This may happen when fro instance the number of errors is greater than t.

these cases and consider them as failures decoding. The complexity of this algorithm is of order $q^\ell T(\mathsf{dec})$ where $T(\mathsf{dec})$ is the time complexity of the decoding algorithm $\mathsf{dec}(\cdot)$.

Although the public code seems to be random in this description, a major problem rests on the choice of the code family to use and how to reduce the size of the keys. Wieschebrink had proposed the use of Reed-Solomon codes but in [12,13,5] an attack is presented that can recover the random secret matrix $\boldsymbol{R}$.

5 Recovering the Random Columns in Polynomial Time

Recently, the paper [14] suggested the use of Reed-Muller codes along with Wieschebrink's masking technique to propose a McEliece-type encryption scheme. In the next section, we describe how to find the random columns of $\boldsymbol{R}$ in this case. Our attack uses the same technique as the one presented in [12,13,5] for the case of Reed-Solomon codes.

5.1 Reed-Muller Based Encryption Scheme

In this section, we draw inspiration from [12,13,5] to mount an attack on the version presented in [14]. But before doing so, we present some properties of Reed-Muller codes.

Definition 4 (Reed-Muller Code). *Let $\mathbb{F}_2[x_1,\dots,x_m]$ be the set of boolean polynomials with m variables. Let us set $\{a_1,\dots,a_n\} \overset{def}{=} \mathbb{F}_2^m$ and $n \overset{def}{=} 2^m$. The Reed-Muller code denoted by $\mathcal{RM}(r,m)$ with $0 \leqslant r \leqslant m$ is the linear space defined by:*

$$\mathcal{RM}(r,m) \overset{def}{=} \Big\{ \big(f(a_1),\dots,f(a_n)\big) \ : \ f \in \mathbb{F}_2[x_1,\dots,x_m], \deg f \leqslant r \Big\}$$

We recall an immediate fact about the dimension of Reed-Muller codes.

Fact 1. *The dimension of $\mathcal{RM}(r,m)$ is equal to $\sum_{i=0}^{r} \binom{m}{i}$.*

Theorem 2 ([15] Chapter 13).

$$\mathcal{RM}(r,m)^\perp = \mathcal{RM}(m-r-1,m)$$

Proposition 2.

$$\mathcal{RM}(r,m)^2 = \mathcal{RM}(2r,m)$$

Proof. Let $c_1 = \big(f(a_1),\dots,f(a_n)\big)$ and $c_2 = \big(g(a_1),\dots,g(a_n)\big)$ be elements of $\mathcal{RM}(r,m)$ with $\deg f \leqslant r$ and $\deg g \leqslant r$. Hence, $c_1 \star c_2$ is the vector of evaluation $\big(fg(a_1),\dots,fg(a_n)\big)$ which corresponds to polynomial fg. This means $c_1 \star c_2 \in \mathcal{RM}(2r,m)$.

Conversely, each monomial $x_1^{e_1},\dots,x_m^{e_m}$ with $e_i \geqslant 0$ and $\sum_i e_i \leqslant 2r$ is the product of two polynomials of degree $\leqslant r$. This proves that a basis of $\mathcal{RM}(2r,m)$ is contained in $\mathcal{RM}(r,m)^2$.

Proposition 3. *Let $\boldsymbol{G}$ be a $k \times (n+\ell)$ matrix obtained by inserting ℓ random columns in the generating matrix of a Reed-Muller code $\mathcal{RM}(r,m)$ and let $\mathscr{C}$ be the code spanned by the rows of $\boldsymbol{G}$. Assume that $\ell \leqslant \binom{k}{2}$ and $\sum_{i=0}^{2r}\binom{m}{i} \leqslant n$. Then we have:*

$$\sum_{i=0}^{2r}\binom{m}{i} \leqslant \dim \mathscr{C}^2 \leqslant \sum_{i=0}^{2r}\binom{m}{i} + \ell \tag{2}$$

Proof. Let $\mathscr{D}_1$ be the code with generating matrix G_1 obtained from $\boldsymbol{G}$ by replacing the last ℓ columns by all-zero columns and let $\mathscr{D}_2$ be the code with generating matrix $\boldsymbol{G}_2$ obtained by replacing in $\boldsymbol{G}$ the first n columns by zero columns. Hence $\boldsymbol{G} = \boldsymbol{G}_1 + \boldsymbol{G}_2$ which implies $\mathscr{D}_1 \subseteq \mathscr{C} \subseteq \mathscr{D}_1 + \mathscr{D}_2$. We have $\mathscr{D}_1 \star \mathscr{D}_2 = 0$ and the following inclusion:

$${\mathscr{D}_1}^2 \subseteq \mathscr{C}^2 \subseteq {\mathscr{D}_1}^2 + {\mathscr{D}_2}^2 + \mathscr{D}_1 \star \mathscr{D}_2.$$

Observe we have $\mathscr{D}_1 \star \mathscr{D}_2 = 0$. By also remarking $\dim {\mathscr{D}_1}^2 = \dim \mathcal{RM}(2r,m)$ and $\dim {\mathscr{D}_2}^2 = \min\left\{\ell, \binom{k}{2}\right\} = \ell$, one can conclude (2) is proven.

5.2 Description of the Attack

It is easy for an adversary to use Prop. 3 to identify the random columns by computing the dimension of $\mathscr{C}^2$ where $\mathscr{C}$ is the code generated by the public matrix $\boldsymbol{G}$ as defined in Sec. 4. We recall that $\mathscr{C}$ is permuted version of a Reed-Muller code $\mathcal{RM}(r,m)$. We assume that $\sum_{i=0}^{2r}\binom{m}{i} \leqslant n_0$ where $n_0 = 2^m$ and $\ell < \binom{k}{2}$ where $k = \sum_{i=0}^{r}\binom{m}{i}$. We denote by $\mathscr{C}_i$ the code generated by the generating matrix $\boldsymbol{G}_i$ obtained by deleting the i-th column of $\boldsymbol{G}$. We also denote by $I \subset \{1,\dots,n\}$ the set of positions that define the random columns inserted in $\boldsymbol{G}$. Two cases occur with high probability:

$$\dim {\mathscr{C}_i}^2 = \begin{cases} \dim \mathscr{C}^2 - 1 & \text{if } \ i \in I, \\[2ex] \dim \mathscr{C}^2 & \text{if } \ i \notin I. \end{cases} \tag{3}$$

Once the set I is recovered, it is then easy to find the secret $\mathcal{RM}(r,m)$ using usual attacks on Reed-Muller code [18].

Remark 1. For the parameters in [14], we observed experimentally that (3) is always true, and the upper-bound given in (2) is always reached, that is to say:

$$\dim \mathscr{C}^2 = \sum_{i=0}^{2r}\binom{m}{i} + \ell.$$

This is way of distinguishing the random positions of the public code assumes that $\sum_{i=0}^{2r}\binom{m}{i} + \ell \geqslant n$. We will see how to deal with parameters that do not satisfy this assumption. The idea is to look at $\dim \mathscr{D}^2$ where $\mathscr{D}$ is the dual of $\mathscr{C}$. Indeed, like generalized Reed-Solomon codes, the family of Reed-Muller codes is stable under duality (Theorem 2).

Proposition 4. *Keeping with notation of Proposition 3, let $\mathscr{D}$ be the dual of $\mathscr{C}$. Assuming $\sum_{i=0}^{2r}\binom{m}{i} > n_0$, we have:*

$$\dim \mathscr{D}^2 \leqslant \frac{1}{2}\ell(\ell+1) + \ell \sum_{i=0}^{m-r-1}\binom{m}{i} + \sum_{i=0}^{2(m-r-1)}\binom{m}{i}. \qquad (4)$$

Proof. Let us set $k = \sum_{i=0}^{r}\binom{m}{i}$. We may assume without loss of generality that a generating matrix of $\mathscr{C}$ is in systematic form: $\begin{pmatrix} \boldsymbol{I}_k & \boldsymbol{A} & \boldsymbol{R} \end{pmatrix}$ where $\boldsymbol{R}$ form the random columns and $\begin{pmatrix} \boldsymbol{I}_k & \boldsymbol{A} \end{pmatrix}$ generates $\mathcal{RM}(r,m)$. A parity-check matrix for $\mathscr{C}$ is then:

$$\boldsymbol{H} = \begin{pmatrix} -\boldsymbol{A}^T & \boldsymbol{I}_{n_0-k} & \boldsymbol{0} \\ -\boldsymbol{R}^T & \boldsymbol{0} & \boldsymbol{I}_\ell \end{pmatrix}.$$

The upper-bound (4) can be readily derived from this last matrix $\boldsymbol{H}$.

5.3 Complexity of the Attack

Proposition 5. *Let $\mathscr{A} \subset \mathbb{F}_q^n$ be a code of dimension k. The complexity of the computation of a basis of $\mathscr{A}^2$ is $O(k^2n^2)$ operations in $\mathbb{F}_q$.*

Proof. The computation, consists first in the computation of $\binom{k+1}{2}$ generators of $\mathscr{A}^2$. This computation costs $O(k^2n)$ operations. Then, we have to apply a Gaussian elimination to a $\binom{k+1}{2} \times n$ matrix, which costs $O(k^2n^2)$ operations. This second step is dominant, which yields the result.

Our attack relies on the computation of the rank of n square codes so the overall complexity for guessing the random columns is $O(n^5)$ operations in the binary field.

6 Conclusion

In this paper, we study the security of the modified version of the Sidelnikov scheme [23] given in [14]. We have presented a polynomial-time method that finds the random columns inserted in a secret matrix. This cryptanalysis uses the same approach as [12,13,5] which computes the square codes. The resulting complexity is $O(n^5)$ operations in the binary field. The last step that aims to fully break the scheme ressort to using the attacks developed in [18,3]. Our work shows that the insertion of random columns in the Sidelnikov scheme does not bring any security improvement.

References

1. Berger, T.P., Cayrel, P.-L., Gaborit, P., Otmani, A.: Reducing key length of the McEliece cryptosystem. In: Preneel, B. (ed.) AFRICACRYPT 2009. LNCS, vol. 5580, pp. 77–97. Springer, Heidelberg (2009)
2. Berger, T.P., Loidreau, P.: How to mask the structure of codes for a cryptographic use. Des. Codes Cryptogr. 35(1), 63–79 (2005)
3. Chizhov, I.V., Borodin, M.A.: The failure of McEliece PKC based on Reed-Muller codes. IACR Cryptology ePrint Archive, Report 2013/287 (2013), http://eprint.iacr.org/
4. Courtois, N.T., Finiasz, M., Sendrier, N.: How to achieve a McEliece-based digital signature scheme. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 157–174. Springer, Heidelberg (2001)
5. Couvreur, A., Gaborit, P., Gauthier-Umaña, V., Otmani, A., Tillich, J.P.: Distinguisher-based attacks on public-key cryptosystems using Reed-Solomon codes. Des. Codes Cryptogr. 73(2), 641–666 (2014), http://dx.doi.org/10.1007/s10623-014-9967-z
6. Faugère, J.C., Gauthier, V., Otmani, A., Perret, L., Tillich, J.P.: A distinguisher for high rate McEliece cryptosystems. In: Proc. IEEE Inf. Theory Workshop, ITW 2011, Paraty, Brasil, pp. 282–286 (October 2011)
7. Faugère, J.C., Gauthier, V., Otmani, A., Perret, L., Tillich, J.P.: A distinguisher for high rate McEliece cryptosystems. IEEE Trans. Inf. Theory 59(10), 6830–6844 (2013)
8. Faugère, J.C., Otmani, A., Perret, L., de Portzamparc, F., Tillich, J.P.: Structural weakness of compact variants of the McEliece cryptosystem. In: Proc. IEEE Int. Symposium Inf. Theory, ISIT 2014, Honolulu, HI, USA, pp. 1717–1721 (July 2014)
9. Faugère, J.C., Otmani, A., Perret, L., de Portzamparc, F., Tillich, J.P.: Structural cryptanalysis of McEliece schemes with compact keys. Des. Codes Cryptogr. (2015), to appear, see also IACR Cryptology ePrint Archive, Report2014/210
10. Faugère, J.-C., Otmani, A., Perret, L., Tillich, J.-P.: Algebraic cryptanalysis of McEliece variants with compact keys. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 279–298. Springer, Heidelberg (2010)
11. Gaborit, P.: Shorter keys for code based cryptography. In: Proceedings of the 2005 International Workshop on Coding and Cryptography (WCC 2005), Bergen, Norway, pp. 81–91 (March 2005)
12. Gauthier, V., Otmani, A., Tillich, J.P.: A distinguisher-based attack of a homomorphic encryption scheme relying on Reed-Solomon codes. CoRR abs/1203.6686 (2012)
13. Gauthier, V., Otmani, A., Tillich, J.P.: A distinguisher-based attack on a variant of McEliece's cryptosystem based on Reed-Solomon codes. CoRR abs/1204.6459 (2012)
14. Gueye, C.T., Mboup, E.H.M.: Secure cryptographic scheme based on modified Reed Muller codes. International Journal of Security and its Applications 7(3), 55–64 (2013)
15. MacWilliams, F.J., Sloane, N.J.A.: The Theory of Error-Correcting Codes, 4th edn. North–Holland, Amsterdam (1986)
16. Márquez-Corbella, I., Pellikaan, R.: Error-correcting pairs for a public-key cryptosystem. preprint (2012) (preprint)
17. McEliece, R.J.: A Public-Key System Based on Algebraic Coding Theory, pp. 114–116. Jet Propulsion Lab (1978), dSN Progress Report 44

18. Minder, L., Shokrollahi, M.A.: Cryptanalysis of the Sidelnikov cryptosystem. In: Naor, M. (ed.) EUROCRYPT 2007. LNCS, vol. 4515, pp. 347–360. Springer, Heidelberg (2007)
19. Misoczki, R., Barreto, P.: Compact McEliece keys from Goppa codes. In: Selected Areas in Cryptography, Calgary, Canada (August 13-14, 2009)
20. Niederreiter, H.: Knapsack-type cryptosystems and algebraic coding theory. Problems of Control and Information Theory 15(2), 159–166 (1986)
21. Sendrier, N.: Cryptosystèmes à clé publique basés sur les codes correcteurs d'erreurs. Ph.D. thesis, Université Paris 6, France (2002)
22. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM J. Comput. 26(5), 1484–1509 (1997)
23. Sidelnikov, V.M.: A public-key cryptosytem based on Reed-Muller codes. Discrete Mathematics and Applications 4(3), 191–207 (1994)
24. Sidelnikov, V.M., Shestakov, S.: On the insecurity of cryptosystems based on generalized Reed-Solomon codes. Discrete Mathematics and Applications 1(4), 439–444 (1992)
25. Wieschebrink, C.: Two NP-complete problems in coding theory with an application in code based cryptography. In: Proc. IEEE Int. Symposium Inf. Theory, ISIT 2006, pp. 1733–1737 (2006)
26. Wieschebrink, C.: Cryptanalysis of the Niederreiter public key scheme based on GRS subcodes. IACR Cryptology ePrint Archive, Report 2009/452 (2009), http://eprint.iacr.org/2009/452.pdf
27. Wieschebrink, C.: Cryptanalysis of the Niederreiter public key scheme based on GRS subcodes. In: Post-Quantum Cryptography 2010, pp. 61–72 (2010)