

# Code-Based Cryptography

1. Error-Correcting Codes and Cryptography
2. McEliece Cryptosystem
3. Message Attacks (ISD)
4. **Key Attacks**
5. Other Cryptographic Constructions Relying on Coding Theory

## 4. Key Attacks

1. Introduction
2. Support Splitting Algorithm
3. Distinguisher for GRS codes
4. **Attack against subcodes of GRS codes**
5. Error-Correcting Pairs
6. Attack against GRS codes
7. Attack against Reed-Muller codes
8. Attack against Algebraic Geometry codes
9. Goppa codes still resist

# Subcodes of GRS codes for the McEliece scheme



## Subcodes of GRS codes



T. Berger and P. Loidreau.

*How to mask the structure of codes for a cryptographic use.*

Des. Codes Cryptogr., 35:63—79, 2005.

# Subcodes of GRS codes for the McEliece scheme



## Subcodes of GRS codes



T. Berger and P. Loidreau.

*How to mask the structure of codes for a cryptographic use.*

Des. Codes Cryptogr., 35:63–79, 2005.



## Attack against this proposal:



C. Wieschebrink.

*Cryptanalysis of the Niederreiter public key scheme based on GRS subcodes.*

In Post-Quantum Cryptography, volume 6061 of Lecture Notes in Comput. Sci., pages 61–72, 2010.

## Attack - If $2k - 1 \leq n - 2$

Public Key:  $\mathcal{K}_{\text{pub}} = \begin{cases} \text{a gen. matrix of } \mathcal{C} \subseteq \text{GRS}_k(\mathbf{a}, \mathbf{b}) \\ \text{and } t = \left\lfloor \frac{n-k}{2} \right\rfloor \end{cases}$

## Attack - If $2k - 1 \leq n - 2$

Public Key:  $\mathcal{K}_{\text{pub}} = \left\{ \begin{array}{l} \text{a gen. matrix of } \mathcal{C} \subseteq \text{GRS}_k(\mathbf{a}, \mathbf{b}) \\ \text{and } t = \left\lfloor \frac{n-k}{2} \right\rfloor \end{array} \right.$

### The Algorithm:

STEP 1 Compute  $\mathcal{C}^{(2)}$ .

With **High Probability**:

$$\mathcal{C}^{(2)} = \text{GRS}_k(\mathbf{a}, \mathbf{b})^{(2)} = \text{GRS}_{2k-1}(\mathbf{a}, \mathbf{b} * \mathbf{b})$$

## Attack - If $2k - 1 \leq n - 2$

Public Key:  $\mathcal{K}_{\text{pub}} = \begin{cases} \text{a gen. matrix of } \mathcal{C} \subseteq \text{GRS}_k(\mathbf{a}, \mathbf{b}) \\ \text{and } t = \left\lfloor \frac{n-k}{2} \right\rfloor \end{cases}$

### The Algorithm:

STEP 1 Compute  $\mathcal{C}^{(2)}$ .

With **High Probability**:

$$\mathcal{C}^{(2)} = \text{GRS}_k(\mathbf{a}, \mathbf{b})^{(2)} = \text{GRS}_{2k-1}(\mathbf{a}, \mathbf{b} * \mathbf{b})$$

STEP 2 Apply the **Sidelnikov-Shestakov** attack to recover

**a**    and    **b \* b**

# Shortened code

Let:

→  $\mathcal{C}$  be an  $[n, k]_q$  code



# Shortened code

Let:

- $\mathcal{C}$  be an  $[n, k]_q$  code
- $(J, \bar{J})$  be a partition of  $\{1, \dots, n\}$

# Shortened code

Let:

- $\mathcal{C}$  be an  $[n, k]_q$  code
- $(J, \bar{J})$  be a partition of  $\{1, \dots, n\}$
- $\mathbf{x}_J$  the **restriction** of  $\mathbf{x} \in \mathbb{F}_q^n$  to the coordinates indexed by  $J$

# Shortened code

Let:

- $\mathcal{C}$  be an  $[n, k]_q$  code
- $(J, \bar{J})$  be a partition of  $\{1, \dots, n\}$
- $\mathbf{x}_J$  the **restriction** of  $\mathbf{x} \in \mathbb{F}_q^n$  to the coordinates indexed by  $J$

## Shortened code $S_J(\mathcal{C})$

The words of  $S_J(\mathcal{C})$  are codewords of  $\mathcal{C}$  that have a zero in the  $J$ -locations, i.e.

$$S_J(\mathcal{C}) = \{\mathbf{c}_{\bar{J}} \mid \mathbf{c} \in \mathcal{C} \text{ and } c_j = 0 \text{ for all } j \in J\}$$

# Shortened code

Let:

- $\mathcal{C}$  be an  $[n, k]_q$  code
- $(J, \bar{J})$  be a partition of  $\{1, \dots, n\}$
- $\mathbf{x}_J$  the **restriction** of  $\mathbf{x} \in \mathbb{F}_q^n$  to the coordinates indexed by  $J$

$$G = \begin{array}{c|c} \begin{array}{ccc} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{array} & \begin{array}{c} \text{ } \end{array} \\ \hline \begin{array}{ccc} 0 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 0 \end{array} & \begin{array}{c} \text{ } \end{array} \end{array}$$

Diagram illustrating the generator matrix  $G$  for a shortened code. The matrix is partitioned into four blocks. The top-left block is an identity matrix of size  $|J| \times |J|$ . The top-right block is a zero matrix of size  $|J| \times (n - |J|)$ . The bottom-left block is a zero matrix of size  $(k - |J|) \times |J|$ . The bottom-right block is a zero matrix of size  $(k - |J|) \times (n - |J|)$ . The dimensions are indicated by arrows:  $|J|$  for the top-left block,  $n - |J|$  for the top-right block,  $|J|$  for the bottom-left block, and  $k - |J|$  for the bottom-right block.

## Shortened code $S_J(\mathcal{C})$

The words of  $S_J(\mathcal{C})$  are codewords of  $\mathcal{C}$  that have a zero in the  $J$ -locations, i.e.

$$S_J(\mathcal{C}) = \{\mathbf{c}_{\bar{J}} \mid \mathbf{c} \in \mathcal{C} \text{ and } c_j = 0 \text{ for all } j \in J\}$$

# Shortened code

Let:

- $\mathcal{C}$  be an  $[n, k]_q$  code
- $(J, \bar{J})$  be a partition of  $\{1, \dots, n\}$
- $\mathbf{x}_J$  the **restriction** of  $\mathbf{x} \in \mathbb{F}_q^n$  to the coordinates indexed by  $J$

$$G = \begin{array}{c|c} \begin{array}{ccc} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{array} & \begin{array}{c} \text{ } \end{array} \\ \hline \begin{array}{ccc} 0 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 0 \end{array} & \begin{array}{c} \text{Generator matrix} \\ \text{for } S_J(\mathcal{C}) \end{array} \end{array}$$

Diagram illustrating the structure of the generator matrix  $G$  for the shortened code  $S_J(\mathcal{C})$ . The matrix is partitioned into four blocks. The top-left block is an identity matrix of size  $|J| \times |J|$ . The top-right block is a  $|J| \times (n - |J|)$  matrix. The bottom-left block is a  $(k - |J|) \times |J|$  matrix of zeros. The bottom-right block is a  $(k - |J|) \times (n - |J|)$  matrix labeled "Generator matrix for  $S_J(\mathcal{C})$ ". The total number of rows is  $k$  and the total number of columns is  $n$ .

## Shortened code $S_J(\mathcal{C})$

The words of  $S_J(\mathcal{C})$  are codewords of  $\mathcal{C}$  that have a zero in the  $J$ -locations, i.e.

$$S_J(\mathcal{C}) = \{\mathbf{c}_{\bar{J}} \mid \mathbf{c} \in \mathcal{C} \text{ and } c_j = 0 \text{ for all } j \in J\}$$

# Shortening a GRS code

The shortened code of a GRS code is GRS code

For GRS code we always have:

$$S_J(\text{GRS}_k(\mathbf{a}, \mathbf{b})) = \text{GRS}_{n-|J|}(\mathbf{a}_{\bar{J}}, \mathbf{b}') \text{ with } b'_i = b_i \prod_{j \in J} (a_i - a_j)$$

**Proof:** Assume  $J = \{1\}$ . Let  $G$  be a gen. matrix for  $\text{GRS}_k(\mathbf{a}, \mathbf{b})$ .

$$G = \begin{array}{cccc} b_1 & b_2 & \dots & b_n \\ b_1 a_1 & b_2 a_2 & \dots & b_n a_n \\ \vdots & \vdots & \ddots & \vdots \\ b_1 a_1^{k-1} & b_2 a_2^{k-1} & \dots & b_n a_n^{k-1} \end{array}$$

# Shortening a GRS code

The shortened code of a GRS code is GRS code

For GRS code we always have:

$$S_J(\text{GRS}_k(\mathbf{a}, \mathbf{b})) = \text{GRS}_{n-|J|}(\mathbf{a}_{\bar{J}}, \mathbf{b}') \text{ with } b'_i = b_i \prod_{j \in J} (a_i - a_j)$$

**Proof:** Assume  $J = \{1\}$ . Let  $G$  be a gen. matrix for  $\text{GRS}_k(\mathbf{a}, \mathbf{b})$ .

$$G = \begin{array}{cccc} \boxed{b_1} & \boxed{b_2} & \dots & \boxed{b_n} \\ b_1 a_1 & b_2 a_2 & \dots & b_n a_n \\ \vdots & \vdots & \ddots & \vdots \\ b_1 a_1^{k-1} & b_2 a_2^{k-1} & \dots & b_n a_n^{k-1} \end{array} \xrightarrow{\text{red arrow}} \mathbf{g}_1$$

# Shortening a GRS code

The shortened code of a GRS code is GRS code

For GRS code we always have:

$$S_J(\text{GRS}_k(\mathbf{a}, \mathbf{b})) = \text{GRS}_{n-|J|}(\mathbf{a}_{\bar{J}}, \mathbf{b}') \text{ with } b'_i = b_i \prod_{j \in J} (a_i - a_j)$$

**Proof:** Assume  $J = \{1\}$ . Let  $G$  be a gen. matrix for  $\text{GRS}_k(\mathbf{a}, \mathbf{b})$ .

$$G = \begin{array}{cccc} b_1 & b_2 & \dots & b_n \\ b_1 a_1 & b_2 a_2 & \dots & b_n a_n \\ \vdots & \vdots & \ddots & \vdots \\ b_1 a_1^{k-1} & b_2 a_2^{k-1} & \dots & b_n a_n^{k-1} \end{array} \xrightarrow{\text{red arrow}} \mathbf{g}_2$$



# Shortening a GRS code

The shortened code of a GRS code is GRS code

For GRS code we always have:

$$S_J(\text{GRS}_k(\mathbf{a}, \mathbf{b})) = \text{GRS}_{n-|J|}(\mathbf{a}_{\bar{J}}, \mathbf{b}') \text{ with } b'_i = b_i \prod_{j \in J} (a_i - a_j)$$

**Proof:** Assume  $J = \{1\}$ . Let  $G$  be a gen. matrix for  $\text{GRS}_k(\mathbf{a}, \mathbf{b})$ .

$$G = \begin{array}{cccc} b_1 & b_2 & \dots & b_n \\ b_1 a_1 & b_2 a_2 & \dots & b_n a_n \\ \vdots & \vdots & \ddots & \vdots \\ b_1 a_1^{k-1} & b_2 a_2^{k-1} & \dots & b_n a_n^{k-1} \end{array} \xrightarrow{\text{red arrow}} \mathbf{g}_k$$

# Shortening a GRS code

The shortened code of a GRS code is GRS code

For GRS code we always have:

$$S_J(\text{GRS}_k(\mathbf{a}, \mathbf{b})) = \text{GRS}_{n-|J|}(\mathbf{a}_{\bar{J}}, \mathbf{b}') \text{ with } b'_i = b_i \prod_{j \in J} (a_i - a_j)$$

**Proof:** Assume  $J = \{1\}$ . Let  $G$  be a gen. matrix for  $\text{GRS}_k(\mathbf{a}, \mathbf{b})$ .

$$G = \begin{array}{cccc} \boxed{1} & * & \dots & * \\ b_1 a_1 & b_2 a_2 & \dots & b_n a_n \\ \vdots & \vdots & \ddots & \vdots \\ b_1 a_1^{k-1} & b_2 a_2^{k-1} & \dots & b_n a_n^{k-1} \end{array} \xrightarrow{\text{red arrow}} \mathbf{g}'_1 = \frac{\mathbf{g}_1}{b_1}$$

# Shortening a GRS code

The shortened code of a GRS code is GRS code

For GRS code we always have:

$$S_J(\text{GRS}_k(\mathbf{a}, \mathbf{b})) = \text{GRS}_{n-|J|}(\mathbf{a}_{\bar{J}}, \mathbf{b}') \text{ with } b'_i = b_i \prod_{j \in J} (a_i - a_j)$$

**Proof:** Assume  $J = \{1\}$ . Let  $G$  be a gen. matrix for  $\text{GRS}_k(\mathbf{a}, \mathbf{b})$ .

$$G = \begin{array}{|c|} \hline \begin{array}{cccc} 1 & * & \dots & * \end{array} \\ \hline \end{array} \begin{array}{l} \xrightarrow{\text{red}} \mathbf{g}'_1 = \frac{\mathbf{g}_1}{b_1} \\ \xrightarrow{\text{red}} \mathbf{g}'_i = \mathbf{g}_i - a_1 \mathbf{g}_{i-1}, \text{ for all } i \geq 2 \end{array}$$

# Shortening a GRS code

The shortened code of a GRS code is GRS code

For GRS code we always have:

$$S_J(\text{GRS}_k(\mathbf{a}, \mathbf{b})) = \text{GRS}_{n-|J|}(\mathbf{a}_{\bar{J}}, \mathbf{b}') \text{ with } b'_i = b_i \prod_{j \in J} (a_i - a_j)$$

**Proof:** Assume  $J = \{1\}$ . Let  $G$  be a gen. matrix for  $\text{GRS}_k(\mathbf{a}, \mathbf{b})$ .

$$G = \begin{array}{cccc} 1 & * & \dots & * \\ 0 & b'_2 & \dots & b'_n \\ \vdots & \vdots & \ddots & \vdots \\ 0 & b'_2 a_2^{k-2} & \dots & b'_n a_n^{k-2} \end{array} \begin{array}{l} \rightarrow \mathbf{g}'_1 = \frac{\mathbf{g}_1}{b_1} \\ \rightarrow \mathbf{g}'_i = \mathbf{g}_i - a_1 \mathbf{g}_{i-1}, \text{ for all } i \geq 2 \end{array}$$

$$\text{Thus, } g'_{ij} = \begin{cases} 0 & \text{if } j = 1 \\ \underbrace{b_j (a_j - a_1)}_{b'_j} a_j^{i-1} & \text{if } j \geq 2 \end{cases}$$

# Shortening a GRS code

The shortened code of a GRS code is GRS code

For GRS code we always have:

$$S_J(\text{GRS}_k(\mathbf{a}, \mathbf{b})) = \text{GRS}_{n-|J|}(\mathbf{a}_{\bar{J}}, \mathbf{b}')$$

$\text{with } b'_i = b_i \prod_{j \in J} (a_i - a_j)$

**Proof:** Assume  $J = \{1\}$ . Let  $G$  be a gen. matrix for  $\text{GRS}_k(\mathbf{a}, \mathbf{b})$ .

$$G = \begin{array}{c|cccc} & 1 & * & \dots & * \\ \hline 0 & b'_2 & \dots & b'_n \\ \vdots & \vdots & \ddots & \vdots \\ 0 & b'_2 a_2^{k-2} & \dots & b'_n a_n^{k-2} \end{array}$$

Generator matrix  
for  $S_1(\text{GRS}_k(\mathbf{a}, \mathbf{b}))$

$\mathbf{g}'_1 = \frac{\mathbf{g}_1}{b_1}$

$\mathbf{g}'_i = \mathbf{g}_i - a_1 \mathbf{g}_{i-1}, \text{ for all } i \geq 2$

Thus,  $g'_{ij} = \begin{cases} 0 & \text{if } j = 1 \\ \underbrace{b_j (a_j - a_1)}_{b'_j} a_j^{i-1} & \text{if } j \geq 2 \end{cases}$

## Attack - If $2k - 1 > n - 2$

Public Key:  $\mathcal{K}_{\text{pub}} = \left\{ \begin{array}{l} \text{a gen. matrix of } \mathcal{C} \subseteq \text{GRS}_k(\mathbf{a}, \mathbf{b}) \\ \text{and } t = \left\lfloor \frac{n-k}{2} \right\rfloor \end{array} \right.$

## Attack - If $2k - 1 > n - 2$

Public Key:  $\mathcal{K}_{\text{pub}} = \left\{ \begin{array}{l} \text{a gen. matrix of } \mathcal{C} \subseteq \text{GRS}_k(\mathbf{a}, \mathbf{b}) \\ \text{and } t = \left\lfloor \frac{n-k}{2} \right\rfloor \end{array} \right.$

### The Algorithm:

STEP 1 Chose a set of indices

$$J = \{i_1, \dots, i_N\} \subseteq \{1, \dots, n\} \text{ such that } 2(k - N) \leq n - 2$$

## Attack - If $2k - 1 > n - 2$

Public Key:  $\mathcal{K}_{\text{pub}} = \begin{cases} \text{a gen. matrix of } \mathcal{C} \subseteq \text{GRS}_k(\mathbf{a}, \mathbf{b}) \\ \text{and } t = \left\lfloor \frac{n-k}{2} \right\rfloor \end{cases}$

### The Algorithm:

**STEP 1** Chose a set of indices

$$J = \{i_1, \dots, i_N\} \subseteq \{1, \dots, n\} \text{ such that } 2(k - N) \leq n - 2$$

**STEP 2** Compute a generator matrix of the shortened code  $S_J(\mathcal{C})$

$$S_J(\mathcal{C}) \subseteq \text{GRS}_{k-N}(\mathbf{a}_J, \mathbf{b}')$$

Recall that

$$\text{with } b'_i = b_i \prod_{j \in J} (a_i - a_j) \text{ for all } i \in J$$



## Attack - If $2k - 1 > n - 2$

Public Key:  $\mathcal{K}_{\text{pub}} = \begin{cases} \text{a gen. matrix of } \mathcal{C} \subseteq \text{GRS}_k(\mathbf{a}, \mathbf{b}) \\ \text{and } t = \left\lfloor \frac{n-k}{2} \right\rfloor \end{cases}$

### The Algorithm:

**STEP 1** Chose a set of indices

$$J = \{i_1, \dots, i_N\} \subseteq \{1, \dots, n\} \text{ such that } 2(k - N) \leq n - 2$$

**STEP 2** Compute a generator matrix of the shortened code  $S_J(\mathcal{C}) \subseteq \text{GRS}_{k-N}(\mathbf{a}_J, \mathbf{b}')$

**STEP 3** Apply the previous algorithm to retrieve  $\mathbf{a}_J$  and  $\mathbf{b}'$ .

Note that  $2(k - N) \leq n - 2$ .

## Attack - If $2k - 1 > n - 2$

Public Key:  $\mathcal{K}_{\text{pub}} = \begin{cases} \text{a gen. matrix of } \mathcal{C} \subseteq \text{GRS}_k(\mathbf{a}, \mathbf{b}) \\ \text{and } t = \left\lfloor \frac{n-k}{2} \right\rfloor \end{cases}$

### The Algorithm:

**STEP 1** Chose a set of indices

$$J = \{i_1, \dots, i_N\} \subseteq \{1, \dots, n\} \text{ such that } 2(k - N) \leq n - 2$$

**STEP 2** Compute a generator matrix of the shortened code  $S_J(\mathcal{C}) \subseteq \text{GRS}_{k-N}(\mathbf{a}_J, \mathbf{b}')$

**STEP 3** Apply the previous algorithm to retrieve  $\mathbf{a}_J$  and  $\mathbf{b}'$ .

Note that  $2(k - N) \leq n - 2$ .

**STEP 4** Return to **STEP 1** until  $\mathbf{a}$  is completely retrieved.

## 4. Key Attacks

1. Introduction
2. Support Splitting Algorithm
3. Distinguisher for GRS codes
4. Attack against subcodes of GRS codes
5. **Error-Correcting Pairs**
6. Attack against GRS codes
7. Attack against Reed-Muller codes
8. Attack against Algebraic Geometry codes
9. Goppa codes still resist