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
GRS codes for the McEliece scheme

Generalized Reed-Solomon codes

H. Niederreiter.

*Knapsack-type cryptosystems and algebraic coding theory.*

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Key size</th>
<th>Security level</th>
</tr>
</thead>
<tbody>
<tr>
<td>[256, 128, 129]_{256}</td>
<td>67 ko</td>
<td>$2^{95}$</td>
</tr>
</tbody>
</table>
GRS codes for the McEliece scheme

Generalized Reed-Solomon codes

H. Niederreiter.
*Knapsack-type cryptosystems and algebraic coding theory.*

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Key size</th>
<th>Security level</th>
</tr>
</thead>
<tbody>
<tr>
<td>([256, 128, 129]_{256})</td>
<td>67 ko</td>
<td>(2^{95})</td>
</tr>
</tbody>
</table>

Attack against this proposal:

V. M. Sidelnikov and S. O. Shestakov.
*On the insecurity of cryptosystems based on generalized Reed-Solomon codes.*
Filtration Attack for GRS codes

Suppose that we know:

\[ C_k = \text{GRS}_k(a, b) \quad \text{and} \quad C_{k-1} = \text{GRS}_{k-1}(a, b) \]

Proposition: Assume that \( 2k - 1 \leq n - 2 \)

\[ C_{k-2} = \text{GRS}_{k-2}(a, b) \]

is the solution space of the following problem

\[ c \in C_{k-1} \quad \text{and} \quad c \ast C_k \subseteq (C_{k-1})^2 \]
Filtration Attack for GRS codes

Suppose that we know:

\[ C_k = \text{GRS}_k(a, b) \quad \text{and} \quad C_{k-1} = \text{GRS}_{k-1}(a, b) \]

**Proposition:** Assume that \(2k - 1 \leq n - 2\)

\[ C_{k-2} = \text{GRS}_{k-2}(a, b) \]

is the solution space of the following problem

\[ c \in C_{k-1} \quad \text{and} \quad c \times C_k \subseteq (C_{k-1})^2 \]

**Proof:** [Sketch of the Proof]
Filtration Attack for GRS codes

Suppose that we know:

\[ C_k = \text{GRS}_k(a, b) \quad \text{and} \quad C_{k-1} = \text{GRS}_{k-1}(a, b) \]

**Proposition:** Assume that \( 2k - 1 \leq n - 2 \)

\[ C_{k-2} = \text{GRS}_{k-2}(a, b) \]

is the solution space of the following problem

\[ c \in C_{k-1} \quad \text{and} \quad c \ast C_k \subseteq (C_{k-1})^2 \]

**Proof:** [Sketch of the Proof]

\[ C_{k-1} \ast C_k = (C_{k-1})^2 \]
Filtration Attack for GRS codes

Suppose that we know:

\[ C_k = \text{GRS}_k(a, b) \quad \text{and} \quad C_{k-1} = \text{GRS}_{k-1}(a, b) \]

Proposition: Assume that \( 2k - 1 \leq n - 2 \)

\[ C_{k-2} = \text{GRS}_{k-2}(a, b) \]

is the solution space of the following problem

\[ c \in C_{k-1} \quad \text{and} \quad c \ast C_k \subseteq (C_{k-1})^2 \]

Proof: [Sketch of the Proof]

\[ C_{k-1} \ast C_k = (C_{k-1})^2 \]

\[ (b \ast f(a)) \ast (b \ast g(a)) = (b \ast b)(fg)(a) \]

with \( \deg(f) < k - 1 \), \( \deg(g) < k \) \( \Rightarrow \) \( \deg(fg) < 2k - 2 \)
Filtration Attack for GRS codes

Suppose that we know:

\[ C_k = \text{GRS}_k(a, b) \quad \text{and} \quad C_{k-1} = \text{GRS}_{k-1}(a, b) \]

Proposition: Assume that \( 2k - 1 \leq n - 2 \)

\[ C_{k-2} = \text{GRS}_{k-2}(a, b) \]

is the solution space of the following problem

\[ c \in C_{k-1} \quad \text{and} \quad c \times C_k \subseteq (C_{k-1})^2 \]

In this way we build the following filtration

\[ \text{GRS}_k(a, b) \supseteq \text{GRS}_{k-1}(a, b) \supseteq \text{GRS}_{k-2}(a, b) \supseteq \cdots \supseteq \text{GRS}_1(a, b) \]
Filtration Attack for GRS codes

Suppose that we know:

\[ C_k = \text{GRS}_k(a, b) \quad \text{and} \quad C_{k-1} = \text{GRS}_{k-1}(a, b) \]

Proposition: Assume that \(2k - 1 \leq n - 2\)

\[ C_{k-2} = \text{GRS}_{k-2}(a, b) \]

is the solution space of the following problem

\[ c \in C_{k-1} \quad \text{and} \quad c \cdot C_k \subseteq (C_{k-1})^2 \]

In this way we build the following filtration

\[ \text{GRS}_k(a, b) \supseteq \text{GRS}_{k-1}(a, b) \supseteq \text{GRS}_{k-2}(a, b) \supseteq \cdots \supseteq \text{GRS}_1(a, b) \]

Note that:

\[ \text{GRS}_1(a, b) = \{ \alpha b \mid \alpha \in \mathbb{F}_q^* \} \]
Filtration Attack for GRS codes

Suppose that we know:

\[ C_k = \text{GRS}_k(a, b) \quad \text{and} \quad C_{k-1} = \text{GRS}_{k-1}(a, b) \]

**Proposition:** Assume that \( 2k - 1 \leq n - 2 \)

\[ C_{k-2} = \text{GRS}_{k-2}(a, b) \]

is the solution space of the following problem

\[ c \in C_{k-1} \quad \text{and} \quad c \ast C_k \subseteq (C_{k-1})^2 \]

In this way we build the following filtration

\[ \text{GRS}_k(a, b) \supseteq \text{GRS}_{k-1}(a, b) \supseteq \text{GRS}_{k-2}(a, b) \supseteq \cdots \supseteq \text{GRS}_1(a, b) \]

Note that: \( \text{GRS}_1(a, b) = \{ \alpha b \mid \alpha \in \mathbb{F}_q^* \} \)

So we get the **column multiplier b.**
Filtration Attack for GRS codes

Suppose that we know:
\[ C_k = \text{GRS}_k(a, b) \quad \text{and} \quad C_{k-1} = \text{GRS}_{k-1}(a, b) \]

**Proposition:** Assume that \( 2k - 1 \leq n - 2 \)

\[ C_{k-2} = \text{GRS}_{k-2}(a, b) \] is the solution space of the following problem

\[ \mathbf{c} \in C_{k-1} \quad \text{and} \quad \mathbf{c} \ast C_k \subseteq (C_{k-1})^2 \]

In this way we build the following filtration

\[ \text{GRS}_k(a, b) \supseteq \text{GRS}_{k-1}(a, b) \supseteq \text{GRS}_{k-2}(a, b) \supseteq \cdots \supseteq \text{GRS}_1(a, b) \]

Note that:
\[ \text{GRS}_1(a, b) = \{ \alpha \mathbf{b} \mid \alpha \in \mathbb{F}_q^* \} \]

So we get the **column multiplier** \( b \). If \( b \) is known, \( a \) can be computed by solving a linear system.
Filtration Attack for GRS codes

1. Suppose that $C_k = \text{GRS}_k(a, b)$ is known.

2. Shortening at the first position (i.e. $S_1(C_k)$) we get $C_0_k$ where $a_0 = (a_2, ..., a_n)$ and $b_0 = (b_0^2, ..., b_0^n)$ with $b_0^j = b^j(a_1 a_0)$.

3. We can build the filtration: $\text{GRS}_k(a, b) \Rightarrow \text{GRS}_k^1(a_0, b_0) \Rightarrow \text{GRS}_k^2(a_0, b_0) \Rightarrow \text{GRS}_1(a_0, b_0)$.

4. Second we get the column multiplier $b_0$ and the support $a_0$.

5. Repeat the process shortening in another position to recover $a$ completely.
Filtration Attack for GRS codes

1. Suppose that $C_k = \text{GRS}_k(a, b)$ is known.

2. **Shortening** at the first position (i.e. $S_1(C_k)$) we get $C_{k-1} = \text{GRS}_{k-1}(a', b')$ where

   $a' = (a_2, \ldots, a_n)$ and $b' = (b'_2, \ldots, b'_n)$ with $b'_j = b_j(a_j - a_1)$
Filtration Attack for GRS codes

1. Suppose that $C_k = \text{GRS}_k(a, b)$ is known.

2. **Shortening** at the first position (i.e. $S_1(C_k)$) we get $C'_{k-1} = \text{GRS}_{k-1}(a', b')$ where

   $$a' = (a_2, \ldots, a_n) \quad \text{and} \quad b' = (b'_2, \ldots, b'_n) \quad \text{with} \quad b'_j = b_j(a_j - a_1)$$

   It’s easy to get a generator matrix of $S_1(C_k)$

   $$G = \begin{pmatrix}
   1 & a_{12} & \ldots & a_{1n} \\
   0 & a_{22} & \ldots & a_{2n} \\
   \vdots & \vdots & \ddots & \vdots \\
   0 & a_{k2} & \ldots & a_{kn}
   \end{pmatrix}$$

   Generator matrix for $C_k$
Filtration Attack for GRS codes

1. Suppose that $C_k = \text{GRS}_k(a, b)$ is known.

2. **Shortening** at the first position (i.e. $S_1(C_k)$) we get $C_0 = \text{GRS}_{k-1}(a_0, b_0)$ where

   \[
   a' = (a_2, \ldots, a_n) \quad \text{and} \quad b' = (b'_2, \ldots, b'_n) \quad \text{with} \quad b'_j = b_j(a_j - a_1)
   \]

   It’s easy to get a generator matrix of $S_1(C_k)$:

   \[
   G = \begin{pmatrix}
   1 & a_{12} & \ldots & a_{1n} \\
   0 & a_{22} & \ldots & a_{2n} \\
   \vdots & \vdots & & \vdots \\
   0 & a_{k2} & \ldots & a_{kn}
   \end{pmatrix}
   \]
Filtration Attack for GRS codes

1. Suppose that \( C_k = \text{GRS}_k(a, b) \) is known.

2. Shortening at the first position (i.e. \( S_1(C_k) \)) we get \( C'_{k-1} = \text{GRS}_{k-1}(a', b') \)
   where
   \[
   a' = (a_2, \ldots, a_n) \quad \text{and} \quad b' = (b'_2, \ldots, b'_n) \quad \text{with} \quad b'_j = b_j(a_j - a_1)
   \]

3. We can build the filtration:
   \[
   \text{GRS}_k(a, b) \supseteq \text{GRS}_{k-1}(a', b') \supseteq \text{GRS}_{k-2}(a', b') \supseteq \text{GRS}_1(a', b')
   \]
Filtration Attack for GRS codes

1. Suppose that \( C_k = \text{GRS}_k(a, b) \) is known.

2. **Shortening** at the first position (i.e. \( S_1(C_k) \)) we get \( C_{k-1} = \text{GRS}_{k-1}(a', b') \)

   where

   \[
   a' = (a_2, \ldots, a_n) \quad \text{and} \quad b' = (b'_2, \ldots, b'_n) \quad \text{with} \quad b'_j = b_j(a_j - a_1)
   \]

3. We can build the filtration:

   \[
   \text{GRS}_k(a, b) \supseteq \text{GRS}_{k-1}(a', b') \supseteq \text{GRS}_{k-2}(a', b') \supseteq \text{GRS}_1(a', b')
   \]

4. Se we get the column **multiplier** \( b' \) and the **support** \( a' \).
Filtration Attack for GRS codes

1. Suppose that $C_k = \text{GRS}_k(a, b)$ is known.

2. **Shortening** at the first position (i.e. $S_1(C_k)$) we get

   $C_{k-1} = \text{GRS}_{k-1}(a', b')$

   where

   $a' = (a_2, \ldots, a_n)$ and $b' = (b'_2, \ldots, b'_n)$ with $b'_j = b_j(a_j - a_1)$

3. We can build the filtration:

   $$\text{GRS}_k(a, b) \supseteq \text{GRS}_{k-1}(a', b') \supseteq \text{GRS}_{k-2}(a', b') \supseteq \text{GRS}_1(a', b')$$

4. So we get the column **multiplier** $b'$ and the **support** $a'$.

5. Repeat the process **shortening** in another position to recover $a$ completely.
Filtration Attack for GRS codes

**Public Key:**
\[ \mathcal{K}_{\text{pub}} = \begin{cases} 
\text{a generator matrix of } C_{\text{pub}} = \text{GRS}_k(a, b) \\
\text{and } t = \left\lfloor \frac{d(C) - 1}{2} \right\rfloor 
\end{cases} \]

**The Algorithm:** Assume that \( 2k - 1 \leq n - 2 \)
Filtration Attack for GRS codes

Public Key: \( \mathcal{K}_{\text{pub}} = \begin{cases} \text{a generator matrix of } C_{\text{pub}} = \text{GRS}_k(a, b) \\ \text{and } t = \left\lceil \frac{d(C) - 1}{2} \right\rceil \end{cases} \)

The Algorithm: Assume that \( 2k - 1 \leq n - 2 \)

1. Determine the code

\[ S_1(C_{\text{pub}}) = \text{GRS}_{k-1}(a', b') \]
Filtration Attack for GRS codes

Public Key: \[ \mathcal{K}_{\text{pub}} = \begin{cases} \text{a generator matrix of } C_{\text{pub}} = \text{GRS}_k(a, b) \\ \text{and } t = \left\lfloor \frac{d(C) - 1}{2} \right\rfloor \end{cases} \]

The Algorithm: Assume that \( 2k - 1 \leq n - 2 \)

1. Determine the code

\[ S_1(C_{\text{pub}}) = \text{GRS}_{k-1}(a', b') \]

2. Build the filtration:

\[ C_{\text{pub}} \supseteq \text{GRS}_{k-1}(a', b') \supseteq \ldots \supseteq \text{GRS}_1(a', b') \]
Filtration Attack for GRS codes

Public Key: \( \mathcal{K}_{\text{pub}} = \begin{cases} \text{a generator matrix of } C_{\text{pub}} = \text{GRS}_k(a, b) \\ \text{and } t = \left\lceil \frac{d(c) - 1}{2} \right\rceil \end{cases} \)

The Algorithm: Assume that \( 2k - 1 \leq n - 2 \)

1. Determine the code

\[ S_1(C_{\text{pub}}) = \text{GRS}_{k - 1}(a', b') \]

2. Build the filtration:

\[ \underbrace{\text{GRS}_k(a, b)}_{C_{\text{pub}}} \supseteq \underbrace{\text{GRS}_{k - 1}(a', b')}_{S_1(C_{\text{pub}})} \supseteq \ldots \supseteq \text{GRS}_1(a', b') \]

3. Return \( b' \) and \( a' \)
Another (Filtration) Attack - Retrieving an ECP

**Proposition 1:**

Let $C = \text{GRS}_k(c, d)$, then $C = \text{GRS}_{n-k}(c, d^\perp)$ is an $t$-ECP for $C$ over $\mathbb{F}_q$.

**Proposition 2:**

To compute a $t$-ECP for $C = \text{GRS}_k(a, b)$, it suffices to compute a code of type $B = \text{GRS}_t(c, d^\perp)$ if we know $C = \text{GRS}_k(c, d)$ and $B = \text{GRS}_t(c, d^\perp)$. 

Then, $A = \text{GRS}_t+1(a, 1) = \downarrow^t B \uparrow C$.
Another (Filtration) Attack - Retrieving an ECP

**Proposition 1:**

Let\( \mathcal{C} = \text{GRS}_k(c, d) \implies \mathcal{C}^\perp = \text{GRS}_{n-k}(c, d^\perp) \)

Then, \( \mathcal{A} = \text{GRS}_{t+1}(c, 1) \) and \( \mathcal{B} = \text{GRS}_t(c, d^\perp) \) is a \( t \)-ECP for \( \mathcal{C} \) over \( \mathbb{F}_q \).
Another (Filtration) Attack - Retrieving an ECP

**Proposition 1:**

Let \( C = \text{GRS}_k(c, d) \implies C^\perp = \text{GRS}_{n-k}(c, d^\perp) \)

Then, \( A = \text{GRS}_{t+1}(c, 1) \) and \( B = \text{GRS}_t(c, d^\perp) \)

is a \( t \)-ECP for \( C \) over \( \mathbb{F}_q \).

**Proposition 2:** To compute a \( t \)-ECP for \( C = \text{GRS}_k(a, b) \) it suffices to compute a code of type \( B = \text{GRS}_t(c, d^\perp) \).

If we know \( C = \text{GRS}_k(c, d) \) and \( B = \text{GRS}_t(c, d^\perp) \)
Another (Filtration) Attack - Retrieving an ECP

Proposition 1:
Let \( C = \text{GRS}_k(c, d) \) \( \implies \) \( C^\perp = \text{GRS}_{n-k}(c, d^\perp) \)

Then, \( A = \text{GRS}_{t+1}(c, 1) \) and \( B = \text{GRS}_t(c, d^\perp) \)

is a \( t \)-ECP for \( C \) over \( \mathbb{F}_q \)

Proposition 2: To compute a \( t \)-ECP for \( C = \text{GRS}_k(a, b) \) it suffice to compute a code of type \( B = \text{GRS}_t(c, d^\perp) \)

If we know \( C = \text{GRS}_k(c, d) \) and \( B = \text{GRS}_t(c, d^\perp) \)

Then, \( A = \text{GRS}_{t+1}(a, 1) = (B \ast C)^\perp \)
Another (Filtration) Attack - Retrieving an ECP

Public Key: \[ \mathcal{K}_{\text{pub}} = \begin{cases} \text{a generator matrix of } C_{\text{pub}} = \text{GRS}_k(a, b) \\ \text{and } t = \left\lfloor \frac{d(c)-1}{2} \right\rfloor \end{cases} \]

The Algorithm: Assume that \(2k - 1 \leq n - 2\)
Another (Filtration) Attack - Retrieving an ECP

Public Key:  \[ \mathcal{K}_{\text{pub}} = \left\{ \begin{array}{l}
\text{a generator matrix of } C_{\text{pub}} = \text{GRS}_k(a, b) \\
\text{and } t = \left\lfloor \frac{d(C) - 1}{2} \right\rfloor
\end{array} \right. \]

The Algorithm: Assume that \(2k - 1 \leq n - 2\)

1. Determine the codes

\[ C_{\text{pub}}^\perp = \text{GRS}_{2t}(a, b^\perp) \quad \text{and} \quad S_1(C_{\text{pub}}^\perp) = \text{GRS}_{2t-1}(a', \sim b^\perp) \]
Another (Filtration) Attack - Retrieving an ECP

**Public Key:**
\[ \mathcal{K}_{pub} = \begin{cases} 
\text{a generator matrix of } C_{pub} = \text{GRS}_k(a, b) \\
\text{and } t = \left\lfloor \frac{d(C)-1}{2} \right\rfloor
\end{cases} \]

**The Algorithm:**
Assume that \(2k - 1 \leq n - 2\)

1. Determine the codes
\[ C_{pub}^\perp = \text{GRS}_{2t}(a, b^\perp) \quad \text{and} \quad S_1(C_{pub}^\perp) = \text{GRS}_{2t-1}(a', \sim b^\perp) \]

2. Build the filtration:
\[ C_{pub}^\perp \supseteq \text{GRS}_{2t-1}(a', \sim b^\perp) \supseteq \ldots \supseteq \text{GRS}_t(a', \sim b^\perp) \]
Another (Filtration) Attack - Retrieving an ECP

Public Key: $\mathcal{K}_{\text{pub}} = \left\{ \begin{array}{l}
\text{a generator matrix of } C_{\text{pub}} = \text{GRS}_k(a, b) \\
\text{and } t = \left\lfloor \frac{d(C) - 1}{2} \right\rfloor
\end{array} \right.$

The Algorithm: Assume that $2k - 1 \leq n - 2$

1. Determine the codes

$$C_{\text{pub}}^\perp = \text{GRS}_{2t}(a, b^\perp)$$ and $$S_1(C_{\text{pub}}^\perp) = \text{GRS}_{2t-1}(a', \sim b^\perp)$$

2. Build the filtration:

$$ \underbrace{\text{GRS}_{2t}(a, b^\perp)}_{C_{\text{pub}}^\perp} \supseteq \underbrace{\text{GRS}_{2t-1}(a', \sim b^\perp)}_{S_1(C_{\text{pub}}^\perp)} \supseteq \ldots \supseteq \underbrace{\text{GRS}_t(a', \sim b^\perp)}_{B} $$

3. Return $(\mathcal{A}, \mathcal{B})$ which is an $ECP$ for $S_1(C)$ where: $\mathcal{A} = (\mathcal{B} \ast S_1(C))^\perp$
Another (Filtration) Attack - Retrieving an ECP

Public Key: \( \mathcal{K}_{\text{pub}} = \begin{cases} 
\text{a generator matrix of } C_{\text{pub}} = \text{GRS}_k(a, b) \\
\text{and } t = \left\lfloor \frac{d(C)-1}{2} \right\rfloor 
\end{cases} \)

The Algorithm: Assume that \( 2k - 1 \leq n - 2 \)

1. Determine the codes

\[ C_{\text{pub}}^\perp = \text{GRS}_{2t}(a, b^\perp) \quad \text{and} \quad S_1(C_{\text{pub}}^\perp) = \text{GRS}_{2t-1}(a', \sim b^\perp) \]

2. Build the filtration:

\[ \text{GRS}_{2t}(a, b^\perp) \supseteq \text{GRS}_{2t-1}(a', \sim b^\perp) \supseteq \ldots \supseteq \text{GRS}_t(a', \sim b^\perp) \]

\[ C_{\text{pub}}^\perp \supseteq S_1(C_{\text{pub}}^\perp) \supseteq B \]

3. Return \((A, B)\) which is an ECP for \( S_1(C) \) where: \( A = (B \ast S_1(C))^\perp \)

4. Note that: Correcting an error in the first position is not a difficult problem.
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