# Recovering short secret keys of RLCE KEM in polynomial time

GT "Butte aux Cailles", January 17, 2019

---

Alain Couvreur[1], Matthieu Lequesne[2,3] and Jean-Pierre Tillich[3]

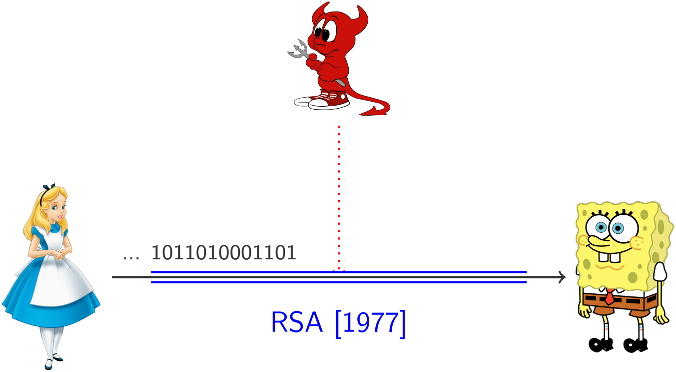
1 - Inria Saclay – team Grace, École polytechnique

2 - Sorbonne Université Paris

3 - Inria Paris – team Secret

# Context

# Public Key Cryptography



# Public Key Cryptography

 +  + $i\hbar \frac{\partial}{\partial t}\Psi = \hat{H}\Psi$

[1994]

… 1011010001101

~~RSA [1977]~~

???

# Public Key Cryptography

[1994]

+

+

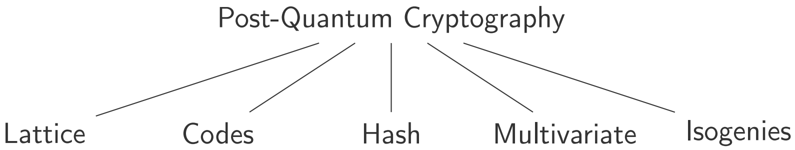
$$i\hbar \frac{\partial}{\partial t} \Psi = \hat{H} \Psi$$

… 1011010001101

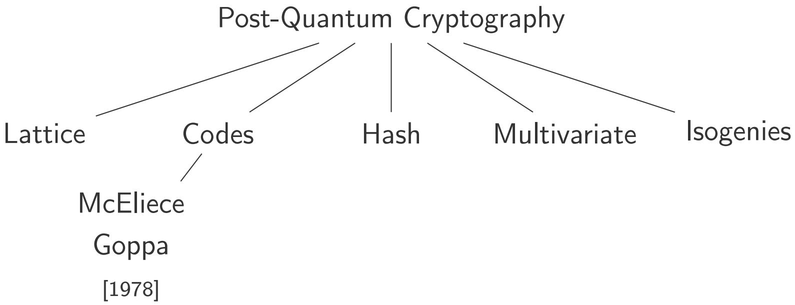
~~RSA [1977]~~

???

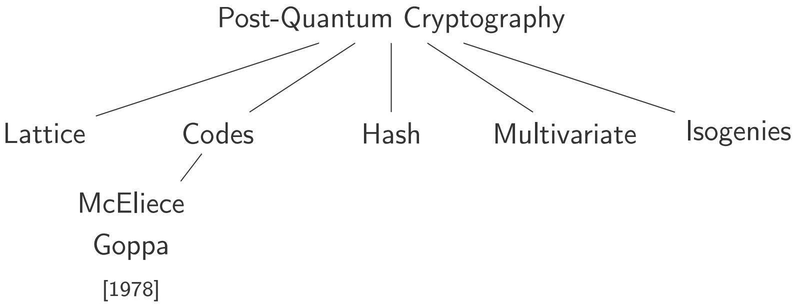
# Post-Quantum Cryptography



# Post-Quantum Cryptography

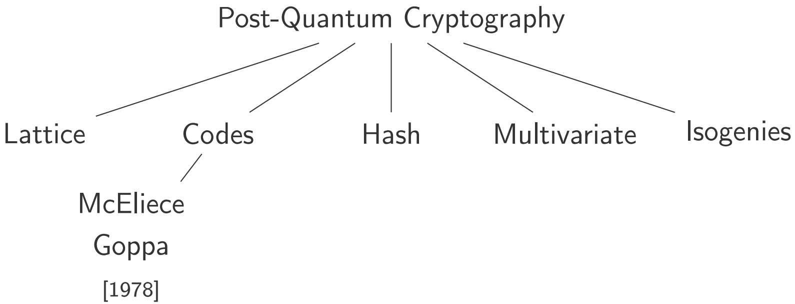


# Post-Quantum Cryptography




**Code-based cryptosystem** (à la McEliece)

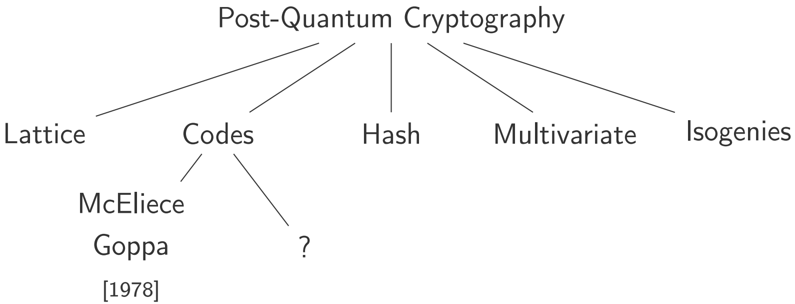
# Post-Quantum Cryptography




**Code-based cryptosystem** (à la McEliece)

**Main goal:** achieve relatively short keys

# Post-Quantum Cryptography




**Code-based cryptosystem** (à la McEliece)

**Main goal:** achieve relatively short keys

# Code Based Cryptography

To instanciate a secure code-based cryptographic scheme, one needs a family of codes such that:

# Code Based Cryptography

To instanciate a secure code-based cryptographic scheme, one needs a family of codes such that:

1. there exists good decoding algorithm ;

# Code Based Cryptography

To instanciate a secure code-based cryptographic scheme, one needs a family of codes such that:

1. there exists good decoding algorithm ;
2. the randomized version of the code is indistinguishable from a random code;
   - $\rightarrow$ key security

# Code Based Cryptography

To instanciate a secure code-based cryptographic scheme, one needs a family of codes such that:

1. there exists good decoding algorithm ;
2. the randomized version of the code is indistinguishable from a random code;
   $\rightarrow$ key security
3. it is computationaly hard to correct the errors whithout knowing of the structure of the code (message security).
   $\rightarrow$ message security

# The RLCE Scheme

# Context: RLCE

- NIST call for post-quantum cryptography standardization;
- Key Encapsulation Mechanism;
- Proposed by Yonggee Wang (UNC Charlotte);

# Context: RLCE

- NIST call for post-quantum cryptography standardization;
- Key Encapsulation Mechanism;
- Proposed by Yonggee Wang (UNC Charlotte);

- **Code-based cryptosystem** (à la McEliece);
- **Idea:** mix a GRS code with random columns.

# Generalised Reed Solomon codes

## Definition (Generalised Reed Solomon codes)

The generalised Reed–Solomon (GRS) code with support $\boldsymbol{x}$ and multiplier $\boldsymbol{y}$ of dimension $k$ is defined as

$$\mathbf{GRS}_k(\boldsymbol{x}, \boldsymbol{y}) \stackrel{\text{def}}{=} \{(y_1 f(x_1), \ldots, y_n f(x_n)) \mid f \in \mathbb{F}_q[x]_{<k}\}.$$

# Generalised Reed Solomon codes

**Definition (Generalised Reed Solomon codes)**

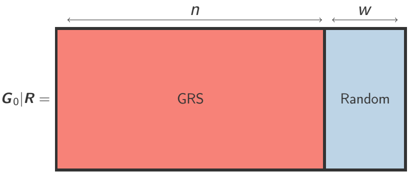
The generalised Reed–Solomon (GRS) code with support $\boldsymbol{x}$ and multiplier $\boldsymbol{y}$ of dimension $k$ is defined as

$$\mathbf{GRS}_k(\boldsymbol{x}, \boldsymbol{y}) \stackrel{\text{def}}{=} \{(y_1 f(x_1), \ldots, y_n f(x_n)) \mid f \in \mathbb{F}_q[x]_{<k}\}.$$

**Sidelnikov Shestakov (1992)**

Given a generator matrix of a GRS code $\mathscr{C}$, it is possible to find $\boldsymbol{x}$ and $\boldsymbol{y}$ such that $\mathscr{C} = \mathbf{GRS}_k(\boldsymbol{x}, \boldsymbol{y})$.

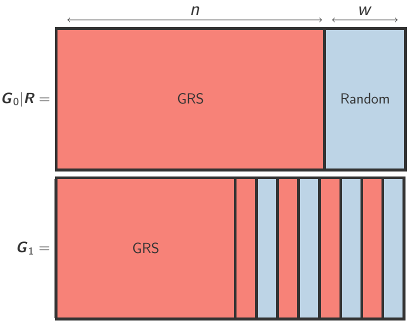
# The Scheme




$$\boldsymbol{G}_0 \leftarrow GRS(n, k)$$

$$\boldsymbol{R} \leftarrow \mathbb{F}_q^{k \times w}$$

# The Scheme

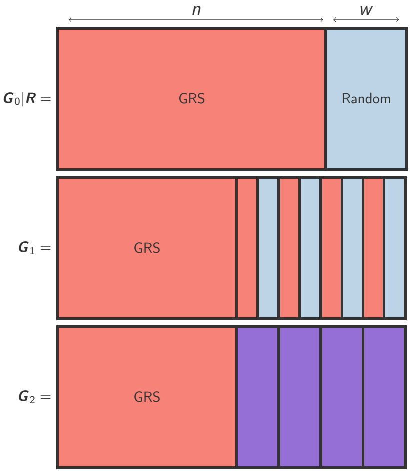



$$\boldsymbol{G}_0 \leftarrow GRS(n, k)$$

$$\boldsymbol{R} \leftarrow \mathbb{F}_q^{k \times w}$$

$$\boldsymbol{G}_1 \stackrel{\text{def}}{=} \mathrm{mix}(\boldsymbol{G}_0, \boldsymbol{R})$$

# The Scheme




$$G_0 \leftarrow GRS(n, k)$$

$$R \leftarrow \mathbb{F}_q^{k \times w}$$

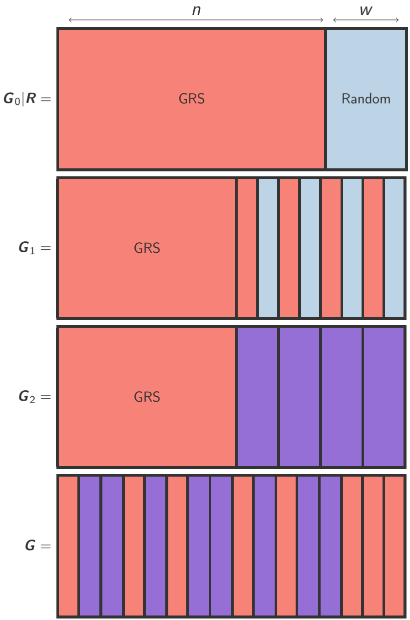
$$G_1 \stackrel{\text{def}}{=} \mathrm{mix}(G_0, R)$$

$$A_i \leftarrow \mathbb{F}_q^{2 \times 2}$$

$$A \stackrel{\text{def}}{=} \begin{pmatrix} I_{n-w} & & & (0) \\ & A_1 & & \\ & & \ddots & \\ (0) & & & A_w \end{pmatrix}$$

$$G_2 \stackrel{\text{def}}{=} G_1 A$$

# The Scheme




$$G_0 \leftarrow GRS(n, k)$$

$$R \leftarrow \mathbb{F}_q^{k \times w}$$

$$G_1 \stackrel{\text{def}}{=} \mathrm{mix}(G_0, R)$$

$$A_i \leftarrow \mathbb{F}_q^{2 \times 2}$$

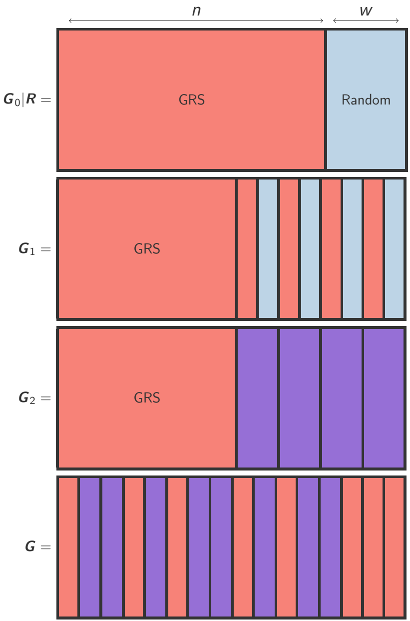
$$A \stackrel{\text{def}}{=} \begin{pmatrix} I_{n-w} & & & (0) \\ & A_1 & & \\ & & \ddots & \\ (0) & & & A_w \end{pmatrix}$$

$$G_2 \stackrel{\text{def}}{=} G_1 A$$

$$P \leftarrow \mathfrak{S}_{n+w}$$

$$G \stackrel{\text{def}}{=} G_2 P$$

# The Scheme

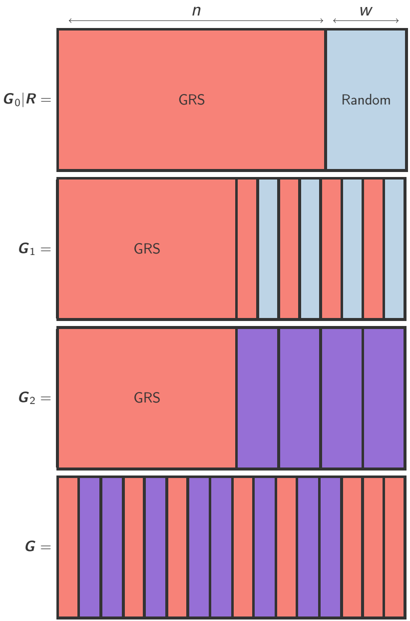



**Attacker's point of view:**

Given $\boldsymbol{G}$:

- which column is ? $\rightarrow$ *GRS*
- which column is ? $\rightarrow$ *PR*

# The Scheme




**Attacker's point of view:**

Given $\boldsymbol{G}$:

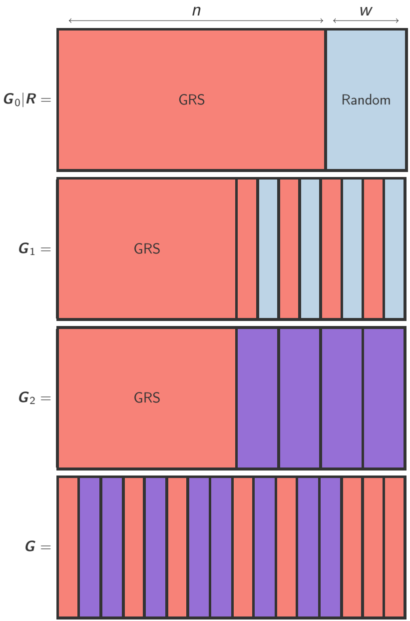
- which column is [red] ? $\rightarrow$ *GRS*
- which column is [purple] ? $\rightarrow$ *PR*

**RLCE-short** *vs.* **RLCE-long**:

Depends on the size of $w$.

- **RLCE-short**: $w \approx \frac{n-k}{2}$ ;
- **RLCE-long**: $w = n - k$.

# The Scheme




**Attacker's point of view:**

Given $\boldsymbol{G}$:

- which column is [red] ? $\rightarrow$ *GRS*
- which column is [purple] ? $\rightarrow$ *PR*

**RLCE-short** *vs.* **RLCE-long**:

Depends on the size of $w$.

- **RLCE-short**: $w \approx \frac{n-k}{2}$ ;
- **RLCE-long**: $w = n - k$.

**Purpose of this talk :**

Understand why we manage to break **RLCE-short** but not **RLCE-long**.

# Parameters

Table: Set of parameters for **RLCE-short**.

| Claimed security | $n$ | $k$ | $t$ | $w$ | $q$ | Public key size (kB) |
|---|---|---|---|---|---|---|
| 128 | 532 | 376 | 78 | 96 | $2^{10}$ | 118 |
| 192 | 846 | 618 | 114 | 144 | $2^{10}$ | 287 |
| 256 | 1160 | 700 | 230 | 311 | $2^{11}$ | 742 |

Table: Set of parameters for **RLCE-long**.

| Claimed security | $n$ | $k$ | $t$ | $w$ | $q$ | Public key size (kB) |
|---|---|---|---|---|---|---|
| 128 | 630 | 470 | 80 | 160 | $2^{10}$ | 188 |
| 192 | 1000 | 764 | 118 | 236 | $2^{10}$ | 450 |
| 256 | 1360 | 800 | 280 | 560 | $2^{11}$ | 1232 |

# The Tools

# The Tools: Schur product

**Definition (Schur product)**

Schur product of vectors: $\boldsymbol{a} \star \boldsymbol{b} \stackrel{\text{def}}{=} (a_1 b_1, \ldots, a_n b_n).$

# The Tools: Schur product

### Definition (Schur product)

Schur product of vectors: $\boldsymbol{a} \star \boldsymbol{b} \stackrel{\text{def}}{=} (a_1 b_1, \ldots, a_n b_n)$.

Schur product of codes:

$$\mathcal{A} \star \mathcal{B} \stackrel{\text{def}}{=} \mathbf{Span}_{\mathbb{F}_q} \{\boldsymbol{a} \star \boldsymbol{b} \mid \boldsymbol{a} \in \mathcal{A},\ \boldsymbol{b} \in \mathcal{B}\}.$$

# The Tools: Schur product

**Definition (Schur product)**

Schur product of vectors: $\boldsymbol{a} \star \boldsymbol{b} \stackrel{\text{def}}{=} (a_1 b_1, \ldots, a_n b_n)$.

Schur product of codes:

$$\mathcal{A} \star \mathcal{B} \stackrel{\text{def}}{=} \mathbf{Span}_{\mathbb{F}_q} \left\{ \boldsymbol{a} \star \boldsymbol{b} \mid \boldsymbol{a} \in \mathcal{A},\ \boldsymbol{b} \in \mathcal{B} \right\}.$$

**Notation:** $\mathscr{C}^{\star 2} \stackrel{\text{def}}{=} \mathscr{C} \star \mathscr{C}$.

# The Tools: Square-code Distinguisher

**Question**

Given a code $\mathscr{C}$ of dimension $k$, what is the value of $\dim \mathscr{C}^{\star 2}$?

# The Tools: Square-code Distinguisher

### Question

Given a code $\mathscr{C}$ of dimension $k$, what is the value of $\dim \mathscr{C}^{\star 2}$?

### Square-code Distinguisher

$$\mathscr{C} \text{ random} \qquad \Rightarrow \quad \dim \mathscr{C}^{\star 2} = \binom{k+1}{2} = \frac{k(k+1)}{2}.$$

# The Tools: Square-code Distinguisher

### Question

Given a code $\mathscr{C}$ of dimension $k$, what is the value of $\dim \mathscr{C}^{\star 2}$?

### Square-code Distinguisher

$$\mathscr{C} \text{ random} \quad \Rightarrow \quad \dim \mathscr{C}^{\star 2} = \binom{k+1}{2} = \frac{k(k+1)}{2}.$$

$$\mathscr{C} = \mathbf{GRS}_k(\boldsymbol{x}, \boldsymbol{y}) \quad \Rightarrow \quad \dim \mathscr{C}^{\star 2} = 2k - 1.$$

# The Tools: Square-code Distinguisher

**Proof.**

Let $\boldsymbol{c}$ and $\boldsymbol{c}' \in \mathbf{GRS}_k(\boldsymbol{x}, \boldsymbol{y})$.

$$\boldsymbol{c} = (y_1 p(x_1), \ldots, y_n p(x_n)), \quad \boldsymbol{c}' = (y_1 q(x_1), \ldots, y_n q(x_n))$$

where $p$ and $q$ are two polynomials of degree at most $k-1$.

# The Tools: Square-code Distinguisher

Proof.

Let $\boldsymbol{c}$ and $\boldsymbol{c}' \in \mathbf{GRS}_k(\boldsymbol{x}, \boldsymbol{y})$.

$$\boldsymbol{c} = (y_1 p(x_1), \ldots, y_n p(x_n)), \quad \boldsymbol{c}' = (y_1 q(x_1), \ldots, y_n q(x_n))$$

where $p$ and $q$ are two polynomials of degree at most $k-1$.

$$\begin{aligned} \boldsymbol{c} \star \boldsymbol{c}' &= y_1^2 p(x_1) q(x_1), \ldots, y_n^2 p(x_n) q(x_n) \\ &= y_1^2 r(x_1), \ldots, y_n^2 r(x_n). \end{aligned}$$

where $r$ is a polynomial of degree at most $2k-2$.

# The Tools: Square-code Distinguisher

**Proof.**

Let $\boldsymbol{c}$ and $\boldsymbol{c}' \in \mathbf{GRS}_k(\boldsymbol{x}, \boldsymbol{y})$.

$$\boldsymbol{c} = (y_1 p(x_1), \dots, y_n p(x_n)), \quad \boldsymbol{c}' = (y_1 q(x_1), \dots, y_n q(x_n))$$

where $p$ and $q$ are two polynomials of degree at most $k-1$.

$$\begin{aligned} \boldsymbol{c} \star \boldsymbol{c}' &= y_1^2 p(x_1) q(x_1), \dots, y_n^2 p(x_n) q(x_n) \\ &= y_1^2 r(x_1), \dots, y_n^2 r(x_n). \end{aligned}$$

where $r$ is a polynomial of degree at most $2k-2$.

Hence,

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

□

# The Tools: Square-code Distinguisher

## Square-code Distinguisher

$\mathscr{C}$ a code of length $n$ and dimension $k$.

$$\mathscr{C} \text{ random} \quad \Rightarrow \quad \dim \mathscr{C}^{\star 2} = \frac{k(k+1)}{2}.$$

$$\mathscr{C} = \mathbf{GRS}_k(\boldsymbol{x}, \boldsymbol{y}) \quad \Rightarrow \quad \dim \mathscr{C}^{\star 2} = 2k - 1.$$

# The Tools: Square-code Distinguisher

## Square-code Distinguisher

$\mathscr{C}$ a code of length $n$ and dimension $k$. DIMENSION $\leqslant$ LENGTH.

$$\mathscr{C} \text{ random} \quad \Rightarrow \quad \dim \mathscr{C}^{\star 2} = \min\left(\frac{k(k+1)}{2}, n\right).$$

$$\mathscr{C} = \mathbf{GRS}_k(\boldsymbol{x}, \boldsymbol{y}) \quad \Rightarrow \quad \dim \mathscr{C}^{\star 2} = \min(2k-1, n).$$

# The Tools: Square-code Distinguisher

## Square-code Distinguisher

$\mathscr{C}$ a code of length $n$ and dimension $k$. DIMENSION $\leqslant$ LENGTH.

$$\mathscr{C} \text{ random} \quad \Rightarrow \quad \dim \mathscr{C}^{\star 2} = \min\left(\frac{k(k+1)}{2}, n\right).$$

$$\mathscr{C} = \mathbf{GRS}_k(\boldsymbol{x}, \boldsymbol{y}) \quad \Rightarrow \quad \dim \mathscr{C}^{\star 2} = \min(2k-1, n).$$

Distinguisher works if: $\begin{cases} \dim \mathscr{C}^{\star 2} < \frac{k(k+1)}{2}, \\ \dim \mathscr{C}^{\star 2} < n. \end{cases}$

# The Tools: Square-code Distinguisher

How to reach the parameter range where the distinguisher works?

# The Tools: Square-code Distinguisher

How to reach the parameter range where the distinguisher works?

How to reach the parameter range where the distinguisher works?

# The Tools: Square-code Distinguisher

How to reach the parameter range where the distinguisher works?

# The Tools: Punctured and Shortened Codes

**Definition (punctured code)**

Let $\mathscr{C} \subseteq \mathbb{F}_q^n$ and $j \in [\![1, n]\!]$.

$$\mathcal{P}_{\{j\}}(\mathscr{C}) \stackrel{\text{def}}{=} \{(c_i)_{i \in [\![1,n]\!], i \neq j} \text{ s.t. } \boldsymbol{c} \in \mathscr{C}\}.$$

# The Tools: Punctured and Shortened Codes

### Definition (punctured code)

Let $\mathscr{C} \subseteq \mathbb{F}_q^n$ and $j \in [\![1, n]\!]$.

$$\mathcal{P}_{\{j\}}(\mathscr{C}) \stackrel{\text{def}}{=} \{(c_i)_{i \in [\![1,n]\!], i \neq j} \text{ s.t. } \boldsymbol{c} \in \mathscr{C}\}.$$

### Definition (shortened code)

Let $\mathscr{C} \subseteq \mathbb{F}_q^n$ and $j \in [\![1, n]\!]$.

$$\mathcal{S}_{\{j\}}(\mathscr{C}) \stackrel{\text{def}}{=} \mathcal{P}_{\{j\}}(\{\boldsymbol{c} \in \mathscr{C} \text{ s.t. } c_j = 0\}).$$

# The Tools: Punctured and Shortened Codes

For $\mathscr{C}$ a **random** code of dimension $k$ and length $n$:

| $\mathscr{C}$ **random** | | $\mathscr{C}' = \mathcal{S}(\mathscr{C})$ |
|---|---|---|
| length $= n$ | $\longrightarrow$ | length $n' = n - 1$ |
| dimension $= k$ | | dimension $k' = k - 1$ |

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

# The Tools: Punctured and Shortened Codes

For $\mathscr{C} = \mathbf{GRS}_k(\boldsymbol{x}, \boldsymbol{y})$:

$$\begin{array}{ccc} \mathscr{C} & & \mathscr{C}' = \mathcal{S}(\mathscr{C}) \\ \text{length } = n & \longrightarrow & \text{length } n' = n - 1 \\ \text{dimension } = k & & \text{dimension } k' = k - 1 \end{array}$$

$$\dim \mathscr{C}^{\star 2} = \min(2k - 1, n).$$

# The Tools: Punctured and Shortened Codes

For $\mathscr{C} = \mathbf{GRS}_k(\boldsymbol{x}, \boldsymbol{y})$:

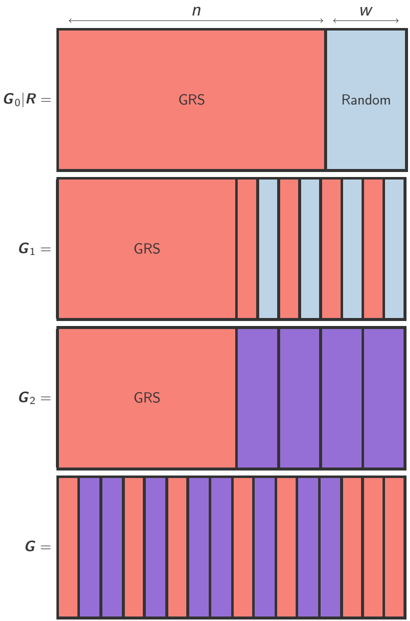
| $\mathscr{C}$ | | $\mathscr{C}' = \mathcal{S}(\mathscr{C})$ |
|---|---|---|
| length $= n$ | $\longrightarrow$ | length $n' = n - 1$ |
| dimension $= k$ | | dimension $k' = k - 1$ |

$$\dim \mathscr{C}^{\star 2} = \min(2k - 1, n).$$

**Repeat** until $\dim \mathscr{C}^{\star 2} < n$.

# A Distinguisher on RLCE

# The Scheme

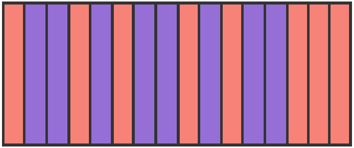



**Attacker's point of view:**

Given $\boldsymbol{G}$:

- which column is  ? $\longrightarrow$ *GRS*
- which column is  ? $\longrightarrow$ *PR*

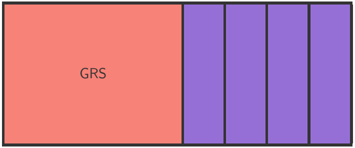
# The Attack



+

= ?

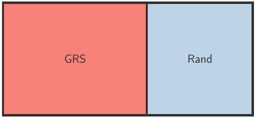
# The Attack

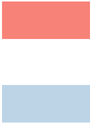



+

= ?

# Lemma : GRS+Rand

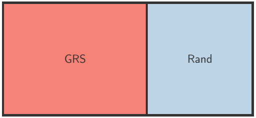
$\mathscr{A} \stackrel{\text{def}}{=}$ 

subcode of length $n$
of a GRS code
of dimension $k_{GRS}$

random code of length $r$

# Lemma : GRS+Rand

$\mathscr{A} \stackrel{\text{def}}{=}$ 

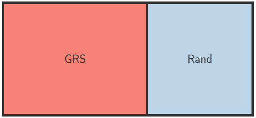

- subcode of length $n$ of a GRS code of dimension $k_{GRS}$
- random code of length $r$

## Lemma

$$\dim \mathscr{A}^{\star 2} \leqslant 2k_{GRS} + r - 1.$$

# Lemma : GRS+Rand

$\mathscr{A} \stackrel{\text{def}}{=}$




subcode of length $n$ of a GRS code of dimension $k_{GRS}$

random code of length $r$

## Lemma

$$\dim \mathscr{A}^{\star 2} \leqslant 2k_{GRS} + r - 1.$$

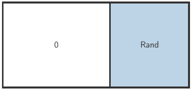
*If the equality holds, then for every* $i \in [\![n+1, n+w]\!]$*:*

$$\dim \mathcal{P}_{\{i\}}\left(\mathscr{A}^{\star 2}\right) = \dim \mathscr{A}^{\star 2} - 1.$$

# Lemma : GRS+Rand (Proof)

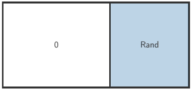
$\mathscr{A}_{GRS} \stackrel{\text{def}}{=}$


$\mathscr{A}_{Rand} \stackrel{\text{def}}{=}$



# Lemma : GRS+Rand (Proof)

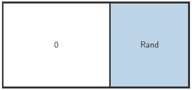
$\mathscr{A}_{GRS} \stackrel{\text{def}}{=}$ 


$\mathscr{A}_{Rand} \stackrel{\text{def}}{=}$ 


$$\mathscr{A} \subseteq \mathscr{A}_{GRS} + \mathscr{A}_{Rand}$$

# Lemma : GRS+Rand (Proof)

$\mathscr{A}_{GRS} \stackrel{\text{def}}{=}$ 

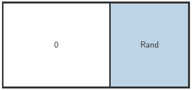

$\mathscr{A}_{Rand} \stackrel{\text{def}}{=}$ 


$$\mathscr{A} \subseteq \mathscr{A}_{GRS} + \mathscr{A}_{Rand}$$

$$\begin{aligned}
\mathscr{A}^{\star 2} &\subseteq \left(\mathscr{A}_{GRS} + \mathscr{A}_{Rand}\right)^{\star 2} \\
&\subseteq \mathscr{A}_{GRS}^{\star 2} + \mathscr{A}_{Rand}^{\star 2} + \mathscr{A}_{GRS} \star \mathscr{A}_{Rand} \\
&\subseteq \mathscr{A}_{GRS}^{\star 2} + \mathscr{A}_{Rand}^{\star 2}
\end{aligned}$$

# Lemma : GRS+Rand (Proof)

$\mathscr{A}_{GRS} \stackrel{\text{def}}{=}$ 


$\mathscr{A}_{Rand} \stackrel{\text{def}}{=}$ 


$$\mathscr{A} \subseteq \mathscr{A}_{GRS} + \mathscr{A}_{Rand}$$

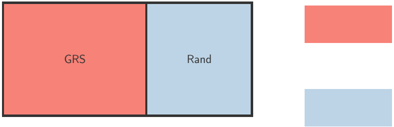
$$\begin{aligned}
\mathscr{A}^{\star 2} &\subseteq (\mathscr{A}_{GRS} + \mathscr{A}_{Rand})^{\star 2} \\
&\subseteq \mathscr{A}_{GRS}^{\star 2} + \mathscr{A}_{Rand}^{\star 2} + \mathscr{A}_{GRS} \star \mathscr{A}_{Rand} \\
&\subseteq \mathscr{A}_{GRS}^{\star 2} + \mathscr{A}_{Rand}^{\star 2}
\end{aligned}$$

$$\begin{aligned}
\dim \mathscr{A}^{\star 2} &\leqslant \dim \mathscr{A}_{GRS}^{\star 2} + \dim \mathscr{A}_{Rand}^{\star 2} \\
&\leqslant 2k_{GRS} - 1 + r
\end{aligned}$$

$\square$

# Lemma : GRS+Rand

$\mathscr{A} \stackrel{\text{def}}{=}$




subcode of length $n$ of a GRS code of dimension $k_{GRS}$

random code of length $r$

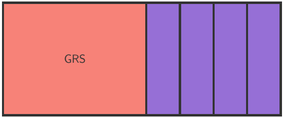
## Lemma

$$\dim \mathscr{A}^{\star 2} \leqslant 2k_{GRS} + r - 1.$$

*If the equality holds, then for every* $i \in [\![n+1, n+w]\!]$*:*

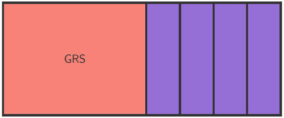
$$\dim \mathcal{P}_{\{i\}} \left(\mathscr{A}^{\star 2}\right) = \dim \mathscr{A}^{\star 2} - 1.$$

# Shortening RLCE

$\mathscr{C} \stackrel{\text{def}}{=}$ 

# Shortening RLCE

$\mathscr{C} \stackrel{\text{def}}{=}$

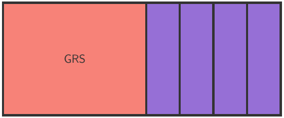



**Theorem**

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

# Shortening RLCE

$\mathscr{C}' \stackrel{\text{def}}{=}$




$+ \quad 1 \times$

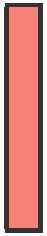
.

## Theorem

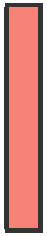
$$\dim \mathscr{C}'^{\star 2} = \min\left(2(k+w-1)-1,\ n+w-1\right).$$

# Proof

# Proof

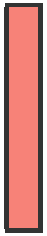
- Case 1:  $+$ 

# Proof

- Case 1:  + 

$$k + w$$
$$\downarrow$$
$$(k - 1) + w$$

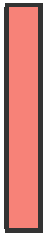
# Proof

- Case 1:  $+$ 

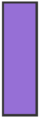
$$k + w$$
$$\downarrow$$
$$(k-1) + w$$

- Case 2:  $+$ 

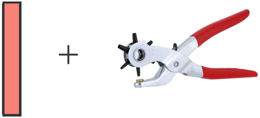
# Proof

- Case 1:  + 

$$\begin{array}{c} k + w \\ \downarrow \\ (k-1) + w \end{array}$$

- Case 2:  + 

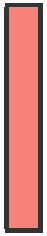
# Proof

- Case 1: 

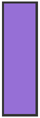
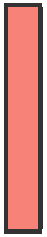
$$k + w \downarrow (k-1) + w$$

- Case 2: 

# Proof

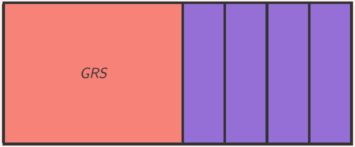
- Case 1:  + 

$$k + w$$
$$\downarrow$$
$$(k-1) + w$$

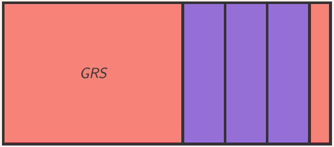
- Case 2:  +  =  **Why?**

# Derandomization Lemma

## Lemma




↓ *shortening one PR column* ↓



# Derandomization Lemma

## Proof.

By construction, there is

- a polynomial $f \in \mathbb{F}_q[x]_{<k}$ (the GRS part) ;
- a linear form $\psi$ (the random part) ;
- elements $a, b, c, d \in \mathbb{F}_q$ (the mixing)

such that, at position $i$, for any $\boldsymbol{c} \in \mathscr{C}$, we have

$$c_i = a \cdot y_j f(x_j) + c \cdot \psi(f),$$

$$c_{\tau(i)} = b \cdot y_j f(x_j) + d \cdot \psi(f).$$

# Derandomization Lemma

**Proof.**

$$c_i = a \cdot y_j f(x_j) + c \cdot \psi(f),$$

$$c_{\tau(i)} = b \cdot y_j f(x_j) + d \cdot \psi(f).$$

# Derandomization Lemma

## Proof.

$$c_i = a \cdot y_j f(x_j) + c \cdot \psi(f),$$

$$c_{\tau(i)} = b \cdot y_j f(x_j) + d \cdot \psi(f).$$

Shortening in $i \quad \Leftrightarrow \quad f \in \mathbb{F}_q[x]_{<k}$ s.t. $c_i = 0$.

*i.e.* $\psi(f) = -c^{-1} a y_j f(x_j)$.

# Derandomization Lemma

## Proof.

$$c_i = a \cdot y_j f(x_j) + c \cdot \psi(f),$$

$$c_{\tau(i)} = b \cdot y_j f(x_j) + d \cdot \psi(f).$$

$$\text{Shortening in } i \quad \Leftrightarrow \quad f \in \mathbb{F}_q[x]_{<k} \text{ s.t. } c_i = 0.$$

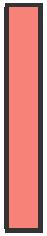
$$\textit{i.e. } \psi(f) = -c^{-1} a y_j f(x_j).$$

Therefore, for any $\boldsymbol{c} \in \mathcal{S}_{\{i\}}(\mathscr{C})$, we have

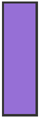
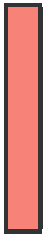
$$c_{\tau(i)} = (b - dac^{-1}) y_j f(x_j).$$

□

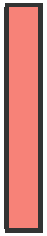
# Proof

- Case 1:  + 

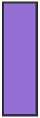
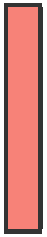
$$k + w$$
$$\downarrow$$
$$(k - 1) + w$$

- Case 2:  +  =  **Why?**

# Proof

- Case 1:  + 

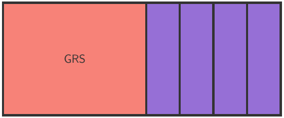
$$k + w \\ \downarrow \\ (k-1) + w$$

- Case 2:  +  = 

$$k + w \\ \downarrow \\ (k+1) + (w-2)$$

# Shortening RLCE

$\mathscr{C}' \stackrel{\text{def}}{=}$




$+ \quad 1 \times$

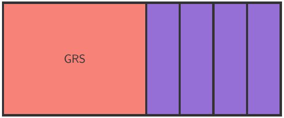
.

## Theorem

$$\dim \mathscr{C}'^{\star 2} = \min\left(2(k+w-1)-1,\ n+w-1\right).$$

# Shortening RLCE

$\mathscr{C}' \stackrel{\text{def}}{=}$



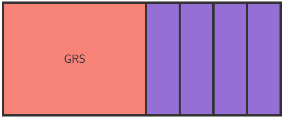

$+ \quad \ell \times$

.

**Theorem**

$$\dim \mathscr{C}'^{\star 2} = \min\left( 2(k + w - \ell) - 1 \,,\; n + w - \ell \right).$$

# Shortening RLCE

$\mathscr{C}' \stackrel{\text{def}}{=}$ 

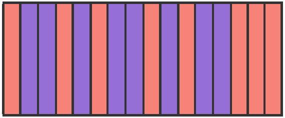
$+ \quad \ell \times$  .

## Theorem

$$\dim \mathscr{C}'^{\star 2} = \min \left( 2(k + w - \ell) - 1 \, , \, n + w - \ell \right).$$

**Independently of the shortened positions!**

# Shortening RLCE

$\mathscr{C}' \stackrel{\text{def}}{=}$  $+ \ \ell \times$  .

**Theorem**

$$\dim \mathscr{C}'^{\star 2} = \min\left( 2(k + w - \ell) - 1 \, , \, n + w - \ell \right).$$

**Independently of the shortened positions!**

# When can we use the distinguisher?

## Conditions

$$\dim \mathscr{C}'^{\star 2} < \binom{k+1-\ell}{2},$$

$$\dim \mathscr{C}'^{\star 2} < n + w - \ell.$$

# When can we use the distinguisher?

## Conditions

$$\min(2(k+w-\ell)-1, n+w-\ell) < \binom{k+1-\ell}{2},$$

$$\min(2(k+w-\ell)-1, n+w-\ell) < n+w-\ell.$$

# When can we use the distinguisher?

## Conditions

$$\ell < k - \frac{3 + \sqrt{16w + 1}}{2},$$

$$w + 2k - n \geqslant \ell.$$

# When can we use the distinguisher?

## Conditions

$$\ell < k - \frac{3 + \sqrt{16w + 1}}{2},$$

$$w + 2k - n \geqslant \ell.$$

**Consequence: works only if**

$$n - k > w + \frac{3 + \sqrt{16w + 1}}{2} = w + O(\sqrt{w}),$$

*i.e.* works up to values of $w$ that are close to $n - k$.

# When can we use the distinguisher?

## Conditions

$$\ell < k - \frac{3 + \sqrt{16w + 1}}{2},$$

$$w + 2k - n \geqslant \ell.$$

**Consequence: works only if**

$$n - k > w + \frac{3 + \sqrt{16w + 1}}{2} = w + O(\sqrt{w}),$$

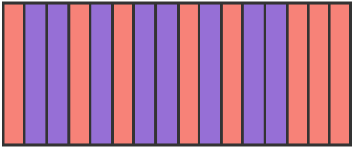
*i.e.* works up to values of $w$ that are close to $n - k$.

## Conclusion

The distinguisher works for **RLCE-short** but not for **RLCE-long**.

# The Attack

# The Attack



+

= ?

# Attack Outline

1. Choose the value of $\ell$.

# Attack Outline

1. Choose the value of $\ell$.
2. Shorten on $\ell$ positions.
   Identify pairs of twin positions.

# Attack Outline

1. Choose the value of $\ell$.
2. Shorten on $\ell$ positions.
   Identify pairs of twin positions.
   Repeat.

# Attack Outline

1. Choose the value of $\ell$.
2. Shorten on $\ell$ positions.
   Identify pairs of twin positions.
   Repeat.
3. Puncture the twin positions to get a GRS code.
   Apply the Sidelnikov Shestakov attack.

# Attack Outline

1. Choose the value of $\ell$.
2. Shorten on $\ell$ positions.
   Identify pairs of twin positions.
   Repeat.
3. Puncture the twin positions to get a GRS code.
   Apply the Sidelnikov Shestakov attack.
4. For each pair of twin positions, recover the mixing matrix.

# Attack Outline

1. Choose the value of $\ell$.
2. Shorten on $\ell$ positions.
   Identify pairs of twin positions.
   Repeat.
3. Puncture the twin positions to get a GRS code.
   Apply the Sidelnikov Shestakov attack.
4. For each pair of twin positions, recover the mixing matrix.
5. Finish to recover the structure of the GRS code.

# Step 1: choose $\ell$

**Constraint:** $\ell_{\min} \leqslant \ell < \ell_{\max}$, where:

$$\begin{aligned} \ell_{\min} &= w + 2k - n \\ \ell_{\max} &= \left\lceil k - \frac{3 + \sqrt{16w+1}}{2} - 1 \right\rceil. \end{aligned}$$

**Choice:**

$$\ell \stackrel{\text{def}}{=} \ell_{\max} - 1.$$

# Attack Outline

1. Choose the value of $\ell$.

# Attack Outline

1. Choose the value of $\ell$.
2. Shorten on $\ell$ positions.
   Identify pairs of twin positions.

# Step 2a: identify PR positions

Choose a subset of columns to shorten:

$$\mathcal{L} \subseteq [\![1, n+w]\!] \text{ s.t. } |\mathcal{L}| = \ell.$$

# Step 2a: identify PR positions

Choose a subset of columns to shorten:

$$\mathcal{L} \subseteq [\![1, n+w]\!] \text{ s.t. } |\mathcal{L}| = \ell.$$

**Idea:** Check for all positions $i \in [\![1, n+w]\!] \setminus \mathcal{L}$:

$$\dim \left(\mathcal{S}_{\mathcal{L}}\left(\mathscr{C}\right)\right)^{\star 2} \stackrel{?}{=} \dim \left(\mathcal{P}_{\{i\}}\left(\mathcal{S}_{\mathcal{L}}\left(\mathscr{C}\right)\right)\right)^{\star 2}.$$

# Step 2a: identify PR positions

Choose a subset of columns to shorten:

$$\mathcal{L} \subseteq [\![1, n+w]\!] \text{ s.t. } |\mathcal{L}| = \ell.$$

**Idea:** Check for all positions $i \in [\![1, n+w]\!] \setminus \mathcal{L}$:

$$\dim \left(\mathcal{S}_{\mathcal{L}}\left(\mathscr{C}\right)\right)^{\star 2} \stackrel{?}{=} \dim \left(\mathcal{P}_{\{i\}}\left(\mathcal{S}_{\mathcal{L}}\left(\mathscr{C}\right)\right)\right)^{\star 2}.$$

Case 1 $i \in \mathcal{I}_{\mathrm{GRS}} \Rightarrow$ the dimension does not change ;

# Step 2a: identify PR positions

Choose a subset of columns to shorten:

$$\mathcal{L} \subseteq [\![1, n+w]\!] \text{ s.t. } |\mathcal{L}| = \ell.$$

**Idea:** Check for all positions $i \in [\![1, n+w]\!] \setminus \mathcal{L}$:

$$\dim \left(\mathcal{S}_{\mathcal{L}}\left(\mathscr{C}\right)\right)^{\star 2} \stackrel{?}{=} \dim \left(\mathcal{P}_{\{i\}}\left(\mathcal{S}_{\mathcal{L}}\left(\mathscr{C}\right)\right)\right)^{\star 2}.$$

Case 1 $i \in \mathcal{I}_{\mathrm{GRS}} \Rightarrow$ the dimension does not change ;

Case 2 $i \in \mathcal{I}_{\mathrm{PR}}$ and $\tau(i) \in \mathcal{L}$: position $i$ is "derandomized" in $\mathcal{S}_{\mathcal{L}}\left(\mathscr{C}\right)$ and behaves like a GRS position $\Rightarrow$ see Case 1 ;

# Step 2a: identify PR positions

Choose a subset of columns to shorten:

$$\mathcal{L} \subseteq [\![1, n+w]\!] \text{ s.t. } |\mathcal{L}| = \ell.$$

**Idea:** Check for all positions $i \in [\![1, n+w]\!] \setminus \mathcal{L}$:

$$\dim \left(\mathcal{S}_{\mathcal{L}}\left(\mathscr{C}\right)\right)^{\star 2} \stackrel{?}{=} \dim \left(\mathcal{P}_{\{i\}}\left(\mathcal{S}_{\mathcal{L}}\left(\mathscr{C}\right)\right)\right)^{\star 2}.$$

Case 1 $i \in \mathcal{I}_{\mathrm{GRS}} \Rightarrow$ the dimension does not change ;

Case 2 $i \in \mathcal{I}_{\mathrm{PR}}$ and $\tau(i) \in \mathcal{L}$: position $i$ is "derandomized" in $\mathcal{S}_{\mathcal{L}}\left(\mathscr{C}\right)$ and behaves like a GRS position $\Rightarrow$ see Case 1 ;

Case 3 $i \in \mathcal{I}_{\mathrm{PR}}$ and $\tau(i) \notin \mathcal{L}$: the column behaves like a random one $\Rightarrow$ puncturing reduces the dimension.

# Step 2a: identify PR positions

Choose a subset of columns to shorten:

$$\mathcal{L} \subseteq [\![1, n+w]\!] \text{ s.t. } |\mathcal{L}| = \ell.$$

**Idea:** Check for all positions $i \in [\![1, n+w]\!] \setminus \mathcal{L}$:

$$\dim \left(\mathcal{S}_{\mathcal{L}}\left(\mathscr{C}\right)\right)^{\star 2} \stackrel{?}{=} \dim \left(\mathcal{P}_{\{i\}}\left(\mathcal{S}_{\mathcal{L}}\left(\mathscr{C}\right)\right)\right)^{\star 2}.$$

Case 1 $i \in \mathcal{I}_{\mathrm{GRS}} \Rightarrow$ the dimension does not change ;

Case 2 $i \in \mathcal{I}_{\mathrm{PR}}$ and $\tau(i) \in \mathcal{L}$: position $i$ is "derandomized" in $\mathcal{S}_{\mathcal{L}}\left(\mathscr{C}\right)$ and behaves like a GRS position $\Rightarrow$ see Case 1 ;

Case 3 $i \in \mathcal{I}_{\mathrm{PR}}$ and $\tau(i) \notin \mathcal{L}$: the column behaves like a random one $\Rightarrow$ puncturing reduces the dimension.

This gives $\mathcal{T}_{\mathcal{L}} \stackrel{\text{def}}{=} \mathcal{I}_{\mathrm{PR}} \cap \left([\![1, n+w]\!] \setminus \mathcal{L}\right)$.

# Step 2b: match twin positions

For $i \in \mathcal{T}_{\mathcal{L}}$, how to identify $\tau(i)$ ?

# Step 2b: match twin positions

For $i \in \mathcal{T}_{\mathcal{L}}$, how to identify $\tau(i)$ ?

**Idea:** Check for all positions $j \in \mathcal{T}_{\mathcal{L}} \setminus \{i\}$:

$$\dim\left(\mathcal{S}_{\mathcal{L}\cup\{i\}}\left(\mathscr{C}\right)\right)^{\star 2} \stackrel{?}{=} \dim\left(\mathcal{P}_{\{j\}}\left(\mathcal{S}_{\mathcal{L}\cup\{i\}}\left(\mathscr{C}\right)\right)\right)^{\star 2}.$$

# Step 2b: match twin positions

For $i \in \mathcal{T}_{\mathcal{L}}$, how to identify $\tau(i)$ ?

**Idea:** Check for all positions $j \in \mathcal{T}_{\mathcal{L}} \setminus \{i\}$:

$$\dim\left(\mathcal{S}_{\mathcal{L}\cup\{i\}}\left(\mathscr{C}\right)\right)^{\star 2} \stackrel{?}{=} \dim\left(\mathcal{P}_{\{j\}}\left(\mathcal{S}_{\mathcal{L}\cup\{i\}}\left(\mathscr{C}\right)\right)\right)^{\star 2}.$$

Case 1 $j = \tau(i)$ : position $j$ is "derandomized" in $\mathcal{S}_{\mathcal{L}\cup\{i\}}\left(\mathscr{C}\right)$ and behaves like a GRS position
$\Rightarrow$ the dimension does not change ;

# Step 2b: match twin positions

For $i \in \mathcal{T}_{\mathcal{L}}$, how to identify $\tau(i)$ ?

**Idea:** Check for all positions $j \in \mathcal{T}_{\mathcal{L}} \setminus \{i\}$:

$$\dim \left(\mathcal{S}_{\mathcal{L}\cup\{i\}}\left(\mathscr{C}\right)\right)^{\star 2} \stackrel{?}{=} \dim \left(\mathcal{P}_{\{j\}}\left(\mathcal{S}_{\mathcal{L}\cup\{i\}}\left(\mathscr{C}\right)\right)\right)^{\star 2}.$$

Case 1 $j = \tau(i)$ : position $j$ is "derandomized" in $\mathcal{S}_{\mathcal{L}\cup\{i\}}(\mathscr{C})$ and behaves like a GRS position
$\Rightarrow$ the dimension does not change ;

Case 2 $j \neq \tau(i)$ : the column behaves like a random one
$\Rightarrow$ puncturing reduces the dimension.

# Step 2b: match twin positions

For $i \in \mathcal{T}_{\mathcal{L}}$, how to identify $\tau(i)$ ?

**Idea:** Check for all positions $j \in \mathcal{T}_{\mathcal{L}} \setminus \{i\}$:

$$\dim \left(\mathcal{S}_{\mathcal{L}\cup\{i\}}\left(\mathscr{C}\right)\right)^{\star 2} \stackrel{?}{=} \dim \left(\mathcal{P}_{\{j\}}\left(\mathcal{S}_{\mathcal{L}\cup\{i\}}\left(\mathscr{C}\right)\right)\right)^{\star 2}.$$

Case 1 $j = \tau(i)$ : position $j$ is "derandomized" in $\mathcal{S}_{\mathcal{L}\cup\{i\}}\left(\mathscr{C}\right)$ and behaves like a GRS position
$\Rightarrow$ the dimension does not change ;

Case 2 $j \neq \tau(i)$ : the column behaves like a random one
$\Rightarrow$ puncturing reduces the dimension.

**Repeat Step 2 with random choices of $\mathcal{L}$ until you identify all twin positions.**

# Attack Outline

1. Choose the value of $\ell$.
2. Shorten on $\ell$ positions.
   Identify pairs of twin positions.
   Repeat.

# Attack Outline

1. Choose the value of $\ell$.
2. Shorten on $\ell$ positions.
   Identify pairs of twin positions.
   Repeat.
3. Puncture the twin positions to get a GRS code.
   Apply the Sidelnikov Shestakov attack.

# Attack Outline

1. Choose the value of $\ell$.
2. Shorten on $\ell$ positions.
   Identify pairs of twin positions.
   Repeat.
3. Puncture the twin positions to get a GRS code.
   Apply the Sidelnikov Shestakov attack.
4. For each pair of twin positions, recover the mixing matrix.

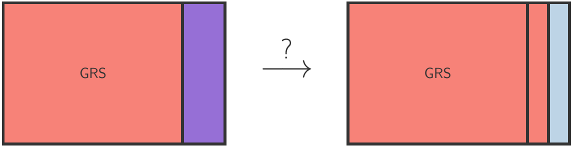
# Step 4: recover the mixing matrix

**For $\{i, \tau(i)\}$ a pair of twin positions,**
**how to recover the GRS and the random column ?**

# Step 4: recover the mixing matrix

**For $\{i, \tau(i)\}$ a pair of twin positions,**
**how to recover the GRS and the random column ?**

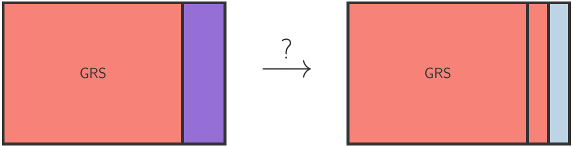
$\mathscr{C}(i) \stackrel{\text{def}}{=}$



# Step 4: recover the mixing matrix

**For $\{i, \tau(i)\}$ a pair of twin positions,**
**how to recover the GRS and the random column ?**

$\mathscr{C}(i) \stackrel{\text{def}}{=}$




**Idea: Use derandomization!**

# Step 4: recover the mixing matrix (2)

By construction, there is

- a polynomial $f \in \mathbb{F}_q[x]_{<k}$ (the GRS part) ;
- a linear form $\psi$ (the random part) ;
- elements $a, b, c, d \in \mathbb{F}_q$ (the mixing)

such that, at position $i$, for any $\boldsymbol{c} \in \mathscr{C}^{(i)}$, we have

$$c_i = a \cdot y_j f(x_j) + c \cdot \psi(f),$$

$$c_{\tau(i)} = b \cdot y_j f(x_j) + d \cdot \psi(f).$$

# Step 4: recover the mixing matrix (2)

By construction, there is

- a polynomial $f \in \mathbb{F}_q[x]_{<k}$ (the GRS part) ;
- a linear form $\psi$ (the random part) ;
- elements $a, b, c, d \in \mathbb{F}_q$ (the mixing)

such that, at position $i$, for any $\boldsymbol{c} \in \mathscr{C}^{(i)}$, we have

$$c_i = a \cdot f(x_j) + c \cdot \psi(f),$$
$$c_{\tau(i)} = b \cdot f(x_j) + \psi(f).$$

# Step 4: recover the mixing matrix (2)

By construction, there is

- a polynomial $f \in \mathbb{F}_q[x]_{<k}$ (the GRS part) ;
- a linear form $\psi$ (the random part) ;
- elements $a, b, c, d \in \mathbb{F}_q$ (the mixing)

such that, at position $i$, for any $\boldsymbol{c} \in \mathscr{C}^{(i)}$, we have

$$c_i = a \cdot f(x_j) + c \cdot \psi(f),$$
$$c_{\tau(i)} = b \cdot f(x_j) + \psi(f).$$

Shortening in $i \quad \Leftrightarrow \quad f \in \mathbb{F}_q[x]_{<k}$ s.t. $c_i = 0$.

# Step 4: recover the mixing matrix (2)

By construction, there is

- a polynomial $f \in \mathbb{F}_q[x]_{<k}$ (the GRS part) ;
- a linear form $\psi$ (the random part) ;
- elements $a, b, c, d \in \mathbb{F}_q$ (the mixing)

such that, at position $i$, for any $\boldsymbol{c} \in \mathscr{C}^{(i)}$, we have

$$c_i = a \cdot f(x_j) + c \cdot \psi(f),$$
$$c_{\tau(i)} = b \cdot f(x_j) + \psi(f).$$

Shortening in $i \quad \Leftrightarrow \quad f \in \mathbb{F}_q[x]_{<k}$ s.t. $c_i = 0$.

Therefore, for any $\boldsymbol{c} \in \mathcal{S}_{\{i\}}\left(\mathscr{C}^{(i)}\right)$, we have

$$c_{\tau(i)} = (b - ac^{-1})f(x_j).$$

# Step 4: recover the mixing matrix (3)

$$c_{\tau(i)} = (b - ac^{-1})f(x_j).$$

# Step 4: recover the mixing matrix (3)

$$c_{\tau(i)} = (b - ac^{-1}) f(x_j).$$

- Collect a basis of codewords in $\mathcal{S}_{\{i\}}\left(\mathscr{C}^{(i)}\right)$,
- Find the corresponding $f \in \mathbb{F}_q[x]_{<k}$ by interpolation on the known GRS positions,
- Deduce the value of $(b - ac^{-1})$ and $x_j$.

# Step 4: recover the mixing matrix (3)

$$c_{\tau(i)} = (b - ac^{-1})f(x_j).$$

- Collect a basis of codewords in $\mathcal{S}_{\{i\}}\left(\mathscr{C}^{(i)}\right)$,
- Find the corresponding $f \in \mathbb{F}_q[x]_{<k}$ by interpolation on the known GRS positions,
- Deduce the value of $(b - ac^{-1})$ and $x_j$.

**Still need to find $a$, $b$ and $c$.**

# Step 4: recover the mixing matrix (3)

$$c_{\tau(i)} = (b - ac^{-1}) f(x_j).$$

- Collect a basis of codewords in $\mathcal{S}_{\{i\}}\left(\mathscr{C}^{(i)}\right)$,
- Find the corresponding $f \in \mathbb{F}_q[x]_{<k}$ by interpolation on the known GRS positions,
- Deduce the value of $(b - ac^{-1})$ and $x_j$.

**Still need to find $a$, $b$ and $c$.**

Bruteforce works ($\mathbb{F}_q = 2^{10}$).
Or use another technical trick.

# Attack Outline

1. Choose the value of $\ell$.
2. Shorten on $\ell$ positions.
   Identify pairs of twin positions.
   Repeat.
3. Puncture the twin positions to get a GRS code.
   Apply the Sidelnikov Shestakov attack.
4. For each pair of twin positions, recover the mixing matrix.

# Attack Outline

1. Choose the value of $\ell$.
2. Shorten on $\ell$ positions.
   Identify pairs of twin positions.
   Repeat.
3. Puncture the twin positions to get a GRS code.
   Apply the Sidelnikov Shestakov attack.
4. For each pair of twin positions, recover the mixing matrix.
5. Finish to recover the structure of the GRS code.

# Conclusion

# Conclusion

# Conclusion

- Recovered the GRS structure of **RLCE-short**.

# Conclusion

- Recovered the GRS structure of **RLCE-short**.
- Complexity in $\mathcal{O}(wn^2k^2)$ operations in $\mathbb{F}_q$.

# Conclusion

- Recovered the GRS structure of **RLCE-short**.
- Complexity in $\mathcal{O}(wn^2k^2)$ operations in $\mathbb{F}_q$.
- Parameters of **RLCE-long** remain out of reach.

# Conclusion

- Recovered the GRS structure of **RLCE-short**.
- Complexity in $\mathcal{O}(wn^2k^2)$ operations in $\mathbb{F}_q$.
- Parameters of **RLCE-long** remain out of reach.

`https://arxiv.org/abs/1805.11489`

Thank you for your attention!