Rank-metric based cryptography and its implementations

Nicolas Aragon

Univ Rennes, CNRS, Inria, IRISA - EMSEC team

November 10, 2021

Journée thématique :
Algorithmes de chiffrement post-quantiques et sécurité matérielle
Introduction
- Context and definitions
- Code-based cryptography

Rank-metric based cryptography
- Definitions
- Encryption
- Digital signatures

The RBC library
- Implementation choices
- Performances
- Security

Conclusion
Post-quantum cryptography

The recent developments of quantum computers threaten currently used public-key cryptography.

Public-key cryptography needs difficult mathematical problems:

- Factorization
- Discrete logarithm
- Decoding random codes
- Searching for short vectors in lattice
- Solving multivariate systems
- ...

Nicolas Aragon
Rank-based cryptography
November 10, 2021
Post-quantum cryptography

The recent developments of quantum computers threaten currently used public-key cryptography.

Public-key cryptography needs difficult mathematical problems:
- Factorization
- Discrete logarithm
- Decoding random codes
  - Searching for short vectors in lattice
  - Solving multivariate systems
  - ...
Post-quantum cryptography

The recent developments of quantum computers threaten currently used public-key cryptography.

Public-key cryptography needs difficult mathematical problems:
- Factorization
- Discrete logarithm
- **Decoding random codes**
- Searching for short vectors in lattice
- Solving multivariate systems
- ...

Nicolas Aragon
NIST standardization process

In 2017, the National Institute for Standards and Technology (NIST) started a standardization process to choose post-quantum encryption and digital signature schemes.

Code-based proposals

There were 19 code-base proposals:

- 17 encryption schemes,
- 2 digital signature schemes.

Code-based cryptography was the second most represented "family", behind lattice-based cryptography.
In 2017, the National Institute for Standards and Technology (NIST) started a standardization process to choose post-quantum encryption and digital signature schemes.

**Code-based proposals**

There were 19 code-base proposals:

- 17 encryption schemes,
- 2 digital signature schemes.

Code-based cryptography was the second most represented "family", behind lattice-based cryptography.
# Error correcting codes

## Definition (Linear code)

An \([n, k] \) linear code \(C\) over \(\mathbb{F}_q\) is a vector subspace of \(\mathbb{F}_q^n\). Elements of \(C\) are called codewords.

## Definition (Generator matrix)

A generator matrix \(G \in \mathbb{F}_q^{k \times n}\) of an \([n, k]_{\mathbb{F}_q}\) code \(C\) is a matrix such that its lines form a basis of the vector space \(C\).

## Definition (Parity-check matrix)

A parity-check matrix \(H \in \mathbb{F}_q^{(n-k) \times n}\) of an \([n, k]_{\mathbb{F}_q}\) code \(C\) is a matrix such that:

\[ x \in C \iff H.x^t = 0 \]
Definition (Linear code)

An \([n, k]\) linear code \(C\) over \(\mathbb{F}_q\) is a vector subspace of \(\mathbb{F}_q^n\). Elements of \(C\) are called codewords.

Definition (Generator matrix)

A generator matrix \(G \in \mathbb{F}_q^{k \times n}\) of an \([n, k]\) \(\mathbb{F}_q\) code \(C\) is a matrix such that its lines form a basis of the vector space \(C\).

Definition (Parity-check matrix)

A parity-check matrix \(H \in \mathbb{F}_q^{(n-k) \times n}\) of an \([n, k]\) \(\mathbb{F}_q\) code \(C\) is a matrix such that:

\[ x \in C \iff H.x^t = 0 \]
Error correcting codes

**Definition (Linear code)**

An \([n, k]\) linear code \(C\) over \(\mathbb{F}_q\) is a vector subspace of \(\mathbb{F}_q^n\). Elements of \(C\) are called codewords.

**Definition (Generator matrix)**

A generator matrix \(G \in \mathbb{F}_q^{k \times n}\) of an \([n, k]\) \(\mathbb{F}_q\) code \(C\) is a matrix such that its lines form a basis of the vector space \(C\).

**Definition (Parity-check matrix)**

A parity-check matrix \(H \in \mathbb{F}_q^{(n-k) \times n}\) of an \([n, k]\) \(\mathbb{F}_q\) code \(C\) is a matrix such that:

\[x \in C \iff H.x^t = 0\]
Hamming metric

In the Hamming metric, we generally consider codes with coefficients in $\mathbb{F}_2$.

**Definition (Hamming metric)**

Let $\mathbf{x}$ a vector $(x_1, \ldots, x_n) \in \mathbb{F}_q^n$. The Hamming weight of $\mathbf{x}$ is the number of non null coordinates of $\mathbf{x}$. The Hamming distance between $\mathbf{x}$ and $\mathbf{y}$ is the number of non null coordinates of $\mathbf{x} - \mathbf{y}$. 
Definition (Syndrome decoding)

Given $H$ a random full-rank $(n - k) \times n$ matrix over $\mathbb{F}_q$, and a syndrome $s \in \mathbb{F}_q^{n-k}$, the problem is to find $e$ such that $He^\top = s$ and $e$ has weight $t$. This problem has been proved NP-complete\(^a\).

The McEliece cryptosystem

McEliece described the first code-based cryptosystem in 78\(^1\).

<table>
<thead>
<tr>
<th>Alice</th>
<th>Bob</th>
</tr>
</thead>
<tbody>
<tr>
<td>( G \leftarrow $ ) generator matrix of a structured code</td>
<td>( G' \leftarrow ) scrambled version of ( G )</td>
</tr>
<tr>
<td>( G' \leftarrow ) scrambled version of ( G )</td>
<td>( e \leftarrow $ ) error of weight ( r )</td>
</tr>
<tr>
<td>( m \leftarrow \text{Decode}(G, c) )</td>
<td>( c \leftarrow mG' + e )</td>
</tr>
</tbody>
</table>

Introduction
  • Context and definitions
  • Code-based cryptography

Rank-metric based cryptography
  • Definitions
  • Encryption
  • Digital signatures

The RBC library
  • Implementation choices
  • Performances
  • Security

Conclusion
Rank metric

In the rank metric, we consider codes with coefficients in $\mathbb{F}_{q^m}$.
Let $(\beta_1, \ldots, \beta_m)$ a basis of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$. To a vector $\mathbf{x} \in \mathbb{F}_{q^m}^n$ We can associate a matrix $\mathbf{M}_x$

$$\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{F}_{q^m}^n \leftrightarrow \mathbf{M}_x = \begin{pmatrix} x_{11} & \cdots & x_{n1} \\ \vdots & \ddots & \vdots \\ x_{1m} & \cdots & x_{nm} \end{pmatrix} \in \mathbb{F}_q^{m \times n}$$

Rank weight and distance

$|\mathbf{x}|_R = \text{Rank}(\mathbf{M}_x)$ and $d(\mathbf{x}, \mathbf{y}) = \text{Rank}(\mathbf{M}_x - \mathbf{M}_y)$
In the rank metric, we consider codes with coefficients in $\mathbb{F}_{q^m}$.

Let $(\beta_1, \ldots, \beta_m)$ a basis of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$. To a vector $x \in \mathbb{F}_{q^m}^n$ we can associate a matrix $M_x$

$$x = (x_1, \ldots, x_n) \in \mathbb{F}_{q^m}^n \leftrightarrow M_x = \begin{pmatrix} x_{11} & \cdots & x_{n1} \\ \vdots & \ddots & \vdots \\ x_{1m} & \cdots & x_{nm} \end{pmatrix} \in \mathbb{F}_{q}^{m \times n}$$

**Rank weight and distance**

$|x|_R = \text{Rank}(M_x)$ and $d(x, y) = \text{Rank}(M_x - M_y)$
Ideal codes

To reduce the size of the parameters, some rank-metric based schemes are based on ideal codes.

**Ideal matrix**

Let $\mathcal{R} = \mathbb{F}_{q^m}[X]/\langle P \rangle$ where $P$ is a polynomial of degree $n$ with coefficients in $\mathbb{F}_q$. Let $y = (y_0, \ldots, y_{n-1}) \in \mathcal{R}$. We can associate the polynomial $y$ with the following ideal matrix:

$$
\mathcal{M}(y) = \begin{pmatrix}
y \\
y \mod P \\
x^2y \mod P \\
\vdots \\
x^{(n-1)}y \mod P
\end{pmatrix} \in \mathbb{F}_{q^m}^{n \times n}.
$$
Encryption schemes

The GPT cryptosystem

McEliee with Gabidulin codes\(^a\).


NIST standardization process

- ROLLO: "McEliee" with LRPC codes.
- RQC: Better security reduction, larger parameters.
Digital signatures

Hash and sign

Most "natural" idea: invert a random syndrome using a structured code ⇒ Ranksign, with LRPC codes, but it was broken\(^a\).


Proof of knowledge

Two approaches:

- Authentication scheme and the Fiat-Shamir transform,
- Adaptation of the Lyubashevsky\(^a\) approach ⇒ Durandal.

Introduction
- Context and definitions
- Code-based cryptography

Rank-metric based cryptography
- Definitions
- Encryption
- Digital signatures

The RBC library
- Implementation choices
- Performances
- Security

Conclusion
### Mathematical objects

#### Elements of $\mathbb{F}_{q^m}$
- Addition
- Multiplication
- Inversion
- ...

#### Elements of $\mathbb{F}_{q^m}^n$
- Rank computation
- Addition, multiplication, inversion
- ...

#### More complex functions
- Operations on vector subspaces of $\mathbb{F}_{q^m}$
- Decoding LRPC and Gabidulin codes
- Sampling low weight vectors
- ...

Nicolas Aragon

Rank-based cryptography

November 10, 2021
Mathematical objects

Elements of $\mathbb{F}_{q^m}$
- Addition
- Multiplication
- Inversion
- ...

Elements of $\mathbb{F}_{q^m}^n$
- Rank computation
- Addition, multiplication, inversion
- ...

More complex functions
- Operations on vector subspaces of $\mathbb{F}_{q^m}$
- Decoding LRPC and Gabidulin codes
- Sampling low weight vectors
- ...

Nicolas Aragon
Rank-based cryptography
November 10, 2021
### Mathematical objects

#### Elements of $\mathbb{F}_{q^m}$
- Addition
- Multiplication
- Inversion
- ...

#### Elements of $\mathbb{F}_{q^n}^m$
- Rank computation
- Addition, multiplication, inversion
- ...

#### More complex functions
- Operations on vector subspaces of $\mathbb{F}_{q^m}$
- Decoding LRPC and Gabidulin codes
- Sampling low weight vectors
- ...

---
Overview

General choices

The RBC library is written in C and focuses on performance without sacrificing usability. It is released under the LGPL license and is available at:

http://rbc-lib.org/

Target users

- Users who want to use the cryptographic schemes implemented in the library,
- Users who want to implement new schemes.
Overview

General choices

The RBC library is written in C and focuses on performance without sacrificing usability.
It is released under the LGPL license and is available at:

http://rbc-lib.org/

Target users

- Users who want to use the cryptographic schemes implemented in the library,
- Users who want to implement new schemes.
Design choices

Implementation of $\mathbb{F}_{q^m}$

- Only $q = 2$.
- $\mathbb{F}_{q^m}$ is implemented as $\mathbb{F}_2[X]/\langle P \rangle$ where $P$ is an irreducible polynomial of degree $m$ over $\mathbb{F}_2$.
- Specific algorithms for each value of $m$.

Memory representation

An element of $\mathbb{F}_{q^m}$ is stored using $\lceil \frac{m}{64} \rceil$ integers, where each bit represents a coefficient of the polynomial.
Design choices

Implementation of $\mathbb{F}_{q^m}$

- Only $q = 2$.
- $\mathbb{F}_{q^m}$ is implemented as $\mathbb{F}_2[X]/\langle P \rangle$ where $P$ is an irreducible polynomial of degree $m$ over $\mathbb{F}_2$.
- Specific algorithms for each value of $m$.

Memory representation

An element of $\mathbb{F}_{q^m}$ is stored using $\lceil \frac{m}{64} \rceil$ integers, where each bit represents a coefficient of the polynomial.
## Preprocessing system

The library uses a preprocessing system that generates optimized code for each finite field:
- Improves performances,
- Avoids duplicate code,
- Adds complexity.

## Build system

The library offers three possible build targets:
- `c32`, for x86 processors.
- `c64`, for x64 processors.
- `avx`, for x64 processors with CLMUL and AVX support.
Design choices

### Preprocessing system

The library uses a preprocessing system that generates optimized code for each finite field:

- Improves performances,
- Avoids duplicate code,
- Adds complexity.

### Build system

The library offers three possible build targets:

- **c32**, for x86 processors.
- **c64**, for x64 processors.
- **avx**, for x64 processors with CLMUL and AVX support.
RBC on microcontroller

Available implementations
- c32 code is compatible with ARM Cortex-M microcontrollers.
- ROLLO and RQC are available in the round 2 mupq project:
  https://github.com/mupq/mupq

Limitations
- Generated files are split into multiple folders.
- OpenSSL used for symmetric cryptography.
RBC on microcontroller

Available implementations

- c32 code is compatible with ARM Cortex-M microcontrollers.
- ROLLO and RQC are available in the round 2 mupq project:
  
  https://github.com/mupq/mupq

Limitations

- Generated files are split into multiple folders.
- OpenSSL used for symmetric cryptography.
Cortex-M4 performance

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Keygen</th>
<th>Encaps</th>
<th>Decaps</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROLLO-I</td>
<td>16 927 603</td>
<td>1 926 332</td>
<td>7 009 943</td>
</tr>
<tr>
<td>RQC</td>
<td>5 756 747</td>
<td>11 340 541</td>
<td>71 551 978</td>
</tr>
<tr>
<td>frodokem640shake</td>
<td>91 940 068</td>
<td>109 310 982</td>
<td>109 009 172</td>
</tr>
<tr>
<td>kyber512</td>
<td>653 616</td>
<td>883 740</td>
<td>981 642</td>
</tr>
<tr>
<td>sikep434</td>
<td>672 303 199</td>
<td>1 100 796 989</td>
<td>1 174 307 957</td>
</tr>
</tbody>
</table>

*Figure 1:* Performances of several KEM on ARM Cortex-M4 in cycles. These implementations are in plain C and target 128 bits security.
Implementation security

Constant time implementations

The goal of the library is to provide efficient and secure implementations, for as many platforms as possible.

Right now the library only focuses on constant time.

There is the need to develop side-channel attacks and countermeasures.\(^a\)

1 Introduction
   - Context and definitions
   - Code-based cryptography

2 Rank-metric based cryptography
   - Definitions
   - Encryption
   - Digital signatures

3 The RBC library
   - Implementation choices
   - Performances
   - Security

4 Conclusion
Future work

Security
- Patch existing security issues,
- Further analyze side-channel leakages,
- Provide a better review of existing code.

Features
- Include additional cryptosystems,
- Explore algorithmic improvements,
- Include additional mathematical objects.
Contributions are welcomed!

https://rbc-lib.org
Questions ?