Decoding Supercodes of Gabidulin Codes and Applications to Cryptanalysis

Maxime BOMBAR, Alain COUVREUR

PQCrypto 2021

July, 2021

Outline

Error correcting codes

General linear code

- Linear subspace $\mathscr{C} \subset \mathbb{F}_q^n$, dimension k, length n, $\mathbb{F}_q$ finite field.
- $(\mathbb{F}_q^n, d)$ metric space.

A hard problem: Bounding distance decoding (BDD)

Given a word $\boldsymbol{y} \in \mathbb{F}_q^n$, and a bound t, find (if exists) a codeword $\boldsymbol{c}$, and $\boldsymbol{e} \in \mathbb{F}_q^n$ such that $\boldsymbol{y} = \boldsymbol{c} + \boldsymbol{e}$ and $d(\boldsymbol{y}, \boldsymbol{c}) \leqslant t$.

$$
\begin{array}{lll}
\text{Hamming weight} & w_H(\boldsymbol{x}) & \stackrel{\text{def}}{=} \#\{i \mid x_i \neq 0\}. \\
\text{Hamming distance} & d_H(\boldsymbol{x}, \boldsymbol{y}) & \stackrel{\text{def}}{=} w_H(\boldsymbol{x} - \boldsymbol{y}).
\end{array}
$$

Code-based cryptography in a nutshell

- McEliece cryptosystem (1978)
  - Public key = random code
  - **quadratic** in security parameter.
  - Security = Hardness of decoding random code.
- Augot-Finiasz (2003)
  - Public key = noisy codeword with large error.
  - **linear** in security parameter.
  - Security = Hardness of decoding Reed-Solomon code above Johnson radius ?

Code-based cryptography in a nutshell

- McEliece cryptosystem (1978)
  - Public key = random code
  - **quadratic** in security parameter.
  - Security = Hardness of decoding random code.
- ~~Augot-Finiasz~~ (2004)
  - Public key = noisy codeword with large error.
  - **linear** in security parameter.
  - Message recovery attack by Coron.

J.S. Coron, *Cryptanalysis of a Public-Key Encryption Scheme Based on the Polynomial Reconstruction Problem*, PKC, 2004

Code-based cryptography in a nutshell

- McEliece cryptosystem (1978)
  - Public key = random code
  - **quadratic** in security parameter.
  - Security = Hardness of decoding random code.
- ~~Augot-Finiasz~~ (2004)
- Faure-Loidreau (2005)
  - Rank-metric version of Augot-Finiasz.
  - **linear** in security parameter.
  - Resist Coron's attack.
  - Security = Hardness of decoding Gabidulin code above half the minimum distance ?

Code-based cryptography in a nutshell

- McEliece cryptosystem (1978)
  - Public key = random code
  - **quadratic** in security parameter.
  - Security = Hardness of decoding random code.
- ~~Augot-Finiasz~~ (2004)
- ~~Faure-Loidreau~~ (2016)
  - Rank-metric version of Augot-Finiasz.
  - **linear** in security parameter.
  - Resist Coron's attack.
  - Key recovery attack.

P. Gaborit, A. Otmani, H. Talé Kalachi *Polynomial-time key recovery attack on the Faure-Loidreau scheme base on Gabidulin codes*, Designs, Codes and Cryptography 2016.

Code-based cryptography in a nutshell

- McEliece cryptosystem (1978)
  - Public key = random code
  - **quadratic** in security parameter.
  - Security = Hardness of decoding random code.
- ~~Augot-Finiasz~~ (2004)
- ~~Faure-Loidreau~~ (2016)
- Two recent repairs, RAMESSES (2020) & LIGA (2021)

J. Lavauzelle, P. Loidreau, B-D. Pham RAMESSES *, a Rank Metric Encryption Scheme with Short Keys*, available on ArXiv (2020).

J. Renner, S. Puchinger, A. Wachter-Zeh LIGA*: A cryptosystem based on the hardness of rank-metric list and interleaved decoding*, accepted for Designs, Codes and Cryptography 2021.

This work

Polynomial time message recovery attack against RAMESSES and LIGA

Implementation in SageMath.

Name	Security Level	Running Time
LIGA-128	128 bits	8 minutes
LIGA-192	192 bits	27 minutes
LIGA-256	256 bits	92 minutes

Outline

Rank metric error correcting codes

Want to see a vector $\boldsymbol{x} \in (\mathbb{F}_{q^m})^n$ as a matrix $\boldsymbol{X}$ over $\mathbb{F}_q$.

$\mathbb{F}_{q^m}$-linear rank metric codes

- $\mathscr{C} \subset \mathbb{F}_{q^m}^n$ linear code of dimension k.
- Rank distance: $d(\boldsymbol{x}, \boldsymbol{y}) := \mathbf{rank}_q(\boldsymbol{X} - \boldsymbol{Y})$.

$$\mathscr{B} = (b_1, \ldots, b_m) \text{ basis of } \mathbb{F}_{q^m}/\mathbb{F}_q, \qquad x_i = \sum_{j=1}^{m} x_{i,j} b_j$$

Extension map

$$\mathrm{Ext}_{\mathscr{B}} : \begin{cases} \mathbb{F}_{q^m}^n & \to & \mathbb{F}_q^{m \times n} \\ \boldsymbol{x} := (x_1, \ldots, x_n) & \mapsto & \boldsymbol{X} := \begin{bmatrix} x_{1,1} & \cdots & x_{n,1} \\ \vdots & \ddots & \vdots \\ x_{1,m} & \cdots & x_{n,m} \end{bmatrix}. \end{cases}$$

Remark. The rank distance doesn't depend on the chosen basis.

Gabidulin codes

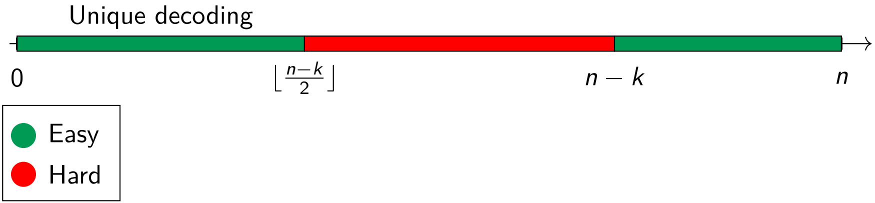
$\mathbb{F}_{q^m}/\mathbb{F}_q$ algebraic extension of degree m.

q-polynomial

- $P = p_0X + p_1X^q + \cdots + p_tX^{q^t}, \quad p_i \in \mathbb{F}_{q^m}, \quad p_t \neq 0.$
- $\deg_q(P) := t.$

Let $\boldsymbol{g} = (g_1, \ldots, g_n) \in \mathbb{F}_{q^m}^n$ whose coordinates are linearly independent. The **Gabidulin code** of dimension k and evaluation vector $\boldsymbol{g}$ is

$$Gab_k(\boldsymbol{g}) = \{(P(g_1), \ldots, P(g_n)) \mid \deg_q(P) < k\}.$$



Outline

Faure-Loidreau PKE

A PKE based on the hardness of decoding a Gabidulin code above half the minimum distance.

Public parameters

$n, k, u \in \mathbb{N}^*$; $\mathbf{G}$ a generator matrix of $Gab_k(\boldsymbol{g}) \subset (\mathbb{F}_{q^n})^n$, $\lfloor \frac{n-k}{2} \rfloor < w < n - k$.

$$\begin{array}{ll} \mathbb{F}_{q^{nu}} & Tr(x) := x + x^{q^n} + \cdots + x^{q^{n(u-1)}} \in \mathbb{F}_{q^n} \text{ is the trace of } \mathbb{F}_{q^{nu}}/\mathbb{F}_{q^n}, \\ \quad \big| \, u & \text{with notation } Tr(x_1, \ldots, x_l) := (Tr(x_1), \ldots, Tr(x_l)). \\ \mathbb{F}_{q^n} & \\ \quad \big| \, n & \\ \mathbb{F}_q & \text{Rank distance is over } \mathbb{F}_q. \end{array}$$

Faure-Loidreau PKE

Keys: $\mathbf{x} \in (\mathbb{F}_{q^{nu}})^k, \mathbf{z} \in (\mathbb{F}_{q^{nu}})^n$ and $\lfloor \frac{n-k}{2} \rfloor < \mathbf{rank}_q(\mathbf{z}) := w < n-k$.
with $(x_{k-u+1}, \dots, x_u)$ a basis of $\mathbb{F}_{q^{nu}}/\mathbb{F}_{q^n}$.

$$\underbrace{\mathbf{k}_{pub}}_{\text{public}} = \underbrace{\mathbf{x}\mathbf{G} + \mathbf{z}}_{\text{private}} \in (\mathbb{F}_{q^{nu}})^n$$

Encrypt: Plaintext is some $\boldsymbol{m} = (m_1, \dots, m_{k-u}, 0, \dots, 0) \in (\mathbb{F}_{q^n})^k$.

- Pick $\alpha \in \mathbb{F}_{q^{nu}}$ at random and $\boldsymbol{e} \in \mathbb{F}_{q^n}^n$ of rank $t := \lfloor \frac{n-k-w}{2} \rfloor$.
- Ciphertext is $\boldsymbol{c} := \boldsymbol{m}\mathbf{G} + \mathrm{Tr}_{q^{nu}/q^n}(\alpha \mathbf{k}_{pub}) + \boldsymbol{e}$.

Faure-Loidreau PKE

$$\underset{\substack{\nearrow \\ \text{public}}}{\mathbf{k}_{pub}} = \underbrace{\mathbf{x}\mathbf{G} + \mathbf{z}}_{\text{private}} \in (\mathbb{F}_{q^{nu}})^n$$

Encrypt: Note that

$$\boldsymbol{c} := \boldsymbol{m}\mathbf{G} + \mathrm{Tr}_{q^{nu}/q^n}(\alpha \mathbf{k}_{pub}) + \boldsymbol{e} = \underbrace{(\boldsymbol{m} + \mathrm{Tr}_{q^{nu}/q^n}(\alpha \mathbf{x}))}_{\mathbf{m}'}\mathbf{G} + (\mathrm{Tr}_{q^{nu}/q^n}(\alpha \mathbf{z}) + \boldsymbol{e}).$$

Decrypt:

- *Puncture* at $\mathrm{Supp}(\mathbf{z})$ and decode $\rightarrow \mathbf{m}'$.
- Knowledge of $\mathbf{x}$ $\rightarrow$ Recover α with linear algebra $\rightarrow \boldsymbol{m}$.

Attack on Faure-Loidreau PKE

$$\mathbf{k}_{pub} = \mathbf{x}\mathbf{G} + \mathbf{z} \in \mathbb{F}_{q^{nu}}^n$$

$\gamma = (\gamma_1, \ldots, \gamma_u)$ basis of $\mathbb{F}_{q^{nu}}/\mathbb{F}_{q^n}$.

Interleaving

$$\mathbf{K}_{pub} := \begin{pmatrix} Tr(\gamma_1 \mathbf{k}_{pub}) \\ \vdots \\ Tr(\gamma_u \mathbf{k}_{pub}) \end{pmatrix}, \mathbf{C} := \begin{pmatrix} Tr(\gamma_1 \mathbf{x})\mathbf{G} \\ \vdots \\ Tr(\gamma_u \mathbf{x})\mathbf{G} \end{pmatrix}, \mathbf{Z} := \begin{pmatrix} Tr(\gamma_1 \mathbf{z}) \\ \vdots \\ Tr(\gamma_u \mathbf{z}) \end{pmatrix} \rightarrow \mathbf{K}_{pub} = \mathbf{C} + \mathbf{Z}.$$

$\boldsymbol{Z}_i$ have a same *row support* over $\mathbb{F}_q$ (namely the support of $\mathbf{z}$).

P. Gaborit, A. Otmani, H. Talé-Kalachi (2016)

$w \leqslant \frac{u}{u+1}(n-k) \Rightarrow$ Recover $\mathbf{x}, \mathbf{z}$ with high probability in polynomial time.

A. Wachter-Zeh, S. Puchinger, J. Renner (2018)

Attack fails if $\mathbf{rank}_{\mathbb{F}_{q^n}}(\mathbf{z})$ is small, *i.e.* errors not independent anymore.

Liga

$$\mathbf{k}_{pub} = \mathbf{x}\mathbf{G} + \mathbf{z} \in (\mathbb{F}_{q^{nu}})^n$$

public   private

$\lfloor \frac{n-k}{2} \rfloor < \mathbf{rank}_q(\mathbf{z}) := w < n - k$

$\zeta \stackrel{\text{def}}{=} \mathbf{rank}_{\mathbb{F}_{q^n}}(\mathbf{z}) < u.$

e.g $\zeta = 2$:

$$\mathbf{z} = \mu_1 \boldsymbol{z}_1 + \mu_2 \boldsymbol{z}_2$$

$\mu_1, \mu_2 \in \mathbb{F}_{q^{nu}}, \boldsymbol{z}_1, \boldsymbol{z}_2 \in \mathbb{F}_{q^n}^n.$

Encrypt: Note that

$$\boldsymbol{c} := \boldsymbol{m}\mathbf{G} + \mathrm{Tr}_{q^{nu}/q^n}(\alpha \mathbf{k}_{pub}) + \boldsymbol{e} = (\mathbf{m}'\mathbf{G} + \mathrm{Tr}(\alpha\mu_1)\boldsymbol{z}_1 + \mathrm{Tr}(\alpha\mu_2)\boldsymbol{z}_2) + \boldsymbol{e}.$$

RAMESSES

Somehow, a dual version of LIGA.
Idea: Avoid linearity of the trace.

- Public key is now a *syndrome*.
- Private key is an *error* of too large weight.
- Plaintext is encoded into an *error*.
- Ciphertext is a noisy codeword.
- **Decrypt:**
  - "*Puncture*" at $\mathrm{Supp}(\boldsymbol{k}_{\mathrm{priv}})$.
  - Perform syndrome decoding.
  - Recovering the plaintext might fail with small probability.

Outline

Decoding supercodes of Gabidulin codes

- $\mathcal{L}_{<d}$ = Set of q-polynomials P with $\deg_q(P) < d \simeq$ Gabidulin code.
- Supercode $\mathscr{C} = \mathcal{L}_{<k} + \mathscr{T}$.
- Received word $\boldsymbol{Y} = \boldsymbol{C} + \boldsymbol{E} = \boldsymbol{C}_0 + \boldsymbol{T} + \boldsymbol{E}$

Berlekamp-Welch key equation

- Take $\boldsymbol{\Lambda} \in \mathcal{L}_{\leqslant t}$ such that $\boldsymbol{\Lambda} \circ \boldsymbol{E} = 0$.
- $\boldsymbol{\Lambda} \circ \boldsymbol{Y} = \boldsymbol{\Lambda} \circ \boldsymbol{C}_0 + \boldsymbol{\Lambda} \circ \boldsymbol{T} \xrightarrow{\text{Linearization}} \boldsymbol{\Lambda} \circ \boldsymbol{Y} = \boldsymbol{N}$
- $\boldsymbol{N} \in (\mathcal{L}_{\leqslant t} \circ \mathcal{L}_{<k}) + (\mathcal{L}_{\leqslant t} \circ \mathscr{T}) = \mathcal{L}_{<k+t} + (\mathcal{L}_{\leqslant t} \circ \mathscr{T})$.

Claim. If $(\mathcal{L}_{<k+t} + \mathcal{L}_{\leqslant t} \circ \mathscr{T}) \cap (\mathcal{L}_{\leqslant t} \circ \boldsymbol{E}) = \{0\}$ then any nonzero solution $(\boldsymbol{\Lambda}, \boldsymbol{N})$ satisfies $\boldsymbol{\Lambda} \circ \boldsymbol{E} = 0$.

Decoding supercodes of Gabidulin codes

- Supercode $\mathscr{C} \stackrel{\text{def}}{=} \mathcal{L}_{<k} + \mathscr{T}$ with $\mathscr{T} \subseteq \mathcal{L}_{<m}$
- Received word $\boldsymbol{Y} = \boldsymbol{C} + \boldsymbol{E} = \boldsymbol{C}_0 + \boldsymbol{T} + \boldsymbol{E}$

Description of the algorithm

(1) Solve Berlekamp-Welch linear system.

(2) Take any nonzero solution $(\boldsymbol{\Lambda}, \boldsymbol{N})$ and compute the right kernel of $\boldsymbol{\Lambda}$ to recover the support of $\boldsymbol{E}$.

(3) Knowing the support, recover the error $\boldsymbol{E}$ (Syndrome decoding).
Warning. Euclidean division is not able to recover $\boldsymbol{C}$ anymore.

Expect to correct almost any error of rank t as soon as

$$k + 2t + \dim(\mathcal{L}_{\leqslant t} \circ \mathscr{T}) \leqslant n.$$

Remark: Attack on RAMESSES needs a *right-hand side* variant of this algorithm.

Outline

Attack on LIGA

(For simplicity, we take $\zeta = 2$).

Recall the ciphertext

$$\boldsymbol{c} = \boldsymbol{m}'\mathbf{G} + \mathrm{Tr}(\alpha\mu_1)\boldsymbol{z}_1 + \mathrm{Tr}(\alpha\mu_2)\boldsymbol{z}_2 + \boldsymbol{e}$$

Idea of the attack:

- **Step 1.** Decode in a *suitable* supercode of Gabidulin code and remove the small error $\boldsymbol{e}$.
- **Step 2.** Remove the $\boldsymbol{z}$ dependency.
- **Step 3.** Recover the plaintext.

Attack on LIGA (Step 1).

(For simplicity, we take $\zeta = 2$).

$\boldsymbol{c} = \boldsymbol{m}'\mathbf{G} + \mathrm{Tr}(\alpha\mu_1)\boldsymbol{z}_1 + \mathrm{Tr}(\alpha\mu_2)\boldsymbol{z}_2 + \boldsymbol{e}$ is a noisy codeword of $\mathscr{G} + \langle \boldsymbol{z}_1, \boldsymbol{z}_2 \rangle =: \mathscr{C}$

Claim.

Set $\mathscr{C}_{pub} \stackrel{\text{def}}{=} \mathscr{G} + \langle \mathrm{Tr}(\gamma_1 \mathbf{k}_{pub}), \mathrm{Tr}(\gamma_2 \mathbf{k}_{pub}) \rangle$ where $\gamma_1, \gamma_2 \in \mathbb{F}_{q^{mu}}$ are linearly independent over $\mathbb{F}_{q^m}$.

$\mathscr{C}_{pub} = \mathscr{C}$ with overwhelming probability over the choices of the γ_i.

One can expect to get rid of $\boldsymbol{e}$ as long as $k + 2t + \zeta(t+1) \leqslant n$.

Attack on LIGA (Step 2).

$\boldsymbol{c} = \boldsymbol{m}\mathbf{G} + \mathrm{Tr}(\alpha \mathbf{k}_{pub}) + \boldsymbol{e}$

Attack on LIGA (Step 2).

$$\boldsymbol{c} = \underbrace{\boldsymbol{m}\mathbf{G} + \mathrm{Tr}(\alpha \mathbf{k}_{pub})}_{\mathbf{c}' = (\boldsymbol{m} + \mathrm{Tr}(\alpha \mathbf{x}))\mathbf{G} + \mathrm{Tr}(\alpha \mathbf{z})} + \cancel{\boldsymbol{e}}$$

Decoding supercode

Attack on LIGA (Step 2).

$\mathbf{c}' := \boldsymbol{m}\mathbf{G} + \mathrm{Tr}(\alpha \mathbf{k}_{pub}) = (\boldsymbol{m} + \mathrm{Tr}(\alpha \mathbf{x}))\mathbf{G} + \mathrm{Tr}(\alpha \mathbf{z})$.

$\boldsymbol{m} = (m_1, \ldots, m_{k-u}, 0, \ldots, 0)$ and $(x_{k-u+1}, \ldots, x_k)$ is a basis of $\mathbb{F}_{q^{nu}}/\mathbb{F}_{q^n}$.

- $\{\beta \in \mathbb{F}_{q^{nu}} \mid \mathbf{c}' - \mathrm{Tr}(\beta \mathbf{k}_{pub}) \in \mathscr{G}\} = \alpha + \bigcap_{i=1}^{\zeta} \langle \mu_i \rangle^{\perp} \stackrel{\mathrm{def}}{=} \alpha + \mathscr{E}$ (Linear algebra).
- $\xrightarrow{\textit{unencode}} \boldsymbol{m} + \{\mathrm{Tr}(\gamma \mathbf{x}) \mid \gamma \in \mathscr{E}\} \stackrel{\mathrm{def}}{=} \boldsymbol{m} + \mathscr{F}$ (Linear algebra).

- (Almost) no more $\mathbf{z}$ dependency !
- The last u components of $\boldsymbol{m} + \mathrm{Tr}(\gamma \mathbf{x})$ are 0 iff $\gamma = 0$.

Attack on LIGA (Step 3).

(i) Take a random element $\boldsymbol{s} = \boldsymbol{m} + \mathrm{Tr}(\gamma \mathbf{x}), \gamma \in \mathscr{E}$.

(ii) Find a generating set $(\mathbf{e_1}, \ldots, \mathbf{e_{u-1}})$ of $\mathscr{F}$.

m is the **only solution** of

$$\begin{cases} \boldsymbol{m} + \sum_{i=1}^{u-1} \lambda_i \mathbf{e_i} & = \boldsymbol{s} \\ m_{k-u+1} = \cdots = m_k & = 0 \end{cases}$$

$k + u$ equations and $k + u - 1$ unknowns $\Rightarrow$ recover $\boldsymbol{m}$.

Efficiency of the attack

Attack on LIGA is in polynomial time.

Implementation in SageMath.

Name	Parameters $(\mathbf{q}, \mathbf{n}, \mathbf{m}, \mathbf{k}, \mathbf{w}, \mathbf{u}, \zeta)$	Security Level	Running Time
LIGA-128	$(2, 92, 92, 53, 27, 5, 2)$	128 bits	8 minutes
LIGA-192	$(2, 120, 120, 69, 35, 5, 2)$	192 bits	27 minutes
LIGA-256	$(2, 148, 148, 85, 43, 5, 2)$	256 bits	92 minutes

Conclusion and perspectives

Contributions.

- Decoding algorithm for supercodes of Gabidulin codes.
- Cryptanalysis of two rank metric encryption schemes with short keys.

Open questions.

- Find a set of parameters that avoids both this attack and the key recovery ?
- Find another cryptosystem based on the hardness of decoding Gabidulin codes above the unique decoding radius ?

The End.

Thanks for your attention !