Cryptanalysis of the Faure-Loidreau PKE, a rank-metric code-based cryptosystem with short keys

Maxime Bombar , Alain Couvreur

GT Grace

November 3, 2020

Outline

1. Introduction
2. Gabidulin codes and the Faure-Loidreau PKE
3. Contribution 1: Alternative attack on the Faure-Loidreau PKE
4. Contribution 2: Attack on the repaired version

Code based cryptography

- *McEliece* : Based on decoding an error of **small** Hamming weight in a (look-alike) **random** code. $\rightarrow$ Usually **huge keys**.

- *Reducing key size ?*

Code based cryptography

- *McEliece* : Based on decoding an error of **small** Hamming weight in a (look-alike) **random** code. $\rightarrow$ Usually **huge keys**.

- *Reducing key size ?*

Code based cryptography

- *McEliece* : Based on decoding an error of **small** Hamming weight in a (look-alike) **random** code. $\rightarrow$ Usually **huge keys**.
- *Reducing key size ?*

Code based cryptography

- *McEliece* : Based on decoding an error of **small** Hamming weight in a (look-alike) **random** code. $\rightarrow$ Usually **huge keys**.
- *Reducing key size ?*
  (1) Large automorphism group $\rightarrow$ Quasi-cyclic, quasi-dyadic . . . .

Code based cryptography

- *McEliece* : Based on decoding an error of **small** Hamming weight in a (look-alike) **random** code. $\rightarrow$ Usually **huge keys**.
- *Reducing key size ?*
  (1) Large automorphism group.
  (2) Rank metric $\rightarrow$ e.g. *GPT* (Eurocrypt 1991) broken by Overbeck in 2005.

Code based cryptography

- *McEliece* : Based on decoding an error of **small** Hamming weight in a (look-alike) **random** code. $\rightarrow$ Usually **huge keys**.

- *Reducing key size ?*
  (1) Large automorphism group.
  (2) Rank metric.
  (3) Another setting $\rightarrow$ *Augot-Finiasz*.

D. Augot, M. Finiasz, *A Public-Key Encryption Scheme based on the Polynomial Reconstruction Problem*, Eurocrypt, 2003

Code based cryptography

- *McEliece* : Based on decoding an error of **small** Hamming weight in a (look-alike) **random** code. $\rightarrow$ Usually **huge keys**.

- *Reducing key size ?*
  1. Large automorphism group.
  2. Rank metric.
  3. Using another setting $\rightarrow$ ~~*Augot–Finiasz*~~ **Message recovery attack**.

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

Code based cryptography

- *McEliece* : Based on decoding an error of **small** Hamming weight in a (look-alike) **random** code. → Usually **huge keys**.
- *Reducing key size ?*
  (1) Large automorphism group.
  (2) **Rank metric**.
  (3) **Using another setting** → *Faure-Loidreau*.

C. Faure, P. Loidreau, *A new public-key cryptosystem based on the problem of reconstructing q-polynomials*, WCC 2005

Error correcting codes

General linear code

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

Bounding distance decoding problem (BDD)

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

Unique decoding radius

- $\delta := d_{min}(\mathscr{C}) := \min\limits_{x \neq y \in \mathscr{C}} d(x, y)$
- $t \leq \lfloor \frac{\delta - 1}{2} \rfloor \Rightarrow$ the BDD problem has at most **one solution**.

Rank metric error correcting codes

Want to see a vector $\mathbf{x} \in (\mathbb{F}_{q^m})^n$ as a matrix $\mathbf{X}$ 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: $d(\mathbf{x}, \mathbf{y}) := \mathbf{Rank}(\mathbf{X} - \mathbf{Y})$.

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

Extension map

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

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

Outline

Non commutative ring of q-polynomials

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

- $P = p_0 X + p_1 X^q + \cdots + p_t X^{q^t}, \quad p_i \in \mathbb{F}_{q^m}, \quad p_t \neq 0.$
- $\deg_q(P) := t$.

- Addition of classical polynomials.
- ~~Multiplication~~ $\rightarrow$ Composition of q-polynomials.

Notations.

$\mathcal{L}\mathbb{F}_{q^m}[X]$ set of q-polynomials.

$\mathcal{L}\mathbb{F}_{q^m}[X]_{\leq t}$ set of q-polynomials of q-degree bounded by t.

Non commutative ring of q-polynomials

Theorem : $(\mathcal{L}\mathbb{F}_{q^m}[X], +, \circ)$ is a **non commutative ring**.
Example. $aX \cdot X^q = aX^q$ while $X^q \cdot aX = a^q X^q$.

A left and right euclidean ring

Let A, B be two q-polynomials.

- $\exists!(Q, R), \quad A = B \circ Q + R$ and $\deg_q(R) < \deg_q(B)$.
- $\exists!(S, T), \quad A = S \circ B + T$ and $\deg_q(S) < \deg_q(B)$.

Roots and interpolation of q-polynomials

A q-polynomial induces an $\mathbb{F}_q$-linear map of $\mathbb{F}_{q^m}$.

Roots of a q-polynomial

- $\mathrm{Ker}(P)$ is linear subspace of dimension at most $\deg_q(P)$.
- For any linear subspace of dimension t there exists a (unique) monic q-polynomial V of q-degree t such that $E = \mathrm{Ker}(V)$.

Lagrange interpolation

Let $\mathbf{g} = (g_1, \ldots, g_n) \in \mathbb{F}_{q^m}^n$ be linearly independent. Let $\mathbf{y} = (y_1, \ldots, y_n) \in \mathbb{F}_{q^m}^n$. There exists a **unique** q-polynomial P of q-degree $< n$ such that:

$$\forall 1 \leq i \leq n, \quad P(g_i) = y_i.$$

Gabidulin codes

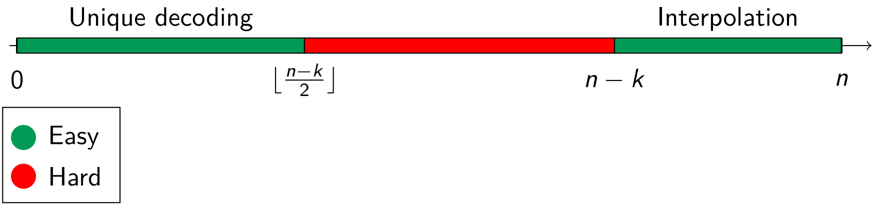
Definition

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

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

$Gab_k(\mathbf{g})$ has minimum distance $n - k + 1$.

Decoding error of rank t in $Gab_k(\mathbf{g})$:



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 $Gab_k(\mathbf{g}) \subset (\mathbb{F}_{q^n})^n$, $\lfloor \frac{n-k}{2} \rfloor < w < n - k$.

$\mathbb{F}_{q^{nu}}$ $\Big|$ u $\mathbb{F}_{q^n}$ $\Big|$ n $\mathbb{F}_q$

$Tr(x) := x + x^{q^n} + \cdots + x^{q^{n(u-1)}} \in \mathbb{F}_{q^n}$ is the trace of $\mathbb{F}_{q^{nu}}/\mathbb{F}_{q^n}$, with notation $Tr(x_1, \ldots, x_l) := (Tr(x_1), \ldots, Tr(x_l))$.

Rank distance is over $\mathbb{F}_q$.

Faure-Loidreau PKE

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

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

public ($\mathbf{k}_{pub}$, $\mathbf{G}$) — private ($\mathbf{x}$, $\mathbf{z}$)

Originality: Short public key, linear in security level.

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

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

Faure-Loidreau PKE

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

public → $\mathbf{k}_{pub}$; private → $\mathbf{x}$, $\mathbf{z}$

Encrypt: Note that

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

Decrypt:

- "Projection" to remove $\mathbf{z}$ dependencies and decode $\to \mathbf{m}'$
- Knowledge of $\mathbf{x} \to$ Recover α with linear algebra $\to \mathbf{m}$.

Attack and repair

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

$(\mathbf{x}, \mathbf{z})$ can be efficiently recovered from $\mathbf{k}_{pub}$ provided that $w \leq \frac{u}{u+1}(n-k)$.

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.

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

Let $\zeta := \mathbf{Rank}_{\mathbb{F}_{q^n}}(z)$.

- Attack fails if $\zeta < \frac{w}{n-k-w}$.
- Repair: Choose $\zeta = 1$.

Attack and repair

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

$(\mathbf{x}, \mathbf{z})$ can be efficiently recovered from $\mathbf{k}_{pub}$ provided that $w \leq \frac{u}{u+1}(n-k)$.

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

Let $\zeta := \mathbf{Rank}_{\mathbb{F}_{q^n}}(z)$.

- Attack fails if $\zeta < \frac{w}{n-k-w}$.
- Repair: Choose $\zeta = 1$.

A. Wachter-Zeh, S. Puchinger, J. Renner, *Repairing the Faure–Loidreau Public-Key Cryptosystem*, ISIT 2018.

Outline

Attack on Faure-Loidreau PKE

Let $\gamma = (\gamma_1, \ldots, \gamma_u)$ be a basis of $\mathbb{F}_{q^{nu}}/\mathbb{F}_{q^n}$, and γ^* be its dual basis : $Tr(\gamma_i \gamma_j^*) = \delta_{i,j}$.

Interleaving

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

Same row support

Claim 1. There exists $\mathscr{E} \subset (\mathbb{F}_q)^n$ of dimension w such that

$$\mathbf{RowSpace}(Tr(\gamma_i \mathbf{z})) \subseteq \mathscr{E}$$

for all $1 \leq i \leq u$.

$\Rightarrow$ Want to work on the right side.

Attack on Faure-Loidreau PKE

Let $\gamma = (\gamma_1, \ldots, \gamma_u)$ be a basis of $\mathbb{F}_{q^{nu}}/\mathbb{F}_{q^n}$, and γ^* be its dual basis : $Tr(\gamma_i \gamma_j^*) = \delta_{i,j}$.

Interleaving

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

Same row support

Claim 1. There exists $\mathscr{E} \subset (\mathbb{F}_q)^n$ of dimension w such that

$$\mathbf{RowSpace}(Tr(\gamma_i \mathbf{z})) \subseteq \mathscr{E}$$

for all $1 \leq i \leq u$.

$\Rightarrow$ Want to work on the right side.

Right Berlekamp-Welch decoding algorithm

$\mathscr{G} = Gab_k(\mathbf{g})$, with $\mathbf{g} \in \mathbb{F}_{q^n}^n$ (here $m = n$).

Interpolation and decoding

$$\begin{array}{c} \mathbf{y} = \mathbf{c} + \mathbf{e}, \\ \mathbf{c} \in \mathscr{G},\ \mathbf{e} \in \mathbb{F}_{q^n}^n \text{ of rank } t \end{array} \xrightarrow{\textit{Interpolation}} \begin{array}{c} \mathbf{Y} = \mathbf{C} + \mathbf{E}, \\ \deg_q(C) < k, \mathbf{Rank}(\mathbf{E}) = t. \end{array}$$

Claim 2. There exists a q-polynomial V with $\deg_q(V) \le t$ such that $E \circ V = 0$.

Systems of equations over $\mathbb{F}_{q^n}$.

$$\begin{cases} Y \circ V = \mathbf{C} \circ V \\ \deg_q V \le t \\ \deg_q \mathbf{C} \le k - 1. \end{cases} \xrightarrow[N := \mathbf{C} \circ V]{\text{Linearization}} \begin{cases} Y \circ V = N \\ \deg_q V \le t \\ \deg_q N \le k + t - 1. \end{cases}$$

n equations, $k + t + 1$ unknowns

Non linear

n equations, $k + 2t + 1$ unknowns $\in \mathbb{F}_{q^n}$

Linear over $\mathbb{F}_q$

Right Berlekamp-Welch decoding algorithm

Claim 3. If $\mathbf{Rank}(\mathbf{E}) \leq \lfloor \frac{n-k}{2} \rfloor$, and if (V, N) is a (non-zero) solution of the linearized system, then $N = \mathbf{C} \circ V$ where $\mathbf{C} = \mathbf{Y} - \mathbf{E}$.

$\Longrightarrow$ Solve the system and recover $\mathbf{C}$ by right euclidean division.

In fact, the system is semi-linear $\longrightarrow$ Adjoint ($\sim$ transpose) of a q-polynomial for bilinear form associated to $Tr_{\mathbb{F}_{q^n}/\mathbb{F}_q}$.

Claim 4. Let $d \leq n$, and $P := \sum_{i=0}^{d} a_i X^{q^i}$. Then $P^* = \sum_{i=0}^{d} a_i^{q^{n-i}} X^{q^{n-i}}$.
(Almost) Same coefficients !

$$Y \circ V = N \xrightarrow{\text{Adjoint}} V^* \circ Y^* = N^* \xrightarrow{\text{Evaluation}} V^*(y_i^*) = N^*(g_i) \text{ for } 1 \leq i \leq n \,.$$

Implementation with **SageMath**.

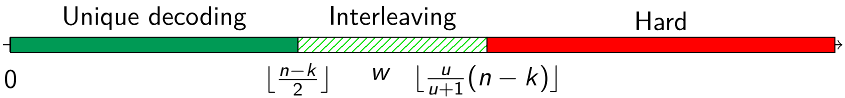
Back to the attack on Faure-Loidreau PKE

$$\mathbf{k}_{pub} = \mathbf{x}\mathbf{G} + \mathbf{z} \xrightarrow[+Interpolation]{Trace} K_i = C_i + Z_i.$$

Claim 5. Z_i have a **common** annulator of q-degree w.

$$\begin{cases} V^*(y_j^*) = N_i^*(g_j) \text{ for } 1 \leq i \leq u \text{ and } 1 \leq j \leq n \\ \deg_q V \leq t \\ \deg_q N_i \leq k + t - 1. \end{cases}$$

- $\mathbb{F}_{q^n}$-Linear system
- $n \times u$ equations $\longrightarrow$ correct up to $\lfloor \frac{u}{u+1}(n-k) \rfloor$ errors.
- $t + 1 + u(k + t)$ unknowns



Limits of the attack

- If all error patterns are the same → As if decoding only one codeword beyond unique decoding radius → Supposed to be hard.
- Need to count independent rows in **Z** → $\zeta := \mathbf{Rank}_{\mathbb{F}_{q^n}}(\mathbf{z})$.
- ⇒ Attack fails if $w > \lfloor \frac{\zeta}{\zeta+1}(n-k) \rfloor$.

⋯ which is exactly the setting used in the 2018 repair of Renner, Puchinger and Wachter-Zeh.

Limits of the attack

- If all error patterns are the same → As if decoding only one codeword beyond unique decoding radius → Supposed to be hard.
- Need to count independent rows in $\mathbf{Z} \to \zeta := \mathbf{Rank}_{\mathbb{F}_{q^n}}(\mathbf{z})$.
- $\Rightarrow$ Attack fails if $w > \lfloor \frac{\zeta}{\zeta+1}(n-k) \rfloor$.

$\cdots$ which is exactly the setting used in the 2018 repair of Renner, Puchinger and Wachter-Zeh.

Outline

Our attack

- Remainder: $\mathbf{c} = (\mathbf{m} + Tr(\alpha \mathbf{x}))\mathbf{G} + Tr(\alpha \mathbf{z}) + \mathbf{e}$ with small error $\mathbf{e}$.
- $\mathbf{Rank}_{\mathbb{F}_{q^n}}(\mathbf{z}) = 1 \Rightarrow \mathbf{z} = \xi \mathbf{z_0}$, $\xi \in \mathbb{F}_{q^{nu}}$ and $\mathbf{z_0} \in \mathbb{F}_{q^n}^n$.

Outline of the attack

- **Step 1 :** Decode **c** in some **computable** code and get rid of **e**.
- **Step 2 :** Find a linear system with **m** as the only solution.
- **Step 3 :** Recover **m**.

Practical experiments

- Implementation with **SageMath**.
- Intel® Core™ i7-5600U 2.60GHz CPU.

q	n	k	u	w	Claimed security level	Time to recover **m**
2	61	31	3	16	90	$\sim$ 4 min
2	62	31	3	17	128	$\sim$ 4 min
2	83	48	24	4	256	$\sim$ 8 min

Step 1: Get rid of $\mathbf{e}$

$\mathbf{c} = \mathbf{m}'\mathbf{G} + Tr(\alpha\xi)\mathbf{z_0} + \mathbf{e}$ is a noisy codeword of $\mathscr{G} + \langle \mathbf{z_0} \rangle =: \mathscr{C}$

Claim.

If $Tr(\xi) \neq 0$ then $\mathscr{C} = \mathscr{G} \oplus \langle Tr(\mathbf{k}_{pub}) \rangle$.

Remark.

$\mathbb{P}(Tr(\xi) = 0) = \frac{1}{q^n}$.

Decoding in the supercode

$\mathscr{C} := \mathscr{G} \oplus \langle Tr(\mathbf{k}_{pub}) \rangle$.

$\mathbf{y} := c_{\mathscr{G}} + \lambda Tr(\mathbf{k}_{pub}) + \mathbf{e}$ noisy codeword of $\mathscr{C}$ with $\mathbf{e}$ of **small** rank t.

A Berlekamp-Welch like decoding algorithm

- **Interpolation :** $\mathbf{Y} = \mathbf{C} + \lambda \mathbf{T} + \mathbf{E}$ with $\deg_q(\mathbf{C}) < k$.

Decoding in the supercode

$\mathscr{C} := \mathscr{G} \oplus \langle Tr(\mathbf{k}_{pub}) \rangle$.

$\mathbf{y} := c_{\mathscr{G}} + \lambda Tr(\mathbf{k}_{pub}) + \mathbf{e}$ noisy codeword of $\mathscr{C}$ with $\mathbf{e}$ of **small** rank t.

A Berlekamp-Welch like decoding algorithm

- **Interpolation :** $\mathbf{Y} = \mathbf{C} + \lambda \mathbf{T} + \mathbf{E}$ with $\deg_q(\mathbf{C}) < k$.
- **Vanishing polynomial :** $\mathbf{V} \circ \mathbf{Y} = \mathbf{V} \circ \mathbf{C} + \mathbf{V} \circ (\lambda \mathbf{T})$ and $\deg_q(\mathbf{V}) = t$.

Decoding in the supercode

$\mathscr{C} := \mathscr{G} \oplus \langle Tr(\mathbf{k}_{pub}) \rangle$.

$\mathbf{y} := c_{\mathscr{G}} + \lambda Tr(\mathbf{k}_{pub}) + \mathbf{e}$ noisy codeword of $\mathscr{C}$ with $\mathbf{e}$ of **small** rank t.

A Berlekamp-Welch like decoding algorithm

- **Interpolation :** $\mathbf{Y} = \mathbf{C} + \lambda \mathbf{T} + \mathbf{E}$ with $\deg_q(\mathbf{C}) < k$.
- **Vanishing polynomial :** $\mathbf{V} \circ \mathbf{Y} = \mathbf{V} \circ \mathbf{C} + \mathbf{V} \circ (\lambda \mathbf{T})$ and $\deg_q(\mathbf{V}) = t$.
- **Linearization :** $\mathbf{V} \circ \mathbf{Y} = \mathbf{N}$ with $\mathbf{N} \in \mathcal{L}\mathbb{F}_{q^n}[X]_{\leq t+k-1} + \mathcal{L}\mathbb{F}_{q^n}[X]_{\leq t} \cdot \mathbf{T}$

Decoding in the supercode

$\mathscr{C} := \mathscr{G} \oplus \langle Tr(\mathbf{k}_{pub}) \rangle$.

$\mathbf{y} := c_{\mathscr{G}} + \lambda Tr(\mathbf{k}_{pub}) + \mathbf{e}$ noisy codeword of $\mathscr{C}$ with $\mathbf{e}$ of **small** rank t.

A Berlekamp-Welch like decoding algorithm

- **Interpolation :** $\mathbf{Y} = \mathbf{C} + \lambda\mathbf{T} + \mathbf{E}$ with $\deg_q(\mathbf{C}) < k$.
- **Vanishing polynomial :** $\mathbf{V} \circ \mathbf{Y} = \mathbf{V} \circ \mathbf{C} + \mathbf{V} \circ (\lambda\mathbf{T})$ and $\deg_q(\mathbf{V}) = t$.
- **Linearization :** $\mathbf{V} \circ \mathbf{Y} = \mathbf{N}$ with $\mathbf{N} \in \mathcal{L}\mathbb{F}_{q^n}[X]_{\leq t+k-1} + \mathcal{L}\mathbb{F}_{q^n}[X]_{\leq t} \cdot \mathbf{T}$
- $3t + k + 2 < n$ equations $\rightarrow$ recover $(\mathbf{V}, \mathbf{N})$.

Decoding in the supercode

$\mathscr{C} := \mathscr{G} \oplus \langle Tr(\mathbf{k}_{pub}) \rangle$.

$\mathbf{y} := c_{\mathscr{G}} + \lambda Tr(\mathbf{k}_{pub}) + \mathbf{e}$ noisy codeword of $\mathscr{C}$ with $\mathbf{e}$ of **small** rank t.

A Berlekamp-Welch like decoding algorithm

- **Interpolation :** $\mathbf{Y} = \mathbf{C} + \lambda \mathbf{T} + \mathbf{E}$ with $\deg_q(\mathbf{C}) < k$.
- **Vanishing polynomial :** $\mathbf{V} \circ \mathbf{Y} = \mathbf{V} \circ \mathbf{C} + \mathbf{V} \circ (\lambda \mathbf{T})$ and $\deg_q(\mathbf{V}) = t$.
- **Linearization :** $\mathbf{V} \circ \mathbf{Y} = \mathbf{N}$ with $\mathbf{N} \in \mathcal{L}\mathbb{F}_{q^n}[X]_{\leq t+k-1} + \mathcal{L}\mathbb{F}_{q^n}[X]_{\leq t} \cdot \mathbf{T}$
- $3t + k + 2 < n$ equations $\rightarrow$ recover $(\mathbf{V}, \mathbf{N})$.

What did we get ?

Decoding in the supercode

$\mathscr{C} := \mathscr{G} \oplus \langle Tr(\mathbf{k}_{pub}) \rangle$.
$\mathbf{y} := c_{\mathscr{G}} + \lambda Tr(\mathbf{k}_{pub}) + \mathbf{e}$ noisy codeword of $\mathscr{C}$ with **e** of **small** rank t.

A Berlekamp-Welch like decoding algorithm

- **Interpolation :** $\mathbf{Y} = \mathbf{C} + \lambda\mathbf{T} + \mathbf{E}$ with $\deg_q(\mathbf{C}) < k$.
- **Vanishing polynomial :** $\mathbf{V} \circ \mathbf{Y} = \mathbf{V} \circ \mathbf{C} + \mathbf{V} \circ (\lambda\mathbf{T})$ and $\deg_q(\mathbf{V}) = t$.
- **Linearization :** $\mathbf{V} \circ \mathbf{Y} = \mathbf{N}$ with $\mathbf{N} \in \mathcal{L}\mathbb{F}_{q^n}[X]_{\leq t+k-1} + \mathcal{L}\mathbb{F}_{q^n}[X]_{\leq t} \cdot \mathbf{T}$
- $3t + k + 2 < n$ equations $\rightarrow$ recover $(\mathbf{V}, \mathbf{N})$.

What did we get ?

- We have $\mathbf{V} \mid \mathbf{N}$ but left division won't give much information about $\mathbf{C}$ ...

Decoding in the supercode

$\mathscr{C} := \mathscr{G} \oplus \langle Tr(\mathbf{k}_{pub}) \rangle$.
$\mathbf{y} := c_{\mathscr{G}} + \lambda Tr(\mathbf{k}_{pub}) + \mathbf{e}$ noisy codeword of $\mathscr{C}$ with $\mathbf{e}$ of **small** rank t.

A Berlekamp-Welch like decoding algorithm

- **Interpolation :** $\mathbf{Y} = \mathbf{C} + \lambda \mathbf{T} + \mathbf{E}$ with $\deg_q(\mathbf{C}) < k$.
- **Vanishing polynomial :** $\mathbf{V} \circ \mathbf{Y} = \mathbf{V} \circ \mathbf{C} + \mathbf{V} \circ (\lambda \mathbf{T})$ and $\deg_q(\mathbf{V}) = t$.
- **Linearization :** $\mathbf{V} \circ \mathbf{Y} = \mathbf{N}$ with $\mathbf{N} \in \mathcal{L}\mathbb{F}_{q^n}[X]_{\leq t+k-1} + \mathcal{L}\mathbb{F}_{q^n}[X]_{\leq t} \cdot \mathbf{T}$
- $3t + k + 2 < n$ equations $\rightarrow$ recover $(\mathbf{V}, \mathbf{N})$.

What did we get ?

- We have $\mathbf{V} \mid \mathbf{N}$ but left division won't give much information about $\mathbf{C}$ ...
- ... However $\mathbf{V}$ vanishes on $\mathbf{Supp}(\mathbf{e})$! $\Rightarrow$ Enables to recover $\mathbf{e}$ efficiently.

Step 2: Recover the plaintext m

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

Step 2: Recover the plaintext m

$$\mathbf{c} = \underbrace{\mathbf{m}\mathbf{G} + Tr(\alpha \mathbf{k}_{pub})}_{\mathbf{c}'=(\mathbf{m}+Tr(\alpha\mathbf{x}))\mathbf{G}+Tr(\alpha\xi)\mathbf{z_0}} + \cancel{\mathbf{e}} \quad \textbf{Step 1}.$$

Step 2: Recover the plaintext m

$$\mathbf{c}' := \mathbf{m}\mathbf{G} + Tr(\alpha \mathbf{k}_{pub}) = (\mathbf{m} + Tr(\alpha \mathbf{x}))\mathbf{G} + Tr(\alpha\xi)\mathbf{z_0}.$$

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

- $\{\beta \in \mathbb{F}_{q^{nu}} \mid \mathbf{c}' - Tr(\beta \mathbf{k}_{pub}) \in \mathscr{G}\} = \alpha + \langle \xi \rangle^{\perp} \xrightarrow{unencode} \mathbf{m} + \{Tr(\gamma \mathbf{x}) \mid \gamma \in \langle \xi \rangle^{\perp}\}$
- The last u components of $\mathbf{m} + Tr(\gamma \mathbf{x})$ are 0 iff $\gamma = 0$.

Step 2: Recover the plaintext m

$$\mathbf{c}' = (\mathbf{m} + Tr(\alpha\mathbf{x}))\mathbf{G} + Tr(\alpha\xi)\mathbf{z_0}.$$

(i) Take a random element $\mathbf{s} = \mathbf{m} + Tr(\gamma\mathbf{x}), \gamma \in \langle\xi\rangle^{\perp}$.

(ii) Find a generating set $(\mathbf{e_1}, \dots, \mathbf{e_{u-1}})$ of $\{ Tr(\gamma\mathbf{x}) \mid \gamma \in \langle\xi\rangle^{\perp}\}$.

m is the **only solution** of

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

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

A new hope ?

- Can we increase ζ ?

A new hope ?

- Can we increase ζ ? No !
  - $\rightarrow$ The attack can be generalized while $n + 1 \geq k + t + (\zeta + 1)(t + 1)$.
  - $\rightarrow$ Increasing $\zeta \Rightarrow$ increasing w to resist key-recovery attack $\Rightarrow$ decreasing $t := \lfloor \frac{n-k-w}{2} \rfloor$... which must be ≥ 1.

A new hope ?

- Can we increase ζ ?   No !
  - $\rightarrow$ The attack can be generalized while $n+1 \geq k+t+(\zeta+1)(t+1)$.
  - $\rightarrow$ Increasing $\zeta \Rightarrow$ increasing w to resist key-recovery attack $\Rightarrow$ decreasing $t := \lfloor \frac{n-k-w}{2} \rfloor$... which must be ≥ 1.
- LIGA cryptosystem (J. Renner, S. Puchinger, A. Wachter-Zeh) on arxiv ... seems still vulnerable to our attack.

A new hope ?

- Can we increase ζ ? No !
  - $\rightarrow$ The attack can be generalized while $n+1 \geq k+t+(\zeta+1)(t+1)$.
  - $\rightarrow$ Increasing $\zeta \Rightarrow$ increasing w to resist key-recovery attack $\Rightarrow$ decreasing $t := \lfloor \frac{n-k-w}{2} \rfloor$... which must be ≥ 1.
- LIGA cryptosystem (J. Renner, S. Puchinger, A. Wachter-Zeh) on arxiv ... seems still vulnerable to our attack.
- RAMESSES cryptosystem (J. Lavauzelle, P. Loidreau, B.-D. Pham) on arxiv another encryption scheme with short keys can be attacked by a similar method (need right hand side decoding).

Conclusion and perspectives

Contributions.

- Alternative decoding algorithm for (interleaved) Gabidulin codes.
- Alternative attack on the original Faure-Loidreau PKE.
- A new message recovery attack on the repair.

Open question.

- Build a provably secure PKE based on decoding Gabidulin codes above unique decoding radius ?

The End.

Thanks for your attention !