On the Security of Some Cryptosystems Based on Gabidulin Codes

Ayoub Otmani[1] **Hervé Talé Kalachi** [2] Sélestin Ndjeya [2]

University of Rouen, France.

University of Yaounde 1, Cameroon.

April 3, 2018

Introduction

Linear code

1. $(\mathbb{F}^n, \|\cdot\|)$, $\mathbb{F}$ a finite field and $\|\cdot\|$ a norm
2. Linear code $\mathscr{C}$ = v.ss of $(\mathbb{F}^n, \|\cdot\|)$

$$\mathscr{C} = \bigoplus_{i=1}^{k} \mathbb{F}\,\vec{v}_i$$

   where $\vec{v}_i$ are linearly independent.

3. The matrix $G = \begin{pmatrix} \vec{v}_1 \\ \cdot \\ \cdot \\ \cdot \\ \vec{v}_k \end{pmatrix}$ is called a generator matrix of $\mathscr{C}$
4. Any $k \times n$ matrix whose rows form a basis of $\mathscr{C}$ is also a generator matrix of $\mathscr{C}$

Linear code

1. $(\mathbb{F}^n, \|\cdot\|)$, $\mathbb{F}$ a finite field and $\|\cdot\|$ a norm
2. Linear code $\mathscr{C}$ = v.ss of $(\mathbb{F}^n, \|\cdot\|)$

$$\mathscr{C} = \bigoplus_{i=1}^{k} \mathbb{F}\, \vec{v}_i$$

where $\vec{v}_i$ are linearly independent.

3. The matrix $\boldsymbol{G} = \begin{pmatrix} \vec{v}_1 \\ . \\ . \\ . \\ \vec{v}_k \end{pmatrix}$ is called a generator matrix of $\mathscr{C}$
4. Any $k \times n$ matrix whose rows form a basis of $\mathscr{C}$ is also a generator matrix of $\mathscr{C}$

Introduction

Linear code

1. $(\mathbb{F}^n, \|\cdot\|)$, $\mathbb{F}$ a finite field and $\|\cdot\|$ a norm
2. Linear code $\mathscr{C}$ = v.ss of $(\mathbb{F}^n, \|\cdot\|)$

$$\mathscr{C} = \bigoplus_{i=1}^{k} \mathbb{F}\, \vec{v}_i$$

where $\vec{v}_i$ are linearly independent.

3. The matrix $G = \begin{pmatrix} \vec{v}_1 \\ \cdot \\ \cdot \\ \cdot \\ \vec{v}_k \end{pmatrix}$ is called a generator matrix of $\mathscr{C}$
4. Any $k \times n$ matrix whose rows form a basis of $\mathscr{C}$ is also a generator matrix of $\mathscr{C}$

Linear code

1. $(\mathbb{F}^n, \|\cdot\|)$, $\mathbb{F}$ a finite field and $\|\cdot\|$ a norm
2. Linear code $\mathscr{C}$ = v.ss of $(\mathbb{F}^n, \|\cdot\|)$

$$\mathscr{C} = \bigoplus_{i=1}^{k} \mathbb{F}\,\vec{v}_i$$

   where $\vec{v}_i$ are linearly independent.

3. The matrix $\boldsymbol{G} = \begin{pmatrix} \vec{v}_1 \\ . \\ . \\ . \\ \vec{v}_k \end{pmatrix}$ is called a generator matrix of $\mathscr{C}$
4. Any $k \times n$ matrix whose rows form a basis of $\mathscr{C}$ is also a generator matrix of $\mathscr{C}$

Linear code

1. $(\mathbb{F}^n, \|\cdot\|)$, $\mathbb{F}$ a finite field and $\|\cdot\|$ a norm
2. Linear code $\mathscr{C}$ = v.ss of $(\mathbb{F}^n, \|\cdot\|)$

$$\mathscr{C} = \bigoplus_{i=1}^{k} \mathbb{F}\,\vec{v}_i$$

where $\vec{v}_i$ are linearly independent.

3. The matrix $\boldsymbol{G} = \begin{pmatrix} \vec{v}_1 \\ . \\ . \\ . \\ \vec{v}_k \end{pmatrix}$ is called a generator matrix of $\mathscr{C}$
4. Any $k \times n$ matrix whose rows form a basis of $\mathscr{C}$ is also a generator matrix of $\mathscr{C}$

Some usual metrics

Let $\mathbb{F}_{q^m}/\mathbb{F}_q$ and $\vec{x} = (x_1 \cdots x_n) \in \mathbb{F}_{q^m}^n$.

1. **Hamming metric:**
$$\|\vec{x}\|_h = \#\{\ i\ :\ x_i \neq 0\}$$
2. **Rank metric:**
$$\|\vec{x}\|_q = \dim < x_1, \cdots x_n >_{\mathbb{F}_q}$$

Example

- $\mathbb{F} = \mathbb{F}_{2^5} = \mathbb{F}_2 < w > = < 1, w, w^2, w^3, w^4 >_{\mathbb{F}_2}$
- $\vec{x}_1 = (w, 0, 0, w)$

1. **Hamming metric:**
   - $\|\vec{x}_1\|_h = 2$
2. **Rank metric:**
   - $\|\vec{x}_1\|_2 = \dim\left(< w, w >_{\mathbb{F}_2}\right) = 1$

Introduction

Some usual metrics

Let $\mathbb{F}_{q^m}/\mathbb{F}_q$ and $\vec{x} = (x_1 \cdots x_n) \in \mathbb{F}_{q^m}^n$.

1. **Hamming metric:**

$$\|\vec{x}\|_h = \#\{\ i\ :\ x_i \neq 0\}$$

2. **Rank metric:**

$$\|\vec{x}\|_q = \dim < x_1, \cdots x_n >_{\mathbb{F}_q}$$

Example

- $\mathbb{F} = \mathbb{F}_{2^5} = \mathbb{F}_2 < w >=< 1, w, w^2, w^3, w^4 >_{\mathbb{F}_2}$
- $\vec{x}_1 = (w, 0, 0, w)$

1. **Hamming metric:**
   - $\|\vec{x}_1\|_h = 2$
2. **Rank metric:**
   - $\|\vec{x}_1\|_2 = \dim(< w, w >_{\mathbb{F}_2}) = 1$

Introduction

Some usual metrics

Let $\mathbb{F}_{q^m}/\mathbb{F}_q$ and $\vec{x} = (x_1 \cdots x_n) \in \mathbb{F}_{q^m}^n$.

1. **Hamming metric:**

$$\|\vec{x}\|_h = \#\{\ i\ :\ x_i \neq 0\}$$

2. **Rank metric:**

$$\|\vec{x}\|_q = \dim < x_1, \cdots x_n >_{\mathbb{F}_q}$$

Example

- $\mathbb{F} = \mathbb{F}_{2^5} = \mathbb{F}_2 < w >=< 1, w, w^2, w^3, w^4 >_{\mathbb{F}_2}$
- $\vec{x}_1 = (w, 0, 0, w)$

1. **Hamming metric:**
   - $\|\vec{x}_1\|_h = 2$
2. **Rank metric:**
   - $\|\vec{x}_1\|_2 = \dim\left(< w, w >_{\mathbb{F}_2}\right) = 1$

Introduction

Some usual metrics

Let $\mathbb{F}_{q^m}/\mathbb{F}_q$ and $\vec{x} = (x_1 \cdots x_n) \in \mathbb{F}_{q^m}^n$.

1. **Hamming metric:**

$$\|\vec{x}\|_h = \#\{\ i\ :\ x_i \neq 0\}$$

2. **Rank metric:**

$$\|\vec{x}\|_q = \dim < x_1, \cdots x_n >_{\mathbb{F}_q}$$

Example

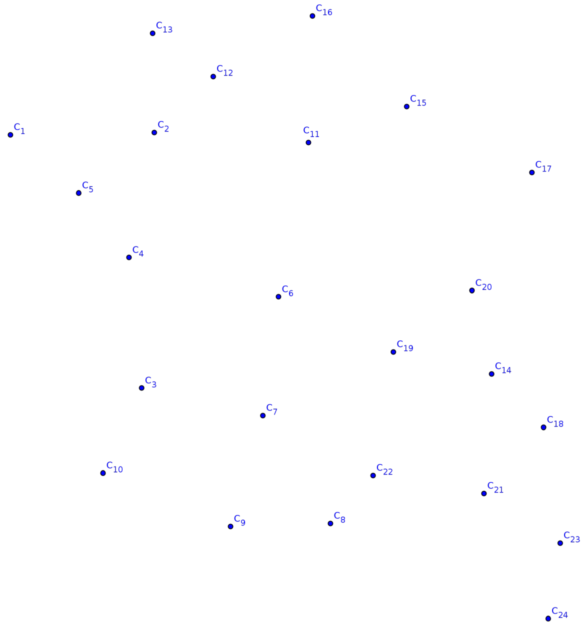
- $\mathbb{F} = \mathbb{F}_{2^5} = \mathbb{F}_2 < w >=< 1, w, w^2, w^3, w^4 >_{\mathbb{F}_2}$
- $\vec{x}_1 = (w, 0, 0, w)$

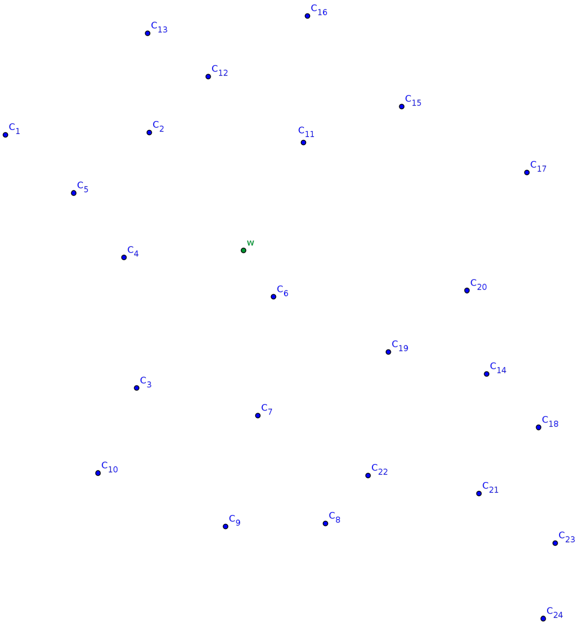
1. **Hamming metric:**
   - $\|\vec{x}_1\|_h = 2$
2. **Rank metric:**
   - $\|\vec{x}_1\|_2 = \dim\left(< w, w >_{\mathbb{F}_2}\right) = 1$

Introduction

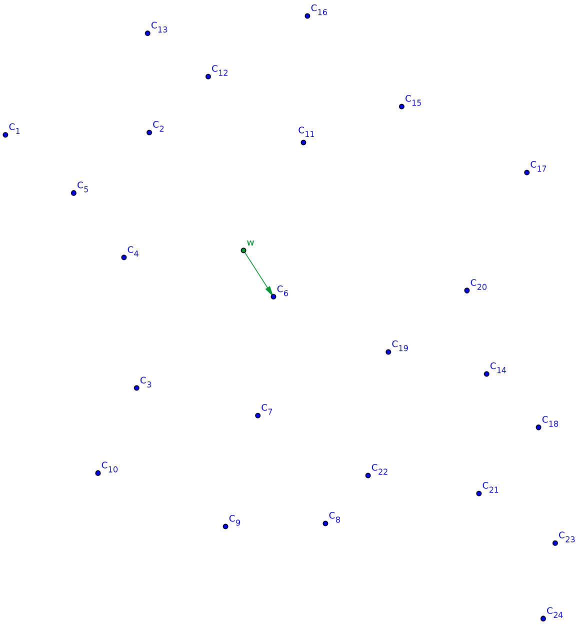
Decoding $\vec{w} \in \mathbb{F}^n$ in $\mathscr{C}$ = Closest Vector Problem (CVP) with Hamming / Rank metric.

Introduction

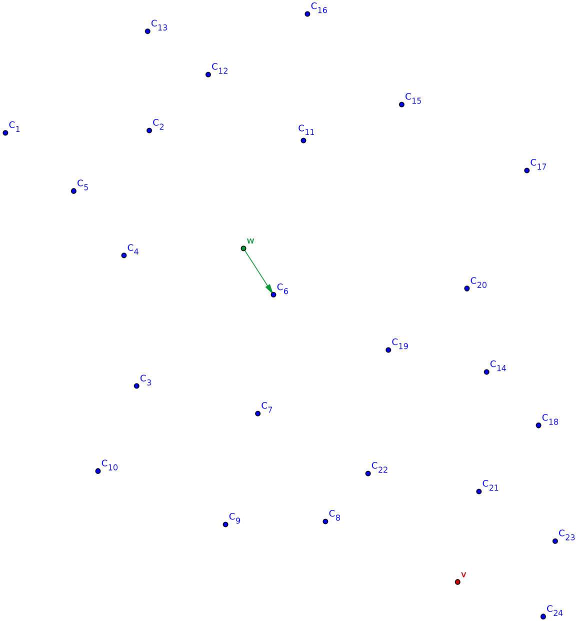


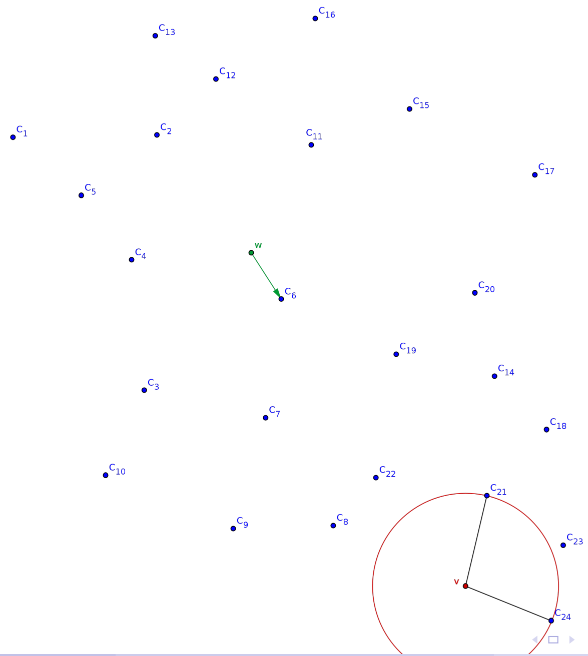

c_{16}
c_{13}
c_{12}
c_{15}
c_1
c_2
c_{11}
c_{17}
c_5
w
c_4
c_6
c_{20}
c_{19}
c_{14}
c_3
c_7
c_{18}
c_{10}
c_{22}
c_{21}
c_9
c_8
c_{23}
c_{24}

Introduction - Decoding

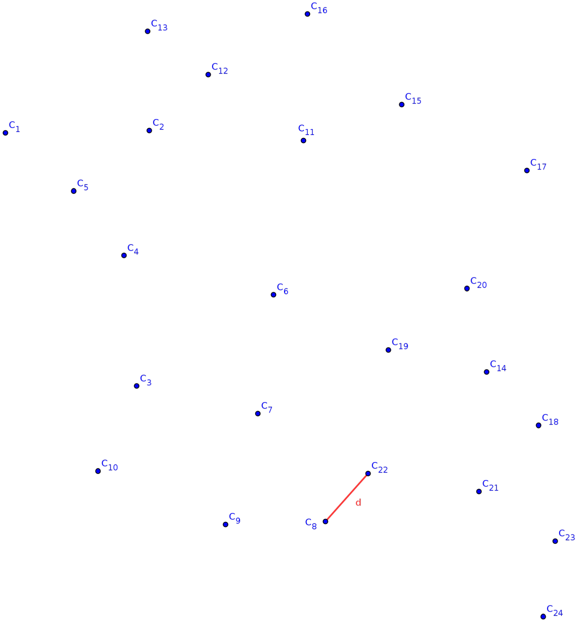


Introduction - Decoding

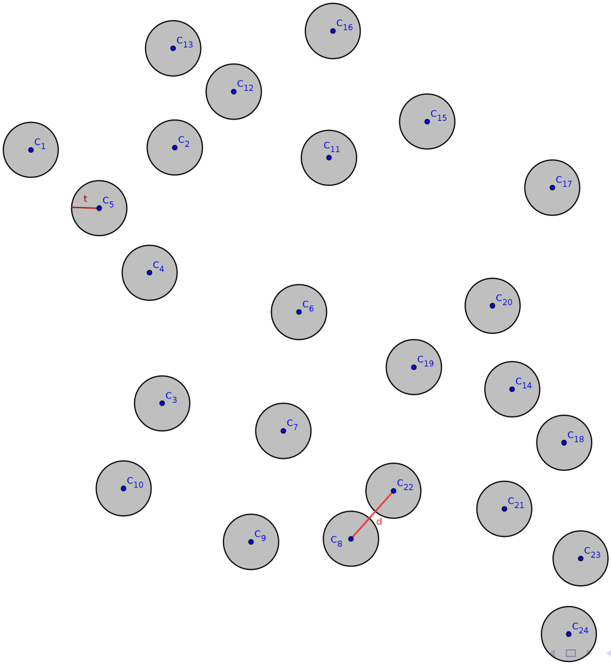



c_{16}
c_{13}
c_{12}
c_{15}
c_1
c_2
c_{11}
c_{17}
c_5
w
c_4
c_6
c_{20}
c_{19}
c_{14}
c_3
c_7
c_{18}
c_{10}
c_{22}
c_{21}
c_9
c_8
c_{23}
v
c_{24}

Introduction - Decoding problem



Introduction - Decoding problem



Hardness of decoding

- Decoding is NP-Hard for a "random" linear code
  - ∗ For Hamming metric: Berlekamp-McEliece-Van Tilborg '78
  - ∗ For Rank metric: Gaborit-Zémor '16

Solving the decoding problem

1. Hamming metric
   - Information set decoding
   - Introduced by Prange '62
   - Complexity: $2^{at(1+o(1))}$

     $$a = constante(\frac{k}{n}, \frac{t}{n})$$

2. Rank metric (the best):
   - Ourivski-Johannsson '02

     $$(tm)^3 2^{kt+f(k,t)}$$

   - Gaborit-Ruatta-Shreck '16 (pour $n \geqslant m$)

     $$(n-k)^3 m^3 2^{(kt+f(k,t))m/n}$$

Introduction

Hardness of decoding

- Decoding is NP-Hard for a "random" linear code
  - ∗ For Hamming metric: Berlekamp-McEliece-Van Tilborg '78
  - ∗ For Rank metric: Gaborit-Zémor '16

Solving the decoding problem

1. Hamming metric
   - Information set decoding
   - Introduced by Prange '62
   - Complexity: $2^{at(1+o(1))}$

     $$a = constante(\frac{k}{n}, \frac{t}{n})$$

2. Rank metric (the best):
   - Ourivski-Johannsson '02

     $$(tm)^3 2^{kt+f(k,t)}$$

   - Gaborit-Ruatta-Shreck '16 (pour $n \geqslant m$)

     $$(n-k)^3 m^3 2^{(kt+f(k,t))m/n}$$

Hardness of decoding

- Decoding is NP-Hard for a "random" linear code
  - ∗ For Hamming metric: Berlekamp-McEliece-Van Tilborg '78
  - ∗ For Rank metric: Gaborit-Zémor '16

Solving the decoding problem

1. Hamming metric
   - Information set decoding
   - Introduced by Prange '62
   - Complexity: $2^{at(1+o(1))}$

$$a = constante(\frac{k}{n}, \frac{t}{n})$$

2. Rank metric (the best):
   - Ourivski-Johannsson '02

$$(tm)^3 2^{kt+f(k,t)}$$

   - Gaborit-Ruatta-Shreck '16 (pour $n \geqslant m$)

$$(n-k)^3 m^3 2^{(kt+f(k,t))m/n}$$

Hardness of decoding

- Decoding is NP-Hard for a "random" linear code
  - ∗ For Hamming metric: Berlekamp-McEliece-Van Tilborg '78
  - ∗ For Rank metric: Gaborit-Zémor '16

Solving the decoding problem

1. Hamming metric
   - Information set decoding
   - Introduced by Prange '62
   - Complexity: $2^{at(1+o(1))}$

   $$a = constante(\frac{k}{n}, \frac{t}{n})$$

2. Rank metric (the best):
   - Ourivski-Johannsson '02

   $$(tm)^3 2^{kt+f(k,t)}$$

   - Gaborit-Ruatta-Shreck '16 (pour $n \geqslant m$)

   $$(n-k)^3 m^3 2^{(kt+f(k,t))m/n}$$

Some codes with efficient decoding algorithms

1. **Generalized Reed-Solomon (GRS)** codes '60 — One-variable polynomials
2. **Goppa** codes '70 — Sub-field sub-codes of GRS codes
3. **Reed-Muller** codes '54 — Multivariate polynomials
4. **Gabidulin** codes '85 — Linearized polynomials with one variable.

Some codes with efficient decoding algorithms

1. **Generalized Reed-Solomon (GRS)** codes '60 — One-variable polynomials
2. **Goppa** codes '70 — Sub-field sub-codes of GRS codes
3. **Reed-Muller** codes '54 — Multivariate polynomials
4. **Gabidulin** codes '85 — Linearized polynomials with one variable.

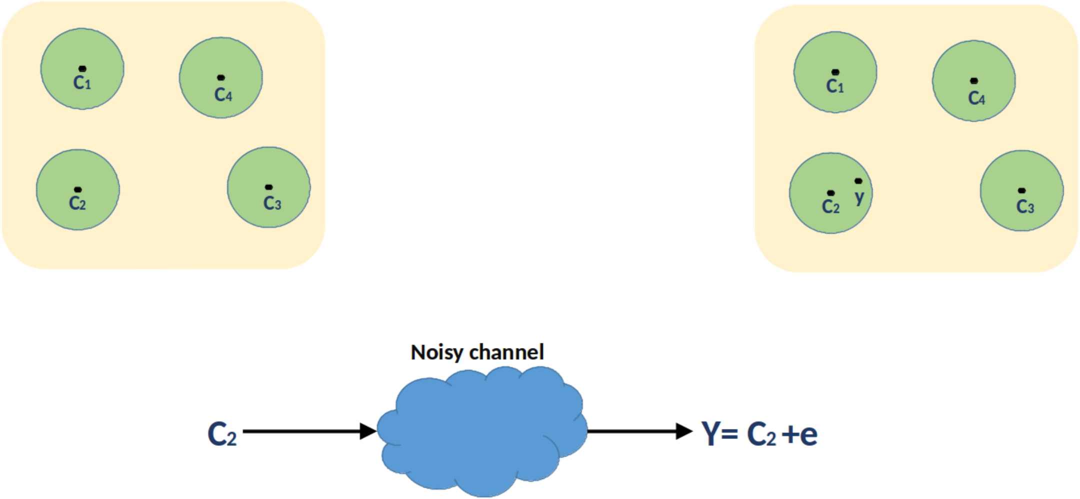
Some codes with efficient decoding algorithms

1. **Generalized Reed-Solomon (GRS)** codes '60 — One-variable polynomials
2. **Goppa** codes '70 — Sub-field sub-codes of GRS codes
3. **Reed-Muller** codes '54 — Multivariate polynomials
4. **Gabidulin** codes '85 — Linearized polynomials with one variable.

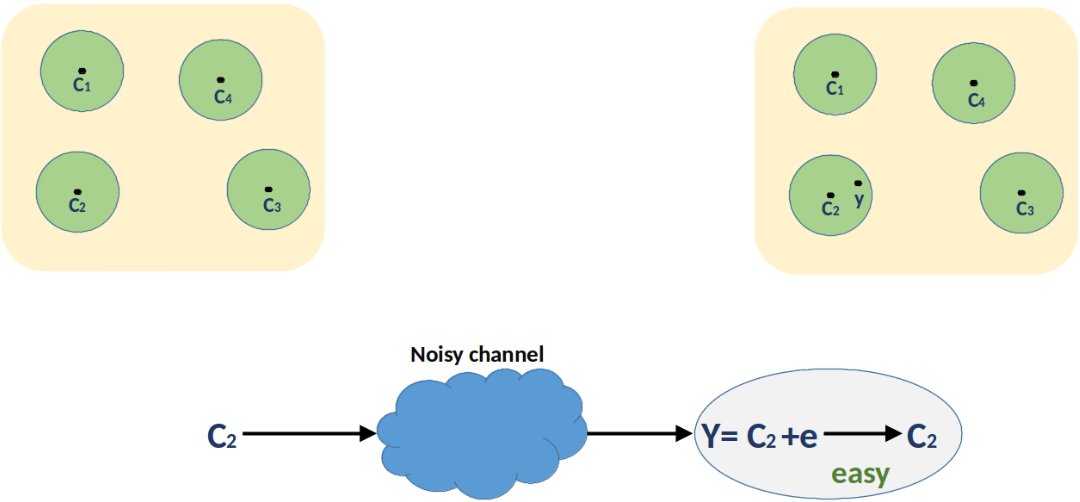
Some codes with efficient decoding algorithms

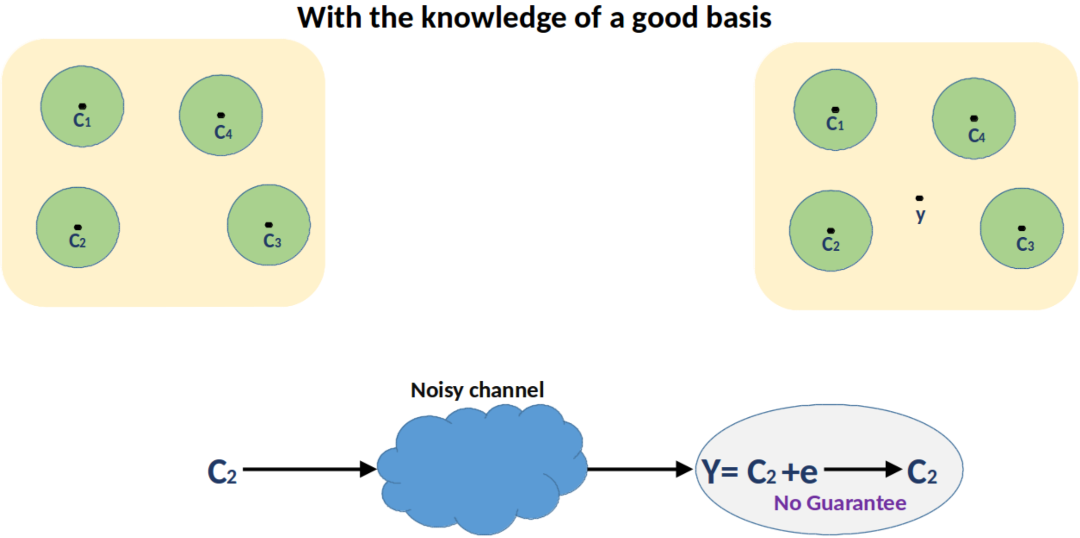
1. **Generalized Reed-Solomon (GRS)** codes '60 — One-variable polynomials
2. **Goppa** codes '70 — Sub-field sub-codes of GRS codes
3. **Reed-Muller** codes '54 — Multivariate polynomials
4. **Gabidulin** codes '85 — Linearized polynomials with one variable.

Theory of error correcting codes



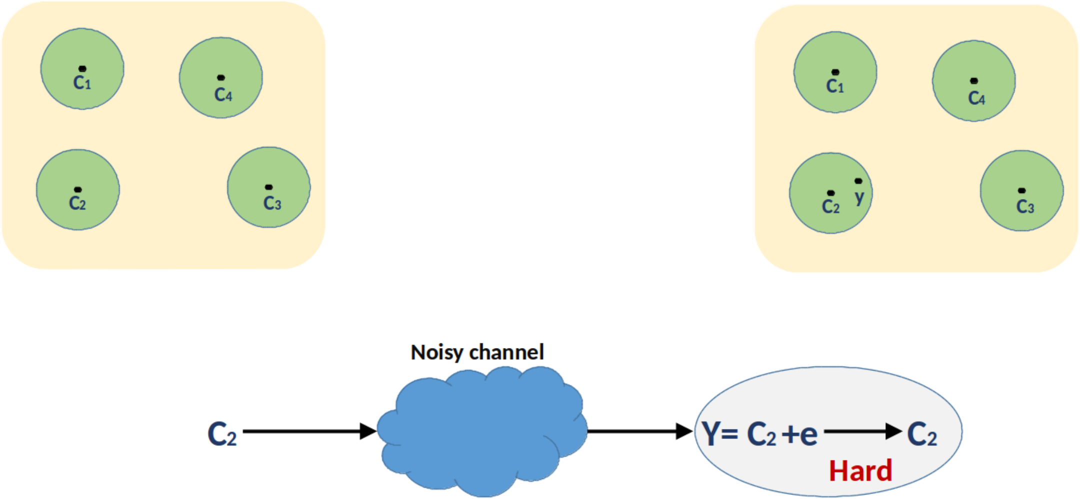
With the knowledge of a good basis




With the knowledge of a good basis
C1
C4
C2
C3
C1
C4
y
C2
C3
Noisy channel
C2
Y= C2 +e
C2
No Guarantee

Theory of error correcting codes

Without the knowledge of a good basis



McEliece Cryptosystem ('78)

1. Use code in Hamming metric
2. Based on linear codes equipped with an efficient decoding algorithm
   - Public key = random basis
   - Private key = decoding algorithm (good basis)
3. McEliece proposed binary Goppa codes

Security assumptions

- Indistinguishability of Goppa codes **Courtois-Finiasz-Sendrier '01**
- Hardness of decoding a "random" linear code

McEliece Cryptosystem

McEliece Cryptosystem ('78)

1. Use code in Hamming metric
2. Based on linear codes equipped with an efficient decoding algorithm
   - Public key = random basis
   - Private key = decoding algorithm (good basis)
3. McEliece proposed binary Goppa codes

Security assumptions

- Indistinguishability of Goppa codes **Courtois-Finiasz-Sendrier '01**
- Hardness of decoding a "random" linear code

McEliece Cryptosystem ('78)

1. Use code in Hamming metric
2. Based on linear codes equipped with an efficient decoding algorithm
   - Public key = random basis
   - Private key = decoding algorithm (good basis)
3. McEliece proposed binary Goppa codes

Security assumptions

- Indistinguishability of Goppa codes **Courtois-Finiasz-Sendrier '01**
- Hardness of decoding a "random" linear code

McEliece Cryptosystem ('78)

1. Use code in Hamming metric
2. Based on linear codes equipped with an efficient decoding algorithm
   - Public key = random basis
   - Private key = decoding algorithm (good basis)
3. McEliece proposed binary Goppa codes

Security assumptions

- Indistinguishability of Goppa codes **Courtois-Finiasz-Sendrier '01**
- Hardness of decoding a "random" linear code

McEliece Cryptosystem

McEliece Cryptosystem ('78)

1. Use code in Hamming metric
2. Based on linear codes equipped with an efficient decoding algorithm
   - Public key = random basis
   - Private key = decoding algorithm (good basis)
3. McEliece proposed binary Goppa codes

Security assumptions

- Indistinguishability of Goppa codes **Courtois-Finiasz-Sendrier '01**
- Hardness of decoding a "random" linear code

McEliece Cryptosystem

McEliece Cryptosystem ('78)

1. Use code in Hamming metric
2. Based on linear codes equipped with an efficient decoding algorithm
   - Public key = random basis
   - Private key = decoding algorithm (good basis)
3. McEliece proposed binary Goppa codes

Security assumptions

- Indistinguishability of Goppa codes **Courtois-Finiasz-Sendrier '01**
- Hardness of decoding a "random" linear code

McEliece Cryptosystem ('78)

1. Use code in Hamming metric
2. Based on linear codes equipped with an efficient decoding algorithm
   - Public key = random basis
   - Private key = decoding algorithm (good basis)
3. McEliece proposed binary Goppa codes

Security assumptions

- Indistinguishability of Goppa codes **Courtois-Finiasz-Sendrier '01**
- Hardness of decoding a "random" linear code

McEliece Cryptosystem

McEliece Cryptosystem ('78)

1. Use code in Hamming metric
2. Based on linear codes equipped with an efficient decoding algorithm
   - Public key = random basis
   - Private key = decoding algorithm (good basis)
3. McEliece proposed binary Goppa codes

Security assumptions

- Indistinguishability of Goppa codes **Courtois-Finiasz-Sendrier '01**
- Hardness of decoding a "random" linear code

McEliece Cryptosystem

Advantages

1. Encryption and decryption are very fast
2. No efficient attack
3. Candidate for Post-Quantum Cryptography

Drawbacks

1. Enormous size of the Public Key : More than 460 000 bits for a security level of only 80 bits.

McEliece Cryptosystem

Advantages

1. Encryption and decryption are very fast
2. No efficient attack
3. Candidate for Post-Quantum Cryptography

Drawbacks

1. Enormous size of the Public Key : More than 460 000 bits for a security level of only 80 bits.

McEliece Cryptosystem

Advantages

1. Encryption and decryption are very fast
2. No efficient attack
3. Candidate for Post-Quantum Cryptography

Drawbacks

1. Enormous size of the Public Key : More than 460 000 bits for a security level of only 80 bits.

McEliece Cryptosystem

Advantages

1. Encryption and decryption are very fast
2. No efficient attack
3. Candidate for Post-Quantum Cryptography

Drawbacks

1. Enormous size of the Public Key : More than 460 000 bits for a security level of only 80 bits.

McEliece Cryptosystem

Advantages

1. Encryption and decryption are very fast
2. No efficient attack
3. Candidate for Post-Quantum Cryptography

Drawbacks

1. Enormous size of the Public Key : More than 460 000 bits for a security level of only 80 bits.

McEliece Cryptosystem - Reduction of key size

Use another family of code

1. GRS codes by **Niederreiter '86**
2. Reed-Muller codes by **Sidelnikov '94**
3. Algebraic geometric codes by **Janwa-Moreno '96**
4. LDPC codes by **Monico-Rosenthal-Shokrollahi '00**
5. Wild Goppa (non-binary) by **Bernstein-Lange-Peters '10**
6. Polar codes by **Shrestha-Kim '14**

Use another family of code

1. GRS codes by **Niederreiter '86**
2. Reed-Muller codes by **Sidelnikov '94**
3. Algebraic geometric codes by **Janwa-Moreno '96**
4. LDPC codes by **Monico-Rosenthal-Shokrollahi '00**
5. Wild Goppa (non-binary) by **Bernstein-Lange-Peters '10**
6. Polar codes by **Shrestha-Kim '14**

McEliece Cryptosystem - Reduction of key size

Use another family of code

1. GRS codes by **Niederreiter '86**
2. Reed-Muller codes by **Sidelnikov '94**
3. Algebraic geometric codes by **Janwa-Moreno '96**
4. LDPC codes by **Monico-Rosenthal-Shokrollahi '00**
5. Wild Goppa (non-binary) by **Bernstein-Lange-Peters '10**
6. Polar codes by **Shrestha-Kim '14**

McEliece Cryptosystem - Reduction of key size

Use another family of code

1. GRS codes by **Niederreiter '86**
2. Reed-Muller codes by **Sidelnikov '94**
3. Algebraic geometric codes by **Janwa-Moreno '96**
4. LDPC codes by **Monico-Rosenthal-Shokrollahi '00**
5. Wild Goppa (non-binary) by **Bernstein-Lange-Peters '10**
6. Polar codes by **Shrestha-Kim '14**

McEliece Cryptosystem - Reduction of key size

Use another family of code

1. GRS codes by **Niederreiter '86**
2. Reed-Muller codes by **Sidelnikov '94**
3. Algebraic geometric codes by **Janwa-Moreno '96**
4. LDPC codes by **Monico-Rosenthal-Shokrollahi '00**
5. Wild Goppa (non-binary) by **Bernstein-Lange-Peters '10**
6. Polar codes by **Shrestha-Kim '14**

McEliece Cryptosystem - Reduction of key size

Use more structured codes

1. Quasi-cyclic BCH codes : **Gaborit '05**
2. Quasi-cyclic LDPC codes : **Baldi-Chiaraluce '07**
3. Quasi-cyclic alternant codes : **Berger-Cayrel-Gaborit-Otmani '09**
4. Quasi-dyadic Goppa codes : **Misoczki-Barreto '09**
5. Quasi-cyclic MDPC codes : **Misoczki-Tillich-Sendrier-Barreto '13**

McEliece Cryptosystem - Reduction of key size

Use more structured codes

1. Quasi-cyclic BCH codes : **Gaborit '05**
2. Quasi-cyclic LDPC codes : **Baldi-Chiaraluce '07**
3. Quasi-cyclic alternant codes : **Berger-Cayrel-Gaborit-Otmani '09**
4. Quasi-dyadic Goppa codes : **Misoczki-Barreto '09**
5. Quasi-cyclic MDPC codes : **Misoczki-Tillich-Sendrier-Barreto '13**

McEliece Cryptosystem - Reduction of key size

Use more structured codes

1. Quasi-cyclic BCH codes : **Gaborit '05**
2. Quasi-cyclic LDPC codes : **Baldi-Chiaraluce '07**
3. Quasi-cyclic alternant codes : **Berger-Cayrel-Gaborit-Otmani '09**
4. Quasi-dyadic Goppa codes : **Misoczki-Barreto '09**
5. Quasi-cyclic MDPC codes : **Misoczki-Tillich-Sendrier-Barreto '13**

McEliece Cryptosystem - Reduction of key size

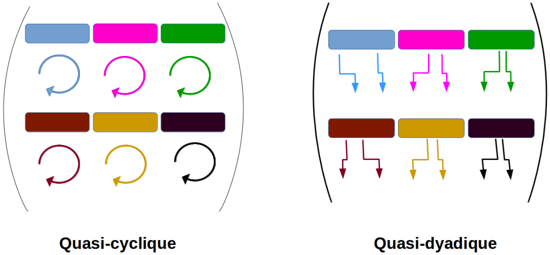
Use more structured codes

1. Quasi-cyclic BCH codes : **Gaborit '05**
2. Quasi-cyclic LDPC codes : **Baldi-Chiaraluce '07**
3. Quasi-cyclic alternant codes : **Berger-Cayrel-Gaborit-Otmani '09**
4. Quasi-dyadic Goppa codes : **Misoczki-Barreto '09**
5. Quasi-cyclic MDPC codes : **Misoczki-Tillich-Sendrier-Barreto '13**

McEliece Cryptosystem - Reduction of key size

Use more structured codes

1. Quasi-cyclic BCH codes : **Gaborit '05**
2. Quasi-cyclic LDPC codes : **Baldi-Chiaraluce '07**
3. Quasi-cyclic alternant codes : **Berger-Cayrel-Gaborit-Otmani '09**
4. Quasi-dyadic Goppa codes : **Misoczki-Barreto '09**
5. Quasi-cyclic MDPC codes : **Misoczki-Tillich-Sendrier-Barreto '13**



Several families do not behave like random codes

Example: GRS Codes - Distinguisher based on code product

- Schur / Star product of $\vec{a} = (a_1, \dots, a_n)$, $\vec{b} = (b_1, \dots, b_n) \in \mathbb{F}_q^n$

$$\vec{a} \star \vec{b} \stackrel{def}{=} (a_1 b_1, \dots, a_n b_n)$$

- $\mathscr{A}$ and $\mathscr{B}$ are two codes of length n.
- $\mathscr{A} \star \mathscr{B} \stackrel{\text{def}}{=} \left\{ \vec{a} \star \vec{b} : \vec{a} \in \mathscr{A}, \vec{b} \in \mathscr{B} \right\}$
- $\mathscr{B} = \mathscr{A} \rightarrow \mathscr{A}^2$

- "Random" code $\mathscr{A}$

$$\dim(\mathscr{A}^2) = \binom{\dim(\mathscr{A}) + 1}{2}$$

- GRS code

$$\dim(GRS^2) = 2\dim(GRS) - 1$$

Several families do not behave like random codes

Example: GRS Codes - Distinguisher based on code product

- Schur / Star product of $\vec{a} = (a_1, ..., a_n)$, $\vec{b} = (b_1, ..., b_n) \in \mathbb{F}_q^n$

$$\vec{a} \star \vec{b} \overset{def}{=} (a_1 b_1, ..., a_n b_n)$$

- $\mathscr{A}$ and $\mathscr{B}$ are two codes of length n.
- $\mathscr{A} \star \mathscr{B} \overset{\mathrm{def}}{=} \left\{ \vec{a} \star \vec{b} : \vec{a} \in \mathscr{A}, \vec{b} \in \mathscr{B} \right\}$
- $\mathscr{B} = \mathscr{A} \rightarrow \mathscr{A}^2$

- "Random" code $\mathscr{A}$

$$\dim(\mathscr{A}^2) = \binom{\dim(\mathscr{A}) + 1}{2}$$

- GRS code

$$\dim(GRS^2) = 2\dim(GRS) - 1$$

Several families do not behave like random codes

Example: GRS Codes - Distinguisher based on code product

- Schur / Star product of $\vec{a} = (a_1, \ldots, a_n),\ \vec{b} = (b_1, \ldots, b_n) \in \mathbb{F}_q^n$

$$\vec{a} \star \vec{b} \stackrel{def}{=} (a_1 b_1, \ldots, a_n b_n)$$

- $\mathscr{A}$ and $\mathscr{B}$ are two codes of length n.
- $\mathscr{A} \star \mathscr{B} \stackrel{\text{def}}{=} \left\{ \vec{a} \star \vec{b} : \vec{a} \in \mathscr{A}, \vec{b} \in \mathscr{B} \right\}$
- $\mathscr{B} = \mathscr{A} \rightarrow \mathscr{A}^2$

- "Random" code $\mathscr{A}$

$$\dim(\mathscr{A}^2) = \binom{\dim(\mathscr{A}) + 1}{2}$$

- GRS code

$$\dim(GRS^2) = 2\dim(GRS) - 1$$

Several families do not behave like random codes

Example: GRS Codes - Distinguisher based on code product

- Schur / Star product of $\vec{a} = (a_1, \ldots, a_n),\ \vec{b} = (b_1, \ldots, b_n) \in \mathbb{F}_q^n$

$$\vec{a} \star \vec{b} \overset{def}{=} (a_1 b_1, \ldots, a_n b_n)$$

- $\mathscr{A}$ and $\mathscr{B}$ are two codes of length n.
- $\mathscr{A} \star \mathscr{B} \overset{\text{def}}{=} \left\{ \vec{a} \star \vec{b} : \vec{a} \in \mathscr{A}, \vec{b} \in \mathscr{B} \right\}$
- $\mathscr{B} = \mathscr{A} \to \mathscr{A}^2$

- "Random" code $\mathscr{A}$

$$\dim(\mathscr{A}^2) = \binom{\dim(\mathscr{A}) + 1}{2}$$

- GRS code

$$\dim(GRS^2) = 2\dim(GRS) - 1$$

McEliece Cryptosystem - Reduction of key size

Several families do not behave like random codes

Example: GRS Codes - Distinguisher based on code product

- Schur / Star product of $\vec{a} = (a_1, \dots, a_n)$, $\vec{b} = (b_1, \dots, b_n) \in \mathbb{F}_q^n$

$$\vec{a} \star \vec{b} \overset{def}{=} (a_1 b_1, \dots, a_n b_n)$$

- $\mathscr{A}$ and $\mathscr{B}$ are two codes of length n.
- $\mathscr{A} \star \mathscr{B} \overset{\text{def}}{=} \left\{ \vec{a} \star \vec{b} : \vec{a} \in \mathscr{A}, \vec{b} \in \mathscr{B} \right\}$
- $\mathscr{B} = \mathscr{A} \rightarrow \mathscr{A}^2$

- "Random" code $\mathscr{A}$

$$\dim(\mathscr{A}^2) = \binom{\dim(\mathscr{A}) + 1}{2}$$

- GRS code

$$\dim(GRS^2) = 2\dim(GRS) - 1$$

McEliece Cryptosystem - Reduction of key size

Date	Scheme	Attack	Complexity
1994	GRS	Sidelnikov-Shestakov	polynomial
2007	Reed-Muller	Minder-Shokrollahi	Sub-exponential
2013	GRS	Couvreur-Gaborit-Gauthier-Otmani-Tillich	polynomial
2010	quasi-cyclic alternants	Faugère-Otmani-Tillich	polynomial
2013	Reed-Muller	Chizhov-Borodin	polynomial
2014	Wild Goppa (non-binary) $m = 2$	Couvreur-Otmani-Tillich	polynomial
2014	AG Codes	Couvreur-Màrquez Corbella-Pellikaan	polynomial
2014	quasi-dyadic Goppa	Faugère-Otmani-Perret-Portzamparc-Tillich	polynomial
2014	AG codes	Couvreur-Màrquez Corbella-Pellikaan	polynomial

New masking techniques

Adding some randomness

- Berger-Loidreau '05 $\rightarrow$ Random subcode of a GRS
  - ⋆ Wieschebrink '10: Square code based attack.
- Wieschebrink '06 $\rightarrow$ Random columns with GRS
  - ⋆ Couvreur-Gaborit-Gauthier-Otmani-Tillich '14: Square code based attack.
- Gueye-Mboup '13 $\rightarrow$ Random columns with Reed-Muller codes
  - ⋆ Otmani-Tale '15: Square code based attack.

New masking techniques

Adding some randomness

- Berger-Loidreau '05 → Random subcode of a GRS
  - ⋆ Wieschebrink '10: Square code based attack.
- Wieschebrink '06 → Random columns with GRS
  - ⋆ Couvreur-Gaborit-Gauthier-Otmani-Tillich '14: Square code based attack.
- Gueye-Mboup '13 → Random columns with Reed-Muller codes
  - ⋆ Otmani-Tale '15: Square code based attack.

New masking techniques

Adding some randomness

- Berger-Loidreau '05 → Random subcode of a GRS
  - ⋆ Wieschebrink '10: Square code based attack.
- Wieschebrink '06 → Random columns with GRS
  - ⋆ Couvreur-Gaborit-Gauthier-Otmani-Tillich '14: Square code based attack.
- Gueye-Mboup '13 → Random columns with Reed-Muller codes
  - ⋆ Otmani-Tale '15: Square code based attack.

New masking techniques

Adding some randomness

- Berger-Loidreau '05 $\to$ Random subcode of a GRS
  - $\star$ Wieschebrink '10: Square code based attack.
- Wieschebrink '06 $\to$ Random columns with GRS
  - $\star$ Couvreur-Gaborit-Gauthier-Otmani-Tillich '14: Square code based attack.
- Gueye-Mboup '13 $\to$ Random columns with Reed-Muller codes
  - $\star$ Otmani-Tale '15: Square code based attack.

New masking techniques

Adding some randomness

- Berger-Loidreau '05 → Random subcode of a GRS
  - ⋆ Wieschebrink '10: Square code based attack.
- Wieschebrink '06 → Random columns with GRS
  - ⋆ Couvreur-Gaborit-Gauthier-Otmani-Tillich '14: Square code based attack.
- Gueye-Mboup '13 → Random columns with Reed-Muller codes
  - ⋆ Otmani-Tale '15: Square code based attack.

New masking techniques

Adding some randomness

- Berger-Loidreau '05 → Random subcode of a GRS
  - ⋆ Wieschebrink '10: Square code based attack.
- Wieschebrink '06 → Random columns with GRS
  - ⋆ Couvreur-Gaborit-Gauthier-Otmani-Tillich '14: Square code based attack.
- Gueye-Mboup '13 → Random columns with Reed-Muller codes
  - ⋆ Otmani-Tale '15: Square code based attack.

New masking techniques

Adding some randomness

- Berger-Loidreau '05 → Random subcode of a GRS
  - ⋆ Wieschebrink '10: Square code based attack.
- Wieschebrink '06 → Random columns with GRS
  - ⋆ Couvreur-Gaborit-Gauthier-Otmani-Tillich '14: Square code based attack.
- Gueye-Mboup '13 → Random columns with Reed-Muller codes
  - ⋆ Otmani-Tale '15: Square code based attack.

Rank metric cryptography

Gabidulin-Paramonov-Tretjakov cryptosystem '91

1. Rank metric with Gabidulin codes
2. But many attacks
   - Gibson's attacks '95, '96
   - Overbeck's attack '05

Some GPT Variants

- **Gabidulin '08**
- **Rashwan-Gabidulin-Honary '10**

Rank metric cryptography

Gabidulin-Paramonov-Tretjakov cryptosystem '91

1. Rank metric with Gabidulin codes
2. But many attacks
   - Gibson's attacks '95, '96
   - Overbeck's attack '05

Some GPT Variants

- **Gabidulin '08**
- **Rashwan-Gabidulin-Honary '10**

Rank metric cryptography

Gabidulin-Paramonov-Tretjakov cryptosystem '91

1. Rank metric with Gabidulin codes
2. But many attacks
   - Gibson's attacks '95, '96
   - Overbeck's attack '05

Some GPT Variants

- **Gabidulin '08**
- **Rashwan-Gabidulin-Honary '10**

Rank metric cryptography

Gabidulin-Paramonov-Tretjakov cryptosystem '91

1. Rank metric with Gabidulin codes
2. But many attacks
   - Gibson's attacks '95, '96
   - Overbeck's attack '05

Some GPT Variants

- **Gabidulin '08**
- **Rashwan-Gabidulin-Honary '10**

Rank metric cryptography

Gabidulin-Paramonov-Tretjakov cryptosystem ’91

1. Rank metric with Gabidulin codes
2. But many attacks
   - Gibson’s attacks ’95, ’96
   - Overbeck’s attack ’05

Some GPT Variants

- Gabidulin ’08
- Rashwan-Gabidulin-Honary ’10

Rank metric cryptography

Gabidulin-Paramonov-Tretjakov cryptosystem ’91

1. Rank metric with Gabidulin codes
2. But many attacks
   - Gibson’s attacks ’95, ’96
   - Overbeck’s attack ’05

Some GPT Variants

- **Gabidulin ’08**
- **Rashwan-Gabidulin-Honary ’10**

Rank metric cryptography

Some recent progress in rank metric

- ⋆ Identity based encryption scheme : **Deneuville-Gaborit-Zémor '17**
- ⋆ Key exchange protocol : **Gaborit-Hauteville-Phan-Tillich '17**
- ⋆ New encryption scheme : **Loidreau '17**
- ⋆ Group signature : **Alamélou-Blazy-Cauchie-Gaborit '16**
- ⋆ Pseudo random generator : **Gaborit-Hauteville-Tillich '16**
- ⋆ Encryption and signature scheme based on LRPC : **Gaborit-Ruatta-Schrek-Zémor '14**

Some recent progress in rank metric

- ⋆ Identity based encryption scheme : **Deneuville-Gaborit-Zémor '17**
- ⋆ Key exchange protocol : **Gaborit-Hauteville-Phan-Tillich '17**
- ⋆ New encryption scheme : **Loidreau '17**
- ⋆ Group signature : **Alamélou-Blazy-Cauchie-Gaborit '16**
- ⋆ Pseudo random generator : **Gaborit-Hauteville-Tillich '16**
- ⋆ Encryption and signature scheme based on LRPC : **Gaborit-Ruatta-Schrek-Zémor '14**

Some recent progress in rank metric

- ⋆ Identity based encryption scheme : **Deneuville-Gaborit-Zémor '17**
- ⋆ Key exchange protocol : **Gaborit-Hauteville-Phan-Tillich '17**
- ⋆ New encryption scheme : **Loidreau '17**
- ⋆ Group signature : **Alamélou-Blazy-Cauchie-Gaborit '16**
- ⋆ Pseudo random generator : **Gaborit-Hauteville-Tillich '16**
- ⋆ Encryption and signature scheme based on LRPC : **Gaborit-Ruatta-Schrek-Zémor '14**

Rank metric cryptography

Some recent progress in rank metric

- ⋆ Identity based encryption scheme : **Deneuville-Gaborit-Zémor ’17**
- ⋆ Key exchange protocol : **Gaborit-Hauteville-Phan-Tillich ’17**
- ⋆ New encryption scheme : **Loidreau ’17**
- ⋆ Group signature : **Alamélou-Blazy-Cauchie-Gaborit ’16**
- ⋆ Pseudo random generator : **Gaborit-Hauteville-Tillich ’16**
- ⋆ Encryption and signature scheme based on LRPC : **Gaborit-Ruatta-Schrek-Zémor ’14**

Some recent progress in rank metric

- ⋆ Identity based encryption scheme : **Deneuville-Gaborit-Zémor '17**
- ⋆ Key exchange protocol : **Gaborit-Hauteville-Phan-Tillich '17**
- ⋆ New encryption scheme : **Loidreau '17**
- ⋆ Group signature : **Alamélou-Blazy-Cauchie-Gaborit '16**
- ⋆ Pseudo random generator : **Gaborit-Hauteville-Tillich '16**
- ⋆ Encryption and signature scheme based on LRPC : **Gaborit-Ruatta-Schrek-Zémor '14**

Rank metric cryptography

Some recent progress in rank metric

- ⋆ Identity based encryption scheme : **Deneuville-Gaborit-Zémor ’17**
- ⋆ Key exchange protocol : **Gaborit-Hauteville-Phan-Tillich ’17**
- ⋆ New encryption scheme : **Loidreau ’17**
- ⋆ Group signature : **Alamélou-Blazy-Cauchie-Gaborit ’16**
- ⋆ Pseudo random generator : **Gaborit-Hauteville-Tillich ’16**
- ⋆ Encryption and signature scheme based on LRPC : **Gaborit-Ruatta-Schrek-Zémor ’14**

Rank metric cryptography

Some recent progress in rank metric

- ⋆ Identity based encryption scheme : **Deneuville-Gaborit-Zémor '17**
- ⋆ Key exchange protocol : **Gaborit-Hauteville-Phan-Tillich '17**
- ⋆ New encryption scheme : **Loidreau '17**
- ⋆ Group signature : **Alamélou-Blazy-Cauchie-Gaborit '16**
- ⋆ Pseudo random generator : **Gaborit-Hauteville-Tillich '16**
- ⋆ Encryption and signature scheme based on LRPC : **Gaborit-Ruatta-Schrek-Zémor '14**

Rank metric cryptography

Some recent progress in rank metric

- ⋆ Identity based encryption scheme : **Deneuville-Gaborit-Zémor '17**
- ⋆ Key exchange protocol : **Gaborit-Hauteville-Phan-Tillich '17**
- ⋆ New encryption scheme : **Loidreau '17**
- ⋆ Group signature : **Alamélou-Blazy-Cauchie-Gaborit '16**
- ⋆ Pseudo random generator : **Gaborit-Hauteville-Tillich '16**
- ⋆ Encryption and signature scheme based on LRPC : **Gaborit-Ruatta-Schrek-Zémor '14**

Outline

Example of isometry for rank metric

- $\vec{x} \in \mathbb{F}_{q^m}^n$
- $\boldsymbol{T} \in \mathrm{GL}_n(\mathbb{F}_q)$

$$\|\vec{x}\boldsymbol{T}\|_q = \|\vec{x}\|_q$$

Definition 1 (Gabidulin code)

- $\vec{g} \in \mathbb{F}_{q^m}^n$ with $\|\vec{g}\|_q = n$

The $(n, k)-$Gabidulin code $\mathscr{G}_k(\vec{g})$ is the code generated by:

$$\boldsymbol{G} = \begin{pmatrix} g_1^{q^0} & g_2^{q^0} & \cdots & g_n^{q^0} \\ g_1^{q^1} & g_2^{q^1} & \cdots & g_n^{q^1} \\ \vdots & \vdots & \ddots & \vdots \\ g_1^{q^{k-1}} & g_2^{q^{k-1}} & \cdots & g_n^{q^{k-1}} \end{pmatrix}$$

$\vec{g}$ is called generator vector of $\mathscr{G}_k(\vec{g})$.

Gabidulin codes

Proposition 1

1. *The correction capability of a Gabidulin code* $\mathscr{G}_k(\vec{g})$ *is* $\lfloor \frac{n-k}{2} \rfloor$
2. $\mathscr{G}_k(\vec{g})^\perp$ *is also a Gabidulin code.*

The dual $\mathscr{C}^\perp$ of a code $\mathscr{C}$ is the v.s.s

$$\mathscr{C}^\perp = \{\vec{y} \in \mathbb{F}^n \ : \ \forall\ \vec{c} \in \mathscr{C},\ \langle \vec{c}, \vec{y} \rangle = 0\} \text{ with } \langle \vec{c}, \vec{y} \rangle = \sum_{i=1}^{n} c_i y_i$$

Gabidulin codes

Proposition 1

1. *The correction capability of a Gabidulin code* $\mathscr{G}_k(\vec{g})$ *is* $\lfloor \frac{n-k}{2} \rfloor$
2. $\mathscr{G}_k(\vec{g})^{\perp}$ *is also a Gabidulin code.*

The dual $\mathscr{C}^{\perp}$ of a code $\mathscr{C}$ is the v.s.s

$$\mathscr{C}^{\perp} = \{\vec{y} \in \mathbb{F}^n \;:\; \forall\, \vec{c} \in \mathscr{C},\; \langle \vec{c}, \vec{y} \rangle = 0\} \text{ with } \langle \vec{c}, \vec{y} \rangle = \sum_{i=1}^{n} c_i y_i$$

Gabidulin codes

Proposition 2

- $\mathscr{G}_k(\vec{g})$ *a* $(n, k)-$*Gabidulin code on* $\mathbb{F}_{q^m}$
- $\boldsymbol{T} \in \mathrm{GL}_n(\mathbb{F}_q)$

$$\mathscr{G}_k(\vec{g})\,\boldsymbol{T} = \mathscr{G}_k(\vec{g}\,\boldsymbol{T})$$

Proof.

For the proof, remark that

$$(\vec{g}\,\boldsymbol{T})^{q^i} = \vec{g}^{q^i}\boldsymbol{T} \quad \text{since} \quad \boldsymbol{T}^{q^i} = \boldsymbol{T}$$

for any integer i. □ □

Proposition 2

- $\mathscr{G}_k(\vec{g})$ *a* $(n, k)-$*Gabidulin code on* $\mathbb{F}_{q^m}$
- $\boldsymbol{T} \in \mathrm{GL}_n(\mathbb{F}_q)$

$$\mathscr{G}_k(\vec{g})\,\boldsymbol{T} = \mathscr{G}_k(\vec{g}\boldsymbol{T})$$

Proof.

For the proof, remark that

$$(\vec{g}\boldsymbol{T})^{q^i} = \vec{g}^{q^i}\boldsymbol{T} \quad \text{since} \quad \boldsymbol{T}^{q^i} = \boldsymbol{T}$$

for any integer i. □ □

Plan

GPT Cryptosystem

Key generation.

- $\boldsymbol{G} \in \mathcal{M}_{k \times n}(\mathbb{F}_{q^m})$ a generator matrix of $\mathscr{G}_k(\vec{g})$
- Pick at random $\boldsymbol{S} \in \mathrm{GL}_k(\mathbb{F}_{q^m})$.
- Pick a random matrix $\boldsymbol{X} \in \mathcal{M}_{k \times \ell}(\mathbb{F}_{q^m})$
- $\boldsymbol{P} \in \mathrm{GL}_{n+\ell}(\mathbb{F}_q)$ be a random non-singular matrix
- Compute

$$\boldsymbol{G}_{pub} \stackrel{\text{def}}{=} \boldsymbol{S}(\boldsymbol{X} \mid \boldsymbol{G})\boldsymbol{P}^{-1} \tag{1}$$

The public key is $(\boldsymbol{G}_{pub}, t)$ where $t \stackrel{\text{def}}{=} \lfloor \frac{n-k}{2} \rfloor$

GPT Cryptosystem

Key generation.

- $\boldsymbol{G} \in \mathcal{M}_{k \times n}(\mathbb{F}_{q^m})$ a generator matrix of $\mathscr{G}_k(\vec{g})$
- Pick at random $\boldsymbol{S} \in \mathrm{GL}_k(\mathbb{F}_{q^m})$.
- Pick a random matrix $\boldsymbol{X} \in \mathcal{M}_{k \times \ell}(\mathbb{F}_{q^m})$
- $\boldsymbol{P} \in \mathrm{GL}_{n+\ell}(\mathbb{F}_q)$ be a random non-singular matrix
- Compute

$$\boldsymbol{G}_{pub} \stackrel{\text{def}}{=} \boldsymbol{S}(\boldsymbol{X} \mid \boldsymbol{G})\boldsymbol{P}^{-1} \quad (1)$$

The public key is $(\boldsymbol{G}_{pub}, t)$ where $t \stackrel{\text{def}}{=} \lfloor \frac{n-k}{2} \rfloor$

Key generation.

- $\boldsymbol{G} \in \mathcal{M}_{k\times n}(\mathbb{F}_{q^m})$ a generator matrix of $\mathscr{G}_k(\vec{g})$
- Pick at random $\boldsymbol{S} \in \mathrm{GL}_k(\mathbb{F}_{q^m})$.
- Pick a random matrix $\boldsymbol{X} \in \mathcal{M}_{k\times \ell}(\mathbb{F}_{q^m})$
- $\boldsymbol{P} \in \mathrm{GL}_{n+\ell}(\mathbb{F}_q)$ be a random non-singular matrix
- Compute

$$\boldsymbol{G}_{pub} \stackrel{\text{def}}{=} \boldsymbol{S}(\boldsymbol{X} \mid \boldsymbol{G})\boldsymbol{P}^{-1} \tag{1}$$

The public key is $(\boldsymbol{G}_{pub}, t)$ where $t \stackrel{\text{def}}{=} \lfloor \frac{n-k}{2} \rfloor$

Key generation.

- $\boldsymbol{G} \in \mathcal{M}_{k\times n}(\mathbb{F}_{q^m})$ a generator matrix of $\mathscr{G}_k(\vec{g})$
- Pick at random $\boldsymbol{S} \in \mathrm{GL}_k(\mathbb{F}_{q^m})$.
- Pick a random matrix $\boldsymbol{X} \in \mathcal{M}_{k\times\ell}(\mathbb{F}_{q^m})$
- $\boldsymbol{P} \in \mathrm{GL}_{n+\ell}(\mathbb{F}_q)$ be a random non-singular matrix
- Compute

$$\boldsymbol{G}_{pub} \stackrel{\text{def}}{=} \boldsymbol{S}(\boldsymbol{X} \mid \boldsymbol{G})\boldsymbol{P}^{-1} \quad (1)$$

The public key is $(\boldsymbol{G}_{pub}, t)$ where $t \stackrel{\text{def}}{=} \lfloor \frac{n-k}{2} \rfloor$

GPT Cryptosystem

Key generation.

- $\boldsymbol{G} \in \mathcal{M}_{k \times n}(\mathbb{F}_{q^m})$ a generator matrix of $\mathscr{G}_k(\vec{g})$
- Pick at random $\boldsymbol{S} \in \mathrm{GL}_k(\mathbb{F}_{q^m})$.
- Pick a random matrix $\boldsymbol{X} \in \mathcal{M}_{k \times \ell}(\mathbb{F}_{q^m})$
- $\boldsymbol{P} \in \mathrm{GL}_{n+\ell}(\mathbb{F}_q)$ be a random non-singular matrix
- Compute

$$\boldsymbol{G}_{pub} \stackrel{\text{def}}{=} \boldsymbol{S}(\boldsymbol{X} \mid \boldsymbol{G})\boldsymbol{P}^{-1} \quad (1)$$

The public key is $(\boldsymbol{G}_{pub}, t)$ where $t \stackrel{\text{def}}{=} \lfloor \frac{n-k}{2} \rfloor$

GPT Cryptosystem

Key generation.

- $\boldsymbol{G} \in \mathcal{M}_{k \times n}(\mathbb{F}_{q^m})$ a generator matrix of $\mathscr{G}_k(\vec{g})$
- Pick at random $\boldsymbol{S} \in \mathrm{GL}_k(\mathbb{F}_{q^m})$.
- Pick a random matrix $\boldsymbol{X} \in \mathcal{M}_{k \times \ell}(\mathbb{F}_{q^m})$
- $\boldsymbol{P} \in \mathrm{GL}_{n+\ell}(\mathbb{F}_q)$ be a random non-singular matrix
- Compute

$$\boldsymbol{G}_{pub} \stackrel{\text{def}}{=} \boldsymbol{S}(\boldsymbol{X} \mid \boldsymbol{G})\boldsymbol{P}^{-1} \tag{1}$$

The public key is $(\boldsymbol{G}_{pub}, t)$ where $t \stackrel{\text{def}}{=} \lfloor \frac{n-k}{2} \rfloor$

GPT Cryptosystem

Encryption.

To encrypt a message $\vec{m} \in \mathbb{F}_{q^m}^k$,

1. Generate $\vec{e} \in \mathbb{F}_{q^m}^n$ such that $\|\vec{e}\|_q \leqslant t$.
2. The cipher-text is the vector

$$\vec{c} = \vec{m}\boldsymbol{G}_{pub} + \vec{e}$$

Decryption.

1. Compute $\vec{c}\boldsymbol{P}$ $\qquad \vec{m}\boldsymbol{S}\left(\boldsymbol{X} \mid \boldsymbol{G}\right) + \vec{e}\boldsymbol{P}$
2. And $\vec{y} = Dec_{\cdot(\boldsymbol{X}|\boldsymbol{G})}(\vec{c}\boldsymbol{P})$ $\qquad \vec{y} = \vec{m}\boldsymbol{S}$ since $\|\vec{e}\boldsymbol{P}\|_q = \|\vec{e}\|_q \leqslant t$
3. Return $\vec{m}' = \vec{y}\boldsymbol{S}^{-1}$ $\qquad \vec{m}' = \vec{m}$

GPT Cryptosystem

Encryption.

To encrypt a message $\vec{m} \in \mathbb{F}_{q^m}^k$,

1. Generate $\vec{e} \in \mathbb{F}_{q^m}^n$ such that $\|\vec{e}\|_q \leqslant t$.
2. The cipher-text is the vector

$$\vec{c} = \vec{m}\boldsymbol{G}_{pub} + \vec{e}$$

Decryption.

1. Compute $\vec{c}\boldsymbol{P}$ $\qquad \vec{m}\boldsymbol{S}\left(\boldsymbol{X} \mid \boldsymbol{G}\right) + \vec{e}\boldsymbol{P}$
2. And $\vec{y} = Dec_{(\boldsymbol{X}|\boldsymbol{G})}(\vec{c}\boldsymbol{P})$ $\qquad \vec{y} = \vec{m}\boldsymbol{S}$ since $\|\vec{e}\boldsymbol{P}\|_q = \|\vec{e}\|_q \leqslant t$
3. Return $\vec{m}' = \vec{y}\boldsymbol{S}^{-1}$ $\qquad \vec{m}' = \vec{m}$

GPT Cryptosystem

Encryption.

To encrypt a message $\vec{m} \in \mathbb{F}_{q^m}^k$,

1. Generate $\vec{e} \in \mathbb{F}_{q^m}^n$ such that $\|\vec{e}\|_q \leqslant t$.
2. The cipher-text is the vector

$$\vec{c} = \vec{m}\boldsymbol{G}_{pub} + \vec{e}$$

Decryption.

1. Compute $\vec{c}\boldsymbol{P}$ — $\vec{m}\boldsymbol{S}\left(\boldsymbol{X} \mid \boldsymbol{G}\right) + \vec{e}\boldsymbol{P}$
2. And $\vec{y} = Dec_{(\boldsymbol{X}|\boldsymbol{G})}(\vec{c}\boldsymbol{P})$ — $\vec{y} = \vec{m}\boldsymbol{S}$ since $\|\vec{e}\boldsymbol{P}\|_q = \|\vec{e}\|_q \leqslant t$
3. Return $\vec{m}' = \vec{y}\boldsymbol{S}^{-1}$ — $\vec{m}' = \vec{m}$

GPT Cryptosystem

Encryption.

To encrypt a message $\vec{m} \in \mathbb{F}_{q^m}^k$,

1. Generate $\vec{e} \in \mathbb{F}_{q^m}^n$ such that $\|\vec{e}\|_q \leqslant t$.
2. The cipher-text is the vector

$$\vec{c} = \vec{m}\boldsymbol{G}_{pub} + \vec{e}$$

Decryption.

1. Compute $\vec{c}\boldsymbol{P}$ — $\vec{m}\boldsymbol{S}\left(\boldsymbol{X} \mid \boldsymbol{G}\right) + \vec{e}\boldsymbol{P}$
2. And $\vec{y} = Dec_{(\boldsymbol{X}|\boldsymbol{G})}(\vec{c}\boldsymbol{P})$ — $\vec{y} = \vec{m}\boldsymbol{S}$ since $\|\vec{e}\boldsymbol{P}\|_q = \|\vec{e}\|_q \leqslant t$
3. Return $\vec{m}' = \vec{y}\boldsymbol{S}^{-1}$ — $\vec{m}' = \vec{m}$

Overbeck's Attack

Definition 2 (Distinguisher)

- f is an integer such that $f \leqslant n - k$

Define the application Λ_f by:

$$\begin{array}{rccl} \Lambda_f : & \mathbb{F}_{q^m}^n & \longrightarrow & \mathbb{F}_{q^m}^n \\ & \mathscr{U} & \longmapsto & \Lambda_f(\mathscr{U}) \stackrel{\text{def}}{=} \mathscr{U} + \mathscr{U}^{q} + \cdots + \mathscr{U}^{q^f} \end{array}$$

- For $\boldsymbol{P} \in \mathrm{GL}_n(\mathbb{F}_q)$

$$\Lambda_f(\mathscr{U}\boldsymbol{P}) = \Lambda_f(\mathscr{U})\boldsymbol{P}$$

Overbeck's Attack

Definition 2 (Distinguisher)

- f is an integer such that $f \leqslant n-k$

Define the application Λ_f by:

$$\begin{array}{rccc} \Lambda_f : & \mathbb{F}_{q^m}^n & \longrightarrow & \mathbb{F}_{q^m}^n \\ & \mathscr{U} & \longmapsto & \Lambda_f(\mathscr{U}) \stackrel{\text{def}}{=} \mathscr{U} + \mathscr{U}^{q} + \cdots + \mathscr{U}^{q^f} \end{array}$$

- For $\boldsymbol{P} \in \mathrm{GL}_n(\mathbb{F}_q)$

$$\Lambda_f(\mathscr{U}\boldsymbol{P}) = \Lambda_f(\mathscr{U})\boldsymbol{P}$$

Overbeck's Attack

Definition 2 (Distinguisher)

- f is an integer such that $f \leqslant n - k$

Define the application Λ_f by:

$$\begin{array}{rccc} \Lambda_f : & \mathbb{F}_{q^m}^n & \longrightarrow & \mathbb{F}_{q^m}^n \\ & \mathscr{U} & \longmapsto & \Lambda_f(\mathscr{U}) \stackrel{\text{def}}{=} \mathscr{U} + \mathscr{U}^q + \cdots + \mathscr{U}^{q^f} \end{array}$$

- For $\boldsymbol{P} \in \mathrm{GL}_n(\mathbb{F}_q)$

$$\Lambda_f(\mathscr{U}\boldsymbol{P}) = \Lambda_f(\mathscr{U})\boldsymbol{P}$$

Proposition 3

- $f \leqslant n - k - 1$

$$\Lambda_f(\mathscr{G}_k(\vec{g})) = \mathscr{G}_{k+f}(\vec{g})$$

In particular,

$$\dim \Lambda_f(\mathscr{G}_k(\vec{g})) = k + f$$

Theorem 3

For a "random" $(n, k)-$code $\mathscr{R}$,

$$\dim \Lambda_f(\mathscr{R}) = min\{n, k(f+1)\}$$

with a high probability.

Overbeck's Attack

Proposition 3

- $f \leqslant n - k - 1$

$$\Lambda_f(\mathscr{G}_k(\vec{g})) = \mathscr{G}_{k+f}(\vec{g})$$

In particular,

$$\dim \Lambda_f(\mathscr{G}_k(\vec{g})) = k + f$$

Theorem 3

For a "random" $(n,k)-$code $\mathscr{R}$,

$$\dim \Lambda_f(\mathscr{R}) = min\{n, k(f+1)\}$$

with a high probability.

Overbeck's Attack

Proposition 3

- $f \leqslant n - k - 1$

$$\Lambda_f(\mathscr{G}_k(\vec{g})) = \mathscr{G}_{k+f}(\vec{g})$$

In particular,

$$\dim \Lambda_f(\mathscr{G}_k(\vec{g})) = k + f$$

Theorem 3

For a "random" $(n, k)-$code $\mathscr{R}$,

$$\dim \Lambda_f(\mathscr{R}) = min\{n, k(f+1)\}$$

with a high probability.

Overbeck's Attack

Proposition 4

- *Let* $\boldsymbol{G}_{pub} = \boldsymbol{S}\left(\boldsymbol{X} \mid \boldsymbol{G}\right)\boldsymbol{P}^{-1}$ *be a generator matrix of* $\mathscr{C}_{\mathrm{pub}}$

$\Lambda_{n-k-1}\left(\mathscr{C}_{pub}\right) \subset \mathbb{F}_{q^m}^{n+\ell}$ *is generated by:*

$$\begin{pmatrix} \boldsymbol{X}_1 & \boldsymbol{G}_{n-1} \\ \boldsymbol{X}_2 & \boldsymbol{0} \end{pmatrix} \boldsymbol{P}^{-1}$$

$\boldsymbol{G}_{n-1}$ *being a generator matrix of* $\mathscr{G}_{n-1}\left(\vec{g}\right)$.

$$\dim \Lambda_{n-k-1}(\mathscr{C}_{pub}) = n - 1 + Rank\left(\boldsymbol{X}_2\right)$$

Theorem 4

If $Rank\left(\boldsymbol{X}_2\right) = \ell$,

- $$\dim \Lambda_{n-k-1}(\mathscr{C}_{pub})^{\perp} = 1$$
- $$\Lambda_{n-k-1}(\mathscr{C}_{pub})^{\perp} = < \left(0 \mid \vec{h}\right) \boldsymbol{P}^T >$$

Overbeck's Attack

$$\dim \Lambda_{n-k-1}(\mathscr{C}_{pub}) = n - 1 + Rank\,(\boldsymbol{X}_2)$$

Theorem 4

If $Rank\,(\boldsymbol{X}_2) = \ell$,

- $$\dim \Lambda_{n-k-1}(\mathscr{C}_{pub})^{\perp} = 1$$
- $$\Lambda_{n-k-1}(\mathscr{C}_{pub})^{\perp} = < \left(0 \mid \vec{h}\right) \boldsymbol{P}^T >$$

Overbeck's Attack

$$\dim \Lambda_{n-k-1}(\mathscr{C}_{pub}) = n - 1 + Rank\left(\boldsymbol{X}_2\right)$$

Theorem 4

If $Rank\left(\boldsymbol{X}_2\right) = \ell$,

- $$\dim \Lambda_{n-k-1}(\mathscr{C}_{pub})^{\perp} = 1$$
- $$\Lambda_{n-k-1}(\mathscr{C}_{pub})^{\perp} =< \left(0 \mid \vec{h}\right) \boldsymbol{P}^{T} >$$

Overbeck's Attack

$$\dim \Lambda_{n-k-1}(\mathscr{C}_{pub}) = n - 1 + Rank\left(\boldsymbol{X}_2\right)$$

Theorem 4

If $Rank\left(\boldsymbol{X}_2\right) = \ell$,

- $$\dim \Lambda_{n-k-1}(\mathscr{C}_{pub})^{\perp} = 1$$
- $$\Lambda_{n-k-1}(\mathscr{C}_{pub})^{\perp} =< \left(0 \mid \vec{h}\right) \boldsymbol{P}^{T} >$$

Overbeck's Attack

Summary

- Compute

$$\Lambda_{n-k-1}(\mathscr{C}_{pub})$$

- If

$$\dim \Lambda_{n-k-1}(\mathscr{C}_{pub})^{\perp} = 1$$

- Choose $\vec{h} \in \Lambda_{n-k-1}(\mathscr{C}_{pub})^{\perp}, \quad \vec{h} \neq \mathbf{0}$
- Find $\boldsymbol{T} \in \mathrm{GL}_{n+\ell}(\mathbb{F}_q)$ such that $\vec{h} = (\mathbf{0} \mid \vec{h}')\boldsymbol{T}$, $\vec{h}' \in \mathbb{F}_{q^m}^n$ Easy : Linear algebra

Overbeck's Attack

Summary

- Compute

$$\Lambda_{n-k-1}(\mathscr{C}_{pub})$$

- If

$$\dim \Lambda_{n-k-1}(\mathscr{C}_{pub})^{\perp} = 1$$

- Choose $\vec{h} \in \Lambda_{n-k-1}(\mathscr{C}_{pub})^{\perp}, \quad \vec{h} \neq \mathbf{0}$
- Find $\boldsymbol{T} \in \mathrm{GL}_{n+\ell}(\mathbb{F}_q)$ such that $\vec{h} = (\mathbf{0} \mid \vec{h}')\boldsymbol{T}$, $\vec{h}' \in \mathbb{F}_{q^m}^n$ Easy : Linear algebra

Overbeck's Attack

Summary

- Compute

$$\Lambda_{n-k-1}(\mathscr{C}_{pub})$$

- If

$$\dim \Lambda_{n-k-1}(\mathscr{C}_{pub})^{\perp} = 1$$

- Choose $\vec{h} \in \Lambda_{n-k-1}(\mathscr{C}_{pub})^{\perp}, \quad \vec{h} \neq \mathbf{0}$
- Find $\boldsymbol{T} \in \mathrm{GL}_{n+\ell}(\mathbb{F}_q)$ such that $\vec{h} = (\mathbf{0} \mid \vec{h}')\boldsymbol{T}$, $\vec{h}' \in \mathbb{F}_{q^m}^n$ Easy : Linear algebra

Overbeck's Attack

Summary

- Compute

$$\Lambda_{n-k-1}(\mathscr{C}_{pub})$$

- If

$$\dim \Lambda_{n-k-1}(\mathscr{C}_{pub})^{\perp} = 1$$

- Choose $\vec{h} \in \Lambda_{n-k-1}(\mathscr{C}_{pub})^{\perp}, \quad \vec{h} \neq \mathbf{0}$
- Find $\boldsymbol{T} \in \mathrm{GL}_{n+\ell}(\mathbb{F}_q)$ such that $\vec{h} = (\mathbf{0} \mid \vec{h}')\boldsymbol{T}$, $\vec{h}' \in \mathbb{F}_{q^m}^n$ Easy : Linear algebra

The success of this attack is based on two facts:

1. $\boldsymbol{P} \in \mathrm{GL}_{n+\ell}(\mathbb{F}_q)$
2. $\boldsymbol{X}_2$ must be a of full rank, $Rank\,(\boldsymbol{X}_2) = \ell$

GPT Reparations

Reparation ideas linked to $\boldsymbol{X}_2$

- **Loidreau '10** : Proposition of parameters such that $Rank\left(\Lambda_f(\mathscr{C}_{pub})^{\perp}\right) > 1$.
- **Rashwan-Gabidulin-Honary '10** : Similar approach called "Smart approach".

GPT Reparations

Reparation ideas linked to $\boldsymbol{X}_2$

- **Loidreau '10** : Proposition of parameters such that $Rank\left(\Lambda_f(\mathscr{C}_{pub})^{\perp}\right) > 1$.
- **Rashwan-Gabidulin-Honary '10** : Similar approach called "Smart approach".

GPT Reparations

Reparation ideas linked to $\boldsymbol{P}$

These variants consist to select $\boldsymbol{P} \in \mathrm{GL}_{n+\ell}(\mathbb{F}_{q^m})$

- **Gabidulin '08**

$$\boldsymbol{P} = \begin{pmatrix} \boldsymbol{Q}_{11} & \boldsymbol{Q}_{12} \\ \boldsymbol{Q}_{21} & \boldsymbol{Q}_{22} \end{pmatrix}$$

- **Rashwan-Gabidulin-Honary '10**

$$\boldsymbol{P} = \left(\boldsymbol{Q}_1 \mid \boldsymbol{Q}_2 \right)$$

Reparation ideas linked to $\boldsymbol{P}$

These variants consist to select $\boldsymbol{P} \in \mathrm{GL}_{n+\ell}(\mathbb{F}_{q^m})$

- **Gabidulin '08**

$$\boldsymbol{P} = \begin{pmatrix} \boldsymbol{Q}_{11} & \boldsymbol{Q}_{12} \\ \boldsymbol{Q}_{21} & \boldsymbol{Q}_{22} \end{pmatrix}$$

- **Rashwan-Gabidulin-Honary '10**

$$\boldsymbol{P} = (\boldsymbol{Q}_1 \mid \boldsymbol{Q}_2)$$

Plan

No proposition of parameters

Key generation.

Choose $\boldsymbol{P} \in \mathrm{GL}_{n+\ell}(\mathbb{F}_{q^m})$ such that

$$\boldsymbol{P} = \begin{pmatrix} \boldsymbol{Q}_{11} & \boldsymbol{Q}_{12} \\ \boldsymbol{Q}_{21} & \boldsymbol{Q}_{22} \end{pmatrix} \quad (2)$$

- $\boldsymbol{Q}_{11} \in \mathcal{M}_{\ell\times\ell}(\mathbb{F}_{q^m})$
- $\boldsymbol{Q}_{21} \in \mathcal{M}_{n\times\ell}(\mathbb{F}_{q^m})$
- $\boldsymbol{Q}_{12} \in \mathcal{M}_{\ell\times n}(\mathbb{F}_{q^m})$ such that $\mathrm{Rank}_{\mathbb{F}_q}(\boldsymbol{Q}_{12}) = s$
- $\boldsymbol{Q}_{22} \in \mathcal{M}_{n\times n}(\mathbb{F}_q)$

Compute

$$\boldsymbol{G}_{pub} \stackrel{\text{def}}{=} \boldsymbol{S}(\boldsymbol{X} \mid \boldsymbol{G})\boldsymbol{P}^{-1} \quad (3)$$

The public key is $(\boldsymbol{G}_{\text{pub}}, t_{\text{pub}})$ where $t_{\text{pub}} \stackrel{\text{def}}{=} t - s$

No proposition of parameters

Key generation.

Choose $P \in \mathrm{GL}_{n+\ell}(\mathbb{F}_{q^m})$ such that

$$P = \begin{pmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{pmatrix} \quad (2)$$

- $Q_{11} \in \mathcal{M}_{\ell \times \ell}(\mathbb{F}_{q^m})$
- $Q_{21} \in \mathcal{M}_{n \times \ell}(\mathbb{F}_{q^m})$
- $Q_{12} \in \mathcal{M}_{\ell \times n}(\mathbb{F}_{q^m})$ such that $\mathrm{Rank}_{\mathbb{F}_q}(Q_{12}) = s$
- $Q_{22} \in \mathcal{M}_{n \times n}(\mathbb{F}_q)$

Compute

$$G_{pub} \stackrel{\text{def}}{=} S(X \mid G)P^{-1} \quad (3)$$

The public key is $(G_{\mathrm{pub}}, t_{\mathrm{pub}})$ where $t_{\mathrm{pub}} \stackrel{\text{def}}{=} t - s$

No proposition of parameters

Key generation.

Choose $\boldsymbol{P} \in \mathrm{GL}_{n+\ell}(\mathbb{F}_{q^m})$ such that

$$\boldsymbol{P} = \begin{pmatrix} \boldsymbol{Q}_{11} & \boldsymbol{Q}_{12} \\ \boldsymbol{Q}_{21} & \boldsymbol{Q}_{22} \end{pmatrix} \quad (2)$$

- $\boldsymbol{Q}_{11} \in \mathcal{M}_{\ell\times\ell}(\mathbb{F}_{q^m})$
- $\boldsymbol{Q}_{21} \in \mathcal{M}_{n\times\ell}(\mathbb{F}_{q^m})$
- $\boldsymbol{Q}_{12} \in \mathcal{M}_{\ell\times n}(\mathbb{F}_{q^m})$ such that $\mathtt{Rank}_{\mathbb{F}_q}(\boldsymbol{Q}_{12}) = s$
- $\boldsymbol{Q}_{22} \in \mathcal{M}_{n\times n}(\mathbb{F}_q)$

Compute

$$\boldsymbol{G}_{pub} \stackrel{\text{def}}{=} \boldsymbol{S}(\boldsymbol{X} \mid \boldsymbol{G})\boldsymbol{P}^{-1} \quad (3)$$

The public key is $(\boldsymbol{G}_{\text{pub}}, t_{\text{pub}})$ where $t_{\text{pub}} \stackrel{\text{def}}{=} t - s$

No proposition of parameters

Key generation.

Choose $\boldsymbol{P} \in \mathrm{GL}_{n+\ell}(\mathbb{F}_{q^m})$ such that

$$\boldsymbol{P} = \begin{pmatrix} \boldsymbol{Q}_{11} & \boldsymbol{Q}_{12} \\ \boldsymbol{Q}_{21} & \boldsymbol{Q}_{22} \end{pmatrix} \tag{2}$$

- $\boldsymbol{Q}_{11} \in \mathcal{M}_{\ell\times\ell}(\mathbb{F}_{q^m})$
- $\boldsymbol{Q}_{21} \in \mathcal{M}_{n\times\ell}(\mathbb{F}_{q^m})$
- $\boldsymbol{Q}_{12} \in \mathcal{M}_{\ell\times n}(\mathbb{F}_{q^m})$ such that $\mathrm{Rank}_{\mathbb{F}_q}(\boldsymbol{Q}_{12}) = s$
- $\boldsymbol{Q}_{22} \in \mathcal{M}_{n\times n}(\mathbb{F}_q)$

Compute

$$\boldsymbol{G}_{pub} \stackrel{\text{def}}{=} \boldsymbol{S}(\boldsymbol{X} \mid \boldsymbol{G})\boldsymbol{P}^{-1} \tag{3}$$

The public key is $(\boldsymbol{G}_{\mathrm{pub}}, t_{\mathrm{pub}})$ where $t_{\mathrm{pub}} \stackrel{\text{def}}{=} t - s$

No proposition of parameters

Key generation.

Choose $\boldsymbol{P} \in \mathrm{GL}_{n+\ell}(\mathbb{F}_{q^m})$ such that

$$\boldsymbol{P} = \begin{pmatrix} \boldsymbol{Q}_{11} & \boldsymbol{Q}_{12} \\ \boldsymbol{Q}_{21} & \boldsymbol{Q}_{22} \end{pmatrix} \quad (2)$$

- $\boldsymbol{Q}_{11} \in \mathcal{M}_{\ell \times \ell}(\mathbb{F}_{q^m})$
- $\boldsymbol{Q}_{21} \in \mathcal{M}_{n \times \ell}(\mathbb{F}_{q^m})$
- $\boldsymbol{Q}_{12} \in \mathcal{M}_{\ell \times n}(\mathbb{F}_{q^m})$ such that $\mathrm{Rank}_{\mathbb{F}_q}(\boldsymbol{Q}_{12}) = s$
- $\boldsymbol{Q}_{22} \in \mathcal{M}_{n \times n}(\mathbb{F}_q)$

Compute

$$\boldsymbol{G}_{pub} \stackrel{\text{def}}{=} \boldsymbol{S}(\boldsymbol{X} \mid \boldsymbol{G})\boldsymbol{P}^{-1} \quad (3)$$

The public key is $(\boldsymbol{G}_{\mathrm{pub}}, t_{\mathrm{pub}})$ where $t_{\mathrm{pub}} \stackrel{\text{def}}{=} t - s$

Cryptanalysis - Gabidulin's variant

1. **Overbeck's Attack**: Principal threat of Gabidulin-based Schemes
2. Taking $\boldsymbol{P} \in \mathrm{GL}(\mathbb{F}_{q^m})$ might protect against it
3. Gabidulin variant,

$$\boldsymbol{P}^{-1} = \begin{pmatrix} \boldsymbol{Q}_{11} & \boldsymbol{Q}_{12} \\ \boldsymbol{Q}_{21} & \boldsymbol{Q}_{22} \end{pmatrix} \quad \text{avec} \quad \boldsymbol{Q}_{22} \in \mathrm{GL}(\mathbb{F}_q) \quad \text{et} \quad \mathtt{Rank}_{\mathbb{F}_q}(\boldsymbol{Q}_{12}) = s$$

$\rightsquigarrow$ **Global idea of our attack**

	Matrix	Code generated	Length	Correction capability
Secret	$\boldsymbol{G}$	$\mathscr{G}_k(\vec{g})$	n	t
Public	$\boldsymbol{G}_{\mathrm{pub}}$	$(n+\ell, k)$−code	$n+\ell$	$t-s$
Attack	$\boldsymbol{G}^*$	$\mathscr{G}_k(\vec{g}^*)$	$n-s$	$t-\frac{s}{2}$

Cryptanalysis - Gabidulin's variant

1. **Overbeck's Attack**: Principal threat of Gabidulin-based Schemes
2. Taking $\boldsymbol{P} \in \mathrm{GL}(\mathbb{F}_{q^m})$ might protect against it
3. Gabidulin variant,

$$\boldsymbol{P}^{-1} = \begin{pmatrix} \boldsymbol{Q}_{11} & \boldsymbol{Q}_{12} \\ \boldsymbol{Q}_{21} & \boldsymbol{Q}_{22} \end{pmatrix} \text{ avec } \boldsymbol{Q}_{22} \in \mathrm{GL}(\mathbb{F}_q) \text{ et } \mathtt{Rank}_{\mathbb{F}_q}(\boldsymbol{Q}_{12}) = s$$

$\rightsquigarrow$ **Global idea of our attack**

	Matrix	Code generated	Length	Correction capability
Secret	$\boldsymbol{G}$	$\mathscr{G}_k(\vec{g})$	n	t
Public	$\boldsymbol{G}_{\mathrm{pub}}$	$(n+\ell, k)$−code	$n+\ell$	$t-s$
Attack	$\boldsymbol{G}^*$	$\mathscr{G}_k(\vec{g}^*)$	$n-s$	$t-\frac{s}{2}$

1. **Overbeck's Attack**: Principal threat of Gabidulin-based Schemes
2. Taking $\boldsymbol{P} \in \mathrm{GL}(\mathbb{F}_{q^m})$ might protect against it
3. Gabidulin variant,

$$\boldsymbol{P}^{-1} = \begin{pmatrix} \boldsymbol{Q}_{11} & \boldsymbol{Q}_{12} \\ \boldsymbol{Q}_{21} & \boldsymbol{Q}_{22} \end{pmatrix} \quad \text{avec} \quad \boldsymbol{Q}_{22} \in \mathrm{GL}(\mathbb{F}_q) \quad \text{et} \quad \mathtt{Rank}_{\mathbb{F}_q}(\boldsymbol{Q}_{12}) = s$$

$\rightsquigarrow$ **Global idea of our attack**

	Matrix	Code generated	Length	Correction capability
Secret	$\boldsymbol{G}$	$\mathscr{G}_k(\vec{g})$	n	t
Public	$\boldsymbol{G}_{\mathrm{pub}}$	$(n+\ell, k)$−code	$n+\ell$	$t-s$
Attack	$\boldsymbol{G}^*$	$\mathscr{G}_k(\vec{g}^*)$	$n-s$	$t-\frac{s}{2}$

1. **Overbeck's Attack**: Principal threat of Gabidulin-based Schemes
2. Taking $\boldsymbol{P} \in \mathrm{GL}(\mathbb{F}_{q^m})$ might protect against it
3. Gabidulin variant,

$$\boldsymbol{P}^{-1} = \begin{pmatrix} \boldsymbol{Q}_{11} & \boldsymbol{Q}_{12} \\ \boldsymbol{Q}_{21} & \boldsymbol{Q}_{22} \end{pmatrix} \quad \text{avec} \quad \boldsymbol{Q}_{22} \in \mathrm{GL}(\mathbb{F}_q) \quad \text{et} \quad \mathtt{Rank}_{\mathbb{F}_q}(\boldsymbol{Q}_{12}) = s$$

$\rightsquigarrow$ **Global idea of our attack**

	Matrix	Code generated	Length	Correction capability
Secret	$\boldsymbol{G}$	$\mathscr{G}_k(\vec{g})$	n	t
Public	$\boldsymbol{G}_{\mathrm{pub}}$	$(n+\ell, k)$−code	$n+\ell$	$t-s$
Attack	$\boldsymbol{G}^*$	$\mathscr{G}_k(\vec{g}^*)$	$n-s$	$t-\frac{s}{2}$

1. **Overbeck's Attack**: Principal threat of Gabidulin-based Schemes
2. Taking $\boldsymbol{P} \in \mathrm{GL}(\mathbb{F}_{q^m})$ might protect against it
3. Gabidulin variant,

$$\boldsymbol{P}^{-1} = \begin{pmatrix} \boldsymbol{Q}_{11} & \boldsymbol{Q}_{12} \\ \boldsymbol{Q}_{21} & \boldsymbol{Q}_{22} \end{pmatrix} \text{ avec } \boldsymbol{Q}_{22} \in \mathrm{GL}(\mathbb{F}_q) \text{ et } \mathtt{Rank}_{\mathbb{F}_q}(\boldsymbol{Q}_{12}) = s$$

$\rightsquigarrow$ **Global idea of our attack**

	Matrix	Code generated	Length	Correction capability
Secret	$\boldsymbol{G}$	$\mathscr{G}_k(\vec{g})$	n	t
Public	$\boldsymbol{G}_{\mathrm{pub}}$	$(n+\ell, k)-$code	$n+\ell$	$t-s$
Attack	$\boldsymbol{G}^*$	$\mathscr{G}_k(\vec{g}^*)$	$n-s$	$t-\frac{s}{2}$

1. **Overbeck's Attack**: Principal threat of Gabidulin-based Schemes
2. Taking $\boldsymbol{P} \in \mathrm{GL}(\mathbb{F}_{q^m})$ might protect against it
3. Gabidulin variant,

$$\boldsymbol{P}^{-1} = \begin{pmatrix} \boldsymbol{Q}_{11} & \boldsymbol{Q}_{12} \\ \boldsymbol{Q}_{21} & \boldsymbol{Q}_{22} \end{pmatrix} \quad \text{avec} \quad \boldsymbol{Q}_{22} \in \mathrm{GL}(\mathbb{F}_q) \quad \text{et} \quad \mathtt{Rank}_{\mathbb{F}_q}(\boldsymbol{Q}_{12}) = s$$

$\rightsquigarrow$ **Global idea of our attack**

	Matrix	Code generated	Length	Correction capability
Secret	$\boldsymbol{G}$	$\mathscr{G}_k(\vec{g})$	n	t
Public	$\boldsymbol{G}_{\mathrm{pub}}$	$(n+\ell, k)-$code	$n+\ell$	$t-s$
Attack	$\boldsymbol{G}^*$	$\mathscr{G}_k(\vec{g}^*)$	$n-s$	$t-\frac{s}{2}$

1. **Overbeck's Attack**: Principal threat of Gabidulin-based Schemes
2. Taking $\boldsymbol{P} \in \mathrm{GL}(\mathbb{F}_{q^m})$ might protect against it
3. Gabidulin variant,

$$\boldsymbol{P}^{-1} = \begin{pmatrix} \boldsymbol{Q}_{11} & \boldsymbol{Q}_{12} \\ \boldsymbol{Q}_{21} & \boldsymbol{Q}_{22} \end{pmatrix} \text{ avec } \boldsymbol{Q}_{22} \in \mathrm{GL}(\mathbb{F}_q) \text{ et } \mathtt{Rank}_{\mathbb{F}_q}(\boldsymbol{Q}_{12}) = s$$

$\rightsquigarrow$ **Global idea of our attack**

	Matrix	Code generated	Length	Correction capability
Secret	$\boldsymbol{G}$	$\mathscr{G}_k(\vec{g})$	n	t
Public	$\boldsymbol{G}_{\mathrm{pub}}$	$(n+\ell, k)$−code	$n+\ell$	$t-s$
Attack	$\boldsymbol{G}^*$	$\mathscr{G}_k(\vec{g}^*)$	$n-s$	$t-\frac{s}{2}$

Cryptanalysis - Gabidulin's variant

1. **Overbeck's Attack**: Principal threat of Gabidulin-based Schemes
2. Taking $\boldsymbol{P} \in \mathrm{GL}(\mathbb{F}_{q^m})$ might protect against it
3. Gabidulin variant,

$$\boldsymbol{P}^{-1} = \begin{pmatrix} \boldsymbol{Q}_{11} & \boldsymbol{Q}_{12} \\ \boldsymbol{Q}_{21} & \boldsymbol{Q}_{22} \end{pmatrix} \text{ avec } \boldsymbol{Q}_{22} \in \mathrm{GL}(\mathbb{F}_q) \text{ et } \mathtt{Rank}_{\mathbb{F}_q}(\boldsymbol{Q}_{12}) = s$$

$\rightsquigarrow$ **Global idea of our attack**

	Matrix	Code generated	Length	Correction capability
Secret	$\boldsymbol{G}$	$\mathscr{G}_k(\vec{g})$	n	t
Public	$\boldsymbol{G}_{\mathrm{pub}}$	$(n+\ell, k)-$code	$n+\ell$	$t-s$
Attack	$\boldsymbol{G}^*$	$\mathscr{G}_k(\vec{g}^*)$	$n-s$	$t-\frac{s}{2}$

Lemma 5

There exist

- $\boldsymbol{P}_{11} \in \mathrm{GL}_{\ell+s}(\mathbb{F}_{q^m})$
- $\boldsymbol{P}_{22} \in \mathrm{GL}_{n-s}(\mathbb{F}_q)$
- $\boldsymbol{P}_{21} \in \mathcal{M}_{(n-s)\times(\ell+s)}(\mathbb{F}_{q^m})$
- $\boldsymbol{L}$ *and* $\boldsymbol{R}$ *in* $\mathrm{GL}_n(\mathbb{F}_q)$

such that

$$\boldsymbol{P}^{-1} = \begin{pmatrix} \boldsymbol{I}_\ell & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{L} \end{pmatrix} \begin{pmatrix} \boldsymbol{P}_{11} & \boldsymbol{0} \\ \boldsymbol{P}_{21} & \boldsymbol{P}_{22} \end{pmatrix} \begin{pmatrix} \boldsymbol{I}_\ell & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{R} \end{pmatrix} \quad (4)$$

Theorem 6

There exist

- $\boldsymbol{X}^* \in \mathcal{M}_{k\times(\ell+s)}(\mathbb{F}_{q^m})$
- $\boldsymbol{P}^* \in \mathrm{GL}_{n+\ell}(\mathbb{F}_q)$
- $\boldsymbol{G}^*$ *generating a* $(n-s,k)-$*Gabidulin code* $\mathscr{G}_k(\vec{g}^*)$ *such that*

$$\boldsymbol{G}_{\mathrm{pub}} = \boldsymbol{S}(\boldsymbol{X}^* \mid \boldsymbol{G}^*)\boldsymbol{P}^*. \quad (5)$$

$\mathscr{G}_k(\vec{g}^*)$ can correct

$$\frac{n-s-k}{2} = \frac{n-k}{2} - \frac{s}{2} = t - \frac{1}{2}s > t - s = t_{\mathrm{pub}}$$

Theorem 6

There exist

- $\boldsymbol{X}^* \in \mathcal{M}_{k\times(\ell+s)}(\mathbb{F}_{q^m})$
- $\boldsymbol{P}^* \in \mathrm{GL}_{n+\ell}(\mathbb{F}_q)$
- $\boldsymbol{G}^*$ *generating a* $(n-s,k)-$*Gabidulin code* $\mathscr{G}_k(\vec{g}^*)$ *such that*

$$\boldsymbol{G}_{\mathrm{pub}} = \boldsymbol{S}(\boldsymbol{X}^* \mid \boldsymbol{G}^*)\boldsymbol{P}^*. \quad (5)$$

$\mathscr{G}_k(\vec{g}^*)$ can correct

$$\frac{n-s-k}{2} = \frac{n-k}{2} - \frac{s}{2} = t - \frac{1}{2}s > t - s = t_{\mathrm{pub}}$$

Cryptanalysis - Gabidulin's variant

Steps of the attack

- Compute

$$\Lambda_{n-s-k-1}(\mathscr{C}_{pub})^{\perp}$$

- If

$$\dim \Lambda_{n-s-k-1}(\mathscr{C}_{pub})^{\perp} = 1$$

- Choose $\vec{h} \in \Lambda_{n-s-k-1}(\mathscr{C}_{pub})^{\perp}, \quad \vec{h} \neq \mathbf{0}$
- Find $\boldsymbol{T} \in \mathrm{GL}_{n+\ell}(\mathbb{F}_q)$ such that $\vec{h} = (\mathbf{0} \mid \vec{h}')\boldsymbol{T}$, $\vec{h} \in \mathbb{F}_{q^m}^{n-s}$.

Key generation

Choose $\boldsymbol{P} \in \mathrm{GL}_n(\mathbb{F}_{q^m})$ such that

$$\boldsymbol{P} = (\boldsymbol{Q}_1 \mid \boldsymbol{Q}_2) \tag{6}$$

- $\boldsymbol{Q}_1 \in \mathcal{M}_{n\times a}(\mathbb{F}_{q^m})$
- while $\boldsymbol{Q}_2 \in \mathcal{M}_{n\times(n-a)}(\mathbb{F}_q)$
- $a \stackrel{\text{def}}{=} t - t_{\text{pub}} \implies t_{\text{pub}} = t - a$

$$(\boldsymbol{Q}_1 \mid \boldsymbol{Q}_2) = \begin{pmatrix} \boldsymbol{Q}_{11} & \boldsymbol{Q}_{12} \\ \boldsymbol{Q}_{21} & \boldsymbol{Q}_{22} \end{pmatrix} = \begin{pmatrix} \boldsymbol{Q}_{11} & \boldsymbol{Q}_{12} \\ \boldsymbol{Q}_{21} & \boldsymbol{Q}_{22} \end{pmatrix}$$

Gabidulin-Rashwan-Honary variant is a particular case of the Gabidulin variant with $s = a$

Gabidulin, Rashwan and Honary Variant

Key generation

Choose $\boldsymbol{P} \in \mathrm{GL}_n(\mathbb{F}_{q^m})$ such that

$$\boldsymbol{P} = (\boldsymbol{Q}_1 \mid \boldsymbol{Q}_2) \tag{6}$$

- $\boldsymbol{Q}_1 \in \mathcal{M}_{n\times a}(\mathbb{F}_{q^m})$
- while $\boldsymbol{Q}_2 \in \mathcal{M}_{n\times(n-a)}(\mathbb{F}_q)$
- $a \stackrel{\text{def}}{=} t - t_{\mathrm{pub}} \implies t_{\mathrm{pub}} = t - a$

$$(\boldsymbol{Q}_1 \mid \boldsymbol{Q}_2) = \begin{pmatrix} \boldsymbol{Q}_{11} & \boldsymbol{Q}_{12} \\ \boldsymbol{Q}_{21} & \boldsymbol{Q}_{22} \end{pmatrix} = \begin{pmatrix} \boldsymbol{Q}_{11} & \boldsymbol{Q}_{12} \\ \boldsymbol{Q}_{21} & \boldsymbol{Q}_{22} \end{pmatrix}$$

Gabidulin-Rashwan-Honary variant is a particular case of the Gabidulin variant with $s = a$

Key generation

Choose $\boldsymbol{P} \in \mathrm{GL}_n(\mathbb{F}_{q^m})$ such that

$$\boldsymbol{P} = (\boldsymbol{Q}_1 \mid \boldsymbol{Q}_2) \tag{6}$$

- $\boldsymbol{Q}_1 \in \mathcal{M}_{n \times a}(\mathbb{F}_{q^m})$
- while $\boldsymbol{Q}_2 \in \mathcal{M}_{n \times (n-a)}(\mathbb{F}_q)$
- $a \stackrel{\text{def}}{=} t - t_{\text{pub}} \implies t_{\text{pub}} = t - a$

$$(\boldsymbol{Q}_1 \mid \boldsymbol{Q}_2) = \begin{pmatrix} \boldsymbol{Q}_{11} & \boldsymbol{Q}_{12} \\ \boldsymbol{Q}_{21} & \boldsymbol{Q}_{22} \end{pmatrix} = \begin{pmatrix} \boldsymbol{Q}_{11} & \boldsymbol{Q}_{12} \\ \boldsymbol{Q}_{21} & \boldsymbol{Q}_{22} \end{pmatrix}$$

Gabidulin-Rashwan-Honary variant is a particular case of the Gabidulin variant with $s = a$

Gabidulin, Rashwan and Honary variant - Cryptanalysis

Steps of the attack

- Compute

$$\Lambda_{n-a-k-1}(\mathscr{C}_{pub})^{\perp}$$

- If

$$\dim \Lambda_{n-a-k-1}(\mathscr{C}_{pub})^{\perp} = 1$$

- Choose $\vec{h} \in \Lambda_{n-a-k-1}(\mathscr{C}_{pub})^{\perp}, \quad \vec{h} \neq \mathbf{0}$
- Find $\boldsymbol{T} \in \mathrm{GL}_n(\mathbb{F}_q)$ such that $\vec{h} = (\mathbf{0} \mid \vec{h}')\boldsymbol{T}$, $\vec{h} \in \mathbb{F}_{q^m}^{n-a}$.

Experimental Results

m	k	t	t_{pub}	Temps (second)
20	10	5	4	$\leqslant 1$
28	14	7	3	$\leqslant 1$
28	14	7	4	$\leqslant 1$
28	14	7	5	$\leqslant 1$
28	14	7	6	$\leqslant 1$
20	10	5	4	$\leqslant 1$

Table : Parameters where $n = m$ and at least 80-bit security.

Plan

1. The General GPT Cryptosystem
2. Some Reparations of the System
3. Conclusion and Related Work

Conclusion

Code based encryption schemes

- **Main drawback**: Enormous size of the Keys
- **Potential solution**: Rank metric codes
  - **Gabidulin codes**
  - Too structured $\rightsquigarrow$ Public code distinguishable

$\rightsquigarrow$ **Our works show that several attempts to mask them have failed**

Conclusion

Code based encryption schemes

- **Main drawback**: Enormous size of the Keys
- **Potential solution**: Rank metric codes
  - **Gabidulin codes**
  - Too structured $\rightsquigarrow$ Public code distinguishable

$\rightsquigarrow$ **Our works show that several attempts to mask them have failed**

Conclusion

Code based encryption schemes

- **Main drawback**: Enormous size of the Keys
- **Potential solution**: Rank metric codes
  - **Gabidulin codes**
  - Too structured $\rightsquigarrow$ Public code distinguishable

$\rightsquigarrow$ **Our works show that several attempts to mask them have failed**

Conclusion

Code based encryption schemes

- **Main drawback**: Enormous size of the Keys
- **Potential solution**: Rank metric codes
  - **Gabidulin codes**
  - Too structured $\rightsquigarrow$ Public code distinguishable

$\rightsquigarrow$ **Our works show that several attempts to mask them have failed**

Conclusion

Code based encryption schemes

- **Main drawback**: Enormous size of the Keys
- **Potential solution**: Rank metric codes
  - **Gabidulin codes**
  - Too structured $\rightsquigarrow$ Public code distinguishable

$\rightsquigarrow$ **Our works show that several attempts to mask them have failed**

Perspectives - Designing

Code based encryption

- Indistinguishability proof of the public code
  - **Wang** '16
- Schemes without masking phase
  - **Alekhnovich** '03
  - **Aguilar-Blazy-Deneuville-Gaborit-Zémor** '16

Perspectives - Designing

Code based encryption

- Indistinguishability proof of the public code
  - **Wang** '16
- Schemes without masking phase
  - **Alekhnovich** '03
  - **Aguilar-Blazy-Deneuville-Gaborit-Zémor** '16

Perspectives - Designing

Code based encryption

- Indistinguishability proof of the public code
  - **Wang** '16
- Schemes without masking phase
  - **Alekhnovich** '03
  - **Aguilar-Blazy-Deneuville-Gaborit-Zémor** '16

Perspectives - Designing

Code based encryption

- Indistinguishability proof of the public code
  - **Wang** '16
- Schemes without masking phase
  - **Alekhnovich** '03
  - **Aguilar-Blazy-Deneuville-Gaborit-Zémor '16**

LRPC Cryptosystem

- $\mathscr{V} \subset \mathbb{F}_{q^m}$ a $\mathbb{F}_q-$vector space
- $d = \dim_{\mathbb{F}_q}(\mathscr{V})$
- $\boldsymbol{H} \in \mathcal{M}_{n-k\times n}(\mathscr{V})$, $Rank(\boldsymbol{H}) = n-k$
- $\boldsymbol{G}_{pub} \in \mathcal{M}_{k\times n}(\mathbb{F}_{q^m})$ such that $\boldsymbol{H}\boldsymbol{G}_{pub}^t = \boldsymbol{0}$
- The public key is

$$(\boldsymbol{G}_{pub}, t) \text{ with } t \leqslant \frac{n-k}{d}$$

New masking for Gabidulin codes: P. Loidreau '16

- $\mathscr{V} \subset \mathbb{F}_{q^m}$ a $\mathbb{F}_q-$vector space
- $d = \dim_{\mathbb{F}_q}(\mathscr{V}) \geqslant 3$
- Choose

$$\boldsymbol{P} \in \mathsf{GL}_n(\mathscr{V}) \quad \text{and} \quad \boldsymbol{G}_{\mathrm{pub}} = \boldsymbol{S}\boldsymbol{G}\boldsymbol{P}^{-1}$$

$$\rightarrow t_{\mathrm{pub}} = \frac{n-k}{2d}$$