

Towards the security of McEliece's cryptosystem based on Hermitian subfield subcodes

Gábor P. Nagy (joint work with Sabira El Khalifaoui)

20th International Conference on Cryptology "Siberian Scientific School-seminar" named after G. P. Agibalov

Novosibirsk, September 6-11, 2021

Budapest University of Technology and Economics (Hungary)

University of Szeged (Hungary)

Code-based cryptography

The Schur product distinguisher

Algebraic-geometric (AG) codes

Subfield subcodes of Hermitian codes

Code-based cryptography

The Schur product distinguisher

Algebraic-geometric (AG) codes

Subfield subcodes of Hermitian codes

Introduction

- Recently, there has been a big amount of research addressed to **quantum computers**.
- Quantum computers use quantum mechanical techniques to solve **hard computational problems** in mathematics.
- The existence of these powerful machines threaten many of the **public-key cryptosystem** that are widely in use.
- **Robert McEliece (1978)** introduced the first **code-based public-key cryptosystem**, that still **resists the quantum attack**.

BUT most importantly...

Introduction

- Recently, there has been a big amount of research addressed to **quantum computers**.
- Quantum computers use quantum mechanical techniques to solve **hard computational problems** in mathematics.
- The existence of these powerful machines threaten many of the **public-key cryptosystem** that are widely in use.
- **Robert McEliece (1978)** introduced the first **code-based public-key cryptosystem**, that still **resists the quantum attack**.

BUT most importantly...

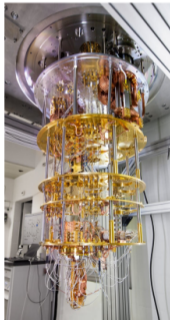


Problem 6. «A promise»

Young cryptographers, Alice, Bob and Carol, are interested in quantum computations and really want to buy a quantum computer. A millionaire gave them a certain amount of money (say, X_A for Alice, X_B for Bob, and X_C for Carol). He also made them promise that they would not tell anyone, including each other, how much money everyone of them had received.

- Could you help the cryptographers to invent an algorithm how to find out (without breaking the promise) whether the total amount of money they have, $X_A + X_B + X_C$, is enough to buy a quantum computer?
- What weaknesses does your algorithm have (if someone breaks the promise)? Does it always protect the secret of the honest participants from the dishonest ones?

Young cryptographers really want to buy a quantum computer!



IBM's 50 qubit quantum computing system

Error correcting codes

$[n, k]_q$ -code \mathcal{C}

- n, k positive integers
- q prime power
- \mathbb{F}_q finite field of order q
- $\mathcal{C} \leq \mathbb{F}_q^n$ linear subspace
- $k = \dim(\mathcal{C})$

The code \mathcal{C} is given by

- the $k \times n$ generator matrix G ,
- the $(n - k) \times n$ parity check matrix H

$$\mathcal{C} = \text{Im}G = \ker H^T$$

Decoding up to t errors

- t positive integer
- $\mathcal{D}_{\mathcal{C}, t} : \mathbb{F}_q^n \rightarrow \mathcal{C} \cup \{*\}$
- $\mathcal{D}_{\mathcal{C}, t}(\mathbf{y}) = \mathbf{x}$, if there is a unique vector \mathbf{e} with $\mathbf{y} = \mathbf{x} + \mathbf{e}$ and $\text{wt}(\mathbf{e}) \leq t$
- $\mathcal{D}_{\mathcal{C}, t}(\mathbf{y}) = *$ otherwise

Theorem (Berlekamp, McEliece, van Tilborg 1978)

Given $\mathcal{C}, \mathbf{y}, t$. It is NP-complete to decide if $\mathcal{D}_{\mathcal{C}, t}(\mathbf{y}) = *$. (Even for $q = 2$.)

Error correcting codes

$[n, k]_q$ -code \mathcal{C}

- n, k positive integers
- q prime power
- \mathbb{F}_q finite field of order q
- $\mathcal{C} \leq \mathbb{F}_q^n$ linear subspace
- $k = \dim(\mathcal{C})$

The code \mathcal{C} is given by

- the $k \times n$ generator matrix G ,
- the $(n - k) \times n$ parity check matrix H

$$\mathcal{C} = \text{Im}G = \ker H^T$$

Decoding up to t errors

- t positive integer
- $\mathcal{D}_{\mathcal{C}, t} : \mathbb{F}_q^n \rightarrow \mathcal{C} \cup \{*\}$
- $\mathcal{D}_{\mathcal{C}, t}(\mathbf{y}) = \mathbf{x}$, if there is a unique vector \mathbf{e} with $\mathbf{y} = \mathbf{x} + \mathbf{e}$ and $\text{wt}(\mathbf{e}) \leq t$
- $\mathcal{D}_{\mathcal{C}, t}(\mathbf{y}) = *$ otherwise

Theorem (Berlekamp, McEliece, van Tilborg 1978)

Given $\mathcal{C}, \mathbf{y}, t$. It is NP-complete to decide if $\mathcal{D}_{\mathcal{C}, t}(\mathbf{y}) = *$. (Even for $q = 2$.)

Error correcting codes

$[n, k]_q$ -code \mathcal{C}

- n, k positive integers
- q prime power
- \mathbb{F}_q finite field of order q
- $\mathcal{C} \leq \mathbb{F}_q^n$ linear subspace
- $k = \dim(\mathcal{C})$

The code \mathcal{C} is given by

- the $k \times n$ generator matrix G ,
- the $(n - k) \times n$ parity check matrix H

$$\mathcal{C} = \text{Im}G = \ker H^T$$

Decoding up to t errors

- t positive integer
- $\mathcal{D}_{\mathcal{C}, t} : \mathbb{F}_q^n \rightarrow \mathcal{C} \cup \{*\}$
- $\mathcal{D}_{\mathcal{C}, t}(\mathbf{y}) = \mathbf{x}$, if there is a unique vector \mathbf{e} with $\mathbf{y} = \mathbf{x} + \mathbf{e}$ and $\text{wt}(\mathbf{e}) \leq t$
- $\mathcal{D}_{\mathcal{C}, t}(\mathbf{y}) = *$ otherwise

Theorem (Berlekamp, McEliece, van Tilborg 1978)

Given $\mathcal{C}, \mathbf{y}, t$. It is NP-complete to decide if $\mathcal{D}_{\mathcal{C}, t}(\mathbf{y}) = *$. (Even for $q = 2$.)

Code-based cryptography

- Let \mathcal{C} be an $[n, k]_q$ -code
- given by a generator matrix $G = [I_k G_0]$ in *systematic form*
- Let $\mathcal{D}_{\mathcal{C}, t}$ be a decoding up to t errors
- given by **an efficient algorithm**

Example: classic McEliece

- $\mathcal{C}_1 = \Gamma(L, g)$ the binary Goppa code with generator matrix G_1 .
- S invertible $k \times k$ matrix, P $n \times n$ permutation matrix
- $G = SG_1P$, $\mathcal{C} = \text{Im}(G)$
- $\mathcal{D}_{\mathcal{C}, t}$ is deduced from S, P and $\mathcal{D}_{\mathcal{C}_1, t}$.

- public key: G, t
- private key: $\mathcal{D}_{\mathcal{C}, t}$

Encryption:

- plain text message $\mathbf{x} \in \mathbb{F}_q^k$
- $\mathbf{e} \in \mathbb{F}_q^n$ random with $\text{wt}(\mathbf{e}) = t$
- cipher text $\mathbf{y} = \mathbf{x}G + \mathbf{e}$

Decryption:

- $\mathbf{z} = \mathcal{D}_{\mathcal{C}, t}(\mathbf{y}) \in \text{Im}G$
- solve the system of linear equations $\mathbf{x}G = \mathbf{z}$

Remark. The private key of classic McEliece is (L, g, S, P)

Code-based cryptography

- Let \mathcal{C} be an $[n, k]_q$ -code
- given by a generator matrix
 $G = [I_k G_0]$ in *systematic form*
- Let $\mathcal{D}_{\mathcal{C},t}$ be a decoding up to t errors
- given by **an efficient algorithm**

Example: classic McEliece

- $\mathcal{C}_1 = \Gamma(L, g)$ the binary Goppa code with generator matrix G_1 .
- S invertible $k \times k$ matrix, P $n \times n$ permutation matrix
- $G = SG_1P$, $\mathcal{C} = \text{Im}(G)$
- $\mathcal{D}_{\mathcal{C},t}$ is deduced from S, P and $\mathcal{D}_{\mathcal{C}_1,t}$.

- public key: G, t
- private key: $\mathcal{D}_{\mathcal{C},t}$

Encryption:

- plain text message $\mathbf{x} \in \mathbb{F}_q^k$
- $\mathbf{e} \in \mathbb{F}_q^n$ random with $\text{wt}(\mathbf{e}) = t$
- cipher text $\mathbf{y} = \mathbf{x}G + \mathbf{e}$

Decryption:

- $\mathbf{z} = \mathcal{D}_{\mathcal{C},t}(\mathbf{y}) \in \text{Im}G$
- solve the system of linear equations $\mathbf{x}G = \mathbf{z}$

Remark. The private key of classic McEliece is (L, g, S, P)

Code-based cryptography

- Let \mathcal{C} be an $[n, k]_q$ -code
- given by a generator matrix $G = [I_k G_0]$ in *systematic form*
- Let $\mathcal{D}_{\mathcal{C},t}$ be a decoding up to t errors
- given by **an efficient algorithm**

Example: classic McEliece

- $\mathcal{C}_1 = \Gamma(L, g)$ the binary Goppa code with generator matrix G_1 .
- S invertible $k \times k$ matrix, P $n \times n$ permutation matrix
- $G = SG_1P$, $\mathcal{C} = \text{Im}(G)$
- $\mathcal{D}_{\mathcal{C},t}$ is deduced from S, P and $\mathcal{D}_{\mathcal{C}_1,t}$.

- public key: G, t
- private key: $\mathcal{D}_{\mathcal{C},t}$

Encryption:

- plain text message $\mathbf{x} \in \mathbb{F}_q^k$
- $\mathbf{e} \in \mathbb{F}_q^n$ random with $\text{wt}(\mathbf{e}) = t$
- cipher text $\mathbf{y} = \mathbf{x}G + \mathbf{e}$

Decryption:

- $\mathbf{z} = \mathcal{D}_{\mathcal{C},t}(\mathbf{y}) \in \text{Im}G$
- solve the system of linear equations $\mathbf{x}G = \mathbf{z}$

Remark. The private key of classic McEliece is (L, g, S, P)

Code-based cryptography

The Schur product distinguisher

Algebraic-geometric (AG) codes

Subfield subcodes of Hermitian codes

Security of code-based cryptosystems

Message recovery attack:

- information set decoding (ISD)
- implementation: Prange (1962)
- time complexity:

$$C_{\text{Prange}} = \frac{\binom{n}{t}}{\binom{n-k}{t}} \cdot \text{poly}(n, k)$$

- Many improvements but still exponential in t

Key recovery attack:

- Hard problem

Distinguishing attack:

- The public key G cannot be distinguished from a random matrix
- In many cases, it leads to key recovery method

Definition: Security level

We say that a code-based cryptosystem has security level L (in bits), if

$$\binom{n}{t} / \binom{n-k}{t} > 2^L.$$

Security of code-based cryptosystems

Message recovery attack:

- information set decoding (ISD)
- implementation: Prange (1962)
- time complexity:

$$C_{\text{Prange}} = \frac{\binom{n}{t}}{\binom{n-k}{t}} \cdot \text{poly}(n, k)$$

- Many improvements but still exponential in t

Key recovery attack:

- Hard problem

Distinguishing attack:

- The public key G cannot be distinguished from a **random matrix**
- In many cases, it leads to key recovery method

Definition: Security level

We say that a code-based cryptosystem has security level L (in bits), if

$$\binom{n}{t} / \binom{n-k}{t} > 2^L.$$

Schur product distinguisher

- Given two elements $\mathbf{a} = (a_1, \dots, a_n)$ and $\mathbf{b} = (b_1, \dots, b_n)$ in \mathbb{F}_q^n , the **Schur product** is the component-wise product

$$\mathbf{a} * \mathbf{b} := (a_1 b_1, \dots, a_n b_n)$$

on \mathbb{F}_q^n .

- For two linear subspaces $A, B \subseteq \mathbb{F}_q^n$, their **Schur product** is the linear subspace

$$A * B := \text{Span}_{\mathbb{F}_q} \{ \mathbf{a} * \mathbf{b} \mid \mathbf{a} \in A \text{ and } \mathbf{b} \in B \}.$$

- $A * A$ is denoted as A^{*2} , and we define A^{*k} by *induction*.

Cascudo, Cramer, Mirandola, Zémor
(2015)

With high probability,

$$\dim(\mathcal{C} * \mathcal{C}) = \frac{k(k+1)}{2}.$$

Definition: s-good codes

\mathcal{C} is **s-good**, if

$$\dim(\mathcal{C} * \mathcal{C}) = \dim(\mathcal{C}^\perp * \mathcal{C}^\perp) = n$$

Code-based cryptography

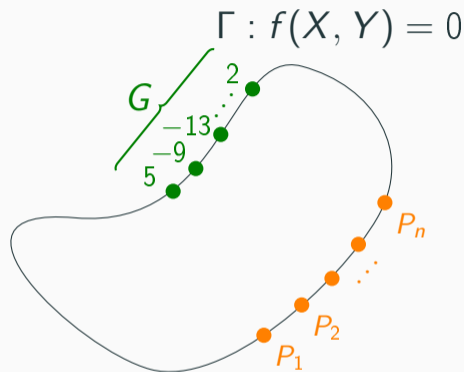
The Schur product distinguisher

Algebraic-geometric (AG) codes

Subfield subcodes of Hermitian codes

Algebraic-geometric (AG) codes

- Let $\Gamma : f(X, Y) = 0$ be a smooth curve over the finite field \mathbb{F}_q .
- Let P_1, \dots, P_n be distinct places of degree one of Γ .
- Define the divisor $D = P_1 + \dots + P_n$.
- Let G be a divisor of Γ whose support is disjoint to $\{P_1, \dots, P_n\}$.
- Let $\mathcal{L}(G)$ be the Riemann-Roch space of G .



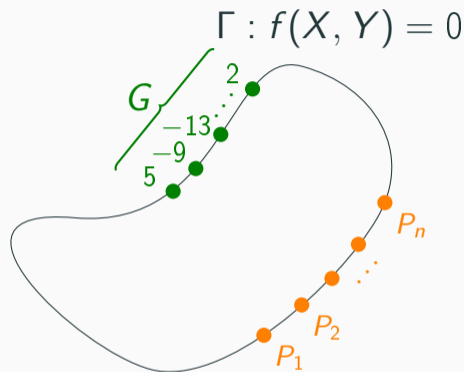
Definition: AG-code

The functional code associated to the divisors D and G is

$$C_L(D, G) = \{(\varphi(P_1), \dots, \varphi(P_n)) \mid \varphi \in \mathcal{L}(G)\} \leq \mathbb{F}_q^n.$$

Algebraic-geometric (AG) codes

- Let $\Gamma : f(X, Y) = 0$ be a smooth curve over the finite field \mathbb{F}_q .
- Let P_1, \dots, P_n be distinct places of degree one of Γ .
- Define the divisor $D = P_1 + \dots + P_n$.
- Let G be a divisor of Γ whose support is disjoint to $\{P_1, \dots, P_n\}$.
- Let $\mathcal{L}(G)$ be the Riemann-Roch space of G .



Definition: AG-code

The functional code associated to the divisors D and G is

$$C_L(D, G) = \{(\varphi(P_1), \dots, \varphi(P_n)) \mid \varphi \in \mathcal{L}(G)\} \leq \mathbb{F}_q^n.$$

Properties of AG-codes

- The **length** of $C_L(D, G)$ is n .
- For $\deg(G) \leq n$, we have

$$\dim C_L(D, G) \leq \deg(G) - g + 1,$$

and equality hold for $\deg(G) \geq 2g - 2$.

- There are **efficient decoding algorithms** for $C_L(D, G)$ up to t errors, where

$$2t < n - \deg(G).$$

"Definition" of differential codes $C_\Omega(D, G)$

The **dual space** of $C_L(D, G)$ is a code equivalent to $C_L(D, G')$, where $\deg(G') = n + 2g - 2 - \deg(G)$.

Subfield subcodes of AG-codes

- Binary Goppa codes, BCH-codes are subfield subcodes of the generalized Reed-Solomon codes.
- Let $r = q^m$, that is, \mathbb{F}_r is an extension of \mathbb{F}_q of degree m .
- Let \mathcal{C} be an $[n, k, t]_r$ linear code.
- The subfield subcode of \mathcal{C} is defined by

$$\mathcal{C}|_{\mathbb{F}_q} = \mathcal{C} \cap \mathbb{F}_q^n.$$

- For $k^* = \dim(\mathcal{C}|_{\mathbb{F}_q})$, we have

$$k^* \geq n - m(n - k).$$

- The true values of k^* are not known in general.

Schur filtration of AG-codes (Wieschebrink, Couvreur et al. 2010-2019)

- In general, **AG-codes are s-bad**:

$$\dim(\mathcal{C}^{*2}) \leq 2k + g - 1 \ll k(k+1)/2,$$

where $\mathcal{C} = \mathcal{C}_L(D, G)$ is an AG-code of dimension k .

- The **Schur product distinguisher** is very effective for AG-codes.
- Using the system of subspaces

$$W_{i,j} = \{z \in \mathbb{F}_q^n \mid z * \mathcal{C}^{*i} \leq \mathcal{C}^{*j}\}$$

one can construct an **error correcting array** and an efficient decoding.

Remark (Couvreur, Marquez-Corbella, Pellikaan 2018)

Subfield subcodes still resist!

- The **Schur filtration attack** can be extended to subfield subcodes of AG-codes provided the **genus g** and the **degree m** are small.

Schur filtration of AG-codes (Wieschebrink, Couvreur et al. 2010-2019)

- In general, **AG-codes are s-bad**:

$$\dim(\mathcal{C}^{*2}) \leq 2k + g - 1 \ll k(k+1)/2,$$

where $\mathcal{C} = C_L(D, G)$ is an AG-code of dimension k .

- The **Schur product distinguisher** is very effective for AG-codes.
- Using the system of subspaces

$$W_{i,j} = \{z \in \mathbb{F}_q^n \mid z * \mathcal{C}^{*i} \leq \mathcal{C}^{*j}\}$$

one can construct an **error correcting array** and an efficient decoding.

Remark (Couvreur, Marquez-Corbella, Pellikaan 2018)

Subfield subcodes still resist!

- The **Schur filtration attack** can be extended to subfield subcodes of AG-codes provided the **genus g** and the **degree m** are small.

Code-based cryptography

The Schur product distinguisher

Algebraic-geometric (AG) codes

Subfield subcodes of Hermitian codes

Definition: Hermitian curve

The equation

$$X^{q+1} = Y^q + Y$$

defines the **Hermitian curve** \mathcal{H}_q over the finite field \mathbb{F}_{q^2} .

- Smooth of **genus** $g = q(q - 1)/2$.
- q^3 **affine points** over \mathbb{F}_{q^2} .
- Point at infinity: $P_\infty = (0 : 1 : 0)$.
- **Maximal curve:** The number of \mathbb{F}_{q^2} -rational points attains the Hasse-Weil bound.

Theorem (Xing, Stichtenoth 2009)

For fixed q , the **genus of a \mathbb{F}_{q^2} -maximal curve Γ** is $\leq q(q - 1)/2$, and equality holds if and only if Γ is isomorphic to \mathcal{H}_q .

Definition: Hermitian curve

The equation

$$X^{q+1} = Y^q + Y$$

defines the **Hermitian curve** \mathcal{H}_q over the finite field \mathbb{F}_{q^2} .

- Smooth of **genus** $g = q(q - 1)/2$.
- q^3 **affine points** over \mathbb{F}_{q^2} .
- Point at infinity: $P_\infty = (0 : 1 : 0)$.
- **Maximal curve**: The number of \mathbb{F}_{q^2} -rational points attains the Hasse-Weil bound.

Theorem (Xing, Stichtenoth 2009)

For fixed q , the **genus of a \mathbb{F}_{q^2} -maximal curve** Γ is $\leq q(q - 1)/2$, and equality holds if and only if Γ is **isomorphic to \mathcal{H}_q** .

S-goodness of 1-point Hermitian subfield subcodes

Definition: 1-point Hermitian subfield subcodes

Let $n = q^3$, $r = q^2$, $D = P_1 + \cdots + P_n$. For $s = 1, \dots, n + 2g - 2$, define

$$C_q(s) = C_L(D, sP_\infty)|_{\mathbb{F}_q}.$$

Proposition

There are positive integers a_q, b_q such that $C_q(s)$ is **s-good** if and only if $a_q \leq s \leq b_q$.

Numerical results using our [GAP package HERmitian](#):

q	4	5	7	8	9	11	13	16
a_q	45	72	192	315	400	720	1 176	2 295
b_q	59	119	335	503	719	1 319	2 183	4 079

Conjecture: $b_q = q^3 - q - 1$.

Comparison of key sizes $k(n - k) \log_2 q$

Classic McEliece		n	k	t	Prange complexity	Key size (bit)
Category 1 (AES-128)		3 488	2 720	64	142.78	2 088 960
Category 3 (AES-192)		4 608	3 360	96	184.89	4 193 280
Category 5 (AES-256)		6 688	5 024	128	262.35	8 359 936
		6 960	5 413	119	263.44	8 373 911
		8 192	6 528	128	300.14	10 862 592
	Code Type	n	k	t	Prange complexity	Key size (bit)
Category 1	$C_{11}(1\,174)$	1 331	927	78	142.33	1 123 524
Category 3	$C_{13}(2\,039)$	2 197	1 735	79	185.89	3 206 280
	$C_{16}(3\,980)$	4 096	3 634	58	187.40	6 715 632
Category 5	$C_{13}(1\,861)$	2 197	1 398	168	263.01	4 468 008
	$C_{16}(3\,874)$	4 096	3 422	111	300.65	9 225 712

Спасибо за Ваше внимание!

Acknowledgement

The presented work was carried out within the project "**Security Enhancing Technologies for the Internet of Things**" 2018-1.2.1-NKP-2018-00004, supported by the National Research, Development and Innovation Fund of Hungary.