Radboud University

The Matrix Code Equivalence problem

Hardness estimates in the rank metric

Krijn Reijnders (joint work with **Simona Samardjiska** and **Monika Trimoska**)
Radboud University

Workshop in Coding and Cryptography, March '22

Today's topic

- **Code Equivalence problem:**
  Given two codes $\mathcal{C}$ and $\mathcal{D}$ over a finite field $\mathbb{F}_q$, find – if any – an isometry μ mapping one to the other, $\mu : \mathcal{C} \to \mathcal{D}$

Today's topic

- **Code Equivalence problem:**
  Given two codes $\mathcal{C}$ and $\mathcal{D}$ over a finite field $\mathbb{F}_q$, find – if any –
  an isometry μ mapping one to the other, $\mu : \mathcal{C} \to \mathcal{D}$

- **Matrix Code Equivalence (MCE):**
  $\mathcal{C}$ and $\mathcal{D}$ are $m \times n$ matrix codes over $\mathbb{F}_q$ equiped with rank metric

Today's topic

- **Code Equivalence problem:**
  Given two codes $\mathcal{C}$ and $\mathcal{D}$ over a finite field $\mathbb{F}_q$, find – if any –
  an isometry μ mapping one to the other, $\mu : \mathcal{C} \to \mathcal{D}$

- **Matrix Code Equivalence (MCE):**
  $\mathcal{C}$ and $\mathcal{D}$ are $m \times n$ matrix codes over $\mathbb{F}_q$ equiped with rank metric

- In this talk:
  - **How hard is MCE?**
  - **How to solve it?**

Matrix Code Equivalence (MCE)

The Matrix Code Equivalence Problem

Matrix code $\mathcal{C}$: a subspace of $\mathcal{M}_{m\times n}(\mathbb{F}_q)$ of dimension k endowed with **rank metric**

$$d(\mathrm{A},\mathrm{B}) = \mathrm{Rank}(\mathrm{A}-\mathrm{B})$$

The Matrix Code Equivalence Problem

Matrix code $\mathcal{C}$: a subspace of $\mathcal{M}_{m\times n}(\mathbb{F}_q)$ of dimension k endowed with **rank metric**

$$d(\mathrm{A},\mathrm{B}) = \mathrm{Rank}(\mathrm{A}-\mathrm{B})$$

Isometry μ: a homomorphism of matrix codes $\mathcal{C} \to \mathcal{D}$ such that for all $\mathrm{C} \in \mathcal{C}$,

$$\mathrm{Rank}\,\mathrm{C} = \mathrm{Rank}\,\mu(\mathrm{C})$$

The Matrix Code Equivalence Problem

Matrix code $\mathcal{C}$: a subspace of $\mathcal{M}_{m\times n}(\mathbb{F}_q)$ of dimension k endowed with **rank metric**

$$d(\mathrm{A},\mathrm{B}) = \mathrm{Rank}(\mathrm{A}-\mathrm{B})$$

Isometry μ: a homomorphism of matrix codes $\mathcal{C}\to\mathcal{D}$ such that for all $\mathrm{C}\in\mathcal{C}$,

$$\mathrm{Rank}\,\mathrm{C} = \mathrm{Rank}\,\mu(\mathrm{C})$$

Matrix Code Equivalence (MCE) problem [Berger, 2003]

MCE$(k,n,m,\mathcal{C},\mathcal{D})$:
Input: Two k-dimensional matrix codes $\mathcal{C},\mathcal{D}\subset\mathcal{M}_{m\times n}(\mathbb{F}_q)$
Question: Find – if any – an isometry $\mu:\mathcal{C}\to\mathcal{D}$.

The Matrix Code Equivalence Problem

Matrix code $\mathcal{C}$: a subspace of $\mathcal{M}_{m\times n}(\mathbb{F}_q)$ of dimension k endowed with **rank metric**

$$d(\mathrm{A},\mathrm{B}) = \mathrm{Rank}(\mathrm{A}-\mathrm{B})$$

Isometry μ: a homomorphism of matrix codes $\mathcal{C}\to\mathcal{D}$ such that for all $\mathrm{C}\in\mathcal{C}$,

$$\mathrm{Rank}\,\mathrm{C} = \mathrm{Rank}\,\mu(\mathrm{C})$$

Matrix Code Equivalence (MCE) problem [Berger, 2003]

MCE$(k,n,m,\mathcal{C},\mathcal{D})$:
Input: Two k-dimensional matrix codes $\mathcal{C},\mathcal{D}\subset\mathcal{M}_{m\times n}(\mathbb{F}_q)$
Question: Find – if any – an isometry $\mu:\mathcal{C}\to\mathcal{D}$.

Known: Any isometry $\mu:\mathcal{C}\to\mathcal{D}$ can be written, for some $\mathrm{A}\in\mathrm{GL}_m(q),\mathrm{B}\in\mathrm{GL}_n(q)$, as

$$\mathrm{C}\mapsto \mathrm{ACB}\in\mathcal{D}$$

Known results [Couvreur, Debris–Alazard & Gaborit, 2021]

$$\mu : \mathrm{C} \mapsto \mathrm{ACB} \in \mathcal{D}, \quad \text{with } \mathrm{A} \in \mathrm{GL}_m(q) \text{ and } \mathrm{B} \in \mathrm{GL}_n(q)$$

- when $A = \mathrm{Id}_m$, or $B = \mathrm{Id}_n$, finding μ is easy (MCRE)

Known results [Couvreur, Debris–Alazard & Gaborit, 2021]

$$\mu : \mathsf{C} \mapsto \mathsf{ACB} \in \mathcal{D}, \quad \text{with } \mathsf{A} \in \mathrm{GL}_m(q) \text{ and } \mathsf{B} \in \mathrm{GL}_n(q)$$

- when $A = \mathrm{Id}_m$, or $B = \mathrm{Id}_n$, finding μ is easy (MCRE)
- code equivalence for $\mathbb{F}_{q^m}$-linear codes with rank metric reduces to MCRE

Known results [Couvreur, Debris–Alazard & Gaborit, 2021]

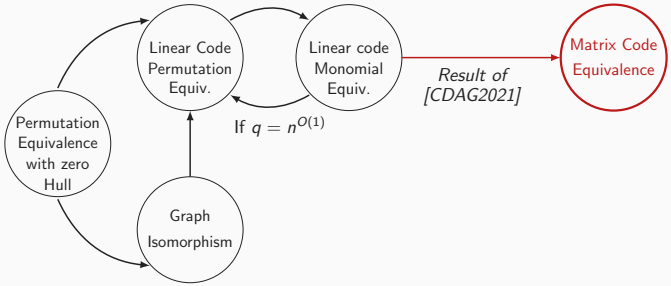
$$\mu : \mathrm{C} \mapsto \mathrm{ACB} \in \mathcal{D}, \quad \text{with } \mathrm{A} \in \mathrm{GL}_m(q) \text{ and } \mathrm{B} \in \mathrm{GL}_n(q)$$

- when $A = \mathrm{Id}_m$, or $B = \mathrm{Id}_n$, finding μ is easy (MCRE)
- code equivalence for $\mathbb{F}_{q^m}$-linear codes with rank metric reduces to MCRE
- MCE is at least as hard as Monomial Equivalence Problem in the Hamming metric

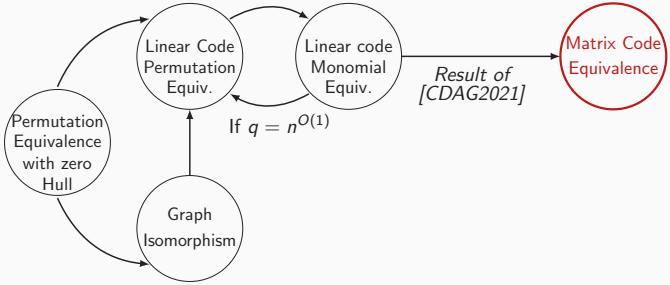
Known results [Couvreur, Debris–Alazard & Gaborit, 2021]

$$\mu : \mathrm{C} \mapsto \mathrm{ACB} \in \mathcal{D}, \quad \text{with } \mathrm{A} \in \mathrm{GL}_m(q) \text{ and } \mathrm{B} \in \mathrm{GL}_n(q)$$

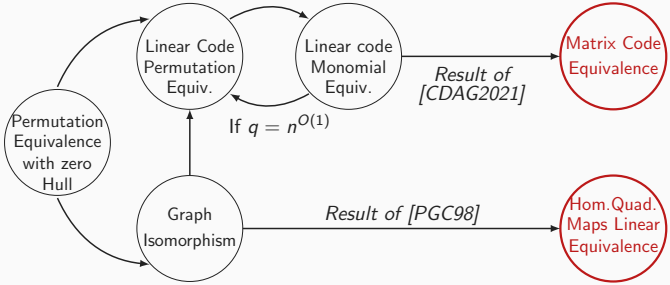
- when $A = \mathrm{Id}_m$, or $B = \mathrm{Id}_n$, finding μ is easy (MCRE)
- code equivalence for $\mathbb{F}_{q^m}$-linear codes with rank metric reduces to MCRE
- MCE is at least as hard as Monomial Equivalence Problem in the Hamming metric



New results

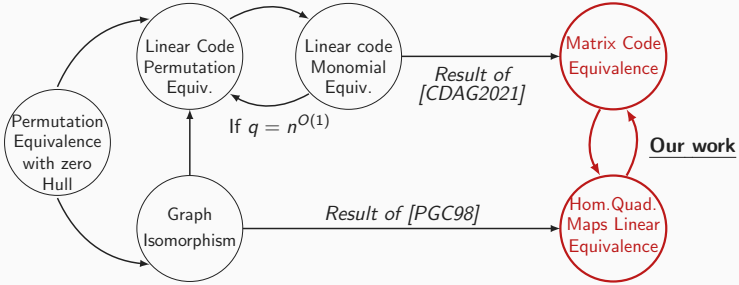

Linear Code Permutation Equiv.
Linear code Monomial Equiv.
Matrix Code Equivalence
Result of [CDAG2021]
If $q = n^{O(1)}$
Permutation Equivalence with zero Hull
Graph Isomorphism

New results



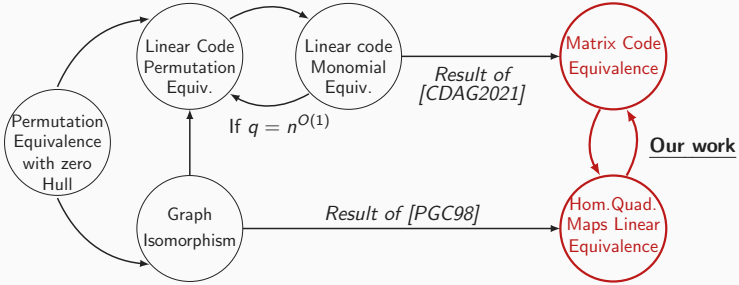
- Homogenous Quadratic Maps Linear Equivalence (hQMLE) problem is well known equivalence problem from multivariate crypto (instance of Isomorphism of Polynomials)

New results



- Homogenous Quadratic Maps Linear Equivalence (hQMLE) problem is well known equivalence problem from multivariate crypto (instance of Isomorphism of Polynomials)
- We show that MCE **is equivalent to** hQMLE

New results



- Homogenous Quadratic Maps Linear Equivalence (hQMLE) problem is well known equivalence problem from multivariate crypto (instance of Isomorphism of Polynomials)
- We show that MCE **is equivalent to** hQMLE
- We upper bound nontrivially the complexity of solving MCE
  - solvable in $\mathcal{O}^*(q^{2/3(m+n)})$ time, for some cases can be improved to $\mathcal{O}^*(q^{\min(m,n)})$
  - previous upper bound $\mathcal{O}^*(q^{\min(m,n)^2})$ time: brute force one side, then solve MCRE

What is hQMLE?

Multivariate crypto basics

- systems of multivariate polynomials $\mathcal{P} = (p_1, p_2, \ldots, p_k)$, every p_s polynomial in N variables $x_1, \ldots, x_N$

Multivariate crypto basics

- systems of multivariate polynomials $\mathcal{P} = (p_1, p_2, \ldots, p_k)$, every p_s polynomial in N variables $x_1, \ldots, x_N$
- most interesting when each p_s is at most degree 2

$$p_s(x_1, \ldots, x_N) = \sum \gamma_{ij}^{(s)} x_i x_j + \sum \beta_i^{(s)} x_i + \alpha^{(s)}, \qquad \alpha^{(s)}, \beta_i^{(s)}, \gamma_{ij}^{(s)} \in \mathbb{F}_q$$

Multivariate crypto basics

- systems of multivariate polynomials $\mathcal{P} = (p_1, p_2, \ldots, p_k)$, every p_s polynomial in N variables $x_1, \ldots, x_N$
- most interesting when each p_s is at most degree 2 and homogeneous part

$$p_s(x_1, \ldots, x_N) = \sum \gamma_{ij}^{(s)} x_i x_j \qquad \gamma_{ij}^{(s)} \in \mathbb{F}_q$$

Multivariate crypto basics

- systems of multivariate polynomials $\mathcal{P} = (p_1, p_2, \ldots, p_k)$, every p_s polynomial in N variables $x_1, \ldots, x_N$
- most interesting when each p_s is at most degree 2 and homogeneous part

$$p_s(x_1, \ldots, x_N) = \sum \gamma_{ij}^{(s)} x_i x_j \qquad \gamma_{ij}^{(s)} \in \mathbb{F}_q$$

homogeneous Quadratic Maps Linear Equivalence (hQMLE) problem

hQMLE$(N, k, \mathcal{F}, \mathcal{P})$:

Input: Two k-tuples of multivariate polynomials

$\mathcal{F} = (f_1, f_2, \ldots, f_k)$, $\mathcal{P} = (p_1, p_2, \ldots, p_k) \in \mathbb{F}_q[x_1, \ldots, x_N]^k$

Question: Find – if any – $\mathsf{S} \in \mathrm{GL}_N(q), \mathsf{T} \in \mathrm{GL}_k(q)$ such that

$$\mathcal{P}(\mathrm{x}) = \mathcal{F}(\mathrm{xS}) \cdot \mathsf{T}$$

Quadratic maps as matrices

$$p_s = \sum \gamma_{ij}^{(s)} x_i x_j \quad = \quad (x_1, \ldots, x_N) \underbrace{\begin{pmatrix} \gamma_{11} & \cdots & \frac{\gamma_{1N}}{2} \\ & & \\ \frac{\gamma_{N1}}{2} & \cdots & \gamma_{NN} \end{pmatrix}}_{P^{(s)} \in \mathcal{M}_{N\times N}(\mathbb{F}_q)} \begin{pmatrix} x_1 \\ \vdots \\ x_N \end{pmatrix}$$

Quadratic maps as matrices

$$p_s = \sum \gamma_{ij}^{(s)} x_i x_j \quad = \quad (x_1, \ldots, x_N) \underbrace{\begin{pmatrix} \gamma_{11} & \cdots & \frac{\gamma_{1N}}{2} \\ & & \\ \frac{\gamma_{N1}}{2} & \cdots & \gamma_{NN} \end{pmatrix}}_{P^{(s)} \in \mathcal{M}_{N\times N}(\mathbb{F}_q)} \begin{pmatrix} x_1 \\ \vdots \\ x_N \end{pmatrix}$$

so with $\mathrm{x} = (x_1, \ldots, x_N)$, we get $p_s(\mathrm{x}) = \mathrm{x} P^{(s)} \mathrm{x}^T$

Quadratic maps as matrices

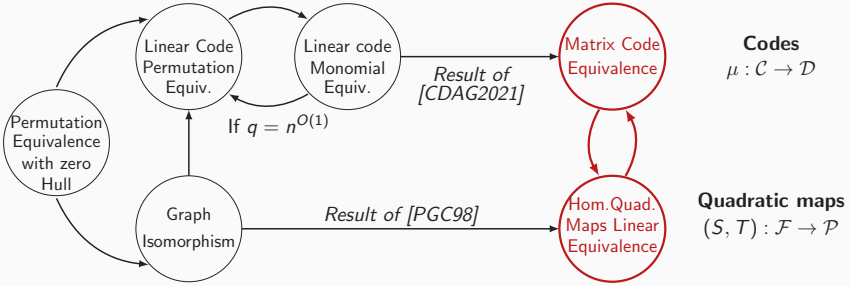
$$p_s = \sum \gamma_{ij}^{(s)} x_i x_j \quad = \quad (x_1, \ldots, x_N) \underbrace{\begin{pmatrix} \gamma_{11} & \ldots & \frac{\gamma_{1N}}{2} \\ & & \\ \frac{\gamma_{N1}}{2} & \ldots & \gamma_{NN} \end{pmatrix}}_{P^{(s)} \in \mathcal{M}_{N\times N}(\mathbb{F}_q)} \begin{pmatrix} x_1 \\ \vdots \\ x_N \end{pmatrix}$$

so with $\mathrm{x} = (x_1, \ldots, x_N)$, we get $p_s(\mathrm{x}) = \mathrm{x} P^{(s)} \mathrm{x}^T$

so $\mathcal{P} = (p_1, \ldots, p_k)$ can be seen as matrices $P^{(1)}, \ldots, P^{(k)}$ spanning some code $\mathcal{D}$

How hard is MCE?

Remember


Linear Code Permutation Equiv.
Linear code Monomial Equiv.
Matrix Code Equivalence
Permutation Equivalence with zero Hull
Graph Isomorphism
Hom.Quad. Maps Linear Equivalence
If $q = n^{O(1)}$
Result of [CDAG2021]
Result of [PGC98]
Codes
$\mu : \mathcal{C} \to \mathcal{D}$
Quadratic maps
$(S, T) : \mathcal{F} \to \mathcal{P}$

How hard is MCE?

MCE **is at least as hard as** hQMLE

Proof *(Sketch)*.

- Given $\mathcal{F}$ and $\mathcal{P}$ such that $\mathcal{P}(\mathrm{x}) = \mathcal{F}(\mathrm{xS}) \cdot \mathrm{T}$

How hard is MCE?

MCE **is at least as hard as** hQMLE

Proof *(Sketch)*.

- Given $\mathcal{F}$ and $\mathcal{P}$ such that $\mathcal{P}(\mathrm{x}) = \mathcal{F}(\mathrm{xS}) \cdot \mathrm{T}$
- $\mathcal{F} = (f_1, \ldots, f_k)$ gives $\langle F^{(1)}, \ldots, F^{(k)} \rangle = \mathcal{C} \subset \mathcal{M}_{N \times N}(\mathbb{F}_q)$, similarly $\mathcal{P}$ gives $\mathcal{D}$

How hard is MCE?

MCE **is at least as hard as** hQMLE

Proof *(Sketch)*.

- Given $\mathcal{F}$ and $\mathcal{P}$ such that $\mathcal{P}(\mathrm{x}) = \mathcal{F}(\mathrm{xS}) \cdot \mathrm{T}$
- $\mathcal{F} = (f_1, \ldots, f_k)$ gives $\langle F^{(1)}, \ldots, F^{(k)} \rangle = \mathcal{C} \subset \mathcal{M}_{N \times N}(\mathbb{F}_q)$, similarly $\mathcal{P}$ gives $\mathcal{D}$
- as before $f_s(\mathrm{xS}) = (\mathrm{xS})F^{(s)}(\mathrm{xS})^T = \mathrm{x}(\mathrm{S}F^{(s)}\mathrm{S}^T)\mathrm{x}^T$ for every quadratic map in $\mathcal{F}$

How hard is MCE?

MCE **is at least as hard as** hQMLE

Proof *(Sketch)*.

- Given $\mathcal{F}$ and $\mathcal{P}$ such that $\mathcal{P}(\mathrm{x}) = \mathcal{F}(\mathrm{xS}) \cdot \mathrm{T}$
- $\mathcal{F} = (f_1, \ldots, f_k)$ gives $\langle F^{(1)}, \ldots, F^{(k)} \rangle = \mathcal{C} \subset \mathcal{M}_{N \times N}(\mathbb{F}_q)$, similarly $\mathcal{P}$ gives $\mathcal{D}$
- as before $f_s(\mathrm{xS}) = (\mathrm{xS})F^{(s)}(\mathrm{xS})^T = \mathrm{x}(\mathrm{S}F^{(s)}\mathrm{S}^T)\mathrm{x}^T$ for every quadratic map in $\mathcal{F}$
- so isometry: $F^{(s)} \mapsto \mathrm{S}F^{(s)}\mathrm{S}^{\mathsf{T}} \in \mathcal{D}$, hence $\mathcal{D} = \mathrm{S}\mathcal{C}\mathrm{S}^T$ □

How hard is MCE?

MCE **is at least as hard as** hQMLE

Proof *(Sketch)*.

- Given $\mathcal{F}$ and $\mathcal{P}$ such that $\mathcal{P}(\mathsf{x}) = \mathcal{F}(\mathsf{xS}) \cdot \mathsf{T}$
- $\mathcal{F} = (f_1, \ldots, f_k)$ gives $\langle F^{(1)}, \ldots, F^{(k)} \rangle = \mathcal{C} \subset \mathcal{M}_{N \times N}(\mathbb{F}_q)$, similarly $\mathcal{P}$ gives $\mathcal{D}$
- as before $f_s(\mathsf{xS}) = (\mathsf{xS})F^{(s)}(\mathsf{xS})^T = \mathsf{x}(\mathsf{S}F^{(s)}\mathsf{S}^T)\mathsf{x}^T$ for every quadratic map in $\mathcal{F}$
- so isometry: $F^{(s)} \mapsto \mathsf{S}F^{(s)}\mathsf{S}^\mathsf{T} \in \mathcal{D}$, hence $\mathcal{D} = \mathsf{S}\mathcal{C}\mathsf{S}^T$ □

where is T...?

How hard is MCE?

MCE **is at least as hard as** hQMLE

Proof *(Sketch)*.

- Given $\mathcal{F}$ and $\mathcal{P}$ such that $\mathcal{P}(\mathrm{x}) = \mathcal{F}(\mathrm{xS}) \cdot \mathrm{T}$
- $\mathcal{F} = (f_1, \ldots, f_k)$ gives $\langle F^{(1)}, \ldots, F^{(k)} \rangle = \mathcal{C} \subset \mathcal{M}_{N \times N}(\mathbb{F}_q)$, similarly $\mathcal{P}$ gives $\mathcal{D}$
- as before $f_s(\mathrm{xS}) = (\mathrm{xS})F^{(s)}(\mathrm{xS})^T = \mathrm{x}(\mathrm{S}F^{(s)}\mathrm{S}^T)\mathrm{x}^T$ for every quadratic map in $\mathcal{F}$
- so isometry: $F^{(s)} \mapsto \mathrm{S}F^{(s)}\mathrm{S}^\mathsf{T} \in \mathcal{D}$, hence $\mathcal{D} = \mathrm{S}\mathcal{C}\mathrm{S}^T$ □

where is T...? Just a change of basis for the code $\mathcal{D}$!

How hard is MCE? (the other direction)

hQMLE **is at least as hard as** MCE

Proof *(Sketch)*.

- Given codes $\mathcal{C}, \mathcal{D} \subset \mathcal{M}_{m\times n}(\mathbb{F}_q)$ such that $\mu : \mathcal{C} \to \mathcal{D}$ by $\mathsf{C} \mapsto \mathsf{ACB}$

How hard is MCE? (the other direction)

hQMLE **is at least as hard as** MCE

Proof *(Sketch)*.

- Given codes $\mathcal{C}, \mathcal{D} \subset \mathcal{M}_{m\times n}(\mathbb{F}_q)$ such that $\mu : \mathcal{C} \to \mathcal{D}$ by $C \mapsto ACB$
- Close already... if $n = m$ and $B = A^T$, map to $\mathcal{F}$ and $\mathcal{P}$ for hQMLE

How hard is MCE? (the other direction)

hQMLE **is at least as hard as** MCE

Proof *(Sketch)*.

- Given codes $\mathcal{C}, \mathcal{D} \subset \mathcal{M}_{m\times n}(\mathbb{F}_q)$ such that $\mu : \mathcal{C} \to \mathcal{D}$ by $\mathrm{C} \mapsto \mathrm{ACB}$
- Close already... if $n = m$ and $B = A^T$, map to $\mathcal{F}$ and $\mathcal{P}$ for hQMLE
- Trick: move from $n \times m$ codes to $(n + m) \times (n + m)$ codes!

How hard is MCE? (the other direction)

hQMLE **is at least as hard as** MCE

Proof *(Sketch)*.

- Given codes $\mathcal{C}, \mathcal{D} \subset \mathcal{M}_{m\times n}(\mathbb{F}_q)$ such that $\mu : \mathcal{C} \to \mathcal{D}$ by $\mathsf{C} \mapsto \mathsf{ACB}$
- Close already... if $n = m$ and $B = A^T$, map to $\mathcal{F}$ and $\mathcal{P}$ for hQMLE
- Trick: move from $n \times m$ codes to $(n + m) \times (n + m)$ codes!
- So: $\mathsf{C} \mapsto \left[\begin{smallmatrix} 0 & \mathsf{C} \\ \mathsf{C}^T & 0 \end{smallmatrix}\right]$ maps $\mathcal{C}$ to bigger code $\tilde{\mathcal{C}} \subset \mathcal{M}_{(n+m)\times(n+m)}(\mathbb{F}_q)$, same for $\tilde{\mathcal{D}}$

How hard is MCE? (the other direction)

hQMLE **is at least as hard as** MCE

Proof *(Sketch)*.

- Given codes $\mathcal{C}, \mathcal{D} \subset \mathcal{M}_{m\times n}(\mathbb{F}_q)$ such that $\mu : \mathcal{C} \to \mathcal{D}$ by $\mathsf{C} \mapsto \mathsf{ACB}$
- Close already... if $n = m$ and $B = A^T$, map to $\mathcal{F}$ and $\mathcal{P}$ for hQMLE
- Trick: move from $n \times m$ codes to $(n+m) \times (n+m)$ codes!
- So: $\mathsf{C} \mapsto \left[\begin{smallmatrix} 0 & \mathsf{C} \\ \mathsf{C}^T & 0 \end{smallmatrix}\right]$ maps $\mathcal{C}$ to bigger code $\tilde{\mathcal{C}} \subset \mathcal{M}_{(n+m)\times(n+m)}(\mathbb{F}_q)$, same for $\tilde{\mathcal{D}}$
- Define $\mathsf{S} = \left[\begin{smallmatrix} \mathsf{A} & 0 \\ 0 & \mathsf{B}^T \end{smallmatrix}\right]$

How hard is MCE? (the other direction)

hQMLE **is at least as hard as** MCE

Proof *(Sketch)*.

- Given codes $\mathcal{C}, \mathcal{D} \subset \mathcal{M}_{m\times n}(\mathbb{F}_q)$ such that $\mu : \mathcal{C} \to \mathcal{D}$ by $\mathsf{C} \mapsto \mathsf{ACB}$
- Close already... if $n = m$ and $B = A^T$, map to $\mathcal{F}$ and $\mathcal{P}$ for hQMLE
- Trick: move from $n \times m$ codes to $(n+m) \times (n+m)$ codes!
- So: $\mathsf{C} \mapsto \left[\begin{smallmatrix} 0 & \mathsf{C} \\ \mathsf{C}^T & 0 \end{smallmatrix}\right]$ maps $\mathcal{C}$ to bigger code $\tilde{\mathcal{C}} \subset \mathcal{M}_{(n+m)\times(n+m)}(\mathbb{F}_q)$, same for $\tilde{\mathcal{D}}$
- Define $\mathsf{S} = \left[\begin{smallmatrix} \mathsf{A} & 0 \\ 0 & \mathsf{B}^T \end{smallmatrix}\right]$, then

$$\mathsf{SCS}^T = \left[\begin{smallmatrix} \mathsf{A} & 0 \\ 0 & \mathsf{B}^T \end{smallmatrix}\right]\left[\begin{smallmatrix} 0 & \mathsf{C} \\ \mathsf{C}^T & 0 \end{smallmatrix}\right]\left[\begin{smallmatrix} \mathsf{A}^T & 0 \\ 0 & \mathsf{B} \end{smallmatrix}\right] = \left[\begin{smallmatrix} 0 & \mathsf{ACB} \\ (\mathsf{ACB})^\mathsf{T} & 0 \end{smallmatrix}\right] \in \tilde{\mathcal{D}}$$

How hard is MCE? (the other direction)

hQMLE **is at least as hard as** MCE

Proof *(Sketch)*.

- Given codes $\mathcal{C}, \mathcal{D} \subset \mathcal{M}_{m\times n}(\mathbb{F}_q)$ such that $\mu : \mathcal{C} \to \mathcal{D}$ by $\mathsf{C} \mapsto \mathsf{ACB}$
- Close already... if $n = m$ and $B = A^T$, map to $\mathcal{F}$ and $\mathcal{P}$ for hQMLE
- Trick: move from $n \times m$ codes to $(n+m) \times (n+m)$ codes!
- So: $\mathsf{C} \mapsto \left[\begin{smallmatrix} 0 & \mathsf{C} \\ \mathsf{C}^T & 0 \end{smallmatrix}\right]$ maps $\mathcal{C}$ to bigger code $\tilde{\mathcal{C}} \subset \mathcal{M}_{(n+m)\times(n+m)}(\mathbb{F}_q)$, same for $\tilde{\mathcal{D}}$
- Define $\mathsf{S} = \left[\begin{smallmatrix} \mathsf{A} & 0 \\ 0 & \mathsf{B}^T \end{smallmatrix}\right]$, then

$$\mathsf{SCS}^T = \left[\begin{smallmatrix} \mathsf{A} & 0 \\ 0 & \mathsf{B}^T \end{smallmatrix}\right]\left[\begin{smallmatrix} 0 & \mathsf{C} \\ \mathsf{C}^T & 0 \end{smallmatrix}\right]\left[\begin{smallmatrix} \mathsf{A}^T & 0 \\ 0 & \mathsf{B} \end{smallmatrix}\right] = \left[\begin{smallmatrix} 0 & \mathsf{ACB} \\ (\mathsf{ACB})^\mathsf{T} & 0 \end{smallmatrix}\right] \in \tilde{\mathcal{D}}$$

- so $\tilde{\mu} : \tilde{C} \to \tilde{D}$ gives hQMLE instance with $\mathcal{F}$ and $\mathcal{P}$ in $(n+m)$ variables. □

Solving MCE

Algorithms for QMLE:

- [Faugére, Peret, 2006] – inhomogenous version solved in $\mathcal{O}(N^9)$ (heuristically)

Solving MCE as hQMLE

Algorithms for QMLE:

- [Faugére, Peret, 2006] – inhomogenous version solved in $\mathcal{O}(N^9)$ (heuristically)
- [Bouillaguet, Fouque & Véber, 2013] – **Graph-based approach** $\mathcal{O}(q^{\frac{2}{3}N}N^9)$

Algorithms for QMLE:

- [Faugére, Peret, 2006] – inhomogenous version solved in $\mathcal{O}(N^9)$ (heuristically)
- [Bouillaguet, Fouque & Véber, 2013] – **Graph-based approach** $\mathcal{O}(q^{\frac{2}{3}N}N^9)$
  - turn **homogenous** instance $\mathcal{P}(\mathrm{x}) = \mathcal{F}(\mathrm{xS}) \cdot \mathrm{T}$ to **inhomogenous** using **collision**

Algorithms for QMLE:

- [Faugére, Peret, 2006] – inhomogenous version solved in $\mathcal{O}(N^9)$ (heuristically)
- [Bouillaguet, Fouque & Véber, 2013] – **Graph-based approach** $\mathcal{O}(q^{\frac{2}{3}N}N^9)$
  - turn **homogenous** instance $\mathcal{P}(\mathsf{x}) = \mathcal{F}(\mathsf{xS}) \cdot \mathsf{T}$ to **inhomogenous** using **collision**
    if $\beta = \alpha\mathsf{S}$ for $\alpha, \beta \in \mathbb{F}_q^N$, then $\mathcal{P}(\mathsf{x} + \alpha) = \mathcal{F}(\mathsf{xS} + \beta) \cdot \mathsf{T}$.

Algorithms for QMLE:

- [Faugére, Peret, 2006] – inhomogenous version solved in $\mathcal{O}(N^9)$ (heuristically)
- [Bouillaguet, Fouque & Véber, 2013] – **Graph-based approach** $\mathcal{O}(q^{\frac{2}{3}N}N^9)$
  - turn **homogenous** instance $\mathcal{P}(\mathrm{x}) = \mathcal{F}(\mathrm{xS}) \cdot \mathrm{T}$ to **inhomogenous** using **collision**
    if $\beta = \alpha\mathrm{S}$ for $\alpha, \beta \in \mathbb{F}_q^N$, then $\mathcal{P}(\mathrm{x}+\alpha) = \mathcal{F}(\mathrm{xS}+\beta) \cdot \mathrm{T}$.
    then $\mathcal{P}'(\mathrm{x}) = \mathcal{P}(\mathrm{x}+\alpha)$ and $\mathcal{F}'(\mathrm{x}) = \mathcal{F}(\mathrm{x}+\beta)$ are **easy instance!**

Algorithms for QMLE:

- [Faugére, Peret, 2006] – inhomogenous version solved in $\mathcal{O}(N^9)$ (heuristically)
- [Bouillaguet, Fouque & Véber, 2013] – **Graph-based approach** $\mathcal{O}(q^{\frac{2}{3}N}N^9)$
  - turn **homogenous** instance $\mathcal{P}(\mathrm{x}) = \mathcal{F}(\mathrm{xS}) \cdot \mathrm{T}$ to **inhomogenous** using **collision**
    if $\beta = \alpha\mathrm{S}$ for $\alpha, \beta \in \mathbb{F}_q^N$, then $\mathcal{P}(\mathrm{x}+\alpha) = \mathcal{F}(\mathrm{xS}+\beta) \cdot \mathrm{T}$.
    then $\mathcal{P}'(\mathrm{x}) = \mathcal{P}(\mathrm{x}+\alpha)$ and $\mathcal{F}'(\mathrm{x}) = \mathcal{F}(\mathrm{x}+\beta)$ are **easy instance!**
  - Part 1: find potential collisions (α, β) using birthday-based approach

Algorithms for QMLE:

- [Faugére, Peret, 2006] – inhomogenous version solved in $\mathcal{O}(N^9)$ (heuristically)
- [Bouillaguet, Fouque & Véber, 2013] – **Graph-based approach** $\mathcal{O}(q^{\frac{2}{3}N}N^9)$
  - turn **homogenous** instance $\mathcal{P}(\mathrm{x}) = \mathcal{F}(\mathrm{xS}) \cdot \mathrm{T}$ to **inhomogenous** using **collision**
    if $\beta = \alpha\mathrm{S}$ for $\alpha, \beta \in \mathbb{F}_q^N$, then $\mathcal{P}(\mathrm{x}+\alpha) = \mathcal{F}(\mathrm{xS}+\beta) \cdot \mathrm{T}$.
    then $\mathcal{P}'(\mathrm{x}) = \mathcal{P}(\mathrm{x}+\alpha)$ and $\mathcal{F}'(\mathrm{x}) = \mathcal{F}(\mathrm{x}+\beta)$ are **easy instance!**
  - Part 1: find potential collisions (α, β) using birthday-based approach
  - Part 2: apply inhomogeneous solver to (α, β) until it gives S and T

Algorithms for QMLE:

- [Faugére, Peret, 2006] – inhomogenous version solved in $\mathcal{O}(N^9)$ (heuristically)
- [Bouillaguet, Fouque & Véber, 2013] – **Graph-based approach** $\mathcal{O}(q^{\frac{2}{3}N}N^9)$
  - turn **homogenous** instance $\mathcal{P}(\mathrm{x}) = \mathcal{F}(\mathrm{xS}) \cdot \mathrm{T}$ to **inhomogenous** using **collision**
    if $\beta = \alpha\mathrm{S}$ for $\alpha, \beta \in \mathbb{F}_q^N$, then $\mathcal{P}(\mathrm{x} + \alpha) = \mathcal{F}(\mathrm{xS} + \beta) \cdot \mathrm{T}$.
    then $\mathcal{P}'(\mathrm{x}) = \mathcal{P}(\mathrm{x} + \alpha)$ and $\mathcal{F}'(\mathrm{x}) = \mathcal{F}(\mathrm{x} + \beta)$ are **easy instance!**
  - Part 1: find potential collisions (α, β) using birthday-based approach
  - Part 2: apply inhomogeneous solver to (α, β) until it gives S and T
  - find potential collisions using differential $\mathrm{D}_\alpha\mathcal{F}(x) := \mathcal{F}(\alpha + x) - \mathcal{F}(\alpha) - \mathcal{F}(x)$
  - reduces complexity to $\mathcal{O}(q^{\frac{2}{3}N})$ with $\approx 63\%$ success probability

Algorithms for QMLE:

- [Faugére, Peret, 2006] – inhomogenous version solved in $\mathcal{O}(N^9)$ (heuristically)
- [Bouillaguet, Fouque & Véber, 2013] – **Graph-based approach** $\mathcal{O}(q^{\frac{2}{3}N}N^9)$
  - turn **homogenous** instance $\mathcal{P}(\mathrm{x}) = \mathcal{F}(\mathrm{xS}) \cdot \mathrm{T}$ to **inhomogenous** using **collision**
    if $\beta = \alpha\mathrm{S}$ for $\alpha, \beta \in \mathbb{F}_q^N$, then $\mathcal{P}(\mathrm{x}+\alpha) = \mathcal{F}(\mathrm{xS}+\beta) \cdot \mathrm{T}$.
    then $\mathcal{P}'(\mathrm{x}) = \mathcal{P}(\mathrm{x}+\alpha)$ and $\mathcal{F}'(\mathrm{x}) = \mathcal{F}(\mathrm{x}+\beta)$ are **easy instance!**
  - Part 1: find potential collisions (α, β) using birthday-based approach
  - Part 2: apply inhomogeneous solver to (α, β) until it gives S and T
  - find potential collisions using differential $\mathrm{D}_\alpha\mathcal{F}(x) := \mathcal{F}(\alpha + x) - \mathcal{F}(\alpha) - \mathcal{F}(x)$
  - reduces complexity to $\mathcal{O}(q^{\frac{2}{3}N})$ with $\approx 63\%$ success probability

Complexity of solving MCE: $\mathcal{O}^*(q^{\frac{2}{3}(n+m)})$ (success prob. $\approx 63\%$)

- map $\mathcal{C}, \mathcal{D}$ to $\tilde{\mathcal{C}}, \tilde{\mathcal{D}}$, gives $N = n + m$ instance of hQMLE
- apply above approach, assuming polynomial-time solver for $\mathcal{P}', \mathcal{F}'$

Solving MCE as hQMLE - taking advantage of bilinearity

- Remark: MCE reduces to **bilinear** hQMLE for $\mathrm{x} = (x_1, \ldots, x_n)$ and $\mathrm{y} = (x_{n+1}, \ldots, x_{n+m})$

Solving MCE as hQMLE - taking advantage of bilinearity

- Remark: MCE reduces to **bilinear** hQMLE for $\mathrm{x} = (x_1, \ldots, x_n)$ and $\mathrm{y} = (x_{n+1}, \ldots, x_{n+m})$
- The differential in this case is special, define $\mathrm{F_a}(y) = \mathcal{F}(a, y)$ and $\mathrm{F_b}(x) = \mathcal{F}(x, b)$ then

$$D_{(\mathrm{a},\mathrm{b})}\mathcal{F}(\mathrm{x},\mathrm{y}) = \begin{pmatrix} \mathrm{F_b} & \mathrm{F_a} \end{pmatrix} \begin{pmatrix} \mathrm{x}^\top \\ \mathrm{y}^\top \end{pmatrix}.$$

Solving MCE as hQMLE - taking advantage of bilinearity

- Remark: MCE reduces to **bilinear** hQMLE for $\mathsf{x} = (x_1, \ldots, x_n)$ and $\mathsf{y} = (x_{n+1}, \ldots, x_{n+m})$
- The differential in this case is special, define $\mathsf{F}_\mathsf{a}(y) = \mathcal{F}(a, y)$ and $\mathsf{F}_\mathsf{b}(x) = \mathcal{F}(x, b)$ then

$$D_{(\mathsf{a},\mathsf{b})}\mathcal{F}(\mathsf{x},\mathsf{y}) = \begin{pmatrix} \mathsf{F}_\mathsf{b} & \mathsf{F}_\mathsf{a} \end{pmatrix} \begin{pmatrix} \mathsf{x}^\top \\ \mathsf{y}^\top \end{pmatrix}.$$

- **Idea:** Focus only on the smaller matrices F_b or F_a
- $\mathfrak{F}_b = \{\mathsf{b} \in \mathbb{F}_q^m \mid \dim \ker \mathsf{F}_\mathsf{b} > 0\}$ (similarly $\mathfrak{F}_a$, $\mathfrak{P}_a$, $\mathfrak{P}_b$)

Solving MCE as hQMLE - taking advantage of bilinearity

- Remark: MCE reduces to **bilinear** hQMLE for $\mathsf{x} = (x_1, \ldots, x_n)$ and $\mathsf{y} = (x_{n+1}, \ldots, x_{n+m})$
- The differential in this case is special, define $\mathsf{F_a}(y) = \mathcal{F}(a, y)$ and $\mathsf{F_b}(x) = \mathcal{F}(x, b)$ then

$$D_{(\mathsf{a},\mathsf{b})}\mathcal{F}(\mathsf{x},\mathsf{y}) = \begin{pmatrix} \mathsf{F_b} & \mathsf{F_a} \end{pmatrix} \begin{pmatrix} \mathsf{x}^\top \\ \mathsf{y}^\top \end{pmatrix}.$$

- **Idea:** Focus only on the smaller matrices $\mathsf{F_b}$ or $\mathsf{F_a}$
- $\mathfrak{F}_b = \{\mathsf{b} \in \mathbb{F}_q^m \mid \dim \ker \mathsf{F_b} > 0\}$ (similarly $\mathfrak{F}_a$, $\mathfrak{P}_a$, $\mathfrak{P}_b$)
- Crucial:
  - For $\mathsf{a} \in \ker \mathsf{F_b}$, we have $\mathcal{F}(\mathsf{a}, \mathsf{b}) = 0$
  - $\mathfrak{F}_a \xrightarrow{\mu} \mathfrak{P}_a, \quad \mathfrak{F}_b \xrightarrow{\mu} \mathfrak{P}_b$
  - Can find collisions (α, β) in the roots!

Solving MCE as hQMLE - taking advantage of bilinearity

- Remark: MCE reduces to **bilinear** hQMLE for $\mathrm{x} = (x_1, \ldots, x_n)$ and $\mathrm{y} = (x_{n+1}, \ldots, x_{n+m})$
- The differential in this case is special, define $\mathrm{F_a}(y) = \mathcal{F}(a, y)$ and $\mathrm{F_b}(x) = \mathcal{F}(x, b)$ then

$$D_{(\mathrm{a},\mathrm{b})}\mathcal{F}(\mathrm{x},\mathrm{y}) = \begin{pmatrix} \mathrm{F_b} & \mathrm{F_a} \end{pmatrix} \begin{pmatrix} \mathrm{x}^\top \\ \mathrm{y}^\top \end{pmatrix}.$$

- **Idea:** Focus only on the smaller matrices $\mathrm{F_b}$ or $\mathrm{F_a}$
- $\mathfrak{F}_b = \{\mathrm{b} \in \mathbb{F}_q^m \mid \dim \ker \mathrm{F_b} > 0\}$ (similarly $\mathfrak{F}_a$, $\mathfrak{P}_a$, $\mathfrak{P}_b$)
- Crucial:
  - For $\mathrm{a} \in \ker \mathrm{F_b}$, we have $\mathcal{F}(\mathrm{a}, \mathrm{b}) = 0$
  - $\mathfrak{F}_a \xrightarrow{\mu} \mathfrak{P}_a, \quad \mathfrak{F}_b \xrightarrow{\mu} \mathfrak{P}_b$
  - Can find collisions (α, β) in the roots!
- However:
  - assuming $n \approx m$, number of roots approximately $q^{n+m-k-1}$
  - so, no non-trivial zeros for $k > n + m$ with reasonable probability

Solving MCE as QMLE - taking advantage of bilinearity

Algorithm for MCE "with non-trivial roots" (essentially when $k \leq n + m$):

Solving MCE as QMLE - taking advantage of bilinearity

Algorithm for MCE "with non-trivial roots" (essentially when $k \leq n + m$):

- Turn $\mathcal{C}, \mathcal{D}$ into $\mathcal{F}$ and $\mathcal{P}$ as before

Solving MCE as QMLE - taking advantage of bilinearity

Algorithm for MCE "with non-trivial roots" (essentially when $k \leq n + m$):

- Turn $\mathcal{C}, \mathcal{D}$ into $\mathcal{F}$ and $\mathcal{P}$ as before
- Calculate $\ker \mathsf{F}_\mathsf{b}$ for all $\mathsf{b} \in \mathbb{F}_q^m$
  - if non-trivial, store b with $\ker \mathsf{F}_\mathsf{b}$ (usually 1-dimensional)
- Do the same for $\mathcal{P}$, and feed possible zero-pairs of $\mathcal{F}$ and $\mathcal{P}$ to solver

Complexity of solving MCE with roots: $\mathcal{O}^*(q^m)$

- MCE has roots when $k \leqslant n + m$ with $n = m$
- notice: 100% success instead of $\approx 63\%$
- assumes polynomial-time solver for $\mathcal{P}', \mathcal{F}'$

Solving MCE as QMLE - taking advantage of bilinearity

Algorithm for MCE "with non-trivial roots" (essentially when $k \leq n + m$):

- Turn $\mathcal{C}, \mathcal{D}$ into $\mathcal{F}$ and $\mathcal{P}$ as before
- Calculate $\ker \mathsf{F}_\mathsf{b}$ for all $\mathsf{b} \in \mathbb{F}_q^m$
  - if non-trivial, store b with $\ker \mathsf{F}_\mathsf{b}$ (usually 1-dimensional)
- Do the same for $\mathcal{P}$, and feed possible zero-pairs of $\mathcal{F}$ and $\mathcal{P}$ to solver

Complexity of solving MCE with roots: $\mathcal{O}^*(q^m)$

- MCE has roots when $k \leqslant n + m$ with $n = m$
- notice: 100% success instead of $\approx 63\%$
- assumes polynomial-time solver for $\mathcal{P}', \mathcal{F}'$

TO DO / exercise for the listener:

- Can we use bilineairity of MCE when $k > n + m$?
- Can we use zeros in birthday-based approach?

Experiments

Runtimes of Algorithm 1.

$m = n$	k	Runtime (s) SampleSet	Runtime (s) CollisionFind	**Runtime (s) total**	Success probability
11	22	47	180	**227**	0.79
12	24	76	903	**979**	0.71

Runtimes of Algorithm 2.

$m = n$	k	Runtime (s) SampleZeros	Runtime (s) CollisionFind	**Runtime (s) total**	% instances with roots
11	22	0.15	104	**104**	59%
	20	0.16	93	**93**	99%
12	24	0.32	230	**230**	63%
	22	0.29	235	**235**	99%

Thank you for listening!