Code Equivalence

Violetta Weger

Finite Geometry and Friends Summer School 2025

September 2025

Code Equivalence

Violetta Weger

Finite Geometry and Friends Summer School 2025

September 2025

Finite Friends and Geometry, 2023

Classic Heroes vs. Quantum Avengers

Classical attackers  → Classical heroes 

X ~~RSA~~
X ~~DLP~~
X ~~EC-DLP~~

Quantum attackers  → Quantum heroes  post-quantum

 Lattice-based

 Hash-based

 Multivariate

 Isogeny-based

 Code-based

Finite Friends and Geometry, 2023

Motivation

Put differently

Technology Innovation Institute

On the mathematics of post-quantum cryptography, 2025


LESS
CF-LESS

THANK YOU VERY MUCH FOR THE ATTENTION!

Quantum attackers  → Quantum heroes  post-quantum

 Lattice-based

 Hash-based

 Multivariate

 Isogeny-based

Code-based Cryptography

 Decoding-based

 Code-Equivalence

Code Equivalence - The Problem

Given two codes $\mathcal{C}, \mathcal{C}'$, find a linear isometry φ such that $\varphi(\mathcal{C}) = \mathcal{C}$.

"Is code equivalence easy to decide?" Petrank, Roth. 2002.

Code Equivalence - The Problem

Given two codes $\mathcal{C}, \mathcal{C}'$, find a linear isometry φ such that $\varphi(\mathcal{C}) = \mathcal{C}$.

"Is code equivalence easy to decide?" Petrank, Roth. 2002.

< LESS signature scheme in 2nd round of NIST standardization call

Code Equivalence - The Problem

Given two codes $\mathcal{C}, \mathcal{C}'$, find a linear isometry φ such that $\varphi(\mathcal{C}) = \mathcal{C}$.

"Is code equivalence easy to decide?" Petrank, Roth. 2002.

< LESS signature scheme in 2nd round of NIST standardization call

Plan

- Basics of Coding Theory
- Hardness of Code Equivalence
- Connections to other Problems
- LESS Signature Scheme
- Solvers
- Some new Results
- Introduction to Complexity Theory
- Finite Friends
- Summary

Code Equivalence - Organization

Material:

Lecture Notes

Exercises

$\mathbb{F}_q$ finite field of order q a prime power

$\mathbb{F}_q$ finite field of order q a prime power

Definition ∘ $[n,k]_q$ linear code $\mathcal{C}$: $\mathbb{F}_q$-linear subspace of $\mathbb{F}_q^n$ of dimension k

$\mathbb{F}_q$ finite field of order q a prime power

Definition

- $[n,k]_q$ linear code $\mathcal{C}$: $\mathbb{F}_q$-linear subspace of $\mathbb{F}_q^n$ of dimension k
- $G \in \mathbb{F}_q^{k\times n}$ generator matrix: $\mathcal{C} = \{mG \mid m \in \mathbb{F}_q^k\} = \langle G \rangle$

$\mathbb{F}_q$ finite field of order q a prime power

Definition

- $[n,k]_q$ linear code $\mathcal{C}$: $\mathbb{F}_q$-linear subspace of $\mathbb{F}_q^n$ of dimension k
- $G \in \mathbb{F}_q^{k\times n}$ generator matrix: $\mathcal{C} = \{mG \mid m \in \mathbb{F}_q^k\} = \langle G \rangle$
- $c \in \mathcal{C}$ is codeword

$\mathbb{F}_q$ finite field of order q a prime power

Definition

- $[n,k]_q$ linear code $\mathcal{C}$: $\mathbb{F}_q$-linear subspace of $\mathbb{F}_q^n$ of dimension k
- $G \in \mathbb{F}_q^{k\times n}$ generator matrix: $\mathcal{C} = \{mG \mid m \in \mathbb{F}_q^k\} = \langle G \rangle$
- $c \in \mathcal{C}$ is codeword
- $H \in \mathbb{F}_q^{(n-k)\times n}$ parity-check matrix: $\mathcal{C} = \{x \in \mathbb{F}_q^n \mid xH^\top = 0\} = \ker(H^\top)$

$\mathbb{F}_q$ finite field of order q a prime power

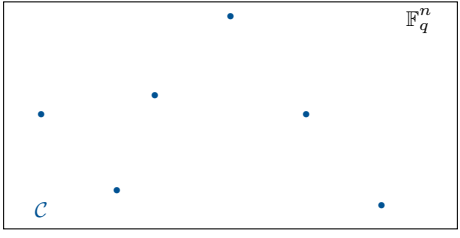
Definition

- $[n,k]_q$ linear code $\mathcal{C}$: $\mathbb{F}_q$-linear subspace of $\mathbb{F}_q^n$ of dimension k
- $G \in \mathbb{F}_q^{k\times n}$ generator matrix: $\mathcal{C} = \{mG \mid m \in \mathbb{F}_q^k\} = \langle G \rangle$
- $c \in \mathcal{C}$ is codeword
- $H \in \mathbb{F}_q^{(n-k)\times n}$ parity-check matrix: $\mathcal{C} = \{x \in \mathbb{F}_q^n \mid xH^\top = 0\} = \ker(H^\top)$
- $xH^\top = s$ is syndrome of x

$\mathbb{F}_q$ finite field of order q a prime power

Definition

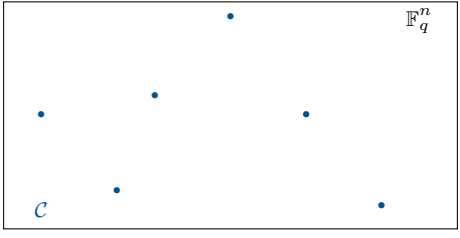
- $[n,k]_q$ linear code $\mathcal{C}$: $\mathbb{F}_q$-linear subspace of $\mathbb{F}_q^n$ of dimension k
- $G \in \mathbb{F}_q^{k\times n}$ generator matrix: $\mathcal{C} = \{mG \mid m \in \mathbb{F}_q^k\} = \langle G \rangle$
- $c \in \mathcal{C}$ is codeword
- $H \in \mathbb{F}_q^{(n-k)\times n}$ parity-check matrix: $\mathcal{C} = \{x \in \mathbb{F}_q^n \mid xH^\top = 0\} = \ker(H^\top)$
- $xH^\top = s$ is syndrome of x

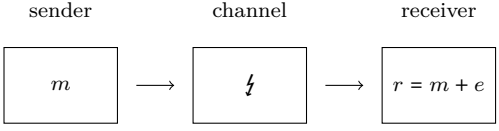


$\mathbb{F}_q$ finite field of order q a prime power

Definition

- $[n,k]_q$ linear code $\mathcal{C}$: $\mathbb{F}_q$-linear subspace of $\mathbb{F}_q^n$ of dimension k
- $G \in \mathbb{F}_q^{k\times n}$ generator matrix: $\mathcal{C} = \{mG \mid m \in \mathbb{F}_q^k\} = \langle G \rangle$
- $c \in \mathcal{C}$ is codeword
- $H \in \mathbb{F}_q^{(n-k)\times n}$ parity-check matrix: $\mathcal{C} = \{x \in \mathbb{F}_q^n \mid xH^\top = 0\} = \ker(H^\top)$
- $xH^\top = s$ is syndrome of x

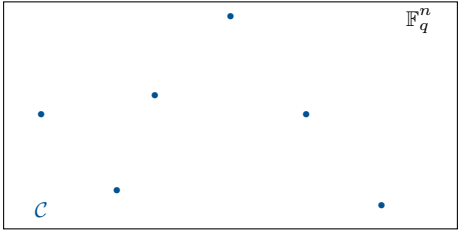


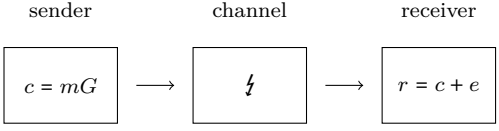


$\mathbb{F}_q$ finite field of order q a prime power

Definition

- $[n,k]_q$ linear code $\mathcal{C}$: $\mathbb{F}_q$-linear subspace of $\mathbb{F}_q^n$ of dimension k
- $G \in \mathbb{F}_q^{k\times n}$ generator matrix: $\mathcal{C} = \{mG \mid m \in \mathbb{F}_q^k\} = \langle G \rangle$
- $c \in \mathcal{C}$ is codeword
- $H \in \mathbb{F}_q^{(n-k)\times n}$ parity-check matrix: $\mathcal{C} = \{x \in \mathbb{F}_q^n \mid xH^\top = 0\} = \ker(H^\top)$
- $xH^\top = s$ is syndrome of x

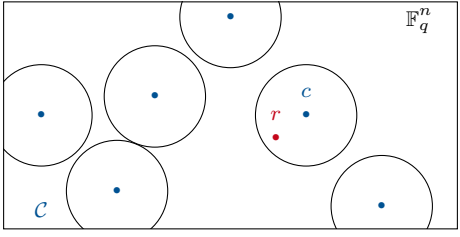


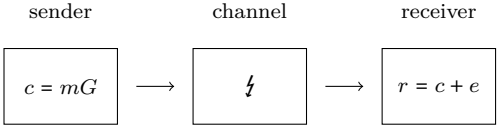


$\mathbb{F}_q$ finite field of order q a prime power

Definition

- $[n,k]_q$ linear code $\mathcal{C}$: $\mathbb{F}_q$-linear subspace of $\mathbb{F}_q^n$ of dimension k
- $G \in \mathbb{F}_q^{k\times n}$ generator matrix: $\mathcal{C} = \{mG \mid m \in \mathbb{F}_q^k\} = \langle G \rangle$
- $c \in \mathcal{C}$ is codeword
- $H \in \mathbb{F}_q^{(n-k)\times n}$ parity-check matrix: $\mathcal{C} = \{x \in \mathbb{F}_q^n \mid xH^\top = 0\} = \ker(H^\top)$
- $xH^\top = s$ is syndrome of x



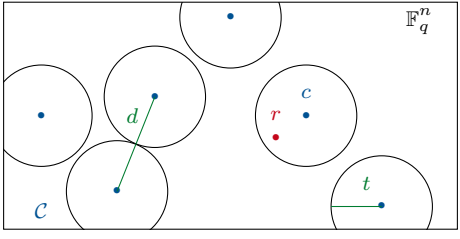


Definition

- The Hamming weight of $x \in \mathbb{F}_q^n$ is $\mathrm{wt}(x) = |\{i \mid x_i \neq 0\}|$
- The Hamming distance between $x, y \in \mathbb{F}_q^n$ is
$$d(x, y) = \mathrm{wt}(x - y) = |\{i \mid x_i \neq y_i\}|$$
- The minimum Hamming distance of $\mathcal{C} \subseteq \mathbb{F}_q^n$ is
$$d(\mathcal{C}) = \min\{\mathrm{wt}(c) \mid c \in \mathcal{C}, c \neq 0\}$$

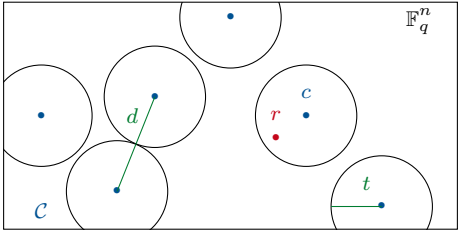
Definition

- The Hamming weight of $x \in \mathbb{F}_q^n$ is $\mathrm{wt}(x) = |\{i \mid x_i \neq 0\}|$
- The Hamming distance between $x, y \in \mathbb{F}_q^n$ is
$$d(x, y) = \mathrm{wt}(x - y) = |\{i \mid x_i \neq y_i\}|$$
- The minimum Hamming distance of $\mathcal{C} \subseteq \mathbb{F}_q^n$ is
$$d(\mathcal{C}) = \min\{\mathrm{wt}(c) \mid c \in \mathcal{C}, c \neq 0\}$$



Definition

- The Hamming weight of $x \in \mathbb{F}_q^n$ is $\mathrm{wt}(x) = |\{i \mid x_i \neq 0\}|$
- The Hamming distance between $x, y \in \mathbb{F}_q^n$ is
$$d(x, y) = \mathrm{wt}(x - y) = |\{i \mid x_i \neq y_i\}|$$
- The minimum Hamming distance of $\mathcal{C} \subseteq \mathbb{F}_q^n$ is
$$d(\mathcal{C}) = \min\{\mathrm{wt}(c) \mid c \in \mathcal{C}, c \neq 0\}$$



A $[n, k, d]_q$ code $\mathcal{C}$ can correct $t = \lfloor \frac{d-1}{2} \rfloor$ errors

$\mathcal{C} = \langle G \rangle = \ker(H^\top) \subseteq \mathbb{F}_q^n$ of dimension k

Definition

- G is in systematic form if $G = \begin{pmatrix} \mathrm{Id}_k & A \end{pmatrix}$
- H is in systematic form if $H = \begin{pmatrix} B & \mathrm{Id}_{n-k} \end{pmatrix}$

Properties

- For $S \in \mathrm{GL}_q(k)$ also $\langle SG \rangle = \mathcal{C}$
- For some permutation matrix P, SGP is in systematic form
- For $S \in \mathrm{GL}_q(n-k)$ also $\ker((SH)^\top) = \mathcal{C}$
- For some permutation matrix P, SHP is in systematic form
- If $G = \begin{pmatrix} \mathrm{Id}_k & A \end{pmatrix}$, then $H = \begin{pmatrix} -A^\top & \mathrm{Id}_{n-k} \end{pmatrix}$

$\mathcal{C} = \langle G \rangle = \ker(H^\top) \subseteq \mathbb{F}_q^n$ of dimension k

Definition

- The dual code of $\mathcal{C}$ is
$$\mathcal{C}^\perp = \{x \in \mathbb{F}_q^n \mid \langle x, y \rangle = 0 \ \forall \ c \in \mathcal{C}\}$$
- $\mathcal{C}^\perp = \langle H \rangle = \ker(G^\top) \subseteq \mathbb{F}_q^n$ of dimension $n - k$
- If $\mathcal{C} = \mathcal{C}^\perp$ then $\mathcal{C}$ is called self-dual
- If $\mathcal{C} \subset \mathcal{C}^\perp$ then $\mathcal{C}$ is called self-orthogonal
- The hull of $\mathcal{C}$ is $\mathcal{H}(\mathcal{C}) = \mathcal{C} \cap \mathcal{C}^\perp$

$\mathcal{C} = \langle G \rangle = \ker(H^\top) \subseteq \mathbb{F}_q^n$ of dimension k

Definition

- The dual code of $\mathcal{C}$ is
$$\mathcal{C}^\perp = \{x \in \mathbb{F}_q^n \mid \langle x, y \rangle = 0 \ \forall \ c \in \mathcal{C}\}$$
- $\mathcal{C}^\perp = \langle H \rangle = \ker(G^\top) \subseteq \mathbb{F}_q^n$ of dimension $n - k$
- If $\mathcal{C} = \mathcal{C}^\perp$ then $\mathcal{C}$ is called self-dual
- If $\mathcal{C} \subset \mathcal{C}^\perp$ then $\mathcal{C}$ is called self-orthogonal
- The hull of $\mathcal{C}$ is $\mathcal{H}(\mathcal{C}) = \mathcal{C} \cap \mathcal{C}^\perp$

Exercises

- Show that $\langle H \rangle = \mathcal{C}^\perp$
- Show that $(\mathcal{C}^\perp)^\perp = \mathcal{C}$.
- Show that if $GG^\top = 0$, then $\mathcal{C}$ is self-orthogonal
- Show that $\mathcal{C}$ is self-dual if and only if $\mathcal{C}$ is self-orthogonal and $n = 2k$
- Show that $\mathcal{H}(\mathcal{C}) = \ker\left(\begin{pmatrix} G \\ H \end{pmatrix}^\top\right)$

How large is this hull?

How large is this hull?

Folklore If $\mathcal{C}$ is random, then $\mathcal{H}(\mathcal{C}) = \{0\}$ with high probability for large n

How large is this hull?

Folklore If $\mathcal{C}$ is random, then $\mathcal{H}(\mathcal{C}) = \{0\}$ with high probability for large n

Theorem If $\mathcal{C}$ is random, then

$$\mathbb{P}(\dim(\mathcal{H}(\mathcal{C})) = h) = \prod_{i=1}^{\infty} q^i \frac{q^i - 1}{q^{2i} - 1} \prod_{i=1}^{n} (q^i - 1)^{-1} \sim (1 - 1/q) q^{-h(h+1)/2}$$

"On the dimension of the hull" N. Sendrier, 1997

How large is this hull?

Folklore If $\mathcal{C}$ is random, then $\mathcal{H}(\mathcal{C}) = \{0\}$ with high probability for large n

Theorem If $\mathcal{C}$ is random, then

$$\mathbb{P}(\dim(\mathcal{H}(\mathcal{C})) = h) = \prod_{i=1}^{\infty} q^i \frac{q^i-1}{q^{2i}-1} \prod_{i=1}^{n} (q^i - 1)^{-1} \sim (1 - 1/q) q^{-h(h+1)/2}$$

"On the dimension of the hull" N. Sendrier, 1997

Theorem If $\mathcal{C}$ is random, then $\mathbb{P}(\mathcal{H}(\mathcal{C}) = \{0\}) \geq 1 - 1/q$ for large n

How large is this hull?

Folklore If $\mathcal{C}$ is random, then $\mathcal{H}(\mathcal{C}) = \{0\}$ with high probability for large n

Theorem If $\mathcal{C}$ is random, then

$$\mathbb{P}(\dim(\mathcal{H}(\mathcal{C})) = h) = \prod_{i=1}^{\infty} q^i \frac{q^i-1}{q^{2i}-1} \prod_{i=1}^{n} (q^i-1)^{-1} \sim (1-1/q)q^{-h(h+1)/2}$$

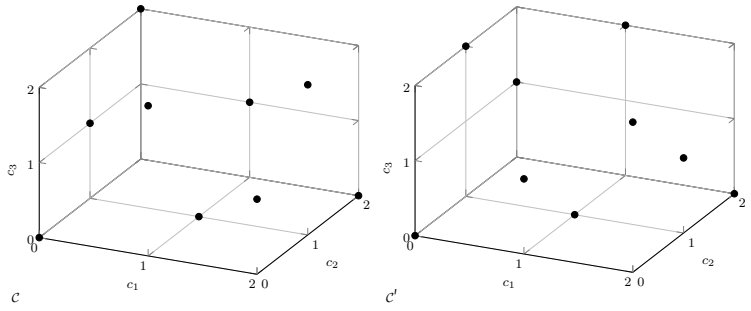
"On the dimension of the hull" N. Sendrier, 1997

Theorem If $\mathcal{C}$ is random, then $\mathbb{P}(\mathcal{H}(\mathcal{C}) = \{0\}) \geq 1 - 1/q$ for large n

Exercise

- If $G = \begin{pmatrix} \mathrm{Id}_k & A \end{pmatrix}$ and $AA^\top + \mathrm{Id}_{n-k}$ has full rank, then $\mathcal{H}(\mathcal{C}) = \{0\}$
- If $GG^\top$ has full rank, then $\mathcal{H}(\mathcal{C}) = \{0\}$

$$G = \begin{pmatrix} 1 & 0 & 2 \\ 0 & 1 & 1 \end{pmatrix}, \quad G' = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}.$$



$$\mathcal{C} = \{(0,0,0),(1,0,2),(2,0,1),(1,1,0),(2,1,2),(0,1,1),(0,2,2),(1,2,1),(2,2,0)\}$$
$$\mathcal{C}' = \{(0,0,0),(0,1,2),(0,2,1),(1,1,0),(1,2,2),(1,0,1),(2,0,2),(2,1,1),(2,2,0)\}$$

Definition ∘ A linear isometry for a distance function d is a linear map

$$\varphi : \mathbb{F}_q^n \to \mathbb{F}_q^n \text{ s.t. } \forall\ x, y \in \mathbb{F}_q^n\colon\ d(x, y) = d(\varphi(x), \varphi(y))$$

Definition ∘ A linear isometry for a distance function d is a linear map

$$\varphi : \mathbb{F}_q^n \to \mathbb{F}_q^n \text{ s.t. } \forall\ x, y \in \mathbb{F}_q^n\text{: } d(x,y) = d(\varphi(x), \varphi(y))$$

Proposition For the Hamming metric: $\varphi \in (\mathbb{F}_q^\star)^n \rtimes S_n$

Definition

- A linear isometry for a distance function d is a linear map
$$\varphi : \mathbb{F}_q^n \to \mathbb{F}_q^n \text{ s.t. } \forall\ x, y \in \mathbb{F}_q^n\text{: } d(x,y) = d(\varphi(x), \varphi(y))$$

Proposition

For the Hamming metric: $\varphi \in (\mathbb{F}_q^\star)^n \rtimes S_n$

Definition

- $\varphi = (d, \sigma) \in (\mathbb{F}_q^\star)^n \rtimes S_n$ called monomial transformation
- $D = \mathrm{diag}(d)$, permutation matrix P, DP called monomial matrix
- The semi-linear isometries are $(\mathbb{F}_q^\star)^n \rtimes (\mathrm{Aut}(\mathbb{F}_q) \times S_n)$

Definition

- A linear isometry for a distance function d is a linear map $\varphi : \mathbb{F}_q^n \to \mathbb{F}_q^n$ s.t. $\forall\ x, y \in \mathbb{F}_q^n$: $d(x, y) = d(\varphi(x), \varphi(y))$

Proposition

For the Hamming metric: $\varphi \in (\mathbb{F}_q^\star)^n \rtimes S_n$

Definition

- $\varphi = (d, \sigma) \in (\mathbb{F}_q^\star)^n \rtimes S_n$ called monomial transformation
- $D = \mathrm{diag}(d)$, permutation matrix P, DP called monomial matrix
- The semi-linear isometries are $(\mathbb{F}_q^\star)^n \rtimes (\mathrm{Aut}(\mathbb{F}_q) \times S_n)$

If $\varphi : \mathcal{C} \to \mathcal{C}'$ linear such that $\mathrm{wt}(c) = \mathrm{wt}(\varphi(c))$ for all $c \in \mathcal{C}$?

Definition

- A linear isometry for a distance function d is a linear map $\varphi : \mathbb{F}_q^n \to \mathbb{F}_q^n$ s.t. $\forall\ x, y \in \mathbb{F}_q^n$: $d(x, y) = d(\varphi(x), \varphi(y))$

Proposition

For the Hamming metric: $\varphi \in (\mathbb{F}_q^\star)^n \rtimes S_n$

Definition

- $\varphi = (d, \sigma) \in (\mathbb{F}_q^\star)^n \rtimes S_n$ called monomial transformation
- $D = \mathrm{diag}(d)$, permutation matrix P, DP called monomial matrix
- The semi-linear isometries are $(\mathbb{F}_q^\star)^n \rtimes (\mathrm{Aut}(\mathbb{F}_q) \times S_n)$

If $\varphi : \mathcal{C} \to \mathcal{C}'$ linear such that $\mathrm{wt}(c) = \mathrm{wt}(\varphi(c))$ for all $c \in \mathcal{C}$?

Theorem

If $\varphi : \mathcal{C} \to \mathcal{C}'$ linear isometry, then exists $\mu \in (\mathbb{F}_q^\star)^n \rtimes S_n$ s.t. $\mu_{|\mathcal{C}} = \varphi$

"Combinatorial problems of elementary abelian groups" F.J. MacWilliams, 1962

Let $\mathcal{C}, \mathcal{C}'$ be $[n,k]_q$ linear codes

Definition

- $\mathcal{C}$ is linearly equivalent to $\mathcal{C}'$ if $\exists \varphi \in (\mathbb{F}_q^\star)^n \rtimes S_n$ s.t. $\varphi(\mathcal{C}) = \mathcal{C}'$
- $\mathcal{C}$ is permutation equivalent to $\mathcal{C}'$ if $\exists \varphi \in S_n$ s.t. $\varphi(\mathcal{C}) = \mathcal{C}'$

Let $\mathcal{C}, \mathcal{C}'$ be $[n,k]_q$ linear codes

Definition

- $\mathcal{C}$ is linearly equivalent to $\mathcal{C}'$ if $\exists \varphi \in (\mathbb{F}_q^\star)^n \rtimes S_n$ s.t. $\varphi(\mathcal{C}) = \mathcal{C}'$
- $\mathcal{C}$ is permutation equivalent to $\mathcal{C}'$ if $\exists \varphi \in S_n$ s.t. $\varphi(\mathcal{C}) = \mathcal{C}'$

Proposition

If $\mathcal{C} = \langle G \rangle$ is linearly equivalent to $\mathcal{C}' = \langle G' \rangle$, then there exist $S \in \mathrm{GL}_q(k), D = \mathrm{diag}(d)$, permutation matrix P, s.t. $SGDP = G'$

Let $\mathcal{C}, \mathcal{C}'$ be $[n,k]_q$ linear codes

Definition

- $\mathcal{C}$ is linearly equivalent to $\mathcal{C}'$ if $\exists \varphi \in (\mathbb{F}_q^\star)^n \rtimes S_n$ s.t. $\varphi(\mathcal{C}) = \mathcal{C}'$
- $\mathcal{C}$ is permutation equivalent to $\mathcal{C}'$ if $\exists \varphi \in S_n$ s.t. $\varphi(\mathcal{C}) = \mathcal{C}'$

Proposition

If $\mathcal{C} = \langle G \rangle$ is linearly equivalent to $\mathcal{C}' = \langle G' \rangle$, then there exist $S \in \mathrm{GL}_q(k), D = \mathrm{diag}(d)$, permutation matrix P, s.t. $SGDP = G'$

Definition

- The automorphism group of $\mathcal{C}$ is $\mathrm{Aut}(\mathcal{C}) = \{\varphi \in (\mathbb{F}_q^\star)^n \rtimes S_n \mid \varphi(\mathcal{C}) = \mathcal{C}\}$

Let $\mathcal{C}, \mathcal{C}'$ be $[n,k]_q$ linear codes

Definition

- $\mathcal{C}$ is linearly equivalent to $\mathcal{C}'$ if $\exists \varphi \in (\mathbb{F}_q^\star)^n \rtimes S_n$ s.t. $\varphi(\mathcal{C}) = \mathcal{C}'$
- $\mathcal{C}$ is permutation equivalent to $\mathcal{C}'$ if $\exists \varphi \in S_n$ s.t. $\varphi(\mathcal{C}) = \mathcal{C}'$

Proposition

If $\mathcal{C} = \langle G \rangle$ is linearly equivalent to $\mathcal{C}' = \langle G' \rangle$, then there exist $S \in \mathrm{GL}_q(k), D = \mathrm{diag}(d)$, permutation matrix P, s.t. $SGDP = G'$

Definition

- The automorphism group of $\mathcal{C}$ is $\mathrm{Aut}(\mathcal{C}) = \{\varphi \in (\mathbb{F}_q^\star)^n \rtimes S_n \mid \varphi(\mathcal{C}) = \mathcal{C}\}$

Property

- If $\mathcal{C}$ is random, then $\mathrm{Aut}(\mathcal{C}) = \{\mathrm{id}\}$ with high probability for large n

"Rigid linear binary codes" H. Lefmann, K. Phelps, V. Rödl, 1993

Let $\mathcal{C}, \mathcal{C}'$ be $[n,k]_q$ linear codes

Proposition If $\varphi \in S_n$ is s.t. $\varphi(\mathcal{C}) = \mathcal{C}'$ then $\varphi(\mathcal{C}^\perp) = \mathcal{C}'^\perp$

Let $\mathcal{C}, \mathcal{C}'$ be $[n,k]_q$ linear codes

Proposition If $\varphi \in S_n$ is s.t. $\varphi(\mathcal{C}) = \mathcal{C}'$ then $\varphi(\mathcal{C}^\perp) = \mathcal{C}'^\perp$

Proposition If $\varphi \in S_n$ is s.t. $\varphi(\mathcal{C}) = \mathcal{C}'$ then $\varphi(\mathcal{H}(\mathcal{C})) = \mathcal{H}(\mathcal{C}')$

Let $\mathcal{C}, \mathcal{C}'$ be $[n,k]_q$ linear codes

Proposition If $\varphi \in S_n$ is s.t. $\varphi(\mathcal{C}) = \mathcal{C}'$ then $\varphi(\mathcal{C}^\perp) = \mathcal{C}'^\perp$

Proposition If $\varphi \in S_n$ is s.t. $\varphi(\mathcal{C}) = \mathcal{C}'$ then $\varphi(\mathcal{H}(\mathcal{C})) = \mathcal{H}(\mathcal{C}')$

Exercises

- If $\varphi \in S_n$ is s.t. $\varphi \in \mathrm{Aut}(\mathcal{C})$ then $\varphi \in \mathrm{Aut}(\mathcal{H}(\mathcal{C}))$
- If $\varphi \in (\mathbb{F}_q^\star)^n \rtimes S_n$ is s.t. $\varphi(\mathcal{C}) = \mathcal{C}'$ then $\exists \varphi' \in (\mathbb{F}_q^\star)^n \rtimes S_n : \varphi'(\mathcal{C}^\perp) = \mathcal{C}'^\perp$

Let $\mathcal{C}, \mathcal{C}'$ be $[n,k]_q$ linear codes

If $\mathcal{C}$ is linearly equivalent to $\mathcal{C}'$, which properties remain the same?

Let $\mathcal{C}, \mathcal{C}'$ be $[n,k]_q$ linear codes

If $\mathcal{C}$ is linearly equivalent to $\mathcal{C}'$, which properties remain the same?

Definition The weight enumerator of $\mathcal{C}$ is $A_w(\mathcal{C}) = |\{c \in \mathcal{C} \mid \mathrm{wt}(c) = w\}|$

Exercise $A_w(\mathcal{C}) = A_w(\mathcal{C}')$ for all $w \in \{1, \ldots, n\}$

Let $\mathcal{C}, \mathcal{C}'$ be $[n,k]_q$ linear codes

If $\mathcal{C}$ is linearly equivalent to $\mathcal{C}'$, which properties remain the same?

Definition The weight enumerator of $\mathcal{C}$ is $A_w(\mathcal{C}) = |\{c \in \mathcal{C} \mid \mathrm{wt}(c) = w\}|$

Exercise $A_w(\mathcal{C}) = A_w(\mathcal{C}')$ for all $w \in \{1, \ldots, n\}$

What about the other direction?

Let $\mathcal{C}, \mathcal{C}'$ be $[n,k]_q$ linear codes

If $\mathcal{C}$ is linearly equivalent to $\mathcal{C}'$, which properties remain the same?

Definition The weight enumerator of $\mathcal{C}$ is $A_w(\mathcal{C}) = |\{c \in \mathcal{C} \mid \mathrm{wt}(c) = w\}|$

Exercise $A_w(\mathcal{C}) = A_w(\mathcal{C}')$ for all $w \in \{1, \ldots, n\}$

What about the other direction?

Proposition $A_w(\mathcal{C}) = A_w(\tilde{\mathcal{C}}) \not\Rightarrow \mathcal{C}$ is linearly equivalent to $\tilde{\mathcal{C}}$

Let $\mathcal{C}, \mathcal{C}'$ be $[n,k]_q$ linear codes

If $\mathcal{C}$ is linearly equivalent to $\mathcal{C}'$, which properties remain the same?

Definition The weight enumerator of $\mathcal{C}$ is $A_w(\mathcal{C}) = |\{c \in \mathcal{C} \mid \mathrm{wt}(c) = w\}|$

Exercise $A_w(\mathcal{C}) = A_w(\mathcal{C}')$ for all $w \in \{1, \ldots, n\}$

What about the other direction?

Proposition $A_w(\mathcal{C}) = A_w(\tilde{\mathcal{C}}) \not\Rightarrow \mathcal{C}$ is linearly equivalent to $\tilde{\mathcal{C}}$

Proposition $|\mathrm{Aut}(\mathcal{C})| = |\mathrm{Aut}(\mathcal{C}')|$

Let $\mathcal{C}, \mathcal{C}'$ be $[n,k]_q$ linear codes

If $\mathcal{C}$ is linearly equivalent to $\mathcal{C}'$, which properties remain the same?

Let $\mathcal{C}, \mathcal{C}'$ be $[n,k]_q$ linear codes

If $\mathcal{C}$ is linearly equivalent to $\mathcal{C}'$, which properties remain the same?

Definition

- The support of $\mathcal{C}$ is $\text{supp}(\mathcal{C}) = \{i \mid \exists c \in \mathcal{C} : c_i \neq 0\}$
- The weight of $\mathcal{C}$ is $\text{wt}(\mathcal{C}) = |\text{supp}(\mathcal{C})|$

Let $\mathcal{C}, \mathcal{C}'$ be $[n,k]_q$ linear codes

If $\mathcal{C}$ is linearly equivalent to $\mathcal{C}'$, which properties remain the same?

Definition

- The support of $\mathcal{C}$ is $\text{supp}(\mathcal{C}) = \{i \mid \exists c \in \mathcal{C} : c_i \neq 0\}$
- The weight of $\mathcal{C}$ is $\text{wt}(\mathcal{C}) = |\text{supp}(\mathcal{C})|$
- Let $r \in \{1, \ldots, k\}$, the rth generalized weight of $\mathcal{C}$ is

$$d_r(\mathcal{C}) = \min\{\text{wt}(\mathcal{D}) \mid \mathcal{D} \subseteq \mathcal{C}, \dim(\mathcal{D}) = r\}$$

Let $\mathcal{C}, \mathcal{C}'$ be $[n,k]_q$ linear codes

If $\mathcal{C}$ is linearly equivalent to $\mathcal{C}'$, which properties remain the same?

Definition

- The support of $\mathcal{C}$ is $\text{supp}(\mathcal{C}) = \{i \mid \exists c \in \mathcal{C} : c_i \neq 0\}$
- The weight of $\mathcal{C}$ is $\text{wt}(\mathcal{C}) = |\text{supp}(\mathcal{C})|$
- Let $r \in \{1, \dots, k\}$, the rth generalized weight of $\mathcal{C}$ is
$$d_r(\mathcal{C}) = \min\{\text{wt}(\mathcal{D}) \mid \mathcal{D} \subseteq \mathcal{C}, \dim(\mathcal{D}) = r\}$$

Exercises

- Show that $d_r(\mathcal{C}) = d_r(\mathcal{C}')$
- For $r \in \{1, \dots, k-1\}$ show that $d_r(\mathcal{C}) < d_{r+1}(\mathcal{C})$

Let $\mathcal{C}$ be an $[n, k, d]_q$ linear code. $\langle G \rangle = \mathcal{C} = \ker(H^\top)$

Summary

- n is called

Let $\mathcal{C}$ be an $[n, k, d]_q$ linear code. $\langle G \rangle = \mathcal{C} = \ker(H^\top)$

Summary

- n is called length

Let $\mathcal{C}$ be an $[n, k, d]_q$ linear code. $\langle G \rangle = \mathcal{C} = \ker(H^\top)$

Summary

- n is called length
- k is called

Let $\mathcal{C}$ be an $[n, k, d]_q$ linear code. $\langle G \rangle = \mathcal{C} = \ker(H^\top)$

Summary

- n is called length
- k is called dimension

Let $\mathcal{C}$ be an $[n, k, d]_q$ linear code. $\langle G \rangle = \mathcal{C} = \ker(H^\top)$

Summary

- n is called length
- k is called dimension
- d is called

Let $\mathcal{C}$ be an $[n, k, d]_q$ linear code. $\langle G \rangle = \mathcal{C} = \ker(H^\top)$

Summary

- n is called length
- k is called dimension
- d is called minimum distance

Let $\mathcal{C}$ be an $[n, k, d]_q$ linear code. $\langle G \rangle = \mathcal{C} = \ker(H^\top)$

Summary

- n is called length
- k is called dimension
- d is called minimum distance
- G is called

Let $\mathcal{C}$ be an $[n, k, d]_q$ linear code. $\langle G \rangle = \mathcal{C} = \ker(H^\top)$

Summary

- n is called length
- k is called dimension
- d is called minimum distance
- G is called generator matrix
- H is called

Let $\mathcal{C}$ be an $[n,k,d]_q$ linear code. $\langle G \rangle = \mathcal{C} = \ker(H^\top)$

Summary

- n is called length
- k is called dimension
- d is called minimum distance
- G is called generator matrix
- H is called parity-check matrix
- $c \in \mathcal{C}$ is called

Let $\mathcal{C}$ be an $[n, k, d]_q$ linear code. $\langle G \rangle = \mathcal{C} = \ker(H^\top)$

Summary

- n is called length
- k is called dimension
- d is called minimum distance
- G is called generator matrix
- H is called parity-check matrix
- $c \in \mathcal{C}$ is called codeword

Let $\mathcal{C}$ be an $[n,k,d]_q$ linear code. $\langle G \rangle = \mathcal{C} = \ker(H^\top)$

Summary

- n is called length
- k is called dimension
- d is called minimum distance
- G is called generator matrix
- H is called parity-check matrix
- $c \in \mathcal{C}$ is called codeword
- $s = xH^\top$ is called

Let $\mathcal{C}$ be an $[n,k,d]_q$ linear code. $\langle G \rangle = \mathcal{C} = \ker(H^\top)$

Summary

- n is called length
- k is called dimension
- d is called minimum distance
- G is called generator matrix
- H is called parity-check matrix
- $c \in \mathcal{C}$ is called codeword
- $s = xH^\top$ is called syndrome

Let $\mathcal{C}$ be an $[n, k, d]_q$ linear code. $\langle G \rangle = \mathcal{C} = \ker(H^\top)$

Summary

- n is called length
- k is called dimension
- d is called minimum distance
- G is called generator matrix
- H is called parity-check matrix
- $c \in \mathcal{C}$ is called codeword
- $s = xH^\top$ is called syndrome
- $\mathcal{C}^\perp$ is called

Let $\mathcal{C}$ be an $[n, k, d]_q$ linear code. $\langle G \rangle = \mathcal{C} = \ker(H^\top)$

Summary

- n is called length
- k is called dimension
- d is called minimum distance
- G is called generator matrix
- H is called parity-check matrix
- $c \in \mathcal{C}$ is called codeword
- $s = xH^\top$ is called syndrome
- $\mathcal{C}^\perp$ is called dual code

Let $\mathcal{C}$ be an $[n,k,d]_q$ linear code. $\langle G \rangle = \mathcal{C} = \ker(H^\top)$

Summary

- n is called length
- k is called dimension
- d is called minimum distance
- G is called generator matrix
- H is called parity-check matrix
- $c \in \mathcal{C}$ is called codeword
- $s = xH^\top$ is called syndrome
- $\mathcal{C}^\perp$ is called dual code
- $\mathcal{H}(\mathcal{C})$ is called

Let $\mathcal{C}$ be an $[n, k, d]_q$ linear code. $\langle G \rangle = \mathcal{C} = \ker(H^\top)$

Summary

- n is called length
- k is called dimension
- d is called minimum distance
- G is called generator matrix
- H is called parity-check matrix
- $c \in \mathcal{C}$ is called codeword
- $s = xH^\top$ is called syndrome
- $\mathcal{C}^\perp$ is called dual code
- $\mathcal{H}(\mathcal{C})$ is called hull

Two $\mathcal{C}, \mathcal{C}'$ $[n,k]_q$ linear codes are said to be

Summary

- linearly equivalent if

Two $\mathcal{C}, \mathcal{C}'$ $[n,k]_q$ linear codes are said to be

Summary

- linearly equivalent if $\exists \varphi \in (\mathbb{F}_q^\star)^n \rtimes S_n : \varphi(\mathcal{C}) = \mathcal{C}'$

Two $\mathcal{C}, \mathcal{C}'$ $[n,k]_q$ linear codes are said to be

Summary

- linearly equivalent if $\exists \varphi \in (\mathbb{F}_q^\star)^n \rtimes S_n : \varphi(\mathcal{C}) = \mathcal{C}'$
- permutation equivalent if

Two $\mathcal{C}, \mathcal{C}'$ $[n,k]_q$ linear codes are said to be

Summary

- linearly equivalent if $\exists \varphi \in (\mathbb{F}_q^\star)^n \rtimes S_n : \varphi(\mathcal{C}) = \mathcal{C}'$
- permutation equivalent if $\exists \varphi \in S_n : \varphi(\mathcal{C}) = \mathcal{C}'$

Two $\mathcal{C}, \mathcal{C}'$ $[n,k]_q$ linear codes are said to be

Summary

- linearly equivalent if $\exists \varphi \in (\mathbb{F}_q^\star)^n \rtimes S_n : \varphi(\mathcal{C}) = \mathcal{C}'$
- permutation equivalent if $\exists \varphi \in S_n : \varphi(\mathcal{C}) = \mathcal{C}'$
- $\exists S \in \mathrm{GL}_q(k), D = \mathrm{diag}(d), P$ perm. matrix, s.t. $SGDP = G'$

Two $\mathcal{C}, \mathcal{C}'$ $[n,k]_q$ linear codes are said to be

Summary

- linearly equivalent if $\exists \varphi \in (\mathbb{F}_q^\star)^n \rtimes S_n : \varphi(\mathcal{C}) = \mathcal{C}'$
- permutation equivalent if $\exists \varphi \in S_n : \varphi(\mathcal{C}) = \mathcal{C}'$
- $\exists S \in \mathrm{GL}_q(k), D = \mathrm{diag}(d), P$ perm. matrix, s.t. $SGDP = G'$

If $\exists \varphi \in S_n$ s.t. $\varphi(\mathcal{C}) = \mathcal{C}' \rightarrow$

Two $\mathcal{C}, \mathcal{C}'$ $[n,k]_q$ linear codes are said to be

Summary

- linearly equivalent if $\exists \varphi \in (\mathbb{F}_q^\star)^n \rtimes S_n : \varphi(\mathcal{C}) = \mathcal{C}'$
- permutation equivalent if $\exists \varphi \in S_n : \varphi(\mathcal{C}) = \mathcal{C}'$
- $\exists S \in \mathrm{GL}_q(k), D = \mathrm{diag}(d), P$ perm. matrix, s.t. $SGDP = G'$

If $\exists \varphi \in S_n$ s.t. $\varphi(\mathcal{C}) = \mathcal{C}' \rightarrow \varphi(\mathcal{C}^\perp) = \mathcal{C}'^\perp$

Two $\mathcal{C}, \mathcal{C}'$ $[n,k]_q$ linear codes are said to be

Summary

- linearly equivalent if $\exists \varphi \in (\mathbb{F}_q^\star)^n \rtimes S_n : \varphi(\mathcal{C}) = \mathcal{C}'$
- permutation equivalent if $\exists \varphi \in S_n : \varphi(\mathcal{C}) = \mathcal{C}'$
- $\exists S \in \mathrm{GL}_q(k), D = \mathrm{diag}(d), P$ perm. matrix, s.t. $SGDP = G'$

If $\exists \varphi \in S_n$ s.t. $\varphi(\mathcal{C}) = \mathcal{C}' \rightarrow \varphi(\mathcal{C}^\perp) = \mathcal{C}'^\perp$

If $\exists \varphi = (D, P) \in (\mathbb{F}_q^\star)^n \rtimes S_n$ s.t. $\varphi(\mathcal{C}) = \mathcal{C}' \rightarrow$

Two $\mathcal{C}, \mathcal{C}'$ $[n,k]_q$ linear codes are said to be

Summary

- linearly equivalent if $\exists \varphi \in (\mathbb{F}_q^\star)^n \rtimes S_n : \varphi(\mathcal{C}) = \mathcal{C}'$
- permutation equivalent if $\exists \varphi \in S_n : \varphi(\mathcal{C}) = \mathcal{C}'$
- $\exists S \in \mathrm{GL}_q(k), D = \mathrm{diag}(d), P$ perm. matrix, s.t. $SGDP = G'$

If $\exists \varphi \in S_n$ s.t. $\varphi(\mathcal{C}) = \mathcal{C}' \rightarrow \varphi(\mathcal{C}^\perp) = \mathcal{C}'^\perp$

If $\exists \varphi = (D, P) \in (\mathbb{F}_q^\star)^n \rtimes S_n$ s.t. $\varphi(\mathcal{C}) = \mathcal{C}' \rightarrow \exists \varphi' = (D^{-1}, P)$ s.t. $\varphi'(\mathcal{C}^\perp) = \mathcal{C}'^\perp$

Two $\mathcal{C}, \mathcal{C}'$ $[n,k]_q$ linear codes are said to be

Summary

- linearly equivalent if $\exists \varphi \in (\mathbb{F}_q^\star)^n \rtimes S_n : \varphi(\mathcal{C}) = \mathcal{C}'$
- permutation equivalent if $\exists \varphi \in S_n : \varphi(\mathcal{C}) = \mathcal{C}'$
- $\exists S \in \mathrm{GL}_q(k), D = \mathrm{diag}(d), P$ perm. matrix, s.t. $SGDP = G'$

If $\exists \varphi \in S_n$ s.t. $\varphi(\mathcal{C}) = \mathcal{C}' \rightarrow \varphi(\mathcal{C}^\perp) = \mathcal{C}'^\perp$

If $\exists \varphi = (D, P) \in (\mathbb{F}_q^\star)^n \rtimes S_n$ s.t. $\varphi(\mathcal{C}) = \mathcal{C}' \rightarrow \exists \varphi' = (D^{-1}, P)$ s.t. $\varphi'(\mathcal{C}^\perp) = \mathcal{C}'^\perp$

Invariants

Two $\mathcal{C}, \mathcal{C}'$ $[n,k]_q$ linear codes are said to be

Summary

- linearly equivalent if $\exists \varphi \in (\mathbb{F}_q^\star)^n \rtimes S_n : \varphi(\mathcal{C}) = \mathcal{C}'$
- permutation equivalent if $\exists \varphi \in S_n : \varphi(\mathcal{C}) = \mathcal{C}'$
- $\exists S \in \mathrm{GL}_q(k), D = \mathrm{diag}(d), P$ perm. matrix, s.t. $SGDP = G'$

If $\exists \varphi \in S_n$ s.t. $\varphi(\mathcal{C}) = \mathcal{C}' \rightarrow \varphi(\mathcal{C}^\perp) = \mathcal{C}'^\perp$

If $\exists \varphi = (D, P) \in (\mathbb{F}_q^\star)^n \rtimes S_n$ s.t. $\varphi(\mathcal{C}) = \mathcal{C}' \rightarrow \exists \varphi' = (D^{-1}, P)$ s.t. $\varphi'(\mathcal{C}^\perp) = \mathcal{C}'^\perp$

Invariants

- Automorphism group $\mathrm{Aut}(\mathcal{C})$

Two $\mathcal{C}, \mathcal{C}'$ $[n,k]_q$ linear codes are said to be

Summary

- linearly equivalent if $\exists \varphi \in (\mathbb{F}_q^\star)^n \rtimes S_n : \varphi(\mathcal{C}) = \mathcal{C}'$
- permutation equivalent if $\exists \varphi \in S_n : \varphi(\mathcal{C}) = \mathcal{C}'$
- $\exists S \in \mathrm{GL}_q(k), D = \mathrm{diag}(d), P$ perm. matrix, s.t. $SGDP = G'$

If $\exists \varphi \in S_n$ s.t. $\varphi(\mathcal{C}) = \mathcal{C}' \rightarrow \varphi(\mathcal{C}^\perp) = \mathcal{C}'^\perp$

If $\exists \varphi = (D, P) \in (\mathbb{F}_q^\star)^n \rtimes S_n$ s.t. $\varphi(\mathcal{C}) = \mathcal{C}' \rightarrow \exists \varphi' = (D^{-1}, P)$ s.t. $\varphi'(\mathcal{C}^\perp) = \mathcal{C}'^\perp$

Invariants

- Automorphism group $\mathrm{Aut}(\mathcal{C})$
- Weight enumerator $A_w(\mathcal{C})$

Two $\mathcal{C}, \mathcal{C}'$ $[n,k]_q$ linear codes are said to be

Summary

- linearly equivalent if $\exists \varphi \in (\mathbb{F}_q^\star)^n \rtimes S_n : \varphi(\mathcal{C}) = \mathcal{C}'$
- permutation equivalent if $\exists \varphi \in S_n : \varphi(\mathcal{C}) = \mathcal{C}'$
- $\exists S \in \mathrm{GL}_q(k), D = \mathrm{diag}(d), P$ perm. matrix, s.t. $SGDP = G'$

If $\exists \varphi \in S_n$ s.t. $\varphi(\mathcal{C}) = \mathcal{C}' \rightarrow \varphi(\mathcal{C}^\perp) = \mathcal{C}'^\perp$

If $\exists \varphi = (D,P) \in (\mathbb{F}_q^\star)^n \rtimes S_n$ s.t. $\varphi(\mathcal{C}) = \mathcal{C}' \rightarrow \exists \varphi' = (D^{-1}, P)$ s.t. $\varphi'(\mathcal{C}^\perp) = \mathcal{C}'^\perp$

Invariants

- Automorphism group $\mathrm{Aut}(\mathcal{C})$
- Weight enumerator $A_w(\mathcal{C})$
- rth generalized weight $d_r(\mathcal{C})$

Goal: secure communication

Goal: secure communication

Symmetric cryptography: both have same key

Goal: secure communication

Symmetric cryptography: both have same key

How to exchange the keys?

Goal: secure communication

Symmetric cryptography: both have same key

How to exchange the keys?

Asymmetric/ public-key cryptography

Public-key encryption (PKE)

Goal: secure communication

Symmetric cryptography: both have same key

How to exchange the keys?

Asymmetric/ public-key cryptography

Key encapsulation mechanism (KEM)

Goal: secure communication

Symmetric cryptography: both have same key

How to exchange the keys?

Asymmetric/ public-key cryptography

Key encapsulation mechanism (KEM)

Signature scheme

Code Equivalence - Cryptography

Signer

Signature Scheme

Verifier

secret key

public key

message m

Code Equivalence - Cryptography

Signer | Signature Scheme | Verifier

secret key → public key

message m

, $m \rightarrow$ signature s

Code Equivalence - Cryptography

Signer | Signature Scheme | Verifier

secret key → public key

message m

, $m \rightarrow$ signature s

, $m, s, \rightarrow$ ✓

Code Equivalence - Cryptography

Signer | Signature Scheme | Verifier

secret key → public key

message m

, $m \rightarrow$ signature s →

, $m, s, \rightarrow$ 

- authentication
- integrity

Code Equivalence - Cryptography

Signer | Signature Scheme | Verifier

secret key → public key

message m

, $m \rightarrow$ signature s →

, $m, s, \rightarrow$

- authentication
- **integrity**

  attacker: recover

  from and many (m, s)

Code Equivalence - Cryptography

Signer | Signature Scheme | Verifier

??? ⟶ ♀ public key

message m

??? , $m \rightarrow$ signature s ⟶

♀ , $m, s, \rightarrow$

- authentication
- **integrity**

  attacker: recover ♀

  from ♀ and many (m, s)

probability of getting accepted:

cheating probability α

Code Equivalence - Cryptography

Prover

Zero-Knowledge Protocol

Verifier

secret key

public key

Code Equivalence - Cryptography

Prover

Zero-Knowledge Protocol

Verifier

secret key

public key

c_0, c_1 commitments

Code Equivalence - Cryptography

Zero-Knowledge Protocol

Prover

secret key

c_0, c_1 commitments

Verifier

public key

challenge $b \in \{0, 1\}$

Prover

Zero-Knowledge Protocol

Verifier

secret key

public key

c_0, c_1 commitments

challenge $b \in \{0, 1\}$

response r_b

Code Equivalence - Cryptography

Prover — Zero-Knowledge Protocol — Verifier

secret key → public key

c_0, c_1 commitments →

← challenge $b \in \{0, 1\}$

response r_b →

check $\varphi, r_b \to c_b$

Code Equivalence - Cryptography

Zero-Knowledge Protocol

Prover		Verifier
		
???	$\longrightarrow$	public key
c_0, c_1 commitments	$\longrightarrow$	
	$\longleftarrow$	challenge $b \in \{0,1\}$
response r_b	$\longrightarrow$	check $, r_b \to c_b$

- zero-knowledge
- complete
- soundness error α

Code Equivalence - Cryptography

Prover — Zero-Knowledge Protocol — Verifier

secret key → public key

c_0, c_1 commitments →

← challenge $b \in \{0, 1\}$

response r_b → check $\mathrm{pk}, r_b \to c_b$

t rounds

- zero-knowledge
- complete
- soundness error $\alpha \to \alpha^t$

Signer

Fiat-Shamir Transform

Verifier

ZK Protocol → Signature Scheme

secret key

public key

c_0, c_1 commitments

$b = \mathsf{Hash}(m, c_0, c_1)$

response r_b

$m, s = (c_0, c_1, r_b)$

$b = \mathsf{Hash}(m, c_0, c_1)$

check $\mathrm{pk}, r_b \to c_b$

Main motivation: LESS

Code Equivalence - Cryptography

Main motivation: LESS

- code-based signature scheme
- 2nd round candidate in NIST call

Main motivation: LESS

- code-based signature scheme
- 2nd round candidate in NIST call

Main motivation: LESS

- code-based signature scheme
- 2nd round candidate in NIST call

- 14 surviving schemes
- 6 code-based schemes

Main motivation: LESS

- code-based signature scheme
- 2nd round candidate in NIST call

- 14 surviving schemes
- 6 code-based schemes

Urgent:

- until 2030 all critical use cases should update
- until 2035 all use cases should update

Main motivation: LESS

- code-based signature scheme
- 2nd round candidate in NIST call

- 14 surviving schemes
- 6 code-based schemes

Urgent:

- until 2030 all critical use cases should update
- until 2035 all use cases should update

Problem:

- Standardizations take time
- All based on novel problems: secure?

Code Equivalence - Cryptography

Prover | LESS ZK-Protocol | Verifier

$\varphi = D, P$ ⟶ G, G' s.t. $SGDP = G'$

Code Equivalence - Cryptography

Prover | LESS ZK-Protocol | Verifier

$\varphi = D, P$

commitment $\tilde{G} = \tilde{\varphi}(G)$

$\longrightarrow$

$\longrightarrow$

G, G' s.t. $SGDP = G'$

Code Equivalence - Cryptography

Prover

LESS ZK-Protocol

Verifier

♀ $\varphi = D, P$

commitment $\tilde{G} = \tilde{\varphi}(G)$

♀ G, G^{l} s.t. $SGDP = G^{l}$

challenge $b \in \{0, 1\}$

Code Equivalence - Cryptography

Prover | LESS ZK-Protocol | Verifier

$\varphi = D, P$ →

G, G^I s.t. $SGDP = G^I$

commitment $\tilde{G} = \tilde{\varphi}(G)$ →

←

challenge $b \in \{0, 1\}$

response $r_0 = \tilde{\varphi}, r_1 = \tilde{\varphi} \circ \varphi^{-1}$ →

Code Equivalence - Cryptography

Prover | LESS ZK-Protocol | Verifier

$\varphi = D, P$ →

commitment $\tilde{G} = \tilde{\varphi}(G)$ →

← challenge $b \in \{0, 1\}$

response $r_0 = \tilde{\varphi}, r_1 = \tilde{\varphi} \circ \varphi^{-1}$ →

G, G' s.t. $SGDP = G'$

check $\tilde{\varphi}(G) = \tilde{G}$ or $\tilde{\varphi} \circ \varphi^{-1}(G') = \tilde{G}$

Code Equivalence - Cryptography

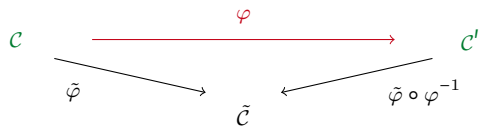
Prover | LESS ZK-Protocol | Verifier

$\varphi = D, P$ → G, G' s.t. $SGDP = G'$

commitment $\tilde{G} = \tilde{\varphi}(G)$ →

← challenge $b \in \{0, 1\}$

response $r_0 = \tilde{\varphi}, r_1 = \tilde{\varphi} \circ \varphi^{-1}$ → check $\tilde{\varphi}(G) = \tilde{G}$ or $\tilde{\varphi} \circ \varphi^{-1}(G') = \tilde{G}$



Prover | LESS ZK-Protocol | Verifier

$\varphi = D, P$

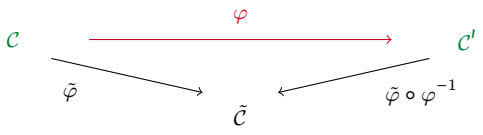
G, G' s.t. $SGDP = G'$

commitment $\tilde{G} = \tilde{\varphi}(G)$

challenge $b \in \{0, 1\}$

response $r_0 = \tilde{\varphi}, r_1 = \tilde{\varphi} \circ \varphi^{-1}$

check $\tilde{\varphi}(G) = \tilde{G}$ or $\tilde{\varphi} \circ \varphi^{-1}(G') = \tilde{G}$



soundness error $\frac{1}{2}$

Set up $\mathcal{P}$ a decisional problem, I an instance, s a solution

Set up $\mathcal{P}$ a decisional problem, I an instance, s a solution

Example Syndrome Decoding Problem:

Given H, s, t, does there exist a e s.t. $eH^\top = s, \mathrm{wt}(e) \leq t$

Instance $=(H, s, t)$ Solution $=$ yes/no

Set up $\mathcal{P}$ a computational problem, I an instance, s a solution

Example Syndrome Decoding Problem:

Given H, s, t, find error vector e s.t. $eH^\top = s, \mathrm{wt}(e) \leq t$

Instance $=(H, s, t)$ Solution $= e$

Set up $\mathcal{P}$ a computational problem, I an instance, s a solution

Example Syndrome Decoding Problem:

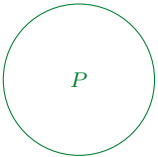
Given H, s, t, find error vector e s.t. $eH^\top = s, \mathrm{wt}(e) \leq t$

Instance $=(H, s, t)$ Solution $= e$

Aim complexity theory: How hard are such problems?

Is SDP harder than sorting / determining minimum distance/ code equivalence?

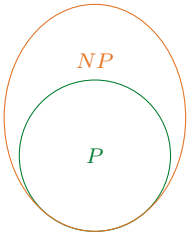
Complexity Classes



- $\mathcal{P} \in P$ if can solve $\mathcal{P}$ in poly. time by a deterministic Turing machine

- polynomial time: $\mathcal{O}(n^c)$ for some constant c

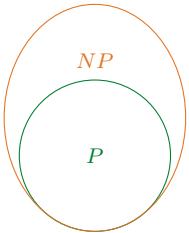
Complexity Classes



- $\mathcal{P} \in P$ if can solve $\mathcal{P}$ in poly. time by a deterministic Turing machine

- polynomial time: $\mathcal{O}(n^c)$ for some constant c

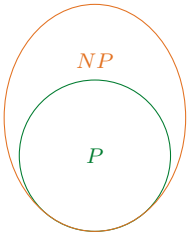
Complexity Classes



- $\mathcal{P} \in P$ if can solve $\mathcal{P}$ in poly. time by a deterministic Turing machine
- $\mathcal{P} \in NP$ if can solve $\mathcal{P}$ in poly. time by a non-deterministic Turing machine

- polynomial time: $\mathcal{O}(n^c)$ for some constant c

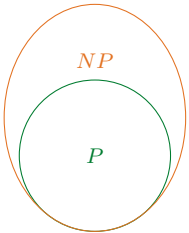
Complexity Classes



- $\mathcal{P} \in P$ if can solve $\mathcal{P}$ in poly. time by a deterministic Turing machine
- $\mathcal{P} \in NP$ if can check candidate is a solution in poly. time

- polynomial time: $\mathcal{O}(n^c)$ for some constant c

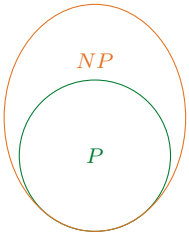
Complexity Classes



- $\mathcal{P} \in P$ if can solve $\mathcal{P}$ in poly. time by a deterministic Turing machine
- $\mathcal{P} \in NP$ if can check candidate is a solution in poly. time
- $P \subset NP$

- polynomial time: $\mathcal{O}(n^c)$ for some constant c

Complexity Classes



- $\mathcal{P} \in P$ if can solve $\mathcal{P}$ in poly. time by a deterministic Turing machine
- $\mathcal{P} \in NP$ if can check candidate is a solution in poly. time
- $P \subset NP$

- polynomial time: $\mathcal{O}(n^c)$ for some constant c
- quasi- polynomial time: $\mathcal{O}(2^{\log(n)^c})$ for some constant c
- exponential time: $\mathcal{O}(2^{nc})$ for some constant c

How to compare hardness of problems?

Polynomial-time reduction form $\mathcal{R}$ to $\mathcal{P}$

1.	take any instance I of $\mathcal{R}$	$\rightarrow$	2.	transform to a instance I' of $\mathcal{P}$
			$\downarrow$	
4.	transform to a solution s of I	$\leftarrow$	3.	oracle gives solution s' to I'

How to compare hardness of problems?

Polynomial-time reduction form $\mathcal{R}$ to $\mathcal{P}$

1.	take any instance I of $\mathcal{R}$	$\rightarrow$	2.	transform to a instance I' of $\mathcal{P}$
			$\downarrow$	
4.	transform to a solution s of I	$\leftarrow$	3.	oracle gives solution s' to I'

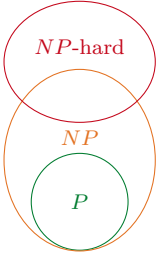
$\rightarrow$ hardness($\mathcal{P}$) $\geq$ hardness($\mathcal{R}$)

How to compare hardness of problems?

Polynomial-time reduction form $\mathcal{R}$ to $\mathcal{P}$

1. take any instance I of $\mathcal{R}$ →
2. transform to a instance I' of $\mathcal{P}$ ↓
3. oracle gives solution s' to I' ←
4. transform to a solution s of I

→ hardness($\mathcal{P}$) ≥ hardness($\mathcal{R}$)



- $\mathcal{P} \in NP$-hard if $\exists$ poly. time reduction from every $\mathcal{R} \in NP$ to $\mathcal{P}$
- NP-complete = $NP \cap NP$-hard
- if $\mathcal{R} \in NP$-hard and $\mathcal{R} \to \mathcal{P}$ then $\mathcal{P} \in NP$-hard

Code Equivalence - Coffee break

Code Equivalence - Exercises

$\mathcal{C} = \langle G \rangle = \ker(H^\top)$ a $[n,k]_q$ linear code

1. Show that $\langle H \rangle = \mathcal{C}^\perp$.
2. Show that $(\mathcal{C}^\perp)^\perp = \mathcal{C}$.
3. Show that if $GG^\top = 0$, then $\mathcal{C}$ is self-orthogonal.
4. Show that $\mathcal{C}$ is self-dual iff $\mathcal{C}$ is self-orthogonal and $n = 2k$.
5. Show that $\mathcal{H}(\mathcal{C}) = \ker\left(\begin{pmatrix} G \\ H \end{pmatrix}^\top\right)$.
6. Let G be in systematic form, i.e., $G = \begin{pmatrix} \mathrm{Id}_k & A \end{pmatrix}$ for $A \in \mathbb{F}_q^{k\times(n-k)}$.
   Show that if $AA^\top + \mathrm{Id}_{n-k}$ is full rank, then $\dim(\mathcal{H}(\mathcal{C})) = 0$.
7. Show that if $GG^\top$ has full rank, then $\dim(\mathcal{H}(\mathcal{C})) = 0$.

$\mathcal{C} = \langle G \rangle$ and $\mathcal{C}' = \langle G' \rangle$

1. Show that the linear isometries form a group with respect to the composition.
2. Give the automorphism group of $\mathcal{C} = \langle (1,0,0), (0,1,1) \rangle \subseteq \mathbb{F}_2^3$.
3. Let $\varphi \in \mathrm{Aut}(\mathcal{C})$ be a permutation. Show that $\varphi \in \mathrm{Aut}(\mathcal{C} \cap \mathcal{C}^\perp)$.
4. Show that $\mathcal{C}^\perp$ is linearly equivalent to $\mathcal{C}'^\perp$.
5. Show that for all $w \in \{1, \ldots, n\}$ we have that $A_w(\mathcal{C}) = A_w(\mathcal{C}')$.
6. Show that for $r \in \{1, \ldots, k-1\}$ we have $d_r(\mathcal{C}) < d_{r+1}(\mathcal{C})$.
7. Show that for all $r \in \{1, \ldots, k\}$ we have that $d_r(\mathcal{C}) = d_r(\mathcal{C}')$.
8. Consider the code $\mathcal{C}_1 \subseteq \mathbb{F}_3^3$ generated by

$$G_1 = \begin{pmatrix} 1 & 0 & 2 \\ 0 & 1 & 1 \end{pmatrix}$$ and the code $\mathcal{C}_2 \subseteq \mathbb{F}_3^3$ generated by $G_2 = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$.

Are the two codes linear equivalent, permutation equivalent or not equivalent?

How hard is code equivalence?

How hard is code equivalence?

Linear Equivalence Problem (LEP)

Given $\mathcal{C}, \mathcal{C}'$ two $[n,k]_q$ linear codes, find $\varphi \in (\mathbb{F}_q^\star)^n \rtimes S_n$ s.t. $\varphi(\mathcal{C}) = \mathcal{C}'$

How hard is code equivalence?

Linear Equivalence Problem (LEP)

Given $\mathcal{C}, \mathcal{C}'$ two $[n,k]_q$ linear codes, find $\varphi \in (\mathbb{F}_q^\star)^n \rtimes S_n$ s.t. $\varphi(\mathcal{C}) = \mathcal{C}'$

Permutation Equivalence Problem (PEP)

Given $\mathcal{C}, \mathcal{C}'$ two $[n,k]_q$ linear codes, find $\varphi \in S_n$ s.t. $\varphi(\mathcal{C}) = \mathcal{C}'$

How hard is code equivalence?

Linear Equivalence Problem (LEP)

Given $\mathcal{C}, \mathcal{C}'$ two $[n,k]_q$ linear codes, find $\varphi \in (\mathbb{F}_q^\star)^n \rtimes S_n$ s.t. $\varphi(\mathcal{C}) = \mathcal{C}'$

Permutation Equivalence Problem (PEP)

Given $\mathcal{C}, \mathcal{C}'$ two $[n,k]_q$ linear codes, find $\varphi \in S_n$ s.t. $\varphi(\mathcal{C}) = \mathcal{C}'$

hardness(LEP) $\geq$ hardness(PEP)

How hard is code equivalence?

Linear Equivalence Problem (LEP)

Given $\mathcal{C}, \mathcal{C}'$ two $[n,k]_q$ linear codes, find $\varphi \in (\mathbb{F}_q^\star)^n \rtimes S_n$ s.t. $\varphi(\mathcal{C}) = \mathcal{C}'$

Permutation Equivalence Problem (PEP)

Given $\mathcal{C}, \mathcal{C}'$ two $[n,k]_q$ linear codes, find $\varphi \in S_n$ s.t. $\varphi(\mathcal{C}) = \mathcal{C}'$

hardness(LEP) $\geq$ hardness(PEP)

Are they NP-hard?

No: any isomorphism problem is not NP-hard

No: any isomorphism problem is not NP-hard

Merlin

Arthur

No: any isomorphism problem is not NP-hard

Merlin

Arthur

- $\mathcal{P} \in$ AM if Merlin can convince Arthur that the answer to instance I is yes
- if PH $\neq$ AM: $\mathcal{P} \in$ co-AM is not NP-hard

No: any isomorphism problem is not NP-hard

Merlin

Arthur

- $\mathcal{P} \in$ co-AM if Merlin can convince Arthur that the answer to instance I is no
- if PH $\neq$ AM: $\mathcal{P} \in$ co-AM is not NP-hard

No: any isomorphism problem is not NP-hard

Merlin

Arthur

- $\mathcal{P} \in$ co-AM if Merlin can convince Arthur that the answer to instance I is no
- if PH $\neq$ AM: $\mathcal{P} \in$ co-AM is not NP-hard

$I = (\mathcal{C}_1, \mathcal{C}_2)$

find $\mathcal{C}_b$ equivalent to $\mathcal{C}$

$\mathcal{C}$ ←

→ b

choose $b \in \{1, 2\}$ and φ

compute $\mathcal{C} = \varphi(\mathcal{C}_b)$

No: any isomorphism problem is not NP-hard

Merlin

Arthur

- $\mathcal{P} \in$ co-AM if Merlin can convince Arthur that the answer to instance I is no
- if PH $\neq$ AM: $\mathcal{P} \in$ co-AM is not NP-hard

$I = (\mathcal{C}_1, \mathcal{C}_2)$

choose $b \in \{1, 2\}$ and φ

$\mathcal{C}$ ←

compute $\mathcal{C} = \varphi(\mathcal{C}_b)$

find $\mathcal{C}_b$ equivalent to $\mathcal{C}$

→ b

soundness error: $1/2$ t rounds $\rightarrow 1/2^t$ $\rightarrow$ LEP $\in$ co-AM

LEP not NP-hard, but is it easy to solve?

LEP not NP-hard, but is it easy to solve?

Solvers Given $G, G' \in \mathbb{F}_q^{k \times n}$ find $S \in \mathrm{GL}_q(k), D = \mathrm{diag}(d), P$ $n \times n$ permutation matrix s.t. $SGDP = G'$

LEP not NP-hard, but is it easy to solve?

Solvers Given $G, G' \in \mathbb{F}_q^{k\times n}$ find $S \in \mathrm{GL}_q(k), D = \mathrm{diag}(d), P\ n \times n$ permutation matrix s.t. $SGDP = G'$

- algebraic solvers

LEP not NP-hard, but is it easy to solve?

Solvers Given $G, G' \in \mathbb{F}_q^{k\times n}$ find $S \in \mathrm{GL}_q(k), D = \mathrm{diag}(d), P$ $n \times n$ permutation matrix s.t. $SGDP = G'$

- algebraic solvers

$$G'H'^{\top} = 0 \text{ and } \langle GDP \rangle = \mathcal{C}' \qquad \rightarrow GDPH'^{\top} = 0$$

LEP not NP-hard, but is it easy to solve?

Solvers Given $G, G' \in \mathbb{F}_q^{k\times n}$ find $S \in \mathrm{GL}_q(k), D = \mathrm{diag}(d), P$ $n \times n$ permutation matrix s.t. $SGDP = G'$

- algebraic solvers

$$G'H'^{\top} = 0 \text{ and } \langle GDP \rangle = \mathcal{C}' \qquad \rightarrow GMH'^{\top} = 0$$

LEP not NP-hard, but is it easy to solve?

Solvers Given $G, G' \in \mathbb{F}_q^{k\times n}$ find $S \in \mathrm{GL}_q(k), D = \mathrm{diag}(d), P$ $n \times n$ permutation matrix s.t. $SGDP = G'$

- algebraic solvers

$G'H'^{\top} = 0$ and $\langle GDP \rangle = \mathcal{C}'$ $\rightarrow GMH'^{\top} = 0$

$\rightarrow k(n-k)$ equations

$\rightarrow n$ variables

LEP not NP-hard, but is it easy to solve?

Solvers Given $G, G' \in \mathbb{F}_q^{k\times n}$ find $S \in \mathrm{GL}_q(k), D = \mathrm{diag}(d), P$ $n \times n$ permutation matrix s.t. $SGDP = G'$

- algebraic solvers

$G'H'^{\top} = 0$ and $\langle GDP \rangle = \mathcal{C}'$ $\rightarrow GMH'^{\top} = 0$

$\rightarrow k(n-k)$ equations

$\rightarrow n^2$ variables

LEP not NP-hard, but is it easy to solve?

Solvers Given $G, G' \in \mathbb{F}_q^{k\times n}$ find $S \in \mathrm{GL}_q(k), D = \mathrm{diag}(d), P$ $n \times n$ permutation matrix s.t. $SGDP = G'$

- algebraic solvers

$G'H'^{\top} = 0$ and $\langle GDP \rangle = \mathcal{C}'$ $\rightarrow GMH'^{\top} = 0$

$\rightarrow k(n-k)$ equations

$\rightarrow n^2$ variables

- combinatorial solvers

  idea: choose subset $S \subset \mathcal{C}, S' \subset \mathcal{C}'$ invariant: $\varphi(S) = S'$

- combinatorial solvers

  idea: choose subset $S \subset \mathcal{C}, S' \subset \mathcal{C}'$ invariant: $\varphi(S) = S'$

  - Leon: weight enumerator

    "Computing automorphism groups of error-correcting codes" J. Leon, 1982

$$S = \{c \in \mathcal{C} \mid \mathrm{wt}(c) = w\} \xrightarrow{\varphi} S' = \{c' \in \mathcal{C}' \mid \mathrm{wt}(c') = w\}$$

- combinatorial solvers

idea: choose subset $S \subset \mathcal{C}, S' \subset \mathcal{C}'$ invariant: $\varphi(S) = S'$

  - Leon: weight enumerator

    "Computing automorphism groups of error-correcting codes" J. Leon, 1982

$$S = \{c \in \mathcal{C} \mid \mathrm{wt}(c) = w\} \xrightarrow{\varphi} S' = \{c' \in \mathcal{C}' \mid \mathrm{wt}(c') = w\}$$

→ cost = cost of solving SDP

- combinatorial solvers

idea: choose subset $S \subset \mathcal{C}, S' \subset \mathcal{C}'$ invariant: $\varphi(S) = S'$

  - Leon: weight enumerator

    "Computing automorphism groups of error-correcting codes" J. Leon, 1982

$$S = \{c \in \mathcal{C} \mid \mathrm{wt}(c) = w\} \xrightarrow{\varphi} S' = \{c' \in \mathcal{C}' \mid \mathrm{wt}(c') = w\}$$

→ cost = cost of solving SDP (NP-hard)

- combinatorial solvers

idea: choose subset $S \subset \mathcal{C}, S' \subset \mathcal{C}'$ invariant: $\varphi(S) = S'$

  - Leon: weight enumerator

    "Computing automorphism groups of error-correcting codes" J. Leon, 1982

$$S = \{c \in \mathcal{C} \mid \mathrm{wt}(c) = w\} \xrightarrow{\varphi} S' = \{c' \in \mathcal{C}' \mid \mathrm{wt}(c') = w\}$$

→ cost = cost of solving SDP $\in \mathcal{O}(2^{nc})$ (NP-hard)

- combinatorial solvers

idea: choose subset $S \subset \mathcal{C}, S' \subset \mathcal{C}'$ invariant: $\varphi(S) = S'$

  - Leon: weight enumerator

    "Computing automorphism groups of error-correcting codes" J. Leon, 1982

$$S = \{c \in \mathcal{C} \mid \mathrm{wt}(c) = w\} \xrightarrow{\varphi} S' = \{c' \in \mathcal{C}' \mid \mathrm{wt}(c') = w\}$$

→ cost = cost of solving SDP $\in \mathcal{O}(2^{nc})$ (NP-hard)

  - Beullens: 2nd generalized weight

    "Not enough LESS" W. Beullens, 2020

$$S = \{\mathcal{D} < \mathcal{C} \mid \dim(\mathcal{D}) = 2, \mathrm{wt}(\mathcal{D}) = w\} \xrightarrow{\varphi} S' = \{\mathcal{D}' < \mathcal{C}' \mid \dim(\mathcal{D}') = 2, \mathrm{wt}(\mathcal{D}') = w\}$$

- combinatorial solvers

idea: choose subset $S \subset \mathcal{C}, S' \subset \mathcal{C}'$ invariant: $\varphi(S) = S'$

  - Leon: weight enumerator

    "Computing automorphism groups of error-correcting codes" J. Leon, 1982

$$S = \{c \in \mathcal{C} \mid \mathrm{wt}(c) = w\} \xrightarrow{\varphi} S' = \{c' \in \mathcal{C}' \mid \mathrm{wt}(c') = w\}$$

→ cost = cost of solving SDP $\in \mathcal{O}(2^{nc})$ (NP-hard)

  - Beullens: 2nd generalized weight

    "Not enough LESS" W. Beullens, 2020

$$S = \{\mathcal{D} < \mathcal{C} \mid \dim(\mathcal{D}) = 2, \mathrm{wt}(\mathcal{D}) = w\} \xrightarrow{\varphi} S' = \{\mathcal{D}' < \mathcal{C}' \mid \dim(\mathcal{D}') = 2, \mathrm{wt}(\mathcal{D}') = w\}$$

→ cost = cost of solving SDP

- combinatorial solvers

idea: choose subset $S \subset \mathcal{C}, S' \subset \mathcal{C}'$ invariant: $\varphi(S) = S'$

  - Leon: weight enumerator

    "Computing automorphism groups of error-correcting codes" J. Leon, 1982

$$S = \{c \in \mathcal{C} \mid \mathrm{wt}(c) = w\} \xrightarrow{\varphi} S' = \{c' \in \mathcal{C}' \mid \mathrm{wt}(c') = w\}$$

→ cost = cost of solving SDP $\in \mathcal{O}(2^{nc})$ (NP-hard)

  - Beullens: 2nd generalized weight

    "Not enough LESS" W. Beullens, 2020

$$S = \{\mathcal{D} < \mathcal{C} \mid \dim(\mathcal{D}) = 2, \mathrm{wt}(\mathcal{D}) = w\} \xrightarrow{\varphi} S' = \{\mathcal{D}' < \mathcal{C}' \mid \dim(\mathcal{D}') = 2, \mathrm{wt}(\mathcal{D}') = w\}$$

→ cost = cost of solving SDP $\in \mathcal{O}(2^{nc})$

- Sendrier: Support Splitting Algorithm (SSA)

  "The support splitting algorithm" N. Sendrier, 2002

Code Equivalence - Solvers

- Sendrier: Support Splitting Algorithm (SSA)

  "The support splitting algorithm" N. Sendrier, 2002

→ only for PEP: puncture in position i: $\mathcal{P}(\mathcal{C}, \{i\}) = \mathcal{C}_i$ and consider the hull

- Sendrier: Support Splitting Algorithm (SSA)

  "The support splitting algorithm" N. Sendrier, 2002

→ only for PEP: puncture in position i: $\mathcal{P}(\mathcal{C}, \{i\}) = \mathcal{C}_i$ and consider the hull

$$\mathcal{H}(\mathcal{C}_i) \xrightarrow{\varphi} \mathcal{H}(\mathcal{C}'_j)$$

- Sendrier: Support Splitting Algorithm (SSA)

  "The support splitting algorithm" N. Sendrier, 2002

→ only for PEP: puncture in position i: $\mathcal{P}(\mathcal{C}, \{i\}) = \mathcal{C}_i$ and consider the hull

$$\mathcal{H}(\mathcal{C}_i) \xrightarrow{\varphi} \mathcal{H}(\mathcal{C}'_j)$$

→ cost $\in \mathcal{O}(q^{\dim(\mathcal{H}(\mathcal{C}))})$

Code Equivalence - Solvers

- Sendrier: Support Splitting Algorithm (SSA)

  "The support splitting algorithm" N. Sendrier, 2002

→ only for PEP: puncture in position i: $\mathcal{P}(\mathcal{C}, \{i\}) = \mathcal{C}_i$ and consider the hull

$$\mathcal{H}(\mathcal{C}_i) \xrightarrow{\varphi} \mathcal{H}(\mathcal{C}'_j)$$

→ cost $\in \mathcal{O}(q^{\dim(\mathcal{H}(\mathcal{C}))})$

- $\mathcal{C}$ random then $\dim(\mathcal{H}(\mathcal{C}))$ constant w.h.p. → PEP is easy for random codes

Code Equivalence - Solvers

- Sendrier: Support Splitting Algorithm (SSA)

  "The support splitting algorithm" N. Sendrier, 2002

→ only for PEP: puncture in position i: $\mathcal{P}(\mathcal{C}, \{i\}) = \mathcal{C}_i$ and consider the hull

$$\mathcal{H}(\mathcal{C}_i) \xrightarrow{\varphi} \mathcal{H}(\mathcal{C}'_j)$$

→ cost $\in \mathcal{O}(q^{\dim(\mathcal{H}(\mathcal{C}))})$

- $\mathcal{C}$ random then $\dim(\mathcal{H}(\mathcal{C}))$ constant w.h.p. → PEP is easy for random codes

→ if $\mathcal{C}$ has constant hull → polynomial time solver

- Sendrier: Support Splitting Algorithm (SSA)

  "The support splitting algorithm" N. Sendrier, 2002

→ only for PEP: puncture in position i: $\mathcal{P}(\mathcal{C}, \{i\}) = \mathcal{C}_i$ and consider the hull

$$\mathcal{H}(\mathcal{C}_i) \xrightarrow{\varphi} \mathcal{H}(\mathcal{C}'_j)$$

→ cost $\in \mathcal{O}(q^{\dim(\mathcal{H}(\mathcal{C}))})$

- $\mathcal{C}$ random then $\dim(\mathcal{H}(\mathcal{C}))$ constant w.h.p. → PEP is easy for random codes
- → if $\mathcal{C}$ has constant hull → polynomial time solver
- if puncture in information set I → $\mathcal{H}(\mathcal{C}_{I^C}) = \{0\}$

- Sendrier: Support Splitting Algorithm (SSA)

  "The support splitting algorithm" N. Sendrier, 2002

→ only for PEP: puncture in position i: $\mathcal{P}(\mathcal{C}, \{i\}) = \mathcal{C}_i$ and consider the hull

$$\mathcal{H}(\mathcal{C}_i) \xrightarrow{\varphi} \mathcal{H}(\mathcal{C}'_j)$$

→ cost $\in \mathcal{O}(q^{\dim(\mathcal{H}(\mathcal{C}))})$

- $\mathcal{C}$ random then $\dim(\mathcal{H}(\mathcal{C}))$ constant w.h.p. → PEP is easy for random codes
- → if $\mathcal{C}$ has constant hull → polynomial time solver
- if puncture in information set $I \rightarrow \mathcal{H}(\mathcal{C}_{I^C}) = \{0\}$
- → only need to find $\varphi(I)$ to puncture also $\mathcal{C}'$

- Sendrier: Support Splitting Algorithm (SSA)

  "The support splitting algorithm" N. Sendrier, 2002

→ only for PEP: puncture in position i: $\mathcal{P}(\mathcal{C}, \{i\}) = \mathcal{C}_i$ and consider the hull

$$\mathcal{H}(\mathcal{C}_i) \xrightarrow{\varphi} \mathcal{H}(\mathcal{C}'_j)$$

→ cost $\in \mathcal{O}(q^{\dim(\mathcal{H}(\mathcal{C}))})$

- $\mathcal{C}$ random then $\dim(\mathcal{H}(\mathcal{C}))$ constant w.h.p. → PEP is easy for random codes
- → if $\mathcal{C}$ has constant hull → polynomial time solver
- if puncture in information set $I \rightarrow \mathcal{H}(\mathcal{C}_{I^C}) = \{0\}$
- → only need to find $\varphi(I)$ to puncture also $\mathcal{C}'$
- → if we know $\varphi(I)$ → easy

- Sendrier: Support Splitting Algorithm (SSA)

  "The support splitting algorithm" N. Sendrier, 2002

→ only for PEP: puncture in position i: $\mathcal{P}(\mathcal{C}, \{i\}) = \mathcal{C}_i$ and consider the hull

$$\mathcal{H}(\mathcal{C}_i) \xrightarrow{\varphi} \mathcal{H}(\mathcal{C}'_j)$$

→ cost $\in \mathcal{O}(q^{\dim(\mathcal{H}(\mathcal{C}))})$

- $\mathcal{C}$ random then $\dim(\mathcal{H}(\mathcal{C}))$ constant w.h.p. → PEP is easy for random codes

→ if $\mathcal{C}$ has constant hull → polynomial time solver

- if puncture in information set I → $\mathcal{H}(\mathcal{C}_{I^C}) = \{0\}$

→ only need to find $\varphi(I)$ to puncture also $\mathcal{C}'$

→ if we know $\varphi(I)$ → easy

→ other solvers using canonical forms → cost $\in \mathcal{O}\left(\sqrt{\binom{n}{k}}\right)$

"On linear equivalence, canonical forms, and digital signatures", T. Chou, E. Persichetti, P. Santini, 2025

Summary

- LEP, PEP not NP-hard

Summary

- LEP, PEP not NP-hard
- solvers for LEP have exponential cost

Summary

- LEP, PEP not NP-hard
- solvers for LEP have exponential cost
- solvers for PEP have cost in $\mathcal{O}(q^{\dim(\mathcal{H}(\mathcal{C}))})$

Summary

- LEP, PEP not NP-hard
- solvers for LEP have exponential cost
- solvers for PEP have cost in $\mathcal{O}(q^{\dim(\mathcal{H}(\mathcal{C}))})$
- PEP easy for random codes

Summary

- LEP, PEP not NP-hard
- solvers for LEP have exponential cost
- solvers for PEP have cost in $\mathcal{O}(q^{\dim(\mathcal{H}(\mathcal{C}))})$
- PEP easy for random codes
- PEP hardest instance: self-orthogonal codes $\mathcal{H}(\mathcal{C}) = \mathcal{C}$

Code Equivalence - Different View Point

Finite Geometry

Finite Geometry

Definition Finite projective geometry of dimension k and order q

$$\mathrm{PG}(k,q) = (\mathbb{F}_q^{k+1} \setminus \{0\})/\sim$$

where $u \sim v$ iff $u = \lambda v$ for some $\lambda \in \mathbb{F}_q^\star$

Finite Geometry

Definition Finite projective geometry of dimension k and order q

$$\mathrm{PG}(k,q) = (\mathbb{F}_q^{k+1} \setminus \{0\})/\sim$$

where $u \sim v$ iff $u = \lambda v$ for some $\lambda \in \mathbb{F}_q^\star$

Definition $\mathcal{M}$ is a projective $[n,k,d]_q$ system if $\mathcal{M}$ is a finite set of n points of $\mathrm{PG}(k-1,q)$ not all on a hyperplane and

$$d = n - \max\{|H \cap \mathcal{M}| \mid H \subseteq \mathrm{PG}(k-1,q), \dim(H) = k-2\}$$

Code Equivalence - Different View Point

Connection

Code Equivalence - Different View Point

Connection

$$G = \begin{pmatrix} g_{1,1} & \cdots & g_{1,n} \\ \vdots & & \vdots \\ g_{k,1} & \cdots & g_{k,n} \end{pmatrix}$$

$\mathcal{C}$ a $[n, k, d]_q$ linear non-degenerate code

Code Equivalence - Different View Point

Connection

$$G = \begin{pmatrix} g_{1,1} & \cdots & g_{1,n} \\ \vdots & & \vdots \\ g_{k,1} & \cdots & g_{k,n} \end{pmatrix}$$

$\mathcal{C}$ a $[n,k,d]_q$ linear non-degenerate code

$$G = \begin{pmatrix} g_{1,1} & \cdots & g_{1,n} \\ \vdots & & \vdots \\ g_{k,1} & \cdots & g_{k,n} \end{pmatrix}$$

$\mathcal{M}$ a projective $[n,k,d]_q$ system

Code Equivalence - Different View Point

Matroids

Matroids

Definition A matroid M is a pair (E, I) where E is a finite set and I is a collection of subsets of E, called independent sets, s.t.

1. $\varnothing \in I$
2. if $A \in I, B \subseteq A$ then $B \in I$
3. if $A, B \in I$, $|A| < |B|$, then $\exists b \in B \setminus A$ s.t. $A \cup \{b\} \in I$

Matroids

Definition A matroid M is a pair (E, I) where E is a finite set and I is a collection of subsets of E, called independent sets, s.t.

1. $\varnothing \in I$
2. if $A \in I, B \subseteq A$ then $B \in I$
3. if $A, B \in I$, $|A| < |B|$, then $\exists b \in B \setminus A$ s.t. $A \cup \{b\} \in I$

Connection

$G \in \mathbb{F}_q^{k \times n}$ generator matrix $\rightarrow$ representable matroid $M(G) = (E, I)$ where $E = \{1, \ldots, n\}$ and $I = \{S \subset E \mid G_S \text{ has full rank }\}$

Code Equivalence - Different View Point

Matroids

Matroids

Definition

A matroid M is a pair (E, r) where E is a finite set and $r : \mathcal{P}(E) \to \mathbb{N}_0$ is a rank function, s.t.

1. $0 \leq r(X) \leq |X|$ for all $X \subseteq E$
2. if $X \subseteq Y \subseteq E$ then $r(X) \leq r(Y)$
3. for all $X, Y \subseteq E$: $r(X \cup Y) + r(X \cap Y) \leq r(X) + r(Y)$

Matroids

Definition A matroid M is a pair (E, r) where E is a finite set and $r : \mathcal{P}(E) \to \mathbb{N}_0$ is a rank function, s.t.

1. $0 \leq r(X) \leq |X|$ for all $X \subseteq E$
2. if $X \subseteq Y \subseteq E$ then $r(X) \leq r(Y)$
3. for all $X, Y \subseteq E$: $r(X \cup Y) + r(X \cap Y) \leq r(X) + r(Y)$

Connection

$G \in \mathbb{F}_q^{k \times n}$ generator matrix $\to$ representable matroid $M(G) = (E, I)$ where $E = \{1, \ldots, n\}$ and for all $S \in \mathcal{P}(E)$: $r(S) = \dim(\langle G_S \rangle)$

Code Equivalence - Different View Point

Designs

Designs

Definition A $t-(v,k,\lambda)$ design is a pair (X,B), where X = set of v points
B = collection of k-elements subsets of X (blocks), s.t.
every t-element subset of X is contained in exactly λ blocks

Code Equivalence - Different View Point

Designs

Definition A $t-(v,k,\lambda)$ design is a pair (X,B), where X = set of v points
B = collection of k-elements subsets of X (blocks), s.t.
every t-element subset of X is contained in exactly λ blocks

Connection

$\mathcal{C}$ a $[n,k,d]_q$ linear code $\rightarrow$ $X = \{1,\ldots,n\}$ and

$$B = \{\mathrm{supp}(c_1),\ldots,\mathrm{supp}(c_N) \mid c_i \in \mathcal{C}, \mathrm{wt}(c_i) = d\}$$

Designs

Assmus-Mattson Theorem

$\mathcal{C}$ a $[n, k, d]_q$ linear code with weight enumerators A_i

$\mathcal{C}^{\perp}$ a $[n, n-k, d']_q$ linear code with weight enumerators A'_i

For $t < d$, s the number of $i < n - t$ s.t. $A'_i \neq 0$

If $s \leq d - t$, then the supports of all codewords in $\mathcal{C}$ of weight u with $d \leq u \leq n$ form a t-design

"New 5-designs" E.F. Assmus, H.F. Mattson, 1969

Reductions

- PEP → LEP ✓

Reduction $\mathcal{R} \to \mathcal{P}$
if can solve $\mathcal{P} \to$ can solve $\mathcal{R}$
hardness($\mathcal{P}$) ≥ hardness($\mathcal{R}$)

Reductions

- PEP → LEP ✓
- LEP → PEP

Reduction $\mathcal{R} \to \mathcal{P}$
if can solve $\mathcal{P} \to$ can solve $\mathcal{R}$
hardness$(\mathcal{P})$ ≥ hardness$(\mathcal{R})$

Reductions

- PEP → LEP ✓
- LEP → PEP
- PEP → GI

Reduction $\mathcal{R} \to \mathcal{P}$
if can solve $\mathcal{P} \to$ can solve $\mathcal{R}$
hardness$(\mathcal{P})$ ≥ hardness$(\mathcal{R})$

Reductions

- PEP → LEP ✓
- LEP → PEP

- PEP → GI
- GI → PEP

Reduction $\mathcal{R} \to \mathcal{P}$
if can solve $\mathcal{P} \to$ can solve $\mathcal{R}$
hardness($\mathcal{P}$) ≥ hardness($\mathcal{R}$)

favorite finite friend: graphs

Reduction from LEP to PEP

Reduction from LEP to PEP

Definition $\mathcal{C}$ a $[n,k]_q$ linear code, $\alpha \in \mathbb{F}_q$ be a primitive element and $\lambda = (1, \alpha, \ldots, \alpha^{q-2}) \in \mathbb{F}_q^{q-1}$. The closure of $\mathcal{C}$ is $\lambda \otimes \mathcal{C}$

"How easy is code equivalence over $\mathbb{F}_q$?" N. Sendrier, D. Simos, 2013

Reduction from LEP to PEP

Definition $\mathcal{C}$ a $[n,k]_q$ linear code, $\alpha \in \mathbb{F}_q$ be a primitive element and $\lambda = (1, \alpha, \ldots, \alpha^{q-2}) \in \mathbb{F}_q^{q-1}$. The closure of $\mathcal{C}$ is $\lambda \otimes \mathcal{C}$

"How easy is code equivalence over $\mathbb{F}_q$?" N. Sendrier, D. Simos, 2013

Proposition $\mathcal{C}, \mathcal{C}'$ $[n,k]_q$ linear codes, $\varphi \in (\mathbb{F}_q^\star)^n \rtimes S_n$ s.t. $\varphi(\mathcal{C}) = \mathcal{C}'$
Then exists $\sigma \in S_{n(q-1)}$ s.t. $\sigma(\lambda \otimes \mathcal{C}) = \lambda \otimes \mathcal{C}'$

Reduction from PEP to GI

Reduction from PEP to GI

Definition A graph $\mathcal{G}$ is a pair (V, E) with vertices V and edges $E \subseteq V \times V$

Reduction from PEP to GI

Definition A graph $\mathcal{G}$ is a pair (V, E) with vertices V and edges $E \subseteq V \times V$

Definition Two graphs $\mathcal{G} = (V, E)$ and $\mathcal{G}' = (V', E')$ are isomorphic if
$\exists f : V \to V' \ \{u, v\} \in E \leftrightarrow \{f(u), f(v)\} \in E'$

Reduction from PEP to GI

Definition A undirected, weighted graph $\mathcal{G} = (V, E)$ is s.t.
with $\{u, v\} \in E$ iff $\{v, u\} \in E$ and edges have weight $w(u, v)$

Definition Two graphs $\mathcal{G} = (V, E)$ and $\mathcal{G}' = (V', E')$ are isomorphic if
$\exists f : V \to V'$ with $\{u, v\} \in E \leftrightarrow \{f(u), f(v)\} \in E'$
and $w(u, v) = w(f(u), f(v))$

Reduction from PEP to GI

Definition A undirected, weighted graph $\mathcal{G} = (V, E)$ is s.t.
with $\{u, v\} \in E$ iff $\{v, u\} \in E$ and edges have weight $w(u, v)$

Definition Two graphs $\mathcal{G} = (V, E)$ and $\mathcal{G}' = (V', E')$ are isomorphic if
$\exists f : V \to V'$ with $\{u, v\} \in E \leftrightarrow \{f(u), f(v)\} \in E'$
and $w(u, v) = w(f(u), f(v))$

Reduction from PEP to GI

Graph Isomorphism (GI) Problem

Given $\mathcal{G} = (V, E), \mathcal{G}' = (V, E')$ with $V = \{1, \ldots, n\}$

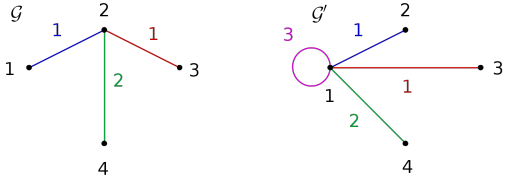
find $\varphi \in S_n$ s.t. $\{u, v\} \in E \leftrightarrow \{\varphi(u), \varphi(v)\} \in E'$

Reduction from PEP to GI

Graph Isomorphism (GI) Problem

Given $\mathcal{G} = (V, E), \mathcal{G}' = (V, E')$ with $V = \{1, \ldots, n\}$

find $\varphi \in S_n$ s.t. $\{u, v\} \in E \leftrightarrow \{\varphi(u), \varphi(v)\} \in E'$

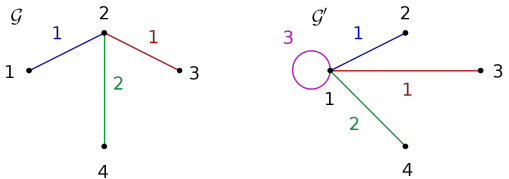


Reduction from PEP to GI

Graph Isomorphism (GI) Problem

Given $\mathcal{G} = (V, E), \mathcal{G}' = (V, E')$ with $V = \{1, \ldots, n\}$

find $\varphi \in S_n$ s.t. $\{u, v\} \in E \leftrightarrow \{\varphi(u), \varphi(v)\} \in E'$



Babai's algorithm: GI is quasi-polynomial time! cost in $\mathcal{O}(2^{\log(n)^c})$

"Graph isomorphism in quasipolynomial time" L. Babai, 2016

Reduction from PEP to GI

Reduction from PEP to GI

Definition The adjacency matrix A of a weighted graph $\mathcal{G}$ is

$$A_{i,j} = \begin{cases} w(i,j) & \text{if } \{i,j\} \in E \\ 0 & \text{else} \end{cases}$$

Reduction from PEP to GI

Definition The adjacency matrix A of a weighted graph $\mathcal{G}$ is

$$A_{i,j} = \begin{cases} w(i,j) & \text{if } \{i,j\} \in E \\ 0 & \text{else} \end{cases}$$

Proposition Two graphs $\mathcal{G}, \mathcal{G}'$ are isomorphic iff

$\exists P$ permutation matrix s.t. $P^\top A P = A'$

Reduction from PEP to GI

Definition The adjacency matrix A of a weighted graph $\mathcal{G}$ is

$$A_{i,j} = \begin{cases} w(i,j) & \text{if } \{i,j\} \in E \\ 0 & \text{else} \end{cases}$$

Proposition Two graphs $\mathcal{G}, \mathcal{G}'$ are isomorphic iff
$\exists P$ permutation matrix s.t. $P^\top A P = A'$

Theorem If $\mathcal{H}(\mathcal{C}) = \{0\}$ then PEP can be reduced to GI $\rightarrow$ PEP is easier than GI

"Permutation code equivalence is not harder than GI" M. Bardet, A. Otmani, M. Saeed-Taha, 2019

Reduction from GI to PEP

Reduction from GI to PEP

Definition The incidence matrix B of a graph $\mathcal{G}$ with $|V| = v, |E| = e$ is

$$B_{i,j} = \begin{cases} 1 & \text{if } i = \{\ell, j\} \in E \\ 0 & \text{else} \end{cases} \qquad B \in \mathbb{F}_2^{e \times v}$$

Reduction from GI to PEP

Definition The incidence matrix B of a graph $\mathcal{G}$ with $|V| = v, |E| = e$ is

$$B_{i,j} = \begin{cases} 1 & \text{if } i = \{\ell, j\} \in E \\ 0 & \text{else} \end{cases} \qquad B \in \mathbb{F}_2^{e \times v}$$

Proposition Two graphs $\mathcal{G}, \mathcal{G}'$ are isomorphic iff
$\exists Q \in S_e, P \in S_v$, such that $QBP = B'$

Reduction from GI to PEP

Definition The incidence matrix B of a graph $\mathcal{G}$ with $|V| = v, |E| = e$ is

$$B_{i,j} = \begin{cases} 1 & \text{if } i = \{\ell, j\} \in E \\ 0 & \text{else} \end{cases} \qquad B \in \mathbb{F}_2^{e \times v}$$

Proposition Two graphs $\mathcal{G}, \mathcal{G}'$ are isomorphic iff
$\exists Q \in S_e, P \in S_v$, such that $QBP = B'$

Theorem We can reduce GI to PEP

"Is code equivalence easy to decide?" E. Petrank, M. Roth, 2002

If LEP → PEP and PEP → GI then LEP → GI

If LEP → PEP and PEP → GI then LEP → GI NO

If LEP → PEP and PEP → GI then LEP → GI NO

Under the rug

We can only reduce PEP to GI if $\mathcal{H}(\mathcal{C}) = \{0\}$ is $\mathcal{H}(\lambda \otimes \mathcal{C}) = \{0\}$?

Code Equivalence - Connections

If LEP → PEP and PEP → GI then LEP → GI NO

Under the rug

We can only reduce PEP to GI if $\mathcal{H}(\mathcal{C}) = \{0\}$ is $\mathcal{H}(\lambda \otimes \mathcal{C}) = \{0\}$?

Exercise Show that $\sum_{\alpha \in \mathbb{F}_q^\star} \alpha^\ell = \begin{cases} 0 & \text{if } (q-1) \nmid \ell \\ -1 & \text{if } (q-1) \mid \ell \end{cases}$

If LEP → PEP and PEP → GI then LEP → GI NO

Under the rug

We can only reduce PEP to GI if $\mathcal{H}(\mathcal{C}) = \{0\}$ is $\mathcal{H}(\lambda \otimes \mathcal{C}) = \{0\}$?

Exercise Show that $\sum_{\alpha \in \mathbb{F}_q^\star} \alpha^\ell = \begin{cases} 0 & \text{if } (q-1) \nmid \ell \\ -1 & \text{if } (q-1) \mid \ell \end{cases}$

Proposition If $q \geq 4$, then $\lambda \otimes \mathcal{C}$ is self-orthogonal

A bit of hope

$q = p^{2m}$

A bit of hope $q = p^{2m}$

Definition

- Let $x, y \in \mathbb{F}_q^n$. The Hermitian inner product is

$$\langle x, y \rangle_H = \sum_{i=1}^{n} x_i y_i^{p^m}$$

A bit of hope $q = p^{2m}$

Definition

- Let $x, y \in \mathbb{F}_q^n$. The Hermitian inner product is
$$\langle x, y \rangle_H = \sum_{i=1}^{n} x_i y_i^{p^m}$$
- Let $\mathcal{C}$ be a $[n, k]_q$ linear code. The Hermitian dual is
$$\mathcal{C}^\star = \{x \in \mathbb{F}_q^n \mid \langle x, y \rangle_H = 0 \ \forall \ y \in \mathcal{C}\}$$

A bit of hope $q = p^{2m}$

Definition

- Let $x, y \in \mathbb{F}_q^n$. The Hermitian inner product is
$$\langle x, y \rangle_H = \sum_{i=1}^{n} x_i y_i^{p^m}$$
- Let $\mathcal{C}$ be a $[n, k]_q$ linear code. The Hermitian dual is
$$\mathcal{C}^\star = \{x \in \mathbb{F}_q^n \mid \langle x, y \rangle_H = 0 \ \forall \ y \in \mathcal{C}\}$$
- A Hermitian parity-check matrix $H^\star$ is s.t. $\langle H^\star \rangle = \mathcal{C}^\star$

"How easy is code equivalence over $\mathbb{F}_q$?" N. Sendrier, D. Simos, 2013

A bit of hope

$q = p^{2m}$

Let $\mathcal{C} = \langle G \rangle = \ker(H^\top)$

A bit of hope $q = p^{2m}$ Let $\mathcal{C} = \langle G \rangle = \ker(H^\top)$

Exercises

- Show that $H^\star (G^{p^m})^\top = 0$. That is $\mathcal{C}^\star = \ker((G^{p^m})^\top)$

A bit of hope $q = p^{2m}$ Let $\mathcal{C} = \langle G \rangle = \ker(H^\top)$

Exercises

- Show that $H^\star (G^{p^m})^\top = 0$. That is $\mathcal{C}^\star = \ker((G^{p^m})^\top)$
- Show that $H^\star = H^{p^m}$ is a Hermitian parity-check matrix

A bit of hope

$q = p^{2m}$

Let $\mathcal{C} = \langle G \rangle = \ker(H^\top)$

Exercises

- Show that $H^\star (G^{p^m})^\top = 0$. That is $\mathcal{C}^\star = \ker((G^{p^m})^\top)$
- Show that $H^\star = H^{p^m}$ is a Hermitian parity-check matrix
- Show that $(\mathcal{C}^\star)^\star = \mathcal{C}$

A bit of hope $q = p^{2m}$ Let $\mathcal{C} = \langle G \rangle = \ker(H^\top)$

Exercises

- Show that $H^\star(G^{p^m})^\top = 0$. That is $\mathcal{C}^\star = \ker((G^{p^m})^\top)$
- Show that $H^\star = H^{p^m}$ is a Hermitian parity-check matrix
- Show that $(\mathcal{C}^\star)^\star = \mathcal{C}$
- Show that $\mathcal{H}^\star(\mathcal{C}) = \ker\left(\begin{pmatrix} G^{p^m} \\ H \end{pmatrix}^\top\right)$

A bit of hope $q = p^{2m}$ Let $\mathcal{C} = \langle G \rangle = \ker(H^\top)$

Exercises

- Show that $H^\star(G^{p^m})^\top = 0$. That is $\mathcal{C}^\star = \ker((G^{p^m})^\top)$
- Show that $H^\star = H^{p^m}$ is a Hermitian parity-check matrix
- Show that $(\mathcal{C}^\star)^\star = \mathcal{C}$
- Show that $\mathcal{H}^\star(\mathcal{C}) = \ker\left(\begin{pmatrix} G^{p^m} \\ H \end{pmatrix}^\top\right)$
- Let $\mathcal{C} \subset \mathbb{F}_q^n$ be linearly equivalent to $\mathcal{C}'$.
  Show that $\mathcal{C}^\star$ is linearly equivalent to $(\mathcal{C}')^\star$

A bit of hope $q = p^{2m}$ Let $\mathcal{C} = \langle G \rangle = \ker(H^\top)$

Exercises

- Show that $H^\star(G^{p^m})^\top = 0$. That is $\mathcal{C}^\star = \ker((G^{p^m})^\top)$
- Show that $H^\star = H^{p^m}$ is a Hermitian parity-check matrix
- Show that $(\mathcal{C}^\star)^\star = \mathcal{C}$
- Show that $\mathcal{H}^\star(\mathcal{C}) = \ker\left(\begin{pmatrix} G^{p^m} \\ H \end{pmatrix}^\top\right)$
- Let $\mathcal{C} \subset \mathbb{F}_q^n$ be linearly equivalent to $\mathcal{C}'$.
  Show that $\mathcal{C}^\star$ is linearly equivalent to $(\mathcal{C}')^\star$
- Let $\mathcal{C} \subset \mathbb{F}_q^n$ be permutation equivalent to $\mathcal{C}'$.
  Show that $\mathcal{H}^\star(\mathcal{C})$ is permutation equivalent to $\mathcal{H}^\star(\mathcal{C}')$

Two New results

How many pairs $(c, \varphi(c))$ needed to recover φ?

Two New results

How many pairs $(c, \varphi(c))$ needed to recover φ?

Rouché-Capelli Test

Let $A \in \mathbb{F}_q^{k \times n}$ of rank r and $b \in \mathbb{F}_q^k$

The system $Ax^\top = b^\top$ has a solution iff $\mathrm{rk}([A \mid b]) = r$

→ only 2! (with some heuristics)

"Two Is All It Takes" A. Budroni, A. Esser, E. Franch, A. Natale, 2025

Two New results

How many pairs $(c, \varphi(c))$ needed to recover φ?

Rouché-Capelli Test

Let $A \in \mathbb{F}_q^{k \times n}$ of rank r and $b \in \mathbb{F}_q^k$

The system $Ax^\top = b^\top$ has a solution iff $\mathrm{rk}([A \mid b]) = r$

→ only 2! (with some heuristics)

"Two Is All It Takes" A. Budroni, A. Esser, E. Franch, A. Natale, 2025

How many pairs $(\mathcal{C}, \varphi(\mathcal{C}))$ needed to recover φ?

→ only 2!

"Don't use it twice!" A. Budroni, J. Chi-Domínguez, D. D'Alconzo, A. Di Scala, M. Kulkarni, 2024

Definition

- Let $x, y \in \mathbb{F}_q^n$. The Schur product is $x * y = (x_1 y_1, \ldots, x_n y_n)$

Definition

- Let $x, y \in \mathbb{F}_q^n$. The Schur product is $x * y = (x_1 y_1, \ldots, x_n y_n)$
- Let $\mathcal{C}_i$ be $[n, k_i]_q$ linear codes. The Schur product is

$$\mathcal{C}_1 * \mathcal{C}_2 = \langle \{ c_1 * c_2 \mid c_1 \in \mathcal{C}_1, c_2 \in \mathcal{C}_2 \} \rangle$$

Definition

- Let $x, y \in \mathbb{F}_q^n$. The Schur product is $x * y = (x_1 y_1, \ldots, x_n y_n)$
- Let $\mathcal{C}_i$ be $[n, k_i]_q$ linear codes. The Schur product is
  $\mathcal{C}_1 * \mathcal{C}_2 = \langle \{c_1 * c_2 \mid c_1 \in \mathcal{C}_1, c_2 \in \mathcal{C}_2\} \rangle$
- Let $\mathcal{C}$ be an $[n, k]_q$ linear code. The square code is $\mathcal{C}^{(2)} = \mathcal{C} * \mathcal{C}$

Definition

- Let $x, y \in \mathbb{F}_q^n$. The Schur product is $x * y = (x_1 y_1, \ldots, x_n y_n)$
- Let $\mathcal{C}_i$ be $[n, k_i]_q$ linear codes. The Schur product is
  $\mathcal{C}_1 * \mathcal{C}_2 = \langle \{c_1 * c_2 \mid c_1 \in \mathcal{C}_1, c_2 \in \mathcal{C}_2\} \rangle$
- Let $\mathcal{C}$ be an $[n, k]_q$ linear code. The square code is $\mathcal{C}^{(2)} = \mathcal{C} * \mathcal{C}$

Exercise

$\langle G \rangle = \mathcal{C}$. Show that $\langle G^{(2)} \rangle = \mathcal{C}^{(2)}$

where $G^{(2)} = \begin{pmatrix} g_1 * g_1 \\ \vdots \\ g_1 * g_k \\ \vdots \\ g_k * g_k \end{pmatrix} \in \mathbb{F}_q^{\binom{k+1}{2} \times n}$

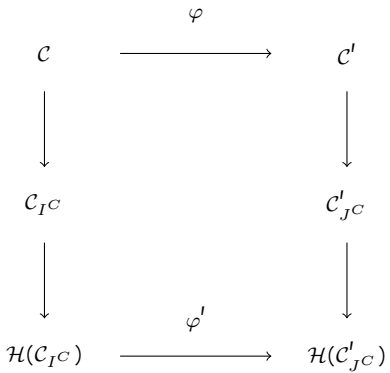
Theorem Let $\mathcal{C}$ be a $[n,k]_q$ linear code. Then $\dim(\mathcal{C}^{(2)}) = \min\left\{n, \binom{k+1}{2}\right\}$

Theorem Let $\mathcal{C}$ be a $[n,k]_q$ linear code. Then $\dim(\mathcal{C}^{(2)}) = \min\left\{n, \binom{k+1}{2}\right\}$

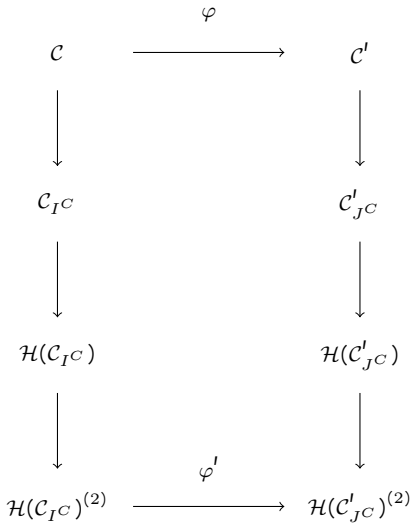
Exercises

- Let $\mathcal{C}, \mathcal{C}'$ be $[n,k]_q$ linear codes and $\varphi = (D,P) \in (\mathbb{F}_q^\star)^n \rtimes S_n$ s.t. $\varphi(\mathcal{C}) = \mathcal{C}'$. Then $\varphi' = (D^2, P) \in (\mathbb{F}_q^\star)^n \rtimes S_n$ is s.t. $\varphi'(\mathcal{C}^{(2)}) = \mathcal{C}'^{(2)}$
- Show that $\mathcal{H}(\mathcal{C}^{(2)}) \neq \mathcal{H}(\mathcal{C})^{(2)}$

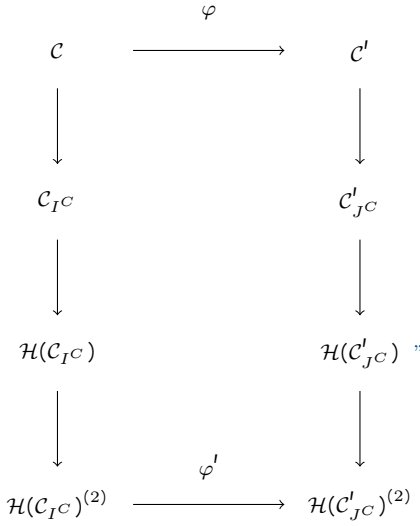
Recall SSA


φ
$\mathcal{C}$
$\mathcal{C}'$
$\mathcal{C}_{I^C}$
$\mathcal{C}'_{J^C}$
φ'
$\mathcal{H}(\mathcal{C}_{I^C})$
$\mathcal{H}(\mathcal{C}'_{J^C})$

Recall SSA


φ
$\mathcal{C}$
$\mathcal{C}'$
$\mathcal{C}_{I^C}$
$\mathcal{C}'_{J^C}$
$\mathcal{H}(\mathcal{C}_{I^C})$
$\mathcal{H}(\mathcal{C}'_{J^C})$
φ'
$\mathcal{H}(\mathcal{C}_{I^C})^{(2)}$
$\mathcal{H}(\mathcal{C}'_{J^C})^{(2)}$

Recall SSA



"Using the Schur Product to Solve the Code Equivalence Problem"
M. Battagliola, R. Mora, Rocco, P. Santini, 2025

recent attack on

"Hollow LWE" M. Albrecht, B. Benčina, R. Lai, 2025

Definition Let $\mathcal{C}$ be an $[n,k]_q$ linear code. The ℓ power code is

$$\mathcal{C}^{(\ell)} = \underbrace{\mathcal{C} \star \cdots \star \mathcal{C}}_{\ell}$$

Theorem Let $\mathcal{C}$ be an $[n,k]_q$ linear code. If $\ell < q$, then

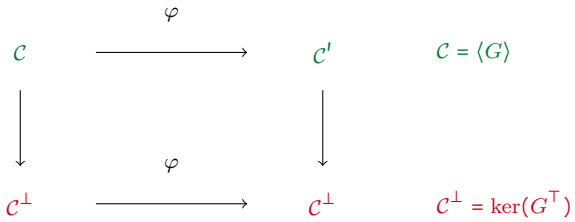
$$\dim(\mathcal{C}^{(\ell)}) = \min\left\{\binom{k+\ell-1}{\ell}, n\right\}$$

Exercises

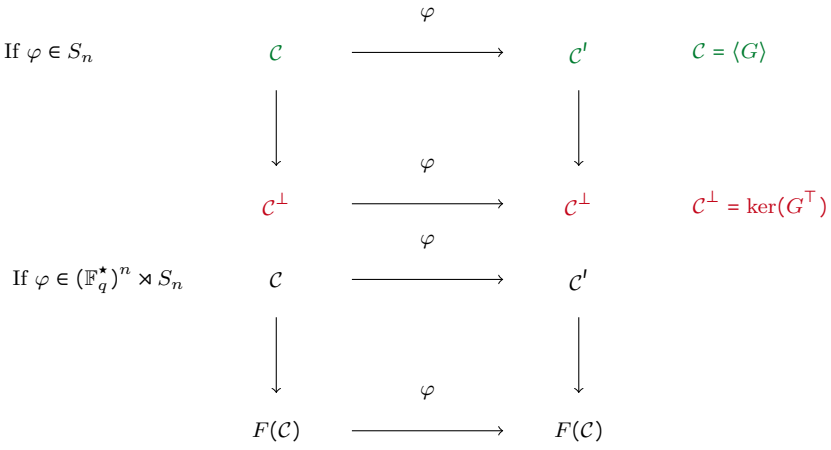
- Show that $(\lambda \otimes \mathcal{C})^{(2)} \neq \lambda \otimes \mathcal{C}^{(2)}$
- Show that $(\lambda \otimes G)^{(\ell)} = \lambda^{\ell} \otimes G^{(\ell)}$

Why $A(G, G^\top) = G^\top (GG^\top)^{-1} G$?

If $\varphi \in S_n$



Why $A(G, G^\top) = G^\top (GG^\top)^{-1} G$?



Summary

Summary

- Differentiate between LEP and PEP

Summary

- Differentiate between LEP and PEP
- If $\mathcal{C}$ is linearly equivalent to $\mathcal{C}'$ then $\mathcal{C}^{\perp}$ is linearly equivalent to $\mathcal{C}'^{\perp}$

Summary

- Differentiate between LEP and PEP
- If $\mathcal{C}$ is linearly equivalent to $\mathcal{C}'$ then $\mathcal{C}^{\perp}$ is linearly equivalent to $\mathcal{C}'^{\perp}$
- Only for PEP is the dual connected through the same permutation

Summary

- Differentiate between LEP and PEP
- If $\mathcal{C}$ is linearly equivalent to $\mathcal{C}'$ then $\mathcal{C}^{\perp}$ is linearly equivalent to $\mathcal{C}'^{\perp}$
- Only for PEP is the dual connected through the same permutation
- Several invariants: weight enumerators, generalized weights

Summary

- Differentiate between LEP and PEP
- If $\mathcal{C}$ is linearly equivalent to $\mathcal{C}'$ then $\mathcal{C}^{\perp}$ is linearly equivalent to $\mathcal{C}'^{\perp}$
- Only for PEP is the dual connected through the same permutation
- Several invariants: weight enumerators, generalized weights
- Hulls of random codes are w.h.p. trivial

Summary

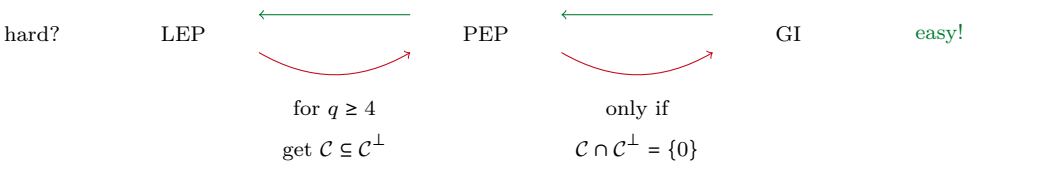
- Differentiate between LEP and PEP
- If $\mathcal{C}$ is linearly equivalent to $\mathcal{C}'$ then $\mathcal{C}^{\perp}$ is linearly equivalent to $\mathcal{C}'^{\perp}$
- Only for PEP is the dual connected through the same permutation
- Several invariants: weight enumerators, generalized weights
- Hulls of random codes are w.h.p. trivial
- LEP, PEP $\notin NP$-hard, they are in co-AM $\cap$ NP

Summary

- Differentiate between LEP and PEP
- If $\mathcal{C}$ is linearly equivalent to $\mathcal{C}'$ then $\mathcal{C}^{\perp}$ is linearly equivalent to $\mathcal{C}'^{\perp}$
- Only for PEP is the dual connected through the same permutation
- Several invariants: weight enumerators, generalized weights
- Hulls of random codes are w.h.p. trivial
- LEP, PEP $\notin NP$-hard, they are in co-AM $\cap$ NP
- Several solvers use invariants, but all exponential cost

Summary

- Differentiate between LEP and PEP
- If $\mathcal{C}$ is linearly equivalent to $\mathcal{C}'$ then $\mathcal{C}^{\perp}$ is linearly equivalent to $\mathcal{C}'^{\perp}$
- Only for PEP is the dual connected through the same permutation
- Several invariants: weight enumerators, generalized weights
- Hulls of random codes are w.h.p. trivial
- LEP, PEP $\notin NP$-hard, they are in co-AM $\cap$ NP
- Several solvers use invariants, but all exponential cost
- There are several reductions:



Other metrics?

Code Equivalence - Bonus Round

Other metrics? Rank metric

Other metrics? Rank metric

- "Matrix code" or $\mathbb{F}_q$-linear code $(\mathbb{F}_q^{m\times n}, \mathrm{wt}_R)$

  $X \in \mathbb{F}_q^{m\times n}$ then $\mathrm{wt}_R(X) = \mathrm{rk}(X)$

Other metrics? Rank metric

- "Matrix code" or $\mathbb{F}_q$-linear code $(\mathbb{F}_q^{m\times n}, \mathrm{wt}_R)$

  $X \in \mathbb{F}_q^{m\times n}$ then $\mathrm{wt}_R(X) = \mathrm{rk}(X)$

  linear isometries:

Other metrics? Rank metric

- "Matrix code" or $\mathbb{F}_q$-linear code $(\mathbb{F}_q^{m\times n}, \mathrm{wt}_R)$

  $X \in \mathbb{F}_q^{m\times n}$ then $\mathrm{wt}_R(X) = \mathrm{rk}(X)$

  linear isometries: $\varphi = (A, B) \in \mathrm{GL}_q(m) \times \mathrm{GL}_q(n)$

Code Equivalence - Bonus Round

Other metrics? Rank metric

- "Matrix code" or $\mathbb{F}_q$-linear code $(\mathbb{F}_q^{m\times n}, \mathrm{wt}_R)$

  $X \in \mathbb{F}_q^{m\times n}$ then $\mathrm{wt}_R(X) = \mathrm{rk}(X)$

  linear isometries: $\varphi = (A, B) \in \mathrm{GL}_q(m) \times \mathrm{GL}_q(n)$

→ no idea

Code Equivalence - Bonus Round

Other metrics? Rank metric

- "Matrix code" or $\mathbb{F}_q$-linear code $(\mathbb{F}_q^{m\times n}, \mathrm{wt}_R)$

  $X \in \mathbb{F}_q^{m\times n}$ then $\mathrm{wt}_R(X) = \mathrm{rk}(X)$

  linear isometries: $\varphi = (A, B) \in \mathrm{GL}_q(m) \times \mathrm{GL}_q(n)$

  → no idea

- "Vector code" or $\mathbb{F}_{q^m}$-linear code $(\mathbb{F}_{q^m}^n, \mathrm{wt}_R)$

Other metrics? Rank metric

- "Matrix code" or $\mathbb{F}_q$-linear code $(\mathbb{F}_q^{m\times n}, \mathrm{wt}_R)$

  $X \in \mathbb{F}_q^{m\times n}$ then $\mathrm{wt}_R(X) = \mathrm{rk}(X)$

  linear isometries: $\varphi = (A, B) \in \mathrm{GL}_q(m) \times \mathrm{GL}_q(n)$

→ no idea

- "Vector code" or $\mathbb{F}_{q^m}$-linear code $(\mathbb{F}_{q^m}^n, \mathrm{wt}_R)$

  $x \in \mathbb{F}_{q^m}^n$ then $\mathrm{wt}_R(x) = \dim_{\mathbb{F}_q}(\langle x_1, \ldots, x_n \rangle_{\mathbb{F}_q}$

Other metrics? Rank metric

- "Matrix code" or $\mathbb{F}_q$-linear code $(\mathbb{F}_q^{m\times n}, \mathrm{wt}_R)$

  $X \in \mathbb{F}_q^{m\times n}$ then $\mathrm{wt}_R(X) = \mathrm{rk}(X)$

  linear isometries: $\varphi = (A, B) \in \mathrm{GL}_q(m) \times \mathrm{GL}_q(n)$

→ no idea

- "Vector code" or $\mathbb{F}_{q^m}$-linear code $(\mathbb{F}_{q^m}^n, \mathrm{wt}_R)$

  $x \in \mathbb{F}_{q^m}^n$ then $\mathrm{wt}_R(x) = \dim_{\mathbb{F}_q}(\langle x_1, \ldots, x_n\rangle_{\mathbb{F}_q}$

  linear isometries:

Code Equivalence - Bonus Round

Other metrics? Rank metric

- "Matrix code" or $\mathbb{F}_q$-linear code $(\mathbb{F}_q^{m\times n}, \mathrm{wt}_R)$

  $X \in \mathbb{F}_q^{m\times n}$ then $\mathrm{wt}_R(X) = \mathrm{rk}(X)$

  linear isometries: $\varphi = (A, B) \in \mathrm{GL}_q(m) \times \mathrm{GL}_q(n)$

→ no idea

- "Vector code" or $\mathbb{F}_{q^m}$-linear code $(\mathbb{F}_{q^m}^n, \mathrm{wt}_R)$

  $x \in \mathbb{F}_{q^m}^n$ then $\mathrm{wt}_R(x) = \dim_{\mathbb{F}_q}(\langle x_1, \ldots, x_n\rangle_{\mathbb{F}_q}$

  linear isometries: $\varphi = (B) \in \mathrm{GL}_q(n)$

→ easy!

"On the hardness of code equivalence problems in rank metric" A. Couvreur, T. Debris-Alazard, P. Gaborit, 2020

Other metrics?

Other metrics? Lee metric

Other metrics? Lee metric $(\mathbb{Z}/p^s\mathbb{Z}^n, \mathrm{wt}_L)$

$x \in \mathbb{Z}/p^s\mathbb{Z}^n$ then $\mathrm{wt}_L(x) = \sum_{i=1}^{n} \min\{x_i, |p^s - x_i|\}$

Other metrics? Lee metric $(\mathbb{Z}/p^s\mathbb{Z}^n, \mathrm{wt}_L)$

$x \in \mathbb{Z}/p^s\mathbb{Z}^n$ then $\mathrm{wt}_L(x) = \sum_{i=1}^{n} \min\{x_i, |p^s - x_i|\}$

linear isometries:

Code Equivalence - Bonus Round

Other metrics? Lee metric $(\mathbb{Z}/p^s\mathbb{Z}^n, \mathrm{wt}_L)$

$x \in \mathbb{Z}/p^s\mathbb{Z}^n$ then $\mathrm{wt}_L(x) = \sum_{i=1}^n \min\{x_i, |p^s - x_i|\}$

linear isometries: $\varphi = (D, P) \in \{\pm 1\}^n \rtimes S_n$

Code Equivalence - Bonus Round

Other metrics? Lee metric $(\mathbb{Z}/p^s\mathbb{Z}^n, \mathrm{wt}_L)$

$x \in \mathbb{Z}/p^s\mathbb{Z}^n$ then $\mathrm{wt}_L(x) = \sum_{i=1}^n \min\{x_i, |p^s - x_i|\}$

linear isometries: $\varphi = (D, P) \in \{\pm 1\}^n \rtimes S_n$

→ like PEP

Code Equivalence - Bonus Round

Other metrics? Lee metric $(\mathbb{Z}/p^s\mathbb{Z}^n, \mathrm{wt}_L)$

$x \in \mathbb{Z}/p^s\mathbb{Z}^n$ then $\mathrm{wt}_L(x) = \sum_{i=1}^n \min\{x_i, |p^s - x_i|\}$

linear isometries: $\varphi = (D, P) \in \{\pm 1\}^n \rtimes S_n$

→ like PEP

Homogeneous metric $(\mathbb{Z}/p^s\mathbb{Z}^n, \mathrm{wt}_{\mathrm{Hom}})$

Code Equivalence - Bonus Round

Other metrics? Lee metric $(\mathbb{Z}/p^s\mathbb{Z}^n, \mathrm{wt}_L)$

$x \in \mathbb{Z}/p^s\mathbb{Z}^n$ then $\mathrm{wt}_L(x) = \sum_{i=1}^n \min\{x_i, |p^s - x_i|\}$

linear isometries: $\varphi = (D, P) \in \{\pm 1\}^n \rtimes S_n$

→ like PEP

Homogeneous metric $(\mathbb{Z}/p^s\mathbb{Z}^n, \mathrm{wt}_{\mathrm{Hom}})$

$$x \in \mathbb{Z}/p^s\mathbb{Z}^n \text{ then } \mathrm{wt}_{\mathrm{Hom}}(x) = \sum_{i=1}^n \begin{cases} 0 & \text{if } x_i = 0, \\ 1 & \text{if } x_i \notin \langle p^{s-1} \rangle, \\ p/(p-1) & \text{if } x_i \in \langle p^{s-1} \rangle \setminus \{0\} \end{cases}$$

Other metrics? Lee metric $(\mathbb{Z}/p^s\mathbb{Z}^n, \mathrm{wt}_L)$

$x \in \mathbb{Z}/p^s\mathbb{Z}^n$ then $\mathrm{wt}_L(x) = \sum_{i=1}^n \min\{x_i, |p^s - x_i|\}$

linear isometries: $\varphi = (D, P) \in \{\pm 1\}^n \rtimes S_n$

→ like PEP

Homogeneous metric $(\mathbb{Z}/p^s\mathbb{Z}^n, \mathrm{wt}_{\mathrm{Hom}})$

$$x \in \mathbb{Z}/p^s\mathbb{Z}^n \text{ then } \mathrm{wt}_{\mathrm{Hom}}(x) = \sum_{i=1}^n \begin{cases} 0 & \text{if } x_i = 0, \\ 1 & \text{if } x_i \notin \langle p^{s-1} \rangle, \\ p/(p-1) & \text{if } x_i \in \langle p^{s-1} \rangle \setminus \{0\} \end{cases}$$

linear isometries:

Code Equivalence - Bonus Round

Other metrics? Lee metric $(\mathbb{Z}/p^s\mathbb{Z}^n, \mathrm{wt}_L)$

$x \in \mathbb{Z}/p^s\mathbb{Z}^n$ then $\mathrm{wt}_L(x) = \sum_{i=1}^n \min\{x_i, |p^s - x_i|\}$

linear isometries: $\varphi = (D, P) \in \{\pm 1\}^n \rtimes S_n$

→ like PEP

Homogeneous metric $(\mathbb{Z}/p^s\mathbb{Z}^n, \mathrm{wt}_{\mathrm{Hom}})$

$$x \in \mathbb{Z}/p^s\mathbb{Z}^n \text{ then } \mathrm{wt}_{\mathrm{Hom}}(x) = \sum_{i=1}^n \begin{cases} 0 & \text{if } x_i = 0, \\ 1 & \text{if } x_i \notin \langle p^{s-1} \rangle, \\ p/(p-1) & \text{if } x_i \in \langle p^{s-1} \rangle \setminus \{0\} \end{cases}$$

linear isometries: $\varphi = (D, P) \in (\mathbb{Z}/p^s\mathbb{Z}^\times)^n \rtimes S_n$

→ easier than Hamming

Every code is linearly equivalent to a code with trivial hull

"Linear codes over $\mathbb{F}_q$ are equivalent to LCD codes for $q > 3$"

C. Carlet, S. Mesnager, C. Tang,Y. Qi, R. Pellikaan, 2018

Every code is linearly equivalent to a code with trivial hull

"Linear codes over $\mathbb{F}_q$ are equivalent to LCD codes for $q > 3$"

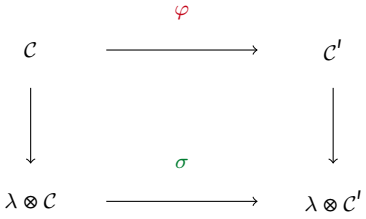
C. Carlet, S. Mesnager, C. Tang,Y. Qi, R. Pellikaan, 2018

$$\mathcal{C} \xrightarrow{\varphi} \mathcal{C}'$$

Every code is linearly equivalent to a code with trivial hull

"Linear codes over $\mathbb{F}_q$ are equivalent to LCD codes for $q > 3$"

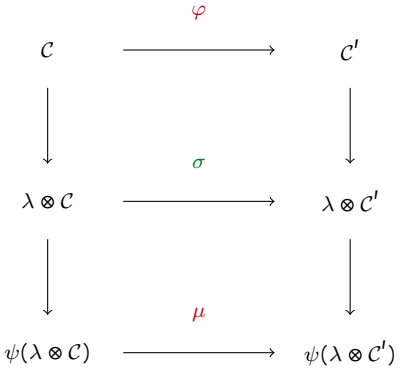
C. Carlet, S. Mesnager, C. Tang,Y. Qi, R. Pellikaan, 2018



Every code is linearly equivalent to a code with trivial hull

"Linear codes over $\mathbb{F}_q$ are equivalent to LCD codes for $q > 3$"

C. Carlet, S. Mesnager, C. Tang,Y. Qi, R. Pellikaan, 2018



Computational → decisional

"A search-to-decision reduction for the permutation code equivalence problem" J.-F. Biasse, G. Micheli, 2023

Workshop on the Mathematics of Post-Quantum Cryptography

Munich, September 7–11, 2026

https://mathpqc26.cry.cit.tum.de/

Code Equivalence - Exercises

$\mathcal{C} = \langle G \rangle = \ker(H^\top)$ a $[n,k]_q$ linear code

1. Let $H^\star \in \mathbb{F}_q^{(n-k)\times n}$ be a Hermitian parity-check matrix of $\mathcal{C}$.
   Show that $H^\star (G^{p^m})^\top = 0$. That is $\mathcal{C}^\star = \ker((G^{p^m})^\top)$.
2. Show that $H^\star = H^{p^m}$ is a Hermitian parity-check matrix.
3. Show that $(\mathcal{C}^\star)^\star = \mathcal{C}$.
4. Show that $\mathcal{H}^\star(\mathcal{C}) = \ker\left(\begin{pmatrix} G^{p^m} \\ H \end{pmatrix}^\top\right)$.
5. Let $\mathcal{C}$ be linearly equivalent to $\mathcal{C}'$. Show that $\mathcal{C}^\star$ is linearly equivalent to $(\mathcal{C}')^\star$.
6. Show that if $\varphi \in S_n$ is such that $\varphi(\mathcal{C}) = \mathcal{C}'$,
   then $\mathcal{H}^\star(\mathcal{C})$ is permutation equivalent to $\mathcal{H}^\star(\mathcal{C}')$.
7. Show that $A^\star$ is independent on the choice of G.
   Show that if $G(G^{p^m})^\top$ has full rank, then $\dim(\mathcal{H}^\star(\mathcal{C})) = 0$.

$\mathcal{C} = \langle G \rangle = \ker(H^\top)$ a $[n,k]_q$ linear code

1. Show that $\sum_{\alpha \in \mathbb{F}_q^\star} \alpha^\ell = \begin{cases} 0 & \text{if } (q-1) \nmid \ell, \\ -1 & \text{if } (q-1) \mid \ell. \end{cases}$
2. Show that $\mathcal{C}^{(2)}$ is generated by $G^{(2)}$.
3. Show that if $\varphi = (D, P) \in (\mathbb{F}_q^\star)^n \rtimes S_n$ is such that $\varphi(\mathcal{C}) = \mathcal{C}'$, then $\varphi' = (D^2, P) \in (\mathbb{F}_q^\star)^n \rtimes S_n$ is such that $\varphi'(\mathcal{C}^{(2)}) = \mathcal{C}'^{(2)}$.
4. Show that $\mathcal{H}(\mathcal{C})^{(2)} \neq \mathcal{H}(\mathcal{C}^{(2)})$.
5. Reduce the following LEP instance to GI using the square code:

$$G = \begin{pmatrix} 1 & 0 & 2 & 1 \\ 0 & 1 & 3 & 0 \end{pmatrix} \in \mathbb{F}_5^{2\times 4} \text{ and } G' = \begin{pmatrix} 4 & 1 & 0 & 2 \\ 0 & 4 & 2 & 0 \end{pmatrix}.$$

6. Show that $(\lambda \otimes \mathcal{C})^{(2)} \neq \lambda \otimes \mathcal{C}^{(2)}$.
7. Show that $(\lambda \otimes G)^{(\ell)} = \lambda^\ell \otimes G^{(\ell)}$.

Code Equivalence - Solutions

Solutions

Slides