

On the Hardness of Finding Permutation Automorphisms of Linear Codes

Paolo Santini

Dipartimento di Ingegneria dell'Informazione (DII), Università Politecnica delle Marche
`p.santini@staff.univpm.it`

Joint work with:

Michele Battagliola, Giuseppe d'Alconzo, Laura Mattiuz, Federica Zanetti

Outline



Why looking at automorphisms?



Code automorphisms: a computational point of view



From isomorphisms to automorphisms, and viceversa



Graphs, codes and automorphisms

Why looking at automorphisms?

Isomorphisms of codes

Let S_n : group of length- n permutations

Permutation Equivalent Codes

Let $\pi \in S_n$ and $\mathcal{C} \subseteq \mathbb{F}_q^n$:

$$\mathcal{C}' = \pi(\mathcal{C}) = \{\pi(\mathbf{c}) \mid \mathbf{c} \in \mathcal{C}\}$$

The codes $\mathcal{C}$ and $\mathcal{C}'$ are said to be **equivalent** (or **isomorphic**)

Isomorphisms of codes

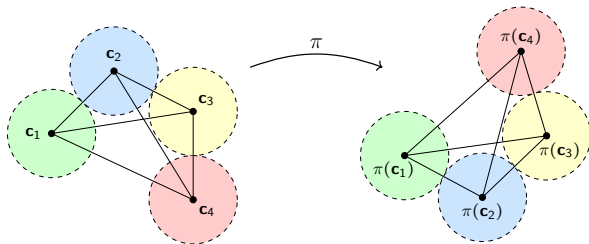
Let S_n : group of length- n permutations

Permutation Equivalent Codes

Let $\pi \in S_n$ and $\mathcal{C} \subseteq \mathbb{F}_q^n$:

$$\mathcal{C}' = \pi(\mathcal{C}) = \{\pi(\mathbf{c}) \mid \mathbf{c} \in \mathcal{C}\}$$

The codes $\mathcal{C}$ and $\mathcal{C}'$ are said to be **equivalent** (or **isomorphic**)



Isomorphisms of codes

Let S_n : group of length- n permutations

Permutation Equivalent Codes

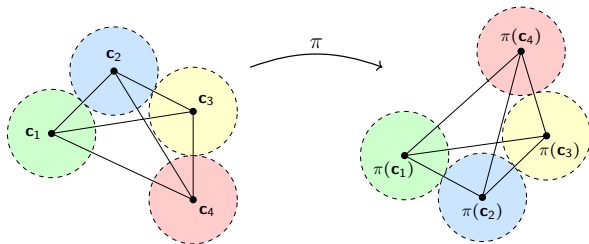
Let $\pi \in S_n$ and $\mathcal{C} \subseteq \mathbb{F}_q^n$:

$$\mathcal{C}' = \pi(\mathcal{C}) = \{\pi(\mathbf{c}) \mid \mathbf{c} \in \mathcal{C}\}$$

The codes $\mathcal{C}$ and $\mathcal{C}'$ are said to be **equivalent** (or **isomorphic**)

If $\mathbf{A}, \mathbf{B} \in \mathbb{F}_q^{k \times n}$ are generator matrices for $\mathcal{C}, \mathcal{C}'$:

$$\exists \mathbf{S} \in GL_k(\mathbb{F}_q), \mathbf{P} \in S_n : \quad \mathbf{B} = \mathbf{SAP}$$



Automorphisms of codes

Definition

Let $\pi \in S_n$ and $\mathcal{C} \subseteq \mathbb{F}_q^n$:

$$\pi \text{ is an automorphism for } \mathcal{C} \iff \pi(\mathcal{C}) = \mathcal{C}$$

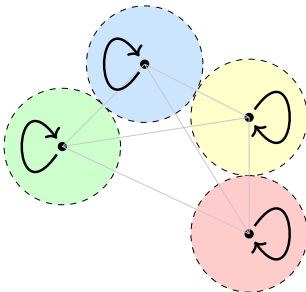
Automorphisms of codes

Definition

Let $\pi \in S_n$ and $\mathcal{C} \subseteq \mathbb{F}_q^n$:

$$\pi \text{ is an automorphism for } \mathcal{C} \iff \pi(\mathcal{C}) = \mathcal{C}$$

Examples:



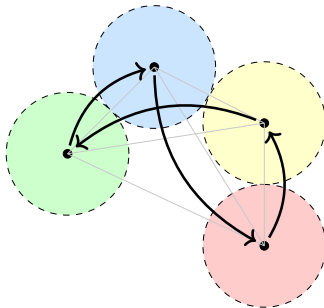
Automorphisms of codes

Definition

Let $\pi \in S_n$ and $\mathcal{C} \subseteq \mathbb{F}_q^n$:

$$\pi \text{ is an automorphism for } \mathcal{C} \iff \pi(\mathcal{C}) = \mathcal{C}$$

Examples:



The permutation automorphism group

Definition

Permutation automorphisms of any code form a subgroup of S_n :

$$\text{P-Aut}(\mathcal{C}) = \{\pi \in S_n \mid \pi(\mathcal{C}) = \mathcal{C}\} \leq S_n$$

We say $\text{P-Aut}(\mathcal{C})$ is non trivial if $\text{P-Aut}(\mathcal{C}) \neq \{id\}$

The permutation automorphism group

Definition

Permutation automorphisms of any code form a subgroup of S_n :

$$\text{P-Aut}(\mathcal{C}) = \{\pi \in S_n \mid \pi(\mathcal{C}) = \mathcal{C}\} \leq S_n$$

We say $\text{P-Aut}(\mathcal{C})$ is non trivial if $\text{P-Aut}(\mathcal{C}) \neq \{id\}$

The permutation automorphism group

Definition

Permutation automorphisms of any code form a subgroup of S_n :

$$\text{P-Aut}(\mathcal{C}) = \{\pi \in S_n \mid \pi(\mathcal{C}) = \mathcal{C}\} \leq S_n$$

We say $\text{P-Aut}(\mathcal{C})$ is non trivial if $\text{P-Aut}(\mathcal{C}) \neq \{id\}$

For $\mathbf{G} \in \mathbb{F}_q^{k \times n}$ a generator matrix for $\mathcal{C}$:

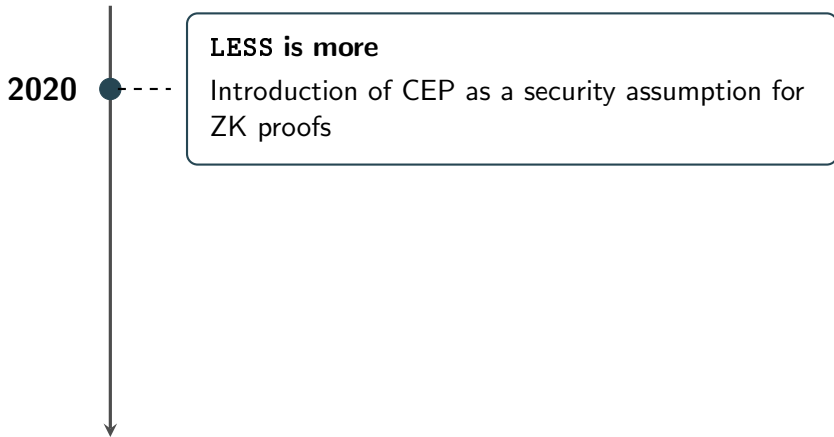
$$\mathbf{P} \in S_n \in \text{P-Aut}(\mathcal{C}) \iff \exists \mathbf{S} \in GL_k(\mathbb{F}_q) : \mathbf{SGP} = \mathbf{G}$$

The Code Equivalence Problem and LESS

Code Equivalence Problem (CEP): find isomorphism between two input codes

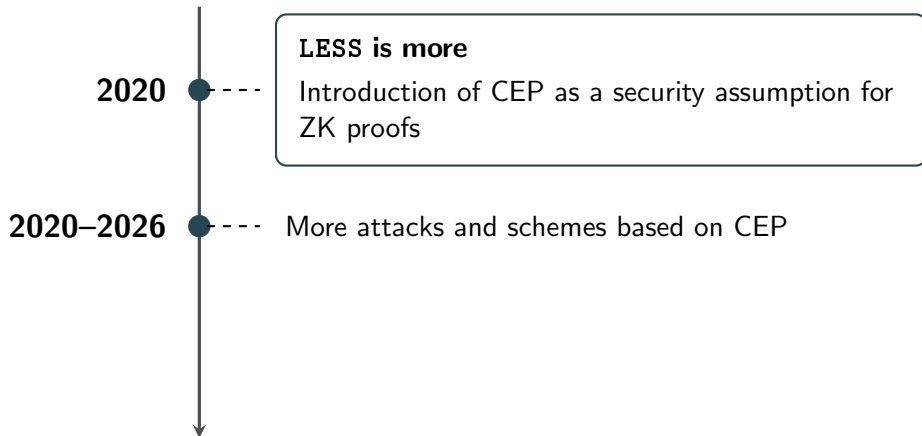
The Code Equivalence Problem and LESS

Code Equivalence Problem (CEP): find isomorphism between two input codes



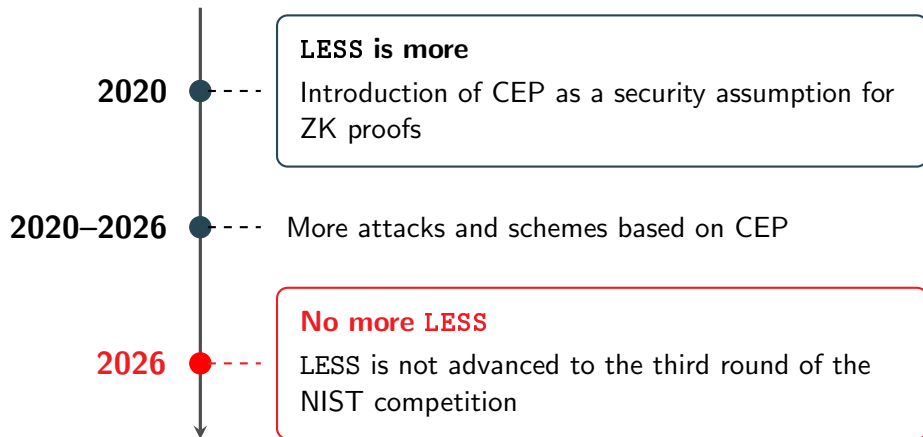
The Code Equivalence Problem and LESS

Code Equivalence Problem (CEP): find isomorphism between two input codes



The Code Equivalence Problem and LESS

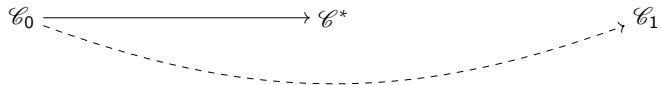
Code Equivalence Problem (CEP): find isomorphism between two input codes



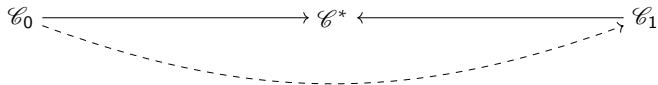
Advanced proofs of knowledge for CEP?



Advanced proofs of knowledge for CEP?

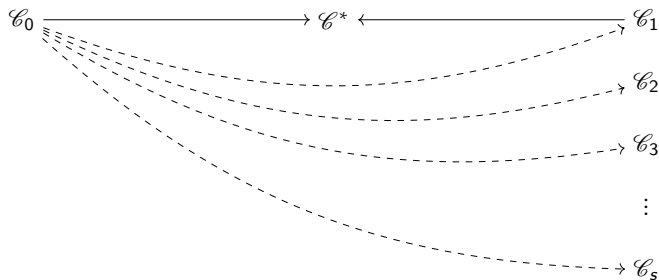


Advanced proofs of knowledge for CEP?



$$\text{Special soundness} \iff \text{Soundness error} = \frac{1}{|\text{Challenge space}|}$$

Advanced proofs of knowledge for CEP?

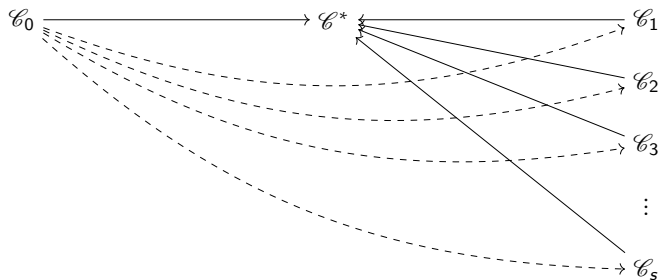


Special soundness

$\iff$

$$\text{Soundness error} = \frac{1}{|\text{Challenge space}|}$$

Advanced proofs of knowledge for CEP?

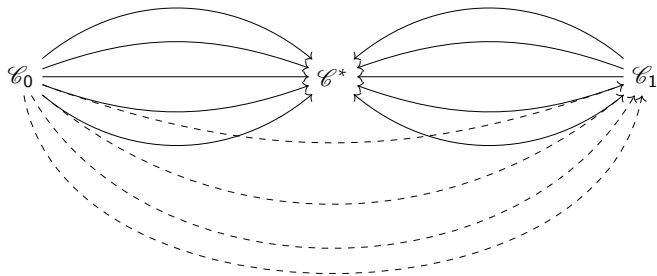


Special soundness



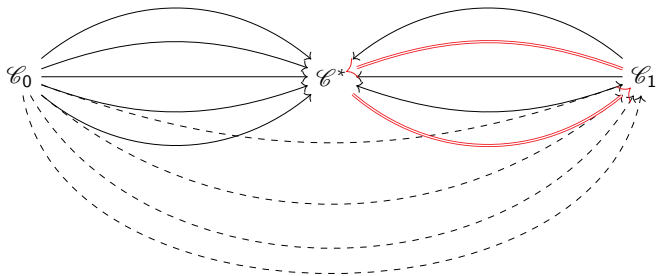
$$\text{Soundness error} = \frac{1}{|\text{Challenge space}|}$$

Advanced proofs of knowledge for CEP?



Special soundness $\iff$ Soundness error = $\frac{1}{|\text{Challenge space}|}$

Advanced proofs of knowledge for CEP?



Special soundness



$$\text{Soundness error} = \frac{1}{|\text{Challenge space}|}$$



Intuition: extractor either finds isomorphisms or automorphisms

Where are automorphisms employed?

- Cryptographic constructions
- Cryptanalysis of CEP

Where are automorphisms employed?

- Cryptographic constructions
- Cryptanalysis of CEP
- Algebraic characterization of codes
- Coding theory conjectures (e.g., extremal $[72, 36, 16]$ self-dual code)

Where are automorphisms employed?

- Cryptographic constructions
- Cryptanalysis of CEP
- Algebraic characterization of codes
- Coding theory conjectures (e.g., extremal $[72, 36, 16]$ self-dual code)
- Decoding algorithms

Where are automorphisms employed?

- Cryptographic constructions
- Cryptanalysis of CEP
- Algebraic characterization of codes
- Coding theory conjectures (e.g., extremal $[72, 36, 16]$ self-dual code)
- Decoding algorithms
- Fault tolerant quantum error correcting codes
- ...

**Code automorphisms: a
computational point of view**

State-of-the-art

Lefmann, Phelps, Rödl, 1993, Li, Wang, Zhou, 2026: random codes with sufficiently large dimension (and co-dimension) have only trivial automorphisms with overwhelming probability

State-of-the-art

Lefmann, Phelps, Rödl, 1993, Li, Wang, Zhou, 2026: random codes with sufficiently large dimension (and co-dimension) have only trivial automorphisms with overwhelming probability

Algorithms from the literature:

- Leon, 1982

State-of-the-art

Lefmann, Phelps, Rödl, 1993, Li, Wang, Zhou, 2026: random codes with sufficiently large dimension (and co-dimension) have only trivial automorphisms with overwhelming probability

Algorithms from the literature:

- Leon, 1982
- Feulner, 2009

State-of-the-art

Lefmann, Phelps, Rödl, 1993, Li, Wang, Zhou, 2026: random codes with sufficiently large dimension (and co-dimension) have only trivial automorphisms with overwhelming probability

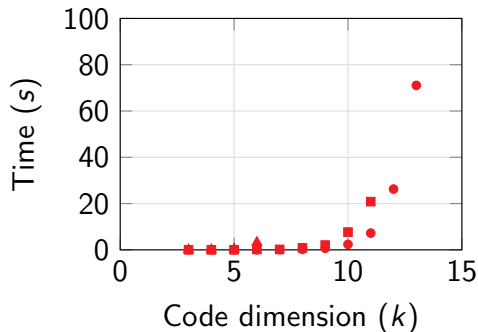
Algorithms from the literature:

- Leon, 1982
- Feulner, 2009

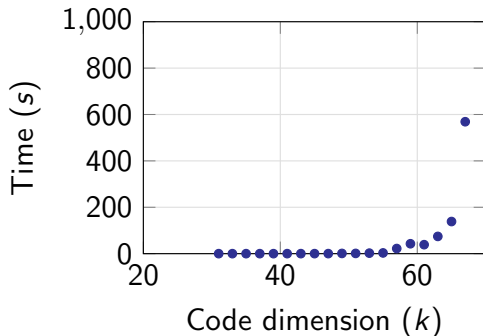
Unless "extreme" codes are considered, Leon's and Feulner's algorithms run in **exponential time**.

Finding automorphisms in practice

Sagemath and Magma implement Leon's and Feulner's algorithms



(a) Sagemath, $q = 3$



(b) Magma, $q = 2$

Figure: Timings for $n = 2k$, random codes with a cyclic automorphism group.

List of problems about code automorphisms

AUT

On input $\mathcal{C} \subseteq \mathbb{F}_q^n$, find a non trivial automorphism $\pi \in \text{P-Aut}(\mathcal{C})$

Glossary:

- Generating set: set $S = \{\phi_1, \dots, \phi_m\} \subseteq G$ such that every $g \in G$ can be expressed as a combination of finitely many elements of S (and their inverses)
- Order of a group: number of its elements

List of problems about code automorphisms

AUT

On input $\mathcal{C} \subseteq \mathbb{F}_q^n$, find a non trivial automorphism $\pi \in \text{P-Aut}(\mathcal{C})$

AUT-Gen

On input $\mathcal{C} \subseteq \mathbb{F}_q^n$, find a (poly-size) generating set for $\text{P-Aut}(\mathcal{C})$

Glossary:

- Generating set: set $S = \{\phi_1, \dots, \phi_m\} \subseteq G$ such that every $g \in G$ can be expressed as a combination of finitely many elements of S (and their inverses)
- Order of a group: number of its elements

List of problems about code automorphisms

AUT

On input $\mathcal{C} \subseteq \mathbb{F}_q^n$, find a non trivial automorphism $\pi \in \text{P-Aut}(\mathcal{C})$

AUT-Gen

On input $\mathcal{C} \subseteq \mathbb{F}_q^n$, find a (poly-size) generating set for $\text{P-Aut}(\mathcal{C})$

AUT-Count

On input $\mathcal{C} \subseteq \mathbb{F}_q^n$, find the order of $\text{P-Aut}(\mathcal{C})$

Glossary:

- Generating set: set $S = \{\phi_1, \dots, \phi_m\} \subseteq G$ such that every $g \in G$ can be expressed as a combination of finitely many elements of S (and their inverses)
- Order of a group: number of its elements

List of other problems

PEP

Given $\mathcal{A}, \mathcal{B} \subseteq \mathbb{F}_q^n$, find $\pi \in S_n$ such that $\mathcal{B} = \pi(\mathcal{A})$

List of other problems

PEP

Given $\mathcal{A}, \mathcal{B} \subseteq \mathbb{F}_q^n$, find $\pi \in S_n$ such that $\mathcal{B} = \pi(\mathcal{A})$

GIP

Given two graphs $\mathcal{G}, \mathcal{G}'$, find an isomorphism from $\mathcal{G}$ to $\mathcal{G}'$

List of other problems

PEP

Given $\mathcal{A}, \mathcal{B} \subseteq \mathbb{F}_q^n$, find $\pi \in S_n$ such that $\mathcal{B} = \pi(\mathcal{A})$

GIP

Given two graphs $\mathcal{G}, \mathcal{G}'$, find an isomorphism from $\mathcal{G}$ to $\mathcal{G}'$

GAP-Gen

Given a graph $\mathcal{G}$, find a (poly-size) generating set for $\text{P-Aut}(\mathcal{G})$

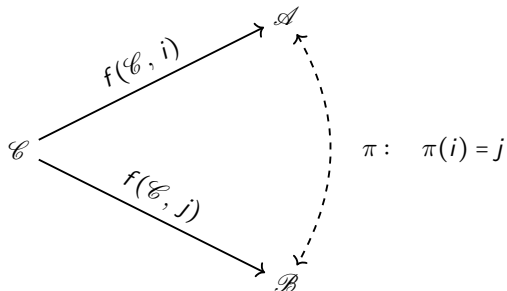
From isomorphisms to automorphisms, and viceversa

From AUT to PEP: the main idea



Do "something" on coordinates i, j , get codes $\mathcal{A}$ and $\mathcal{B}$ such that

$$\exists \varphi \in \text{P-Aut}(\mathcal{C}) : \quad \varphi(i) = j \quad \implies \quad \mathcal{A} \cong \mathcal{B}$$

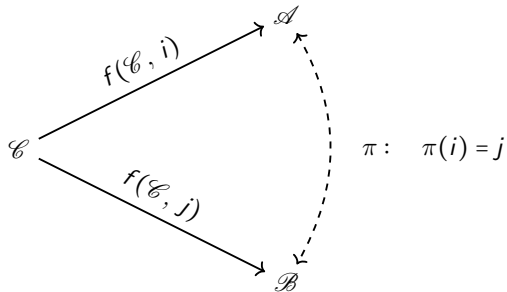


From AUT to PEP: the main idea



Do "something" on coordinates i, j , get codes $\mathcal{A}$ and $\mathcal{B}$ such that

$$\exists \varphi \in \text{P-Aut}(\mathcal{C}) : \quad \varphi(i) = j \quad \implies \quad \mathcal{A} \cong \mathcal{B}$$



Observation: if $\varphi \in \text{P-Aut}(\mathcal{C})$ is non trivial, then there exists $i, j \in \{1, \dots, n\}$, $i \neq j$ such that $\varphi(i) = j$

From AUT to PEP

On input $\mathcal{C} \subseteq \mathbb{F}_q^n$ with no repeated coordinates:

- 1) For $i \in \{1, \dots, n\}$
- 2) $\mathcal{A} \leftarrow \mathcal{C}$ with repeated i -th coordinate $\backslash\backslash$ First input for PEP
- 3) For $j \in \{1, \dots, n\} \setminus \{i\}$: $\backslash\backslash$ Brute force image of i , under action of $\text{P-Aut}(\mathcal{C})$
- 4) $\mathcal{B} \leftarrow \mathcal{C}$ with repeated j -th coordinate $\backslash\backslash$ Second input for PEP
- 5) Solve PEP on $\{\mathcal{A}, \mathcal{B}\}$: if solution is found, reconstruct automorphism of $\mathcal{C}$
- 6) If PEP never has solutions, $\mathcal{C}$ has no non-trivial automorphism

From AUT to PEP

On input $\mathcal{C} \subseteq \mathbb{F}_q^n$ with no repeated coordinates:

- 1) For $i \in \{1, \dots, n\}$
- 2) $\mathcal{A} \leftarrow \mathcal{C}$ with repeated i -th coordinate $\backslash\backslash$ First input for PEP
- 3) For $j \in \{1, \dots, n\} \setminus \{i\}$: $\backslash\backslash$ Brute force image of i , under action of $\text{P-Aut}(\mathcal{C})$
- 4) $\mathcal{B} \leftarrow \mathcal{C}$ with repeated j -th coordinate $\backslash\backslash$ Second input for PEP
- 5) Solve PEP on $\{\mathcal{A}, \mathcal{B}\}$: if solution is found, reconstruct automorphism of $\mathcal{C}$
- 6) If PEP never has solutions, $\mathcal{C}$ has no non-trivial automorphism

Example: $n = 4$

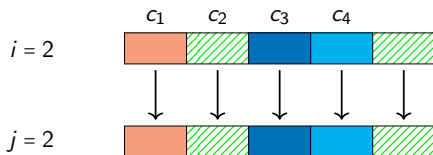


From AUT to PEP

On input $\mathcal{C} \subseteq \mathbb{F}_q^n$ with no repeated coordinates:

- 1) For $i \in \{1, \dots, n\}$
- 2) $\mathcal{A} \leftarrow \mathcal{C}$ with repeated i -th coordinate $\backslash\backslash$ First input for PEP
- 3) For $j \in \{1, \dots, n\} \setminus \{i\}$: $\backslash\backslash$ Brute force image of i , under action of $\text{P-Aut}(\mathcal{C})$
- 4) $\mathcal{B} \leftarrow \mathcal{C}$ with repeated j -th coordinate $\backslash\backslash$ Second input for PEP
- 5) Solve PEP on $\{\mathcal{A}, \mathcal{B}\}$: if solution is found, reconstruct automorphism of $\mathcal{C}$
- 6) If PEP never has solutions, $\mathcal{C}$ has no non-trivial automorphism

Example: $n = 4$



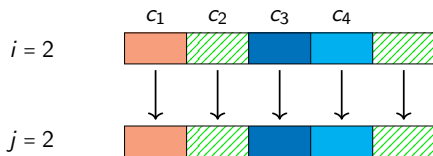
$$\pi(1) = 1, \quad \pi(2) = 2, \quad \pi(3) = 3, \quad \pi(4) = 4, \quad \pi(5) = 5$$

From AUT to PEP

On input $\mathcal{C} \subseteq \mathbb{F}_q^n$ with no repeated coordinates:

- 1) For $i \in \{1, \dots, n\}$
- 2) $\mathcal{A} \leftarrow \mathcal{C}$ with repeated i -th coordinate $\backslash\backslash$ First input for PEP
- 3) For $j \in \{1, \dots, n\} \setminus \{i\}$: $\backslash\backslash$ Brute force image of i , under action of $\text{P-Aut}(\mathcal{C})$
- 4) $\mathcal{B} \leftarrow \mathcal{C}$ with repeated j -th coordinate $\backslash\backslash$ Second input for PEP
- 5) Solve PEP on $\{\mathcal{A}, \mathcal{B}\}$: if solution is found, reconstruct automorphism of $\mathcal{C}$
- 6) If PEP never has solutions, $\mathcal{C}$ has no non-trivial automorphism

Example: $n = 4$



$$\pi(1) = 1, \quad \pi(2) = 2, \quad \pi(3) = 3, \quad \pi(4) = 4, \quad \pi(5) = 5$$

$$\varphi = id \in \text{P-Aut}(\mathcal{C})$$

From AUT to PEP

On input $\mathcal{C} \subseteq \mathbb{F}_q^n$ with no repeated coordinates:

- 1) For $i \in \{1, \dots, n\}$
- 2) $\mathcal{A} \leftarrow \mathcal{C}$ with repeated i -th coordinate $\backslash\backslash$ First input for PEP
- 3) For $j \in \{1, \dots, n\} \setminus \{i\}$: $\backslash\backslash$ Brute force image of i , under action of $\text{P-Aut}(\mathcal{C})$
- 4) $\mathcal{B} \leftarrow \mathcal{C}$ with repeated j -th coordinate $\backslash\backslash$ Second input for PEP
- 5) Solve PEP on $\{\mathcal{A}, \mathcal{B}\}$: if solution is found, reconstruct automorphism of $\mathcal{C}$
- 6) If PEP never has solutions, $\mathcal{C}$ has no non-trivial automorphism

Example: $n = 4$

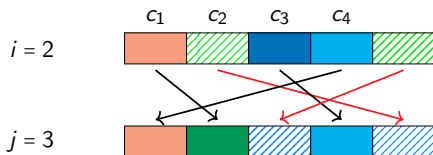


From AUT to PEP

On input $\mathcal{C} \subseteq \mathbb{F}_q^n$ with no repeated coordinates:

- 1) For $i \in \{1, \dots, n\}$
- 2) $\mathcal{A} \leftarrow \mathcal{C}$ with repeated i -th coordinate $\backslash\backslash$ First input for PEP
- 3) For $j \in \{1, \dots, n\} \setminus \{i\}$: $\backslash\backslash$ Brute force image of i , under action of $\text{P-Aut}(\mathcal{C})$
- 4) $\mathcal{B} \leftarrow \mathcal{C}$ with repeated j -th coordinate $\backslash\backslash$ Second input for PEP
- 5) Solve PEP on $\{\mathcal{A}, \mathcal{B}\}$: if solution is found, reconstruct automorphism of $\mathcal{C}$
- 6) If PEP never has solutions, $\mathcal{C}$ has no non-trivial automorphism

Example: $n = 4$

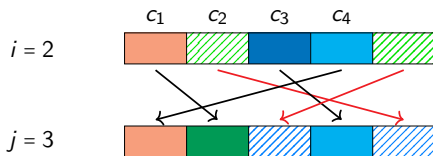


From AUT to PEP

On input $\mathcal{C} \subseteq \mathbb{F}_q^n$ with no repeated coordinates:

- 1) For $i \in \{1, \dots, n\}$
- 2) $\mathcal{A} \leftarrow \mathcal{C}$ with repeated i -th coordinate $\backslash\backslash$ First input for PEP
- 3) For $j \in \{1, \dots, n\} \setminus \{i\}$: $\backslash\backslash$ Brute force image of i , under action of $\text{P-Aut}(\mathcal{C})$
- 4) $\mathcal{B} \leftarrow \mathcal{C}$ with repeated j -th coordinate $\backslash\backslash$ Second input for PEP
- 5) Solve PEP on $\{\mathcal{A}, \mathcal{B}\}$: if solution is found, reconstruct automorphism of $\mathcal{C}$
- 6) If PEP never has solutions, $\mathcal{C}$ has no non-trivial automorphism

Example: $n = 4$



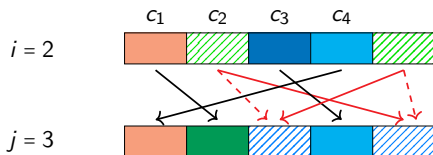
$$\pi(1) = 2, \quad \pi(2) = 5, \quad \pi(3) = 4, \quad \pi(4) = 1, \quad \pi(5) = 2$$

From AUT to PEP

On input $\mathcal{C} \subseteq \mathbb{F}_q^n$ with no repeated coordinates:

- 1) For $i \in \{1, \dots, n\}$
- 2) $\mathcal{A} \leftarrow \mathcal{C}$ with repeated i -th coordinate $\backslash\backslash$ First input for PEP
- 3) For $j \in \{1, \dots, n\} \setminus \{i\}$: $\backslash\backslash$ Brute force image of i , under action of $\text{P-Aut}(\mathcal{C})$
- 4) $\mathcal{B} \leftarrow \mathcal{C}$ with repeated j -th coordinate $\backslash\backslash$ Second input for PEP
- 5) Solve PEP on $\{\mathcal{A}, \mathcal{B}\}$: if solution is found, reconstruct automorphism of $\mathcal{C}$
- 6) If PEP never has solutions, $\mathcal{C}$ has no non-trivial automorphism

Example: $n = 4$



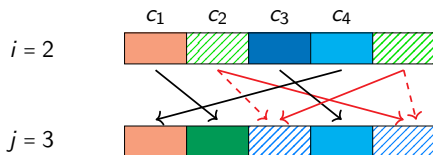
$$\pi(1) = 2, \quad \pi(2) = 3, \quad \pi(3) = 4, \quad \pi(4) = 1, \quad \pi(5) = 5$$

From AUT to PEP

On input $\mathcal{C} \subseteq \mathbb{F}_q^n$ with no repeated coordinates:

- 1) For $i \in \{1, \dots, n\}$
- 2) $\mathcal{A} \leftarrow \mathcal{C}$ with repeated i -th coordinate $\backslash\backslash$ First input for PEP
- 3) For $j \in \{1, \dots, n\} \setminus \{i\}$: $\backslash\backslash$ Brute force image of i , under action of $\text{P-Aut}(\mathcal{C})$
- 4) $\mathcal{B} \leftarrow \mathcal{C}$ with repeated j -th coordinate $\backslash\backslash$ Second input for PEP
- 5) Solve PEP on $\{\mathcal{A}, \mathcal{B}\}$: if solution is found, reconstruct automorphism of $\mathcal{C}$
- 6) If PEP never has solutions, $\mathcal{C}$ has no non-trivial automorphism

Example: $n = 4$



$$\pi(1) = 2, \quad \pi(2) = 3, \quad \pi(3) = 4, \quad \pi(4) = 1, \quad \pi(5) = 5$$

$$\varphi \in \text{P-Aut}(\mathcal{C}), \quad \varphi \neq id$$

The Cartesian product of codes

Definition

Given two component codes $\mathcal{A}, \mathcal{B} \subseteq \mathbb{F}_q^n$, their **Cartesian product** is the code

$$\mathcal{A} \times \mathcal{B} = \{(\mathbf{a}, \mathbf{b}) \mid \mathbf{a} \in \mathcal{A}, \mathbf{b} \in \mathcal{B}\}$$

Given generator matrices $\mathbf{A}, \mathbf{B} \in \mathbb{F}_q^{k \times n}$ for $\mathcal{A}$ and $\mathcal{B}$, $\mathcal{A} \times \mathcal{B}$ is generated by

$$\begin{pmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{B} \end{pmatrix} \in \mathbb{F}_q^{2k \times 2n}$$

The automorphism group of the Cartesian product of codes

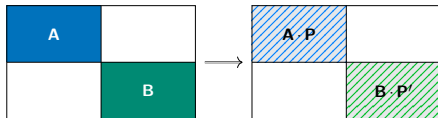
If $\mathcal{A}$ and $\mathcal{B}$ are indencomposable (they are not the Cartesian product of codes), there are only two types of automorphisms: **component-wise** and **swapping**.

Component-wise: $\begin{pmatrix} \mathbf{P} & \mathbf{0} \\ \mathbf{0} & \mathbf{P}' \end{pmatrix}, \quad \begin{matrix} \mathbf{P} \in \text{P-Aut}(\mathcal{A}) \\ \mathbf{P}' \in \text{P-Aut}(\mathcal{B}) \end{matrix}$

The automorphism group of the Cartesian product of codes

If $\mathcal{A}$ and $\mathcal{B}$ are indencomposable (they are not the Cartesian product of codes), there are only two types of automorphisms: **component-wise** and **swapping**.

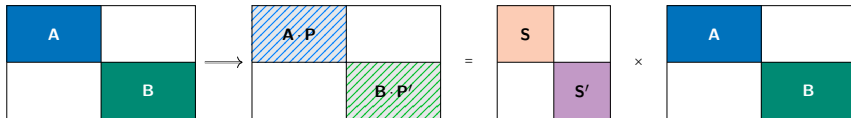
Component-wise: $\begin{pmatrix} \mathbf{P} & \mathbf{0} \\ \mathbf{0} & \mathbf{P}' \end{pmatrix}, \quad \begin{matrix} \mathbf{P} \in \text{P-Aut}(\mathcal{A}) \\ \mathbf{P}' \in \text{P-Aut}(\mathcal{B}) \end{matrix}$



The automorphism group of the Cartesian product of codes

If $\mathcal{A}$ and $\mathcal{B}$ are indencomposable (they are not the Cartesian product of codes), there are only two types of automorphisms: **component-wise** and **swapping**.

Component-wise: $\begin{pmatrix} \mathbf{P} & \mathbf{0} \\ \mathbf{0} & \mathbf{P}' \end{pmatrix}, \quad \begin{matrix} \mathbf{P} \in \text{P-Aut}(\mathcal{A}) \\ \mathbf{P}' \in \text{P-Aut}(\mathcal{B}) \end{matrix}$



The automorphism group of the Cartesian product of codes

If $\mathcal{A}$ and $\mathcal{B}$ are indencomposable (they are not the Cartesian product of codes), there are only two types of automorphisms: **component-wise** and **swapping**.

Component-wise: $\begin{pmatrix} \mathbf{P} & \mathbf{0} \\ \mathbf{0} & \mathbf{P}' \end{pmatrix}, \quad \begin{matrix} \mathbf{P} \in \text{P-Aut}(\mathcal{A}) \\ \mathbf{P}' \in \text{P-Aut}(\mathcal{B}) \end{matrix}$

Swapping: $\begin{pmatrix} \mathbf{0} & \mathbf{P}_{\mathcal{A}} \cdot \mathbf{P} \cdot \mathbf{P}_{\mathcal{B}} \\ \mathbf{P}'_{\mathcal{B}} \cdot \mathbf{P}^{\top} \cdot \mathbf{P}'_{\mathcal{A}} & \mathbf{0} \end{pmatrix}, \quad \mathbf{P} \in S_n : \quad \mathcal{B} = \mathcal{A} \cdot \mathbf{P}, \quad \begin{matrix} \mathbf{P}_{\mathcal{A}}, \mathbf{P}'_{\mathcal{A}} \in \text{P-Aut}(\mathcal{A}) \\ \mathbf{P}_{\mathcal{B}}, \mathbf{P}'_{\mathcal{B}} \in \text{P-Aut}(\mathcal{B}) \end{matrix}$

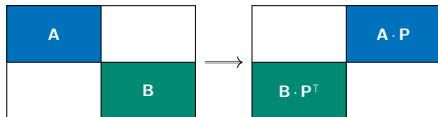
The automorphism group of the Cartesian product of codes

If $\mathcal{A}$ and $\mathcal{B}$ are indencomposable (they are not the Cartesian product of codes), there are only two types of automorphisms: **component-wise** and **swapping**.

Component-wise: $\begin{pmatrix} \mathbf{P} & \mathbf{0} \\ \mathbf{0} & \mathbf{P}' \end{pmatrix}, \quad \mathbf{P} \in \text{P-Aut}(\mathcal{A}), \mathbf{P}' \in \text{P-Aut}(\mathcal{B})$

Swapping: $\begin{pmatrix} \mathbf{0} & \mathbf{P}_{\mathcal{A}} \cdot \mathbf{P} \cdot \mathbf{P}_{\mathcal{B}} \\ \mathbf{P}'_{\mathcal{B}} \cdot \mathbf{P}^{\top} \cdot \mathbf{P}'_{\mathcal{A}} & \mathbf{0} \end{pmatrix}, \quad \mathbf{P} \in S_n: \quad \mathcal{B} = \mathcal{A} \cdot \mathbf{P},$

$\mathbf{P}_{\mathcal{A}}, \mathbf{P}'_{\mathcal{A}} \in \text{P-Aut}(\mathcal{A})$
 $\mathbf{P}_{\mathcal{B}}, \mathbf{P}'_{\mathcal{B}} \in \text{P-Aut}(\mathcal{B})$



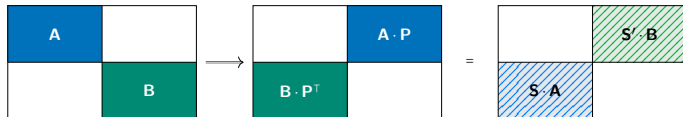
The automorphism group of the Cartesian product of codes

If $\mathcal{A}$ and $\mathcal{B}$ are indencomposable (they are not the Cartesian product of codes), there are only two types of automorphisms: **component-wise** and **swapping**.

Component-wise: $\begin{pmatrix} \mathbf{P} & \mathbf{0} \\ \mathbf{0} & \mathbf{P}' \end{pmatrix}, \quad \begin{matrix} \mathbf{P} \in \text{P-Aut}(\mathcal{A}) \\ \mathbf{P}' \in \text{P-Aut}(\mathcal{B}) \end{matrix}$

Swapping: $\begin{pmatrix} \mathbf{0} & \mathbf{P}_{\mathcal{A}} \cdot \mathbf{P} \cdot \mathbf{P}_{\mathcal{B}} \\ \mathbf{P}'_{\mathcal{B}} \cdot \mathbf{P}^{\top} \cdot \mathbf{P}'_{\mathcal{A}} & \mathbf{0} \end{pmatrix}, \quad \mathbf{P} \in S_n: \quad \mathcal{B} = \mathcal{A} \cdot \mathbf{P},$

$\begin{matrix} \mathbf{P}_{\mathcal{A}}, \mathbf{P}'_{\mathcal{A}} \in \text{P-Aut}(\mathcal{A}) \\ \mathbf{P}_{\mathcal{B}}, \mathbf{P}'_{\mathcal{B}} \in \text{P-Aut}(\mathcal{B}) \end{matrix}$

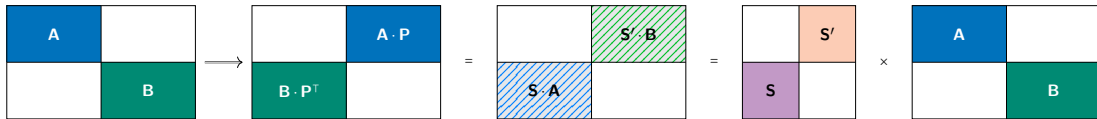


The automorphism group of the Cartesian product of codes

If $\mathcal{A}$ and $\mathcal{B}$ are indencomposable (they are not the Cartesian product of codes), there are only two types of automorphisms: **component-wise** and **swapping**.

Component-wise: $\begin{pmatrix} \mathbf{P} & \mathbf{0} \\ \mathbf{0} & \mathbf{P}' \end{pmatrix}, \quad \begin{matrix} \mathbf{P} \in \text{P-Aut}(\mathcal{A}) \\ \mathbf{P}' \in \text{P-Aut}(\mathcal{B}) \end{matrix}$

Swapping: $\begin{pmatrix} \mathbf{0} & \mathbf{P}_{\mathcal{A}} \cdot \mathbf{P} \cdot \mathbf{P}_{\mathcal{B}} \\ \mathbf{P}'_{\mathcal{B}} \cdot \mathbf{P}^{\top} \cdot \mathbf{P}'_{\mathcal{A}} & \mathbf{0} \end{pmatrix}, \quad \mathbf{P} \in S_n: \quad \mathcal{B} = \mathcal{A} \cdot \mathbf{P},$ $\begin{matrix} \mathbf{P}_{\mathcal{A}}, \mathbf{P}'_{\mathcal{A}} \in \text{P-Aut}(\mathcal{A}) \\ \mathbf{P}_{\mathcal{B}}, \mathbf{P}'_{\mathcal{B}} \in \text{P-Aut}(\mathcal{B}) \end{matrix}$



From PEP to AUT-Gen

On input indecomposable codes $\mathcal{A}, \mathcal{B} \subseteq \mathbb{F}_q^n$:

- 1) Set $\mathcal{C} = \mathcal{A} \times \mathcal{B}$ $\backslash \backslash$ Isomorphisms from $\mathcal{A}$ to $\mathcal{B}$ are automorphisms of $\mathcal{C}$
- 2) $\{\varphi_1, \dots, \varphi_m\} \leftarrow$ solution of AUT-Gen on $\mathcal{C}$ $\backslash \backslash$ Generating set for $\text{P-Aut}(\mathcal{C})$

Note: if $\mathcal{A} \cong \mathcal{B}$, any generating set for $\text{P-Aut}(\mathcal{A} \times \mathcal{B})$ must contain at least one swapping automorphism

From PEP to AUT-Gen

On input indecomposable codes $\mathcal{A}, \mathcal{B} \subseteq \mathbb{F}_q^n$:

- 1) Set $\mathcal{C} = \mathcal{A} \times \mathcal{B}$ $\backslash\backslash$ Isomorphisms from $\mathcal{A}$ to $\mathcal{B}$ are automorphisms of $\mathcal{C}$
- 2) $\{\varphi_1, \dots, \varphi_m\} \leftarrow$ solution of AUT-Gen on $\mathcal{C}$ $\backslash\backslash$ Generating set for $\text{P-Aut}(\mathcal{C})$
- 3) If $\exists i$: φ_i is swapping:
- 4) Extract and output the permutation $\backslash\backslash$ Solution to PEP!

Note: if $\mathcal{A} \cong \mathcal{B}$, any generating set for $\text{P-Aut}(\mathcal{A} \times \mathcal{B})$ must contain at least one swapping automorphism

From PEP to AUT-Gen

On input indecomposable codes $\mathcal{A}, \mathcal{B} \subseteq \mathbb{F}_q^n$:

- 1) Set $\mathcal{C} = \mathcal{A} \times \mathcal{B}$ $\backslash\backslash$ Isomorphisms from $\mathcal{A}$ to $\mathcal{B}$ are automorphisms of $\mathcal{C}$
- 2) $\{\varphi_1, \dots, \varphi_m\} \leftarrow$ solution of AUT-Gen on $\mathcal{C}$ $\backslash\backslash$ Generating set for $\text{P-Aut}(\mathcal{C})$
- 3) If $\exists i$: φ_i is swapping:
- 4) Extract and output the permutation $\backslash\backslash$ Solution to PEP!
- 5) Else:
- 6) $\mathcal{A}, \mathcal{B}$ are not isomorphic $\backslash\backslash$ PEP has no solution!

Note: if $\mathcal{A} \cong \mathcal{B}$, any generating set for $\text{P-Aut}(\mathcal{A} \times \mathcal{B})$ must contain at least one swapping automorphism

From Dec-PEP to AUT-Count

Main intuition:

- swapping automorphisms exist if and only $\mathcal{A} \cong \mathcal{B}$
- if $\mathcal{A} \not\cong \mathcal{B}$: $\text{P-Aut}(\mathcal{A} \times \mathcal{B}) \cong \text{P-Aut}(\mathcal{A}) \times \text{P-Aut}(\mathcal{B})$

From Dec-PEP to AUT-Count

Main intuition:

- swapping automorphisms exist if and only $\mathcal{A} \cong \mathcal{B}$
- if $\mathcal{A} \not\cong \mathcal{B}$: $\text{P-Aut}(\mathcal{A} \times \mathcal{B}) \cong \text{P-Aut}(\mathcal{A}) \times \text{P-Aut}(\mathcal{B})$

Sketch of reduction:

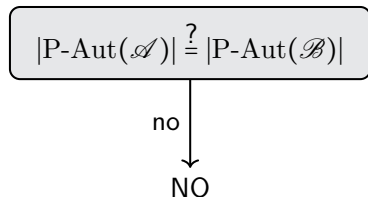
$$|\text{P-Aut}(\mathcal{A})| \stackrel{?}{=} |\text{P-Aut}(\mathcal{B})|$$

From Dec-PEP to AUT-Count

Main intuition:

- swapping automorphisms exist if and only $\mathcal{A} \cong \mathcal{B}$
- if $\mathcal{A} \not\cong \mathcal{B}$: $\text{P-Aut}(\mathcal{A} \times \mathcal{B}) \cong \text{P-Aut}(\mathcal{A}) \times \text{P-Aut}(\mathcal{B})$

Sketch of reduction:

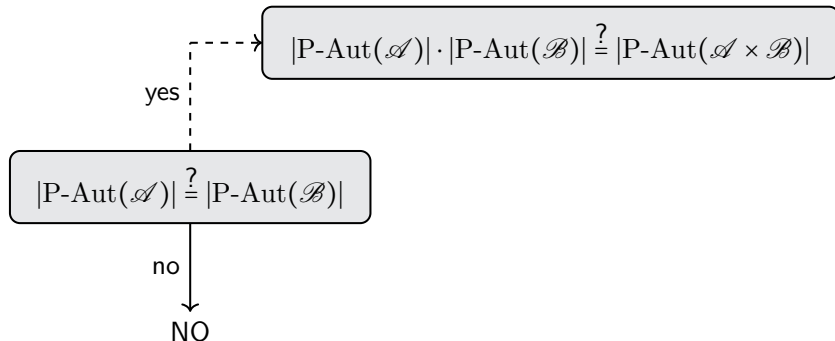


From Dec-PEP to AUT-Count

Main intuition:

- swapping automorphisms exist if and only $\mathcal{A} \cong \mathcal{B}$
- if $\mathcal{A} \not\cong \mathcal{B}$: $\text{P-Aut}(\mathcal{A} \times \mathcal{B}) \cong \text{P-Aut}(\mathcal{A}) \times \text{P-Aut}(\mathcal{B})$

Sketch of reduction:

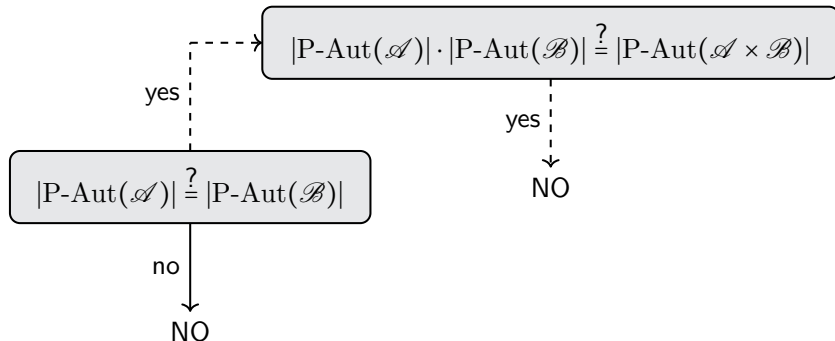


From Dec-PEP to AUT-Count

Main intuition:

- swapping automorphisms exist if and only $\mathcal{A} \cong \mathcal{B}$
- if $\mathcal{A} \not\cong \mathcal{B}$: $\text{P-Aut}(\mathcal{A} \times \mathcal{B}) \cong \text{P-Aut}(\mathcal{A}) \times \text{P-Aut}(\mathcal{B})$

Sketch of reduction:

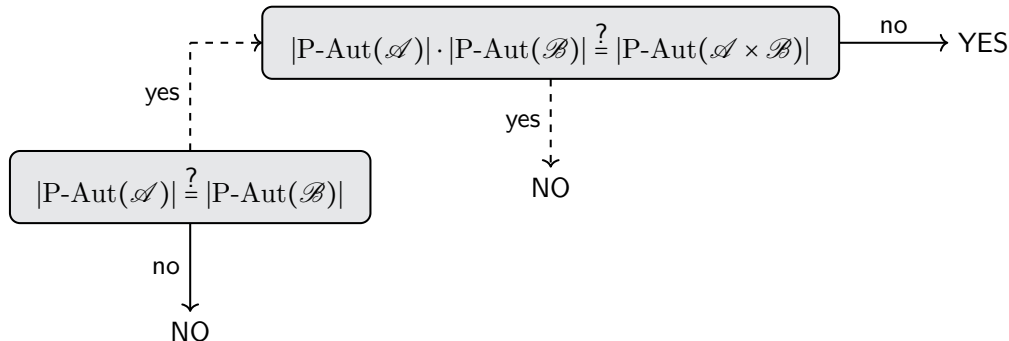


From Dec-PEP to AUT-Count

Main intuition:

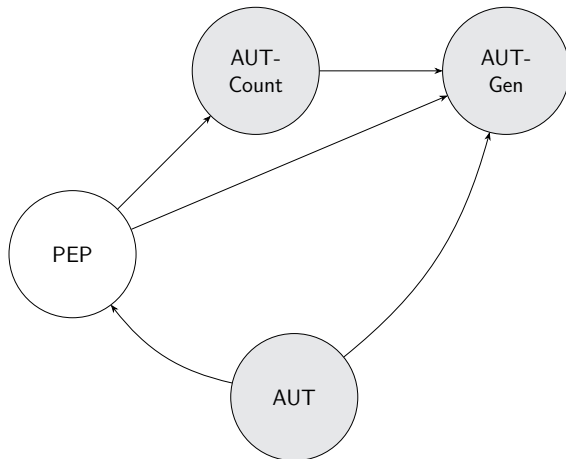
- swapping automorphisms exist if and only $\mathcal{A} \cong \mathcal{B}$
- if $\mathcal{A} \not\cong \mathcal{B}$: $\text{P-Aut}(\mathcal{A} \times \mathcal{B}) \cong \text{P-Aut}(\mathcal{A}) \times \text{P-Aut}(\mathcal{B})$

Sketch of reduction:



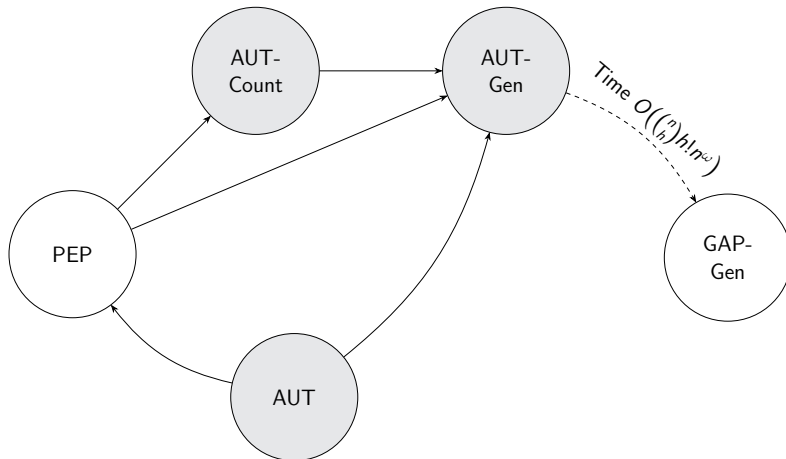
What we've seen so far...

With some technicalities, reductions work also for decomposable codes.



What we've seen so far...

With some technicalities, reductions work also for decomposable codes.



Graphs, codes and automorphisms

The hull of a linear code

Definition

Given a code $\mathcal{C} \subseteq \mathbb{F}_q^n$ with dimension k , its **hull** is

$$\mathcal{H}(\mathcal{C}) = \mathcal{C} \cap \mathcal{C}^\perp$$

The hull dimension is $h \leq \min \{k ; n - k\}$

We say the hull is trivial ($h = 0$) if $\mathcal{H}(\mathcal{C}) = \{\mathbf{0}\}$

Facts:

- for a random code, $\Pr [\text{Dim}(\mathcal{H}(\mathcal{C})) = h] \approx \begin{cases} 1 - 1/q & \text{if } h = 0, \\ q^{-h^2/2} & \text{if } h \geq 1. \end{cases}$
- $h = k - \text{rank}(\mathbf{G}\mathbf{G}^\top)$

From codes to graphs

Definition (Bardet, Otmani, Saeed-Taha, 2019)

For $\mathbf{G} \in \mathbb{F}_q^{k \times n}$ such that $\text{Rank}(\mathbf{G}\mathbf{G}^\top) = k$, let

$$\text{Adj}(\mathbf{G}) = \mathbf{G}^\top \cdot (\mathbf{G} \cdot \mathbf{G}^\top)^{-1} \cdot \mathbf{G}$$

Property: for any $\mathbf{S} \in GL_k(\mathbb{F}_q)$: $\text{Adj}(\mathbf{G}) = \text{Adj}(\mathbf{S} \cdot \mathbf{G})$

From codes to graphs

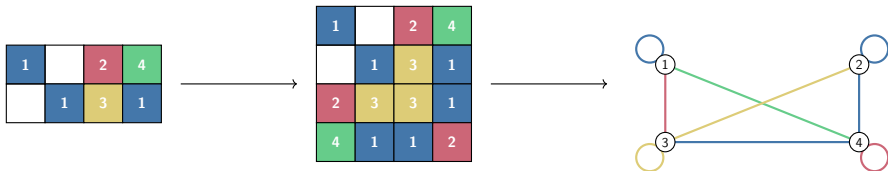
Definition (Bardet, Otmani, Saeed-Taha, 2019)

For $\mathbf{G} \in \mathbb{F}_q^{k \times n}$ such that $\text{Rank}(\mathbf{G}\mathbf{G}^\top) = k$, let

$$\text{Adj}(\mathbf{G}) = \mathbf{G}^\top \cdot (\mathbf{G} \cdot \mathbf{G}^\top)^{-1} \cdot \mathbf{G}$$

Property: for any $\mathbf{S} \in GL_k(\mathbb{F}_q)$: $\text{Adj}(\mathbf{G}) = \text{Adj}(\mathbf{S} \cdot \mathbf{G})$

Example over $\mathbb{F}_5$, $k = 2$, $n = 4$:



From PEP to GIP (trivial hull)

Observation: If undirected graphs $\mathcal{G}, \mathcal{G}'$ with adjacency matrices $\mathbf{X}, \mathbf{Y}$ are isomorphic, then:

$$\exists \mathbf{P} \in S_n : \quad \mathbf{Y} = \mathbf{P}^\top \cdot \mathbf{X} \cdot \mathbf{P}$$

From PEP to GIP (trivial hull)

Observation: If undirected graphs $\mathcal{G}, \mathcal{G}'$ with adjacency matrices $\mathbf{X}, \mathbf{Y}$ are isomorphic, then:

$$\exists \mathbf{P} \in S_n : \quad \mathbf{Y} = \mathbf{P}^\top \cdot \mathbf{X} \cdot \mathbf{P}$$

Theorem (Bardet, Otmani, Saeed-Taha, 2019)

For codes with trivial hull, $\text{PEP} \leq_p \text{GIP}$

Proof

Let $\mathbf{A}, \mathbf{B} = \mathbf{S} \cdot \mathbf{A} \cdot \mathbf{P}$ be generator matrices. Then

$$\begin{aligned} \text{Adj}(\mathbf{B}) &= \text{Adj}(\mathbf{A} \cdot \mathbf{P}) \\ &= \mathbf{P}^\top \cdot \mathbf{A}^\top \cdot (\mathbf{A} \cdot \mathbf{P} \cdot \mathbf{P}^\top \cdot \mathbf{A}^\top)^{-1} \cdot \mathbf{A} \cdot \mathbf{P} \\ &= \mathbf{P}^\top \cdot \mathbf{A}^\top \cdot \underbrace{(\mathbf{A} \cdot \mathbf{A}^\top)^{-1}}_{\text{Adj}(\mathbf{A})} \cdot \mathbf{A} \cdot \mathbf{P} \end{aligned}$$

From AUT-Gen to GAP-Gen (trivial hull)

Theorem

For a code $\mathcal{C} \subseteq \mathbb{F}_q^n$ with trivial hull, $\text{AUT-Gen} \leq_p \text{GAP-Gen}$

From AUT-Gen to GAP-Gen (trivial hull)

Theorem

For a code $\mathcal{C} \subseteq \mathbb{F}_q^n$ with trivial hull, $\text{AUT-Gen} \leq_p \text{GAP-Gen}$

Proof

Just compute $\text{Adj}(\mathbf{G})$ and solve GAP-Gen!

From AUT-Gen to GAP-Gen (trivial hull)

Theorem

For a code $\mathcal{C} \subseteq \mathbb{F}_q^n$ with trivial hull, $\text{AUT-Gen} \leq_p \text{GAP-Gen}$

Proof

Just compute $\text{Adj}(\mathbf{G})$ and solve GAP-Gen!

Example over $\mathbb{F}_5$, $k = 3$, $n = 6$

4		3	3		
1	2	4	4	4	
2	2	4	2	1	4

From AUT-Gen to GAP-Gen (trivial hull)

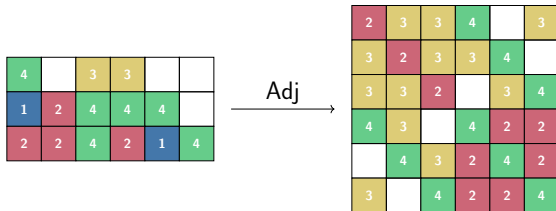
Theorem

For a code $\mathcal{C} \subseteq \mathbb{F}_q^n$ with trivial hull, $\text{AUT-Gen} \leq_p \text{GAP-Gen}$

Proof

Just compute $\text{Adj}(\mathbf{G})$ and solve GAP-Gen!

Example over $\mathbb{F}_5$, $k = 3$, $n = 6$



From AUT-Gen to GAP-Gen (trivial hull)

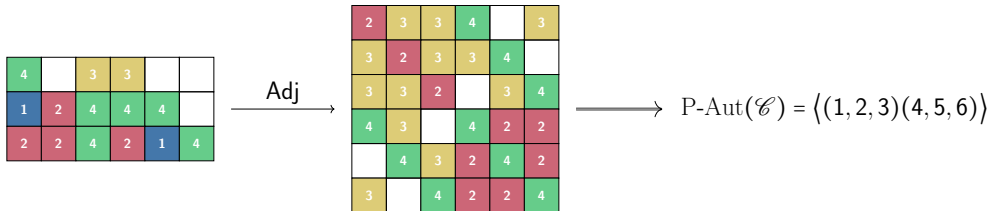
Theorem

For a code $\mathcal{C} \subseteq \mathbb{F}_q^n$ with trivial hull, $\text{AUT-Gen} \leq_p \text{GAP-Gen}$

Proof

Just compute $\text{Adj}(\mathbf{G})$ and solve GAP-Gen!

Example over $\mathbb{F}_5$, $k = 3$, $n = 6$



From PEP to GIP (non trivial hull)

Theorem: If $J \subseteq \{1, \dots, n\}$, $|J| = h$ is an information set for $\mathcal{H}(\mathcal{A})$, then shortening $\mathcal{A}$ on J gives a code with trivial hull.

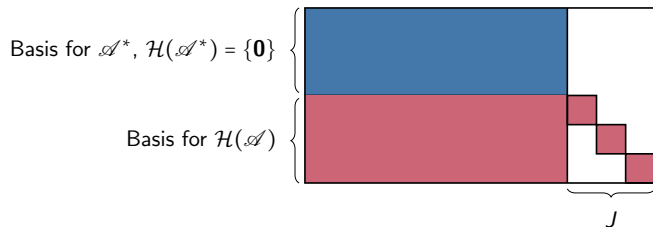
From PEP to GIP (non trivial hull)

Theorem: If $J \subseteq \{1, \dots, n\}$, $|J| = h$ is an information set for $\mathcal{H}(\mathcal{A})$, then shortening $\mathcal{A}$ on J gives a code with trivial hull.

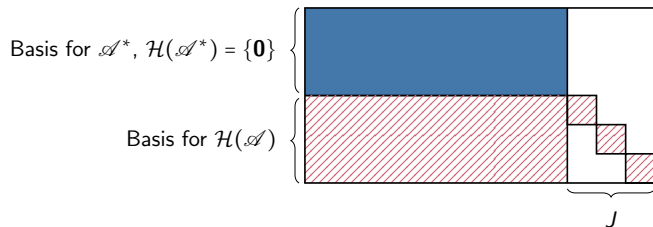
Sketch of reduction (Bardet, Otmani, Saeed-Taha, 2019):

- 1) Choose J : information set for the hull
- 2) Enumerate its image U
- 3) For each (J, U) , shorten the codes
- 4) Solve PEP on shortened codes
- 5) For each solution, enumerate permutations from J to U

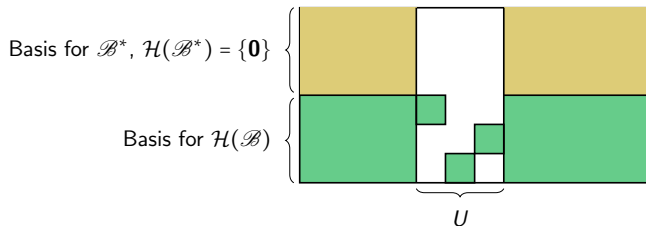
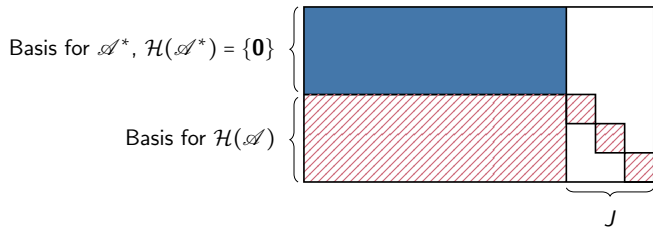
From PEP to GIP (non trivial hull)



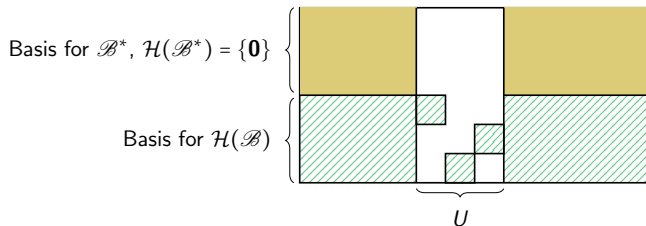
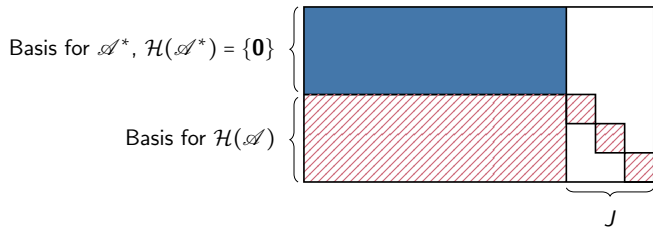
From PEP to GIP (non trivial hull)



From PEP to GIP (non trivial hull)



From PEP to GIP (non trivial hull)



From AUT-Gen to GAP-Gen: is it immediate?



Intuition: Shorten $\mathcal{C}$, find automorphisms

From AUT-Gen to GAP-Gen: is it immediate?



Intuition: Shorten $\mathcal{C}$, find automorphisms



Caution: shortening (or puncturing) can distort the automorphism group!

From AUT-Gen to GAP-Gen: is it immediate?



Intuition: Shorten $\mathcal{C}$, find automorphisms



Caution: shortening (or puncturing) can distort the automorphism group!

Example: $q = 29$, $n = 6$, $k = 3$, $h = 1$

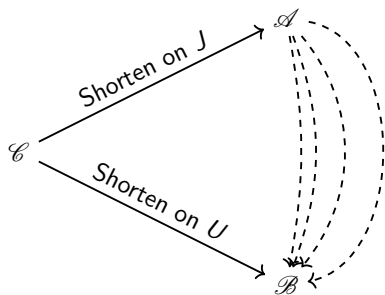
$$\mathbf{G} = \begin{pmatrix} 19 & 18 & 14 & 22 & 10 & 23 \\ 14 & 13 & 10 & 27 & 3 & 0 \\ 18 & 27 & 4 & 25 & 2 & 9 \end{pmatrix}, \quad \text{P-Aut}(\mathcal{C}) = \langle (1, 5, 4)(2, 3, 6) \rangle$$

Every shortening on one position yields a code with only trivial automorphisms.

From AUT-Gen to GAP-Gen: the main idea



To find an automorphism sending $J \subseteq \{1, \dots, n\}$ to $U \subseteq \{1, \dots, n\}$, map into an isomorphism problem!

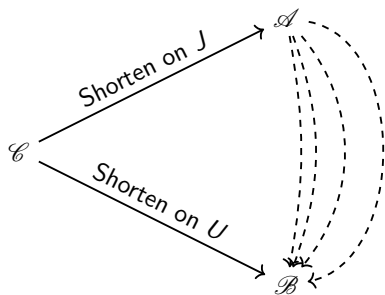


Permutations $\pi \in S_{n-h} : \pi(J) = U$

From AUT-Gen to GAP-Gen: the main idea



To find an automorphism sending $J \subseteq \{1, \dots, n\}$ to $U \subseteq \{1, \dots, n\}$, map into an isomorphism problem!



Permutations $\pi \in S_{n-h} : \pi(J) = U$



To find all isomorphisms from $\mathcal{A}$ to $\mathcal{B}$, map into an automorphism problem!

From AUT-Gen to GAP-Gen: the main ideas

Theorem

Let $\varphi \in \text{P-Aut}(\mathcal{C})$, $J, U \subseteq \{1, \dots, n\}$ of size h and such that $\varphi(J) = U$.

Let $\mathcal{A}, \mathcal{B}$ be the shortenings of $\mathcal{C}$ on J and U , i.e., $\mathcal{A} = \mathcal{C}_{-J}$ and $\mathcal{B} = \mathcal{C}_{-U}$.

Then, $\exists \pi \in S_{n-h}$ such that

- $\mathcal{B} = \pi(\mathcal{A})$
- $\pi(\{1, \dots, n\} \setminus J) = \varphi(\{1, \dots, n\} \setminus J)$

From AUT-Gen to GAP-Gen: the main ideas

Theorem

Let $\varphi \in \text{P-Aut}(\mathcal{C})$, $J, U \subseteq \{1, \dots, n\}$ of size h and such that $\varphi(J) = U$.

Let $\mathcal{A}, \mathcal{B}$ be the shortenings of $\mathcal{C}$ on J and U , i.e., $\mathcal{A} = \mathcal{C}_{-J}$ and $\mathcal{B} = \mathcal{C}_{-U}$.

Then, $\exists \pi \in S_{n-h}$ such that

- $\mathcal{B} = \pi(\mathcal{A})$
- $\pi(\{1, \dots, n\} \setminus J) = \varphi(\{1, \dots, n\} \setminus J)$

Observation: from $\pi \in \text{P-Aut}(\mathcal{A} \times \mathcal{B})$, we get $\varphi \in \text{P-Aut}(\mathcal{C})$ in time $O(h! \cdot n^\omega)$

From AUT-Gen to GAP-Gen (?)

Input: $\mathcal{C} \subseteq \mathbb{F}_q^n$ with $\text{Dim}(\mathcal{H}(\mathcal{C})) = h$

- 1) Set $\text{P-Aut}(\mathcal{C}) = \{id\}$ `\\Initialize P-Aut( $\mathcal{C}$ )`
- 2) Choose J : information set for $\mathcal{H}(\mathcal{C})$

From AUT-Gen to GAP-Gen (?)

Input: $\mathcal{C} \subseteq \mathbb{F}_q^n$ with $\text{Dim}(\mathcal{H}(\mathcal{C})) = h$

- 1) Set $\text{P-Aut}(\mathcal{C}) = \{id\}$ $\backslash\backslash$ Initialize $\text{P-Aut}(\mathcal{C})$
- 2) Choose J : information set for $\mathcal{H}(\mathcal{C})$
- 3) For $U \subseteq \{1, \dots, n\}$, $|U| = h$ $\backslash\backslash$ Brute force images of J
- 4) $S^* \leftarrow$ Solve AUT-Gen on $\mathcal{C}_{-J} \times \mathcal{C}_{-U}$ $\backslash\backslash$ Reduce to GAP-Gen

From AUT-Gen to GAP-Gen (?)

Input: $\mathcal{C} \subseteq \mathbb{F}_q^n$ with $\text{Dim}(\mathcal{H}(\mathcal{C})) = h$

- 1) Set $\text{P-Aut}(\mathcal{C}) = \{id\}$ $\backslash\backslash$ Initialize $\text{P-Aut}(\mathcal{C})$
- 2) Choose J : information set for $\mathcal{H}(\mathcal{C})$
- 3) For $U \subseteq \{1, \dots, n\}$, $|U| = h$ $\backslash\backslash$ Brute force images of J
- 4) $S^* \leftarrow$ Solve AUT-Gen on $\mathcal{C}_{-J} \times \mathcal{C}_{-U}$ $\backslash\backslash$ Reduce to GAP-Gen
- 5) For $\pi \in \langle S^* \rangle$, π swapping: $\backslash\backslash$ Only isomorphisms between shortened codes

From AUT-Gen to GAP-Gen (?)

Input: $\mathcal{C} \subseteq \mathbb{F}_q^n$ with $\text{Dim}(\mathcal{H}(\mathcal{C})) = h$

- 1) Set $\text{P-Aut}(\mathcal{C}) = \{id\}$ `\\Initialize P-Aut(C)`
- 2) Choose J : information set for $\mathcal{H}(\mathcal{C})$
- 3) For $U \subseteq \{1, \dots, n\}$, $|U| = h$ `\\Brute force images of J`
- 4) $S^* \leftarrow \text{Solve AUT-Gen on } \mathcal{C}_{-J} \times \mathcal{C}_{-U}$ `\\Reduce to GAP-Gen`
- 5) For $\pi \in \langle S^* \rangle$, π swapping: `\\Only isomorphisms between shortened codes`
- 6) For $\chi \in S_n$, $\chi(J) = U$: `\\Enumerate permutations from J to U`
- 7) $\varphi \leftarrow$ "combination" of π and χ

From AUT-Gen to GAP-Gen (?)

Input: $\mathcal{C} \subseteq \mathbb{F}_q^n$ with $\text{Dim}(\mathcal{H}(\mathcal{C})) = h$

- 1) Set $\text{P-Aut}(\mathcal{C}) = \{id\}$ `\\Initialize P-Aut(C)`
- 2) Choose J : information set for $\mathcal{H}(\mathcal{C})$
- 3) For $U \subseteq \{1, \dots, n\}$, $|U| = h$ `\\Brute force images of J`
- 4) $S^* \leftarrow$ Solve AUT-Gen on $\mathcal{C}_{-J} \times \mathcal{C}_{-U}$ `\\Reduce to GAP-Gen`
- 5) For $\pi \in \langle S^* \rangle$, π swapping: `\\Only isomorphisms between shortened codes`
- 6) For $\chi \in S_n$, $\chi(J) = U$: `\\Enumerate permutations from J to U`
- 7) $\varphi \leftarrow$ "combination" of π and χ
- 8) If φ is automorphism for $\mathcal{C}$: `\\This takes polynomial time`
- 9) $\text{P-Aut}(\mathcal{C}) \leftarrow \text{P-Aut}(\mathcal{C}) \cup \{\varphi\}$

From AUT-Gen to GAP-Gen (?)

Input: $\mathcal{C} \subseteq \mathbb{F}_q^n$ with $\text{Dim}(\mathcal{H}(\mathcal{C})) = h$

- 1) Set $\text{P-Aut}(\mathcal{C}) = \{id\}$ `\\Initialize P-Aut(C)`
- 2) Choose J : information set for $\mathcal{H}(\mathcal{C})$
- 3) For $U \subseteq \{1, \dots, n\}$, $|U| = h$ `\\Brute force images of J`
- 4) $S^* \leftarrow$ Solve AUT-Gen on $\mathcal{C}_{-J} \times \mathcal{C}_{-U}$ `\\Reduce to GAP-Gen`
- 5) For $\pi \in \langle S^* \rangle$, π swapping: `\\Only isomorphisms between shortened codes`
- 6) For $\chi \in S_n$, $\chi(J) = U$: `\\Enumerate permutations from J to U`
- 7) $\varphi \leftarrow$ "combination" of π and χ
- 8) If φ is automorphism for $\mathcal{C}$: `\\This takes polynomial time`
- 9) $\text{P-Aut}(\mathcal{C}) \leftarrow \text{P-Aut}(\mathcal{C}) \cup \{\varphi\}$
- 10) Return $\text{P-Aut}(\mathcal{C})$

From AUT-Gen to GAP-Gen: can something "go wrong"?

Presumably efficient if h is small, but...

From AUT-Gen to GAP-Gen: can something "go wrong"?

Presumably efficient if h is small, but...

Exponential time complexity if:

- $|\text{P-Aut}(\mathcal{C})|$ is exponentially large

From AUT-Gen to GAP-Gen: can something "go wrong"?

Presumably efficient if h is small, but...

Exponential time complexity if:

- $|\text{P-Aut}(\mathcal{C})|$ is exponentially large
- number of isomorphisms from $\mathcal{C}_J$ to $\mathcal{C}_U$ is exponentially large

From AUT-Gen to GAP-Gen: can something "go wrong"?

Presumably efficient if h is small, but...

Exponential time complexity if:

- $|\text{P-Aut}(\mathcal{C})|$ is exponentially large
- number of isomorphisms from $\mathcal{C}_J$ to $\mathcal{C}_U$ is exponentially large

Example: $q = 17$, $k = 2$, $n = 7$, $h = 1$

$$\mathbf{G} = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 4 & 5 & 14 & 16 & 7 & 8 & 4 \end{pmatrix}, \quad \text{P-Aut}(\mathcal{C}) = \{id\}$$

From AUT-Gen to GAP-Gen: can something "go wrong"?

Presumably efficient if h is small, but...

Exponential time complexity if:

- $|\text{P-Aut}(\mathcal{C})|$ is exponentially large
- number of isomorphisms from $\mathcal{C}_{-J}$ to $\mathcal{C}_{-U}$ is exponentially large

Example: $q = 17$, $k = 2$, $n = 7$, $h = 1$

$$\mathbf{G} = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 4 & 5 & 14 & 16 & 7 & 8 & 4 \end{pmatrix}, \quad \text{P-Aut}(\mathcal{C}) = \{id\}$$

For $J = U = \{1\} \implies \mathcal{C}_{-J} = \text{Span}\left(\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}\right)$: need to test $6! = |S_6|$ isomorphisms from $\mathcal{C}_{-J}$ to $\mathcal{C}_{-U}$

From AUT-Gen to GAP-Gen (finally!)

The reduction can be fixed with a few tricks:

- working with **generating sets**

From AUT-Gen to GAP-Gen (finally!)

The reduction can be fixed with a few tricks:

- working with **generating sets**
- de-hulling by **repeating coordinates** of hull information sets

From AUT-Gen to GAP-Gen (finally!)

The reduction can be fixed with a few tricks:

- working with **generating sets**
- de-hulling by **repeating coordinates** of hull information sets

$$G = \begin{bmatrix} g_1 & g_2 & g_3 & g_4 & g_5 \end{bmatrix}$$

$$A = \begin{bmatrix} g_1 & g_2 & g_3 & g_4 & g_5 & g_3 & g_3 & g_3 & g_3 & g_5 & g_5 & g_5 & g_5 & g_5 & g_5 & g_5 \end{bmatrix}$$

ℓ_1 replicas ℓ_2 replicas

$$B = \begin{bmatrix} g_1 & g_2 & g_3 & g_4 & g_5 & g_4 & g_4 & g_4 & g_4 & g_1 & g_1 & g_1 & g_1 & g_1 & g_1 & g_1 \end{bmatrix}$$

ℓ_1 replicas ℓ_2 replicas

Example over $\mathbb{F}_3$, with hull dimension $h = 2$
and code length $n = 5$.

We are using $J = (3, 5)$ and $U = (4, 1)$.

From AUT-Gen to GAP-Gen (finally!)

The reduction can be fixed with a few tricks:

- working with **generating sets**
- de-hulling by **repeating coordinates** of hull information sets

$$\mathbf{G} = \begin{bmatrix} \mathbf{g}_1 & \mathbf{g}_2 & \mathbf{g}_3 & \mathbf{g}_4 & \mathbf{g}_5 \end{bmatrix}$$

$$\mathbf{A} = \begin{bmatrix} \mathbf{g}_1 & \mathbf{g}_2 & \mathbf{g}_3 & \mathbf{g}_4 & \mathbf{g}_5 & \mathbf{g}_3 & \mathbf{g}_3 & \mathbf{g}_3 & \mathbf{g}_3 & \mathbf{g}_5 & \mathbf{g}_5 & \mathbf{g}_5 & \mathbf{g}_5 & \mathbf{g}_5 & \mathbf{g}_5 & \mathbf{g}_5 \end{bmatrix}$$

$\xleftarrow{\ell_1 \text{ replicas}} \quad \xrightarrow{\ell_2 \text{ replicas}}$

$$\mathbf{B} = \begin{bmatrix} \mathbf{g}_1 & \mathbf{g}_2 & \mathbf{g}_3 & \mathbf{g}_4 & \mathbf{g}_5 & \mathbf{g}_4 & \mathbf{g}_4 & \mathbf{g}_4 & \mathbf{g}_4 & \mathbf{g}_1 & \mathbf{g}_1 & \mathbf{g}_1 & \mathbf{g}_1 & \mathbf{g}_1 & \mathbf{g}_1 & \mathbf{g}_1 \end{bmatrix}$$

$\xleftarrow{\ell_1 \text{ replicas}} \quad \xrightarrow{\ell_2 \text{ replicas}}$

Example over $\mathbb{F}_3$, with hull dimension $h = 2$
and code length $n = 5$.

We are using $J = (3, 5)$ and $U = (4, 1)$.

Properties

- for constant p , $\mathbf{A}$ and $\mathbf{B}$ have length $\tilde{n} = O(n^2)$
- every isomorphism from $\mathbf{A}$ to $\mathbf{B}$ sends J to U
- there is an isomorphism from $\mathbf{A}$ to $\mathbf{B}$ if and only if an automorphism of $\mathbf{G}$ sends J to U

From AUT-Gen to GAP-Gen: running time

Theorem

Let $\mathcal{C} \subseteq \mathbb{F}_q^n$ with hull dimension h . Let $\tilde{n} = n + \sum_{i=1}^h \ell_i$. Let $T_{GAP}(m)$ be the worst-case complexity of an algorithm solving GAP-Gen on graphs with m nodes. Then, on input $\mathcal{C}$, AUT-Gen can be solved in worst case time

$$O\left(\binom{n}{h} \cdot h! \cdot n^\omega \cdot T_{GAP}(2\tilde{n})\right)$$

From AUT-Gen to GAP-Gen: running time

Theorem

Let $\mathcal{C} \subseteq \mathbb{F}_q^n$ with hull dimension h . Let $\tilde{n} = n + \sum_{i=1}^h \ell_i$. Let $T_{GAP}(m)$ be the worst-case complexity of an algorithm solving GAP-Gen on graphs with m nodes. Then, on input $\mathcal{C}$, AUT-Gen can be solved in worst case time

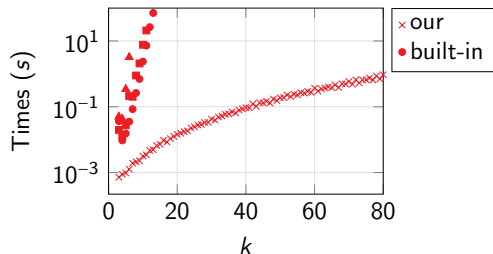
$$O\left(\binom{n}{h} \cdot h! \cdot n^\omega \cdot T_{GAP}(2\tilde{n})\right)$$

Running time:

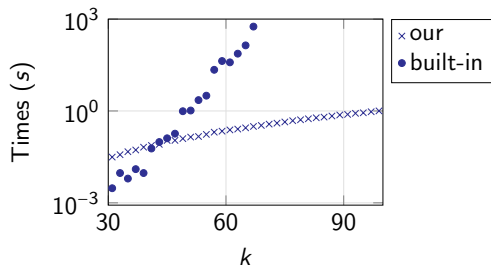
- For $h = O(\log(n)) \rightarrow$ **quasi-polynomial**
- For $h = \Theta(n) \rightarrow$ **exponential**

Practical solvers for AUT-Gen

The first reduction already gives a heuristically pretty good solver, unless one considers "pathological" codes.



(a) Sagemath, $q = 3$, $h = 0$



(b) Magma, $q = 2$, $h = 1$

Figure: Timings for $n = 2k$, random codes with a cyclic automorphism group.

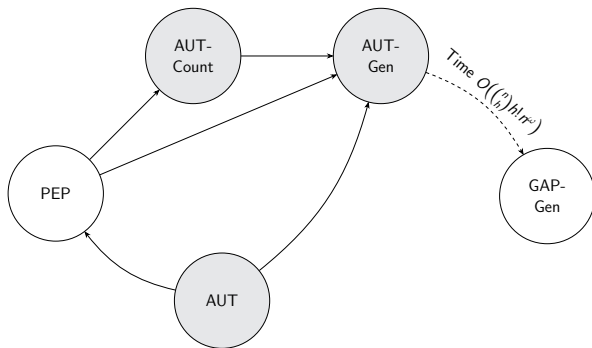
Our contributions

- ✓ Relations between automorphism group and the hull

Our contributions

✓ Relations between automorphism group and the hull

✓ Reductions to and from automorphisms problems



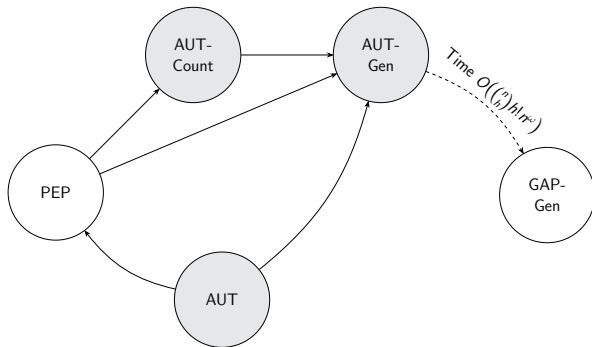
Our contributions

✓ Relations between automorphism group and the hull

✓ Reductions to and from automorphisms problems

✓ Relation with PEP:

- $\text{PEP} \leq_p \text{AUT-Gen}$
- $\text{AUT} \leq_p \text{PEP}$



Our contributions (II)

- ✓ Relation with GAP-Gen:
 - for trivial hull ($h = 0$), $\text{AUT-Gen} \leq_p \text{GAP-Gen}$
 - for $h \geq 1$, reduction to GAP-Gen in time $O\left(\binom{n}{h} h! n^\omega\right)$
 - heuristic solver: efficient when h is small

Our contributions (II)

✓ Relation with GAP-Gen:

- for trivial hull ($h = 0$), $\text{AUT-Gen} \leq_p \text{GAP-Gen}$
- for $h \geq 1$, reduction to GAP-Gen in time $O\left(\binom{n}{h} h! n^\omega\right)$
- heuristic solver: efficient when h is small

✓ Easy instances:

- worst case quasi-polynomial time for $h = O(\log(n))$
- heuristically easy for small h

Our contributions (II)

✓ Relation with GAP-Gen:

- for trivial hull ($h = 0$), $\text{AUT-Gen} \leq_p \text{GAP-Gen}$
- for $h \geq 1$, reduction to GAP-Gen in time $O\left(\binom{n}{h} h! n^\omega\right)$
- heuristic solver: efficient when h is small

✓ Easy instances:

- worst case quasi-polynomial time for $h = O(\log(n))$
- heuristically easy for small h

✗ Some reductions are missing: can they be defined?

Our contributions (II)

✓ Relation with GAP-Gen:

- for trivial hull ($h = 0$), $\text{AUT-Gen} \leq_p \text{GAP-Gen}$
- for $h \geq 1$, reduction to GAP-Gen in time $O\left(\binom{n}{h} h! n^\omega\right)$
- heuristic solver: efficient when h is small

✓ Easy instances:

- worst case quasi-polynomial time for $h = O(\log(n))$
- heuristically easy for small h

✗ Some reductions are missing: can they be defined?

✗ More general notions of code-equivalence? Other metrics?

Our contributions (II)

✓ Relation with GAP-Gen:

- for trivial hull ($h = 0$), $\text{AUT-Gen} \leq_p \text{GAP-Gen}$
- for $h \geq 1$, reduction to GAP-Gen in time $O\left(\binom{n}{h} h! n^\omega\right)$
- heuristic solver: efficient when h is small

✓ Easy instances:

- worst case quasi-polynomial time for $h = O(\log(n))$
- heuristically easy for small h

✗ Some reductions are missing: can they be defined?

✗ More general notions of code-equivalence? Other metrics?

✗ Can some of these results be generalized to "any" group action?

That's it :)

Thank you very much for the attention! Questions?

This work was supported by the Italian Fund for Applied Science (FISA 2022), project **“Quantum-safe cryptographic tools for the protection of national data and information technology assets”** (QSAFEIT) – No. FISA 2022-00618 (CUP I33C24000520001).

