

Key Attack on the ACDGV Matrix Encryption Scheme

Anmoal Porwal^{*}, Antonia Wachter-Zeh^{*}, Pierre Loidreau[†]

^{*}Technical University of Munich

[†]DGA, University of Rennes, CNRS, IRMAR – UMR 6625

25 September 2026
University of Rennes

Introduction

- ▶ A rank-metric public-key encryption scheme was proposed by Aragon, Couvreur, Dyseryn, Gaborit and Vinçotte (ACDGV) at Asiacrypt 2024

Introduction

- ▶ A rank-metric public-key encryption scheme was proposed by Aragon, Couvreur, Dyseryn, Gaborit and Vinçotte (ACDGV) at Asiacrypt 2024
- ▶ It uses matrix codes in the McEliece framework

Introduction

- ▶ A rank-metric public-key encryption scheme was proposed by Aragon, Couvreur, Dyseryn, Gaborit and Vinçotte (ACDGV) at Asiacrypt 2024
- ▶ It uses matrix codes in the McEliece framework

$$\begin{array}{c} \mathcal{C}_{\text{mat}} \subseteq \mathbb{F}_q^{m \times n} \\ \downarrow \text{trapdoor / hiding transform} \\ \mathcal{C}_{\text{pub}} \end{array}$$

Introduction

- ▶ A rank-metric public-key encryption scheme was proposed by Aragon, Couvreur, Dyseryn, Gaborit and Vinçotte (ACDGV) at Asiacrypt 2024
- ▶ It uses matrix codes in the McEliece framework

$$\begin{array}{c} \mathcal{C}_{\text{mat}} \subseteq \mathbb{F}_q^{m \times n} \\ \downarrow \text{trapdoor / hiding transform} \\ \mathcal{C}_{\text{pub}} \end{array}$$

- ▶ We present an exponential-time key-recovery attack that
 - outperforms the previous best distinguisher
 - breaks some proposed parameter sets

Introduction

- ▶ A rank-metric public-key encryption scheme was proposed by Aragon, Couvreur, Dyseryn, Gaborit and Vinçotte (ACDGV) at Asiacrypt 2024
- ▶ It uses matrix codes in the McEliece framework

$$\begin{array}{c} \mathcal{C}_{\text{mat}} \subseteq \mathbb{F}_q^{m \times n} \\ \downarrow \text{trapdoor / hiding transform} \\ \mathcal{C}_{\text{pub}} \end{array}$$

- ▶ We present an exponential-time key-recovery attack that
 - outperforms the previous best distinguisher
 - breaks some proposed parameter sets
- ▶ Note: a newer attack by Le (ePrint 2026/972) is much more efficient; we briefly discuss this at the end

The ACDGV Hiding Transform

- Public
- Secret

$$\mathcal{C}_{\text{mat}} = \langle \mathbf{A}_1, \dots, \mathbf{A}_{km} \rangle \subseteq \mathbb{F}_q^{m \times n}$$

The ACDGV Hiding Transform

• Public

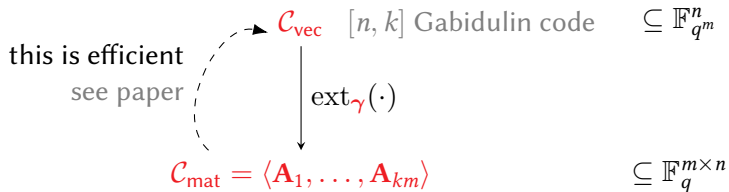
• Secret

$$\begin{array}{ccc} \mathcal{C}_{\text{vec}} & [n, k] \text{ Gabidulin code} & \subseteq \mathbb{F}_{q^m}^n \\ \downarrow \text{ext}_{\gamma}(\cdot) & & \\ \mathcal{C}_{\text{mat}} = \langle \mathbf{A}_1, \dots, \mathbf{A}_{km} \rangle & & \subseteq \mathbb{F}_q^{m \times n} \end{array}$$

The ACDGV Hiding Transform

• Public

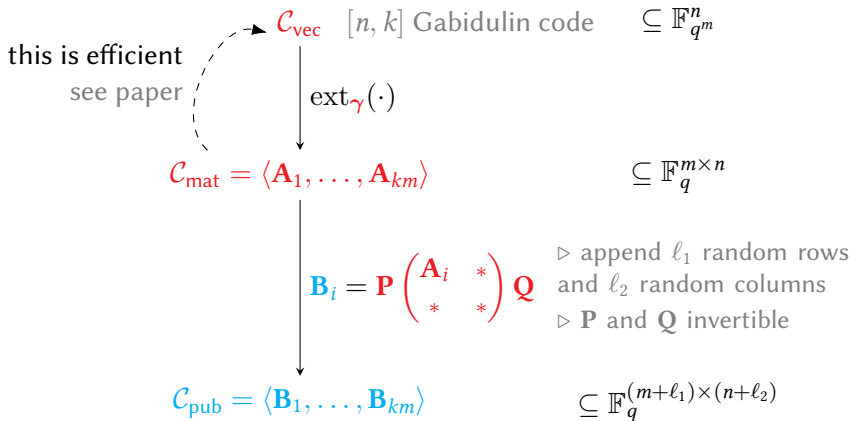
• Secret



The ACDGV Hiding Transform

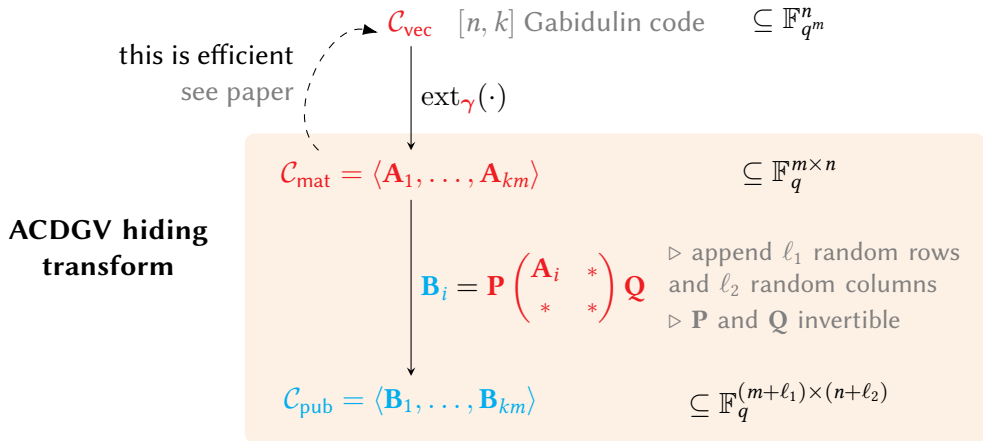
• Public

• Secret



The ACDGV Hiding Transform

- Public
- Secret



The ACDGV Hiding Transform

- Public
- Secret

this is efficient
see paper

$\mathcal{C}_{\text{vec}}$ $[n, k]$ Gabidulin code $\subseteq \mathbb{F}_{q^m}^n$

$\text{ext}_{\gamma}(\cdot)$

$$\mathcal{C}_{\text{mat}} = \langle \mathbf{A}_1, \dots, \mathbf{A}_{km} \rangle \subseteq \mathbb{F}_q^{m \times n}$$

$$\mathbf{B}_i = \mathbf{P} \begin{pmatrix} \mathbf{A}_i & * \\ * & * \end{pmatrix} \mathbf{Q}$$

- ▷ append ℓ_1 random rows and ℓ_2 random columns
- ▷ $\mathbf{P}$ and $\mathbf{Q}$ invertible

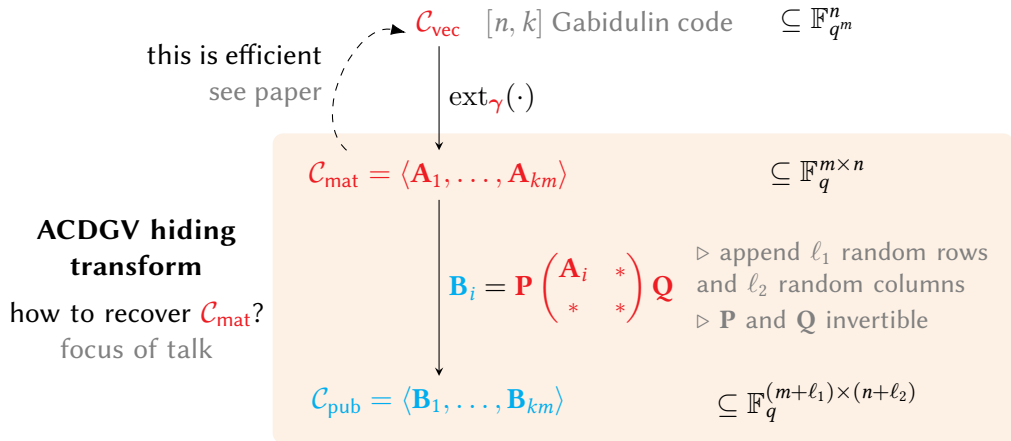
$$\mathcal{C}_{\text{pub}} = \langle \mathbf{B}_1, \dots, \mathbf{B}_{km} \rangle \subseteq \mathbb{F}_q^{(m+\ell_1) \times (n+\ell_2)}$$

ACDGV hiding transform

how to recover $\mathcal{C}_{\text{mat}}$?
focus of talk

The ACDGV Hiding Transform

- Public
- Secret



Note: our attack only exploits $\mathbb{F}_{q^m}$ -linearity and does not need Gabidulin structure

Solution Idea

- Public
- Secret

The *stabilizer algebra* of a matrix code $\mathcal{C} \subseteq \mathbb{F}_q^{m \times n}$ is the set

$$\{\mathbf{T} \in \mathbb{F}_q^{m \times m} \mid \mathbf{T} \cdot \mathcal{C} \subseteq \mathcal{C}\}$$

The *stabilizer algebra* of a matrix code $\mathcal{C} \subseteq \mathbb{F}_q^{m \times n}$ is the set

$$\{\mathbf{T} \in \mathbb{F}_q^{m \times m} \mid \mathbf{T} \cdot \mathcal{C} \subseteq \mathcal{C}\}$$

If $\mathcal{C}$ is a random matrix code $\implies$ stabilizer algebra has dimension 1 w.h.p.
since w.h.p. it equals $\{\lambda \mathbf{I} \mid \lambda \in \mathbb{F}_q\}$

The *stabilizer algebra* of a matrix code $\mathcal{C} \subseteq \mathbb{F}_q^{m \times n}$ is the set

$$\{\mathbf{T} \in \mathbb{F}_q^{m \times m} \mid \mathbf{T} \cdot \mathcal{C} \subseteq \mathcal{C}\}$$

If $\mathcal{C}$ is a random matrix code $\implies$ stabilizer algebra has dimension 1 w.h.p.
since w.h.p. it equals $\{\lambda \mathbf{I} \mid \lambda \in \mathbb{F}_q\}$

If $\mathcal{C} = \mathcal{C}_{\text{mat}}$ $\implies$ stabilizer algebra has dimension m

The *stabilizer algebra* of a matrix code $\mathcal{C} \subseteq \mathbb{F}_q^{m \times n}$ is the set

$$\{\mathbf{T} \in \mathbb{F}_q^{m \times m} \mid \mathbf{T} \cdot \mathcal{C} \subseteq \mathcal{C}\}$$

If $\mathcal{C}$ is a random matrix code $\implies$ stabilizer algebra has dimension 1 w.h.p.
since w.h.p. it equals $\{\lambda \mathbf{I} \mid \lambda \in \mathbb{F}_q\}$

If $\mathcal{C} = \mathcal{C}_{\text{mat}}$ $\implies$ stabilizer algebra has dimension m

We will exploit this property of $\mathcal{C}_{\text{mat}}$

Solution Idea

- Public
- Secret

$$\begin{array}{ccc} \overset{\mathcal{C}_{\text{mat}}}{\langle \mathbf{A}_1, \dots, \mathbf{A}_{km} \rangle} & \xrightarrow{\mathbf{B}_i = \overset{\text{Public}}{\mathbf{P}} \begin{pmatrix} \mathbf{A}_i & * \\ * & * \end{pmatrix} \underset{\text{Secret}}{\mathbf{Q}}} & \overset{\mathcal{C}_{\text{pub}}}{\langle \mathbf{B}_1, \dots, \mathbf{B}_{km} \rangle} \end{array}$$

Solution Idea

- Public
- Secret

$$\begin{array}{ccc} \overset{\mathcal{C}_{\text{mat}}}{\langle \mathbf{A}_1, \dots, \mathbf{A}_{km} \rangle} & \xrightarrow{\mathbf{B}_i = \mathbf{P} \begin{pmatrix} \mathbf{A}_i & * \\ * & * \end{pmatrix} \mathbf{Q}} & \overset{\mathcal{C}_{\text{pub}}}{\langle \mathbf{B}_1, \dots, \mathbf{B}_{km} \rangle} \end{array}$$

$$\dim \{ \mathbf{T} \mid \mathbf{T} \cdot \mathcal{C}_{\text{mat}} \subseteq \mathcal{C}_{\text{mat}} \} = m$$

Solution Idea

- Public
- Secret

$$\begin{array}{ccc} \overset{\mathcal{C}_{\text{mat}}}{\langle \mathbf{A}_1, \dots, \mathbf{A}_{km} \rangle} & \xrightarrow{\mathbf{B}_i = \mathbf{P} \begin{pmatrix} \mathbf{A}_i & * \\ * & * \end{pmatrix} \mathbf{Q}} & \overset{\mathcal{C}_{\text{pub}}}{\langle \mathbf{B}_1, \dots, \mathbf{B}_{km} \rangle} \end{array}$$

$$\dim \{ \mathbf{T} \mid \mathbf{T} \cdot \mathcal{C}_{\text{mat}} \subseteq \mathcal{C}_{\text{mat}} \} = m$$

What about $\mathcal{C}_{\text{pub}}$?

$$\dim \{ \mathbf{F} \mid \mathbf{F} \cdot \mathcal{C}_{\text{pub}} \subseteq \mathcal{C}_{\text{pub}} \} =$$

Solution Idea

- Public
- Secret

$$\begin{array}{ccc} \overset{\mathcal{C}_{\text{mat}}}{\langle \mathbf{A}_1, \dots, \mathbf{A}_{km} \rangle} & \xrightarrow{\mathbf{B}_i = \mathbf{P} \begin{pmatrix} \mathbf{A}_i & * \\ * & * \end{pmatrix} \mathbf{Q}} & \overset{\mathcal{C}_{\text{pub}}}{\langle \mathbf{B}_1, \dots, \mathbf{B}_{km} \rangle} \end{array}$$

$$\dim \{ \mathbf{T} \mid \mathbf{T} \cdot \mathcal{C}_{\text{mat}} \subseteq \mathcal{C}_{\text{mat}} \} = m$$

What about $\mathcal{C}_{\text{pub}}$?

$$\dim \{ \mathbf{F} \mid \mathbf{F} \cdot \mathcal{C}_{\text{pub}} \subseteq \mathcal{C}_{\text{pub}} \} = 1 \quad \text{☹}$$

Solution Idea

- Public
- Secret

$$\begin{array}{ccc} \textcolor{red}{\mathcal{C}_{\text{mat}}} & & \textcolor{blue}{\mathcal{C}_{\text{pub}}} \\ \langle \textcolor{red}{\mathbf{A}}_1, \dots, \textcolor{red}{\mathbf{A}}_{km} \rangle & \xrightarrow{\textcolor{blue}{\mathbf{B}}_i = \textcolor{red}{\mathbf{P}} \begin{pmatrix} \textcolor{blue}{\mathbf{A}}_i & * \\ * & * \end{pmatrix} \textcolor{red}{\mathbf{Q}}} & \langle \textcolor{blue}{\mathbf{B}}_1, \dots, \textcolor{blue}{\mathbf{B}}_{km} \rangle \end{array}$$

$$\dim \{ \mathbf{T} \mid \mathbf{T} \cdot \textcolor{red}{\mathcal{C}_{\text{mat}}} \subseteq \textcolor{red}{\mathcal{C}_{\text{mat}}} \} = m$$

What about $\textcolor{blue}{\mathcal{C}_{\text{pub}}}$?

$$\dim \{ \mathbf{F} \mid \mathbf{F} \cdot \textcolor{blue}{\mathcal{C}_{\text{pub}}} \subseteq \textcolor{blue}{\mathcal{C}_{\text{pub}}} \} = 1 \quad \text{☹}$$

► Suppose $\mathbf{F} = \textcolor{red}{\mathbf{P}} \begin{pmatrix} \mathbf{T} & \mathbf{0} \\ * & * \end{pmatrix} \textcolor{red}{\mathbf{P}}^{-1}$

Solution Idea

- Public
- Secret

$$\langle \overset{\mathcal{C}_{\text{mat}}}{\mathbf{A}_1, \dots, \mathbf{A}_{km}} \rangle \xrightarrow{\mathbf{B}_i = \mathbf{P} \begin{pmatrix} \mathbf{A}_i & * \\ * & * \end{pmatrix} \mathbf{Q}} \langle \overset{\mathcal{C}_{\text{pub}}}{\mathbf{B}_1, \dots, \mathbf{B}_{km}} \rangle$$

$$\dim \{ \mathbf{T} \mid \mathbf{T} \cdot \mathcal{C}_{\text{mat}} \subseteq \mathcal{C}_{\text{mat}} \} = m$$

What about $\mathcal{C}_{\text{pub}}$?

$$\dim \{ \mathbf{F} \mid \mathbf{F} \cdot \mathcal{C}_{\text{pub}} \subseteq \mathcal{C}_{\text{pub}} \} = 1 \quad \text{☹}$$

► Suppose $\mathbf{F} = \mathbf{P} \begin{pmatrix} \mathbf{T} & \mathbf{0} \\ * & * \end{pmatrix} \mathbf{P}^{-1}$

$$\text{Then } \mathbf{F} \mathbf{B}_i = \mathbf{P} \begin{pmatrix} \mathbf{T} \mathbf{A}_i & * \\ * & * \end{pmatrix} \mathbf{Q}$$

Solution Idea

- Public
- Secret

$$\langle \overset{\mathcal{C}_{\text{mat}}}{\mathbf{A}_1, \dots, \mathbf{A}_{km}} \rangle \xrightarrow{\mathbf{B}_i = \mathbf{P} \begin{pmatrix} \mathbf{A}_i & * \\ * & * \end{pmatrix} \mathbf{Q}} \langle \overset{\mathcal{C}_{\text{pub}}}{\mathbf{B}_1, \dots, \mathbf{B}_{km}} \rangle$$

$$\dim \{ \mathbf{T} \mid \mathbf{T} \cdot \mathcal{C}_{\text{mat}} \subseteq \mathcal{C}_{\text{mat}} \} = m$$

What about $\mathcal{C}_{\text{pub}}$?

$$\dim \{ \mathbf{F} \mid \mathbf{F} \cdot \mathcal{C}_{\text{pub}} \subseteq \mathcal{C}_{\text{pub}} \} = 1 \quad \text{☹}$$

► Suppose $\mathbf{F} = \mathbf{P} \begin{pmatrix} \mathbf{T} & \mathbf{0} \\ * & * \end{pmatrix} \mathbf{P}^{-1}$

Then $\mathbf{F} \mathbf{B}_i = \mathbf{P} \begin{pmatrix} \mathbf{T} \mathbf{A}_i & * \\ * & * \end{pmatrix} \mathbf{Q}$

Then for some $\mathbf{C}_i \in \mathcal{C}_{\text{pub}}$, we have $\mathbf{F} \mathbf{B}_i - \mathbf{C}_i = \mathbf{P} \begin{pmatrix} \mathbf{0} & * \\ * & * \end{pmatrix} \mathbf{Q}$

Solution Idea

- Public
- Secret

Strategy: Find matrix $\mathbf{F}$ and matrices $\mathbf{C}_1, \dots, \mathbf{C}_{km} \in \mathcal{C}_{\text{pub}}$ such that

$$\mathbf{F}\mathbf{B}_i - \mathbf{C}_i = \mathbf{P} \begin{pmatrix} \mathbf{0} & * \\ * & * \end{pmatrix} \mathbf{Q} \quad \text{for all } i = 1, \dots, km$$

Solution Idea

- Public
- Secret

Strategy: Find matrix $\mathbf{F}$ and matrices $\mathbf{C}_1, \dots, \mathbf{C}_{km} \in \mathcal{C}_{\text{pub}}$ such that

$$\mathbf{F}\mathbf{B}_i - \mathbf{C}_i = \mathbf{P} \begin{pmatrix} \mathbf{0} & * \\ * & * \end{pmatrix} \mathbf{Q} \quad \text{for all } i = 1, \dots, km$$

Q1 How to solve for $\mathbf{F}$ and $\mathbf{C}_1, \dots, \mathbf{C}_{km}$?

Solution Idea

- Public
- Secret

Strategy: Find matrix $\mathbf{F}$ and matrices $\mathbf{C}_1, \dots, \mathbf{C}_{km} \in \mathcal{C}_{\text{pub}}$ such that

$$\mathbf{F}\mathbf{B}_i - \mathbf{C}_i = \mathbf{P} \begin{pmatrix} \mathbf{0} & * \\ * & * \end{pmatrix} \mathbf{Q} \quad \text{for all } i = 1, \dots, km$$

Q1 How to solve for $\mathbf{F}$ and $\mathbf{C}_1, \dots, \mathbf{C}_{km}$?

Q2 How does this help recover $\mathcal{C}_{\text{mat}}$?

Q1 How to solve for $\mathbf{F}$ and $\mathbf{C}_1, \dots, \mathbf{C}_{km}$?

- Public
- Secret

Find matrix $\mathbf{F}$ and matrices $\mathbf{C}_1, \dots, \mathbf{C}_{km} \in \mathcal{C}_{\text{pub}}$ such that

$$\mathbf{F} \mathbf{B}_i - \mathbf{C}_i = \mathbf{P} \begin{pmatrix} \mathbf{0} & * \\ * & * \end{pmatrix} \mathbf{Q}$$

for all $i = 1, \dots, km$

Q1 How to solve for $\mathbf{F}$ and $\mu_{i,j}$'s?

• Public

• Secret

Find matrix $\mathbf{F}$ and scalars $\mu_{i,j}$ such that

$$\mathbf{F} \mathbf{B}_1 - \sum_j \mu_{1,j} \mathbf{B}_j = \mathbf{P} \begin{pmatrix} \mathbf{0} & * \\ * & * \end{pmatrix} \mathbf{Q}$$

$\vdots$

$$\mathbf{F} \mathbf{B}_{km} - \sum_j \mu_{km,j} \mathbf{B}_j = \mathbf{P} \begin{pmatrix} \mathbf{0} & * \\ * & * \end{pmatrix} \mathbf{Q}$$

Q1 How to solve for $\mathbf{F}$ and $\mu_{i,j}$'s?

• Public

• Secret

Find matrix $\mathbf{F}$ and scalars $\mu_{i,j}$ such that

$$\mathbf{F} \mathbf{B}_1 - \sum_j \mu_{1,j} \mathbf{B}_j = \mathbf{P} \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ * & * \end{pmatrix} \mathbf{Q} + \mathbf{P} \begin{pmatrix} \mathbf{0} & * \\ \mathbf{0} & * \end{pmatrix} \mathbf{Q}$$

$\vdots$

$$\mathbf{F} \mathbf{B}_{km} - \sum_j \mu_{km,j} \mathbf{B}_j = \mathbf{P} \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ * & * \end{pmatrix} \mathbf{Q} + \mathbf{P} \begin{pmatrix} \mathbf{0} & * \\ \mathbf{0} & * \end{pmatrix} \mathbf{Q}$$

Q1 How to solve for $\mathbf{F}$ and $\mu_{i,j}$'s?

• Public

• Secret

Find matrix $\mathbf{F}$ and scalars $\mu_{i,j}$ such that

$$\mathbf{F} \mathbf{B}_1 - \sum_j \mu_{1,j} \mathbf{B}_j = \mathbf{P} \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ * & * \end{pmatrix} \mathbf{Q} + \mathbf{P} \begin{pmatrix} \mathbf{0} & * \\ \mathbf{0} & * \end{pmatrix} \mathbf{Q}$$

$\vdots$

$$\mathbf{F} \mathbf{B}_{km} - \sum_j \mu_{km,j} \mathbf{B}_j = \mathbf{P} \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ * & * \end{pmatrix} \mathbf{Q} + \mathbf{P} \begin{pmatrix} \mathbf{0} & * \\ \mathbf{0} & * \end{pmatrix} \mathbf{Q}$$

The $\mathbf{P} \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ * & * \end{pmatrix} \mathbf{Q}$ matrices share same column space

The $\mathbf{P} \begin{pmatrix} \mathbf{0} & * \\ \mathbf{0} & * \end{pmatrix} \mathbf{Q}$ matrices share same row space

Q1 How to solve for $\mathbf{F}$ and $\mu_{i,j}$'s?

• Public

• Secret

Find matrix $\mathbf{F}$ and scalars $\mu_{i,j}$ such that

$$\mathbf{F} \mathbf{B}_1 - \sum_j \mu_{1,j} \mathbf{B}_j = \mathbf{P} \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ * & * \end{pmatrix} \mathbf{Q} + \mathbf{P} \begin{pmatrix} \mathbf{0} & * \\ \mathbf{0} & * \end{pmatrix} \mathbf{Q}$$

$\vdots$

$$\mathbf{F} \mathbf{B}_{km} - \sum_j \mu_{km,j} \mathbf{B}_j = \mathbf{P} \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ * & * \end{pmatrix} \mathbf{Q} + \mathbf{P} \begin{pmatrix} \mathbf{0} & * \\ \mathbf{0} & * \end{pmatrix} \mathbf{Q}$$

The $\mathbf{P} \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ * & * \end{pmatrix} \mathbf{Q}$ matrices share same column space $\implies$ share same left null space

The $\mathbf{P} \begin{pmatrix} \mathbf{0} & * \\ \mathbf{0} & * \end{pmatrix} \mathbf{Q}$ matrices share same row space $\implies$ share same right null space

Q1 How to solve for $\mathbf{F}$ and $\mu_{i,j}$'s?

• Public

• Secret

Algorithm: Guess vectors $\mathbf{v}$ and $\mathbf{w}$ in these left and right null spaces

$$\begin{aligned}\mathbf{F} \mathbf{B}_1 - \sum_j \mu_{1,j} \mathbf{B}_j &= \mathbf{P} \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ * & * \end{pmatrix} \mathbf{Q} + \mathbf{P} \begin{pmatrix} \mathbf{0} & * \\ \mathbf{0} & * \end{pmatrix} \mathbf{Q} \\ &\vdots \\ \mathbf{F} \mathbf{B}_{km} - \sum_j \mu_{km,j} \mathbf{B}_j &= \mathbf{P} \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ * & * \end{pmatrix} \mathbf{Q} + \mathbf{P} \begin{pmatrix} \mathbf{0} & * \\ \mathbf{0} & * \end{pmatrix} \mathbf{Q}\end{aligned}$$

Q1 How to solve for $\mathbf{F}$ and $\mu_{i,j}$'s?

- Public
- Secret

Algorithm: Guess vectors $\mathbf{v}$ and $\mathbf{w}$ in these left and right null spaces

$$\begin{aligned} \mathbf{v}^\top \left(\mathbf{F} \mathbf{B}_1 - \sum_j \mu_{1,j} \mathbf{B}_j \right) \mathbf{w} &= \mathbf{v}^\top \left(\mathbf{P} \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ * & * \end{pmatrix} \mathbf{Q} + \mathbf{P} \begin{pmatrix} \mathbf{0} & * \\ \mathbf{0} & * \end{pmatrix} \mathbf{Q} \right) \mathbf{w} \\ &\vdots \\ \mathbf{v}^\top \left(\mathbf{F} \mathbf{B}_{km} - \sum_j \mu_{km,j} \mathbf{B}_j \right) \mathbf{w} &= \mathbf{v}^\top \left(\mathbf{P} \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ * & * \end{pmatrix} \mathbf{Q} + \mathbf{P} \begin{pmatrix} \mathbf{0} & * \\ \mathbf{0} & * \end{pmatrix} \mathbf{Q} \right) \mathbf{w} \end{aligned}$$

Q1 How to solve for $\mathbf{F}$ and $\mu_{i,j}$'s?

• Public

• Secret

Algorithm: Guess vectors $\mathbf{v}$ and $\mathbf{w}$ in these left and right null spaces

$$\begin{aligned} \mathbf{v}^\top \left(\mathbf{F} \mathbf{B}_1 - \sum_j \mu_{1,j} \mathbf{B}_j \right) \mathbf{w} &= \mathbf{v}^\top \left(\mathbf{P} \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ * & * \end{pmatrix} \mathbf{Q} + \mathbf{P} \begin{pmatrix} \mathbf{0} & * \\ \mathbf{0} & * \end{pmatrix} \mathbf{Q} \right) \mathbf{w} = 0 \\ &\vdots \\ \mathbf{v}^\top \left(\mathbf{F} \mathbf{B}_{km} - \sum_j \mu_{km,j} \mathbf{B}_j \right) \mathbf{w} &= \mathbf{v}^\top \left(\mathbf{P} \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ * & * \end{pmatrix} \mathbf{Q} + \mathbf{P} \begin{pmatrix} \mathbf{0} & * \\ \mathbf{0} & * \end{pmatrix} \mathbf{Q} \right) \mathbf{w} = 0 \end{aligned}$$

Q1 How to solve for $\mathbf{F}$ and $\mu_{i,j}$'s?

• Public

• Secret

Algorithm: Guess vectors $\mathbf{v}$ and $\mathbf{w}$ in these left and right null spaces

$$\begin{aligned} \mathbf{v}^\top \left(\mathbf{F} \mathbf{B}_1 - \sum_j \mu_{1,j} \mathbf{B}_j \right) \mathbf{w} &= \mathbf{v}^\top \left(\mathbf{P} \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ * & * \end{pmatrix} \mathbf{Q} + \mathbf{P} \begin{pmatrix} \mathbf{0} & * \\ \mathbf{0} & * \end{pmatrix} \mathbf{Q} \right) \mathbf{w} = 0 \\ &\vdots \\ \mathbf{v}^\top \left(\mathbf{F} \mathbf{B}_{km} - \sum_j \mu_{km,j} \mathbf{B}_j \right) \mathbf{w} &= \mathbf{v}^\top \left(\mathbf{P} \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ * & * \end{pmatrix} \mathbf{Q} + \mathbf{P} \begin{pmatrix} \mathbf{0} & * \\ \mathbf{0} & * \end{pmatrix} \mathbf{Q} \right) \mathbf{w} = 0 \end{aligned}$$

With enough guesses, we get enough equations to solve for $\mathbf{F}$ and the $\mu_{i,j}$'s

Q1 How to solve for $\mathbf{F}$ and $\mu_{i,j}$'s?

• Public

• Secret

Algorithm: Guess vectors $\mathbf{v}$ and $\mathbf{w}$ in these left and right null spaces

$$\begin{aligned} \mathbf{v}^\top \left(\mathbf{F} \mathbf{B}_1 - \sum_j \mu_{1,j} \mathbf{B}_j \right) \mathbf{w} &= \mathbf{v}^\top \left(\mathbf{P} \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ * & * \end{pmatrix} \mathbf{Q} + \mathbf{P} \begin{pmatrix} \mathbf{0} & * \\ \mathbf{0} & * \end{pmatrix} \mathbf{Q} \right) \mathbf{w} = 0 \\ &\vdots \\ \mathbf{v}^\top \left(\mathbf{F} \mathbf{B}_{km} - \sum_j \mu_{km,j} \mathbf{B}_j \right) \mathbf{w} &= \mathbf{v}^\top \left(\mathbf{P} \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ * & * \end{pmatrix} \mathbf{Q} + \mathbf{P} \begin{pmatrix} \mathbf{0} & * \\ \mathbf{0} & * \end{pmatrix} \mathbf{Q} \right) \mathbf{w} = 0 \end{aligned}$$

With enough guesses, we get enough equations to solve for $\mathbf{F}$ and the $\mu_{i,j}$'s

This can be seen as an adaptation of the kernel search solver for MinRank

Q1 How to solve for $\mathbf{F}$ and $\mu_{i,j}$'s?

• Public

• Secret

Algorithm: Guess vectors $\mathbf{v}$ and $\mathbf{w}$ in these left and right null spaces

$$\begin{aligned} \mathbf{v}^\top \left(\mathbf{F} \mathbf{B}_1 - \sum_j \mu_{1,j} \mathbf{B}_j \right) \mathbf{w} &= \mathbf{v}^\top \left(\mathbf{P} \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ * & * \end{pmatrix} \mathbf{Q} + \mathbf{P} \begin{pmatrix} \mathbf{0} & * \\ \mathbf{0} & * \end{pmatrix} \mathbf{Q} \right) \mathbf{w} = 0 \\ &\vdots \\ \mathbf{v}^\top \left(\mathbf{F} \mathbf{B}_{km} - \sum_j \mu_{km,j} \mathbf{B}_j \right) \mathbf{w} &= \mathbf{v}^\top \left(\mathbf{P} \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ * & * \end{pmatrix} \mathbf{Q} + \mathbf{P} \begin{pmatrix} \mathbf{0} & * \\ \mathbf{0} & * \end{pmatrix} \mathbf{Q} \right) \mathbf{w} = 0 \end{aligned}$$

With enough guesses, we get enough equations to solve for $\mathbf{F}$ and the $\mu_{i,j}$'s

This can be seen as an adaptation of the kernel search solver for MinRank

A caveat: $\mathbf{F}$ has many spurious solutions... we will address this later

Q1 How to solve for $\mathbf{F}$ and $\mu_{i,j}$'s? *More details...*

• Public

• Secret

Suppose we guess b_1 left vectors $\mathbf{v}_1, \dots, \mathbf{v}_{b_1}$ and b_2 right vectors $\mathbf{w}_1, \dots, \mathbf{w}_{b_2}$

Q1 How to solve for $\mathbf{F}$ and $\mu_{i,j}$'s? *More details...*

• Public

• Secret

Suppose we guess b_1 left vectors $\mathbf{v}_1, \dots, \mathbf{v}_{b_1}$ and b_2 right vectors $\mathbf{w}_1, \dots, \mathbf{w}_{b_2}$

$$\left(\mathbf{F} \quad \mathbf{B}_i - \sum_{j=1}^{km} \mu_{i,j} \mathbf{B}_j \right) \quad \text{for } 1 \leq i \leq km$$

Q1 How to solve for $\mathbf{F}$ and $\mu_{i,j}$'s? *More details...*

• Public

• Secret

Suppose we guess b_1 left vectors $\mathbf{v}_1, \dots, \mathbf{v}_{b_1}$ and b_2 right vectors $\mathbf{w}_1, \dots, \mathbf{w}_{b_2}$

$$\begin{bmatrix} \mathbf{v}_1^\top \\ \vdots \\ \mathbf{v}_{b_1}^\top \end{bmatrix} \left(\mathbf{F} \quad \mathbf{B}_i - \sum_{j=1}^{km} \mu_{i,j} \mathbf{B}_j \right) \quad \text{for } 1 \leq i \leq km$$

Q1 How to solve for $\mathbf{F}$ and $\mu_{i,j}$'s? *More details...*

• Public

• Secret

Suppose we guess b_1 left vectors $\mathbf{v}_1, \dots, \mathbf{v}_{b_1}$ and b_2 right vectors $\mathbf{w}_1, \dots, \mathbf{w}_{b_2}$

$$\begin{bmatrix} \mathbf{v}_1^\top \\ \vdots \\ \mathbf{v}_{b_1}^\top \end{bmatrix} \left(\mathbf{F} \quad \mathbf{B}_i - \sum_{j=1}^{km} \mu_{i,j} \mathbf{B}_j \right) [\mathbf{w}_1 \quad \dots \quad \mathbf{w}_{b_2}] \quad \text{for } 1 \leq i \leq km$$

Q1 How to solve for $\mathbf{F}$ and $\mu_{i,j}$'s? *More details...*

• Public

• Secret

Suppose we guess b_1 left vectors $\mathbf{v}_1, \dots, \mathbf{v}_{b_1}$ and b_2 right vectors $\mathbf{w}_1, \dots, \mathbf{w}_{b_2}$

$$\begin{bmatrix} \mathbf{v}_1^\top \\ \vdots \\ \mathbf{v}_{b_1}^\top \end{bmatrix} \left(\mathbf{F} \quad \mathbf{B}_i - \sum_{j=1}^{km} \mu_{i,j} \mathbf{B}_j \right) \begin{bmatrix} \mathbf{w}_1 & \cdots & \mathbf{w}_{b_2} \end{bmatrix} = \mathbf{0} \quad \text{for } 1 \leq i \leq km$$

Q1 How to solve for $\mathbf{F}$ and $\mu_{i,j}$'s? *More details...*

• Public

• Secret

Suppose we guess b_1 left vectors $\mathbf{v}_1, \dots, \mathbf{v}_{b_1}$ and b_2 right vectors $\mathbf{w}_1, \dots, \mathbf{w}_{b_2}$

$$\begin{bmatrix} \mathbf{v}_1^T \\ \vdots \\ \mathbf{v}_{b_1}^T \end{bmatrix} \begin{pmatrix} \mathbf{F} & \mathbf{B}_i \\ \times \binom{m+\ell_1}{m+\ell_1} & \times \binom{m+\ell_1}{n+\ell_2} \end{pmatrix} - \sum_{j=1}^{km} \mu_{i,j} \mathbf{B}_j \begin{bmatrix} \mathbf{w}_1 & \cdots & \mathbf{w}_{b_2} \end{bmatrix} = \mathbf{0} \quad \text{for } 1 \leq i \leq km$$

Q1 How to solve for $\mathbf{F}$ and $\mu_{i,j}$'s? *More details...*

• Public

• Secret

Suppose we guess b_1 left vectors $\mathbf{v}_1, \dots, \mathbf{v}_{b_1}$ and b_2 right vectors $\mathbf{w}_1, \dots, \mathbf{w}_{b_2}$

$$\begin{bmatrix} \mathbf{v}_1^T \\ \vdots \\ \mathbf{v}_{b_1}^T \end{bmatrix} \begin{pmatrix} \mathbf{F} & \mathbf{B}_i \end{pmatrix} - \sum_{j=1}^{km} \mu_{i,j} \mathbf{B}_j \begin{bmatrix} \mathbf{w}_1 & \cdots & \mathbf{w}_{b_2} \end{bmatrix} = \mathbf{0} \quad \text{for } 1 \leq i \leq km$$

$$\# \text{equations} := b_1 b_2 km$$

Q1 How to solve for $\mathbf{F}$ and $\mu_{i,j}$'s? *More details...*

• Public

• Secret

Suppose we guess b_1 left vectors $\mathbf{v}_1, \dots, \mathbf{v}_{b_1}$ and b_2 right vectors $\mathbf{w}_1, \dots, \mathbf{w}_{b_2}$

$$\begin{bmatrix} \mathbf{v}_1^T \\ \vdots \\ \mathbf{v}_{b_1}^T \end{bmatrix} \begin{pmatrix} \mathbf{F} & \mathbf{B}_i \\ \times \binom{m+\ell_1}{m+\ell_1} & \times \binom{m+\ell_1}{n+\ell_2} \end{pmatrix} - \sum_{j=1}^{km} \mu_{i,j} \mathbf{B}_j \begin{bmatrix} \mathbf{w}_1 & \cdots & \mathbf{w}_{b_2} \end{bmatrix} = \mathbf{0} \quad \text{for } 1 \leq i \leq km$$

$$\# \text{equations} := b_1 b_2 km$$

$$\# \text{unknowns} := (m + \ell_1)^2 + (km)^2$$

Q1 How to solve for $\mathbf{F}$ and $\mu_{i,j}$'s? *More details...*

• Public

• Secret

Suppose we guess b_1 left vectors $\mathbf{v}_1, \dots, \mathbf{v}_{b_1}$ and b_2 right vectors $\mathbf{w}_1, \dots, \mathbf{w}_{b_2}$

$$\begin{bmatrix} \mathbf{v}_1^T \\ \vdots \\ \mathbf{v}_{b_1}^T \end{bmatrix} \begin{pmatrix} \begin{matrix} \times (m + \ell_1) \\ \mathbf{F} \end{matrix} & \begin{matrix} \times (m + \ell_1) \\ \mathbf{B}_i \end{matrix} & - & \begin{matrix} \times (m + \ell_1) \\ \sum_{j=1}^{km} \mu_{i,j} \mathbf{B}_j \end{matrix} \end{pmatrix} \begin{matrix} \times (n + \ell_2) \\ \begin{bmatrix} \mathbf{w}_1 & \cdots & \mathbf{w}_{b_2} \end{bmatrix} \end{matrix} = \begin{matrix} b_1 \times b_2 \\ \mathbf{0} \end{matrix} \quad \text{for } 1 \leq i \leq km$$

#equations := $b_1 b_2 km$

#unknowns := $(m + \ell_1)^2 + (km)^2$

#solution-space dimension := somewhat involved

Q1 How to solve for $\mathbf{F}$ and $\mu_{i,j}$'s? *More details...*

• Public

• Secret

Suppose we guess b_1 left vectors $\mathbf{v}_1, \dots, \mathbf{v}_{b_1}$ and b_2 right vectors $\mathbf{w}_1, \dots, \mathbf{w}_{b_2}$

$$\begin{bmatrix} \mathbf{v}_1^T \\ \vdots \\ \mathbf{v}_{b_1}^T \end{bmatrix} \begin{pmatrix} \mathbf{F} & \mathbf{B}_i - \sum_{j=1}^{km} \mu_{i,j} \mathbf{B}_j \end{pmatrix} \begin{bmatrix} \mathbf{w}_1 & \cdots & \mathbf{w}_{b_2} \end{bmatrix} = \mathbf{0} \quad \text{for } 1 \leq i \leq km$$

$$\# \text{equations} := b_1 b_2 km$$

$$\# \text{unknowns} := (m + \ell_1)^2 + (km)^2$$

$$\# \text{solution-space dimension} := \text{somewhat involved}$$

$$\text{We want} \quad \# \text{equations} \geq \# \text{unknowns} - \# \text{solution-space dimension} + 1 \quad (*)$$

Q1 How to solve for $\mathbf{F}$ and $\mu_{i,j}$'s? *More details...*

• Public

• Secret

Suppose we guess b_1 left vectors $\mathbf{v}_1, \dots, \mathbf{v}_{b_1}$ and b_2 right vectors $\mathbf{w}_1, \dots, \mathbf{w}_{b_2}$

$$\begin{bmatrix} \mathbf{v}_1^T \\ \vdots \\ \mathbf{v}_{b_1}^T \end{bmatrix} \begin{pmatrix} \mathbf{F} & \mathbf{B}_i - \sum_{j=1}^{km} \mu_{i,j} \mathbf{B}_j \end{pmatrix} \begin{bmatrix} \mathbf{w}_1 & \cdots & \mathbf{w}_{b_2} \end{bmatrix} = \mathbf{0} \quad \text{for } 1 \leq i \leq km$$

$$\# \text{equations} := b_1 b_2 km$$

$$\# \text{unknowns} := (m + \ell_1)^2 + (km)^2$$

$$\# \text{solution-space dimension} := \text{somewhat involved}$$

$$\text{We want} \quad \# \text{equations} \geq \# \text{unknowns} - \# \text{solution-space dimension} + 1 \quad (*)$$

$$\text{Probability of correctly guessing the } \mathbf{v}_a \text{'s and } \mathbf{w}_b \text{'s is } \frac{\binom{m}{b_1}_q}{\binom{m+\ell_1}{b_1}_q} \frac{\binom{n}{b_2}_q}{\binom{n+\ell_2}{b_2}_q}$$

Q1 How to solve for $\mathbf{F}$ and $\mu_{i,j}$'s? *More details...*

• Public

• Secret

Suppose we guess b_1 left vectors $\mathbf{v}_1, \dots, \mathbf{v}_{b_1}$ and b_2 right vectors $\mathbf{w}_1, \dots, \mathbf{w}_{b_2}$

$$\begin{bmatrix} \mathbf{v}_1^T \\ \vdots \\ \mathbf{v}_{b_1}^T \end{bmatrix} \begin{pmatrix} \mathbf{F} & \mathbf{B}_i \\ \times \binom{m+\ell_1}{m+\ell_1} & \times \binom{m+\ell_1}{n+\ell_2} \end{pmatrix} - \sum_{j=1}^{km} \mu_{i,j} \mathbf{B}_j \begin{bmatrix} \mathbf{w}_1 & \cdots & \mathbf{w}_{b_2} \end{bmatrix} = \mathbf{0} \quad \text{for } 1 \leq i \leq km$$

$$\# \text{equations} := b_1 b_2 km$$

$$\# \text{unknowns} := (m + \ell_1)^2 + (km)^2$$

$$\# \text{solution-space dimension} := \text{somewhat involved}$$

$$\text{We want} \quad \# \text{equations} \geq \# \text{unknowns} - \# \text{solution-space dimension} + 1 \quad (*)$$

$$\text{Probability of correctly guessing the } \mathbf{v}_a \text{'s and } \mathbf{w}_b \text{'s is } \frac{\binom{m}{b_1}_q}{\binom{m+\ell_1}{b_1}_q} \frac{\binom{n}{b_2}_q}{\binom{n+\ell_2}{b_2}_q} \approx q^{-(b_1 \ell_1 + b_2 \ell_2)}$$

Q1 How to solve for $\mathbf{F}$ and $\mu_{i,j}$'s? *More details...*

• Public

• Secret

Suppose we guess b_1 left vectors $\mathbf{v}_1, \dots, \mathbf{v}_{b_1}$ and b_2 right vectors $\mathbf{w}_1, \dots, \mathbf{w}_{b_2}$

$$\begin{bmatrix} \mathbf{v}_1^T \\ \vdots \\ \mathbf{v}_{b_1}^T \end{bmatrix} \left(\begin{matrix} \times \binom{m+\ell_1}{m+\ell_1} & \times \binom{m+\ell_1}{n+\ell_2} \\ \mathbf{F} & \mathbf{B}_i \end{matrix} - \sum_{j=1}^{km} \mu_{i,j} \mathbf{B}_j \right) \begin{bmatrix} \mathbf{w}_1 & \cdots & \mathbf{w}_{b_2} \end{bmatrix} = \mathbf{0} \quad \text{for } 1 \leq i \leq km$$

$$\# \text{equations} := b_1 b_2 km$$

$$\# \text{unknowns} := (m + \ell_1)^2 + (km)^2$$

$$\# \text{solution-space dimension} := \text{somewhat involved}$$

$$\text{We want} \quad \# \text{equations} \geq \# \text{unknowns} - \# \text{solution-space dimension} + 1 \quad (*)$$

$$\text{Probability of correctly guessing the } \mathbf{v}_a \text{'s and } \mathbf{w}_b \text{'s is } \frac{\binom{m}{b_1}_q}{\binom{m+\ell_1}{b_1}_q} \frac{\binom{n}{b_2}_q}{\binom{n+\ell_2}{b_2}_q} \approx q^{-(b_1 \ell_1 + b_2 \ell_2)}$$

Minimize $b_1 \ell_1 + b_2 \ell_2$ subject to $(*)$

Solution Idea

- Public
- Secret

Strategy: Find matrix $\mathbf{F}$ and matrices $\mathbf{C}_1, \dots, \mathbf{C}_{km} \in \mathcal{C}_{\text{pub}}$ such that

$$\mathbf{F}\mathbf{B}_i - \mathbf{C}_i = \mathbf{P} \begin{pmatrix} \mathbf{0} & * \\ * & * \end{pmatrix} \mathbf{Q} \quad \text{for all } i = 1, \dots, km$$

Strategy: Find matrix $\mathbf{F}$ and matrices $\mathbf{C}_1, \dots, \mathbf{C}_{km} \in \mathcal{C}_{\text{pub}}$ such that

$$\mathbf{F}\mathbf{B}_i - \mathbf{C}_i = \mathbf{P} \begin{pmatrix} \mathbf{0} & * \\ * & * \end{pmatrix} \mathbf{Q} \quad \text{for all } i = 1, \dots, km$$

Q1 How to solve for $\mathbf{F}$ and $\mathbf{C}_1, \dots, \mathbf{C}_{km}$? ✓ (almost...)

Solution Idea

- Public
- Secret

Strategy: Find matrix $\mathbf{F}$ and matrices $\mathbf{C}_1, \dots, \mathbf{C}_{km} \in \mathcal{C}_{\text{pub}}$ such that

$$\mathbf{F}\mathbf{B}_i - \mathbf{C}_i = \mathbf{P} \begin{pmatrix} \mathbf{0} & * \\ * & * \end{pmatrix} \mathbf{Q} \quad \text{for all } i = 1, \dots, km$$

Q1 How to solve for $\mathbf{F}$ and $\mathbf{C}_1, \dots, \mathbf{C}_{km}$? ✓ (almost...)

Q2 How does this help recover $\mathcal{C}_{\text{mat}}$?

Q2 How does a solution help recover $\mathcal{C}_{\text{mat}}$?

Suppose, for now, we have a correct solution $\mathbf{F}$ and $\mathbf{C}_1, \dots, \mathbf{C}_{km} \in \mathcal{C}_{\text{pub}}$

• Public

• Secret

Q2 How does a solution help recover $\mathcal{C}_{\text{mat}}$?

Suppose, for now, we have a correct solution $\mathbf{F}$ and $\mathbf{C}_1, \dots, \mathbf{C}_{km} \in \mathcal{C}_{\text{pub}}$

• Public

• Secret

$$\begin{bmatrix} \mathbf{v}_1^\top \\ \vdots \\ \mathbf{v}_{b_1}^\top \end{bmatrix} \left(\mathbf{F} \mathbf{B}_i - \mathbf{C}_i \right) \begin{bmatrix} \mathbf{w}_1 & \cdots & \mathbf{w}_{b_2} \end{bmatrix} = \mathbf{0}$$

Q2 How does a solution help recover $\mathcal{C}_{\text{mat}}$?

• Public

• Secret

Suppose, for now, we have a correct solution $\mathbf{F}$ and $\mathbf{C}_1, \dots, \mathbf{C}_{km} \in \mathcal{C}_{\text{pub}}$

$$\begin{bmatrix} \mathbf{v}_1^\top \\ \vdots \\ \mathbf{v}_{b_1}^\top \end{bmatrix} \left(\mathbf{F} \mathbf{B}_i - \mathbf{C}_i \right) \begin{bmatrix} \mathbf{w}_1 & \cdots & \mathbf{w}_{b_2} \end{bmatrix} = \mathbf{0} \quad \text{recall } \mathbf{F} \mathbf{B}_i - \mathbf{C}_i = \mathbf{P} \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ * & * \end{pmatrix} \mathbf{Q} + \mathbf{P} \begin{pmatrix} \mathbf{0} & * \\ \mathbf{0} & * \end{pmatrix} \mathbf{Q}$$

Q2 How does a solution help recover $\mathcal{C}_{\text{mat}}$?

• Public

• Secret

Suppose, for now, we have a correct solution $\mathbf{F}$ and $\mathbf{C}_1, \dots, \mathbf{C}_{km} \in \mathcal{C}_{\text{pub}}$

$$\begin{bmatrix} \mathbf{v}_1^\top \\ \vdots \\ \mathbf{v}_{b_1}^\top \end{bmatrix} \left(\mathbf{F} \mathbf{B}_i - \mathbf{C}_i \right) \begin{bmatrix} \mathbf{w}_1 & \cdots & \mathbf{w}_{b_2} \end{bmatrix} = \mathbf{0} \quad \text{recall } \mathbf{F} \mathbf{B}_i - \mathbf{C}_i = \mathbf{P} \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ * & * \end{pmatrix} \mathbf{Q} + \mathbf{P} \begin{pmatrix} \mathbf{0} & * \\ \mathbf{0} & * \end{pmatrix} \mathbf{Q}$$

If we omit $\begin{bmatrix} \mathbf{w}_1 & \cdots & \mathbf{w}_{b_2} \end{bmatrix}$, then

Q2 How does a solution help recover $\mathbf{C}_{\text{mat}}$?

• Public

• Secret

Suppose, for now, we have a correct solution $\mathbf{F}$ and $\mathbf{C}_1, \dots, \mathbf{C}_{km} \in \mathcal{C}_{\text{pub}}$

$$\begin{bmatrix} \mathbf{v}_1^\top \\ \vdots \\ \mathbf{v}_{b_1}^\top \end{bmatrix} \left(\mathbf{F} \mathbf{B}_i - \mathbf{C}_i \right) \begin{bmatrix} \mathbf{w}_1 & \cdots & \mathbf{w}_{b_2} \end{bmatrix} = \mathbf{0} \quad \text{recall } \mathbf{F} \mathbf{B}_i - \mathbf{C}_i = \mathbf{P} \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ * & * \end{pmatrix} \mathbf{Q} + \mathbf{P} \begin{pmatrix} \mathbf{0} & * \\ \mathbf{0} & * \end{pmatrix} \mathbf{Q}$$

If we omit $\begin{bmatrix} \mathbf{w}_1 & \cdots & \mathbf{w}_{b_2} \end{bmatrix}$, then

$$\begin{bmatrix} \mathbf{v}_1^\top \\ \vdots \\ \mathbf{v}_{b_1}^\top \end{bmatrix} \left(\mathbf{F} \mathbf{B}_i - \mathbf{C}_i \right)$$

Q2 How does a solution help recover $\mathcal{C}_{\text{mat}}$?

• Public

• Secret

Suppose, for now, we have a correct solution $\mathbf{F}$ and $\mathbf{C}_1, \dots, \mathbf{C}_{km} \in \mathcal{C}_{\text{pub}}$

$$\begin{bmatrix} \mathbf{v}_1^\top \\ \vdots \\ \mathbf{v}_{b_1}^\top \end{bmatrix} \left(\mathbf{F} \mathbf{B}_i - \mathbf{C}_i \right) \begin{bmatrix} \mathbf{w}_1 & \cdots & \mathbf{w}_{b_2} \end{bmatrix} = \mathbf{0} \quad \text{recall } \mathbf{F} \mathbf{B}_i - \mathbf{C}_i = \mathbf{P} \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ * & * \end{pmatrix} \mathbf{Q} + \mathbf{P} \begin{pmatrix} \mathbf{0} & * \\ \mathbf{0} & * \end{pmatrix} \mathbf{Q}$$

If we omit $\begin{bmatrix} \mathbf{w}_1 & \cdots & \mathbf{w}_{b_2} \end{bmatrix}$, then

$$\begin{bmatrix} \mathbf{v}_1^\top \\ \vdots \\ \mathbf{v}_{b_1}^\top \end{bmatrix} \left(\mathbf{F} \mathbf{B}_i - \mathbf{C}_i \right) = \begin{bmatrix} \mathbf{v}_1^\top \\ \vdots \\ \mathbf{v}_{b_1}^\top \end{bmatrix} \left(\mathbf{P} \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ * & * \end{pmatrix} \mathbf{Q} + \mathbf{P} \begin{pmatrix} \mathbf{0} & * \\ \mathbf{0} & * \end{pmatrix} \mathbf{Q} \right)$$

Q2 How does a solution help recover $\mathcal{C}_{\text{mat}}$?

• Public

• Secret

Suppose, for now, we have a correct solution $\mathbf{F}$ and $\mathbf{C}_1, \dots, \mathbf{C}_{km} \in \mathcal{C}_{\text{pub}}$

$$\begin{bmatrix} \mathbf{v}_1^\top \\ \vdots \\ \mathbf{v}_{b_1}^\top \end{bmatrix} \left(\mathbf{F} \mathbf{B}_i - \mathbf{C}_i \right) \begin{bmatrix} \mathbf{w}_1 & \cdots & \mathbf{w}_{b_2} \end{bmatrix} = \mathbf{0} \quad \text{recall } \mathbf{F} \mathbf{B}_i - \mathbf{C}_i = \mathbf{P} \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ * & * \end{pmatrix} \mathbf{Q} + \mathbf{P} \begin{pmatrix} \mathbf{0} & * \\ \mathbf{0} & * \end{pmatrix} \mathbf{Q}$$

If we omit $\begin{bmatrix} \mathbf{w}_1 & \cdots & \mathbf{w}_{b_2} \end{bmatrix}$, then

$$\begin{aligned} \begin{bmatrix} \mathbf{v}_1^\top \\ \vdots \\ \mathbf{v}_{b_1}^\top \end{bmatrix} \left(\mathbf{F} \mathbf{B}_i - \mathbf{C}_i \right) &= \begin{bmatrix} \mathbf{v}_1^\top \\ \vdots \\ \mathbf{v}_{b_1}^\top \end{bmatrix} \left(\mathbf{P} \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ * & * \end{pmatrix} \mathbf{Q} + \mathbf{P} \begin{pmatrix} \mathbf{0} & * \\ \mathbf{0} & * \end{pmatrix} \mathbf{Q} \right) \\ &= \mathbf{0} + \begin{bmatrix} \mathbf{v}_1^\top \\ \vdots \\ \mathbf{v}_{b_1}^\top \end{bmatrix} \mathbf{P} \begin{pmatrix} \mathbf{0} & * \\ \mathbf{0} & * \end{pmatrix} \mathbf{Q} \end{aligned}$$

Q2 How does a solution help recover $\mathcal{C}_{\text{mat}}$?

• Public

• Secret

Suppose, for now, we have a correct solution $\mathbf{F}$ and $\mathbf{C}_1, \dots, \mathbf{C}_{km} \in \mathcal{C}_{\text{pub}}$

$$\begin{bmatrix} \mathbf{v}_1^\top \\ \vdots \\ \mathbf{v}_{b_1}^\top \end{bmatrix} \left(\mathbf{F} \mathbf{B}_i - \mathbf{C}_i \right) \begin{bmatrix} \mathbf{w}_1 & \cdots & \mathbf{w}_{b_2} \end{bmatrix} = \mathbf{0} \quad \text{recall } \mathbf{F} \mathbf{B}_i - \mathbf{C}_i = \mathbf{P} \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ * & * \end{pmatrix} \mathbf{Q} + \mathbf{P} \begin{pmatrix} \mathbf{0} & * \\ \mathbf{0} & * \end{pmatrix} \mathbf{Q}$$

If we omit $\begin{bmatrix} \mathbf{w}_1 & \cdots & \mathbf{w}_{b_2} \end{bmatrix}$, then

$$\begin{aligned} \begin{bmatrix} \mathbf{v}_1^\top \\ \vdots \\ \mathbf{v}_{b_1}^\top \end{bmatrix} \left(\mathbf{F} \mathbf{B}_i - \mathbf{C}_i \right) &= \begin{bmatrix} \mathbf{v}_1^\top \\ \vdots \\ \mathbf{v}_{b_1}^\top \end{bmatrix} \left(\mathbf{P} \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ * & * \end{pmatrix} \mathbf{Q} + \mathbf{P} \begin{pmatrix} \mathbf{0} & * \\ \mathbf{0} & * \end{pmatrix} \mathbf{Q} \right) \\ &= \mathbf{0} + \begin{bmatrix} \mathbf{v}_1^\top \\ \vdots \\ \mathbf{v}_{b_1}^\top \end{bmatrix} \mathbf{P} \begin{pmatrix} \mathbf{0} & * \\ \mathbf{0} & * \end{pmatrix} \mathbf{Q} \\ &= \begin{pmatrix} \mathbf{0} & * \\ \mathbf{0} & * \end{pmatrix} \mathbf{Q} \end{aligned}$$

Q2 How does a solution help recover $\mathcal{C}_{\text{mat}}$?

• Public

• Secret

Suppose, for now, we have a correct solution $\mathbf{F}$ and $\mathbf{C}_1, \dots, \mathbf{C}_{km} \in \mathcal{C}_{\text{pub}}$

$$\begin{bmatrix} \mathbf{v}_1^\top \\ \vdots \\ \mathbf{v}_{b_1}^\top \end{bmatrix} \left(\mathbf{F} \mathbf{B}_i - \mathbf{C}_i \right) \begin{bmatrix} \mathbf{w}_1 & \cdots & \mathbf{w}_{b_2} \end{bmatrix} = \mathbf{0} \quad \text{recall } \mathbf{F} \mathbf{B}_i - \mathbf{C}_i = \mathbf{P} \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ * & * \end{pmatrix} \mathbf{Q} + \mathbf{P} \begin{pmatrix} \mathbf{0} & * \\ \mathbf{0} & * \end{pmatrix} \mathbf{Q}$$

If we omit $\begin{bmatrix} \mathbf{w}_1 & \cdots & \mathbf{w}_{b_2} \end{bmatrix}$, then

$$\begin{aligned} \begin{bmatrix} \mathbf{v}_1^\top \\ \vdots \\ \mathbf{v}_{b_1}^\top \end{bmatrix} \left(\mathbf{F} \mathbf{B}_i - \mathbf{C}_i \right) &= \begin{bmatrix} \mathbf{v}_1^\top \\ \vdots \\ \mathbf{v}_{b_1}^\top \end{bmatrix} \left(\mathbf{P} \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ * & * \end{pmatrix} \mathbf{Q} + \mathbf{P} \begin{pmatrix} \mathbf{0} & * \\ \mathbf{0} & * \end{pmatrix} \mathbf{Q} \right) \\ &= \mathbf{0} + \begin{bmatrix} \mathbf{v}_1^\top \\ \vdots \\ \mathbf{v}_{b_1}^\top \end{bmatrix} \mathbf{P} \begin{pmatrix} \mathbf{0} & * \\ \mathbf{0} & * \end{pmatrix} \mathbf{Q} \\ &= \begin{pmatrix} \mathbf{0} & * \\ \mathbf{0} & * \end{pmatrix} \mathbf{Q} \implies \begin{array}{l} \text{reveals enough info to} \\ \text{remove random columns} \end{array} \\ &\quad \text{☺} \end{aligned}$$

Q2 How does a solution help recover $\mathcal{C}_{\text{mat}}$?

• Public

• Secret

Suppose, for now, we have a correct solution $\mathbf{F}$ and $\mathbf{C}_1, \dots, \mathbf{C}_{km} \in \mathcal{C}_{\text{pub}}$

$$\begin{bmatrix} \mathbf{v}_1^\top \\ \vdots \\ \mathbf{v}_{b_1}^\top \end{bmatrix} \left(\mathbf{F} \mathbf{B}_i - \mathbf{C}_i \right) \begin{bmatrix} \mathbf{w}_1 & \cdots & \mathbf{w}_{b_2} \end{bmatrix} = \mathbf{0} \quad \text{recall } \mathbf{F} \mathbf{B}_i - \mathbf{C}_i = \mathbf{P} \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ * & * \end{pmatrix} \mathbf{Q} + \mathbf{P} \begin{pmatrix} \mathbf{0} & * \\ \mathbf{0} & * \end{pmatrix} \mathbf{Q}$$

If we omit $\begin{bmatrix} \mathbf{w}_1 & \cdots & \mathbf{w}_{b_2} \end{bmatrix}$, then

$$\begin{bmatrix} \mathbf{v}_1^\top \\ \vdots \\ \mathbf{v}_{b_1}^\top \end{bmatrix} \left(\mathbf{F} \mathbf{B}_i - \mathbf{C}_i \right) = \begin{bmatrix} \mathbf{v}_1^\top \\ \vdots \\ \mathbf{v}_{b_1}^\top \end{bmatrix} \left(\mathbf{P} \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ * & * \end{pmatrix} \mathbf{Q} + \mathbf{P} \begin{pmatrix} \mathbf{0} & * \\ \mathbf{0} & * \end{pmatrix} \mathbf{Q} \right)$$

$$= \mathbf{0} + \begin{bmatrix} \mathbf{v}_1^\top \\ \vdots \\ \mathbf{v}_{b_1}^\top \end{bmatrix} \mathbf{P} \begin{pmatrix} \mathbf{0} & * \\ \mathbf{0} & * \end{pmatrix} \mathbf{Q}$$

$$= \begin{pmatrix} \mathbf{0} & * \\ \mathbf{0} & * \end{pmatrix} \mathbf{Q} \implies \text{reveals enough info to remove random columns}$$



Unfortunately, we cannot do the same from the left and omit the $\mathbf{v}_a$'s because $\mathbf{F}$ is corrupted

Q2 How does a solution help recover $\mathcal{C}_{\text{mat}}$?

• Public

• Secret

Suppose, for now, we have a correct solution $\mathbf{F}$ and $\mathbf{C}_1, \dots, \mathbf{C}_{km} \in \mathcal{C}_{\text{pub}}$

$$\begin{bmatrix} \mathbf{v}_1^\top \\ \vdots \\ \mathbf{v}_{b_1}^\top \end{bmatrix} \left(\mathbf{F} \mathbf{B}_i - \mathbf{C}_i \right) \begin{bmatrix} \mathbf{w}_1 & \cdots & \mathbf{w}_{b_2} \end{bmatrix} = \mathbf{0} \quad \text{recall } \mathbf{F} \mathbf{B}_i - \mathbf{C}_i = \mathbf{P} \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ * & * \end{pmatrix} \mathbf{Q} + \mathbf{P} \begin{pmatrix} \mathbf{0} & * \\ \mathbf{0} & * \end{pmatrix} \mathbf{Q}$$

If we omit $\begin{bmatrix} \mathbf{w}_1 & \cdots & \mathbf{w}_{b_2} \end{bmatrix}$, then

$$\begin{aligned} \begin{bmatrix} \mathbf{v}_1^\top \\ \vdots \\ \mathbf{v}_{b_1}^\top \end{bmatrix} \left(\mathbf{F} \mathbf{B}_i - \mathbf{C}_i \right) &= \begin{bmatrix} \mathbf{v}_1^\top \\ \vdots \\ \mathbf{v}_{b_1}^\top \end{bmatrix} \left(\mathbf{P} \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ * & * \end{pmatrix} \mathbf{Q} + \mathbf{P} \begin{pmatrix} \mathbf{0} & * \\ \mathbf{0} & * \end{pmatrix} \mathbf{Q} \right) \\ &= \mathbf{0} + \begin{bmatrix} \mathbf{v}_1^\top \\ \vdots \\ \mathbf{v}_{b_1}^\top \end{bmatrix} \mathbf{P} \begin{pmatrix} \mathbf{0} & * \\ \mathbf{0} & * \end{pmatrix} \mathbf{Q} \\ &= \begin{pmatrix} \mathbf{0} & * \\ \mathbf{0} & * \end{pmatrix} \mathbf{Q} \implies \text{reveals enough info to} \\ &\quad \text{remove random columns} \end{aligned}$$



Unfortunately, we cannot do the same from the left and omit the $\mathbf{v}_a$'s because $\mathbf{F}$ is corrupted

A solution for $\mathbf{F}$ has the form $\mathbf{F}_{\text{true}} + \mathbf{K}$ with $\mathbf{K}$ such that $\mathbf{v}_a^\top \mathbf{K} = \mathbf{0}$

Q2 How does a solution help recover C_{mat} ?

So now what?

- Public
- Secret

Q2 How does a solution help recover $\mathcal{C}_{\text{mat}}$?

So now what? After removing the random columns, our problem looks like

- Public
- Secret

Q2 How does a solution help recover $\mathcal{C}_{\text{mat}}$?

So now what? After removing the random columns, our problem looks like

$$\mathbf{B}'_1 = \mathbf{P} \begin{pmatrix} \mathbf{A}_1 \\ \mathbf{R}'_1 \end{pmatrix}, \quad \mathbf{B}'_2 = \mathbf{P} \begin{pmatrix} \mathbf{A}_2 \\ \mathbf{R}'_2 \end{pmatrix}, \quad \dots \quad (\text{note } \mathbf{Q} \text{ can be omitted without loss of generality})$$

• Public

• Secret

Q2 How does a solution help recover $\mathcal{C}_{\text{mat}}$?

• Public

• Secret

So now what? After removing the random columns, our problem looks like

$$\mathbf{B}'_1 = \mathbf{P} \begin{pmatrix} \mathbf{A}_1 \\ \mathbf{R}'_1 \end{pmatrix}, \quad \mathbf{B}'_2 = \mathbf{P} \begin{pmatrix} \mathbf{A}_2 \\ \mathbf{R}'_2 \end{pmatrix}, \quad \dots \quad (\text{note } \mathbf{Q} \text{ can be omitted without loss of generality})$$

- We obtain a solution for $\mathbf{F}$ but this solution is corrupted

Q2 How does a solution help recover $\mathcal{C}_{\text{mat}}$?

• Public

• Secret

So now what? After removing the random columns, our problem looks like

$$\mathbf{B}'_1 = \mathbf{P} \begin{pmatrix} \mathbf{A}_1 \\ \mathbf{R}'_1 \end{pmatrix}, \quad \mathbf{B}'_2 = \mathbf{P} \begin{pmatrix} \mathbf{A}_2 \\ \mathbf{R}'_2 \end{pmatrix}, \quad \dots \quad (\text{note } \mathbf{Q} \text{ can be omitted without loss of generality})$$

- We obtain a solution for $\mathbf{F}$ but this solution is corrupted
but we also obtain solutions for $\mathbf{C}_1, \mathbf{C}_2, \dots, \mathbf{C}_{km}$ which *are* correct

Q2 How does a solution help recover $\mathcal{C}_{\text{mat}}$?

• Public

• Secret

So now what? After removing the random columns, our problem looks like

$$\mathbf{B}'_1 = \mathbf{P} \begin{pmatrix} \mathbf{A}_1 \\ \mathbf{R}'_1 \end{pmatrix}, \quad \mathbf{B}'_2 = \mathbf{P} \begin{pmatrix} \mathbf{A}_2 \\ \mathbf{R}'_2 \end{pmatrix}, \quad \dots \quad (\text{note } \mathbf{Q} \text{ can be omitted without loss of generality})$$

- ▶ We obtain a solution for $\mathbf{F}$ but this solution is corrupted
but we also obtain solutions for $\mathbf{C}_1, \mathbf{C}_2, \dots, \mathbf{C}_{km}$ which *are* correct
- ▶ The maps

$$\mathbf{B}'_1 \mapsto \mathbf{C}_1, \quad \dots, \quad \mathbf{B}'_{km} \mapsto \mathbf{C}_{km}$$

multiply the underlying $\mathcal{C}_{\text{mat}}$ matrices by some $\mathbb{F}_{q^m}$ -element t

Q2 How does a solution help recover $\mathcal{C}_{\text{mat}}$?

• Public

• Secret

So now what? After removing the random columns, our problem looks like

$$\mathbf{B}'_1 = \mathbf{P} \begin{pmatrix} \mathbf{A}_1 \\ \mathbf{R}'_1 \end{pmatrix}, \quad \mathbf{B}'_2 = \mathbf{P} \begin{pmatrix} \mathbf{A}_2 \\ \mathbf{R}'_2 \end{pmatrix}, \quad \dots \quad (\text{note } \mathbf{Q} \text{ can be omitted without loss of generality})$$

- ▶ We obtain a solution for $\mathbf{F}$ but this solution is corrupted
but we also obtain solutions for $\mathbf{C}_1, \mathbf{C}_2, \dots, \mathbf{C}_{km}$ which *are* correct

- ▶ The maps

$$\mathbf{B}'_1 \mapsto \mathbf{C}_1, \quad \dots, \quad \mathbf{B}'_{km} \mapsto \mathbf{C}_{km}$$

multiply the underlying $\mathcal{C}_{\text{mat}}$ matrices by some $\mathbb{F}_{q^m}$ -element t

- ▶ We can therefore set up the *linear* system

$$\mathbf{F}^{(1)} \mathbf{B}'_i = \mathbf{F}^{(2)} \mathbf{C}_i, \quad i = 1, \dots, km,$$

Q2 How does a solution help recover $\mathcal{C}_{\text{mat}}$?

• Public

• Secret

So now what? After removing the random columns, our problem looks like

$$\mathbf{B}'_1 = \mathbf{P} \begin{pmatrix} \mathbf{A}_1 \\ \mathbf{R}'_1 \end{pmatrix}, \quad \mathbf{B}'_2 = \mathbf{P} \begin{pmatrix} \mathbf{A}_2 \\ \mathbf{R}'_2 \end{pmatrix}, \quad \dots \quad (\text{note } \mathbf{Q} \text{ can be omitted without loss of generality})$$

- ▶ We obtain a solution for $\mathbf{F}$ but this solution is corrupted but we also obtain solutions for $\mathbf{C}_1, \mathbf{C}_2, \dots, \mathbf{C}_{km}$ which *are* correct

- ▶ The maps

$$\mathbf{B}'_1 \mapsto \mathbf{C}_1, \quad \dots, \quad \mathbf{B}'_{km} \mapsto \mathbf{C}_{km}$$

multiply the underlying $\mathcal{C}_{\text{mat}}$ matrices by some $\mathbb{F}_{q^m}$ -element t

- ▶ We can therefore set up the *linear* system

$$\mathbf{F}^{(1)} \mathbf{B}'_i = \mathbf{F}^{(2)} \mathbf{C}_i, \quad i = 1, \dots, km,$$

where $\mathbf{F}^{(1)}$ removes the random rows and applies multiplication by t , while $\mathbf{F}^{(2)}$ only removes the random rows

Q2 How does a solution help recover $\mathcal{C}_{\text{mat}}$?

• Public

• Secret

So now what? After removing the random columns, our problem looks like

$$\mathbf{B}'_1 = \mathbf{P} \begin{pmatrix} \mathbf{A}_1 \\ \mathbf{R}'_1 \end{pmatrix}, \quad \mathbf{B}'_2 = \mathbf{P} \begin{pmatrix} \mathbf{A}_2 \\ \mathbf{R}'_2 \end{pmatrix}, \quad \dots \quad (\text{note } \mathbf{Q} \text{ can be omitted without loss of generality})$$

- ▶ We obtain a solution for $\mathbf{F}$ but this solution is corrupted but we also obtain solutions for $\mathbf{C}_1, \mathbf{C}_2, \dots, \mathbf{C}_{km}$ which *are* correct

- ▶ The maps

$$\mathbf{B}'_1 \mapsto \mathbf{C}_1, \quad \dots, \quad \mathbf{B}'_{km} \mapsto \mathbf{C}_{km}$$

multiply the underlying $\mathcal{C}_{\text{mat}}$ matrices by some $\mathbb{F}_{q^m}$ -element t

- ▶ We can therefore set up the *linear* system

$$\mathbf{F}^{(1)} \mathbf{B}'_i = \mathbf{F}^{(2)} \mathbf{C}_i, \quad i = 1, \dots, km,$$

where $\mathbf{F}^{(1)}$ removes the random rows and applies multiplication by t , while $\mathbf{F}^{(2)}$ only removes the random rows

$\implies$ we can remove the random rows and recover $\mathcal{C}_{\text{mat}}$ up to equivalence ☺

Complexity (ignoring polynomial factors)

Complexity (ignoring polynomial factors)

Algorithm	Complexity (assume $m = n$)
Naive Key Recovery	$q^{m\ell_1 + m\ell_2}$

Complexity (ignoring polynomial factors)

Algorithm	Complexity (assume $m = n$)
Naive Key Recovery	$q^{m\ell_1 + m\ell_2}$
ACDGV Distinguisher	$q^{m\ell_1 + (k+1)\ell_2}$

Complexity (ignoring polynomial factors)

Algorithm	Complexity (assume $m = n$)
Naive Key Recovery	$q^{m\ell_1 + m\ell_2}$
ACDGV Distinguisher	$q^{m\ell_1 + (k+1)\ell_2}$
Our Key Recovery	$q^{2\sqrt{km\ell_1\ell_2} + \ell_1 + (1+\epsilon)\ell_2}$ if $\frac{k}{m} \lesssim \min\left(\frac{\ell_1}{\ell_2}, \frac{\ell_2}{\ell_1}\right)$ else $\min\{q^{m\ell_1 + (k+1)\ell_2}, q^{(k+1)\ell_1 + m\ell_2}\}$

Complexity (ignoring polynomial factors)

Algorithm	Complexity (assume $m = n$)
Naive Key Recovery	$q^{m\ell_1 + m\ell_2}$
ACDGV Distinguisher	$q^{m\ell_1 + (k+1)\ell_2}$
Our Key Recovery	$q^{2\sqrt{km\ell_1\ell_2} + \ell_1 + (1+\epsilon)\ell_2}$ if $\frac{k}{m} \lesssim \min\left(\frac{\ell_1}{\ell_2}, \frac{\ell_2}{\ell_1}\right)$ else $\min\{q^{m\ell_1 + (k+1)\ell_2}, q^{(k+1)\ell_1 + m\ell_2}\}$

Using exact estimates, it can be shown the exponents satisfy

$$\text{Our Recovery} \leq \text{ACDGV Distinguisher} \leq \text{Naive Recovery}$$

Concrete Costs for ACDGV Parameters*

*Three smallest public keys per security level

Concrete Costs for ACDGV Parameters*

Security Level	q	$\frac{k}{m}$	m	ℓ_1	ℓ_2	Ours	ACDGV Dist.	Naive
143 bits (AES-128)	16	0.30	23	0	5	196	189	489
	16	0.57	23	1	1	176	177	213
	2	0.46	37	4	0	112	179	179
207 bits (AES-192)	2	0.53	43	5	0	161	248	248
	2	0.70	47	5	0	212	268	268
	2	0.77	53	4	0	211	246	246
272 bits (AES-256)	16	0.31	29	2	1	223	303	379
	16	0.59	29	2	1	295	335	379
	2	0.49	47	3	3	240	246	315

Gaussian elimination costs about 30–45 bits (included in the costs above)

*Three smallest public keys per security level

Concrete Costs for ACDGV Parameters*

Security Level	q	$\frac{k}{m}$	m	ℓ_1	ℓ_2	Ours	ACDGV Dist.	Naive
143 bits (AES-128)	16	0.30	23	0	5	196	189	489
	16	0.57	23	1	1	176	177	213
	2	0.46	37	4	0	112	179	179
207 bits (AES-192)	2	0.53	43	5	0	161	248	248
	2	0.70	47	5	0	212	268	268
	2	0.77	53	4	0	211	246	246
272 bits (AES-256)	16	0.31	29	2	1	223	303	379
	16	0.59	29	2	1	295	335	379
	2	0.49	47	3	3	240	246	315

Gaussian elimination costs about 30–45 bits (included in the costs above)

Overall, we break 5 of their 16 parameter sets

*Three smallest public keys per security level

Newer attack by Le (ePrint 2026/972)

Newer attack by Le (ePrint 2026/972)

- ▶ Assume we have already removed the random columns by guessing

Newer attack by Le (ePrint 2026/972)

- ▶ Assume we have already removed the random columns by guessing
Then puncture to a code of blocklength $k + 1$,

Newer attack by Le (ePrint 2026/972)

- Assume we have already removed the random columns by guessing
Then puncture to a code of blocklength $k + 1$,

$$\mathbf{B}_i = \mathbf{P} \begin{pmatrix} \mathbf{A}_i \\ \mathbf{R}'_i \end{pmatrix} \in \mathbb{F}_q^{(m+\ell_1) \times (k+1)} \text{ for } 1 \leq i \leq km$$

Newer attack by Le (ePrint 2026/972)

- Assume we have already removed the random columns by guessing
Then puncture to a code of blocklength $k + 1$,

$$\mathbf{B}_i = \mathbf{P} \begin{pmatrix} \mathbf{A}_i \\ \mathbf{R}'_i \end{pmatrix} \in \mathbb{F}_q^{(m+\ell_1) \times (k+1)} \text{ for } 1 \leq i \leq km$$

- There exists a $\mathbf{u} \in \mathbb{F}_{q^m}^{m+\ell_1}$ such that $\mathbf{u}^T \mathbf{B}_i \in \mathbb{F}_{q^m}^{k+1}$ is the corresponding codeword in $\mathcal{C}_{\text{vec}}$

Newer attack by Le (ePrint 2026/972)

- Assume we have already removed the random columns by guessing
Then puncture to a code of blocklength $k + 1$,

$$\mathbf{B}_i = \mathbf{P} \begin{pmatrix} \mathbf{A}_i \\ \mathbf{R}'_i \end{pmatrix} \in \mathbb{F}_q^{(m+\ell_1) \times (k+1)} \text{ for } 1 \leq i \leq km$$

- There exists a $\mathbf{u} \in \mathbb{F}_{q^m}^{m+\ell_1}$ such that $\mathbf{u}^\top \mathbf{B}_i \in \mathbb{F}_{q^m}^{k+1}$ is the corresponding codeword in $\mathcal{C}_{\text{vec}}$
pick $\mathbf{u}^\top = (\boldsymbol{\alpha}^\top \mathbf{0}^\top) \mathbf{P}^{-1}$ so $\mathbf{u}^\top \mathbf{B}_i = \boldsymbol{\alpha}^\top \mathbf{A}_i \in \mathcal{C}_{\text{vec}}$ (where $\boldsymbol{\alpha}$ is a basis of $\mathbb{F}_{q^m}$)

Newer attack by Le (ePrint 2026/972)

- Assume we have already removed the random columns by guessing
Then puncture to a code of blocklength $k + 1$,

$$\mathbf{B}_i = \mathbf{P} \begin{pmatrix} \mathbf{A}_i \\ \mathbf{R}'_i \end{pmatrix} \in \mathbb{F}_q^{(m+\ell_1) \times (k+1)} \text{ for } 1 \leq i \leq km$$

- There exists a $\mathbf{u} \in \mathbb{F}_{q^m}^{m+\ell_1}$ such that $\mathbf{u}^\top \mathbf{B}_i \in \mathbb{F}_{q^m}^{k+1}$ is the corresponding codeword in $\mathcal{C}_{\text{vec}}$
pick $\mathbf{u}^\top = (\boldsymbol{\alpha}^\top \mathbf{0}^\top) \mathbf{P}^{-1}$ so $\mathbf{u}^\top \mathbf{B}_i = \boldsymbol{\alpha}^\top \mathbf{A}_i \in \mathcal{C}_{\text{vec}}$ (where $\boldsymbol{\alpha}$ is a basis of $\mathbb{F}_{q^m}$)
Further, since $\mathcal{C}_{\text{vec}}$ is a $[k+1, k]_{\mathbb{F}_{q^m}}$ code, there exists $\mathbf{h} \in \mathbb{F}_{q^m}^{k+1}$ such that

$$\mathbf{u}^\top \mathbf{B}_i \mathbf{h} = 0 \quad \text{for } 1 \leq i \leq km$$

Newer attack by Le (ePrint 2026/972)

- Assume we have already removed the random columns by guessing
Then puncture to a code of blocklength $k + 1$,

$$\mathbf{B}_i = \mathbf{P} \begin{pmatrix} \mathbf{A}_i \\ \mathbf{R}'_i \end{pmatrix} \in \mathbb{F}_q^{(m+\ell_1) \times (k+1)} \text{ for } 1 \leq i \leq km$$

- There exists a $\mathbf{u} \in \mathbb{F}_{q^m}^{m+\ell_1}$ such that $\mathbf{u}^\top \mathbf{B}_i \in \mathbb{F}_{q^m}^{k+1}$ is the corresponding codeword in $\mathcal{C}_{\text{vec}}$
pick $\mathbf{u}^\top = (\boldsymbol{\alpha}^\top \mathbf{0}^\top) \mathbf{P}^{-1}$ so $\mathbf{u}^\top \mathbf{B}_i = \boldsymbol{\alpha}^\top \mathbf{A}_i \in \mathcal{C}_{\text{vec}}$ (where $\boldsymbol{\alpha}$ is a basis of $\mathbb{F}_{q^m}$)
Further, since $\mathcal{C}_{\text{vec}}$ is a $[k + 1, k]_{\mathbb{F}_{q^m}}$ code, there exists $\mathbf{h} \in \mathbb{F}_{q^m}^{k+1}$ such that

$$\mathbf{u}^\top \mathbf{B}_i \mathbf{h} = 0 \quad \text{for } 1 \leq i \leq km$$

- This is a bilinear system in the unknowns u_a and h_b

Newer attack by Le (ePrint 2026/972)

- Assume we have already removed the random columns by guessing
Then puncture to a code of blocklength $k + 1$,

$$\mathbf{B}_i = \mathbf{P} \begin{pmatrix} \mathbf{A}_i \\ \mathbf{R}'_i \end{pmatrix} \in \mathbb{F}_q^{(m+\ell_1) \times (k+1)} \text{ for } 1 \leq i \leq km$$

- There exists a $\mathbf{u} \in \mathbb{F}_{q^m}^{m+\ell_1}$ such that $\mathbf{u}^\top \mathbf{B}_i \in \mathbb{F}_{q^m}^{k+1}$ is the corresponding codeword in $\mathcal{C}_{\text{vec}}$
pick $\mathbf{u}^\top = (\boldsymbol{\alpha}^\top \mathbf{0}^\top) \mathbf{P}^{-1}$ so $\mathbf{u}^\top \mathbf{B}_i = \boldsymbol{\alpha}^\top \mathbf{A}_i \in \mathcal{C}_{\text{vec}}$ (where $\boldsymbol{\alpha}$ is a basis of $\mathbb{F}_{q^m}$)
Further, since $\mathcal{C}_{\text{vec}}$ is a $[k + 1, k]_{\mathbb{F}_{q^m}}$ code, there exists $\mathbf{h} \in \mathbb{F}_{q^m}^{k+1}$ such that

$$\mathbf{u}^\top \mathbf{B}_i \mathbf{h} = 0 \quad \text{for } 1 \leq i \leq km$$

- This is a bilinear system in the unknowns u_a and h_b
This system is then reformulated as a system of degree-2 equations

Newer attack by Le (ePrint 2026/972)

- Assume we have already removed the random columns by guessing
Then puncture to a code of blocklength $k + 1$,

$$\mathbf{B}_i = \mathbf{P} \begin{pmatrix} \mathbf{A}_i \\ \mathbf{R}'_i \end{pmatrix} \in \mathbb{F}_q^{(m+\ell_1) \times (k+1)} \text{ for } 1 \leq i \leq km$$

- There exists a $\mathbf{u} \in \mathbb{F}_{q^m}^{m+\ell_1}$ such that $\mathbf{u}^\top \mathbf{B}_i \in \mathbb{F}_{q^m}^{k+1}$ is the corresponding codeword in $\mathcal{C}_{\text{vec}}$
pick $\mathbf{u}^\top = (\boldsymbol{\alpha}^\top \mathbf{0}^\top) \mathbf{P}^{-1}$ so $\mathbf{u}^\top \mathbf{B}_i = \boldsymbol{\alpha}^\top \mathbf{A}_i \in \mathcal{C}_{\text{vec}}$ (where $\boldsymbol{\alpha}$ is a basis of $\mathbb{F}_{q^m}$)
Further, since $\mathcal{C}_{\text{vec}}$ is a $[k + 1, k]_{\mathbb{F}_{q^m}}$ code, there exists $\mathbf{h} \in \mathbb{F}_{q^m}^{k+1}$ such that

$$\mathbf{u}^\top \mathbf{B}_i \mathbf{h} = 0 \quad \text{for } 1 \leq i \leq km$$

- This is a bilinear system in the unknowns u_a and h_b
This system is then reformulated as a system of degree-2 equations
Experimentally, this particular system can be solved efficiently (computing the Macaulay matrix at degree 2 suffices)

Newer attack by Le (ePrint 2026/972)

The reported cost is polynomial when either $\ell_1 = 0$ or $\ell_2 = 0$

Thus, it is much more efficient than our attack

Newer attack by Le (ePrint 2026/972)

The reported cost is polynomial when either $\ell_1 = 0$ or $\ell_2 = 0$

Thus, it is much more efficient than our attack

NIST security level	q	$R = k/m$	m	ℓ_1	ℓ_2	Le's attack	Our attack
143 bits (AES-128)	16	0.30	23	0	5	49	196
	16	0.57	23	1	1	82	176
	2	0.46	37	4	0	35	112
207 bits (AES-192)	2	0.53	43	5	0	38	161
	2	0.70	47	5	0	41	212
	2	0.77	53	4	0	41	211
272 bits (AES-256)	16	0.31	29	2	1	69	223
	16	0.59	29	2	1	103	295
	2	0.49	47	3	3	108	240

Conclusion

- ▶ We presented a key-recovery attack on ACDGV that is better than the previous best distinguisher

Conclusion

- ▶ We presented a key-recovery attack on ACDGV that is better than the previous best distinguisher
- ▶ The newer algebraic attack by Le has a lower cost than our combinatorial attack

Conclusion

- ▶ We presented a key-recovery attack on ACDGV that is better than the previous best distinguisher
- ▶ The newer algebraic attack by Le has a lower cost than our combinatorial attack
- ▶ However, our approach has the following novel features:

Conclusion

- ▶ We presented a key-recovery attack on ACDGV that is better than the previous best distinguisher
- ▶ The newer algebraic attack by Le has a lower cost than our combinatorial attack
- ▶ However, our approach has the following novel features:
 - Our attack in fact applies to any matrix code with a non-trivial stabilizer algebra, and not just those with $\mathbb{F}_{q^m}$ -structure

Conclusion

- ▶ We presented a key-recovery attack on ACDGV that is better than the previous best distinguisher
- ▶ The newer algebraic attack by Le has a lower cost than our combinatorial attack
- ▶ However, our approach has the following novel features:
 - Our attack in fact applies to any matrix code with a non-trivial stabilizer algebra, and not just those with $\mathbb{F}_{q^m}$ -structure
 - This contrasts with Le's attack, which seems to require $\mathbb{F}_{q^m}$ -structure

Conclusion

- ▶ We presented a key-recovery attack on ACDGV that is better than the previous best distinguisher
- ▶ The newer algebraic attack by Le has a lower cost than our combinatorial attack
- ▶ However, our approach has the following novel features:
 - Our attack in fact applies to any matrix code with a non-trivial stabilizer algebra, and not just those with $\mathbb{F}_{q^m}$ -structure
 - This contrasts with Le's attack, which seems to require $\mathbb{F}_{q^m}$ -structure
 - This is useful since there exist efficiently decodable matrix codes without $\mathbb{F}_{q^m}$ -linear structure (Couvreur & Pratihara 2025)

Conclusion

- ▶ We presented a key-recovery attack on ACDGV that is better than the previous best distinguisher
- ▶ The newer algebraic attack by Le has a lower cost than our combinatorial attack
- ▶ However, our approach has the following novel features:
 - Our attack in fact applies to any matrix code with a non-trivial stabilizer algebra, and not just those with $\mathbb{F}_{q^m}$ -structure
 - This contrasts with Le's attack, which seems to require $\mathbb{F}_{q^m}$ -structure
 - This is useful since there exist efficiently decodable matrix codes without $\mathbb{F}_{q^m}$ -linear structure (Couvreur & Pratihara 2025)
 - Our technique of using a *two-sided* kernel search appears new and may be applicable to other cryptosystems

Conclusion

- ▶ We presented a key-recovery attack on ACDGV that is better than the previous best distinguisher
- ▶ The newer algebraic attack by Le has a lower cost than our combinatorial attack
- ▶ However, our approach has the following novel features:
 - Our attack in fact applies to any matrix code with a non-trivial stabilizer algebra, and not just those with $\mathbb{F}_{q^m}$ -structure
 - This contrasts with Le's attack, which seems to require $\mathbb{F}_{q^m}$ -structure
 - This is useful since there exist efficiently decodable matrix codes without $\mathbb{F}_{q^m}$ -linear structure (Couvreur & Pratihara 2025)
 - Our technique of using a *two-sided* kernel search appears new and may be applicable to other cryptosystems

Thank you! Questions?