Private information retrieval with a coding-theoretic perspective

Julien Lavauzelle

LAGA, Université Paris 8 Vincennes – Saint-Denis

Séminaire de l'équipe ECO

23/03/2022

1. Private information retrieval

2. PIR schemes with low computation and storage
   - Transversal designs and codes
   - A PIR scheme with transversal designs
   - Collusion-resistant PIR schemes with weighted lifted codes

3. PIR schemes for common storage systems
   - Distributed storage systems
   - A PIR scheme on RS-coded databases
   - A PIR scheme with regenerating codes

Problem statement

Private information retrieval (PIR):

Given a **remote** database $F \in \Sigma^M$ and $i \in [1, M]$,
can we **retrieve** the entry/file F_i,
without leaking information on the index i?

Private information retrieval (PIR):

Given a **remote** database $F \in \Sigma^M$ and $i \in [1, M]$,
can we **retrieve** the entry/file F_i,
without leaking information on the index i?

Appications: access to medical data, geoprivacy, ...

Private information retrieval (PIR):

> Given a **remote** database $F \in \Sigma^M$ and $i \in [1, M]$,
> can we **retrieve** the entry/file F_i,
> **without leaking** information on the index i?

Appications: access to medical data, geoprivacy, ...

Trivial solution: full download.

Definition of PIR

Introduced in:

Private Information Retrieval. Chor, Goldreich, Kushilevitz, Sudan. FOCS. **1995**.

A database F is stored (in some way) on n servers $S_1, \dots, S_n$.

Definition of PIR

Introduced in:

Private Information Retrieval. Chor, Goldreich, Kushilevitz, Sudan. FOCS. **1995**.

A database F is stored (in some way) on n servers $S_1, \dots, S_n$.
A user U wants to recover F_i privately.

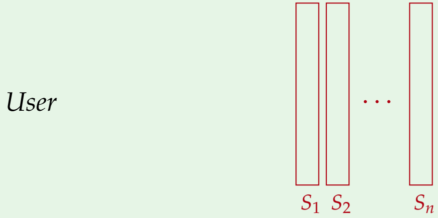
Definition of PIR

Introduced in:

Private Information Retrieval. Chor, Goldreich, Kushilevitz, Sudan. FOCS. **1995**.

A database F is stored (in some way) on n servers $S_1, \dots, S_n$.
A user U wants to recover F_i privately.

A **Private Information Retrieval protocol** is a set of algorithms $(\mathcal{Q}, \mathcal{A}, \mathcal{R})$:



Definition of PIR

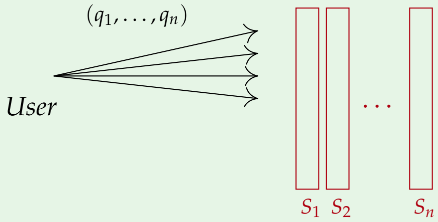
Introduced in:

Private Information Retrieval. Chor, Goldreich, Kushilevitz, Sudan. FOCS. **1995**.

A database F is stored (in some way) on n servers $S_1, \dots, S_n$.
A user U wants to recover F_i privately.

A **Private Information Retrieval protocol** is a set of algorithms $(\mathcal{Q}, \mathcal{A}, \mathcal{R})$:

1. U generates a query vector
   $$\boldsymbol{q} := (q_1, \dots, q_n) \leftarrow \mathcal{Q}(i)$$
   and sends q_j to server S_j.



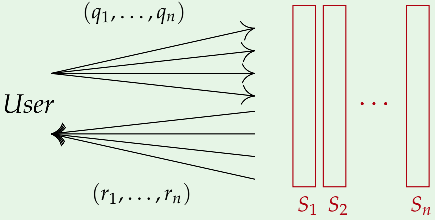
Introduced in:

Private Information Retrieval. Chor, Goldreich, Kushilevitz, Sudan. FOCS. **1995**.

A database F is stored (in some way) on n servers $S_1, \dots, S_n$.
A user U wants to recover F_i privately.

A **Private Information Retrieval protocol** is a set of algorithms $(\mathcal{Q}, \mathcal{A}, \mathcal{R})$:

1. U generates a query vector
$$\boldsymbol{q} := (q_1, \dots, q_n) \leftarrow \mathcal{Q}(i)$$
and sends q_j to server S_j.
2. Each server S_j computes
$$r_j := \mathcal{A}(q_j, F_{|S_j})$$
and sends it back to U.



Definition of PIR

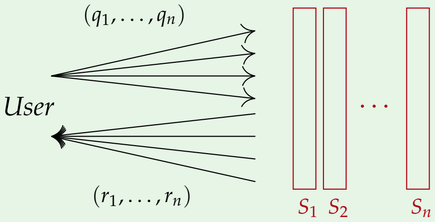
Introduced in:

Private Information Retrieval. Chor, Goldreich, Kushilevitz, Sudan. FOCS. **1995**.

A database F is stored (in some way) on n servers $S_1, \dots, S_n$.
A user U wants to recover F_i privately.

A **Private Information Retrieval protocol** is a set of algorithms $(\mathcal{Q}, \mathcal{A}, \mathcal{R})$:

1. U generates a query vector
   $$\boldsymbol{q} := (q_1, \dots, q_n) \leftarrow \mathcal{Q}(i)$$
   and sends q_j to server S_j.
2. Each server S_j computes
   $$r_j := \mathcal{A}(q_j, F_{|S_j})$$
   and sends it back to U.
3. U recovers the desired entry
   $$F_i = \mathcal{R}(\boldsymbol{q}, \boldsymbol{r}, i)\,.$$



Privacy

The adversary: a **collusion of servers** = a subset of servers $\{S_j : j \in T\}$, where $T \subset [1, n]$, which exchange information about queries, data, etc.

$$t := \max\{|T|, T \subseteq [1, n] \text{ is a collusion}\} \geq 1$$

The adversary: a **collusion of servers** = a subset of servers $\{S_j : j \in T\}$, where $T \subset [1, n]$, which exchange information about queries, data, etc.

$$t := \max\{|T|, T \subseteq [1, n] \text{ is a collusion}\} \geq 1$$

- **Information-theoretic (IT) privacy:**

$$\mathrm{I}(i\,; \boldsymbol{q}_{|T}) = 0, \quad \forall T \subseteq [1, n], |T| \leq t.$$

- **Computational privacy:** by varying the index i, distributions of queries $\boldsymbol{q}_{|T} = \mathcal{Q}(i)_{|T}$ are computationally indistinguishable.

The adversary: a **collusion of servers** = a subset of servers $\{S_j : j \in T\}$, where $T \subset [1, n]$, which exchange information about queries, data, etc.

$$t := \max\{|T|, T \subseteq [1, n] \text{ is a collusion}\} \geq 1$$

- **Information-theoretic (IT) privacy:**

$$\mathrm{I}(i\,; \boldsymbol{q}_{|T}) = 0, \quad \forall T \subseteq [1, n], |T| \leq t.$$

- **Computational privacy:** by varying the index i, distributions of queries $\boldsymbol{q}_{|T} = \mathcal{Q}(i)_{|T}$ are computationally indistinguishable.

Theorem [CGKS95, CG97]. If $t = n$ (in particular if $n = 1$ server), then:

- for **IT** privacy, **no better solution than full download**,
- **computational** privacy is possible, but remains **expensive** as of now.

We focus on **IT-privacy**
(hence we need $n \geq 2$ servers)

We focus on **IT-privacy**
(hence we need $n \geq 2$ servers)

Parameters to be taken into account:

- **communication** complexity (upload and download)

Main parameters of PIR schemes

We focus on **IT-privacy**
(hence we need $n \geq 2$ servers)

Parameters to be taken into account:

- **communication** complexity (upload and download)
- **computation** complexity (client and servers)

Main parameters of PIR schemes

We focus on **IT-privacy**
(hence we need $n \geq 2$ servers)

Parameters to be taken into account:

- **communication** complexity (upload and download)
- **computation** complexity (client and servers)
- global **server storage** overhead

We focus on **IT-privacy**
(hence we need $n \geq 2$ servers)

Parameters to be taken into account:

- **communication** complexity (upload and download)
- **computation** complexity (client and servers)
- global **server storage** overhead
- maximum size of collusions (t)

Main parameters of PIR schemes

We focus on **IT-privacy**
(hence we need $n \geq 2$ servers)

Parameters to be taken into account:

- **communication** complexity (upload and download)
- **computation** complexity (client and servers)
- global **server storage** overhead
- maximum size of collusions (t)

Several possible **settings**:

- **replicated** database vs. **coded** database

Main parameters of PIR schemes

We focus on **IT-privacy**
(hence we need $n \geq 2$ servers)

Parameters to be taken into account:

- **communication** complexity (upload and download)
- **computation** complexity (client and servers)
- global **server storage** overhead
- maximum size of collusions (t)

Several possible **settings**:

- **replicated** database vs. **coded** database
- unresponsive or **byzantine** servers

Main parameters of PIR schemes

We focus on **IT-privacy**
(hence we need $n \geq 2$ servers)

Parameters to be taken into account:

- **communication** complexity (upload and download)
- **computation** complexity (client and servers)
- global **server storage** overhead
- maximum size of collusions (t)

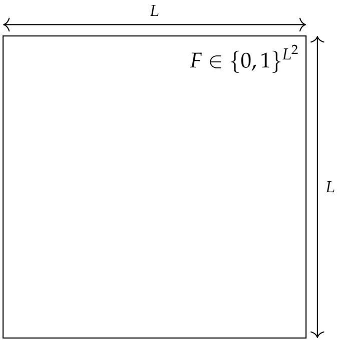
Several possible **settings**:

- **replicated** database vs. **coded** database
- unresponsive or **byzantine** servers
- small entries vs. large entries
- bounded vs. unbounded number of entries in the database
- dynamic database vs. static database

Private Information Retrieval. Chor, Goldreich, Kushilevitz, Sudan. FOCS. **1995**.

Settings: database F stored on n servers, where:

- $|F| = M$ entries (bits), with $M = L^2$, and $[1, M] \simeq [1, L]^2$.
- $n = 4$ servers $S_{00}, S_{01}, S_{10}, S_{11}$, each storing a replica of F.



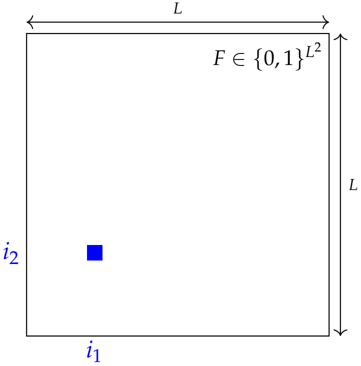
Seminal work [CGKS'95-98]

Private Information Retrieval. Chor, Goldreich, Kushilevitz, Sudan. FOCS. **1995**.

Settings: database F stored on n servers, where:

- $|F| = M$ entries (bits), with $M = L^2$, and $[1, M] \simeq [1, L]^2$.
- $n = 4$ servers $S_{00}, S_{01}, S_{10}, S_{11}$, each storing a replica of F.

Goal: retrieve $F_i = F_{(i_1, i_2)}$, for $1 \leq i_1, i_2 \leq L$.



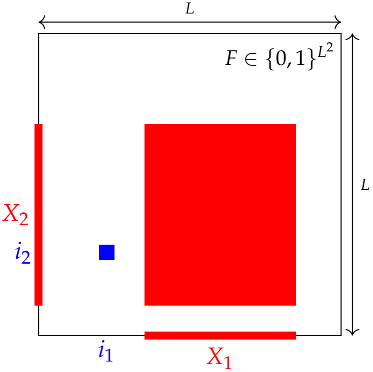
Seminal work [CGKS'95-98]

Private Information Retrieval. Chor, Goldreich, Kushilevitz, Sudan. FOCS. **1995**.

Settings: database F stored on n servers, where:

- $|F| = M$ entries (bits), with $M = L^2$, and $[1, M] \simeq [1, L]^2$.
- $n = 4$ servers $S_{00}, S_{01}, S_{10}, S_{11}$, each storing a replica of F.

Goal: retrieve $F_i = F_{(i_1, i_2)}$, for $1 \leq i_1, i_2 \leq L$.



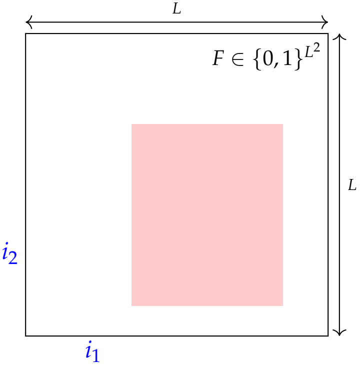
1. U generates at random two subsets X_1, X_2 of $[1, L]$. Then U sends:

Private Information Retrieval. Chor, Goldreich, Kushilevitz, Sudan. FOCS. **1995**.

Settings: database F stored on n servers, where:

- $|F| = M$ entries (bits), with $M = L^2$, and $[1, M] \simeq [1, L]^2$.
- $n = 4$ servers $S_{00}, S_{01}, S_{10}, S_{11}$, each storing a replica of F.

Goal: retrieve $F_i = F_{(i_1, i_2)}$, for $1 \leq i_1, i_2 \leq L$.



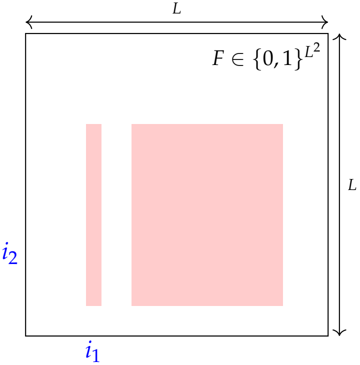
1. U generates at random two subsets X_1, X_2 of $[1, L]$. Then U sends:
   - (X_1 , X_2) to S_{00},

Private Information Retrieval. Chor, Goldreich, Kushilevitz, Sudan. FOCS. **1995**.

Settings: database F stored on n servers, where:

- $|F| = M$ entries (bits), with $M = L^2$, and $[1, M] \simeq [1, L]^2$.
- $n = 4$ servers $S_{00}, S_{01}, S_{10}, S_{11}$, each storing a replica of F.

Goal: retrieve $F_i = F_{(i_1, i_2)}$, for $1 \leq i_1, i_2 \leq L$.



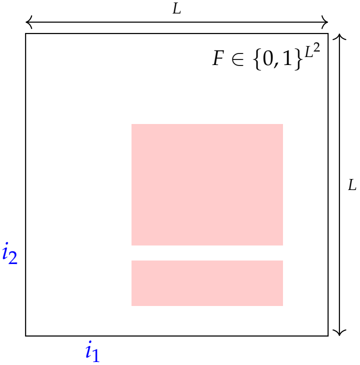
1. U generates at random two subsets X_1, X_2 of $[1, L]$. Then U sends:
   - (X_1, X_2) to S_{00},
   - $(X_1 \Delta \{i_1\}, X_2)$ to S_{10},

Private Information Retrieval. Chor, Goldreich, Kushilevitz, Sudan. FOCS. **1995**.

Settings: database F stored on n servers, where:

- $|F| = M$ entries (bits), with $M = L^2$, and $[1, M] \simeq [1, L]^2$.
- $n = 4$ servers $S_{00}, S_{01}, S_{10}, S_{11}$, each storing a replica of F.

Goal: retrieve $F_i = F_{(i_1, i_2)}$, for $1 \leq i_1, i_2 \leq L$.



1. U generates at random two subsets X_1, X_2 of $[1, L]$. Then U sends:
   - (X_1, X_2) to S_{00},
   - $(X_1 \Delta \{i_1\}, X_2)$ to S_{10},
   - $(X_1, X_2 \Delta \{i_2\})$ to S_{01},

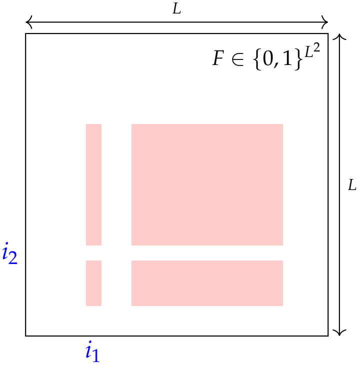
Seminal work [CGKS'95-98]

Private Information Retrieval. Chor, Goldreich, Kushilevitz, Sudan. FOCS. **1995**.

Settings: database F stored on n servers, where:

- $|F| = M$ entries (bits), with $M = L^2$, and $[1, M] \simeq [1, L]^2$.
- $n = 4$ servers $S_{00}, S_{01}, S_{10}, S_{11}$, each storing a replica of F.

Goal: retrieve $F_i = F_{(i_1, i_2)}$, for $1 \le i_1, i_2 \le L$.



1. U generates at random two subsets X_1, X_2 of $[1, L]$. Then U sends:
   - (X_1, X_2) to S_{00},
   - $(X_1 \Delta \{i_1\}, X_2)$ to S_{10},
   - $(X_1, X_2 \Delta \{i_2\})$ to S_{01},
   - $(X_1 \Delta \{i_1\}, X_2 \Delta \{i_2\})$ to S_{11}.

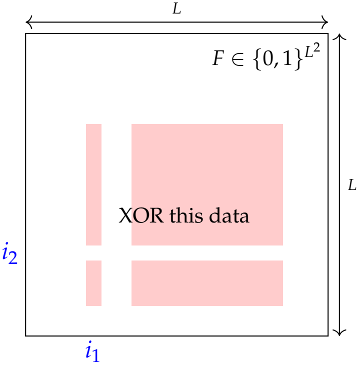
Seminal work [CGKS'95-98]

Private Information Retrieval. Chor, Goldreich, Kushilevitz, Sudan. FOCS. **1995**.

Settings: database F stored on n servers, where:

- $|F| = M$ entries (bits), with $M = L^2$, and $[1, M] \simeq [1, L]^2$.
- $n = 4$ servers $S_{00}, S_{01}, S_{10}, S_{11}$, each storing a replica of F.

Goal: retrieve $F_i = F_{(i_1,i_2)}$, for $1 \le i_1, i_2 \le L$.



1. U generates at random two subsets X_1, X_2 of $[1, L]$. Then U sends:
   - (X_1, X_2) to S_{00},
   - $(X_1 \Delta \{i_1\}, X_2)$ to S_{10},
   - $(X_1, X_2 \Delta \{i_2\})$ to S_{01},
   - $(X_1 \Delta \{i_1\}, X_2 \Delta \{i_2\})$ to S_{11}.
2. At reception of (Z_1, Z_2), each server computes $a = \bigoplus_{z \in Z_1 \times Z_2} F_z$ and sends a to the user.

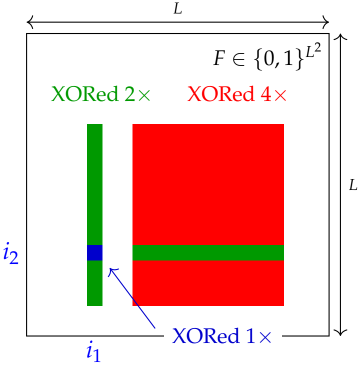
Seminal work [CGKS'95-98]

Private Information Retrieval. Chor, Goldreich, Kushilevitz, Sudan. FOCS. **1995**.

Settings: database F stored on n servers, where:

- $|F| = M$ entries (bits), with $M = L^2$, and $[1, M] \simeq [1, L]^2$.
- $n = 4$ servers $S_{00}, S_{01}, S_{10}, S_{11}$, each storing a replica of F.

Goal: retrieve $F_i = F_{(i_1, i_2)}$, for $1 \leq i_1, i_2 \leq L$.



1. U generates at random two subsets X_1, X_2 of $[1, L]$. Then U sends:
   - (X_1, X_2) to S_{00},
   - $(X_1 \Delta \{i_1\}, X_2)$ to S_{10},
   - $(X_1, X_2 \Delta \{i_2\})$ to S_{01},
   - $(X_1 \Delta \{i_1\}, X_2 \Delta \{i_2\})$ to S_{11}.
2. At reception of (Z_1, Z_2), each server computes $a = \bigoplus_{z \in Z_1 \times Z_2} F_z$ and sends a to the user.
3. User XORs the 4 bits and retrieves F_i.

Correct, and **secure** if no collusion.

Features of the PIR scheme in [CGKS'95-98]

Correct, and **secure** if no collusion.

With $n = 4$ servers:

- **Communication:** $8\sqrt{M}$ uploaded bits, 4 downloaded bits,
- **Storage:** replication of F over $n = 4$ servers,
- **Complexity:**
  - for each server: in average, XOR of $(L/2)^2 = M/4$ bits
  - for the user: XOR of $n = 4$ bits.

Features of the PIR scheme in [CGKS'95-98]

Correct, and **secure** if no collusion.

With $n = 4$ servers:

- **Communication:** $8\sqrt{M}$ uploaded bits, 4 downloaded bits,
- **Storage:** replication of F over $n = 4$ servers,
- **Complexity:**
  - for each server: in average, XOR of $(L/2)^2 = M/4$ bits
  - for the user: XOR of $n = 4$ bits.

Generalisable to $n = 2^b$ servers:

- **Communication:** $b2^b M^{1/b} = n\log(n) M^{1/\log(n)}$ uploaded bits, n downloaded bits,
- **Storage:** replication of F over n servers,
- **Complexity:**
  - for each server: in average, XOR of M/n bits
  - for the user: XOR of n bits.

(Short) state of the art

- 1995: first definition [CGKS95]
- 2000: reduction from smooth locally decodable codes [KT00]
- 2000-10's: many improvements
  - PIR with 3 servers and subpolynomial communication [Yek08, Efr09]
  - PIR with 2 servers and subpolynomial communication [DG16]
  - lower storage overhead with *PIR codes* [FVY15]
- 2016-now: capacity-achieving schemes, schemes dedicated to storage systems
  - capacity of PIR [SJ17, BU18]
  - (nearly) capacity-achieving schemes [SRR14, CHY15, TR16, ...]

1. Private information retrieval

2. PIR schemes with low computation and storage
   - Transversal designs and codes
   - A PIR scheme with transversal designs
   - Collusion-resistant PIR schemes with weighted lifted codes

3. PIR schemes for common storage systems
   - Distributed storage systems
   - A PIR scheme on RS-coded databases
   - A PIR scheme with regenerating codes

Context

Previous scheme:

- moderate communication complexity
- computationally inefficient (linear in $|F|$)
- huge storage overhead (replicas of $|F|$)

Context

Previous scheme:

- moderate communication complexity
- computationally inefficient (linear in $|F|$)
- huge storage overhead (replicas of $|F|$)

Our goal:

- moderate communication complexity
- **optimal computation** (one read for each server)
- **smaller storage overhead** by encoding/distributing the database

Context

Previous scheme:

- moderate communication complexity
- computationally inefficient (linear in $|F|$)
- huge storage overhead (replicas of $|F|$)

Our goal:

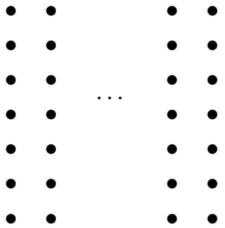
- moderate communication complexity
- **optimal computation** (one read for each server)
- **smaller storage overhead** by encoding/distributing the database

Tools: coding theory

- codes from transversal designs
- “lifted” codes

A **transversal design** $\mathrm{TD}(n, s) = (X, \mathcal{B}, \mathcal{G})$ is given by:

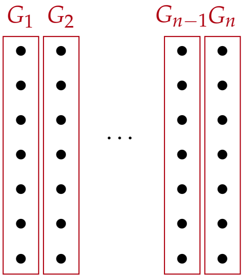
- X a set of *points*, $|X| = N = ns$,



A **transversal design** $\mathrm{TD}(n, s) = (X, \mathcal{B}, \mathcal{G})$ is given by:

- X a set of *points*, $|X| = N = ns$,
- *groups* $\mathcal{G} = \{G_j\}_{1 \le j \le n}$ satisfying

$$X = \coprod_{j=1}^{n} G_j \quad \text{and} \quad |G_j| = s\,,$$



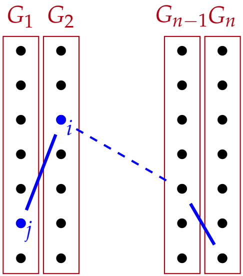
Transversal designs

A **transversal design** $\mathrm{TD}(n,s) = (X, \mathcal{B}, \mathcal{G})$ is given by:

- X a set of *points*, $|X| = N = ns$,
- *groups* $\mathcal{G} = \{G_j\}_{1 \le j \le n}$ satisfying

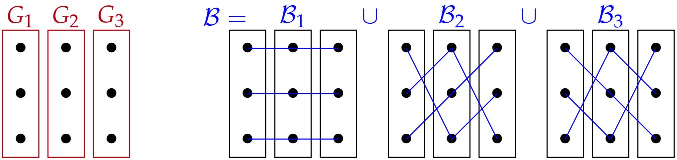
$$X = \coprod_{j=1}^{n} G_j \text{ and } |G_j| = s\,,$$

- *blocks* $B \in \mathcal{B}$ satisfying
  - $B \subset X$ and $|B| = n$;
  - for all $\{i,j\} \subset X$, $\{i,j\}$ lie:
    either in a single group $G \in \mathcal{G}$,
    or in a unique block $B \in \mathcal{B}$



An example for a $\mathrm{TD}(3,3)$:

- $ns = 9$ points
- $s = 3$ groups G_1, G_2, G_3 of size 3
- $ns = 9$ blocks of $n = 3$ points, partitionned into 3 parallel classes $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3$



Codes from designs

Let $\mathcal{T}$ be a transversal design $\mathrm{TD}(n, s) = (X, \mathcal{B}, \mathcal{G})$.

Its **incidence matrix** $\boldsymbol{M}$ has size $|\mathcal{B}| \times |X| = ns \times ns$, and is defined by:

$$M_{i,j} = \begin{cases} 1 & \text{if } x_j \in B_i \\ 0 & \text{otherwise.} \end{cases}$$

Codes from designs

Let $\mathcal{T}$ be a transversal design $\mathrm{TD}(n, s) = (X, \mathcal{B}, \mathcal{G})$.

Its **incidence matrix** $\boldsymbol{M}$ has size $|\mathcal{B}| \times |X| = ns \times ns$, and is defined by:

$$M_{i,j} = \begin{cases} 1 & \text{if } x_j \in B_i \\ 0 & \text{otherwise.} \end{cases}$$

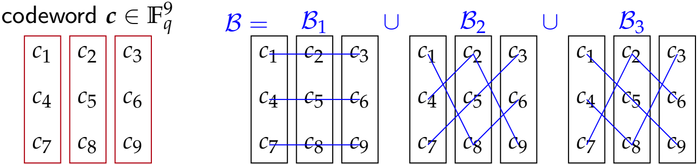
Definition. The **code $\mathcal{C}$ based on $\mathcal{T}$ over $\mathbb{F}_q$** is the $\mathbb{F}_q$-linear code having $\boldsymbol{M}$ as a parity-check matrix (*i.e.* $\mathcal{C}^\perp$ is generated by $\boldsymbol{M}$).

- $\mathrm{length}(\mathcal{C}) = |X| = ns$,
- $\dim(\mathcal{C}) = \dim(\ker \boldsymbol{M})$,
- every block $B \in \mathcal{B}$ gives a parity-check equation $\boldsymbol{h} \in \mathcal{C}^\perp$, such that

$$\mathrm{wt}(\boldsymbol{h}_{|G_j}) = 1, \quad \forall j = 1, \dots, n$$

Example

The transversal design TD$(3,3)$ represented by:

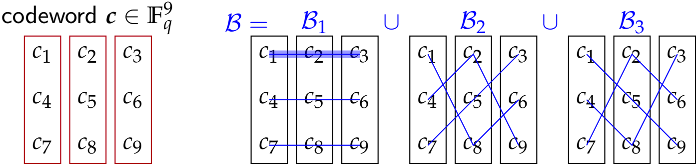


gives a code with the following parity-check matrix:

$$
\boldsymbol{H} = \begin{pmatrix}
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\
1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0
\end{pmatrix}
$$

Example

The transversal design $\mathrm{TD}(3,3)$ represented by:

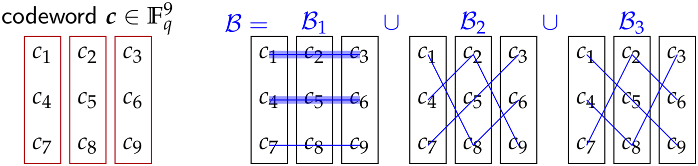


gives a code with the following parity-check matrix:

$$\boldsymbol{H} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Example

The transversal design $\mathrm{TD}(3,3)$ represented by:



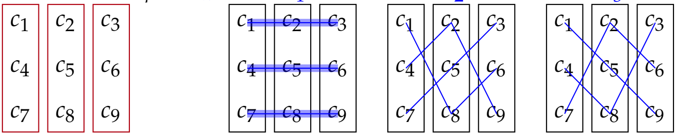
gives a code with the following parity-check matrix:

$$\boldsymbol{H} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Example

The transversal design $\mathrm{TD}(3,3)$ represented by:

codeword $\boldsymbol{c} \in \mathbb{F}_q^9$ $\quad\quad$ $\mathcal{B} = \mathcal{B}_1 \cup \mathcal{B}_2 \cup \mathcal{B}_3$

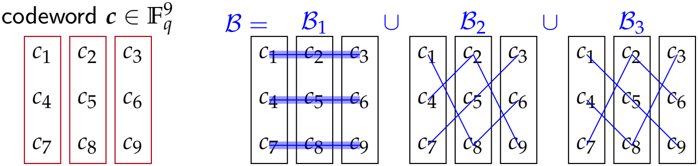


gives a code with the following parity-check matrix:

$$
\boldsymbol{H} = \begin{pmatrix}
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\
1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0
\end{pmatrix}
$$

Example

The transversal design $\mathrm{TD}(3,3)$ represented by:



gives a code with the following parity-check matrix:

$$\boldsymbol{H} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Dimension of the code?

- depends on q
- for instance, over $\mathbb{F}_3$, we have $\mathrm{rk}(\boldsymbol{H}) = 6 \quad \dim(\mathcal{C}) = 3$.

Outline

The PIR scheme

Private Information Retrieval from Transversal Designs. **L.**. IEEE-TIT. **2019**.

Let $\mathcal{C} \subseteq \mathbb{F}_q^N$ be a code based on a $\mathrm{TD}(n, s)$, with $N = ns$.

Private Information Retrieval from Transversal Designs. **L.**. IEEE-TIT. **2019**.

Let $\mathcal{C} \subseteq \mathbb{F}_q^N$ be a code based on a $\mathrm{TD}(n, s)$, with $N = ns$.

- **Initialisation.** User U encodes $F \mapsto \boldsymbol{c} \in \mathcal{C}$, and gives $\boldsymbol{c}_{|G_j}$ to server S_j.

Private Information Retrieval from Transversal Designs. **L.**. IEEE-TIT. **2019**.

Let $\mathcal{C} \subseteq \mathbb{F}_q^N$ be a code based on a $\mathrm{TD}(n, s)$, with $N = ns$.

- **Initialisation.** User U encodes $F \mapsto \boldsymbol{c} \in \mathcal{C}$, and gives $\boldsymbol{c}_{|G_j}$ to server S_j.
- **To recover** $F_i = c_i$, with $i \in X$:
  1. User U randomly picks a block $B \in \mathcal{B}$ containing i.
     Then U defines:

$$q_j = \mathcal{Q}(i)_j := \begin{cases} \text{unique } \in B \cap G_j & \text{if } i \notin G_j \\ \text{a random point in } G_j & \text{otherwise.} \end{cases}$$

Private Information Retrieval from Transversal Designs. **L.**. IEEE-TIT. **2019**.

Let $\mathcal{C} \subseteq \mathbb{F}_q^N$ be a code based on a $\mathrm{TD}(n, s)$, with $N = ns$.

- **Initialisation.** User U encodes $F \mapsto \boldsymbol{c} \in \mathcal{C}$, and gives $\boldsymbol{c}_{|G_j}$ to server S_j.
- **To recover** $F_i = c_i$, with $i \in X$:
  1. User U randomly picks a block $B \in \mathcal{B}$ containing i.
     Then U defines:
     $$q_j = \mathcal{Q}(i)_j := \begin{cases} \text{unique } \in B \cap G_j & \text{if } i \notin G_j \\ \text{a random point in } G_j & \text{otherwise.} \end{cases}$$
  2. Each server S_j sends back c_{q_j}

Private Information Retrieval from Transversal Designs. **L.**. IEEE-TIT. **2019**.

Let $\mathcal{C} \subseteq \mathbb{F}_q^N$ be a code based on a $\mathrm{TD}(n, s)$, with $N = ns$.

- **Initialisation.** User U encodes $F \mapsto \boldsymbol{c} \in \mathcal{C}$, and gives $\boldsymbol{c}_{|G_j}$ to server S_j.
- **To recover** $F_i = c_i$, with $i \in X$:
  1. User U randomly picks a block $B \in \mathcal{B}$ containing i.
     Then U defines:

$$q_j = \mathcal{Q}(i)_j := \begin{cases} \text{unique } \in B \cap G_j & \text{if } i \notin G_j \\ \text{a random point in } G_j & \text{otherwise.} \end{cases}$$

  2. Each server S_j sends back c_{q_j}
  3. U recovers

$$c_i = -\sum_{j:\, i \notin G_j} c_{q_j} = -\sum_{b \in B \setminus \{i\}} c_b$$

Privacy and parameters

Theorem. This PIR protocol is information-theoretically private.

Proof:

- the only server which holds F_i received a random query;
- for each other server S_j, query q_j gives no information on the block B which has been picked $\Rightarrow$ no information leaks on i.

Theorem. This PIR protocol is information-theoretically private.

Proof:

- the only server which holds F_i received a random query;
- for each other server S_j, query q_j gives no information on the block B which has been picked $\Rightarrow$ no information leaks on i.

Features.

- communication complexity: $n \log s$ uploaded bits, $n \log q$ downloaded bits

Theorem. This PIR protocol is information-theoretically private.

Proof:

- the only server which holds F_i received a random query;
- for each other server S_j, query q_j gives no information on the block B which has been picked $\Rightarrow$ no information leaks on i.

Features.

- communication complexity: $n \log s$ uploaded bits, $n \log q$ downloaded bits
- computational complexity:
  - **only 1 read for each server** (optimal)
  - $\leq n$ additions over $\mathbb{F}_q$ for the user

Theorem. This PIR protocol is information-theoretically private.

Proof:

- the only server which holds F_i received a random query;
- for each other server S_j, query q_j gives no information on the block B which has been picked $\Rightarrow$ no information leaks on i.

Features.

- communication complexity: $n \log s$ uploaded bits, $n \log q$ downloaded bits
- computational complexity:
  - **only 1 read for each server** (optimal)
  - $\leq n$ additions over $\mathbb{F}_q$ for the user
- storage overhead: $(ns - k) \log q$ bits, where $k = \dim(\mathcal{C})$

Privacy and parameters

Theorem. This PIR protocol is information-theoretically private.

Proof:

- the only server which holds F_i received a random query;
- for each other server S_j, query q_j gives no information on the block B which has been picked $\Rightarrow$ no information leaks on i.

Features.

- communication complexity: $n \log s$ uploaded bits, $n \log q$ downloaded bits
- computational complexity:
  - **only 1 read for each server** (optimal)
  - $\leq n$ additions over $\mathbb{F}_q$ for the user
- storage overhead: $(ns - k) \log q$ bits, where $k = \dim(\mathcal{C})$

Question: transversal designs leading to large dimension codes?

An example: the **classical affine transversal design**:

- $X = \mathbb{F}_q^m$ for $m \geq 2$,
- $\mathcal{G}$ a partition of X into q hyperplanes $G_1, \dots, G_q$,
- $\mathcal{B} = \{\text{affine lines } L \text{ secant to each } G_j\}$.

The code has:

- length $ns = q^m$,
- "locality" $n = q$.

Instances with geometric designs

An example: the **classical affine transversal design**:

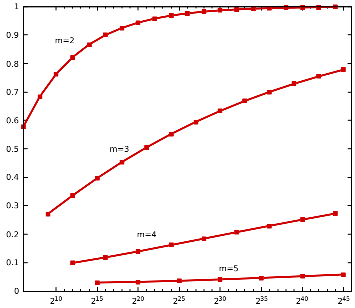
- $X = \mathbb{F}_q^m$ for $m \geq 2$,
- $\mathcal{G}$ a partition of X into q hyperplanes $G_1, \dots, G_q$,
- $\mathcal{B} = \{$affine lines L secant to each $G_j\}$.

The code has:

– length $ns = q^m$,

– "locality" $n = q$.

rate k/N



length $N = ns = 2^{em}$

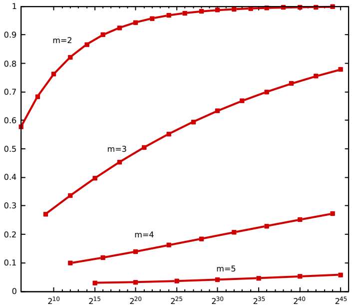
An example: the **classical affine transversal design**:

- $X = \mathbb{F}_q^m$ for $m \geq 2$,
- $\mathcal{G}$ a partition of X into q hyperplanes $G_1, \dots, G_q$,
- $\mathcal{B} = \{$affine lines L secant to each $G_j\}$.

The code has:

- length $ns = q^m$,
- "locality" $n = q$.

rate k/N



length $N = ns = 2^{em}$

Question: how to deal with collusions and byzantine errors?

Definition. The (full-length) **Reed–Solomon code** of dimension k over $\mathbb{F}_q$ is:

$$\mathrm{RS}_q(k) := \{\mathrm{ev}_{\mathbb{A}^1}(f) := (f(x_1), \dots, f(x_q)) \mid \deg(f) \leq k-1\}.$$

Definition. The (full-length) **Reed–Solomon code** of dimension k over $\mathbb{F}_q$ is:

$$\mathrm{RS}_q(k) := \{\mathrm{ev}_{\mathbb{A}^1}(f) := (f(x_1),\dots,f(x_q)) \mid \deg(f) \leq k-1\}\,.$$

The code $\mathcal{C} = \mathrm{RS}_q(k)$ is **MDS**: every codeword $\boldsymbol{c} \in \mathcal{C}$ can be reconstructed from any k-subset of coordinates of $\boldsymbol{c}$.

Definition. The (full-length) **Reed–Solomon code** of dimension k over $\mathbb{F}_q$ is:

$$\mathrm{RS}_q(k) := \{\mathrm{ev}_{\mathbb{A}^1}(f) := (f(x_1), \dots, f(x_q)) \mid \deg(f) \leq k-1\}\,.$$

The code $\mathcal{C} = \mathrm{RS}_q(k)$ is **MDS**: every codeword $\boldsymbol{c} \in \mathcal{C}$ can be reconstructed from any k-subset of coordinates of $\boldsymbol{c}$.

Definition. The **Reed–Muller code** of order m and degree r over $\mathbb{F}_q$ is:

$$\mathrm{RM}_q(m,r) := \{\mathrm{ev}_{\mathbb{A}^m}(f) \mid f \in \mathbb{F}_q[\boldsymbol{X}] \text{ and } \deg(f) \leq r\}\,.$$

Definition. The (full-length) **Reed–Solomon code** of dimension k over $\mathbb{F}_q$ is:

$$\mathrm{RS}_q(k) := \{\mathrm{ev}_{\mathbb{A}^1}(f) := (f(x_1),\dots,f(x_q)) \mid \deg(f) \le k-1\}\,.$$

The code $\mathcal{C} = \mathrm{RS}_q(k)$ is **MDS**: every codeword $\boldsymbol{c} \in \mathcal{C}$ can be reconstructed from any k-subset of coordinates of $\boldsymbol{c}$.

Definition. The **Reed–Muller code** of order m and degree r over $\mathbb{F}_q$ is:

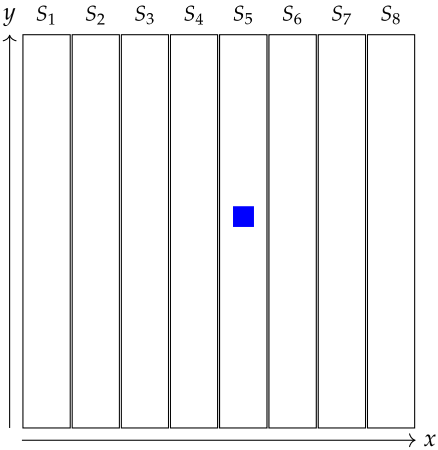
$$\mathrm{RM}_q(m,r) := \{\mathrm{ev}_{\mathbb{A}^m}(f) \mid f \in \mathbb{F}_q[\boldsymbol{X}] \text{ and } \deg(f) \le r\}\,.$$

Reed–Muller codes have the following property:

$$\forall \boldsymbol{c} = \mathrm{ev}_{\mathbb{A}^m}(f) \in \mathrm{RM}_q(m,r) \quad \text{and} \quad \forall \text{ affine line } L \subset \mathbb{A}^m,$$
$$\mathrm{ev}_{\mathbb{A}^1}(f_{|L}) \in \mathrm{RS}_q(r+1)\,.$$

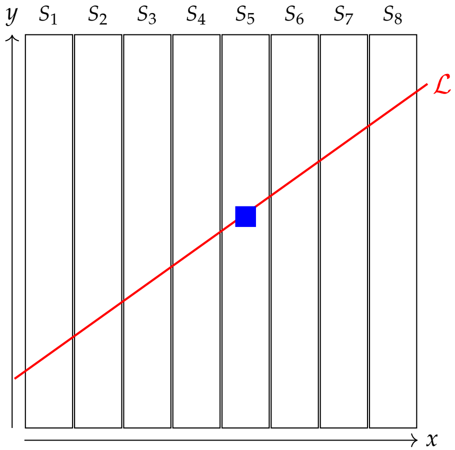
(where $f_{|L}$ is the lowest-degree univariate polynomial interpolating f over L)

A PIR scheme based on Reed–Muller codes



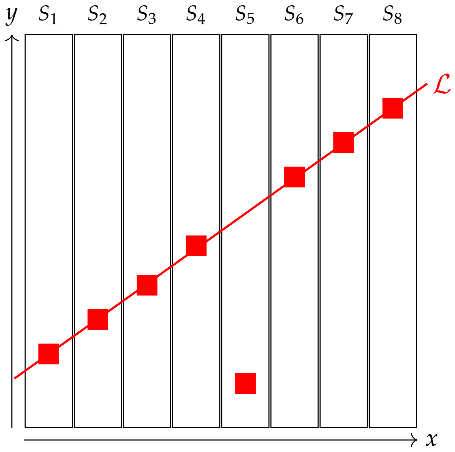
Database F in encoded with $\mathrm{RM}_q(m, r)$, then distributed across the servers

A PIR scheme based on Reed–Muller codes



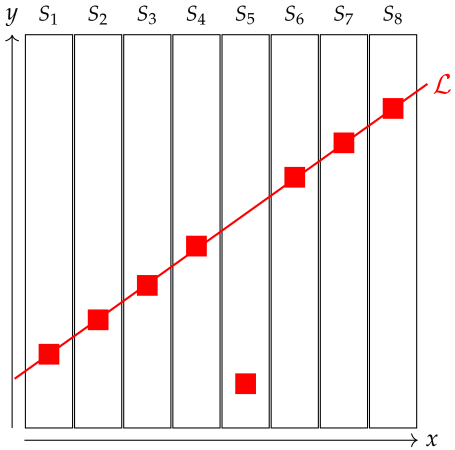
Database F in encoded with $\mathrm{RM}_q(m,r)$, then distributed across the servers

A PIR scheme based on Reed–Muller codes

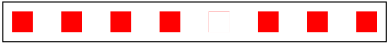


Database F in encoded with $\mathrm{RM}_q(m, r)$, then distributed across the servers

A PIR scheme based on Reed–Muller codes

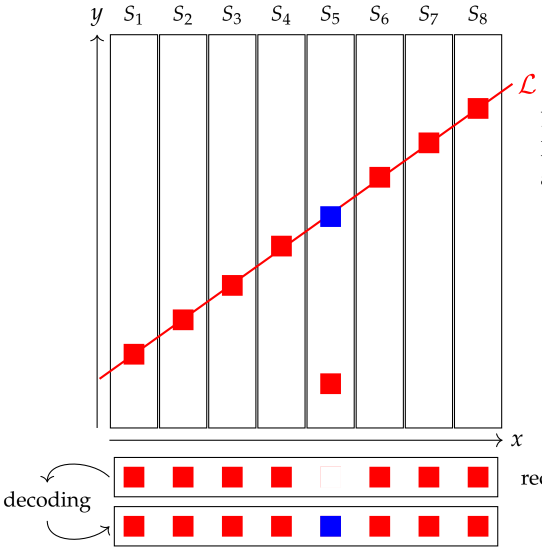


Database F in encoded with $\mathrm{RM}_q(m,r)$, then distributed across the servers



received answer

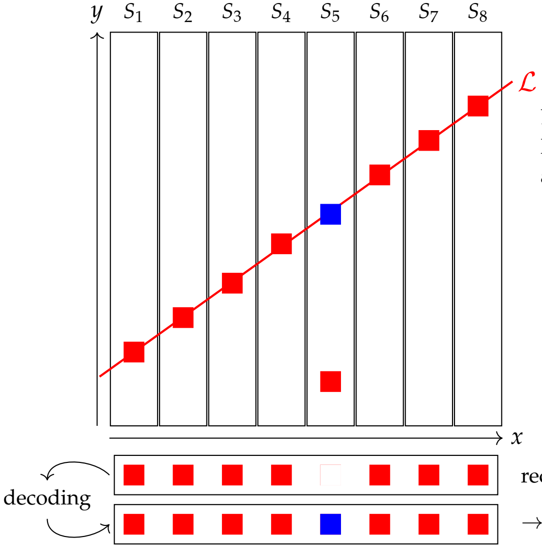
A PIR scheme based on Reed–Muller codes



Database F in encoded with $\mathrm{RM}_q(m,r)$, then distributed across the servers

received answer

A PIR scheme based on Reed–Muller codes

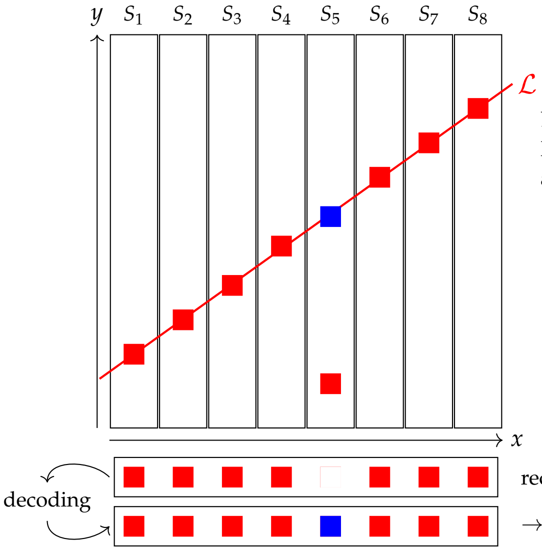


Database F in encoded with $\mathrm{RM}_q(m, r)$, then distributed across the servers

received answer

$\to$ output F_i

A PIR scheme based on Reed–Muller codes



Database F in encoded with $\mathrm{RM}_q(m,r)$, then distributed across the servers

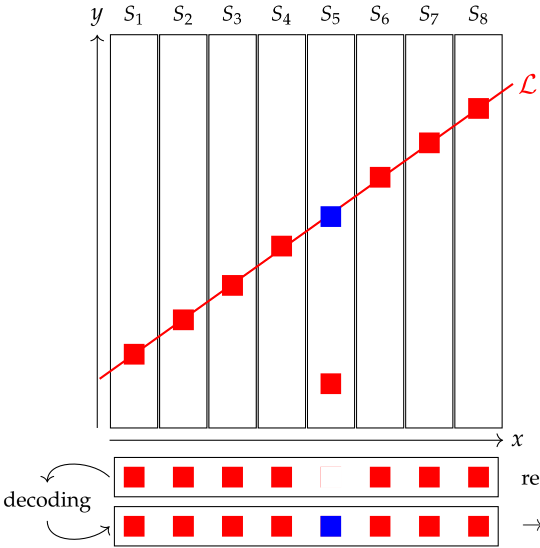
$\mathcal{L}$ is an affine line

$\Longrightarrow$ no individual server can find where is ■

received answer

$\to$ output F_i

A PIR scheme based on Reed–Muller codes



Database F in encoded with $\mathrm{RM}_q(m,r)$, then distributed across the servers

$\mathcal{L}$ is an affine line

$\implies$ no individual server can find where is ■

$\implies$ the PIR scheme is private

Parameter of the Reed–Muller code-based PIR

Features with $\mathrm{RM}_q(m, r)$ of length q^m.

- communication complexity: $(m-1)q \log q$ uploaded bits, $q \log q$ downloaded bits
- computational complexity:
  - **only 1 read for each server** (optimal)
  - a decoding procedure for $\mathrm{RS}_q(r)$ for the user
- storage overhead: the rate of $\mathrm{RM}_q(m, r)$ with $r \leq q-1$ is

$$\simeq \frac{(r/q)^m}{m!} \ldots$$

$\Longrightarrow$ We need codes with the same properties, but larger dimension.

Lifted codes

"Lifted" codes are the largest codes having the same property as Reed–Muller codes.

New affine-invariant codes from lifting. Guo, Kopparty, Sudan. ITCS. **2013**.

"Lifted" codes are the largest codes having the same property as Reed–Muller codes.

New affine-invariant codes from lifting. Guo, Kopparty, Sudan. ITCS. **2013**.

Definition. The ***m*-th lifted Reed-Solomon code** of degree r over $\mathbb{F}_q$ is:

$$\mathrm{Lift}_q(m,r) := \{\mathrm{ev}_{\mathbb{A}^m}(f) \mid f \in \mathbb{F}_q[\mathbf{X}] \text{ and } \forall \text{ affine line } L \subset \mathbb{A}^m, \deg(f_{|L}) \leq r\}.$$

"Lifted" codes are the largest codes having the same property as Reed–Muller codes.

New affine-invariant codes from lifting. Guo, Kopparty, Sudan. ITCS. **2013**.

Definition. The **m-th lifted Reed-Solomon code** of degree r over $\mathbb{F}_q$ is:

$$\mathrm{Lift}_q(m,r) := \{\mathrm{ev}_{\mathbb{A}^m}(f) \mid f \in \mathbb{F}_q[\mathbf{X}] \text{ and } \forall \text{ affine line } L \subset \mathbb{A}^m, \deg(f_{|L}) \le r\}.$$

Lifted codes contain Reed-Muller codes, **sometimes properly**.

"Lifted" codes are the largest codes having the same property as Reed–Muller codes.

New affine-invariant codes from lifting. Guo, Kopparty, Sudan. ITCS. **2013**.

Definition. The **m-th lifted Reed-Solomon code** of degree r over $\mathbb{F}_q$ is:

$$\mathrm{Lift}_q(m,r) := \{\mathrm{ev}_{\mathbb{A}^m}(f) \mid f \in \mathbb{F}_q[\mathbf{X}] \text{ and } \forall \text{ affine line } L \subset \mathbb{A}^m, \deg(f_{|L}) \le r\}\,.$$

Lifted codes contain Reed-Muller codes, **sometimes properly**.

Example. For $q = 4$, $m = 2$, $r = 2$, consider $f(X,Y) = X^2Y^2$.

$$f(aT+b, cT+d) \;\equiv\; (a^2d^2 + b^2c^2)T^2 + a^2c^2T + b^2d^2 \mod (T^4 - T)$$

Hence,

$$\mathrm{ev}(X^2Y^2) \in \mathrm{Lift}_4(2,2) \quad \text{but} \quad \mathrm{ev}(X^2Y^2) \notin \mathrm{RM}_4(2,2)\,.$$

"Lifted" codes are the largest codes having the same property as Reed–Muller codes.

New affine-invariant codes from lifting. Guo, Kopparty, Sudan. ITCS. **2013**.

Definition. The ***m*-th lifted Reed-Solomon code** of degree r over $\mathbb{F}_q$ is:

$$\mathrm{Lift}_q(m,r) := \{\mathrm{ev}_{\mathbb{A}^m}(f) \mid f \in \mathbb{F}_q[\mathbf{X}] \text{ and } \forall \text{ affine line } L \subset \mathbb{A}^m, \deg(f_{|L}) \leq r\}\,.$$

Lifted codes contain Reed-Muller codes, **sometimes properly**.

Example. For $q = 4$, $m = 2$, $r = 2$, consider $f(X,Y) = X^2Y^2$.

$$f(aT+b, cT+d) \;\equiv\; (a^2d^2 + b^2c^2)T^2 + a^2c^2T + b^2d^2 \mod (T^4 - T)$$

Hence,

$$\mathrm{ev}(X^2Y^2) \in \mathrm{Lift}_4(2,2) \quad \text{but} \quad \mathrm{ev}(X^2Y^2) \notin \mathrm{RM}_4(2,2)\,.$$

Fact. For every m, lifted codes reach arbitrarily large information rates.

Weighted lifted codes

Question: how to deal with collusions and byzantine errors?

Question: how to deal with collusions and byzantine errors?

For convenience, here $m = 2$.

Definition. A t-curve is:

$$\mathcal{L} = \{(x, g(x)) \in \mathbb{A}^2 \mid g \in \mathbb{F}_q[X], \deg(g) \leq t\}$$

Weighted lifted codes

Question: how to deal with collusions and byzantine errors?

For convenience, here $m = 2$.

Definition. A t-curve is:

$$\mathcal{L} = \{(x, g(x)) \in \mathbb{A}^2 \mid g \in \mathbb{F}_q[X], \deg(g) \leq t\}$$

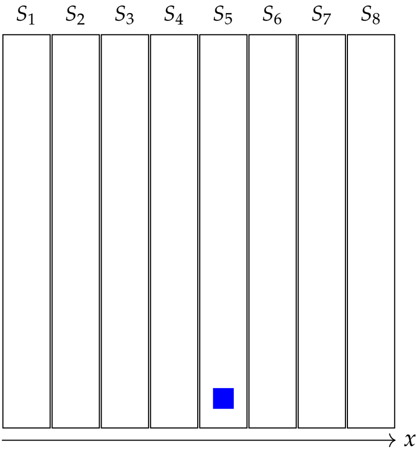
Definition. The **weighted lifted Reed-Solomon code** of degree r and weight t over $\mathbb{F}_q$ is:

$$\mathrm{WLift}_q(t, r) := \{\mathrm{ev}_{\mathbb{A}^2}(f) \mid f \in \mathbb{F}_q[X, Y] \text{ and } \forall\ t\text{-curve } \mathcal{L} \subset \mathbb{A}^2, \deg(f_{|\mathcal{L}}) \leq r\}$$

Consequence: for every codeword $\boldsymbol{c} \in \mathrm{WLift}_q(t, r)$ and every t-curve $\mathcal{L}$, we have:

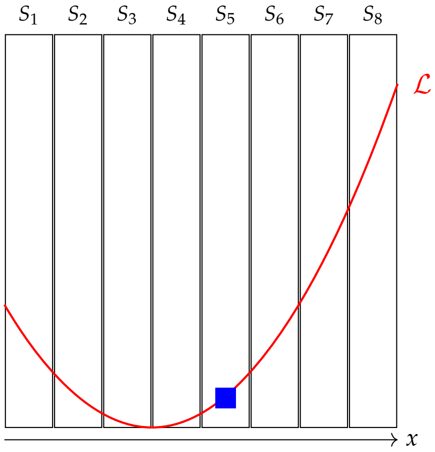
$$\boldsymbol{c}_{|\mathcal{L}} \in \mathrm{RS}_q(r + 1)\,.$$

A PIR scheme based on weighted lifted codes



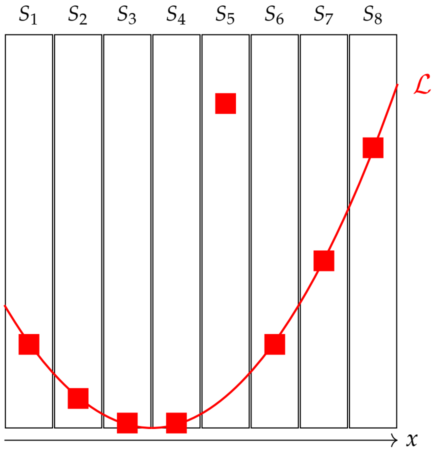
Database F in encoded with $\mathrm{WLift}_q(t, r)$, then distributed across the servers

A PIR scheme based on weighted lifted codes



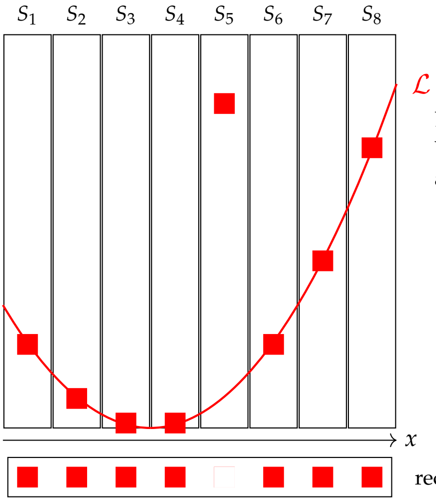
Database F in encoded with $\mathrm{WLift}_q(t, r)$, then distributed across the servers

A PIR scheme based on weighted lifted codes



Database F in encoded with $\mathrm{WLift}_q(t,r)$, then distributed across the servers

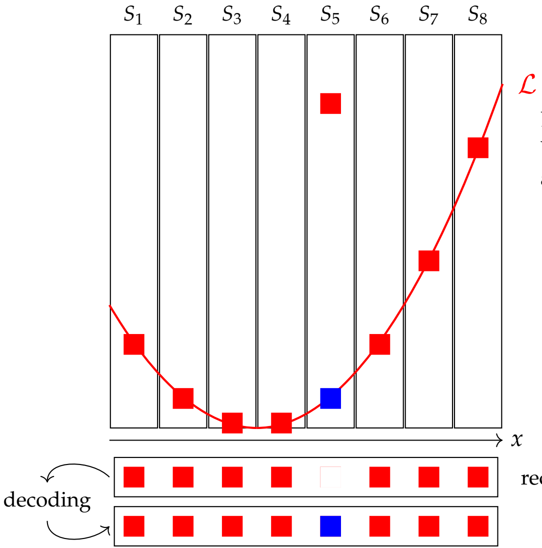
A PIR scheme based on weighted lifted codes



Database F in encoded with $\mathrm{WLift}_q(t, r)$, then distributed across the servers

received answer

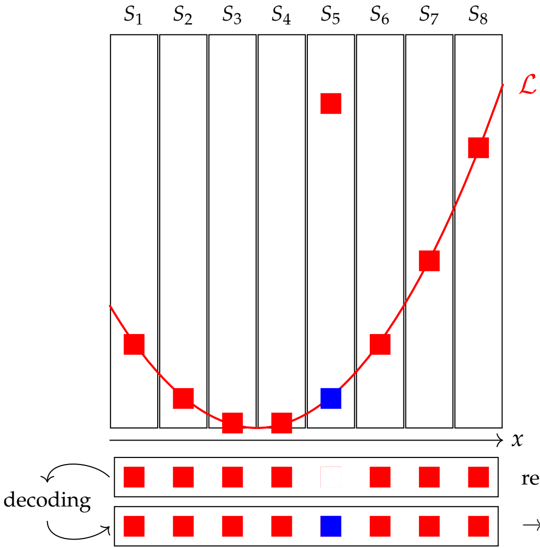
A PIR scheme based on weighted lifted codes



Database F in encoded with $\mathrm{WLift}_q(t, r)$, then distributed across the servers

received answer

A PIR scheme based on weighted lifted codes

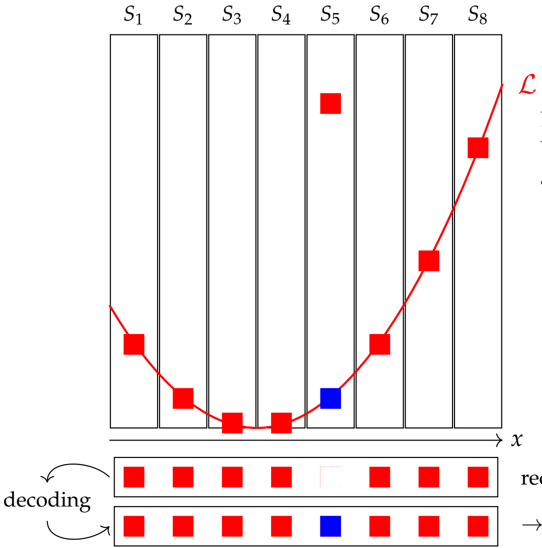


Database F in encoded with $\mathrm{WLift}_q(t,r)$, then distributed across the servers

received answer

$\to$ output F_i

A PIR scheme based on weighted lifted codes



Database F in encoded with $\mathrm{WLift}_q(t, r)$, then distributed across the servers

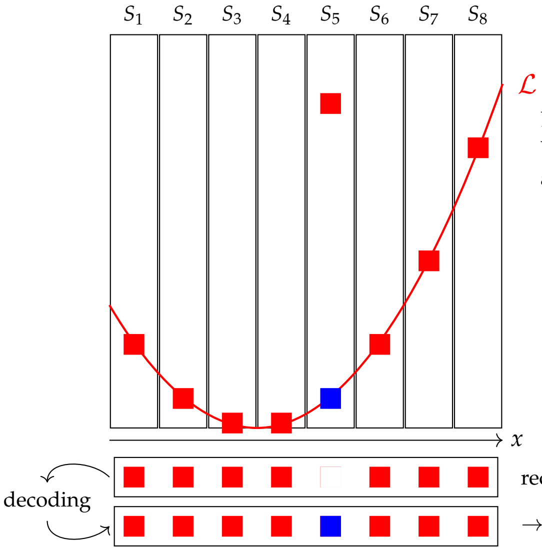
$\mathcal{L}$ is a t-curve

$\implies$ no t-set of servers can find where is ■

received answer

$\rightarrow$ output F_i

A PIR scheme based on weighted lifted codes



Database F in encoded with $\mathrm{WLift}_q(t, r)$, then distributed across the servers

$\mathcal{L}$ is a t-curve

$\Longrightarrow$ no t-set of servers can find where is ■

$\Longrightarrow$ the PIR scheme is t-private

received answer

$\to$ output F_i

Parameters

Weighted Lifted Codes: Local Correctabilities and Application to Robust Private Information Retrieval. **L.**, Nardi. IEEE TIT. **2021**.

Theorem. Let p be a prime number, $t \geq 1$ and $\alpha \geq 2$ be fixed integers. Then, the information rate $\mathrm{WLift}_{p^e}(t, p^e - \alpha)$ grows to 1 when $e \to \infty$.

Corollary: we get PIR schemes with relative storage overhead $\to 0$, for a constant number of adversaries.

Weighted Lifted Codes: Local Correctabilities and Application to Robust Private Information Retrieval. **L.**, Nardi. IEEE TIT. **2021**.

Theorem. Let p be a prime number, $t \geq 1$ and $\alpha \geq 2$ be fixed integers. Then, the information rate $\mathrm{WLift}_{p^e}(t, p^e - \alpha)$ grows to 1 when $e \to \infty$.

Corollary: we get PIR schemes with relative storage overhead $\to 0$, for a constant number of adversaries.

Theorem. Let p be a prime number, $t \geq 1$ and $c \geq 1$ be fixed integers. Let $\gamma = 1 - p^{-c}$ and $\mathcal{C}_e = \mathrm{WLift}_{p^e}(t, \gamma p^e)$. Then, the information rate R_e of $\mathcal{C}_e$ satisfies:

$$\lim_{e \to \infty} R_e = K_{t,p,c} > 0$$

Corollary: we get PIR schemes with **constant relative storage overhead**, for a **constant number of collusions** and a **constant fraction of errors**.

Outline

Context

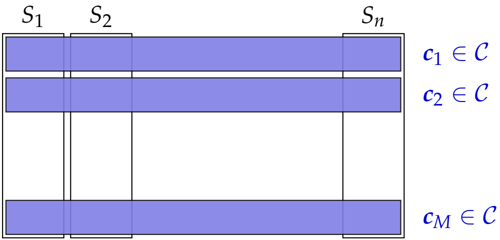
Storage systems use codes to cope with node failures.

- Before 2010: mostly replication or parity-check.
- 2010's: MDS storage (*e.g.* $[14, 10]$ Reed-Solomon code for Facebook).
- Recently: codes with locality (*e.g.* Hadoop Xorbas).

Storage systems use codes to cope with node failures.

- Before 2010: mostly replication or parity-check.
- 2010's: MDS storage (*e.g.* $[14, 10]$ Reed-Solomon code for Facebook).
- Recently: codes with locality (*e.g.* Hadoop Xorbas).

Given a code $\mathcal{C}$ of length n:



Definition (Reed-Solomon code). Let $\boldsymbol{x} = (x_1, \dots, x_n) \in \mathbb{F}_q^n$, pairwise distinct.

$$\mathrm{RS}_q(k,n) := \{(f(x_1), \dots, f(x_n)), f \in \mathbb{F}_q[X], \deg f < k\}$$

Definition (Reed-Solomon code). Let $\boldsymbol{x} = (x_1, \dots, x_n) \in \mathbb{F}_q^n$, pairwise distinct.

$$\mathrm{RS}_q(k,n) := \{(f(x_1), \dots, f(x_n)), f \in \mathbb{F}_q[X], \deg f < k\}$$

File storage:

$$\text{a file} \quad F_i \in \Sigma \simeq \mathbb{F}_{q^s}^k \quad \text{is encoded into} \quad \boldsymbol{c}_i \in \mathrm{RS}_q(k,n) \otimes \mathbb{F}_{q^s}$$

Example: Reed-Solomon storage systems

Definition (Reed-Solomon code). Let $\boldsymbol{x} = (x_1, \dots, x_n) \in \mathbb{F}_q^n$, pairwise distinct.

$$\mathrm{RS}_q(k, n) := \{(f(x_1), \dots, f(x_n)), f \in \mathbb{F}_q[X], \deg f < k\}$$

File storage:

$$\text{a file} \quad F_i \in \Sigma \simeq \mathbb{F}_{q^s}^k \quad \text{is encoded into} \quad \boldsymbol{c}_i \in \mathrm{RS}_q(k, n) \otimes \mathbb{F}_{q^s}$$

Main assumption (can be discussed):

$$s \gg M$$

Outline

Usual goal (assuming $s \gg M$): a large *PIR rate*

$$\rho := \frac{|F_i|}{|\boldsymbol{r}|} \, .$$

Usual goal (assuming $s \gg M$): a large *PIR rate*

$$\rho := \frac{|F_i|}{|\boldsymbol{r}|} .$$

Next, we present a PIR scheme for RS-coded databases.

- Originally [TR16], then extended and reformulated [TGKFH18, TGR18].
- Optimal PIR rate for $t = 1$ and $M \to \infty$.
- PIR rate conjectured optimal for $M \to \infty$.

[TR16] *PIR from MDS Coded Data in Distributed Storage Systems*. Tajeddine, El Rouayheb. ISIT. **2016**.

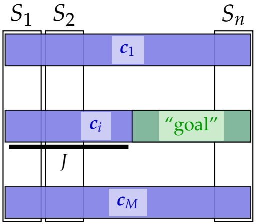
[TGKFH18] *Robust PIR from Coded Systems with Byzantine and Colluding Servers*. Tajeddine, Gnilke, Karpuk, Freij-Hollanti, Hollanti. ISIT. **2018**.

[TGR18] *PIR from MDS Coded Data in Distributed Storage Systems*. Tajeddine, Gnilke, El Rouayheb. IEEE-TIT. **2018**.

Notation:

$$\begin{aligned} \boldsymbol{a} \star \boldsymbol{b} &:= (a_1 b_1, \dots, a_n b_n) \\ \mathcal{C} \star \mathcal{C}' &:= \langle \{ \boldsymbol{c} \star \boldsymbol{c}' \mid \boldsymbol{c} \in \mathcal{C}, \boldsymbol{c}' \in \mathcal{C}' \} \rangle \end{aligned}$$

Notation:
$$\boldsymbol{a} \star \boldsymbol{b} := (a_1 b_1, \dots, a_n b_n)$$
$$\mathcal{C} \star \mathcal{C}' := \langle \{ \boldsymbol{c} \star \boldsymbol{c}' \mid \boldsymbol{c} \in \mathcal{C}, \boldsymbol{c}' \in \mathcal{C}' \} \rangle$$



System parameters:

$\mathcal{C} \subseteq \mathbb{F}_q^n$ the *storage code*, $\boldsymbol{C} \in \mathcal{C}^M$ the coded database

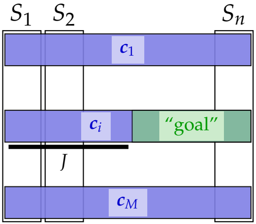
$J \subseteq [1, n]$ an information set for $\mathcal{C} \star \mathcal{D}$, and $\bar{J} := [1, n] \setminus J$

Notation:
$$\boldsymbol{a} \star \boldsymbol{b} := (a_1 b_1, \dots, a_n b_n)$$
$$\mathcal{C} \star \mathcal{C}' := \langle \{ \boldsymbol{c} \star \boldsymbol{c}' \mid \boldsymbol{c} \in \mathcal{C}, \boldsymbol{c}' \in \mathcal{C}' \} \rangle$$

System parameters:

$\mathcal{C} \subseteq \mathbb{F}_q^n$ the *storage code*, $\boldsymbol{C} \in \mathcal{C}^M$ the coded database

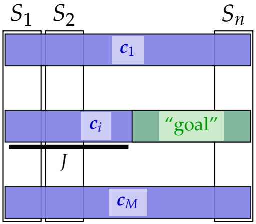
$J \subseteq [1, n]$ an information set for $\mathcal{C} \star \mathcal{D}$, and $\bar{J} := [1, n] \setminus J$



Queries: let $\mathcal{D} \subseteq \mathbb{F}_q^n$ be a *query code* of dual distance $d^\perp(\mathcal{D}) = t + 1$

The protocol: query generation

Notation: $$\boldsymbol{a} \star \boldsymbol{b} := (a_1 b_1, \dots, a_n b_n)$$
$$\mathcal{C} \star \mathcal{C}' := \langle \{ \boldsymbol{c} \star \boldsymbol{c}' \mid \boldsymbol{c} \in \mathcal{C}, \boldsymbol{c}' \in \mathcal{C}' \} \rangle$$



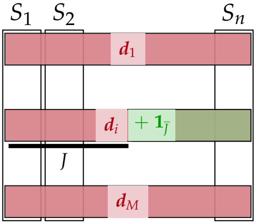
System parameters:

$\mathcal{C} \subseteq \mathbb{F}_q^n$ the *storage code*, $\boldsymbol{C} \in \mathcal{C}^M$ the coded database

$J \subseteq [1, n]$ an information set for $\mathcal{C} \star \mathcal{D}$, and $\bar{J} := [1, n] \setminus J$

Queries: let $\mathcal{D} \subseteq \mathbb{F}_q^n$ be a *query code* of dual distance $d^{\perp}(\mathcal{D}) = t + 1$

1. the user generates at random M words $\boldsymbol{d}_1, \dots, \boldsymbol{d}_M \in \mathcal{D}$ and defines $\boldsymbol{Q}$ as follows:
2. the j-th column of $\boldsymbol{Q}$ is sent to server S_j



Notation:
$$\boldsymbol{a} \star \boldsymbol{b} := (a_1 b_1, \dots, a_n b_n)$$
$$\mathcal{C} \star \mathcal{C}' := \langle \{ \boldsymbol{c} \star \boldsymbol{c}' \mid \boldsymbol{c} \in \mathcal{C}, \boldsymbol{c}' \in \mathcal{C}' \} \rangle$$

System parameters:

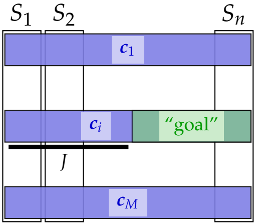
$\mathcal{C} \subseteq \mathbb{F}_q^n$ the *storage code*, $\boldsymbol{C} \in \mathcal{C}^M$ the coded database

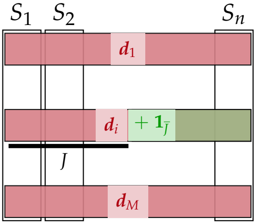
$J \subseteq [1, n]$ an information set for $\mathcal{C} \star \mathcal{D}$, and $\bar{J} := [1, n] \setminus J$

Queries: let $\mathcal{D} \subseteq \mathbb{F}_q^n$ be a *query code* of dual distance $d^\perp(\mathcal{D}) = t + 1$

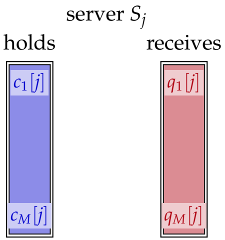
1. the user generates at random M words $\boldsymbol{d}_1, \dots, \boldsymbol{d}_M \in \mathcal{D}$ and defines $\boldsymbol{Q}$ as follows:
2. the j-th column of $\boldsymbol{Q}$ is sent to server S_j

Remark: queries remain private against collusions of servers of size $\leq t$.





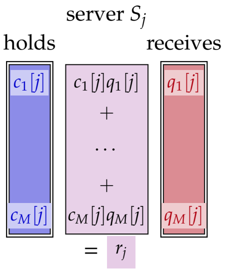
Server answers: server S_j receives as a query a column $\boldsymbol{Q}^{(j)} \in \mathbb{F}_q^M$ of $\boldsymbol{Q}$,



The protocol: server answers and reconstruction

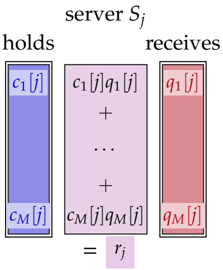
Server answers: server S_j receives as a query a column $\boldsymbol{Q}^{(j)} \in \mathbb{F}_q^M$ of $\boldsymbol{Q}$, and has to compute

$$r_j = \langle \boldsymbol{Q}^{(j)}, \boldsymbol{C}^{(j)} \rangle \in \mathbb{F}_q.$$

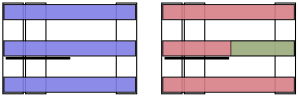


Server answers: server S_j receives as a query a column $\boldsymbol{Q}^{(j)} \in \mathbb{F}_q^M$ of $\boldsymbol{Q}$, and has to compute

$$r_j = \langle \boldsymbol{Q}^{(j)}, \boldsymbol{C}^{(j)} \rangle \in \mathbb{F}_q.$$

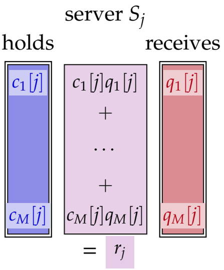


Reconstruction:



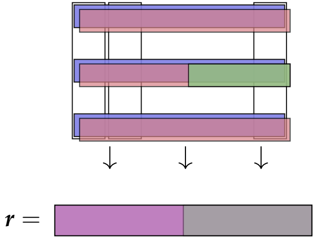
Server answers: server S_j receives as a query a column $\boldsymbol{Q}^{(j)} \in \mathbb{F}_q^M$ of $\boldsymbol{Q}$, and has to compute

$$r_j = \langle \boldsymbol{Q}^{(j)}, \boldsymbol{C}^{(j)} \rangle \in \mathbb{F}_q.$$



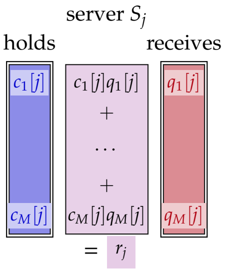
Reconstruction: The user collects

$$\boldsymbol{r} = (r_1, \dots, r_n) = \underbrace{\sum_{m=1}^{M} \boldsymbol{d}_m \star \boldsymbol{c}_m}_{\in \mathcal{C} \star \mathcal{D}} + \underbrace{\boldsymbol{1}_{\bar{J}} \star \boldsymbol{c}_i}_{= \boldsymbol{c}_i \text{ on } \bar{J}}$$



Server answers: server S_j receives as a query a column $\boldsymbol{Q}^{(j)} \in \mathbb{F}_q^M$ of $\boldsymbol{Q}$, and has to compute

$$r_j = \langle \boldsymbol{Q}^{(j)}, \boldsymbol{C}^{(j)} \rangle \in \mathbb{F}_q.$$

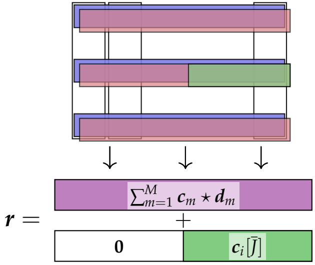


Reconstruction: The user collects

$$\boldsymbol{r} = (r_1, \dots, r_n) = \underbrace{\sum_{m=1}^{M} \boldsymbol{d}_m \star \boldsymbol{c}_m}_{\in \mathcal{C} \star \mathcal{D}} + \underbrace{\mathbf{1}_{\bar{J}} \star \boldsymbol{c}_i}_{= \boldsymbol{c}_i \text{ on } \bar{J}}$$

and interpolates on J to recover

- $\sum_{m=1}^{M} \boldsymbol{d}_m \star \boldsymbol{c}_m$,
- then $\boldsymbol{c}_i[|\bar{J}]$.



Features for 1 run of the protocol.

- download cost: n symbols over $\mathbb{F}_{q^s}$
- upload cost: an $(M \times n)$-matrix over $\mathbb{F}_q$ (negligible if $s \gg M$)
- retrieval of $|\bar{J}| = n - \dim(\mathcal{C} \star \mathcal{D})$ symbols of the desired file
- the protocol is **private** against collusions of size $\leq d^{\perp}(\mathcal{D}) - 1$

Features for 1 run of the protocol.

- download cost: n symbols over $\mathbb{F}_{q^s}$
- upload cost: an $(M \times n)$-matrix over $\mathbb{F}_q$ (negligible if $s \gg M$)
- retrieval of $|\bar{J}| = n - \dim(\mathcal{C} \star \mathcal{D})$ symbols of the desired file
- the protocol is **private** against collusions of size $\leq d^{\perp}(\mathcal{D}) - 1$

For **Reed-Solomon codes**: $\mathcal{C} = \mathrm{RS}_q(k, n)$ and $\mathcal{D} = \mathrm{RS}_q(t, n)$:

$$d^{\perp}(\mathcal{D}) - 1 = t \qquad \text{and} \qquad \mathcal{C} \star \mathcal{D} = \mathrm{RS}_q(k + t - 1, n) \Rightarrow |\bar{J}| = n - k - t + 1$$

Features for 1 run of the protocol.

- download cost: n symbols over $\mathbb{F}_{q^s}$
- upload cost: an $(M \times n)$-matrix over $\mathbb{F}_q$ (negligible if $s \gg M$)
- retrieval of $|\bar{J}| = n - \dim(\mathcal{C} \star \mathcal{D})$ symbols of the desired file
- the protocol is **private** against collusions of size $\leq d^{\perp}(\mathcal{D}) - 1$

For **Reed-Solomon codes**: $\mathcal{C} = \mathrm{RS}_q(k, n)$ and $\mathcal{D} = \mathrm{RS}_q(t, n)$:

$$d^{\perp}(\mathcal{D}) - 1 = t \qquad \text{and} \qquad \mathcal{C} \star \mathcal{D} = \mathrm{RS}_q(k + t - 1, n) \Rightarrow |\bar{J}| = n - k - t + 1$$

If $(n - k - t + 1) \mid k$, then **repeating** several runs gives a (download) **PIR rate**:

$$\rho = \frac{n - k - t + 1}{n} = 1 - \frac{k + t - 1}{n} \,.$$

Features for 1 run of the protocol.

- download cost: n symbols over $\mathbb{F}_{q^s}$
- upload cost: an $(M \times n)$-matrix over $\mathbb{F}_q$ (negligible if $s \gg M$)
- retrieval of $|\bar{J}| = n - \dim(\mathcal{C} \star \mathcal{D})$ symbols of the desired file
- the protocol is **private** against collusions of size $\leq d^{\perp}(\mathcal{D}) - 1$

For **Reed-Solomon codes**: $\mathcal{C} = \mathrm{RS}_q(k, n)$ and $\mathcal{D} = \mathrm{RS}_q(t, n)$:

$$d^{\perp}(\mathcal{D}) - 1 = t \qquad \text{and} \qquad \mathcal{C} \star \mathcal{D} = \mathrm{RS}_q(k + t - 1, n) \Rightarrow |\bar{J}| = n - k - t + 1$$

If $(n - k - t + 1) \mid k$, then **repeating** several runs gives a (download) **PIR rate**:

$$\rho = \frac{n - k - t + 1}{n} = 1 - \frac{k + t - 1}{n}.$$

Otherwise, **striping** methods allow to achieve the same PIR rate.

Outline

!!! Sorry for the notation !!!

!!! Sorry for the notation !!!

Definition: $\mathcal{C}$ is an $(n, k, d, \alpha, \beta, B)$-**regenerating code** if:

- $\mathcal{C}$ is a linear space of dimension B, consisting in $(\alpha \times n)$-matrices over $\mathbb{F}_q$,
- every $\boldsymbol{c} \in \mathcal{C}$ is fully determined by any k-subset of columns,
- every column of $\boldsymbol{c}$ can be "repaired", by downloading $\beta \leq \alpha$ symbols from any d-subset of columns (hence $d\beta \geq \alpha$).

!!! Sorry for the notation !!!

Definition: $\mathcal{C}$ is an $(n, k, d, \alpha, \beta, B)$-**regenerating code** if:

- $\mathcal{C}$ is a linear space of dimension B, consisting in $(\alpha \times n)$-matrices over $\mathbb{F}_q$,
- every $\boldsymbol{c} \in \mathcal{C}$ is fully determined by any k-subset of columns,
- every column of $\boldsymbol{c}$ can be "repaired", by downloading $\beta \leq \alpha$ symbols from any d-subset of columns (hence $d\beta \geq \alpha$).

Main bound (cut-set bound):

$$B \leq \sum_{i=0}^{k-1} \min(\alpha, (d-i)\beta)\,.$$

!!! Sorry for the notation !!!

Definition: $\mathcal{C}$ is an $(n, k, d, \alpha, \beta, B)$-**regenerating code** if:

- $\mathcal{C}$ is a linear space of dimension B, consisting in $(\alpha \times n)$-matrices over $\mathbb{F}_q$,
- every $\boldsymbol{c} \in \mathcal{C}$ is fully determined by any k-subset of columns,
- every column of $\boldsymbol{c}$ can be "repaired", by downloading $\beta \leq \alpha$ symbols from any d-subset of columns (hence $d\beta \geq \alpha$).

Main bound (cut-set bound):

$$B \leq \sum_{i=0}^{k-1} \min(\alpha, (d-i)\beta)\,.$$

A particular optimal point (minimum-bandwidth repair, MBR): $d\beta = \alpha$.
Then,

$$B = \left(kd - \frac{k(k-1)}{2}\right)\beta\,.$$

Product-matrix MBR codes [RSK11]

Optimal Exact-Regenerating Codes for Distributed Storage at the MSR and MBR Points via a Product-Matrix Construction. Rashmi, Shah, Kumar. IEEE-TIT. **2011**.

We set $\beta = 1$, hence $\alpha = d$.

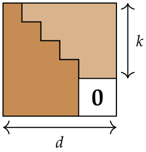
Optimal Exact-Regenerating Codes for Distributed Storage at the MSR and MBR Points via a Product-Matrix Construction. Rashmi, Shah, Kumar. IEEE-TIT. **2011**.

We set $\beta = 1$, hence $\alpha = d$.

1. Message symbols are arranged in a $(d \times d)$-matrix

$$A = \begin{pmatrix} \boldsymbol{S} & \boldsymbol{T}^\top \\ \boldsymbol{T} & \boldsymbol{0} \end{pmatrix}$$

   where $\boldsymbol{S}$ is $(k \times k)$-symmetric.



Product-matrix MBR codes [RSK11]

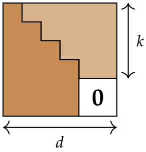
Optimal Exact-Regenerating Codes for Distributed Storage at the MSR and MBR Points via a Product-Matrix Construction. Rashmi, Shah, Kumar. IEEE-TIT. **2011**.

We set $\beta = 1$, hence $\alpha = d$.

1. Message symbols are arranged in a $(d \times d)$-matrix

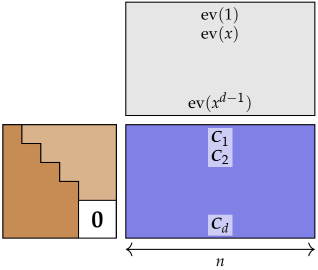
$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{S} & \boldsymbol{T}^\top \\ \boldsymbol{T} & \boldsymbol{0} \end{pmatrix}$$

   where $\boldsymbol{S}$ is $(k \times k)$-symmetric.



2. Let $\boldsymbol{G}$ be a $(d \times n)$ generator matrix for $\mathrm{RS}_q(d, n)$, echelonized in degree (*i.e.* a Vandermonde matrix). Codewords are then:

$$\boldsymbol{C} = \boldsymbol{A}\boldsymbol{G} \in \mathbb{F}_q^{d \times n}\,.$$



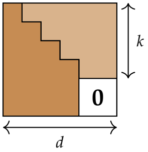
Optimal Exact-Regenerating Codes for Distributed Storage at the MSR and MBR Points via a Product-Matrix Construction. Rashmi, Shah, Kumar. IEEE-TIT. **2011**.

We set $\beta = 1$, hence $\alpha = d$.

1. Message symbols are arranged in a $(d \times d)$-matrix

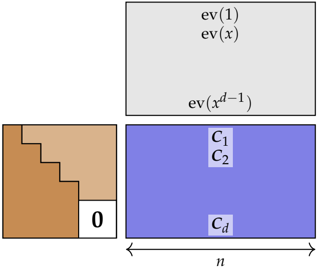
$$A = \begin{pmatrix} S & T^\top \\ T & 0 \end{pmatrix}$$

   where S is $(k \times k)$-symmetric.



2. Let G be a $(d \times n)$ generator matrix for $\mathrm{RS}_q(d, n)$, echelonized in degree (*i.e.* a Vandermonde matrix). Codewords are then:

$$C = AG \in \mathbb{F}_q^{d \times n} .$$

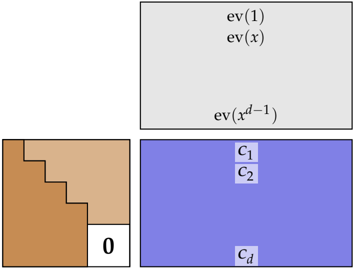


Remark: row C_j of C is a word of a RS code

- of dimension k, if $j > k$,
- of dimension $d > k$ otherwise.

Private Information Retrieval Schemes With Product-Matrix MBR Codes. **L.**, Tajeddine, Freij-Hollanti, Hollanti. IEEE IFS. **2021**.

PIR scheme with **no collusion** ($t = 1$).

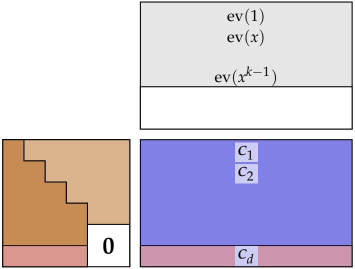


PIR scheme on PM-MBR codes with no collusion

Private Information Retrieval Schemes With Product-Matrix MBR Codes. **L.**, Tajeddine, Freij-Hollanti, Hollanti. IEEE IFS. **2021**.

PIR scheme with **no collusion** ($t = 1$).

- For row $j = d$ down to $k+1$:
  - Run a RS(k)-coded PIR scheme with randomness $\boldsymbol{D}$.
  - Interpolate random values $\sum \boldsymbol{d}_m \star \boldsymbol{C}_{j,m}$.
  - Recover row $\boldsymbol{C}_j$, then row $\boldsymbol{A}_j$.

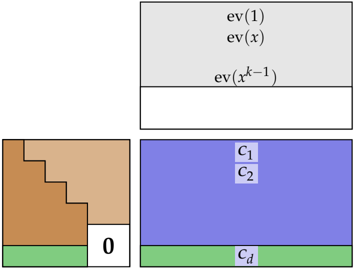


PIR scheme on PM-MBR codes with no collusion

Private Information Retrieval Schemes With Product-Matrix MBR Codes. **L.**, Tajeddine, Freij-Hollanti, Hollanti. IEEE IFS. **2021**.

PIR scheme with **no collusion** ($t = 1$).

- For row $j = d$ down to $k + 1$:
  - Run a RS(k)-coded PIR scheme with randomness $\boldsymbol{D}$.
  - Interpolate random values $\sum \boldsymbol{d}_m \star \boldsymbol{C}_{j,m}$.
  - Recover row $\boldsymbol{C}_j$, then row $\boldsymbol{A}_j$.



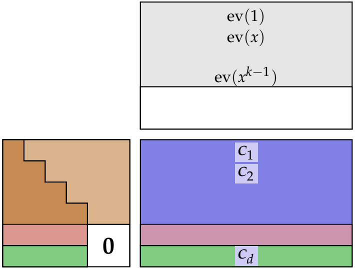
Retrieval rate: $1 - \frac{k}{n}$

PIR scheme on PM-MBR codes with no collusion

Private Information Retrieval Schemes With Product-Matrix MBR Codes. **L.**, Tajeddine, Freij-Hollanti, Hollanti. IEEE IFS. **2021**.

PIR scheme with **no collusion** ($t = 1$).

- For row $j = d$ down to $k + 1$:
  - Run a $\mathrm{RS}(k)$-coded PIR scheme with randomness $\boldsymbol{D}$.
  - Interpolate random values $\sum \boldsymbol{d}_m \star \boldsymbol{C}_{j,m}$.
  - Recover row $\boldsymbol{C}_j$, then row $\boldsymbol{A}_j$.

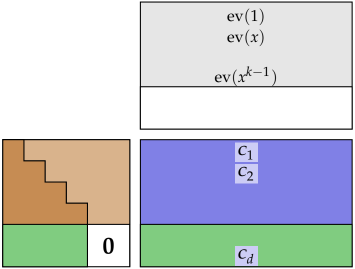


PIR scheme on PM-MBR codes with no collusion

Private Information Retrieval Schemes With Product-Matrix MBR Codes. **L.**, Tajeddine, Freij-Hollanti, Hollanti. IEEE IFS. **2021**.

PIR scheme with **no collusion** ($t = 1$).

- For row $j = d$ down to $k + 1$:
  - Run a $\mathrm{RS}(k)$-coded PIR scheme with randomness $\boldsymbol{D}$.
  - Interpolate random values $\sum \boldsymbol{d}_m \star \boldsymbol{C}_{j,m}$.
  - Recover row $\boldsymbol{C}_j$, then row $\boldsymbol{A}_j$.



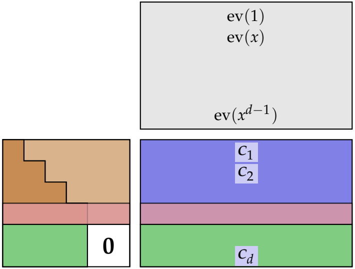
Retrieval rate: $1 - \frac{k}{n}$

PIR scheme on PM-MBR codes with no collusion

Private Information Retrieval Schemes With Product-Matrix MBR Codes. **L.**, Tajeddine, Freij-Hollanti, Hollanti. IEEE IFS. **2021**.

PIR scheme with **no collusion** ($t = 1$).

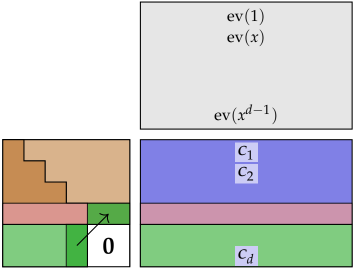
- For row $j = d$ down to $k + 1$:
  - Run a RS(k)-coded PIR scheme with randomness $\boldsymbol{D}$.
  - Interpolate random values $\sum \boldsymbol{d}_m \star \boldsymbol{C}_{j,m}$.
  - Recover row $\boldsymbol{C}_j$, then row $\boldsymbol{A}_j$.
- For row $j = k$ down to 1:
  - Run a RS(j)-coded PIR scheme with randomness $\boldsymbol{D}$.
  - Use symmetry of $\boldsymbol{A}$ and previously recovered data for the reconstruction (high-degree terms can be eliminated).
  - Interpolate random values $\sum \boldsymbol{d}_m \star \boldsymbol{C}_{j,m}$.
  - Recover row $\boldsymbol{C}_j$, then row $\boldsymbol{A}_j$.



Private Information Retrieval Schemes With Product-Matrix MBR Codes. **L.**, Tajeddine, Freij-Hollanti, Hollanti. IEEE IFS. **2021**.

PIR scheme with **no collusion** ($t = 1$).

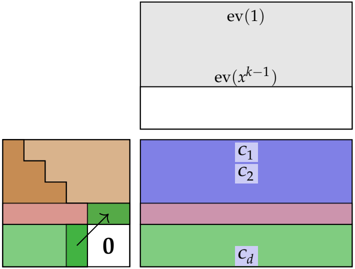
- For row $j = d$ down to $k + 1$:
  - Run a RS(k)-coded PIR scheme with randomness $\boldsymbol{D}$.
  - Interpolate random values $\sum \boldsymbol{d}_m \star \boldsymbol{C}_{j,m}$.
  - Recover row $\boldsymbol{C}_j$, then row $\boldsymbol{A}_j$.
- For row $j = k$ down to 1:
  - Run a RS(j)-coded PIR scheme with randomness $\boldsymbol{D}$.
  - Use symmetry of $\boldsymbol{A}$ and previously recovered data for the reconstruction (high-degree terms can be eliminated).
  - Interpolate random values $\sum \boldsymbol{d}_m \star \boldsymbol{C}_{j,m}$.
  - Recover row $\boldsymbol{C}_j$, then row $\boldsymbol{A}_j$.



Private Information Retrieval Schemes With Product-Matrix MBR Codes. **L.**, Tajeddine, Freij-Hollanti, Hollanti. IEEE IFS. **2021**.

PIR scheme with **no collusion** ($t = 1$).

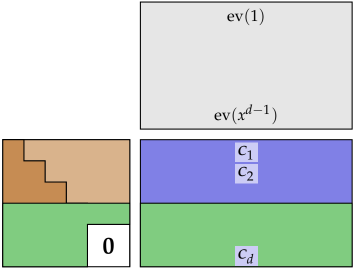
- For row $j = d$ down to $k + 1$:
  - Run a RS(k)-coded PIR scheme with randomness $\boldsymbol{D}$.
  - Interpolate random values $\sum \boldsymbol{d}_m \star \boldsymbol{C}_{j,m}$.
  - Recover row $\boldsymbol{C}_j$, then row $\boldsymbol{A}_j$.
- For row $j = k$ down to 1:
  - Run a RS(j)-coded PIR scheme with randomness $\boldsymbol{D}$.
  - Use symmetry of $\boldsymbol{A}$ and previously recovered data for the reconstruction (high-degree terms can be eliminated).
  - Interpolate random values $\sum \boldsymbol{d}_m \star \boldsymbol{C}_{j,m}$.
  - Recover row $\boldsymbol{C}_j$, then row $\boldsymbol{A}_j$.



Private Information Retrieval Schemes With Product-Matrix MBR Codes. **L.**, Tajeddine, Freij-Hollanti, Hollanti. IEEE IFS. **2021**.

PIR scheme with **no collusion** ($t = 1$).

- For row $j = d$ down to $k + 1$:
  - Run a RS(k)-coded PIR scheme with randomness $\boldsymbol{D}$.
  - Interpolate random values $\sum \boldsymbol{d}_m \star \boldsymbol{C}_{j,m}$.
  - Recover row $\boldsymbol{C}_j$, then row $\boldsymbol{A}_j$.
- For row $j = k$ down to 1:
  - Run a RS(j)-coded PIR scheme with randomness $\boldsymbol{D}$.
  - Use symmetry of $\boldsymbol{A}$ and previously recovered data for the reconstruction (high-degree terms can be eliminated).
  - Interpolate random values $\sum \boldsymbol{d}_m \star \boldsymbol{C}_{j,m}$.
  - Recover row $\boldsymbol{C}_j$, then row $\boldsymbol{A}_j$.

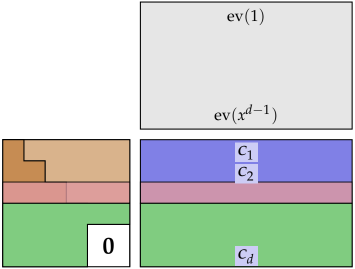


Retrieval rate: $1 - \frac{k}{n}$

Private Information Retrieval Schemes With Product-Matrix MBR Codes. **L.**, Tajeddine, Freij-Hollanti, Hollanti. IEEE IFS. **2021**.

PIR scheme with **no collusion** ($t = 1$).

- For row $j = d$ down to $k + 1$:
  - Run a RS(k)-coded PIR scheme with randomness $\boldsymbol{D}$.
  - Interpolate random values $\sum \boldsymbol{d}_m \star \boldsymbol{C}_{j,m}$.
  - Recover row $\boldsymbol{C}_j$, then row $\boldsymbol{A}_j$.
- For row $j = k$ down to 1:
  - Run a RS(j)-coded PIR scheme with randomness $\boldsymbol{D}$.
  - Use symmetry of $\boldsymbol{A}$ and previously recovered data for the reconstruction (high-degree terms can be eliminated).
  - Interpolate random values $\sum \boldsymbol{d}_m \star \boldsymbol{C}_{j,m}$.
  - Recover row $\boldsymbol{C}_j$, then row $\boldsymbol{A}_j$.

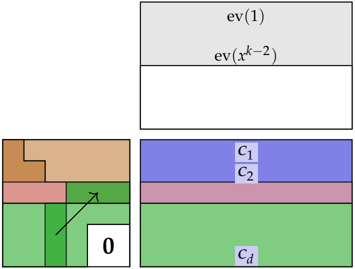


PIR scheme on PM-MBR codes with no collusion

Private Information Retrieval Schemes With Product-Matrix MBR Codes. **L.**, Tajeddine, Freij-Hollanti, Hollanti. IEEE IFS. **2021**.

PIR scheme with **no collusion** ($t = 1$).

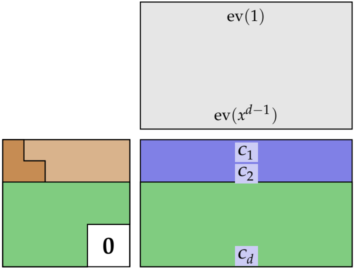
- For row $j = d$ down to $k+1$:
  - Run a RS(k)-coded PIR scheme with randomness $\boldsymbol{D}$.
  - Interpolate random values $\sum \boldsymbol{d}_m \star \boldsymbol{C}_{j,m}$.
  - Recover row $\boldsymbol{C}_j$, then row $\boldsymbol{A}_j$.
- For row $j = k$ down to 1:
  - Run a RS(j)-coded PIR scheme with randomness $\boldsymbol{D}$.
  - Use symmetry of $\boldsymbol{A}$ and previously recovered data for the reconstruction (high-degree terms can be eliminated).
  - Interpolate random values $\sum \boldsymbol{d}_m \star \boldsymbol{C}_{j,m}$.
  - Recover row $\boldsymbol{C}_j$, then row $\boldsymbol{A}_j$.



Private Information Retrieval Schemes With Product-Matrix MBR Codes. **L.**, Tajeddine, Freij-Hollanti, Hollanti. IEEE IFS. **2021**.

PIR scheme with **no collusion** ($t = 1$).

- For row $j = d$ down to $k + 1$:
  - Run a RS(k)-coded PIR scheme with randomness $\boldsymbol{D}$.
  - Interpolate random values $\sum \boldsymbol{d}_m \star \boldsymbol{C}_{j,m}$.
  - Recover row $\boldsymbol{C}_j$, then row $\boldsymbol{A}_j$.
- For row $j = k$ down to 1:
  - Run a RS(j)-coded PIR scheme with randomness $\boldsymbol{D}$.
  - Use symmetry of $\boldsymbol{A}$ and previously recovered data for the reconstruction (high-degree terms can be eliminated).
  - Interpolate random values $\sum \boldsymbol{d}_m \star \boldsymbol{C}_{j,m}$.
  - Recover row $\boldsymbol{C}_j$, then row $\boldsymbol{A}_j$.



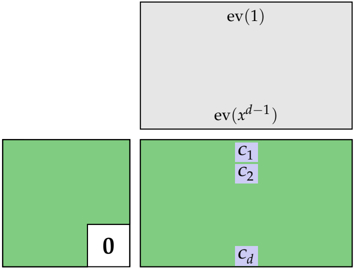
Retrieval rate: $1 - \frac{k-1}{n}$

PIR scheme on PM-MBR codes with no collusion

Private Information Retrieval Schemes With Product-Matrix MBR Codes. **L.**, Tajeddine, Freij-Hollanti, Hollanti. IEEE IFS. **2021**.

PIR scheme with **no collusion** ($t = 1$).

- For row $j = d$ down to $k + 1$:
  - Run a RS(k)-coded PIR scheme with randomness $\boldsymbol{D}$.
  - Interpolate random values $\sum \boldsymbol{d}_m \star \boldsymbol{C}_{j,m}$.
  - Recover row $\boldsymbol{C}_j$, then row $\boldsymbol{A}_j$.
- For row $j = k$ down to 1:
  - Run a RS(j)-coded PIR scheme with randomness $\boldsymbol{D}$.
  - Use symmetry of $\boldsymbol{A}$ and previously recovered data for the reconstruction (high-degree terms can be eliminated).
  - Interpolate random values $\sum \boldsymbol{d}_m \star \boldsymbol{C}_{j,m}$.
  - Recover row $\boldsymbol{C}_j$, then row $\boldsymbol{A}_j$.



Retrieval rate: $1 - \frac{j}{n}$

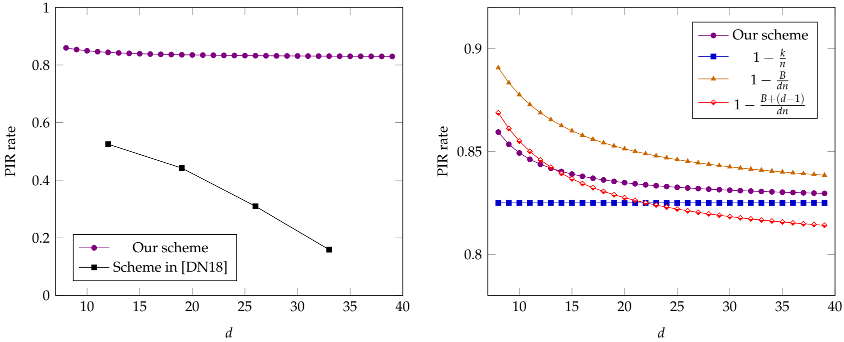
PIR scheme on PM-MBR codes

We get a PIR rate:

$$\rho = \frac{1-\frac{k}{n}}{1-\frac{k(k+1)(k-1)}{nB}} > 1-\frac{k}{n}$$

We get a PIR rate:

$$\rho = \frac{1-\frac{k}{n}}{1-\frac{k(k+1)(k-1)}{nB}} > 1-\frac{k}{n}$$



Comparison of PIR rates for $n = 40$ and $k = 7$.