

# On the construction of PIR schemes

Julien Lavauzelle

IRMAR, Université de Rennes 1

Séminaire de l'équipe Géométrie et Algèbre Effectives

01/03/2019

## 1. Private information retrieval

## 2. PIR schemes from codes with locality

- PIR schemes from locally decodable codes

- Transversal designs and codes

- A PIR scheme with transversal designs

## 3. PIR schemes for common storage systems

- Distributed storage systems

- A PIR scheme on RS-coded databases

- A PIR scheme with regenerating codes

## 4. Conclusion

## 1. Private information retrieval

## 2. PIR schemes from codes with locality

PIR schemes from locally decodable codes

Transversal designs and codes

A PIR scheme with transversal designs

## 3. PIR schemes for common storage systems

Distributed storage systems

A PIR scheme on RS-coded databases

A PIR scheme with regenerating codes

## 4. Conclusion

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  $i$ ?

**Trivial solution:** full download.

Introduced in:

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

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

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

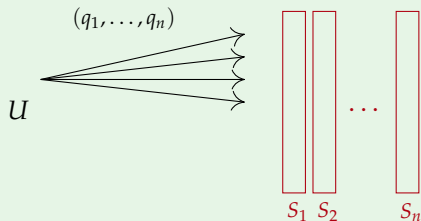
Introduced in:

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

Database  $F$  stored (in some way) on  $n$  servers  $S_1, \dots, S_n$ ,  
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  
 $\mathbf{q} = (q_1, \dots, q_n) \leftarrow \mathcal{Q}(i)$  and  
sends  $q_j$  to server  $S_j$



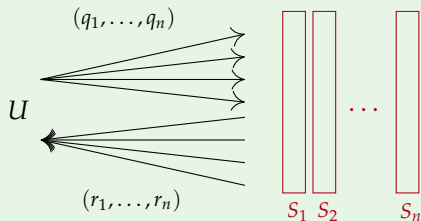
Introduced in:

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

Database  $F$  stored (in some way) on  $n$  servers  $S_1, \dots, S_n$ ,  
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  $\mathbf{q} = (q_1, \dots, q_n) \leftarrow \mathcal{Q}(i)$  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  $U$





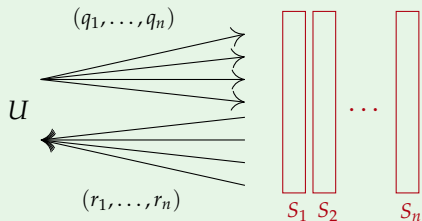
Introduced in:

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

Database  $F$  stored (in some way) on  $n$  servers  $S_1, \dots, S_n$ ,  
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  $\mathbf{q} = (q_1, \dots, q_n) \leftarrow \mathcal{Q}(i)$  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  $F_i = \mathcal{R}(\mathbf{q}, \mathbf{r}, i)$



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

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

A **collusion of servers**: set 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 privacy:**

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

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

A **collusion of servers**: set 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 privacy:**

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

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

**Theorem [CGKS95, CG97].** If  $t = n$  (in particular if  $n = 1$ ), 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)
- computation complexity (client and servers)
- global server storage overhead
- maximum size of collusions ( $t$ )

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 ( $t$ )

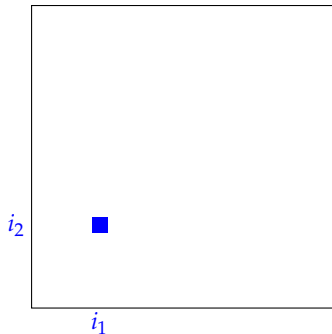
Several possible **settings**:

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

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

### Settings:

- ▶  $|F| = M$  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  $1 \leq i_1, i_2 \leq L$ .

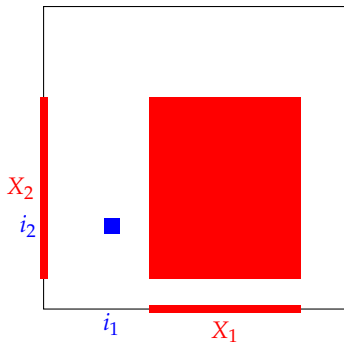




📄 *Private Information Retrieval*. Chor, Goldreich, Kushilevitz, Sudan. FOCS. 1995.

### Settings:

- ▶  $|F| = M$  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  $1 \leq i_1, i_2 \leq L$ .

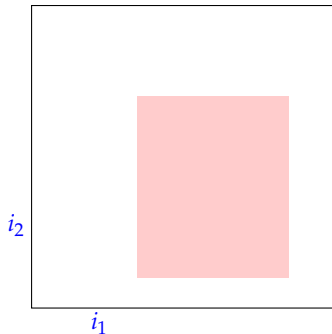


1.  $U$  generates at random two subsets  $X_1, X_2$  of  $[1, L]$ . Then  $U$  sends:

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

### Settings:

- ▶  $|F| = M$  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  $1 \leq i_1, i_2 \leq L$ .

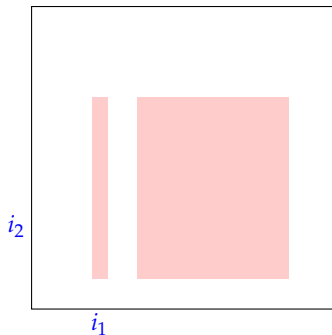


1.  $U$  generates at random two subsets  $X_1, X_2$  of  $[1, L]$ . Then  $U$  sends:
  - $(X_1, X_2)$  to  $S_{00}$ ,

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

### Settings:

- ▶  $|F| = M$  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  $1 \leq i_1, i_2 \leq L$ .

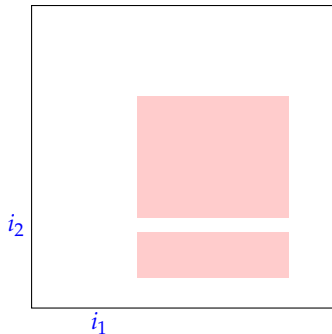


1.  $U$  generates at random two subsets  $X_1, X_2$  of  $[1, L]$ . Then  $U$  sends:
  - $(X_1, X_2)$  to  $S_{00}$ ,
  - $(X_1 \Delta \{i_1\}, X_2)$  to  $S_{10}$ ,

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

### Settings:

- ▶  $|F| = M$  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  $1 \leq i_1, i_2 \leq L$ .

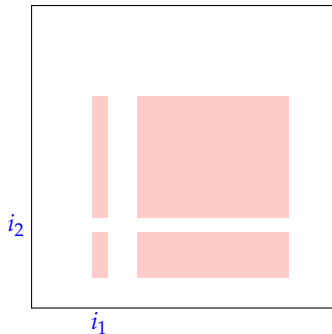


1.  $U$  generates at random two subsets  $X_1, X_2$  of  $[1, L]$ . Then  $U$  sends:
  - $(X_1, X_2)$  to  $S_{00}$ ,
  - $(X_1 \Delta \{i_1\}, X_2)$  to  $S_{10}$ ,
  - $(X_1, X_2 \Delta \{i_2\})$  to  $S_{01}$ ,

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

### Settings:

- ▶  $|F| = M$  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  $1 \leq i_1, i_2 \leq L$ .

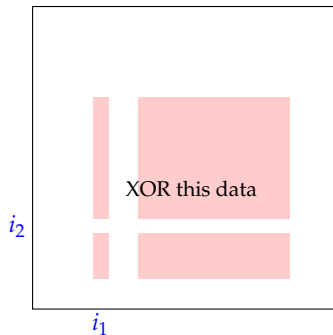


1.  $U$  generates at random two subsets  $X_1, X_2$  of  $[1, L]$ . Then  $U$  sends:
  - $(X_1, X_2)$  to  $S_{00}$ ,
  - $(X_1 \Delta \{i_1\}, X_2)$  to  $S_{10}$ ,
  - $(X_1, X_2 \Delta \{i_2\})$  to  $S_{01}$ ,
  - $(X_1 \Delta \{i_1\}, X_2 \Delta \{i_2\})$  to  $S_{11}$ .

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

### Settings:

- ▶  $|F| = M$  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  $1 \leq i_1, i_2 \leq L$ .

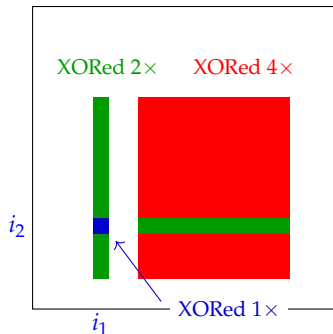


1.  $U$  generates at random two subsets  $X_1, X_2$  of  $[1, L]$ . Then  $U$  sends:
  - $(X_1, X_2)$  to  $S_{00}$ ,
  - $(X_1 \Delta \{i_1\}, X_2)$  to  $S_{10}$ ,
  - $(X_1, X_2 \Delta \{i_2\})$  to  $S_{01}$ ,
  - $(X_1 \Delta \{i_1\}, X_2 \Delta \{i_2\})$  to  $S_{11}$ .
2. At reception of  $(Z_1,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

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

### Settings:

- ▶  $|F| = M$  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.



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

With  $n = 4$  servers:

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

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

With  $n = 4$  servers:

- ▶ **Communication:**  $8\sqrt{M}$  uploaded bits, 4 downloaded bits,
- ▶ **Storage:** replication of  $F$  over 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.

## 1. Private information retrieval

## 2. PIR schemes from codes with locality

- PIR schemes from locally decodable codes

- Transversal designs and codes

- A PIR scheme with transversal designs

## 3. PIR schemes for common storage systems

- Distributed storage systems

- A PIR scheme on RS-coded databases

- A PIR scheme with regenerating codes

## 4. Conclusion

## 1. Private information retrieval

## 2. PIR schemes from codes with locality

- PIR schemes from locally decodable codes

- Transversal designs and codes

- A PIR scheme with transversal designs

## 3. PIR schemes for common storage systems

- Distributed storage systems

- A PIR scheme on RS-coded databases

- A PIR scheme with regenerating codes

## 4. Conclusion

📄 [KT00] *On the efficiency of local decoding procedures for error-correcting codes*. Katz, Trevisan. STOC. 2000.

**Definition (simplified).** A code  $\mathcal{C} \subseteq \mathbb{F}_q^n$  of dimension  $k$  is a **(smooth, complete, no-error)  $\ell$ -locally decodable code (LDC)** if there exists an algorithm  $\mathcal{D}$  such that for every  $\mathbf{m} \in \mathbb{F}_q^k$  and every  $i \in [1, k]$ , we have:

1.  $\mathcal{D}(i) = m_i$ ,
2.  $\mathcal{D}$  reads at most  $\ell$  entries of  $\mathbf{c} = \mathcal{C}(\mathbf{m})$ ,
3. every  $c_j, j \in [1, n]$ , is queried with equal probability.

📄 [KT00] *On the efficiency of local decoding procedures for error-correcting codes*. Katz, Trevisan. STOC. 2000.

**Definition (simplified).** A code  $\mathcal{C} \subseteq \mathbb{F}_q^n$  of dimension  $k$  is a **(smooth, complete, no-error)  $\ell$ -locally decodable code (LDC)** if there exists an algorithm  $\mathcal{D}$  such that for every  $\mathbf{m} \in \mathbb{F}_q^k$  and every  $i \in [1, k]$ , we have:

1.  $\mathcal{D}(i) = m_i$ ,
2.  $\mathcal{D}$  reads at most  $\ell$  entries of  $\mathbf{c} = \mathcal{C}(\mathbf{m})$ ,
3. every  $c_j, j \in [1, n]$ , is queried with equal probability.

**Example:** Hadamard code is a 2-LDC

$$\text{Had}(m) := \{ \mathbf{c} = (\phi(x))_{x \in \mathbb{F}_2^k}, \phi \in \mathcal{L}(\mathbb{F}_2^k, \mathbb{F}_2) \} \subseteq \mathbb{F}_2^{2^k}$$

Using a systematic encoding ( $\mathbf{m} \mapsto \phi = \sum_j m_j e_j^*$ ), for every  $x \in \mathbb{F}_2^k$  we get:

$$c_x = \phi(x) = \phi(x + \mathbf{y}) - \phi(\mathbf{y}) = \underbrace{c_{x+\mathbf{y}} - c_x}_{\text{uniformly distributed queries}}$$

**Context.**  $\ell$  non-colluding servers, each storing a copy of the database  $F = \mathbf{m}$ .

Let  $\mathcal{C}$  be an  $\ell$ -LDC, with decoder  $\mathcal{D}$ , and write  $\mathbf{c} = \mathcal{C}(\mathbf{m})$ .

**Context.**  $\ell$  non-colluding servers, each storing a copy of the database  $F = \mathbf{m}$ .

Let  $\mathcal{C}$  be an  $\ell$ -LDC, with decoder  $\mathcal{D}$ , and write  $\mathbf{c} = \mathcal{C}(\mathbf{m})$ .

We want to retrieve  $m_i = c_i$ .

1.  $U$  uses  $\mathcal{D}$  to generate a query vector  $\mathbf{q} = (q_1, \dots, q_\ell) \leftarrow \mathcal{Q}(i)$  and sends  $q_j$  to server  $S_j$ .
2. Each server  $S_j$  computes  $r_j = \mathcal{A}(q_j, F) := \mathcal{C}(\mathbf{m})_{q_j}$  and sends it back to  $U$ .
3.  $U$  uses the decoder  $\mathcal{D}(i)$  to recovers  $F_i = m_i$ .



**Context.**  $\ell$  non-colluding servers, each storing a copy of the database  $F = \mathbf{m}$ .

Let  $\mathcal{C}$  be an  $\ell$ -LDC, with decoder  $\mathcal{D}$ , and write  $\mathbf{c} = \mathcal{C}(\mathbf{m})$ .

We want to retrieve  $m_i = c_i$ .

1.  $U$  uses  $\mathcal{D}$  to generate a query vector  $\mathbf{q} = (q_1, \dots, q_\ell) \leftarrow \mathcal{Q}(i)$  and sends  $q_j$  to server  $S_j$ .
2. Each server  $S_j$  computes  $r_j = \mathcal{A}(q_j, F) := \mathcal{C}(\mathbf{m})_{q_j}$  and sends it back to  $U$ .
3.  $U$  uses the decoder  $\mathcal{D}(i)$  to recovers  $F_i = m_i$ .

### Features.

- ▶ Correctness of the decoder  $\Rightarrow$  **correctness** of the scheme.
- ▶ Uniform distribution of queries  $\Rightarrow$  **privacy** of the scheme.
- ▶ Communication:  $\ell \times (|F_i| + \log(n))$ .
- ▶ No collusion, heavy server computation.

- 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]

- 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]

**Remark.** The initial PIR scheme [CGKS95] can be embedded in this framework. The code is:

$$\mathcal{C} := \{c = (f(\mathbf{x}, \mathbf{y}))_{\mathbf{x}, \mathbf{y} \in (\mathbb{F}_2^L)^2}, f \text{ bilinear form over } \mathbb{F}_2^L\}.$$

This code is a 4-LDC:

$$\begin{aligned} F_{i_1, i_2} &= c_{(i_1, i_2)} = f(\mathbf{e}_{i_1}, \mathbf{e}_{i_2}) \\ &= f(\mathbf{x}, \mathbf{y}) - f(\mathbf{x} + \mathbf{e}_{i_1}, \mathbf{y}) - f(\mathbf{x}, \mathbf{y} + \mathbf{e}_{i_2}) + f(\mathbf{x} + \mathbf{e}_{i_1}, \mathbf{y} + \mathbf{e}_{i_2}). \end{aligned}$$

Generally, evaluating  $s$ -linear forms over  $\mathbb{F}_2^L$  give a  $2^s$ -LDC of length  $2^{Ls}$ .

The previous series of PIR schemes are:

- adaptive,
- communication-efficient, ...

The previous series of PIR schemes are:

- adaptive,
- communication-efficient, ...
- ... but need a lot of storage capacity
- and are very computationally inefficient (reencoding).

## 1. Private information retrieval

## 2. PIR schemes from codes with locality

PIR schemes from locally decodable codes

**Transversal designs and codes**

A PIR scheme with transversal designs

## 3. PIR schemes for common storage systems

Distributed storage systems

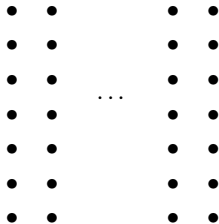
A PIR scheme on RS-coded databases

A PIR scheme with regenerating codes

## 4. Conclusion

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

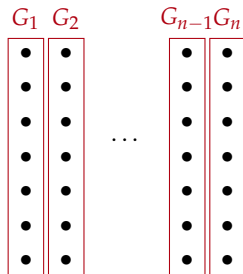
- ▶  $X$  a set of *points*,  $|X| = N = ns$ ,



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

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

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



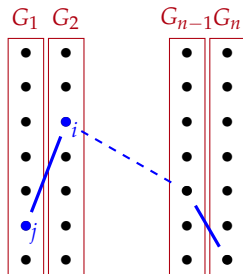


A transversal design  $\text{TD}(n, s) = (X, \mathcal{B}, \mathcal{G})$  is given by:

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

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

- ▶ 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  $B \in \mathcal{B}$



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

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

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

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

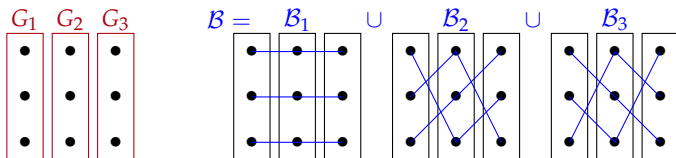
Its **incidence matrix**  $M$  has size  $|\mathcal{B}| \times |X|$  and is defined by:

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

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

- ▶  $\text{length}(\mathcal{C}) = |X|$ ,
- ▶  $\dim(\mathcal{C}) = \dim(\ker M)$ ,
- ▶ every  $B \in \mathcal{B}$  gives an  $\mathbf{h} \in \mathcal{C}^\perp$  such that  $\text{wt}(\mathbf{h}|_{\mathcal{G}_j}) = 1, \forall j = 1, \dots, n$ .

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



gives an incidence matrix

$$M = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Its rank over  $\mathbb{F}_3$  is 6  $\implies$  the associated code  $\mathcal{C}$  is a  $[9,3]_3$  code.

## 1. Private information retrieval

## 2. PIR schemes from codes with locality

PIR schemes from locally decodable codes

Transversal designs and codes

A PIR scheme with transversal designs

## 3. PIR schemes for common storage systems

Distributed storage systems

A PIR scheme on RS-coded databases

A PIR scheme with regenerating codes

## 4. Conclusion

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

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

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

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

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

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



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

**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** (somewhat optimal)
  - ▶  $\leq n$  additions over  $\mathbb{F}_q$  for the user
- ▶ storage overhead:  $(ns - M) \log q$  bits, where  $M = \dim(\mathcal{C})$

**Question:** transversal designs with good  $\dim(\mathcal{C})$  depending on  $(n, s)$ ?

$\mathcal{T}_A$ , the **classical affine transversal design**:

- ▶  $X = \mathbb{F}_q^m, m \geq 2,$
- ▶  $\mathcal{G}$  a set of  $q$  disjoint hyperplanes partitionning  $X,$
- ▶  $\mathcal{B} = \{\text{affine lines } L \text{ secant to each group of } \mathcal{G}\}.$

The code has:

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

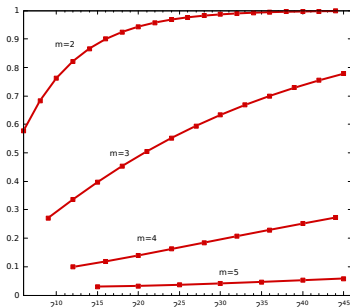
$\mathcal{T}_A$ , the **classical affine transversal design**:

- ▶  $X = \mathbb{F}_q^m, m \geq 2$ ,
- ▶  $\mathcal{G}$  a set of  $q$  disjoint hyperplanes partitioning  $X$ ,
- ▶  $\mathcal{B} = \{\text{affine lines } L \text{ secant to each group of } \mathcal{G}\}$ .

The code has:

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

rate  $M/N$



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



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

## 1. Private information retrieval

## 2. PIR schemes from codes with locality

PIR schemes from locally decodable codes

Transversal designs and codes

A PIR scheme with transversal designs

## 3. PIR schemes for common storage systems

Distributed storage systems

A PIR scheme on RS-coded databases

A PIR scheme with regenerating codes

## 4. Conclusion

## 1. Private information retrieval

## 2. PIR schemes from codes with locality

PIR schemes from locally decodable codes

Transversal designs and codes

A PIR scheme with transversal designs

## 3. PIR schemes for common storage systems

Distributed storage systems

A PIR scheme on RS-coded databases

A PIR scheme with regenerating codes

## 4. Conclusion

**Storage systems** use codes to cope with node failures.

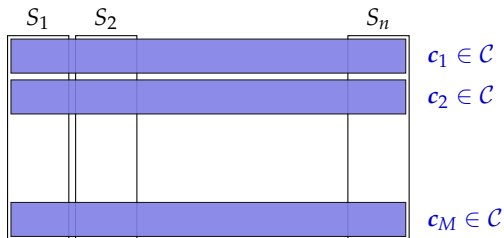
- ▶ Before 2010: mostly replication or parity-check.
- ▶ 2010's: MDS storage (*e.g.*  $[14, 10]$  Reed-Solomon code for Facebook).
- ▶ Recently: codes with locality (*e.g.* Hadoop Xorbas).



**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  $\mathbf{x} = (x_1, \dots, x_n) \in \mathbb{F}_q^n$ , pairwise distinct.

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

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

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

$\mathcal{C} = \text{RS}_q(k, n)$  is **MDS**:

- ▶ every codeword  $\mathbf{c} \in \mathcal{C}$  can be reconstructed from any  $k$ -subset of coordinates of  $\mathbf{c}$ ,
- ▶ any subset of  $d^\perp(\mathcal{C}) - 1 = k$  coordinates of  $\mathbf{c}$  are independent.

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

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

$\mathcal{C} = \text{RS}_q(k, n)$  is **MDS**:

- ▶ every codeword  $\mathbf{c} \in \mathcal{C}$  can be reconstructed from any  $k$ -subset of coordinates of  $\mathbf{c}$ ,
- ▶ any subset of  $d^\perp(\mathcal{C}) - 1 = k$  coordinates of  $\mathbf{c}$  are independent.

**File storage:**

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

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

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

$\mathcal{C} = \text{RS}_q(k, n)$  is **MDS**:

- ▶ every codeword  $\mathbf{c} \in \mathcal{C}$  can be reconstructed from any  $k$ -subset of coordinates of  $\mathbf{c}$ ,
- ▶ any subset of  $d^\perp(\mathcal{C}) - 1 = k$  coordinates of  $\mathbf{c}$  are independent.

**File storage:**

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

**Main assumption** (can be discussed):

$$s \gg M$$

## 1. Private information retrieval

## 2. PIR schemes from codes with locality

PIR schemes from locally decodable codes

Transversal designs and codes

A PIR scheme with transversal designs

## 3. PIR schemes for common storage systems

Distributed storage systems


A PIR scheme on RS-coded databases


A PIR scheme with regenerating codes


## 4. Conclusion

PIR schemes for RS-coded databases.

- ▶ Originally [TR16], then extended and reformulated [TGKFH18, TGR18].
- ▶ Scalable.
- ▶ Optimal download communication for  $t = 1$  and  $M \rightarrow \infty$ .
- ▶ Conjectured optimal for  $t > 1$  and  $M \rightarrow \infty$ .

 [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:**

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



# The protocol: query generation

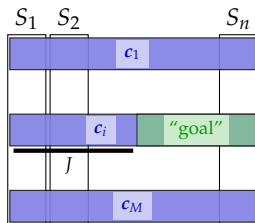
**Notation:**  $a \star b := (a_1 b_1, \dots, a_n b_n)$   
 $\mathcal{C} \star \mathcal{C}' := \langle \{c \star c' \mid c \in \mathcal{C}, c' \in \mathcal{C}'\} \rangle$

## System parameters:

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

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

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



# The protocol: query generation

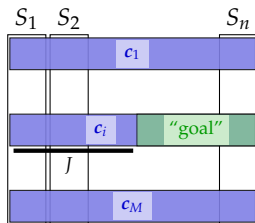
**Notation:**  $a \star b := (a_1 b_1, \dots, a_n b_n)$   
 $\mathcal{C} \star \mathcal{C}' := \langle \{c \star c' \mid c \in \mathcal{C}, c' \in \mathcal{C}'\} \rangle$

## System parameters:

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

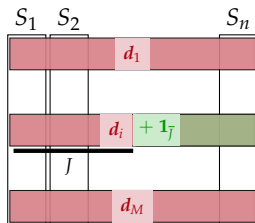
$\mathcal{D} \subseteq \mathbb{F}_q^n$  a query code of dual distance  $d^\perp(\mathcal{D}) = t + 1$

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



## Queries:

1. the user generates at random  $M$  words  $d_1, \dots, d_M \in \mathcal{D}$  and defines  $Q$  as follows:
2. the  $j$ -th column of  $Q$  is sent to server  $S_j$



# The protocol: query generation

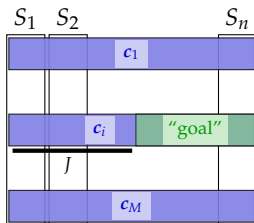
**Notation:**  $a \star b := (a_1 b_1, \dots, a_n b_n)$   
 $\mathcal{C} \star \mathcal{C}' := \langle \{c \star c' \mid c \in \mathcal{C}, c' \in \mathcal{C}'\} \rangle$

## System parameters:

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

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

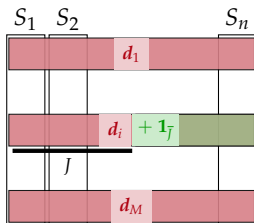
$J \subseteq [1, n]$  an information set for  $\mathcal{C} \star \mathcal{D}$ , and  $\bar{J} := [1, n] \setminus J$



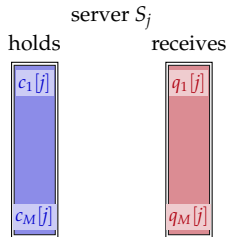
## Queries:

1. the user generates at random  $M$  words  $d_1, \dots, d_M \in \mathcal{D}$  and defines  $Q$  as follows:
2. the  $j$ -th column of  $Q$  is sent to server  $S_j$

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

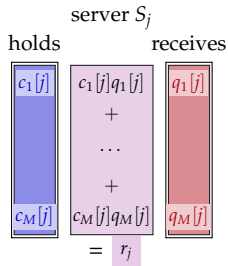


**Server answers:** server  $S_j$  receives as a query a column  $Q^{(j)} \in \mathbb{F}_q^M$  of  $Q$ ,



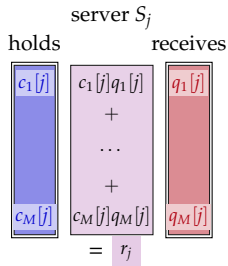
**Server answers:** server  $S_j$  receives as a query a column  $Q^{(j)} \in \mathbb{F}_q^M$  of  $Q$ , and has to compute

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

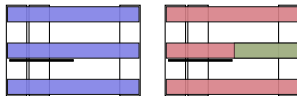


**Server answers:** server  $S_j$  receives as a query a column  $Q^{(j)} \in \mathbb{F}_q^M$  of  $Q$ , and has to compute

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

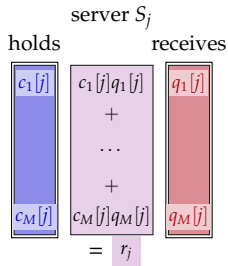


**Reconstruction:**



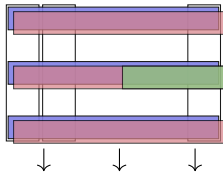
**Server answers:** server  $S_j$  receives as a query a column  $Q^{(j)} \in \mathbb{F}_q^M$  of  $Q$ , and has to compute

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



**Reconstruction:** The user collects

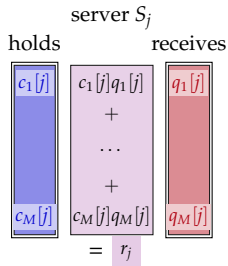
$$r = (r_1, \dots, r_n) = \underbrace{\sum_{m=1}^M d_m \star c_m}_{\in \mathcal{C} \star \mathcal{D}} + \underbrace{\mathbf{1}_{\bar{j}} \star c_i}_{= c_i \text{ on } \bar{j}}$$



$r =$

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

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

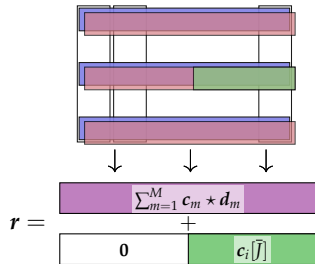


**Reconstruction:** The user collects

$$r = (r_1, \dots, r_n) = \underbrace{\sum_{m=1}^M d_m \star c_m}_{\in \mathcal{C} \star \mathcal{D}} + \underbrace{\mathbf{1}_{\bar{J}} \star c_i}_{= c_i \text{ on } \bar{J}}$$

and interpolates on  $J$  to recover

- $\sum_{m=1}^M d_m \star c_m$ ,
- then  $c_i[[\bar{J}]]$ .





**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**:  $\mathcal{C} = \text{RS}_q(k, n)$  and  $\mathcal{D} = \text{RS}_q(t, n)$ :

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

**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**:  $\mathcal{C} = \text{RS}_q(k, n)$  and  $\mathcal{D} = \text{RS}_q(t, n)$ :

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

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**:  $\mathcal{C} = \text{RS}_q(k, n)$  and  $\mathcal{D} = \text{RS}_q(t, n)$ :

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

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.

## 1. Private information retrieval

## 2. PIR schemes from codes with locality

PIR schemes from locally decodable codes

Transversal designs and codes

A PIR scheme with transversal designs

## 3. PIR schemes for common storage systems

Distributed storage systems

A PIR scheme on RS-coded databases

A PIR scheme with regenerating codes

## 4. Conclusion

**!!! 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  $c \in \mathcal{C}$  is fully determined by any  $k$ -subset of columns,
- ▶ every column of  $c$  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  $c \in \mathcal{C}$  is fully determined by any  $k$ -subset of columns,
- ▶ every column of  $c$  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 [WDR07]):

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



!!! 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  $c \in \mathcal{C}$  is fully determined by any  $k$ -subset of columns,
- ▶ every column of  $c$  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 [WDR07]):

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

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

Then,

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

 *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  $\alpha = d$ .

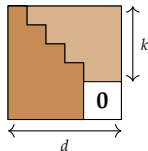
📄 *Optimal Exact-Regenerating Codes for Distributed Storage at the MSR and MBR Points via a Product-Matrix Construction.* Rashmi, Shah, Kumar. IEEE-TIT. 2011.


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

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

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

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



 *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  $\alpha = d$ .

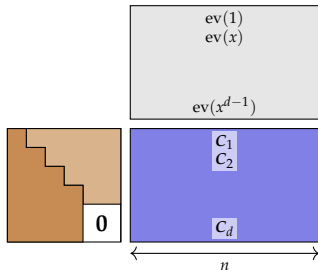
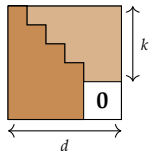
1. Message symbols are arranged in a  $(d \times d)$ -matrix


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

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

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

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



 *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

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

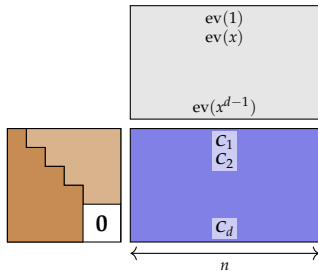
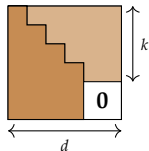
where  $\mathbf{S}$  is  $(k \times k)$ -symmetric.


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

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

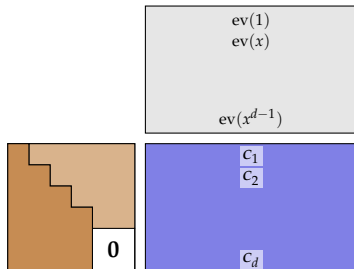
**Remark:** row  $\mathbf{C}_j$  of  $\mathbf{C}$  is a word of a RS code

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



 *Private Information Retrieval Schemes with Regenerating Codes*. L., Tajeddine, Freij-Hollanti, Hollanti. [arxiv:1811.02898](https://arxiv.org/abs/1811.02898). 2018.

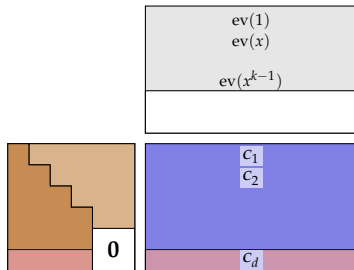
PIR scheme with **no collusion** ( $t = 1$ ).



📄 *Private Information Retrieval Schemes with Regenerating Codes*. L., Tajeddine, Freij-Hollanti, Hollanti. [arxiv:1811.02898](https://arxiv.org/abs/1811.02898). 2018.

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

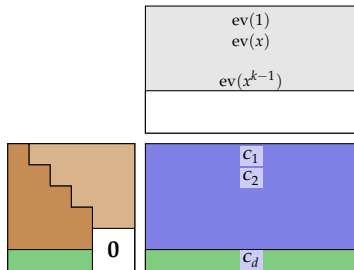
- For row  $j = d$  down to  $k + 1$ :
  - Run a  $RS(k)$ -coded PIR scheme with randomness  $D$ .
  - Interpolate random values  $\sum d_m \star C_{j,m}$ .
  - Recover row  $C_j$ , then row  $A_j$ .



📄 *Private Information Retrieval Schemes with Regenerating Codes*. L., Tajeddine, Freij-Hollanti, Hollanti. [arxiv:1811.02898](https://arxiv.org/abs/1811.02898). 2018.

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

- For row  $j = d$  down to  $k + 1$ :
  - Run a  $RS(k)$ -coded PIR scheme with randomness  $D$ .
  - Interpolate random values  $\sum d_m \star C_{j,m}$ .
  - Recover row  $C_j$ , then row  $A_j$ .



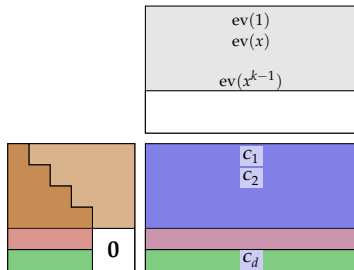
Retrieval rate:  $1 - \frac{k}{n}$



📄 *Private Information Retrieval Schemes with Regenerating Codes*. L., Tajeddine, Freij-Hollanti, Hollanti. [arxiv:1811.02898](https://arxiv.org/abs/1811.02898). 2018.

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

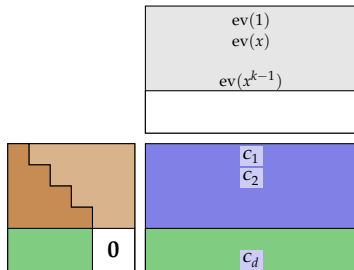
- For row  $j = d$  down to  $k + 1$ :
  - Run a  $RS(k)$ -coded PIR scheme with randomness  $D$ .
  - Interpolate random values  $\sum d_m \star C_{j,m}$ .
  - Recover row  $C_j$ , then row  $A_j$ .



📄 *Private Information Retrieval Schemes with Regenerating Codes*. L., Tajeddine, Freij-Hollanti, Hollanti. [arxiv:1811.02898](https://arxiv.org/abs/1811.02898). 2018.

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

- For row  $j = d$  down to  $k + 1$ :
  - Run a  $RS(k)$ -coded PIR scheme with randomness  $D$ .
  - Interpolate random values  $\sum d_m \star C_{j,m}$ .
  - Recover row  $C_j$ , then row  $A_j$ .

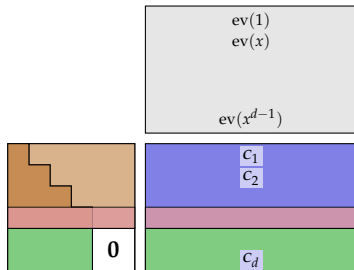


Retrieval rate:  $1 - \frac{k}{n}$

📄 *Private Information Retrieval Schemes with Regenerating Codes*. L., Tajeddine, Freij-Hollanti, Hollanti. [arxiv:1811.02898](https://arxiv.org/abs/1811.02898). 2018.

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

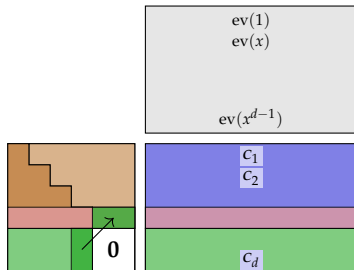
- For row  $j = d$  down to  $k + 1$ :
  - Run a  $RS(k)$ -coded PIR scheme with randomness  $D$ .
  - Interpolate random values  $\sum d_m \star C_{j,m}$ .
  - Recover row  $C_j$ , then row  $A_j$ .
- For row  $j = k$  down to 1:
  - Run a  $RS(j)$ -coded PIR scheme with randomness  $D$ .
  - Use symmetry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



📄 *Private Information Retrieval Schemes with Regenerating Codes*. L., Tajeddine, Freij-Hollanti, Hollanti. [arxiv:1811.02898](https://arxiv.org/abs/1811.02898). 2018.

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

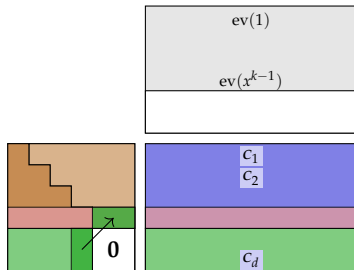
- For row  $j = d$  down to  $k + 1$ :
  - Run a  $RS(k)$ -coded PIR scheme with randomness  $D$ .
  - Interpolate random values  $\sum d_m \star C_{j,m}$ .
  - Recover row  $C_j$ , then row  $A_j$ .
- For row  $j = k$  down to 1:
  - Run a  $RS(j)$ -coded PIR scheme with randomness  $D$ .
  - Use symmetry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



📄 *Private Information Retrieval Schemes with Regenerating Codes*. L., Tajeddine, Freij-Hollanti, Hollanti. [arxiv:1811.02898](https://arxiv.org/abs/1811.02898). 2018.

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

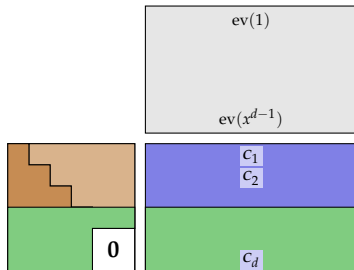
- For row  $j = d$  down to  $k + 1$ :
  - Run a  $RS(k)$ -coded PIR scheme with randomness  $D$ .
  - Interpolate random values  $\sum d_m \star C_{j,m}$ .
  - Recover row  $C_j$ , then row  $A_j$ .
- For row  $j = k$  down to 1:
  - Run a  $RS(j)$ -coded PIR scheme with randomness  $D$ .
  - Use symmetry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



📄 *Private Information Retrieval Schemes with Regenerating Codes*. L., Tajeddine, Freij-Hollanti, Hollanti. [arxiv:1811.02898](https://arxiv.org/abs/1811.02898). 2018.

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

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

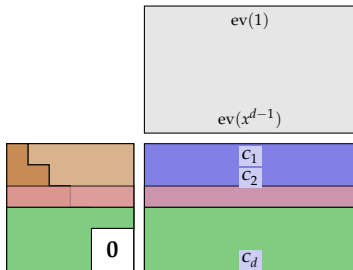


Retrieval rate:  $1 - \frac{k}{n}$

📄 *Private Information Retrieval Schemes with Regenerating Codes*. L., Tajeddine, Freij-Hollanti, Hollanti. [arxiv:1811.02898](https://arxiv.org/abs/1811.02898). 2018.

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

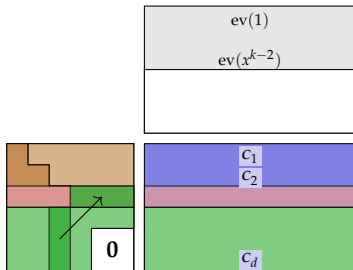
- For row  $j = d$  down to  $k + 1$ :
  - Run a  $RS(k)$ -coded PIR scheme with randomness  $D$ .
  - Interpolate random values  $\sum d_m \star C_{j,m}$ .
  - Recover row  $C_j$ , then row  $A_j$ .
- For row  $j = k$  down to 1:
  - Run a  $RS(j)$ -coded PIR scheme with randomness  $D$ .
  - Use symmetry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



📄 *Private Information Retrieval Schemes with Regenerating Codes*. L., Tajeddine, Freij-Hollanti, Hollanti. [arxiv:1811.02898](https://arxiv.org/abs/1811.02898). 2018.

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

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

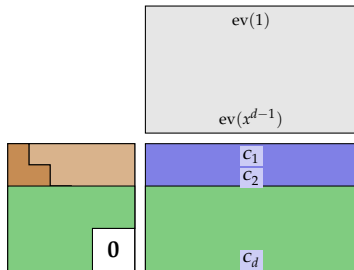




📄 *Private Information Retrieval Schemes with Regenerating Codes*. L., Tajeddine, Freij-Hollanti, Hollanti. [arxiv:1811.02898](https://arxiv.org/abs/1811.02898). 2018.

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

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

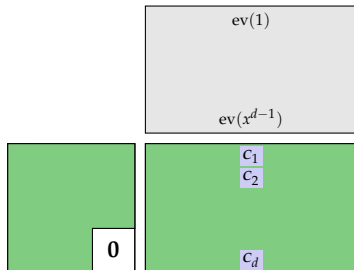


Retrieval rate:  $1 - \frac{k-1}{n}$

📄 *Private Information Retrieval Schemes with Regenerating Codes*. L., Tajeddine, Freij-Hollanti, Hollanti. [arxiv:1811.02898](https://arxiv.org/abs/1811.02898). 2018.

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

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



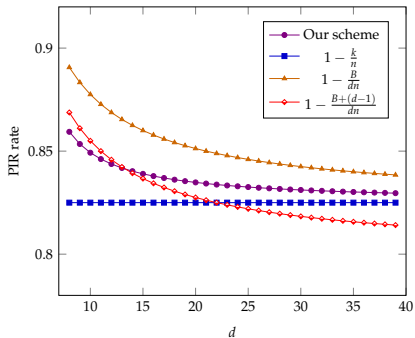
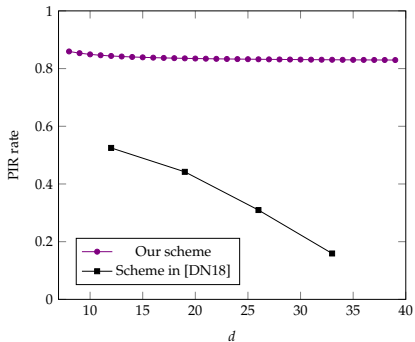
Retrieval rate:  $1 - \frac{j}{n}$

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  $k = 7$ .

## 1. Private information retrieval

## 2. PIR schemes from codes with locality

PIR schemes from locally decodable codes

Transversal designs and codes

A PIR scheme with transversal designs

## 3. PIR schemes for common storage systems

Distributed storage systems

A PIR scheme on RS-coded databases

A PIR scheme with regenerating codes

## 4. Conclusion

## **Private information retrieval:**

- ▶ concentrated a lot of recent research,
- ▶ involves nice mathematical tools,
- ▶ but in practice ... relies on questionable assumptions (collusions, size of entries, communication channels)

## Private information retrieval:

- ▶ concentrated a lot of recent research,
- ▶ involves nice mathematical tools,
- ▶ but in practice ... relies on questionable assumptions (collusions, size of entries, communication channels)

Questions?