Construction of computationally efficient PIR protocols

Julien Lavauzelle

LIX & INRIA Saclay, Université Paris-Saclay

Technical University of Munich, Germany
27/02/2018

1. The PIR issue
2. Transversal designs for efficient PIR protocols
3. Instances

Informal PIR

Given a file F,

can we retrieve the entry F_i

without leaking any information on i?

Informal PIR

Given a file F,

can we retrieve the entry F_i

without leaking any information on i?

Remarks:

- PIR $\neq$ anonymity (where the user is hidden)
- PIR $\neq$ encryption (where data is hidden)

Formal definition of PIR

A file F stored on ℓ servers $S_1, \ldots, S_\ell$. File F may be encoded.

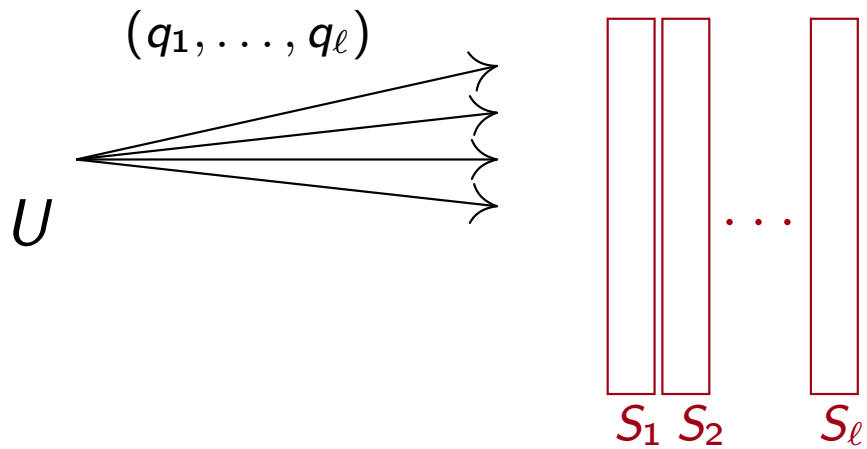
Private Information Retrieval (PIR) protocol:
user U wants to recover F_i privately.

Formal definition of PIR

A file F stored on ℓ servers $S_1, \ldots, S_\ell$. File F may be encoded.

Private Information Retrieval (PIR) protocol:
user U wants to recover F_i privately.

1. U generates a query vector $\mathbf{q} = (q_1, \ldots, q_\ell) \leftarrow \mathcal{Q}(i)$ and sends q_j to S_j

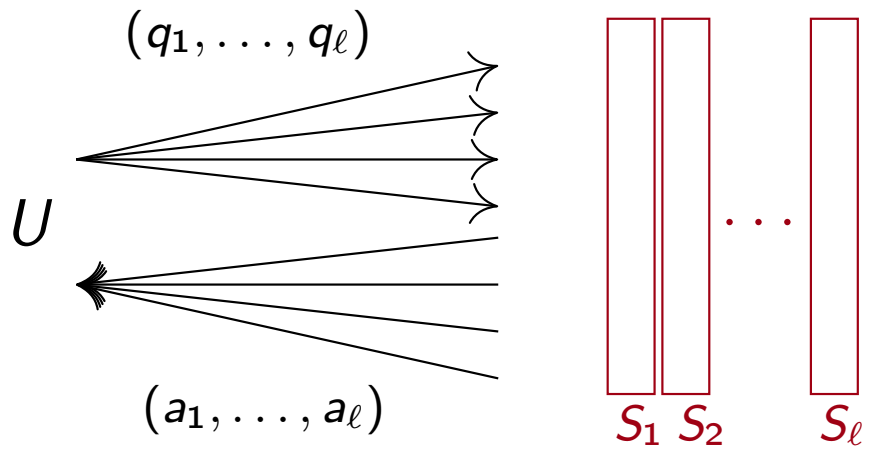


Formal definition of PIR

A file F stored on ℓ servers $S_1, \ldots, S_\ell$. File F may be encoded.

Private Information Retrieval (PIR) protocol:
user U wants to recover F_i privately.

1. U generates a query vector $\mathbf{q} = (q_1, \ldots, q_\ell) \leftarrow \mathcal{Q}(i)$ and sends q_j to S_j
2. Each server S_j computes $a_j = \mathcal{A}_j(q_j, F_{|S_j})$ and sends it back to U

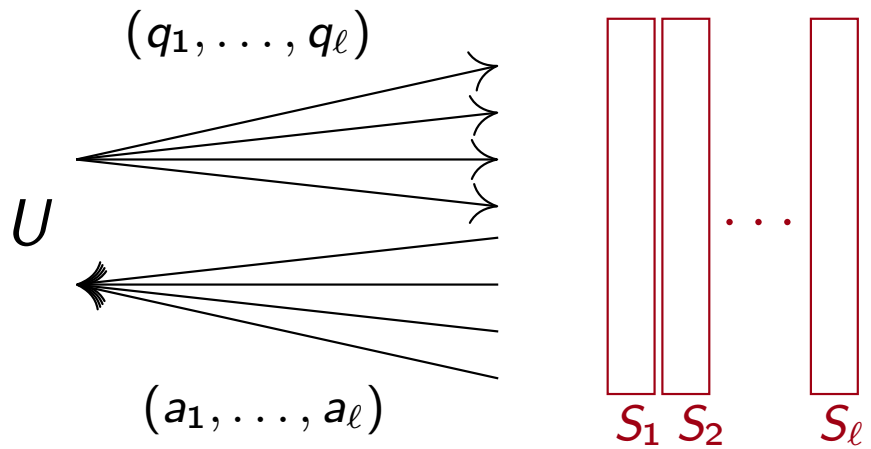


Formal definition of PIR

A file F stored on ℓ servers $S_1, \ldots, S_\ell$. File F may be encoded.

Private Information Retrieval (PIR) protocol:
user U wants to recover F_i privately.

1. U generates a query vector $\mathbf{q} = (q_1, \ldots, q_\ell) \leftarrow \mathcal{Q}(i)$ and sends q_j to S_j
2. Each server S_j computes $a_j = \mathcal{A}_j(q_j, F_{|S_j})$ and sends it back to U
3. U recovers $F_i = \mathcal{R}(\mathbf{q}, \mathbf{a}, i)$

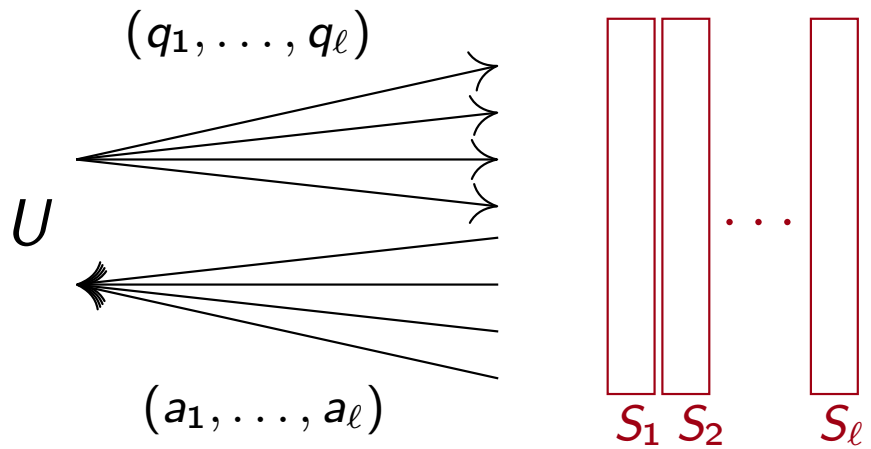


Formal definition of PIR

A file F stored on ℓ servers $S_1, \ldots, S_\ell$. File F may be encoded.

Private Information Retrieval (PIR) protocol:
user U wants to recover F_i privately.

1. U generates a query vector $\mathbf{q} = (q_1, \ldots, q_\ell) \leftarrow \mathcal{Q}(i)$ and sends q_j to S_j
2. Each server S_j computes $a_j = \mathcal{A}_j(q_j, F_{|S_j})$ and sends it back to U
3. U recovers $F_i = \mathcal{R}(\mathbf{q}, \mathbf{a}, i)$



Information-theoretic privacy: $\mathrm{I}(i\,; q_j) = 0,\ \forall j = 1, \ldots, \ell.$

Goals

Common goals for PIR:

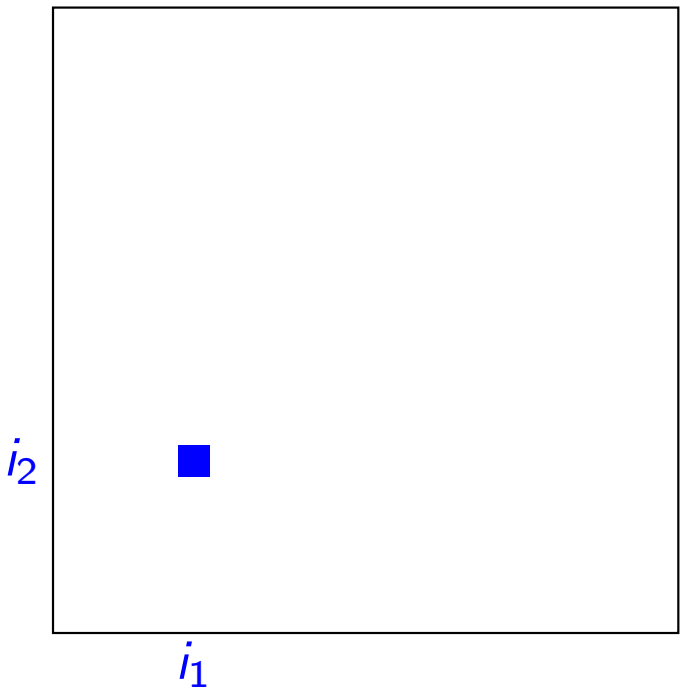
- Low communication complexity (number of bits exchanged between user and servers).
  $\rightarrow$ number of servers ≥ 2.
- Low storage overhead for the servers (if coded file).
- Low computation complexity for algorithms $\mathcal{A}$ (server) and $\mathcal{R}$ (user).

Remark: we mainly focus on privacy without collusion of servers.

Seminal work [CGKS'95-98]

Ref: Chor, Goldreich, Kushilevitz, Sudan, *Private Information Retrieval*, FOCS'95, J.ACM'98

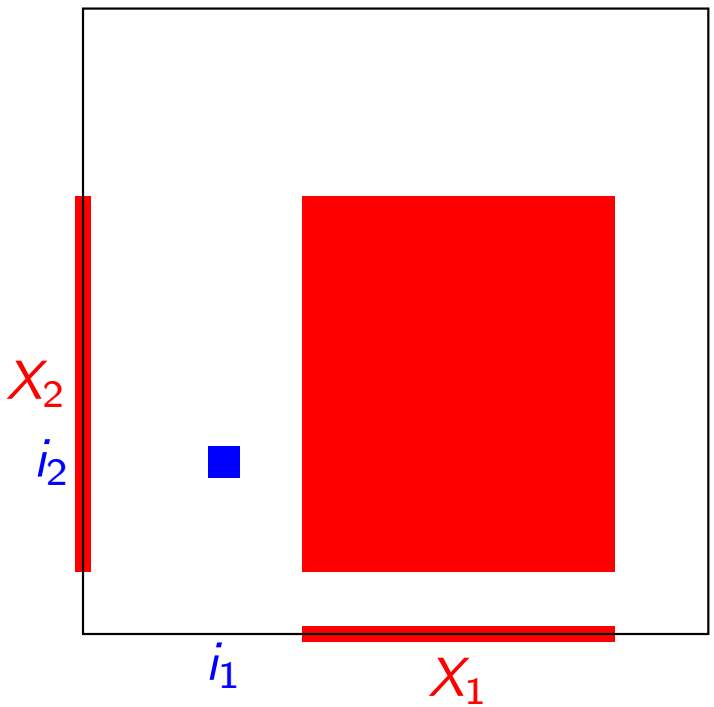
- $|F| = n$ bits, with $n = m^2$, and let's see $[1, n]$ as $[1, m]^2$.
- 4 servers $S_{00}, S_{01}, S_{10}, S_{11}$. Each server holds F.
- Assume user U wants to retrieve $F_{(i_1,i_2)}$, $1 \leq i_1, i_2 \leq m$.



Seminal work [CGKS'95-98]

Ref: Chor, Goldreich, Kushilevitz, Sudan, *Private Information Retrieval*, FOCS'95, J.ACM'98

- $|F| = n$ bits, with $n = m^2$, and let's see $[1, n]$ as $[1, m]^2$.
- 4 servers $S_{00}, S_{01}, S_{10}, S_{11}$. Each server holds F.
- Assume user U wants to retrieve $F_{(i_1,i_2)}$, $1 \leq i_1, i_2 \leq m$.

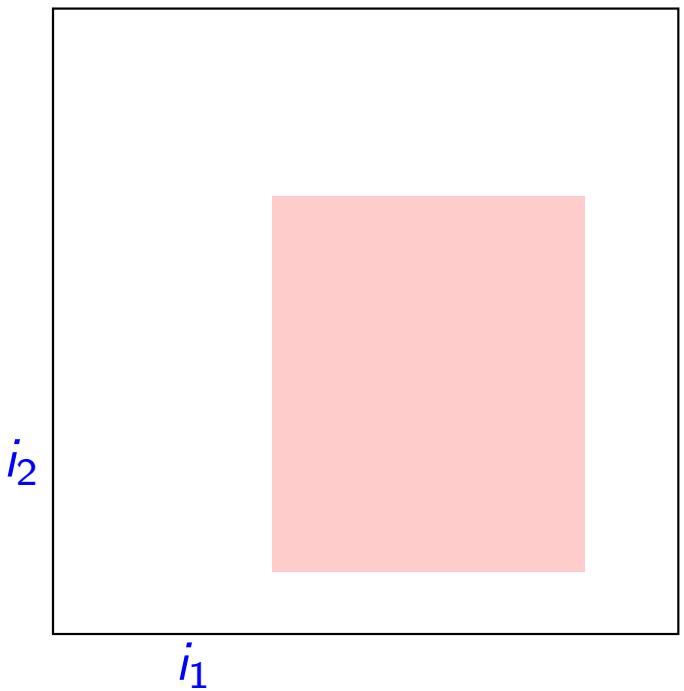


1. U generates at random two subsets X_1, X_2 of $[1, m]$. Then U sends:

Seminal work [CGKS'95-98]

Ref: Chor, Goldreich, Kushilevitz, Sudan, *Private Information Retrieval*, FOCS'95, J.ACM'98

- $|F| = n$ bits, with $n = m^2$, and let's see $[1, n]$ as $[1, m]^2$.
- 4 servers $S_{00}, S_{01}, S_{10}, S_{11}$. Each server holds F.
- Assume user U wants to retrieve $F_{(i_1,i_2)}$, $1 \leq i_1, i_2 \leq m$.

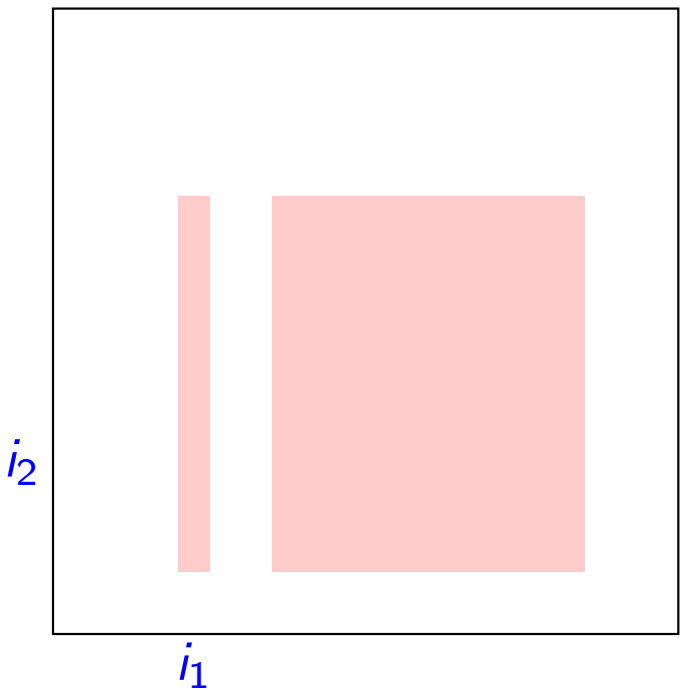


1. U generates at random two subsets X_1, X_2 of $[1, m]$. Then U sends:
   - (X_1 , X_2) to S_{00},

Seminal work [CGKS'95-98]

Ref: Chor, Goldreich, Kushilevitz, Sudan, *Private Information Retrieval*, FOCS'95, J.ACM'98

- $|F| = n$ bits, with $n = m^2$, and let's see $[1, n]$ as $[1, m]^2$.
- 4 servers $S_{00}, S_{01}, S_{10}, S_{11}$. Each server holds F.
- Assume user U wants to retrieve $F_{(i_1,i_2)}$, $1 \leq i_1, i_2 \leq m$.

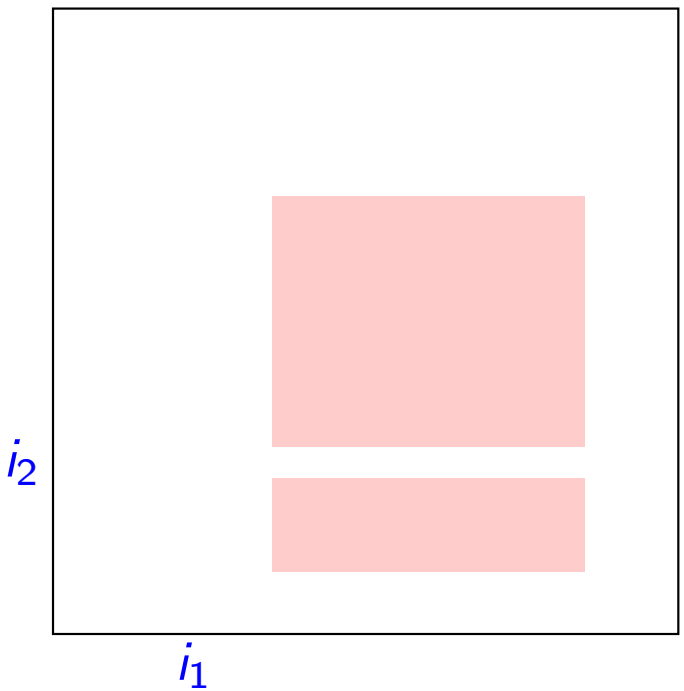


1. U generates at random two subsets X_1, X_2 of $[1, m]$. Then U sends:
   - $(\quad X_1 \quad , \quad X_2 \quad)$ to S_{00},
   - $(X_1 \Delta \{i_1\}, \quad X_2 \quad)$ to S_{10},

Seminal work [CGKS'95-98]

Ref: Chor, Goldreich, Kushilevitz, Sudan, *Private Information Retrieval*, FOCS'95, J.ACM'98

- $|F| = n$ bits, with $n = m^2$, and let's see $[1, n]$ as $[1, m]^2$.
- 4 servers $S_{00}, S_{01}, S_{10}, S_{11}$. Each server holds F.
- Assume user U wants to retrieve $F_{(i_1,i_2)}$, $1 \leq i_1, i_2 \leq m$.

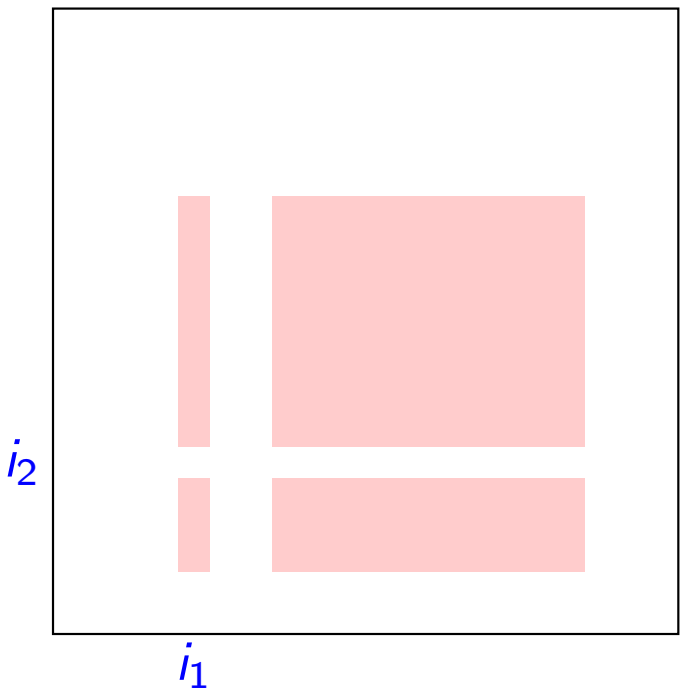


1. U generates at random two subsets X_1, X_2 of $[1, m]$. Then U sends:
   - $(\quad X_1 \quad , \quad X_2 \quad)$ to S_{00},
   - $(X_1 \Delta \{i_1\}, \quad X_2 \quad)$ to S_{10},
   - $(\quad X_1 \quad , X_2 \Delta \{i_2\})$ to S_{01},

Seminal work [CGKS'95-98]

Ref: Chor, Goldreich, Kushilevitz, Sudan, *Private Information Retrieval*, FOCS'95, J.ACM'98

- $|F| = n$ bits, with $n = m^2$, and let's see $[1, n]$ as $[1, m]^2$.
- 4 servers $S_{00}, S_{01}, S_{10}, S_{11}$. Each server holds F.
- Assume user U wants to retrieve $F_{(i_1,i_2)}$, $1 \leq i_1, i_2 \leq m$.

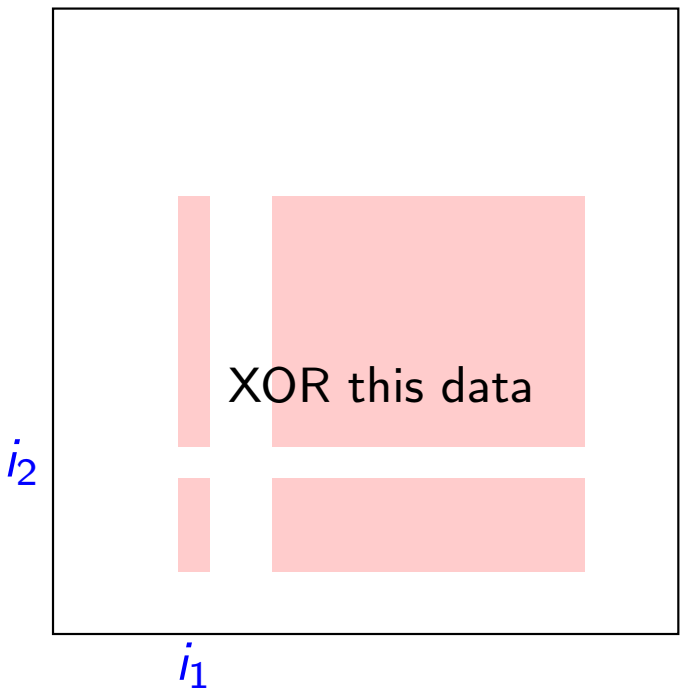


1. U generates at random two subsets X_1, X_2 of $[1, m]$. Then U sends:
   - $(\quad X_1 \quad , \quad X_2 \quad)$ to S_{00},
   - $(X_1 \Delta \{i_1\}, \quad X_2 \quad)$ to S_{10},
   - $(\quad X_1 \quad , X_2 \Delta \{i_2\})$ to S_{01},
   - $(X_1 \Delta \{i_1\}, X_2 \Delta \{i_2\})$ to S_{11}.

Seminal work [CGKS'95-98]

Ref: Chor, Goldreich, Kushilevitz, Sudan, *Private Information Retrieval*, FOCS'95, J.ACM'98

- $|F| = n$ bits, with $n = m^2$, and let's see $[1, n]$ as $[1, m]^2$.
- 4 servers $S_{00}, S_{01}, S_{10}, S_{11}$. Each server holds F.
- Assume user U wants to retrieve $F_{(i_1,i_2)}$, $1 \leq i_1, i_2 \leq m$.

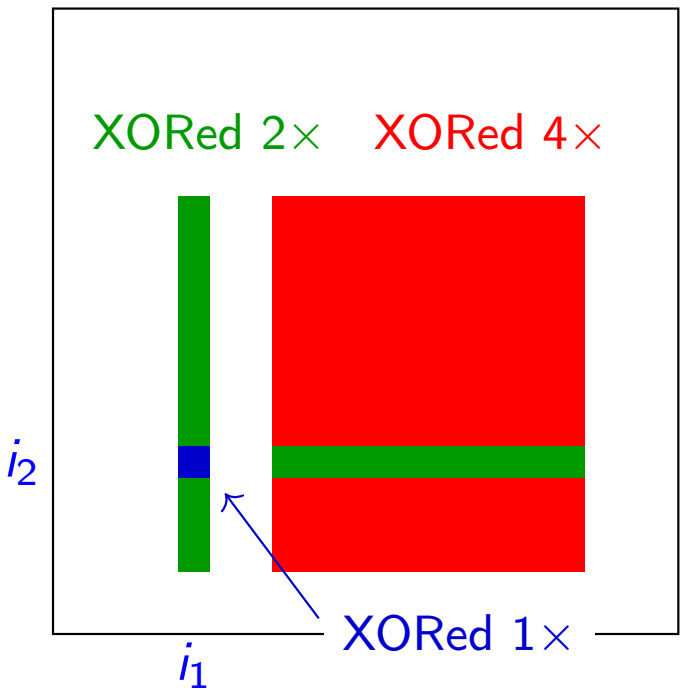


1. U generates at random two subsets X_1, X_2 of $[1, m]$. Then U sends:
   - $(\quad X_1 \quad , \quad X_2 \quad)$ to S_{00},
   - $(X_1 \Delta \{i_1\}, \quad X_2 \quad)$ to S_{10},
   - $(\quad X_1 \quad , X_2 \Delta \{i_2\})$ to S_{01},
   - $(X_1 \Delta \{i_1\}, X_2 \Delta \{i_2\})$ to S_{11}.
2. 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]

Ref: Chor, Goldreich, Kushilevitz, Sudan, *Private Information Retrieval*, FOCS'95, J.ACM'98

- $|F| = n$ bits, with $n = m^2$, and let's see $[1, n]$ as $[1, m]^2$.
- 4 servers $S_{00}, S_{01}, S_{10}, S_{11}$. Each server holds F.
- Assume user U wants to retrieve $F_{(i_1,i_2)}$, $1 \leq i_1, i_2 \leq m$.



1. U generates at random two subsets X_1, X_2 of $[1, m]$. Then U sends:
   - $(\quad X_1 \quad , \quad X_2 \quad)$ to S_{00},
   - $(X_1 \Delta \{i_1\}, \quad X_2 \quad)$ to S_{10},
   - $(\quad X_1 \quad , X_2 \Delta \{i_2\})$ 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 received bits and outputs the result.

Parameters of [CGKS'95-98]

Secure and correct.

Parameters of [CGKS'95-98]

Secure and correct.

With 4 servers:

- Communication: $8\sqrt{n}$ uploaded bits, 4 dowloaded bits,
- Storage: replication of F over 4 servers,
- Complexity: in average, XOR of $n/4$ bits for each server's answer; XOR of 4 bits for the user.

Parameters of [CGKS'95-98]

Secure and correct.

With 4 servers:

- Communication: $8\sqrt{n}$ uploaded bits, 4 dowloaded bits,
- Storage: replication of F over 4 servers,
- Complexity: in average, XOR of $n/4$ bits for each server's answer; XOR of 4 bits for the user.

Generalizable to 2^s servers:

- Communication: $s2^s n^{1/s}$ uploaded bits, 2^s dowloaded bits,
- Storage: replication of F over 2^s servers,
- Complexity: in average, XOR of $n/2^s$ bits for each server's answer; XOR of 2^s bits for the user.

Parameters of [CGKS'95-98]

Secure and correct.

With 4 servers:

- Communication: $8\sqrt{n}$ uploaded bits, 4 dowloaded bits,
- Storage: replication of F over 4 servers,
- Complexity: in average, XOR of $n/4$ bits for each server's answer; XOR of 4 bits for the user.

Generalizable to 2^s servers:

- Communication: $s2^s n^{1/s}$ uploaded bits, 2^s dowloaded bits, **(ok)**
- Storage: replication of F over 2^s servers, **(heavy cost)**
- Complexity: in average, XOR of $n/2^s$ bits for each server's answer; XOR of 2^s bits for the user. **(non trivial cost)**

Notable works

Recent ideas (aiming at communication/storage improvements):

- *Fazeli, Vardy, Yaakobi '15.*
  PIR codes. Transforms a replication-based PIR into a coded PIR.
- *Sun, Jafar '16.*
  Introduced PIR capacity.
- *El Rouayheb, Freij-Hollanti, Gnilke, Hollanti, Karpuk, Tajeddine '16'17.*
  Built optimal constructions according to PIR capacity.

Practicality

Context: file F is frequently queried (*e.g.* a public database). Notion of *price of privacy* (\$), mainly depending on:

- computational complexity for the servers,
- servers' storage overhead.

Practicality

Context: file F is frequently queried (*e.g.* a public database). Notion of *price of privacy* (\$), mainly depending on:

- computational complexity for the servers,
- servers' storage overhead.

Yekhanin (survey, '12): "the overwhelming computational complexity of PIR schemes (...) currently presents the main bottleneck to their practical deployment".

Practicality

Context: file F is frequently queried (*e.g.* a public database). Notion of *price of privacy* (\$), mainly depending on:

- computational complexity for the servers,
- servers' storage overhead.

Yekhanin (survey, '12): "the overwhelming computational complexity of PIR schemes (...) currently presents the main bottleneck to their practical deployment".

We focus on PIR schemes with **optimal computation complexity** on the server side: each $\mathcal{A}_j$ reads at most 1 symbol.

Practicality

Context: file F is frequently queried (*e.g.* a public database). Notion of *price of privacy* (\$), mainly depending on:

- computational complexity for the servers,
- servers' storage overhead.

Yekhanin (survey, '12): "the overwhelming computational complexity of PIR schemes (...) currently presents the main bottleneck to their practical deployment".

We focus on PIR schemes with **optimal computation complexity** on the server side: each $\mathcal{A}_j$ reads at most 1 symbol.

A few years ago...
Katz, Trevisan '00: Smooth locally decodable codes give PIR protocols.

PIR from smooth local decoding

Definition.– A code $\mathcal{C} \subseteq \mathbb{F}_q^n$ equipped with an encoder $E : \mathbb{F}_q^k \to \mathcal{C}$ is said (ℓ, δ)-locally decodable, if there exists a probabilistic algorithm $\mathcal{D}$ such that, for all $m \in \mathbb{F}_q^k$, $y \in \mathbb{F}_q^n$, such that $d(E(m), y) \leq \delta n$, we have:

- $\forall i \in [1, k]$, $\mathbb{P}(\mathcal{D}^{(y)}(i) = m_i) \geq 2/3$,
- $\forall i \in [1, k]$, $\mathcal{D}^{(y)}(i)$ makes at most ℓ queries to symbols of y.

The code is said *smooth* if, given $i \in [1, k]$, the coordinates of the queries made by $\mathcal{D}^{(y)}(i)$ are uniformly distributed over $[1, n]$.

PIR from smooth local decoding

Definition.– A code $\mathcal{C} \subseteq \mathbb{F}_q^n$ equipped with an encoder $E : \mathbb{F}_q^k \to \mathcal{C}$ is said (ℓ, δ)-locally decodable, if there exists a probabilistic algorithm $\mathcal{D}$ such that, for all $m \in \mathbb{F}_q^k$, $y \in \mathbb{F}_q^n$, such that $d(E(m), y) \leq \delta n$, we have:

- $\forall i \in [1, k]$, $\mathbb{P}(\mathcal{D}^{(y)}(i) = m_i) \geq 2/3$,
- $\forall i \in [1, k]$, $\mathcal{D}^{(y)}(i)$ makes at most ℓ queries to symbols of y.

The code is said *smooth* if, given $i \in [1, k]$, the coordinates of the queries made by $\mathcal{D}^{(y)}(i)$ are uniformly distributed over $[1, n]$.

Examples:

- Hadamard code, low degree Reed-Muller codes
- higher rate: multiplicity codes (Kopparty, Saraf, Yekhanin), lifted codes (Guo, Kopparty, Sudan), expander codes (Hemenway, Ostrowsky, Wootters), ...

PIR from smooth local decoding

From local decoding to PIR (Katz-Trevisan '00).
Given a smooth (ℓ, δ)-LDC $\mathcal{C}$, with encoder E.

1. Use ℓ servers, each storing $E(m)$.
2. For recovering m_i:
   - use $\mathcal{D}^{(y)}(i)$'s query generator to define queries $q_1, \dots q_\ell$,
   - send q_j to server S_j which sends back $a_j = E(m)_{q_j}$,
   - continue running $\mathcal{D}^{(y)}(i)$ on $(a_1, \dots, a_\ell)$ and output the result.

PIR from smooth local decoding

From local decoding to PIR (Katz-Trevisan '00).
Given a smooth (ℓ, δ)-LDC $\mathcal{C}$, with encoder E.

1. Use ℓ servers, each storing $E(m)$.
2. For recovering m_i:
   - use $\mathcal{D}^{(y)}(i)$'s query generator to define queries $q_1, \dots q_\ell$,
   - send q_j to server S_j which sends back $a_j = E(m)_{q_j}$,
   - continue running $\mathcal{D}^{(y)}(i)$ on $(a_1, \dots, a_\ell)$ and output the result.

Features:

- IT-privacy, thanks to smoothness
- ℓ servers storing $E(m)$ (still bad)
- communication $\mathcal{O}(\ell)$
- computation complexity
  - servers: $\mathcal{O}(1)$ (fine)
  - user: highly depends on $\mathcal{D}$

Overview of our solution

Basic ideas:

- Encode the file $F \mapsto c \in \mathcal{C} \subseteq \mathbb{F}_q^n$, **split** c in ℓ parts of size n/ℓ, $\ell \ll n$, and share them among ℓ servers.
- Use **low weight** parity-check equations $h \in \mathcal{C}^\perp$ to retrieve symbols F_i. More specifically, when restricted to the support G_j corresponding to a server S_j:
  - *computation complexity:* the weight of $h_{|G_j}$ is 1;
  - *privacy:* the support of $h_{|G_j}$ is uniformly distributed over G_j.

Overview of our solution

Basic ideas:

- Encode the file $F \mapsto c \in \mathcal{C} \subseteq \mathbb{F}_q^n$, **split** c in ℓ parts of size n/ℓ, $\ell \ll n$, and share them among ℓ servers.
- Use **low weight** parity-check equations $h \in \mathcal{C}^\perp$ to retrieve symbols F_i. More specifically, when restricted to the support G_j corresponding to a server S_j:
  - *computation complexity:* the weight of $h_{|G_j}$ is 1;
  - *privacy:* the support of $h_{|G_j}$ is uniformly distributed over G_j.

Practical solution:

- use codes $\mathcal{C}$ based on transversal designs.

1. The PIR issue

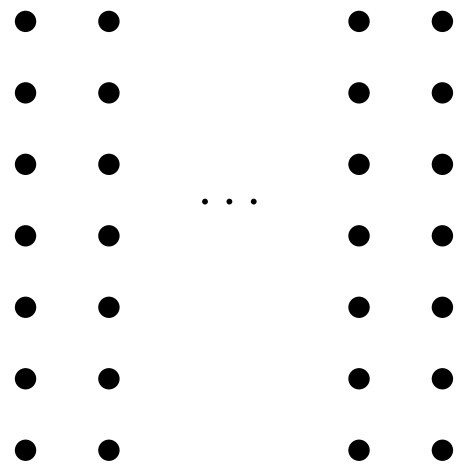
2. Transversal designs for efficient PIR protocols

3. Instances
   - First instance: affine transversal designs
   - Second instance: with orthogonal arrays

Transversal designs

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

- X a set of *points*, $|X| = n = s\ell$,

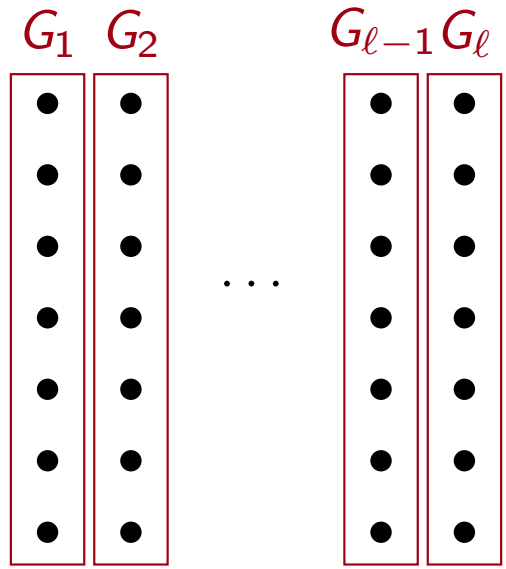


Transversal designs

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

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

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



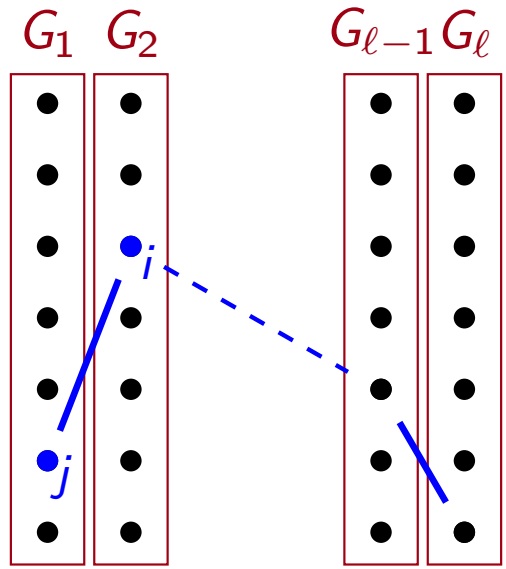
Transversal designs

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

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

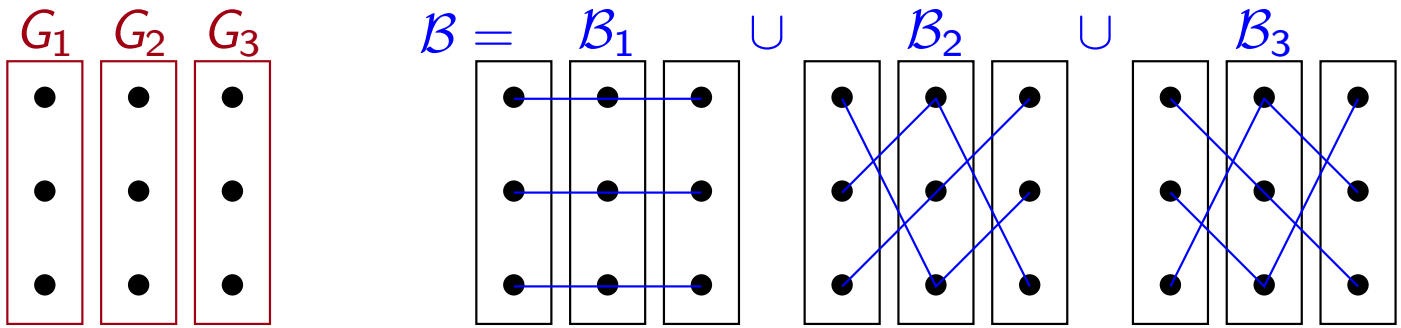
$$X = \coprod_{j=1}^{\ell} G_j \text{ and } |G_j| = s\,,$$

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



Examples of TD

- Points X, parallel hyperplanes $\mathcal{G}$ and transversal lines $\mathcal{B}$ in the affine space $\mathbb{A}^m$. For instance, a $\mathrm{TD}(3,3)$:



- Similar construction in $X = \mathbb{P}^m \setminus A$, $\mathrm{codim}(A) = 2$.
- Combinatorial constructions based on orthogonal arrays, on difference sets...

Codes from designs

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

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

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

Codes from designs

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

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

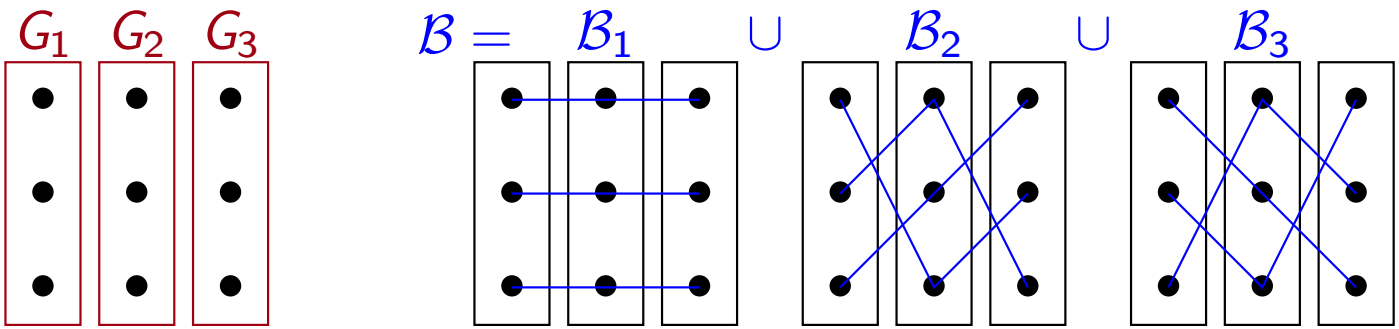
$$M_{i,j} = \begin{cases} 1 & \text{if } x_j \in B_i \\ 0 & \text{otherwise.} \end{cases}$$

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

- $\mathrm{length}(\mathcal{C}) = |X|$,
- $\dim(\mathcal{C}) = \dim(\ker M)$,
- $B \in \mathcal{B} \Rightarrow h \in \mathcal{C}^\perp$, such that $\mathrm{wt}(h_{|G_j}) = 1$, $\forall j = 1, \ldots, \ell$.

Example

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



gives an incidence matrix

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

whose rank over $\mathbb{F}_3$ is 6. $\quad\Longrightarrow\quad$ $\mathcal{C}$ is a $[9,3]_3$ code.

Our PIR protocol construction

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

Our PIR protocol construction

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

- **Initialisation.** User U encodes $F \mapsto c \in \mathcal{C}$, and gives $c_{|G_j}$ to server S_j for $j = 1, \ldots, \ell$.
- **To recover $F_i = c_i$:**
  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 $a_j = \mathcal{A}_j(q_j, c_{|G_j}) := c_{q_j}$
  3. U recovers

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

Privacy and parameters

Theorem.– If the servers do not collude, then our PIR protocol is information-theoretically private.

Privacy and parameters

Theorem.– If the servers do not collude, then our PIR protocol is information-theoretically private.

Proof:

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

Privacy and parameters

Theorem.– If the servers do not collude, then our PIR protocol is information-theoretically private.
Proof:

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

Properties. For $|F| = k \log q$ bits, with $k = \dim \mathcal{C} \leq n = s\ell$.

Privacy and parameters

Theorem.– If the servers do not collude, then our PIR protocol is information-theoretically private.
Proof:

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

Properties. For $|F| = k \log q$ bits, with $k = \dim \mathcal{C} \leq n = s\ell$.

- communication complexity: $\ell(\log s + \log q)$ bits

Privacy and parameters

Theorem.– If the servers do not collude, then our PIR protocol is information-theoretically private.
Proof:

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

Properties. For $|F| = k \log q$ bits, with $k = \dim \mathcal{C} \leq n = s\ell$.

- communication complexity: $\ell(\log s + \log q)$ bits
- computational complexity:
  - $\mathcal{O}(1)$ **for algorithm** $\mathcal{A}$ (somewhat optimal)
  - $\mathcal{O}(\ell)$ $\mathbb{F}_q$-operations for $\mathcal{R}$

Privacy and parameters

Theorem.– If the servers do not collude, then our PIR protocol is information-theoretically private.
Proof:

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

Properties. For $|F| = k \log q$ bits, with $k = \dim \mathcal{C} \leq n = s\ell$.

- communication complexity: $\ell(\log s + \log q)$ bits
- computational complexity:
  - $\mathcal{O}(1)$ **for algorithm** $\mathcal{A}$ (somewhat optimal)
  - $\mathcal{O}(\ell)$ $\mathbb{F}_q$-operations for $\mathcal{R}$
- storage overhead: $(n - k) \log q$ bits

Privacy and parameters

Theorem.– If the servers do not collude, then our PIR protocol is information-theoretically private.
Proof:

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

Properties. For $|F| = k \log q$ bits, with $k = \dim \mathcal{C} \leq n = s\ell$.

- communication complexity: $\ell(\log s + \log q)$ bits
- computational complexity:
  - $\mathcal{O}(1)$ **for algorithm** $\mathcal{A}$ (somewhat optimal)
  - $\mathcal{O}(\ell)$ $\mathbb{F}_q$-operations for $\mathcal{R}$
- storage overhead: $(n-k)\log q$ bits

Question: TDs with good k depending on (ℓ, s)?

1. The PIR issue

2. Transversal designs for efficient PIR protocols

3. Instances

First instance: affine transversal designs

Second instance: with orthogonal arrays

1. The PIR issue

2. Transversal designs for efficient PIR protocols

3. Instances
   - First instance: affine transversal designs
   - Second instance: with orthogonal arrays

A classical TD: points/lines/hyperplanes

Let $\mathcal{T}_A$ be the **classical affine TD**:

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

A classical TD: points/lines/hyperplanes

Let $\mathcal{T}_A$ be the **classical affine TD**:

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

The associated $\mathbb{F}_q$-linear code $\mathcal{C}$ has

- length $n = q^m$
- block size $\ell = q$
- dimension?

A classical TD: points/lines/hyperplanes

Let $\mathcal{T}_A$ be the **classical affine TD**:

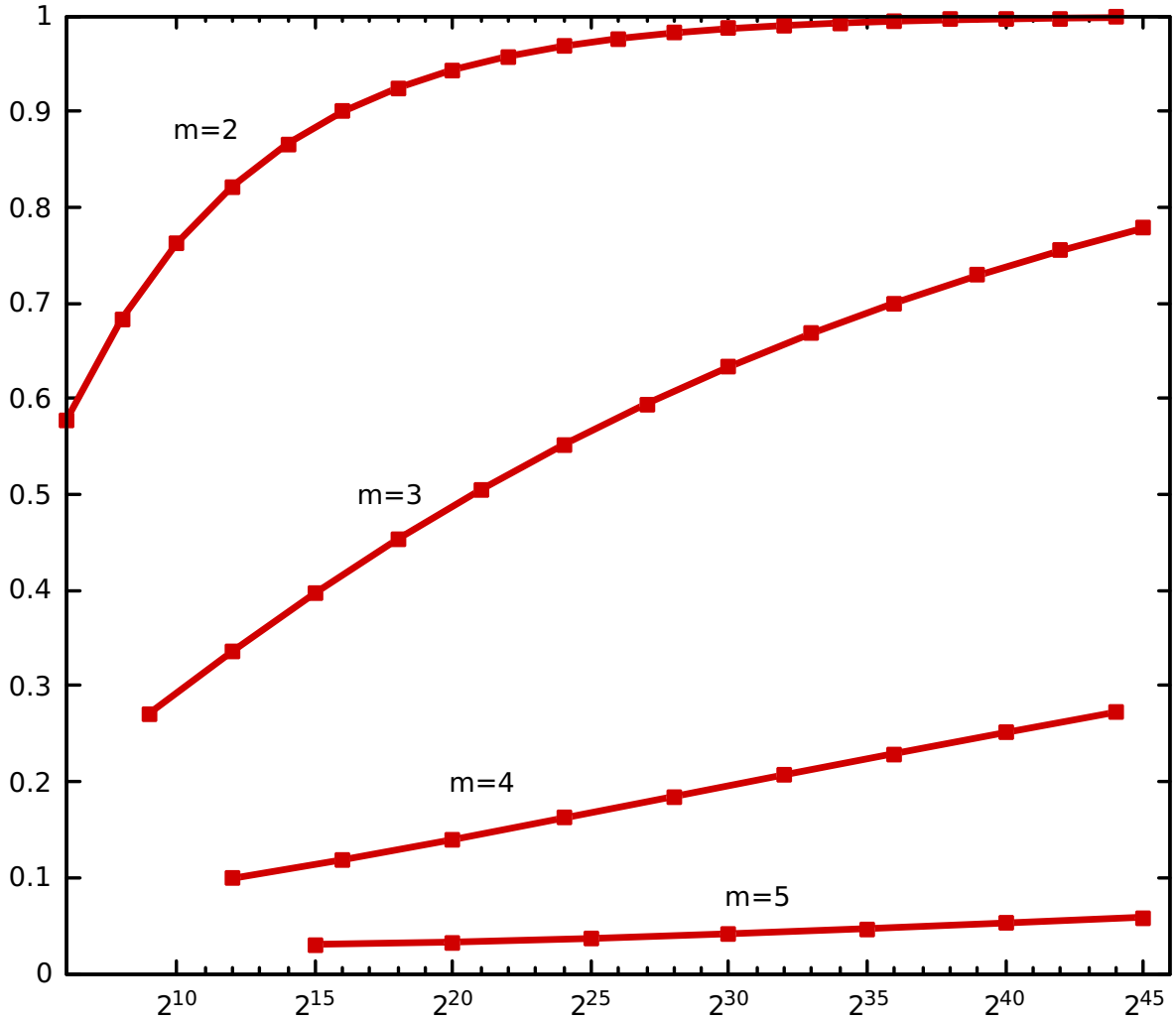
- $X = \mathbb{F}_q^m$, $m \geq 2$,
- $\mathcal{G}$ a set of q disjoint hyperplanes partitionning X,
- $\mathcal{B} = \{\text{affine lines } L \text{ secant to each group of } \mathcal{G}\}$.

The associated $\mathbb{F}_q$-linear code $\mathcal{C}$ has

- length $n = q^m$
- block size $\ell = q$
- dimension?
  - its parity-check matrix has q^m columns and q^{2m-2} rows...
  - ... but $\mathcal{C}$ contains $\mathrm{RM}_q(m, q-2)$ which has rate $\simeq 1/m!$,
  - and sometimes it is even larger.

Lower bounds on rates of TD-based codes

rate $R = k/n$



length $n = 2^{em}$

Particular case: $m = 2$

For $m = 2$, $q = p^e$, using Hamada's formula [Ham68] we obtain:

$$n = p^{2e}, \quad k \geq p^{2e} - \binom{p+1}{2}^e, \quad \ell = \sqrt{n}\,.$$

[Ham68] N Hamada. *The rank of the incidence matrix of points and d-flats in finite geometries*. J. of Science of the Hiroshima Univ., Series A-I (Maths), 32(2):381–396, 1968.

Particular case: $m = 2$

For $m = 2$, $q = p^e$, using Hamada's formula [Ham68] we obtain:

$$n = p^{2e}, \quad k \geq p^{2e} - \binom{p+1}{2}^e, \quad \ell = \sqrt{n}\,.$$

Asymptotically ($e \to \infty$, fixed p):

$$\begin{cases} R = k/n & = 1 - \Theta(n^{c_p}) \\ \ell & = \Theta(\sqrt{n}) \end{cases}$$

where $c_p = \frac{1}{2}(\log_p(\frac{p+1}{2}) - 1) < 0$.

Moreover, $c_p \nearrow$, with $c_2 = -0.208$ and $c_\infty = 0$.

[Ham68] N Hamada. *The rank of the incidence matrix of points and d-flats in finite geometries*. J. of Science of the Hiroshima Univ., Series A-I (Maths), 32(2):381–396, 1968.

Particular case: $m = 2$

For $m = 2$, $q = p^e$, using Hamada's formula [Ham68] we obtain:

$$n = p^{2e}, \quad k \geq p^{2e} - \binom{p+1}{2}^e, \quad \ell = \sqrt{n}\,.$$

Asymptotically ($e \to \infty$, fixed p):

$$\begin{cases} R = k/n &= 1 - \Theta(n^{c_p}) \\ \ell &= \Theta(\sqrt{n}) \end{cases}$$

where $c_p = \frac{1}{2}(\log_p(\frac{p+1}{2}) - 1) < 0$.

Moreover, $c_p \nearrow$, with $c_2 = -0.208$ and $c_\infty = 0$.

Question:

- Here, with ℓ servers, PIR rate $= 1/\ell$ and code rate $= 1 - \Theta(\ell^{-0.41})$
  Is this instance rate-optimal for "read-only" non-collusive PIRs?

[Ham68] N Hamada. *The rank of the incidence matrix of points and d-flats in finite geometries*. J. of Science of the Hiroshima Univ., Series A-I (Maths), 32(2):381–396, 1968.

1. The PIR issue

2. Transversal designs for efficient PIR protocols

3. Instances
   - First instance: affine transversal designs
   - Second instance: with orthogonal arrays

TDs and orthogonal arrays

An *orthogonal array* $\mathrm{OA}(t, \ell, s)$ *of strength* t may be seen as a list of codewords over S, with:

- $|S| = s$,
- length ℓ,
- and dual distance $d^{\perp} = t + 1$

TDs and orthogonal arrays

An *orthogonal array* $\mathrm{OA}(t, \ell, s)$ *of strength* t may be seen as a list of codewords over S, with:

- $|S| = s$,
- length ℓ,
- and dual distance $d^{\perp} = t + 1$

$$S = \{a, b\}$$

$$OA(2, 3, 2) = \begin{bmatrix} a & b & b \\ b & b & a \\ b & a & b \\ a & a & a \end{bmatrix}$$

TDs and orthogonal arrays

An *orthogonal array* $\mathrm{OA}(t, \ell, s)$ *of strength* t may be seen as a list of codewords over S, with:

- $|S| = s$,
- length ℓ,
- and dual distance $d^{\perp} = t + 1$

$$S = \{a, b\}$$

$$OA(2, 3, 2) = \begin{bmatrix} a & b & b \\ b & b & a \\ b & a & b \\ a & a & a \end{bmatrix}$$

Construction OA $\rightarrow$ TD :

- $X = S \times [1, \ell]$
- $\mathcal{G} = \{S \times \{i\}, 1 \leq i \leq \ell\}$

$(a, 1)$	$(a, 2)$	$(a, 3)$
$(b, 1)$	$(b, 2)$	$(b, 3)$

TDs and orthogonal arrays

An *orthogonal array* $\mathrm{OA}(t, \ell, s)$ *of strength* t may be seen as a list of codewords over S, with:

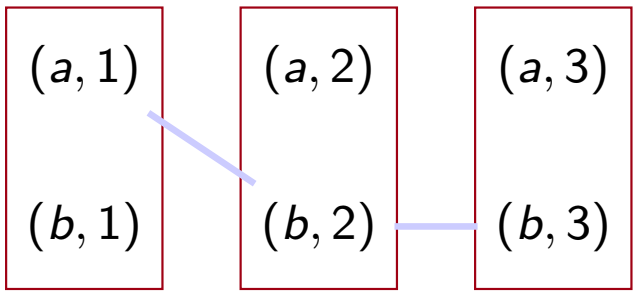
- $|S| = s$,
- length ℓ,
- and dual distance $d^\perp = t + 1$

$S = \{a, b\}$

$$OA(2,3,2) = \begin{bmatrix} a & b & b \\ b & b & a \\ b & a & b \\ a & a & a \end{bmatrix}$$

Construction OA $\to$ TD :

- $X = S \times [1, \ell]$
- $\mathcal{G} = \{S \times \{i\}, 1 \le i \le \ell\}$
- $\mathcal{B} = \{\{(c_i, i), 1 \le i \le \ell\}, c \in \mathrm{OA}\}$



TDs and orthogonal arrays

An *orthogonal array* $\mathrm{OA}(t, \ell, s)$ *of strength* t may be seen as a list of codewords over S, with:

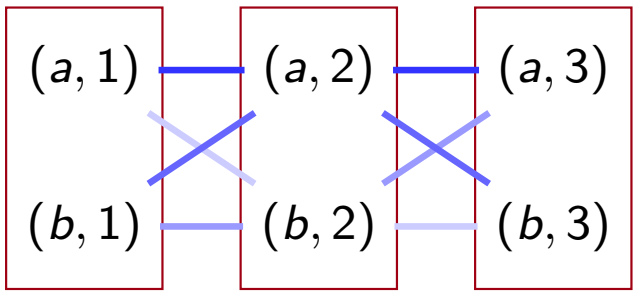
- $|S| = s$,
- length ℓ,
- and dual distance $d^\perp = t + 1$

$S = \{a, b\}$

$$OA(2,3,2) = \begin{bmatrix} a & b & b \\ b & b & a \\ b & a & b \\ a & a & a \end{bmatrix}$$

Construction OA $\to$ TD :

- $X = S \times [1, \ell]$
- $\mathcal{G} = \{S \times \{i\}, 1 \le i \le \ell\}$
- $\mathcal{B} = \{\{(c_i, i), 1 \le i \le \ell\}, c \in \mathrm{OA}\}$



TDs and orthogonal arrays

An *orthogonal array* $\mathrm{OA}(t, \ell, s)$ *of strength* t may be seen as a list of codewords over S, with:

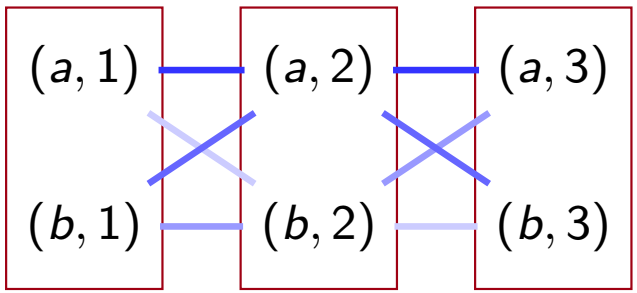
- $|S| = s$,
- length ℓ,
- and dual distance $d^\perp = t + 1$

$S = \{a, b\}$

$$OA(2,3,2) = \begin{bmatrix} a & b & b \\ b & b & a \\ b & a & b \\ a & a & a \end{bmatrix}$$

Construction OA $\to$ TD :

- $X = S \times [1, \ell]$
- $\mathcal{G} = \{S \times \{i\}, 1 \le i \le \ell\}$
- $\mathcal{B} = \{\{(c_i, i), 1 \le i \le \ell\}, c \in \mathrm{OA}\}$



Prop. If $t = 2$, then we obtain a $\mathrm{TD}(\ell, s)$ from an $\mathrm{OA}(t, \ell, s)$.

Resistance to collusions

Experiments: for $t = 2$ and small ℓ and s, the classical affine TD leads to the best code dimension.

Resistance to collusions

Experiments: for $t = 2$ and small ℓ and s, the classical affine TD leads to the best code dimension.

What about $\mathrm{OA}(t, \ell, s)$ with $t > 2$?

Resulting TD satisfies: for each t-tuple of points lying in t different groups, there is a block which contains them all.

$\Rightarrow$ Our PIR protocol resists $t - 1$ collusive servers.

The *incidence code* construction

Definition.– We call *incidence code* of $\mathcal{C}_0$, denoted $I_q(\mathcal{C}_0)$, the $\mathbb{F}_q$-linear code $\mathcal{C}$ coming from the successive constructions:

$$\mathcal{C}_0 = \mathrm{OA}(t, \ell, s) \quad \mapsto \quad \text{generalized } \mathrm{TD}(\ell, s; t) \quad \mapsto \quad \mathcal{C} = I_q(\mathcal{C}_0)$$

The *incidence code* construction

Definition.– We call *incidence code* of $\mathcal{C}_0$, denoted $I_q(\mathcal{C}_0)$, the $\mathbb{F}_q$-linear code $\mathcal{C}$ coming from the successive constructions:

$$\mathcal{C}_0 = \mathrm{OA}(t, \ell, s) \quad \mapsto \quad \text{generalized } \mathrm{TD}(\ell, s; t) \quad \mapsto \quad \mathcal{C} = I_q(\mathcal{C}_0)$$

We derive PIR parameters from those of $\mathcal{C}_0$:

- $d^{\perp}(\mathcal{C}_0) - 2$ is the number of collusive servers the protocol resists
- $I_q(\cdot)$ is decreasing w.r.t. inclusion of codes
  $\Rightarrow$ the larger $\mathcal{C}_0$, the larger PIR storage overhead

The *incidence code* construction

Definition.– We call *incidence code* of $\mathcal{C}_0$, denoted $I_q(\mathcal{C}_0)$, the $\mathbb{F}_q$-linear code $\mathcal{C}$ coming from the successive constructions:

$$\mathcal{C}_0 = \mathrm{OA}(t, \ell, s) \quad \mapsto \quad \text{generalized } \mathrm{TD}(\ell, s; t) \quad \mapsto \quad \mathcal{C} = I_q(\mathcal{C}_0)$$

We derive PIR parameters from those of $\mathcal{C}_0$:

- $d^{\perp}(\mathcal{C}_0) - 2$ is the number of collusive servers the protocol resists
- $I_q(\cdot)$ is decreasing w.r.t. inclusion of codes
  $\Rightarrow$ the larger $\mathcal{C}_0$, the larger PIR storage overhead

let's use MDS codes for $\mathcal{C}_0$

Incidence codes of Reed-Solomon codes

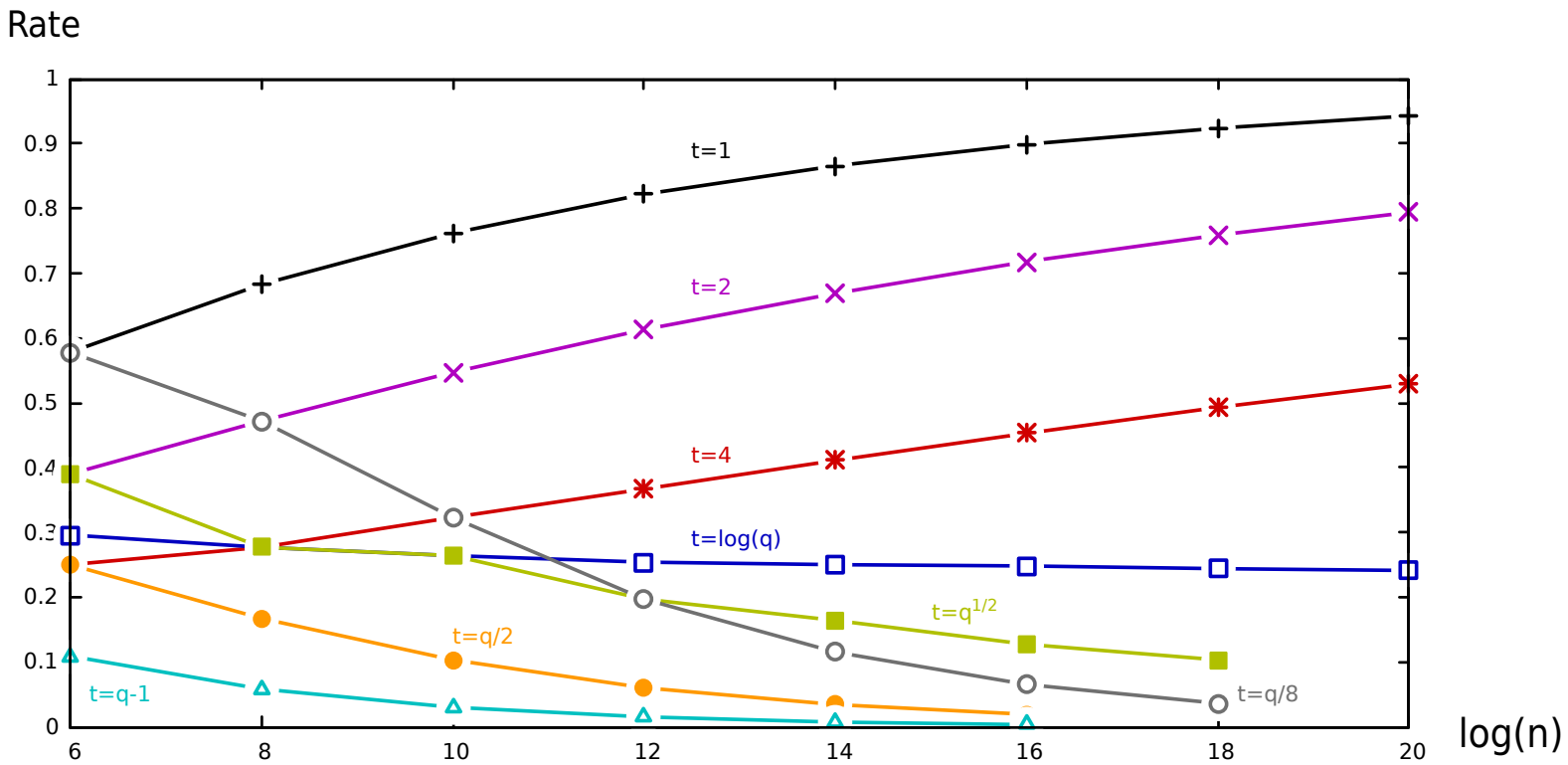
Example: for $\mathcal{C}_0 = \mathrm{RS}(\mathbb{F}_q, t+1)$, file F, $|F| = Rq^2 \log q$ bits, R the rate of the incidence code:

- requires q servers, resists t colluding ones,
- communication complexity $q \log q$ bits,
- optimal computation complexity; rate R given by:

Incidence codes of Reed-Solomon codes

Example: for $\mathcal{C}_0 = \mathrm{RS}(\mathbb{F}_q, t+1)$, file F, $|F| = Rq^2 \log q$ bits, R the rate of the incidence code:

- requires q servers, resists t colluding ones,
- communication complexity $q \log q$ bits,
- optimal computation complexity; rate R given by:



Conclusion

Summary: (server-)efficient PIR protocols can be built with codes based on transversal designs

Conclusion

Summary: (server-)efficient PIR protocols can be built with codes based on transversal designs

Issues related to this construction:

- find transversal designs leading to largest codes,
- bounds, optimal constructions,
- (divisible projective linear codes $\mathcal{C}_0$ over large alphabets?).

Conclusion

Summary: (server-)efficient PIR protocols can be built with codes based on transversal designs

Issues related to this construction:

- find transversal designs leading to largest codes,
- bounds, optimal constructions,
- (divisible projective linear codes $\mathcal{C}_0$ over large alphabets?).

Other issues:

- capacity of PIR with "read-only" servers?
- which PIR protocols with "read-only" servers can be modeled as PIR with incidence codes? (all of them?)
- links between this construction and others?

Thank you for your attention.
Questions?

Remarks on incidence codes

Proposition.– For any code $\mathcal{C}_0$ of length ℓ over $\mathbb{F}_s$, the incidence code $I_q(\mathcal{C}_0)$ is an $[n, k]_q$ code with:

- $n = s\ell$,
- $\ell - 1 \leq k \leq n - \Omega(\sqrt{n})$.

Proposition.– Let H be the parity-check matrix of $I_q(\mathcal{C}_0)$. Then,

$$HH^T = \ell J - D(\mathcal{C}_0)\,,$$

where J is the all-1 matrix and

$$D(\mathcal{C}_0)_{c,c'} = d(c, c'), \quad \forall c, c' \in \mathcal{C}_0\,.$$

Divisible codes for efficient PIR protocols

A p-divisible code is a code whose codewords' weights are divisible by p.

Corollary.– If $\mathcal{C}_0$ is p-divisible for $p = \mathrm{char}(\mathbb{F}_q)$, then:

$$k = \dim I_q(\mathcal{C}_0) \geq \frac{n-1}{2}\,.$$

Furthermore, if $p \,|\, \ell$, then:

$$HH^T = 0 \quad \Rightarrow \quad \mathcal{C}^\perp \subseteq \mathcal{C}$$

Theorem.– If there exists a p-divisible code $\mathcal{C}_0$ of length ℓ and dual distance $t+2$, then there exists a PIR protocol resisting to t colluding servers, with rate $\gtrsim 1/2$ and optimal computational complexity.

Question.– Do there exist projective ($d^\perp \geq 3$) p-divisible codes of length ℓ over $\mathbb{F}_q$ (with $q \gg \ell$, or $d^\perp$ large)?