Constructions for efficient Private Information Retrieval protocols

Julien Lavauzelle

Laboratoire LIX, École Polytechnique, Inria & CNRS UMR 7161
Université Paris-Saclay
julien.lavauzelle@inria.fr

Abstract. Private Information Retrieval (PIR) protocols aim at ensuring a user that he can retrieve some part D_i of a distributed database D without revealing the index i to the server(s). Most of known PIR protocols focus on decreasing the communication complexity between the client and the server(s). Recently, the use of *PIR codes* by Fazeli *et. al.* also lead to a huge reduction of the storage overhead supported by the servers.
However, only a few works address the issue of the computational complexity of the servers. In this paper, we show that transversal designs and their generalizations provide PIR schemes achieving simultaneously reasonable communication complexity, low storage overhead, optimal computational complexity for the servers, and resistance to a collusion of some of them.

1 Introduction

1.1 Private Information Retrieval

A Private Information Retrieval (PIR) protocol allows a user to retrieve entries of a database without revealing the identity of the desired item. Such protocols can be applied in medical data storage where, for example, physicians could access parts of the genome while hiding the specific gene they analyse. The PIR paradigm was originally introduced Chor, Goldreich, Kushilevitz and Sudan in [6, 7].

A naive solution is to download the entire database each time the user wants a single symbol. In this setting the communication complexity is overwhelming, so we look for PIR protocols which exchange less bits. However, Chor *et. al.* proved that, when the n-bit database is stored on a single server, a PIR protocol cannot be information-theoretically secure with less than $\Theta(n)$ bits of communication [7]. Two alternatives were then considered: restricting the protocol to a computational security (initiated by Chor and Gilboa [5]), or allowing several servers to store the database. Our work focuses on the last one.

In most of such PIR protocols, the database is replicated on the servers, and each server is asked to compute some partial information related to a random query sent by the user — this query shall hide the index of the symbol the user wants. Then the user collects all the servers' answers and retrieves the desired symbol with an appropriate algorithm. For example, Chor *et. al.* [7] used XOR properties on $\log(n)$-dimensional vectors to hide queries and retrieve data symbols. A few years later, Katz and Trevisan showed in [13] that any smooth locally decodable code (LDC) gives rise to a PIR protocol whose number of servers and communication complexity correspond to the LDC locality (Yekhanin gives a good survey on LDCs in [17]). Building on this idea, many PIR schemes (notably [3, 16, 10, 9]) successively decreased the communication complexity to $\mathcal{O}(n^{\sqrt{\log\log n/\log n}})$. However, only few of them tried to lighten the computational and storage cost on the server side.

By preprocessing the database, Beimel, Ishai and Malkin [4] were the first to address the minimization of the server storage/computation. Then, initiated by Fazeli, Vardy and Yaakobi [11], recent works used the concept of *PIR codes* to transform a k-server replication-based PIR protocol into a more-than-k-server PIR protocol which uses less storage. The idea

is to encode the database into a codeword and distribute parts of this codeword among the servers. Through this transformation, both communication complexity and computational cost keep the same order of magnitude, but the overall storage overhead ratio is reduced to the PIR code one, which can be arbitrarily brought to 1 when sufficiently many servers are used. Again, while the storage drawback seems to be solved, a huge computational cost may still represent a barrier to PIR practicality.

1.2 Motivations and results

As pointed out by Yekhanin [17], "the overwhelming computational complexity of PIR schemes (...) currently presents the main bottleneck to their practical deployment". Indeed, if public database is frequently queried (for instance, consider a database storing stock exchange prices), one cannot afford a PIR protocol with, for each query, a linear computational complexity in the length of the database.

Naively, this computational cost could be drastically reduced by letting the user precompute and send to the servers all the possible answers to its queries. Of course, storing all these answers dramatically increases the needed storage, and we prefer to focus on another construction due to Augot, Levy-dit-Vehel and Shikfa [2] that we shortly explain in the next section.

In this work, we generalize the construction from [2] by modelling the PIR security constraints in terms of block designs, and build linear codes upon them. It leads to PIR schemes with low communication complexity, low storage overhead and constant computational complexity which can be generalized in order to resist to collusions of servers.

2 Definitions and related constructions

2.1 Definitions

In all what follows, let U be the owner of a database $D = (D_i)_{i \in I} \in \mathbb{F}_q^{|I|}$, and $S_1, \ldots, S_\ell$ be ℓ servers involved in the PIR protocol. The standard definition of (information-theoretically secure) replication-based PIR protocols is the following:

Definition 1 (standard, or replication-based PIR protocol). *Assume that each server S_j stores a copy of the database D. An ℓ-server standard PIR protocol is a set of three algorithms $\mathcal{P} = (\mathcal{Q}, \mathcal{A}, \mathcal{R})$ which run the following steps on input $i \in I$:*

1. Query generation: *the randomized algorithm $\mathcal{Q}$ generates ℓ queries $(q_1, \ldots, q_\ell) = \mathcal{Q}(i)$. Query q_j is sent to the server S_j.*
2. Servers' answer: *each server S_j computes an answer $a_j = \mathcal{A}(q_j, D)$ and sends it back to the user*[1].
3. Reconstruction: *the user reconstructs $r = \mathcal{R}(i, (a_j), (q_j))$.*

The PIR protocol is said:

- correct *if $r = D_i$ when the servers follow the protocol.*
- t-private *if, for every $(i, i') \in I^2$ and $T \subseteq [1, \ell]$ such that $|T| \leq t$, the distributions $\mathcal{Q}(i)_{|T}$ and $\mathcal{Q}(i')_{|T}$ are the same. We also say that the PIR protocol resists to t collusions.*

We call communication complexity *the number of bits sent between the user and the servers, and* server *(resp.* user*)* computational complexity *the overall number of $\mathbb{F}_q$-operations made by $\mathcal{A}$ in order to compute every answer a_j (resp. made by both $\mathcal{Q}$ and $\mathcal{R}$).*

We now want to model a PIR protocol where the database can be encoded and distributed over the servers. Thus, from now on, $D = (D_i)_{i \in I}$ denotes *the encoding of a database.* Besides, we assume that $I = [1, n] \times [1, \ell]$ and for readability we write $D_{(i_1, i_2)} = D_{i_1}^{(i_2)}$ and $D^{(i_2)} = (D_k^{(i_2)})_{k \in [1,n]}$.

[1] algorithm $\mathcal{A} := \mathcal{A}_j$ may generically depend on j

Definition 2 (distributed PIR protocol). *Assume that for $1 \leq j \leq \ell$, the server S_j holds the part $D^{(j)}$ of the database. An ℓ–server distributed* PIR *protocol is a set of three algorithms $\mathcal{P} = (\mathcal{Q}, \mathcal{A}, \mathcal{R})$ running the following steps on input $i \in I$:*

1. Query generation: *the randomized algorithm $\mathcal{Q}$ generates ℓ queries $(q_1, \ldots, q_\ell) = \mathcal{Q}(i)$. Query q_j is sent to the server S_j.*
2. Servers' answer: *each server S_j computes an answer $a_j = \mathcal{A}(q_j, D^{(j)})$ and sends it back to the user.*
3. Reconstruction: *the user reconstructs $r = \mathcal{R}(i, (a_j), (q_j))$.*

The correctness and privacy properties are identical to the standard protocol. Finally, as the database D has been previously encoded, we call storage overhead *the number of redundancy bits stored by the servers.*

2.2 Partition of the database into several servers

Based on [13], we briefly recall how to design a *standard* PIR protocol based on a perfectly smooth locally decodable code (LDC, see [17] for a formal definition of these codes). With the previous notations, let us say the user wants to privately retrieve entries of a database $D \in \mathbb{F}_q^{|I|}$, and assume there exists $\mathcal{C} \subset \mathbb{F}_q^n$, a perfectly smooth LDC of dimension $|I|$ and locality ℓ. A replication-based ℓ-server PIR protocol based on the code $\mathcal{C}$ is described in Figure 1.

We will use the local decoding algorithm $\mathcal{D}$ of the code $\mathcal{C}$. Assume the user wants to retrieve the symbol D_i for $i \in I$, and denote by $\mathcal{C}(D)$ the encoding of D *via* $\mathcal{C}$.

1. *Queries generation.* Using the local decoder $\mathcal{D}$, the user U generates at random queries $(q_1, \ldots, q_\ell)$ for decoding D_i. Query q_j is sent to server S_j.
2. *Server answers.* Each server S_j computes the symbol $\mathcal{C}(D)_{q_j}$ and sends it back to the user.
3. *Reconstruction.* The user runs the local decoder $\mathcal{D}$ on the $(\mathcal{C}(D)_{q_j})_{j \in [1,\ell]}$ and retrieves D_i.

Fig. 1: A standard PIR protocol based on a perfectly smooth locally decodable code.

Augot, Levy-dit-Vehel and Shikfa motivated their work [2] with the perspective of reducing of the total storage carried by the servers. Their construction uses a specific family of LDCs called *multiplicity codes* [14]. But instead of *replicating* the database on several servers, they *split* an encoded version c of the database D and share the parts on these servers. Without going into technicalities — see [2, 14] for more details — we give a sketch of the idea. Multiplicity codes have the property that a codeword c can be seen as the evaluation of a polynomial f_c and its derivatives over the space $\mathbb{F}_q^m$. Now, on every affine line there exist relations between f_c and its derivatives, leading to a local decoder which, when decoding a symbol in position $i \in \mathbb{F}_q^m$, picks random affine lines passing through i and computes linear combinations of the symbols indexed by these lines.

Augot *et. al.* [2] then realized that partitionning $\mathbb{F}_q^m$ into q hyperplanes gives rise to storage improvements. By splitting a codeword according to these hyperplanes and giving one part to each of the q servers, they obtained a huge cutdown on both the total storage and the number of servers, while keeping a reasonable communication complexity. Their transformation induces a minor modification of the query generation process (the only server which holds the desired symbol must receive a random query), but the PIR scheme they built was at that time the only one to let the servers store less than twice the size of the database. Moreover, the precomputation of the encoding of the database ensures a constant-time computational complexity to the servers.

In the subsequent sections, we will focus on this "codeword support splitting" idea and reformulate it in terms of block designs. It produces a large family of codes leading to PIR schemes with low storage and low computational complexity on the server side.

3 Transversal designs and codes

Let us recall the definition of some combinatorial structures and linear codes based on them. See the following books for complementary details: [1], [15] and [8].

Definition 3 (Block design). *A* block design *is a pair $\mathcal{D} = (X, \mathcal{B})$ where X is a finite set called the* points*, and $\mathcal{B}$ is a set of non-empty subsets of X called the* blocks*.*

Definition 4 (Incidence matrix). *Let $\mathcal{D} = (X, \mathcal{B})$ be a block design. The* incidence matrix *$M_{\mathcal{D}}$ of $\mathcal{D}$ is the $|\mathcal{B}| \times |X|$ matrix whose (i, j)-entry is:*

$$\begin{cases} 1 & \text{if the block } i \text{ contains the point } j \\ 0 & \text{otherwise} \end{cases} \qquad \forall i \in \mathcal{B},\, j \in X\,.$$

*The i-th row of this matrix is called incidence vector of the block $B_i \in \mathcal{B}$, and denoted $\mathbb{1}_{B_i}$. We also call q-*rank *of $M_{\mathcal{D}}$ the rank of $M_{\mathcal{D}}$ over the field $\mathbb{F}_q$.*

Example 1. Let $\mathbb{A}^2(\mathbb{F}_3)$ the affine plane over $\mathbb{F}_3$, and X be the 9 points in $\mathbb{A}^2(\mathbb{F}_3)$. We define 12 blocks in $\mathcal{B}$ as the set of lines in $\mathbb{A}^2(\mathbb{F}_3)$. Then the (12×9)–incidence matrix of $\mathcal{D} = (X, \mathcal{B})$ is

$$M_{\mathcal{D}} = \begin{pmatrix} 1&1&1&0&0&0&0&0&0 \\ 1&0&0&1&0&0&1&0&0 \\ 1&0&0&0&1&0&0&0&1 \\ 1&0&0&0&0&1&0&1&0 \\ 0&1&0&1&0&0&0&0&1 \\ 0&1&0&0&1&0&0&1&0 \\ 0&1&0&0&0&1&1&0&0 \\ 0&0&1&1&0&0&0&1&0 \\ 0&0&1&0&1&0&1&0&0 \\ 0&0&1&0&0&1&0&0&1 \\ 0&0&0&1&1&1&0&0&0 \\ 0&0&0&0&0&0&1&1&1 \end{pmatrix}.$$

Over $\mathbb{F}_2$, $M_{\mathcal{D}}$ has full-rank 9, while over $\mathbb{F}_3$, it has rank 6.

Definition 5 (Transversal design). *Let $n, k \geq 2$ and $\lambda \geq 1$ be integers. A* transversal design*, denoted $\mathrm{TD}_\lambda(k, n)$, is a block design $(X, \mathcal{B})$ equipped with a partition $\mathcal{G}$ of X called the* groups *such that:*

- *$|X| = nk$;*
- *any group in $\mathcal{G}$ has size n and any block in $\mathcal{B}$ has size k;*
- *any unordered pair of elements from X is contained in exactly one group or in exacly λ blocks, but not both.*

When $\lambda = 1$, we simply write $\mathrm{TD}(k, n)$.

Remark 1. There are k groups and λn^2 blocks in $\mathrm{TD}_\lambda(k, n)$. Notice that a block cannot be secant to a group with multiplicity more than 1, otherwise the third condition of the definition would be disproved. Moreover, as the block size equals the number of groups, any block must meet any group. Thus the following holds:

$$\forall (B, G) \in \mathcal{B} \times \mathcal{G},\, |B \cap G| = 1\,.$$

Example 2. Start from the design $\mathcal{D} = (X, \mathcal{B})$ defined in Example 1. Define $\mathcal{G}$ to be a set of 3 parallel lines from $\mathcal{B}$ which partitions the point set X. Then the design $(X, \mathcal{B} \setminus \mathcal{G}, \mathcal{G})$ is a $\mathrm{TD}(3, 3)$. Generally, for any prime power q, a $\mathrm{TD}(q, q)$ can be built with the affine plane $\mathbb{A}^2(\mathbb{F}_q)$.

Definition 6 (Code of a design). *Let $\mathbb{F}_q$ be a finite field, and $\mathcal{D} = (X, \mathcal{B})$ be a block design. The code $\mathrm{Code}_q(\mathcal{D})$ is the $\mathbb{F}_q$-linear code of length $|X|$ whose dual code is spanned by the rows of the incidence matrix $M_{\mathcal{D}}$ of the design $\mathcal{D}$. The dimension over $\mathbb{F}_q$ of $\mathrm{Code}_q(\mathcal{D})$ is $n - \mathrm{rank}_p(M_{\mathcal{D}})$ where p is the characteristic of the field $\mathbb{F}_q$.*

Example 3. The design from Example 1 gives a linear code over $\mathbb{F}_3$ of length 9 and dimension 3. One may notice that this code is indeed the generalized Reed-Muller code of degree 1 and order 2 over $\mathbb{F}_3$.

4 Construction of 1-private PIR protocols based on transversal designs

In this section we present our PIR protocol construction which relies on transversal designs. This first idea leads to PIR protocols which are 1-private — see section 5 for t-private PIR with $t > 1$.

4.1 The distributed PIR protocol

To comply with standard coding theory notations, we now denote by ℓ the number of groups of a transversal design (k usually represents the dimension of a code). Let $\mathcal{T} = (X, \mathcal{B}, \mathcal{G})$ be a transversal design $\mathrm{TD}(\ell, n)$ and $N = |X| = \ell n$. Denote by $\mathcal{C} = \mathrm{Code}_q(\mathcal{T}) \subseteq \mathbb{F}_q^N$ the associated $\mathbb{F}_q$-linear code, and write $k = \dim_{\mathbb{F}_q} \mathcal{C}$. Our PIR protocol is defined in Figure 2 and the overall construction is summarized in Figure 3.

1) Initialisation step.

1. *Encoding.* The database $D \in \mathbb{F}_q^k$ is encoded into a codeword $c \in \mathcal{C} = \mathrm{Code}_q(\mathcal{T})$ of length $N = \ell n$.
2. *Distribution.* Denote by $c_{|G_j}$ the symbols of c whose support is the group $G_j \in \mathcal{G}$. Each server S_j receives $c_{|G_j}$.

2) Retrieving step. For each symbol c_i the user U wants to retrieve, denote by $j^* \in [1, \ell]$ the index of the unique group G_{j^*} which contains i. Also denote by $\mathcal{B}^*$ the set of blocks which contains i. The three steps of the distributed PIR protocol are:

1. *Queries generation.* The user U uniformly picks a random block $B \in \mathcal{B}^*$. For every server S_j with $j \neq j^*$, the user sends to S_j the unique index q_j in $B \cap G_j$. The server S_{j^*} receives a random query q_{j^*} uniformly picked in G_{j^*}. To sum up,
$$\begin{cases} B \xleftarrow{\$} \mathcal{B}^* \\ \mathcal{Q}(i)_{j^*} = q_{j^*} \xleftarrow{\$} G_{j^*} \\ \mathcal{Q}(i)_j = q_j \in B \cap G_j,\ j \neq j^* \end{cases}$$
2. *Servers' answer.* Each server S_j (including S_{j^*}) reads $a_j = c_{q_j}$ and sends it back to the user:
$$\mathcal{A}(q_j, c^{(j)}) = c_{q_j}\ .$$
3. *Reconstruction.* The user U computes $x = -\sum_{j \neq j^*} a_j$ and outputs x.

Fig. 2: A 1-private distributed PIR protocol based on a transversal design $\mathcal{T} = (X, \mathcal{B}, \mathcal{G})$

$$\text{Transversal design} \xrightarrow[\text{matrix}]{\text{incidence}} \text{Design-based linear code} \xrightarrow{\text{database encoding}} \text{Distributed PIR scheme}$$

Fig. 3: The generic transversal-design-based PIR scheme.

4.2 Analysis

We analyse our PIR scheme by proving the following theorem:

Theorem 1. *Let D be a database with k entries over a field $\mathbb{F}_q$, and $\mathcal{T} = \mathrm{TD}(\ell, n)$ be a transversal design, whose incidence matrix has rank $\ell n - k$ over $\mathbb{F}_q$. Then, there exists a distributed ℓ-server 1-private PIR protocol with:*

- *only 1 symbol to read for each server,*
- *$\ell - 1$ field operations for the user,*

– $\ell \log(nq)$ *bits of communication,*
– *a storage overhead of* $(\ell n - k) \log q$ *bits on the servers.*

Proof. **Correctness.** Let $\mathcal{C} = \mathrm{Code}_q(\mathcal{T})$. From the definition of the code, the incidence vector $\mathbb{1}_B$ of a block $B \in \mathcal{B}$ belongs to the dual code $\mathcal{C}^\perp$. So $\mathbb{1}_B \cdot c = 0$ which leads to $\sum_{j \in B} c_j = 0$. So our PIR is correct as long as there is no error on the symbols a_j returned by the servers.

Security (1-privacy). We prove that for all $j \in [1, \ell]$, it holds that $\mathbb{P}(i \,|\, q_j) = \mathbb{P}(i)$, where probabilities are taken over the randomness of $B \leftarrow \mathcal{B}^*$. Hence we have

$$\begin{aligned}\mathbb{P}(i \,|\, q_j) &= \mathbb{P}(i \,|\, q_j \text{ and } i \in G_j)\mathbb{P}(i \in G_j) + \mathbb{P}(i \,|\, q_j \text{ and } i \notin G_j)\mathbb{P}(i \notin G_j) \\ &= \mathbb{P}(i \,|\, i \in G_j)\mathbb{P}(i \in G_j) + \mathbb{P}(i \,|\, i \notin G_j)\mathbb{P}(i \notin G_j) = \mathbb{P}(i)\,.\end{aligned}$$

Above, the reasons why we eliminated the random variable q_j in the conditional probabilities are:

– in the case $i \in G_j$ (that is, $j = j^*$), by the very construction of the PIR we know q_j and i are independent;
– in the case $i \notin G_j$, by definition of a transversal design, there are as many blocks containing both q_j and i as there are blocks containing q_j and any i' in $X \setminus G_j$. Indeed, by definition of a transversal design the number of such blocks is always λ. So once again, the value of the random variable q_j is irrelated to i.

Communication complexity. For each server, exactly 1 position in $[1, n]$ and 1 symbol from $\mathbb{F}_q$ are exchanged. So the overall communication complexity is $\ell \times (\log(n) + \log(q))$ bits.

Storage overhead. The number of bits stored on a server is $\frac{N}{\ell} \log |\mathbb{F}_q| = n \log q$, giving a storage overhead of $(\ell n - k) \log q = (N - k) \log q$.

Computation complexity. Each server just needs to read the queried symbol, hence our protocol incurs no extra cost. □

4.3 Explicit constructions

Theorem 1 shows that, if we want to optimize the practical parameters of our PIR scheme, we basically need to decrease ℓ, the number of groups. However, the dimension k of the code strongly depends on ℓ and n, and tiny values of ℓ can lead to trivial or very small codes. Thus, the rest of the section is devoted to the construction of transversal designs leading to codes and PIR protocols with good parameters.

4.3.1 From affine geometries. Transversal designs can be built through incidence properties of subspaces. Let $\mathbb{A}^m(\mathbb{F}_q)$ be the affine space of dimension m over $\mathbb{F}_q$, and $H = (H_1, \ldots, H_q)$ be q hyperplanes that partition $\mathbb{A}^m(\mathbb{F}_q)$. We define a transversal design $\mathcal{T}_A(m, q)$ as follows:

– the point set X consists in all the points in $\mathbb{A}^m(\mathbb{F}_q)$;
– the groups $\mathcal{G}$ are the so-called parallel class H;
– the blocks $\mathcal{B}$ are all the 1-dimensional affine subspaces (lines) which do not entirely lie in one of the H_i.

Such a design is a $\mathrm{TD}(q, q^{m-1})$ because a line is either contained in one of the H_i, or is 1-secant to each of them. To complete the study of the parameters, it remains to compute the dimension of $\mathrm{Code}(\mathcal{T}_A(m, q))$.

First notice that all blocks from $\mathcal{T}_A(m, q)$ are contained in the block set of the affine geometry design $\mathrm{AG}_1(m, q)$, the incidence structure of points and lines in $\mathbb{A}^m(\mathbb{F}_q)$. It implies that $\mathrm{Code}_p(\mathrm{AG}_1(m, q)) \subseteq \mathrm{Code}_p(\mathcal{T}_A(m, q))$ for any field $\mathbb{F}_p$. The benefit to consider $\mathrm{AG}_1(m, q)$ is that the p-rank of its incidence matrix has been well-studied [12, 1]. Besides, when p and q are coprimes, it has been proved that this design-based code has dimension 0 or 1 (Theorem

2.4.1 in [1]). Hence, to have better codes we will assume that p is the characteristic of the field $\mathbb{F}_q$.

In this setting ($q = p^e$), Hamada [12] gives a generic formula to compute the p-rank of a design coming from affine and projective geometries. Although asymptotics are hard to derive from Hamada's formula, we can compute specific values of some p-ranks, which give us lower bounds on the dimension of our transversal-design-based codes presented in Table 1.

m	$\ell = q$	$N = n\ell = q^m$	$\dim \mathcal{C}$ (lower bound)	$R = \dim \mathcal{C}/N$ (lower bound)
2	8	64	37	0.578
2	64	4096	3367	0.822
2	1024	1 048 576	989 527	0.944
2	4096	16 777 216	16 245 775	0.968
2	65 536	4 294 967 296	4 251 920 575	0.990
3	64	262 144	118 873	0.453
3	256	16 777 216	9 263 777	0.552
3	1024	1 073 741 824	680 200 873	0.633
3	8192	549 755 813 888	400 637 408 211	0.729
4	64	16 777 216	2 717 766	0.162
4	256	4 294 967 296	890 445 921	0.207
5	64	1 073 741 824	44 281 594	0.041

Table 1: Lower bounds on the dimension and the rate of codes arising from the transversal designs $\mathcal{T}_A(m, q)$ built on affine spaces — we recall that $\dim \mathrm{Code}_p(\mathcal{T}_A(m, p^e)) \geq p^{em} - \mathrm{rank}_p \mathrm{AG}_1(m, p^e)$. Here, q is a power of 2, and these codes can be defined over any extension of $\mathbb{F}_2$. Remind that in the PIR settings, R is related to the server storage overhead and $q = \ell$ is essentially the communication complexity and the number of servers.

For example, the two following PIR instances arise from our construction:

- choosing $m = 2$ and $\ell = 4096$, there exists a PIR protocol on a $\simeq 2.0$ MB database with only 6 kB of communication and 3.2% storage overhead;
- for a $\simeq 46$ GB file ($m = 3$, $\ell = 8192$), we obtain a PIR protocol with 27.1% storage overhead and 39kB of communication.

4.3.2 From projective geometries. Projective geometries are closely related to affine geometries, but contrary to them, there is no generic hyperplane-partition of the projective space (because every pair of hyperplanes intersects in a projective space of co-dimension 2). To tackle this problem, the idea is to consider the hyperplanes H_i which intersect on a fixed subspace of co-dimension 2 (call it Π_∞). Then, all the sets $H_i \setminus \Pi_\infty$ are disjoint, and their union gives exactly $\mathbb{P}^m(\mathbb{F}_q) \setminus \Pi_\infty$. Besides, any projective line disjoint from Π_∞ is either contained in one of the H_i, or is secant to all of them with multiplicity one. Hence we can define the following transversal design $\mathcal{T}_P(m, q)$:

- the point set $X = \mathbb{P}^m(\mathbb{F}_q) \setminus \Pi_\infty$;
- the group set $\mathcal{G} = \{\text{hyperplanes } H \subset \mathbb{P}^m(\mathbb{F}_q),\ \Pi_\infty \subset H\}$;
- the block set $\mathcal{B} = \{\text{projectives lines } L \subset \mathbb{P}^m(\mathbb{F}_q),\ L \cap \Pi_\infty = \varnothing \text{ and } \forall H \in \mathcal{G},\ L \not\subset H\}$.

This is a $\mathrm{TD}(q + 1, (q + 1)q^{m-1})$, and, as in the affine case, its p-rank can be bounded by that of $\mathrm{PG}_1(m, q)$, the design of point-line incidences in the projective space $\mathbb{P}^m(\mathbb{F}_q)$. Instead of presenting a table of parameters, we draw the results in Figure 4 and show that the parameters obtained in the affine and projective stettings are essentially the same.

4.3.3 From orthogonal arrays.

Definition 7 (Orthogonal array). *An orthogonal array $A = \mathrm{OA}_\lambda(t, \ell, s)$ is an array with λs^t rows of length ℓ with entries in a set S of size s, with the property that in any subarray of A formed by t columns, every row vector from S^t appears exactly λ times. We call λ the index of the orthogonal array, t its strength and ℓ its degree. If t (resp. λ) is omitted, it is*

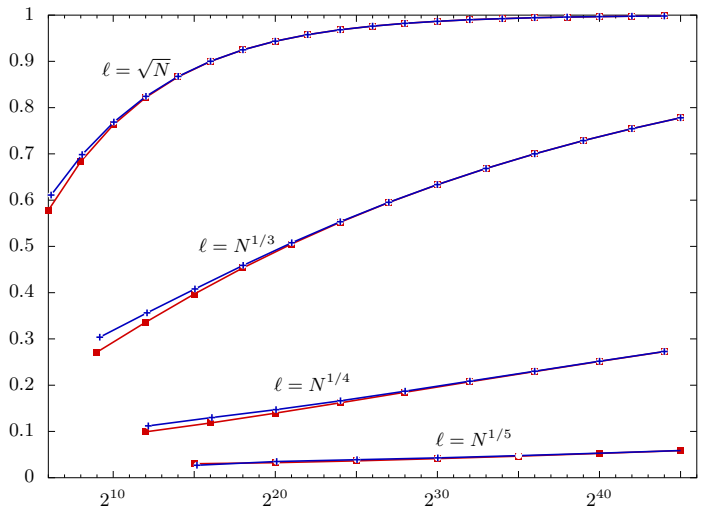

Fig. 4: Rate of codes presented in subsections 4.3.1 (red line) and 4.3.2 (blue line). The x-axis represents the length N in log scaling.

understood to be 2 *(resp.* 1*). When both these parameters are omitted we write* $A = \mathrm{OA}(\ell, s)$*. For convenience we also restrict the definition to orthogonal arrays with no repeated column and no repeated row.*

Construction of transversal designs from orthogonal arrays. We can build a $\mathrm{TD}(\ell, n)$ from an $\mathrm{OA}(\ell, n)$ with the following construction, given as a remark in [8, II.2]. Let A be an $\mathrm{OA}(\ell, n)$ with symbols in S, $|S| = n$, and denote by $\mathrm{Rows}(A)$ the n^2 rows of A. We define a point set $X = S \times [1, \ell]$, and a block set $\mathcal{B}$ as follows:

$$\mathcal{B} = \{\{(c_i, i), i \in [1, \ell]\}, c \in \mathrm{Rows}(A)\}\,.$$

Finally, let $\mathcal{G} = \{S \times \{i\}, i \in [1, \ell]\}$. Then $(X, \mathcal{B}, \mathcal{G})$ is a transversal design $\mathrm{TD}(\ell, n)$.

Remark 2. It is well-known that orthogonal arrays are closely related to codes (see again [8]). Listed in rows, all the codewords of a (possibly non-) linear code $\mathcal{C}$ give rise to an orthogonal array with strength $t = d' - 1$, where d' is the dual distance of $\mathcal{C}$.

Example 4. Let $\mathbf{x} = \{x_1, \ldots, x_\ell\}$ be a subset of $\mathbb{F}_q$ and denote by $\mathrm{RS}(\mathbf{x}, 2)$ the Reed-Solomon code of length ℓ and dimension 2 over $\mathbb{F}_q$ with evaluation points $\mathbf{x}$:

$$\mathrm{RS}(\mathbf{x}, 2) = \{(f(x_1), \ldots, f(x_\ell)),\ f \in \mathbb{F}_q[X],\ \deg f < 2\}\,.$$

Then, all the codewords of $\mathrm{RS}(\mathbf{x}, 2)$ form an orthogonal array $A = \mathrm{OA}(\ell, q)$. Now, use the previous construction to exhibit a transversal design $\mathrm{TD}(\ell, q)$. The point set is $X = \mathbb{F}_q \times [1, \ell]$, and the blocks are "labeled Reed-Solomon codewords", that is, sets of the form $\{(c_i, i),\ i \in [1, \ell]\}$ with $c \in \mathrm{RS}(\mathbf{x}, 2)$. The ℓ groups correspond to the ℓ coordinates of the code: $\mathbb{F}_q \times \{i\}$, $1 \leq i \leq \ell$.

As in previous constructions, for PIR application we can build a code based on a transversal design – itself constructed with an orthogonal array which in turn comes from a code $\mathcal{C}_0$. We call $\mathcal{C}_0$*-coded-queries code* such a stacked construction, which is summarized in Figure 5.

Orthogonal arrays from linear MDS codes of dimension 2**.** We recall that a $[n, k, d]$ linear code is said maximum distance separable (MDS) if it reaches the Singleton bound $n + 1 = k + d$. In this paragraph we analyse the coded-queries codes given by MDS codes of dimension 2. In order to simplify our study, we first prove some results.

Lemma 1. *All* $[\ell, 2, \ell - 1]$ *MDS codes over* $\mathbb{F}_q$ *with* $2 \leq \ell \leq q$ *are generalized Reed-Solomon codes (GRS codes).*

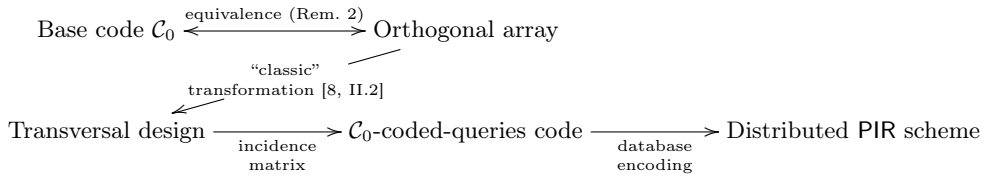

Fig. 5: A distributed PIR scheme using the coded-queries code construction.

Proof. First GRS codes are MDS. Now let $\mathcal{C}$ be an $[\ell, 2, \ell-1]_q$ code with $2 \leq \ell \leq q$. The weight distribution of an MDS code ensures there exists a codeword $c \in \mathcal{C}$ with Hamming weight ℓ. Let $u \in \mathcal{C}$ such that $\{c, u\}$ generates $\mathcal{C}$. We denote by $c * u$ the coordinate-wise product $(c_1u_1, \ldots, c_\ell u_\ell)$ and $\mathbf{1}$ the all-one vector of length ℓ. Then $c = \mathbf{1} * c$ and $u = c * (c^{-1} * u)$, where c^{-1} is the coordinate-wise inverse of c through $*$. Hence, the code $\mathcal{C}$ can be written $c * \mathcal{C}'$ where $\mathcal{C}'$ has $G' = \binom{\mathbf{1}}{c^{-1} * u}$ as generator matrix. It means that $\mathcal{C}$ is the GRS code with evaluation points $\mathbf{x} = c^{-1} * u$, multipliers $\mathbf{y} = c$ and dimension 2. □

A map $\phi : X \to X'$ is an isomorphism between transversal designs $(X, \mathcal{B}, \mathcal{G})$ and $(X', \mathcal{B}', \mathcal{G}')$ if it preverses their structure, that is, if ϕ is invertible on the points, on the blocks ($\phi(\mathcal{B}) = \mathcal{B}'$) and on the groups ($\phi(\mathcal{G}) = \mathcal{G}'$), and if ϕ preserves the incidence relations between them.

Lemma 2. *Let $\mathcal{C}, \mathcal{C}'$ be two codes such that $\mathcal{C}' = \mathbf{y} * \mathcal{C}$ for some $\mathbf{y} \in (\mathbb{F}_q^\times)^\ell$. Denote by $\mathrm{OA}_\mathcal{C}$ and $\mathrm{OA}_{\mathcal{C}'}$ their associated orthogonal arrays and by $\mathrm{TD}_\mathcal{C}$, $\mathrm{TD}_{\mathcal{C}'}$ the transversal designs defined through them. Then these two transversal designs are isomorphic.*

Proof. Write $\mathrm{TD}_\mathcal{C} = (X, \mathcal{B}, \mathcal{G})$ and $\mathrm{TD}_{\mathcal{C}'} = (X', \mathcal{B}', \mathcal{G}')$. By definition $X = X' = \mathbb{F}_q \times [1, n]$ and $\mathcal{G} = \mathcal{G}' = \{\mathbb{F}_q \times \{i\}, 1 \leq i \leq \ell\}$. Now focus on the block sets. We see that $\mathcal{B} = \{\{(c_i, i), 1 \leq i \leq \ell\}, c \in \mathcal{C}\}$ and $\mathcal{B}' = \{\{(y_i c_i, i), 1 \leq i \leq \ell\}, c \in \mathcal{C}\}$. Let:

$$\begin{array}{rcl} \phi_\mathbf{y} : \mathbb{F}_q \times [1, \ell] & \to & \mathbb{F}_q \times [1, \ell] \\ (x, i) & \mapsto & (y_i x, i) \end{array}$$

The vector $\mathbf{y}$ is $*$-invertible so the map $\phi_\mathbf{y}$ is one-to-one on the point set X. Now remark that $\phi_\mathbf{y}$ maps $\mathcal{G}$ to itself, and that $\phi_\mathbf{y}(\mathcal{B})$ is exactly $\mathcal{B}'$. Hence $\phi_\mathbf{y}(\mathrm{TD}_\mathcal{C}) = \mathrm{TD}_{\mathbf{y}*\mathcal{C}} = \mathrm{TD}_{\mathcal{C}'}$. □

Proposition 1. *Let $2 \leq \ell \leq q$ and $\mathcal{C}$ be an $[\ell, 2, \ell-1]_q$ linear (MDS) code. The $\mathcal{C}$-coded-queries code over $\mathbb{F}_p$ is permutation-equivalent to a $\mathrm{RS}(\mathbf{x}, 2)$-coded-queries code, with $\mathbf{x} \in \mathbb{F}_q^\ell$, $x_i \neq x_j$.*

Proof. Lemma 1 shows that all $[\ell, 2, \ell-1]_q$ linear codes $\mathcal{C}$ can be written $\mathbf{y} * \mathrm{RS}(\mathbf{x}, 2)$ for some $\mathbf{x} \in \mathbb{F}_q^\ell$. Moreover, with the previous notations $\phi_\mathbf{y}(\mathrm{TD}_{\mathrm{RS}(\mathbf{x},2)}) = \mathrm{TD}_{\mathbf{y}*\mathrm{RS}(\mathbf{x},2)}$, so we have $u \in \mathrm{Code}_p(\mathrm{TD}_{\mathbf{y}*\mathrm{RS}(\mathbf{x},2)})$ if and only if $u \in \mathrm{Code}_p(\phi_\mathbf{y}(\mathrm{TD}_{\mathrm{RS}(\mathbf{x},2)}))$. Now, let:

$$\begin{array}{rcl} \tilde{\phi}_\mathbf{y} : \mathbb{F}_p^X & \to & \mathbb{F}_p^X \\ u = (u_x)_{x \in X} & \mapsto & (u_{\phi_\mathbf{y}(x)})_{x \in X} \end{array}.$$

Clearly $\tilde{\phi}_\mathbf{y}(\mathrm{Code}_p(\mathrm{TD}_{\mathrm{RS}(\mathbf{x},2)})) = \mathrm{Code}_p(\phi_\mathbf{y}(\mathrm{TD}_{\mathrm{RS}(\mathbf{x},2)}))$ and $\tilde{\phi}_\mathbf{y}$ is a permutation of the coordinates of the code. So $\mathrm{Code}_p(\mathrm{TD}_\mathcal{C})$ is permutation-equivalent to $\mathrm{Code}_p(\mathrm{TD}_{\mathrm{RS}(\mathbf{x},2)})$ which proves the result. □

If we plan to search for MDS codes leading to high-rate coded-queries codes, then the previous proposition allows us to focus only on Reed-Solomon codes $\mathrm{RS}(\mathbf{x}, 2)$. It can even be proved, when $\mathcal{C}_0 = \mathrm{RS}(\mathbb{F}_q, 2)$, that the $\mathcal{C}_0$-coded-queries code is identical to the code presented in subsection 4.3.1.

On the other hand, considering vectors $\mathbf{x}$ of length $\ell < q$ is equivalent to shortening $\mathrm{Code}_p(\mathrm{TD}_{\mathcal{T}_A(2,q)})$ on the coordinates corresponding to some groups. Then, a further direction of research would be to analyse, for a given length $\ell < q$, which evaluation points $\mathbf{x} \subseteq \mathbb{F}_q^\ell$ lead to the largest coded-queries codes. For instance, an exhaustive search shows that, when considering the 4368 different supports $\mathbf{x} \in \mathbb{F}_{16}^5$, 48 of the codes $\mathrm{Code}_{16}(\mathrm{TD}_{\mathrm{RS}(\mathbf{x},2)})$ have dimension 24 while the 4320 others have dimension 22.

5 PIR with better privacy

When servers are colluding, the PIR protocol based on a simple transversal design does not give a sufficient privacy, because the knowledge of two points on a block gives some information on it. To solve this issue, we propose to use orthogonal arrays with higher strength t.

5.1 Generic construction and analysis

In the previous section, classical ($t = 2$) orthogonal arrays were used to build transversal design. Considering higher values of t, we naturally generalize the latter as follows:

Definition 8 (t-transversal designs). *Let $t \geq 1$. A t-transversal design is a block design $\mathcal{D} = (X, \mathcal{B})$ equipped with a group set $\mathcal{G} = \{G_i\}_{1 \leq i \leq \ell}$ partitionning X such that:*

- *$|X| = n\ell$;*
- *any group has size n and any block has size ℓ;*
- *for any $T \subseteq [1, \ell]$ with $|T| = t$ and for any $(x_1, \ldots, x_t) \in G_{T_1} \times \ldots \times G_{T_t}$, there exist exacly λ blocks $B \in \mathcal{B}$ such that $\{x_1, \ldots, x_t\} \subset B$.*

A t-transversal design with parameters n, ℓ, t, λ is denoted t-TD$_\lambda(\ell, n)$, or t-TD(ℓ, n) when $\lambda = 1$.

Given a t-transversal design, we can build a $(t-1)$-private PIR protocol with exactly the same steps as in section 4: build the code $\mathcal{C}$ associated to the design and follow the algorithm given in Figure 2. As a t-transversal design is also a 2-transversal design for $t \geq 2$, the analysis remains identical for every feature, except for the security where it is very similar.

Security ($(t-1)$-privacy). Let T be a collusion of servers of size $|T| \leq t-1$. For varying i, the distributions $\mathcal{Q}(i)_{|T}$ are the same because there are exactly $\lambda n^{t-1-|T|} \geq 1$ blocks which contain both i and the queries known by the servers in T.

To sum up, the following theorem holds:

Theorem 2. *Let D be a database with k entries over a field $\mathbb{F}_q$, and $\mathcal{T} = t$-TD(ℓ, n) be a t-transversal design, whose incidence matrix has rank $\ell n - k$ over $\mathbb{F}_q$. Then, there exists an ℓ-server $(t-1)$-private PIR protocol with:*

- *only 1 symbol to read for each server,*
- *$\ell - 1$ field operations for the user,*
- *$\ell \log(nq)$ bits of communication,*
- *a storage overhead of $(\ell n - k) \log q$ bits on the servers.*

5.2 Instances and results

t-transversal designs from orthogonal arrays of strength t. Let A be an orthogonal array OA$_\lambda(t, \ell, s)$ on a symbol set S. We define the following design:

- points $X = S \times [1, \ell]$;
- groups $\mathcal{G} = \{S \times \{i\}, 1 \leq i \leq \ell\}$;
- blocks $\mathcal{B} = \{\{(a_{i,j}, i), 1 \leq i \leq \ell\}, 1 \leq j \leq \lambda s^t\}$.

Proposition 2. *This design is a t-TD$_\lambda(\ell, s)$.*

Proof. It is clear that $\mathcal{G}$ is a partition of X and that the blocks and groups have the right size. Now focus on the incidence property. Let $T \subset [1, \ell]$ with $|T| = t$, and let $(x_1, \ldots, x_t) \in G_{T_1} \times \ldots \times G_{T_t}$. We need to prove that there are exactly λ blocks $B \in \mathcal{B}$ such that $\{x_1, \ldots, x_t\} \in B$. Consider the map from blocks in $\mathcal{B}$ to rows of A given by:

$$\begin{array}{rccc} \psi : & \mathcal{B} & \to & \text{Rows}(A) \\ & B_j = \{(a_{i,j}, i), 1 \leq i \leq \ell\} & \mapsto & (a_{1,j}, \ldots, a_{\ell,j}) \end{array}$$

As we assumed that orthogonal arrays have no repeated rows, the map ψ is one-to-one. Denote by $x' = (x'_1, , \ldots, x'_t) \in S^t$ the vector formed by the first coordinates of $(x_1, , \ldots, x_t) \in X^t$. From the definition of an orthogonal array of strength t and index λ, we know that x' appears exactly λ times in the submatrix of A defined by the columns indexed by T. These λ appearances have λ preimages in $\mathcal{B}$ which proves the result. □

MDS codes of dimension t have dual distance $t+1$, so they correspond to orthogonal arrays of strength t through which we can build a t-transversal design. In Figure 6 we present how the rate of such $\mathcal{C}_0$-coded-queries codes varies according to the value of t.

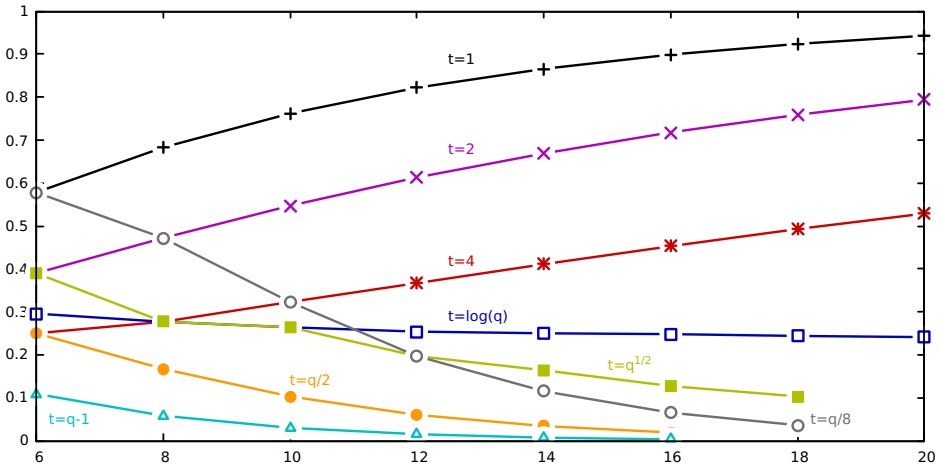

Fig. 6: Rate of $\mathcal{C}_0$-coded-queries codes used for t-private PIR, with various parameters of $\mathcal{C}_0$. The code $\mathcal{C}_0$ is a full-length Reed-Solomon code of dimension $t+1$ (dual distance $t+2$) over $\mathbb{F}_q$. The PIR protocol then needs q servers. The x-axis represents $\log_2 N$ where $N = q^2$ is the code length.

6 Conclusion

We showed that codes from transversal designs give rise to the construction of distributed PIR protocols with low burden on the server side. We point out that our scheme is computationnally optimal for the servers, in a sense that they just have to read one symbol of the word they hold. Moreover, the recovering step is also easy for the user which just needs to compute a linear combination of the symbols it receives.

The genericity of our construction naturally leads to the question of finding transversal designs with the most practical PIR parameters. Indeed, while affine and projective geometries give excellent PIR parameters for the servers (taken individually), their moderate communication and relatively huge number of servers leaves room for improvements and future research.

Acknowledgement

This work is partially funded by French ANR-15-CE39-0013-01 “Manta”. The author would like to thank Daniel Augot and Françoise Levy-dit-Vehel for their very helpful comments and corrections on the paper.

References

1. Edward F. Assmus and Jennifer D. Key. *Designs and Their Codes*. Cambridge University Press, 1992.

2. Daniel Augot, Françoise Levy-dit-Vehel, and Abdullatif Shikfa. A storage-efficient and robust private information retrieval scheme allowing few servers. In Dimitris Gritzalis, Aggelos Kiayias, and Ioannis G. Askoxylakis, editors, *Cryptology and Network Security - 13th International Conference, CANS 2014, Heraklion, Crete, Greece, October 22-24, 2014. Proceedings*, volume 8813 of *Lecture Notes in Computer Science*, pages 222–239. Springer, 2014.
3. Amos Beimel, Yuval Ishai, Eyal Kushilevitz, and Jean-François Raymond. Breaking the $o(n^{1/(2k-1)})$ barrier for information-theoretic private information retrieval. In *43rd Symposium on Foundations of Computer Science (FOCS 2002), 16-19 November 2002, Vancouver, BC, Canada, Proceedings*, pages 261–270. IEEE Computer Society, 2002.
4. Amos Beimel, Yuval Ishai, and Tal Malkin. Reducing the servers' computation in private information retrieval: PIR with preprocessing. *J. Cryptology*, 17(2):125–151, 2004.
5. Benny Chor and Niv Gilboa. Computationally private information retrieval. In Frank Thomson Leighton and Peter W. Shor, editors, *Proceedings of the Twenty-Ninth Annual ACM Symposium on the Theory of Computing, El Paso, Texas, USA, May 4-6, 1997*, pages 304–313. ACM, 1997.
6. Benny Chor, Oded Goldreich, Eyal Kushilevitz, and Madhu Sudan. Private information retrieval. In *36th Annual Symposium on Foundations of Computer Science, Milwaukee, Wisconsin, 23-25 October 1995*, pages 41–50. IEEE Computer Society, 1995.
7. Benny Chor, Eyal Kushilevitz, Oded Goldreich, and Madhu Sudan. Private information retrieval. *J. ACM*, 45(6):965–981, 1998.
8. Charles J. Colbourn and Jeffrey H. Dinitz. *Handbook of Combinatorial Designs, Second Edition*. Chapman & Hall/CRC, 2006.
9. Zeev Dvir and Sivakanth Gopi. 2-server PIR with subpolynomial communication. *J. ACM*, 63(4):39:1–39:15, 2016.
10. Klim Efremenko. 3-query locally decodable codes of subexponential length. *SIAM J. Comput.*, 41(6):1694–1703, 2012.
11. Arman Fazeli, Alexander Vardy, and Eitan Yaakobi. Codes for distributed PIR with low storage overhead. In *IEEE International Symposium on Information Theory, ISIT 2015, Hong Kong, China, June 14-19, 2015*, pages 2852–2856. IEEE, 2015.
12. Noboru Hamada. The rank of the incidence matrix of points and d-flats in finite geometries. *Journal of Science of the Hiroshima University, Series A-I (Mathematics)*, 32(2):381–396, 1968.
13. Jonathan Katz and Luca Trevisan. On the efficiency of local decoding procedures for error-correcting codes. In F. Frances Yao and Eugene M. Luks, editors, *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing, May 21-23, 2000, Portland, OR, USA*, pages 80–86. ACM, 2000.
14. Swastik Kopparty, Shubhangi Saraf, and Sergey Yekhanin. High-rate codes with sublinear-time decoding. *J. ACM*, 61(5):28:1–28:20, 2014.
15. Douglas R. Stinson. *Combinatorial Designs - Constructions and Analysis*. Springer, 2004.
16. Sergey Yekhanin. Towards 3-query locally decodable codes of subexponential length. *J. ACM*, 55(1):1:1–1:16, 2008.
17. Sergey Yekhanin. Locally decodable codes. *Foundations and Trends in Theoretical Computer Science*, 6(3):139–255, 2012.