THEORY OF COMPUTING, Volume 3 (2007), pp. 221–238
http://theoryofcomputing.org

An $\Omega(n^{1/3})$ Lower Bound for Bilinear Group-Based Private Information Retrieval*

Alexander A. Razborov† Sergey Yekhanin‡

Received: May 14, 2007; revised: December 19, 2007; published: December 28, 2007.

Abstract: A two-server *private information retrieval* (PIR) scheme allows a user $\mathcal{U}$ to retrieve the i-th bit of an n-bit string x replicated on two servers while each server individually learns no information about i. The main parameter of interest in a PIR scheme is its communication complexity: the number of bits exchanged by the user and the servers. Substantial effort has been invested by researchers over the last decade in the search for efficient PIR schemes. A number of different schemes (Chor et al., 1998, Beimel et al., 2005, Woodruff and Yekhanin, CCC'05) have been proposed; however, all of them result in the same communication complexity of $O(n^{1/3})$. The best known lower bound to date is $5\log n$ by Wehner and de Wolf (ICALP'05). The tremendous gap between upper and lower bounds is the focus of our paper. We show an $\Omega(n^{1/3})$ lower bound in a restricted model that nevertheless captures all known upper bound techniques.

*A preliminary version of this paper appeared in the proceedings of the 47th IEEE Symposium on Foundations of Computer Science (FOCS'06) [15].

†Supported by the Charles Simonyi Endowment and NSF grant ITR-0324906.

‡Supported by NSF grant CCR 0219218.

ACM Classification: H.3.3, F.1.3.e, F.1.2.b

AMS Classification: 68P20, 68Q17, 20C20

Key words and phrases: lower bounds, private information retrieval, secret sharing, communication complexity, group representations, bilinear schemes

DOI: 10.4086/toc.2007.v003a012

Our lower bound applies to bilinear group-based PIR schemes. A bilinear PIR scheme is a one-round PIR scheme where the user computes the dot product of the servers' responses to obtain the desired value of the i-th bit. Every linear scheme can be turned into a bilinear one with an asymptotically negligible communication overhead. A group-based PIR scheme is a PIR scheme in which the servers represent the database by a function on a certain finite group G and the user retrieves the value of this function at any group element using the natural secret sharing scheme based on G. Our proof relies on the representation theory of finite groups.

1 Introduction

Private information retrieval (PIR) was introduced in a seminal paper by Chor, Goldreich, Kushilevitz, and Sudan [5]. In such a scheme a server holds an n-bit string x representing a database, and a user holds an index $i \in [n]$. At the end of the protocol the user should learn x_i and the server should learn nothing about i. A trivial PIR protocol is to send the whole database x to the user. While this protocol is perfectly private, its communication complexity is prohibitively large. (Note for comparison that in the non-private setting there is a protocol with only $\log n + 1$ bits of communication.) This raises the question of how much communication is necessary to achieve privacy. It has been shown in [5] that when information-theoretic privacy is required the above trivial solution is in fact optimal. To get around this Chor et al. suggested replicating the database among $k > 1$ non-communicating servers.

For the case of two servers Chor et al. [5] obtained a PIR protocol with $O(n^{1/3})$ communication complexity. In spite of the large amount of subsequent research, this bound remains the best known to date. In contrast to two-server PIR schemes, PIR schemes involving three or more servers have undergone several steps of improvement. The initial three-server PIR scheme of Chor et al. [5] had communication complexity $O(n^{1/3})$. Later Ambainis [1] suggested a scheme with $O(n^{1/5})$ communication, and Beimel et al. [4] further reduced the communication to $O(n^{1/5.25})$. Finally, in a recent paper Yekhanin [21] achieved $O(n^{1/32,582,658})$ communication, and showed that communication can be further reduced to $n^{o(1)}$ under a plausible number-theoretic assumption regarding the density of Mersenne primes (see also [12]). The best known (unconditional) upper bound for communication complexity of k-server PIR when k is considered as a parameter can be obtained by a combination of results from [4] and [21] and is $n^{O\left(\frac{\log\log k}{k \log k}\right)}$.

On the lower bounds side the progress has been scarce. We list the known results for the two-server case. The first nontrivial lower bound of $4 \log n$ is due to Mann [14]. Later it was improved to $4.4 \log n$ by Kerenidis and de Wolf [13] using the results of Katz and Trevisan [11]. The current record of $5 \log n$ is due to Wehner and de Wolf [17]. The proofs of the last two bounds use quantum arguments.

The PIR literature existing today is extensive. There are a number of generalizations of the basic PIR setup that have been studied. Most notably those are: computational PIR (i. e., PIR based on computational assumptions), PIR with privacy against coalitions of servers, PIR with fixed answer sizes, robust PIR, etc. Private information retrieval schemes are also closely related to locally decodable codes (LDC). For a survey of PIR and LDC literature see [6, 20].

In the present paper we study the communication complexity of PIR in the most basic, two-server case. There are two reasons why this case in especially attractive. Firstly, determining the communication complexity of optimal two-server PIR schemes is arguably the most challenging problem in the area of PIR research. There has been no quantitative progress for this case since the problem was posed. Although to date a number of different two-server PIR schemes are known [5, 3, 19] all of them have the same communication complexity of $O(n^{1/3})$. Secondly, the work of [4] implies that a large improvement in the upper bound for two-server PIR would yield better PIR protocols for all other values of k.

1.1 Our results

Our main result is an $\Omega(n^{1/3})$ lower bound for a restricted model of two-server PIR. Our restrictions revolve around a novel, though quite natural, combinatorial view of the problem. We show that two-server PIR is essentially a problem regarding the minimal size of an *induced universal graph* for a family of graphs with a certain property.[1] This view allows us to identify two natural models of PIR, namely, *bilinear* PIR, and *bilinear group-based* PIR. A bilinear PIR scheme is a one-round PIR scheme where the user computes the dot product of the servers' responses to obtain the desired value of the i-th bit. A *group-based* PIR scheme is a PIR scheme that involves the servers representing the database by a function on a certain finite group G, and allows the user to retrieve the value of this function at any group element using the natural secret sharing scheme based on G.

We establish an $\Omega(n^{1/3})$ lower bound for communication complexity of any bilinear group-based PIR scheme, that holds regardless of the underlying group G and regardless of the algorithms run by the servers. The model of bilinear group-based PIR generalizes all PIR protocols known to date, thus our lower bound demonstrates a common shortcoming of the existing upper bound techniques. It also helps to explain why the (hitherto somewhat arbitrary looking) numerical value $O(n^{1/3})$ in fact represents quite a natural barrier for techniques of this sort.

It turns out that the communication complexity of bilinear group-based PIR schemes over a group G can be estimated in terms of the number of low-dimensional principal left ideals in the group algebra $\mathbb{F}_q[G]$. Our main technical result is an upper bound for this quantity obtained by an argument relying on the representation theory of finite groups.

1.2 Related work

Apart from the work on general lower bounds for PIR protocols surveyed above, there has been some effort to establish (stronger) lower bounds for various restricted models of PIR [10, 7, 17]. In particular, Itoh [10] obtained polynomial lower bounds on the communication complexity of one-round PIR schemes under the assumption that each server returns a multilinear or affine function of its input. Goldreich et al. [7] introduced the notion of *linear* PIR protocols, i. e., protocols where the servers are restricted to return linear combinations of the database bits to the user, and also the notion of *probe complexity*, i. e., the maximal number of bits the user needs to read from the servers' answers in order to compute x_i. Goldreich et al. obtained polynomial lower bounds for the communication complexity of two-server linear PIR schemes whose probe complexity is constant. Later, their results were extended

[1] We actually prefer to use language of matrices rather than graphs, but of course graph formulations are easy to obtain. A graph G is called induced universal for a graph family $\mathcal{F}$ if every graph $F \in \mathcal{F}$ is an induced subgraph of G.

by Wehner and de Wolf [17] who showed that the restriction of linearity can in fact be dropped. See also [2].

It is not easy to match the restricted models surveyed above against one another or against our model, as the restrictions are quite different. We do not impose any restriction on the functions computed by the servers as in [10], and do not restrict the user to read only a small number of bits from servers' answers as in [7]. We show that our bilinearity restriction is weaker than the linearity restriction of [7], since every linear protocol can be easily turned into a bilinear one. However, we insist that the PIR scheme should employ group-based secret sharing, and that the user should be able to privately reconstruct not only the database bits but also some extra functions of the database (given by the values at group elements that do not correspond to database bits).

1.3 Outline

In Section 2 we introduce our notation and provide some necessary definitions. In Section 3 we present our combinatorial interpretation of two-server PIR, and identify the models of bilinear PIR and bilinear group-based PIR. Section 4 contains the main technical contribution of the paper. We introduce the necessary algebraic tools and establish an $\Omega(n^{1/3})$ lower bound for communication complexity of any bilinear group-based PIR scheme. In Section 5 we discuss possible interpretations of our results and pose an open problem. In the appendix we review currently known two-server PIR schemes and demonstrate that all of them are bilinear group-based.

2 Preliminaries

Let $[s] \stackrel{\text{def}}{=} \{1,\ldots,s\}$. We assume that q is a prime power and use the notation $\mathbb{F}_q$ to denote a finite field of q elements. We assume that the database contains entries from the alphabet $[q]$, rather than just a binary alphabet. We also assume some implicit bijection between $[q]$ and $\mathbb{F}_q$. Throughout log stands for the log base q. The notation $a \circ b$ stands for concatenation of strings a and b.

A two-server PIR scheme involves two servers, $\mathcal{S}_1$ and $\mathcal{S}_2$, each holding the same n-bit string x (the database), and a user $\mathcal{U}$ who knows n and wishes to retrieve some bit x_i, $i \in [n]$, without revealing the value of i. We restrict our attention to one-round information-theoretic PIR protocols. The following definition is a non-uniform variant of the definition from [4].

Definition 2.1. A two-server PIR protocol is a triplet of non-uniform algorithms $\mathcal{P} = (\mathcal{Q},\mathcal{A},\mathcal{C})$. We assume that each algorithm is given n as advice. At the beginning of the protocol, the user $\mathcal{U}$ tosses random coins and obtains a random string r. Next, $\mathcal{U}$ invokes $\mathcal{Q}(i,r)$ to generate a pair of queries $(\mathrm{que}_1,\mathrm{que}_2)$. $\mathcal{U}$ sends que_1 to $\mathcal{S}_1$ and que_2 to $\mathcal{S}_2$. Each server $\mathcal{S}_j$ responds with an answer $\mathrm{ans}_j = \mathcal{A}(j,x,\mathrm{que}_j)$. (We assume without loss of generality that servers are deterministic.) Finally, $\mathcal{U}$ computes its output by applying the reconstruction algorithm $\mathcal{C}(\mathrm{ans}_1,\mathrm{ans}_2,i,r)$. A protocol as above should satisfy the following requirements:

- **Correctness :** For any n, $x \in [q]^n$, and $i \in [n]$, the user outputs the correct value of x_i with probability 1 (where the probability is over the random strings r).

- **Privacy :** Each server individually learns no information about i. More precisely, we require that for any n and for any $j = 1,2$, the distributions $\mathrm{que}_j(i,r)$ are identical for all values $i \in [n]$.

The *communication complexity* of a PIR protocol $\mathcal{P}$ is the function of n measuring the total number of bits communicated between the user and the servers, maximized over all choices of $x \in [q]^n$, $i \in [n]$, and random inputs.

Definition 2.2. [7] A *linear* PIR scheme is a PIR scheme where the answer function $\mathcal{A}(j,x,\mathrm{que}_j)$ is linear in x for arbitrary fixed values of j and que_j. In other words, every bit of an answer is a certain linear combination of the database bits.

3 A combinatorial view of two-server PIR

Definition 3.1. A generalized Latin square ($\mathrm{GLS}[n,T]$ for short) is a square matrix Q of size T by T over an alphabet $[n] \cup \{*\}$ such that:

- For every $i \in [n]$ and $j \in [T]$, there exists a unique $k \in [T]$ such that $Q_{jk} = i$;

- For every $i \in [n]$ and $j \in [T]$, there exists a unique $k \in [T]$ such that $Q_{kj} = i$.

In particular, every row (or column) of a $\mathrm{GLS}[n,T]$ contains precisely $(T-n)$ stars. We call the ratio n/T the *density* of the generalized Latin square. It is easy to see that generalized Latin squares of density 1 are simply Latin squares.

Let Q be a $\mathrm{GLS}[n,T]$, and let $\sigma : [n] \to [q]$ be an arbitrary map. By Q_σ we denote the matrix of size T by T over the alphabet $[q] \cup \{*\}$ that is obtained from Q by replacing every non-star entry i in Q by $\sigma(i)$. We say that a matrix $C \in [q]^{T\times T}$ is a *completion* of Q_σ if $C_{ij} = (Q_\sigma)_{ij}$ whenever $(Q_\sigma)_{ij} \in [q]$.

For matrices $C \in [q]^{c\times c}$ and $A \in [q]^{\ell\times\ell}$ we say that C *reduces* to A if there exist two maps $\pi_1 : [c] \to [\ell]$ and $\pi_2 : [c] \to [\ell]$ such that for any $j,k \in [c]$, $C_{jk} = A_{\pi_1(j),\pi_2(k)}$. Note that we do not impose any restrictions on the maps π_1 and π_2; in particular c can be larger than ℓ.

Definition 3.2. Let Q be a $\mathrm{GLS}[n,T]$ and $A \in [q]^{\ell\times\ell}$. We say that A *covers* Q (notation $Q \hookrightarrow A$) if, for every $\sigma : [n] \to [q]$, there exists a completion C of Q_σ such that C reduces to A.

Theorem 3.3. *The following two implications are valid:*

- *A pair $Q \hookrightarrow A$, where Q is a $\mathrm{GLS}[n,T]$ and $A \in [q]^{\ell\times\ell}$, yields a two-server PIR protocol with communication $\log T$ from $\mathcal{U}$ to each $\mathcal{S}_j$ and communication $\log \ell$ from the $\mathcal{S}_j$'s back to $\mathcal{U}$.*

- *A two-server PIR protocol with queries of length $t(n)$ and answers of length $a(n)$, where the user tosses at most $\tau(n)$ random coins, yields a pair $Q \hookrightarrow A$, where Q is a GLS $\left[n, nq^{t(n)+\tau(n)}\right]$ and A is a q-ary square matrix of size $nq^{t(n)+a(n)}$.*

Proof. We start with the first part. We assume that the matrix A is known to all parties $\mathcal{U}$, $\mathcal{S}_1$, and $\mathcal{S}_2$. At the preprocessing stage, the servers use the database $x \in [q]^n$ to define the map $\sigma : [n] \to [q]$, setting

$\sigma(i) \stackrel{\text{def}}{=} x_i$. Also, they find an appropriate completion C, and fix maps $\pi_1 : [T] \to [\ell]$ and $\pi_2 : [T] \to [\ell]$, such that for all j,k: $C_{jk} = A_{\pi_1(j),\pi_2(k)}$. Next, the following protocol is executed.

$\mathcal{U}$	:	Picks a location j,k in Q such that $Q_{jk} = i$ uniformly at random.
$\mathcal{U} \to \mathcal{S}_1$	:	j
$\mathcal{U} \to \mathcal{S}_2$	:	k
$\mathcal{U} \leftarrow \mathcal{S}_1$	:	$\pi_1(j)$
$\mathcal{U} \leftarrow \mathcal{S}_2$	:	$\pi_2(k)$
$\mathcal{U}$	:	Outputs $A_{\pi_1(j),\pi_2(k)}$.

It is straightforward to verify that the protocol above is private, since a uniformly random choice of a location j,k such that $Q_{jk} = i$ induces uniformly random individual distributions on j and k. Correctness follows from the fact that C reduces to A. Total communication is given by $2(\log T + \log \ell)$.

Now we proceed to the second part. Consider a two-server protocol $\mathcal{P} = (\mathcal{Q}, \mathcal{A}, \mathcal{C})$. First we show that one can modify $\mathcal{P}$ to obtain a new PIR protocol $\mathcal{P}' = (\mathcal{Q}', \mathcal{A}', \mathcal{C}')$ such that $\mathcal{C}'$ depends only on ans'_1 and ans'_2, but not on i or r. The transformation is simple:

- First $\mathcal{Q}'$ obtains a random string r and invokes $\mathcal{Q}(i,r)$ to generate $(\mathrm{que}_1, \mathrm{que}_2)$. Next $\mathcal{Q}'$ tosses $\log n$ extra random coins to represent i as a random sum $i = i_1 + i_2 \mod n$, sets $\mathrm{que}'_1 = \mathrm{que}_1 \circ i_1$, $\mathrm{que}'_2 = \mathrm{que}_2 \circ i_2$, and sends que'_1 to $\mathcal{S}_1$ and que'_2 to $\mathcal{S}_2$.

- For $j = 1,2$, $\mathcal{A}'$ extracts que_j from que'_j, runs $\mathcal{A}$ on (j, x, que_j), and returns $\mathrm{ans}_j \circ \mathrm{que}'_j$.

- Finally, $\mathcal{C}'$ extracts $\mathrm{que}_1, \mathrm{que}_2, \mathrm{ans}_1, \mathrm{ans}_2$, and i from ans'_1 and ans'_2, and performs a brute force search over all possible random coin tosses of $\mathcal{Q}$ to find some random input r' such that $\mathcal{Q}(i,r') = (\mathrm{que}_1, \mathrm{que}_2)$. $\mathcal{C}'$ runs $\mathcal{C}$ on $(\mathrm{ans}_1, \mathrm{ans}_2, i, r')$ and returns the answer (if no such r' exists, $\mathcal{C}'$ answers arbitrarily). Note that the string r' may in fact be different from the string r; however the correctness property of $\mathcal{P}$ implies that even in this case $\mathcal{C}'$ outputs the right value.

Now consider the protocol $\mathcal{P}'$. Let Q'_j denote the range of queries to server j, and A'_j denote the range of answers from server j. The variable que'_j ranges over Q'_j, and the variable ans'_j ranges over A'_j. Let $R(\mathrm{que}'_j, i)$ denote the set of random strings r that lead to the query que'_j to server j on input i. Formally,

$$R(\mathrm{que}'_1, i) = \left\{ r \in [q]^{\tau(n)} \mid \exists\, \mathrm{que}'_2 : Q(i,r) = (\mathrm{que}'_1, \mathrm{que}'_2) \right\},$$
$$R(\mathrm{que}'_2, i) = \left\{ r \in [q]^{\tau(n)} \mid \exists\, \mathrm{que}'_1 : Q(i,r) = (\mathrm{que}'_1, \mathrm{que}'_2) \right\}.$$

Note that the privacy property of the protocol $\mathcal{P}'$ means that the cardinalities of $R(\mathrm{que}'_j, i)$ are independent of i. We denote these cardinalities by $r(\mathrm{que}'_j)$. It is easy to see that $r(\mathrm{que}'_j)$ is always an integer between 1 and $q^{\tau(n)}$. Now we are ready to define the matrices Q and A.

Rows of Q are labelled by pairs (que'_1, s_1), where $s_1 \in [r(\mathrm{que}'_1)]$. Columns of Q are labelled by pairs (que'_2, s_2), where $s_2 \in [r(\mathrm{que}'_2)]$. We set $Q_{(\mathrm{que}'_1, s_1),(\mathrm{que}'_2, s_2)} = i$ if there exists a string $r \in R(\mathrm{que}'_1, i) \cap R(\mathrm{que}'_2, i)$ such that r is the string number s_1 in $R(\mathrm{que}'_1, i)$ and the string number s_2 in $R(\mathrm{que}'_2, i)$ with respect to lexicographic ordering of these sets; otherwise we set $Q_{(\mathrm{que}'_1,s_1),(\mathrm{que}'_2,s_2)} = *$. The definition of the protocol $\mathcal{P}'$ easily implies that for every pair $(\mathrm{que}'_1, \mathrm{que}'_2)$ there can be at most one i such that $R(\mathrm{que}'_1, i) \cap R(\mathrm{que}'_2, i) \neq \emptyset$, therefore Q is correctly defined.

Consider now an arbitrary pair $(i, (\mathrm{que}'_1, s_1))$, where $s_1 \in [r(\mathrm{que}'_1)]$. Let r be the random string number s_1 in lexicographic ordering of $R(\mathrm{que}'_1, i)$. Let $\mathcal{Q}'(i,r) = (\mathrm{que}'_1, \mathrm{que}'_2)$, and let s_2 be the number of r in lexicographic ordering of $R(\mathrm{que}'_2, i)$. The column of Q labelled (que'_2, s_2) is the unique column such that $Q_{(\mathrm{que}'_1,s_1),(\mathrm{que}'_2,s_2)} = i$. Thus we proved that every row of Q contains exactly one entry labelled i. A similar argument proves this claim for columns. Thus Q is a generalized Latin square.

Now we proceed to the matrix A. Rows of A are labelled by possible values of ans'_1, similarly columns of A are labelled by possible values of ans'_2. We set $A_{\mathrm{ans}'_1,\mathrm{ans}'_2} = \mathcal{C}'(\mathrm{ans}'_1, \mathrm{ans}'_2)$. The matrix A defined above may not be square, however one can always pad it to a square shape.

It remains to show that $Q \hookrightarrow A$. Given a map $\sigma : [n] \to [q]$ we consider the database x, where $x_i = \sigma(i)$. We use the protocol $\mathcal{P}'$ to define maps π_1 from the row set of Q to the row set of A, and π_2 from the column set of Q to the column set of A. We set $\pi_1(\mathrm{que}'_1, s_1) = \mathcal{A}'(1, x, \mathrm{que}'_1)$ and $\pi_2(\mathrm{que}'_2, s_2) = \mathcal{A}'(2, x, \mathrm{que}'_2)$. The correctness property of $\mathcal{P}'$ implies that the maps π_1, π_2 reduce a certain completion of Q_σ to A. □

The theorem above represents our combinatorial view of two-server PIR protocols. A PIR protocol is just a pair $Q \hookrightarrow A$, where Q is a generalized Latin square and A is a q-ary matrix. Every PIR protocol can be converted into this form, and in case the number of user's coin tosses is linear in the query length such conversion does not affect the asymptotic communication complexity.

3.1 Bilinear PIR

The combinatorial interpretation of PIR suggested above views PIR as a problem of reducing certain special families of matrices to some fixed matrix. A nice example of a nontrivial matrix where one can say a lot about matrices that reduce to it is a Hadamard matrix.

Definition 3.4. A Hadamard matrix H_m is a q^m by q^m matrix where rows and columns are labelled by elements of $\mathbb{F}_q^m$ and matrix cells contain the dot products of corresponding labels. I.e. $(H_m)_{v_1,v_2} = (v_1, v_2)$.

Lemma 3.5. *Let M be a square matrix with entries from $\mathbb{F}_q$; then M reduces to a Hadamard matrix H_m if and only if the rank of M is at most m.*

Proof. Clearly, the rank of H_m is m; therefore the rank of any matrix that reduces to H_m is at most that large. To prove the converse, observe that M can be written as a sum of m matrices $M = M^1 + \ldots + M^m$, where each M^j is of rank at most one. Let t be the dimension of M. For every $i \in [m]$ set the i-th coordinate of m long vectors $v^1, \ldots, v^t \;\; u^1, \ldots, u^t$ so that $v_i^j u_i^k = M_{jk}^i$. Now the maps $\pi_1 : [t] \to [q^m]$, $\pi_2 : [t] \to [q^m]$ defined by $\pi_1(j) = v^j$, $\pi_2(k) = u^k$ embed M into H_m. □

The above lemma is important since it allows us to reduce the proof that $Q \hookrightarrow H_m$ for some generalized Latin square Q to showing that for every $\sigma : [n] \to \mathbb{F}_q$, Q_σ can be completed to a low rank matrix.

Definition 3.6. We say that a two-server PIR scheme $Q \hookrightarrow A$ is *bilinear* if $A = H_m$ for some value of m.

Another way to formulate the above definition is to say that a PIR scheme is bilinear if $\mathcal{U}$ computes the dot product of the servers' answers to obtain the value of x_i. The next lemma shows that the restriction of bilinearity is weaker than that of linearity.

Lemma 3.7. *Every linear PIR protocol can be turned into a bilinear PIR protocol with the same asymptotic communication complexity.*

Proof. In a linear PIR protocol the user receives two strings $\mathrm{ans}_1, \mathrm{ans}_2$ of linear combinations of database bits from the servers. The n-dimensional unit vector corresponding to the i-th bit of the database is guaranteed to be in the joint span of the combinations from ans_1 and ans_2. The final output of $\mathcal{U}$ is a sum of two dot products $(c_1, \mathrm{ans}_1) + (c_2, \mathrm{ans}_2) = x_i$, for some vectors c_1 and c_2 that are computed by the user along with queries $(\mathrm{que}_1, \mathrm{que}_2)$. The idea behind turning a linear protocol into a bilinear one is simple.

After generating $(\mathrm{que}_1, \mathrm{que}_2)$ along with c_1 and c_2, $\mathcal{U}$ represents c_1 and c_2 as sums of random strings $c_1 = c_{11} + c_{12}$, $c_2 = c_{21} + c_{22}$, and sends $\mathrm{que}_1 \circ c_{11} \circ c_{21}$ to $\mathcal{S}_1$ and $\mathrm{que}_2 \circ c_{12} \circ c_{22}$ to $\mathcal{S}_2$. Each server responds with a string of $2 + |\mathrm{ans}_1| + |\mathrm{ans}_2|$ bits. $\mathcal{S}_1$ sends back $1 \circ (c_{11}, \mathrm{ans}_1) \circ c_{21} \circ \mathrm{ans}_1$. $\mathcal{S}_2$ sends back $(c_{22}, \mathrm{ans}_2) \circ 1 \circ \mathrm{ans}_2 \circ c_{12}$. It is easy to see that the dot product of the servers' answers yields x_i, and that the procedure above increases the overall communication only by a constant factor. $\square$

3.2 Group-based PIR

Finite groups are a natural source of generalized Latin squares. Let $G = \{g_1, \ldots, g_T\}$ be a finite group of size T. Let $S = \{s_1, \ldots, s_n\} \subseteq G$ be an ordered subset of G of size n. A generalized Latin square $Q_{G,S}$ is a T by T square matrix whose rows and columns are labelled by elements of G, and $Q_{g_1,g_2} = i$ if $g_1 g_2^{-1} = s_i$, while all other locations contain stars.

When a PIR protocol $Q \hookrightarrow A$ uses a generalized Latin square $Q_{G,S}$ we say that it *employs a group-based secret sharing scheme*. Essentially, this means that given an index i, $\mathcal{U}$ maps it to a group element s_i, represents s_i as a random product in the group $s_i = g_1 g_2^{-1}$, and sends g_j to $\mathcal{S}_j$.

The notion of a *group-based* PIR protocol (for which we later prove a lower bound) is more restrictive. Let $M \in [q]^{T \times T}$ and G be finite group. Assume that the rows and columns of M are labelled by $g_1, \ldots, g_T$. We say that M *respects* G if, for every $g_1, g_2, g_3, g_4 \in G$ such that $g_1 g_2^{-1} = g_3 g_4^{-1}$, we have $M_{g_1,g_2} = M_{g_3,g_4}$.

Definition 3.8. We say that a PIR protocol $Q \hookrightarrow A$ is *group-based* if it employs a secret sharing scheme based on some group G and, for every $\sigma : [n] \to \mathbb{F}_q$, there exists a completion C such that C reduces to A and C respects G.

Stated in other words, a PIR scheme is group-based if the servers represent the database by a function on a certain finite group G and the scheme allows the user to retrieve the value of this function at any group element using the natural secret sharing based on G.

4 Communication complexity of bilinear group-based PIR

Consider a bilinear group-based PIR scheme $Q \hookrightarrow H_r$ based on a group G, with answer length r. Clearly, the query length is $\log|G|$. Let $N(q,G,r)$ denote the number of $|G|$ by $|G|$ matrices over $\mathbb{F}_q$ that respect G (for some fixed labelling $\{g_1,\ldots,g_T\}$ or rows and columns) and have rank at most r. It is easy to see that

$$q^n \le N(q,G,r)\,, \tag{4.1}$$

since by Lemma 3.5 every database yields such a matrix and distinct databases yield distinct matrices. In Section 4.2 we obtain an equivalent algebraic definition for $N(q,G,r)$, and in Section 4.3 we prove an upper bound for $N(q,G,r)$. Our final result is a constraint on the range of possible values of q, $|G|$, and r. This constraint implies an $\Omega(n^{1/3})$ lower bound for the total communication of any bilinear group-based PIR scheme.

4.1 Algebraic preliminaries

Our proof relies on some basic notions of the representation theory of finite groups. The standard references for this subject are [18], [8]. For a general algebra background see [16].

Let $G = \{g_1,\ldots,g_T\}$ be a finite (not necessarily abelian) group. The *general linear group* $GL_r(\mathbb{F}_q)$ is the multiplicative group of all non-singular r by r matrices over $\mathbb{F}_q$.

- An $\mathbb{F}_q$-*representation* of G of degree r is an homomorphism $\phi : G \to GL_r(\mathbb{F}_q)$.
- The *group algebra* $\mathbb{F}_q[G]$ of G over a field $\mathbb{F}_q$ is the algebra over $\mathbb{F}_q$ consisting of all possible formal linear combinations $\sum_{i=1}^{T} \alpha_i g_i$, $\alpha_i \in \mathbb{F}_q$. The algebraic operations in $\mathbb{F}_q[G]$ are defined by:

$$\sum_i \alpha_i g_i + \sum_i \beta_i g_i = \sum_i (\alpha_i+\beta_i) g_i\,;$$
$$\left(\sum_i \alpha_i g_i\right)\left(\sum_i \beta_i g_i\right) = \sum_{i,j} (\alpha_i\beta_j)(g_i g_j)\,;$$
$$\lambda\left(\sum_i \alpha_i g_i\right) = \sum_i (\lambda\alpha_i) g_i, \quad \lambda \in \mathbb{F}_q\,.$$

- A *left (right) ideal* in the group algebra $\mathbb{F}_q[G]$ is an $\mathbb{F}_q$-linear subspace of $\mathbb{F}_q[G]$ that is closed under left (right) multiplications by elements of $\mathbb{F}_q[G]$.
- A *left* $\mathbb{F}_q[G]$-*module* is an $\mathbb{F}_q$-linear space on which $\mathbb{F}_q[G]$ acts by left multiplication in such a way that for any $m_1,m_2 \in M$ and any $\alpha,\beta \in \mathbb{F}_q[G]$:

$$\alpha(m_1+m_2) = \alpha m_1 + \alpha m_2\,;$$
$$(\alpha+\beta)m_1 = \alpha m_1 + \beta m_1\,;$$
$$(\alpha\beta)m_1 = \alpha(\beta m_1)\,.$$

The *dimension* of a module is its dimension as an $\mathbb{F}_q$-linear space. Let M be an r-dimensional $\mathbb{F}_q[G]$-module, and let $m = \{m_1, \ldots, m_r\}$ be a basis of M. Multiplication by an element $g \in G$ induces a coordinate change in the basis m. Such a change can be expressed by an r by r matrix $\phi_{M,m}(g)$. The map $\phi_{M,m} : G \longrightarrow GL_r(\mathbb{F}_q)$ is an $\mathbb{F}_q$-representation of G of degree r. Two $\mathbb{F}_q[G]$-modules M, M' are called *isomorphic* if there exists an isomorphism between them as linear spaces that preserves multiplication by the elements of $\mathbb{F}_q[G]$. In the matrix form this means that for an arbitrary choice of bases m, m' in M, M' there exists $U \in GL_r(\mathbb{F}_q)$ such that $\phi_{M,m}(g) = U^{-1}\phi_{M',m'}(g)U$ $(g \in G)$. In particular, non-isomorphic modules correspond to different representations (again, for any choice of bases) and thus **the number of pairwise non-isomorphic left modules of dimension r does not exceed the number of different $\mathbb{F}_q$-representations $\phi : G \longrightarrow GL_r(\mathbb{F}_q)$ of degree r.**

Clearly, every left ideal of $\mathbb{F}_q[G]$ is a left $\mathbb{F}_q[G]$-module.

4.2 Algebraic formulation

Let $A = \mathbb{F}_q[G]$. For $\alpha \in A$, let $A\alpha$ denote the *principal left ideal* generated by α, that is, the set $\{\beta\alpha \mid \beta \in A\}$. Let rk $(\alpha) = \dim(A\alpha)$, where dim $(A\alpha)$ is the dimension of $A\alpha$ as a linear space over $\mathbb{F}_q$. Consider the *regular representation* ϕ of G, $\phi : G \rightarrow GL_{|G|}(\mathbb{F}_q)$, defined by

$$(\phi(g))_{g_1,g_2} = \begin{cases} 1, & g_1 g_2^{-1} = g, \\ 0, & \text{otherwise.} \end{cases} \tag{4.2}$$

Extend ϕ to A by linearity. Note that ϕ is an injective algebra homomorphism and that the image of ϕ is the $\mathbb{F}_q$-algebra R of all matrices that respect G. Observe that for any $M \in R$,

$$\mathrm{rk\ M = \dim\{M'M \mid M' \in R\}}\,. \tag{4.3}$$

To verify formula (4.3) one needs to notice that the first row of a matrix $M' \in R$ can be arbitrary. Therefore products $M'M$ contain all possible linear combinations of rows of M as their first row. Also notice that matrices in R are uniquely determined by their first row. Formula (4.3) follows. Since ϕ is injective, it implies that $\mathrm{rk}(\phi(\alpha)) = \mathrm{rk}(\alpha)$ $(\alpha \in \mathrm{A})$, and we arrive at the following alternate definition of $N(q,G,r)$:

$$N(q,G,r) = \#\{\alpha \in A \mid \mathrm{rk}(\alpha) \leq r\}\,. \tag{4.4}$$

4.3 Low-dimensional principal ideals in group algebras

Let V be an $\mathbb{F}_q$-linear subspace of $A = \mathbb{F}_q[G]$. The *left annihilator* of V is defined by $\mathrm{Ann}_L(V) \stackrel{\text{def}}{=} \{\beta \in A \mid \beta V = 0\}$. Similarly, the *right annihilator* is $\mathrm{Ann}_R(V) \stackrel{\text{def}}{=} \{\beta \in A \mid V\beta = 0\}$. Clearly, $\mathrm{Ann}_L(V)$ is a left ideal in A and $\mathrm{Ann}_R(V)$ is a right ideal in A. Let M be a left A-module. The *kernel* of M is defined by Ker $(M) \stackrel{\text{def}}{=} \{\beta \in A \mid \beta M = 0\}$. It is straightforward to verify that Ker (M) is a two-sided ideal that coincides with $\mathrm{Ann}_L(M)$ if M is a left ideal in A.

Lemma 4.1. *The number of r-dimensional left A-modules counted up to isomorphism is at most $q^{r^2 \log_2 |G|}$.*

Proof. As we remarked in Section 4.1, an upper bound on the number of $\mathbb{F}_q$-representations of G of degree r yields an upper bound on the number of non-isomorphic r-dimensional A-modules.

To bound the number of representations, let $g_1,\dots,g_s$ be a set of generators for G, where $s \le \log_2|G|$, and note that every representation $\phi : G \to GL_r(\mathbb{F}_q)$ is uniquely specified by s matrices $\phi(g_1),\dots,\phi(g_s)$ each of size r by r. $\square$

Clearly, isomorphic modules have identical kernels. Now we show that the kernel of a low-dimensional module has high dimension.

Lemma 4.2. *Let M be an r-dimensional left A-module; then the dimension of* $\mathrm{Ker}\,(M)$ *as an $\mathbb{F}_q$-linear space is at least $|G|-r^2$.*

Proof. Fix arbitrarily a basis $m=\{m_1,\dots,m_r\}$ in M, consider the representation $\phi_{M,m}$ from Section 4.1 and extend it by linearity to an algebra homomorphism $\phi_{M,m} : A \to GL_r(\mathbb{F}_q)$ as in Section 4.2. Then $\mathrm{Ker}\,(M)$ is just the kernel of $\phi_{M,m}$, and the statement follows from basic linear algebra: for any linear mapping $\phi : V \to W$ we have $\dim(\mathrm{Ker}(\phi)) \ge \dim(V)-\dim(W)$. $\square$

Lemma 4.3. *Suppose V is an $\mathbb{F}_q$-linear subspace of A; then* $\dim(\mathrm{Ann}_R(V)) \le |G|-\dim(V)$.

Proof. Consider a bilinear map $\ell : A\otimes A \to \mathbb{F}_q$, setting $\ell(x\otimes y)$ equal to the coefficient of 1 in the expansion of xy in the group basis. Recall from basic linear algebra that given any bilinear map $\ell : U\otimes V \to \mathbb{F}_q$ we can define its *rank* $\mathrm{rk}(\ell)$ by choosing bases $\{u_1,\dots,u_m\}$ in U and $\{v_1,\dots,v_n\}$ in V, and letting $\mathrm{rk}(\ell)$ be the rank of the m by n matrix with the entries $\ell(u_i\otimes v_j)$. $\mathrm{rk}(\ell)$ does not depend on the choice of the bases. As a consequence, for every subspaces $U'\subseteq U$, $V'\subseteq V'$ such that $\ell(U'\otimes V')=0$ we have the inequality $\dim(U')+\dim(V') \le m+n-\mathrm{rk}(\ell)$ (if an m by n matrix of rank r contains an m' by n' zero submatrix, then $m'+n' \le m+n-r$).

In our situation, $\mathrm{rk}(\ell)=|G|$ (since in the group basis ℓ is represented by a permutation matrix). Also, $\ell(V\otimes \mathrm{Ann}_R(V))=0$. Therefore $\dim(\mathrm{Ann}_R(V)) \le |G|-\dim(V)$ by the above. $\square$

Our main technical result is given by

Theorem 4.4. *For an arbitrary finite group G and arbitrary values of q and r*

$$N(q,G,r) \le q^{O(r^2\log|G|)}.$$

Proof. Let $\alpha \in A$ be such that $\mathrm{rk}\,(\alpha) \le r$. Consider $A\alpha$ as a left A-module. $\mathrm{Ker}\,(A\alpha)$ is the two-sided ideal $I=\mathrm{Ann}_L(A\alpha)$. Note that $\alpha \in \mathrm{Ann}_R(I)$. By Lemma 4.1, every A-module of dimension up to r has its kernel coming from a family of at most $\sum_{i=1}^r q^{i^2\log_2|G|} \le rq^{r^2\log_2|G|}$ ideals. Also by Lemmas 4.2 and 4.3 there are at most q^{r^2} elements in $\mathrm{Ann}_R(I)$ for every I. $\square$

Combining Equation (4.1) with Theorem 4.4 we obtain our main result.

Theorem 4.5. *Let $Q \hookrightarrow H_r$ be a bilinear group-based PIR scheme over a group G. Let $t=\log|G|$ denote the query length and r denote the answer length; then*

$$n \le O(tr^2).$$

In particular the total communication of any such scheme is $\Omega(n^{1/3})$.

5 Conclusion

We introduced a novel and quite natural combinatorial view of the two-server PIR problem, and obtained a lower bound for the communication complexity of PIR in the corresponding model. Stated informally, our main result is that as long as the servers represent the database by a function on a finite group, the protocol allows the user to retrieve the value of this function at any group element, and the user computes the dot product of the servers' responses to obtain the final answer, the communication complexity has to be $\Omega(n^{1/3})$. Clearly, our result admits two interpretations. On the one hand it can be viewed as a witness in support of the conjecture of Chor et al. from [5] saying that their PIR protocol with $O(n^{1/3})$ communication is asymptotically optimal. On the other hand, our result exhibits a common shortcoming of the existing upper bound techniques and thus hopefully may provide some directions for future work on upper bounds. We would like to stress the first interpretation of our result by revisiting and discussing all restrictions that we introduced in order to prove the lower bound:

1. We restricted ourselves to bilinear protocols, i. e., protocols where $\mathcal{U}$ computes the dot product of the servers' responses. Bilinearity is a weaker assumption than linearity, therefore if one believes that linear PIRs come close to optimal, then so do bilinear ones.

2. We restricted $\mathcal{U}$ to toss a linear number of coins in the length of his queries. Although this restriction seems a technicality, so far we have not been able to remove it. The only justification that we have is that it would seem quite surprising if indeed optimal PIR schemes require a very large amount of randomness. If one accepts restrictions 1-2, then a PIR protocol is just a pair $Q \hookrightarrow H_r$ such that for every $\sigma : [n] \to \mathbb{F}_q$, Q_σ can be completed to a matrix of rank at most r.

3. We further restrict the generalized Latin square Q to be of the form $\mathrm{GLS}_{G,S}$ for certain subset S of a finite group G. Generalized Latin squares of this form constitute a rich and natural class. In other words, this restriction states that $\mathcal{U}$ employs a group-based secret sharing scheme to share the index i between the servers.

4. Our last restriction is a restriction on the structure of low rank completions of matrices Q_σ. We require that for every σ there exists a completion C of Q_σ to a matrix of rank at most r subject to the extra constraint that C respects G. Our only evidence for this restriction is that so far we are unaware of examples of matrices Q_σ (with parameters suitable for nontrivial PIR) whose minimal rank with respect to locations labelled by stars would be substantially smaller than the minimal rank subject to an extra constraint of respecting G.

We proved that the communication complexity of any PIR scheme that satisfies restrictions 1-4 is $\Omega(n^{1/3})$. We leave it up to the reader to decide whether to accept each of the restrictions 1-4 as reasonable. We hope that ideas and techniques that we introduced may lead to further progress towards understanding the true communication complexity of private information retrieval. In particular the following problem is intriguing:

Open problem: Let Q be a $\mathrm{GLS}[n,n^\delta]$ of inverse polynomial density. Show that there exists a map $\sigma : [n] \to \mathbb{F}_q$ such that the minimal $\mathbb{F}_q$-rank of Q_σ (with respect to locations containing stars in Q_σ) is $\omega(\log n)$.

Comment: If true this implies an $\omega(\log n)$ lower bound for every bilinear PIR scheme, where $\mathcal{U}$ tosses a linear number of coins in the length of his queries. If false, this yields a PIR protocol with $c\log n$ communication. It may also be interesting to see if there is any formal connection between this problem and the well-known *matrix rigidity* problem over finite fields.

6 Appendix: Current PIR schemes are bilinear group-based

A number of two-server PIR schemes are known to date [5, 1, 3, 9, 4, 19]. The goal of this section is to show that all of them can be easily transformed into bilinear group-based schemes. We restrict ourselves to schemes from [5, 3, 19] since every other scheme is a variant of one of them. We do not follow the chronological order in which the schemes were proposed.

All known two-server PIR schemes rely on the idea of polynomial interpolation. Specifically, the retrieval of x_i, where the servers hold database x and the user holds index i, is reduced to an evaluation of a cubic polynomial $F(z_1,\ldots,z_m) \in \mathbb{F}_q[z_1,\ldots,z_m]$, held by the servers, on a point $E(i)$, that the user determines based on i. We refer to $E(i)$ as the encoding of i.

We use the encoding function $E:\ [n] \to \mathbb{F}_q^m$ that has been previously used in [5, 3]. Without loss of generality assume that $m' = n^{1/3}$ is an integer. Consider an arbitrary bijection $\gamma : [n] \to [m'] \times [m'] \times [m']$. Let $e'_l \in \{0,1\}^{m'}$ denote a vector whose unique nonzero coordinate is ℓ. Set $m = 3m'$. Put

$$E(i) = e'_{\gamma(i)_1} \circ e'_{\gamma(i)_2} \circ e'_{\gamma(i)_3}\,.$$

Note that for every i, $E(i)$ has three nonzero coordinates. Define

$$F(z_1,\ldots,z_m) = \sum_{i=1}^{n} x_i \prod_{E(i)_\ell=1} z_\ell\,,$$

($E(i)_l$ is the ℓ-th coordinate of $E(i)$). Since each $E(i)$ is of weight three, the degree of F is three. Each assignment $E(i)$ to the variables z_i satisfies exactly one monomial in F (whose coefficient is x_i); thus, $F(E(i)) = x_i$.

6.1 The monomial distribution scheme of [3]

For simplicity we restrict ourselves to the case when the underlying field is $\mathbb{F}_2$. Given a cubic multivariate polynomial $F(z_1,\ldots,z_m) \in \mathbb{F}_2[z_1,\ldots,z_m]$, the servers compute a new polynomial in $2m$ variables

$$\hat{F}(v_1,\ldots,v_m,w_1,\ldots,w_m) = F(v_1+w_1,\ldots,v_m+w_m)\,.$$

The servers rewrite $\hat{F}$ as a sum of two polynomials

$$\hat{F}(v_1,\ldots,v_m,w_1,\ldots,w_m) = \hat{F}_v(v_1,\ldots,v_m,w_1,\ldots,w_m) + \hat{F}_w(v_1,\ldots,v_m,w_1,\ldots,w_m)\,,$$

where $\hat{F}_v$ is the sum of all monomials from $\hat{F}$ that contain at least two variables v_j, and $\hat{F}_w$ is the sum of all monomials from $\hat{F}$ that contain at least two variables w_j. Note that every monomial of $\hat{F}$ goes either to $\hat{F}_v$ or to $\hat{F}_w$. Servers further rewrite $\hat{F}_v$ and $\hat{F}_w$ to obtain

$$\hat{F}_v(v_1,\dots,v_m,w_1,\dots,w_m) = F(v_1,\dots,v_m) + \sum_{\ell=1}^{m} c_\ell(v_1,\dots,v_m)w_\ell\,, \tag{6.1}$$

$$\hat{F}_w(v_1,\dots,v_m,w_1,\dots,w_m) = F(w_1,\dots,w_m) + \sum_{\ell=1}^{m} c_\ell(w_1,\dots,w_m)v_\ell\,. \tag{6.2}$$

The formal description of the scheme is below. Recall that the user holds $P \in \mathbb{F}_2^m$ and wishes to retrieve $F(P)$.

$\mathcal{U}$	:	Represents P as a random sum $P = V + W$ for $V, W \in \mathbb{F}_2^m$.
$\mathcal{U} \to \mathcal{S}_1$	:	$(v_1,\dots,v_m)$
$\mathcal{U} \to \mathcal{S}_2$	:	$(w_1,\dots,w_m)$
$\mathcal{U} \leftarrow \mathcal{S}_1$	:	$F(V), c_1(V),\dots,c_m(V)$
$\mathcal{U} \leftarrow \mathcal{S}_2$	:	$F(W), c_1(W),\dots,c_m(W)$
$\mathcal{U}$	:	Outputs $F(V)+F(W)+(V,(c_1(W),\dots,c_m(W)))+(W,(c_1(V),\dots,c_m(V)))$

Note that the protocol above is group-based, since the user can retrieve $F(P)$ for any $P \in \mathbb{F}_2^m$, and the user's secret sharing scheme is based on $\mathbb{F}_2^m$. Unfortunately, in the current form, the protocol is not bilinear. It is not hard to modify the protocol to achieve bilinearity.

Bilinear group-based form:

$\mathcal{U}$	:	Represents P as a random sum $P = V + W$ for $V, W \in \mathbb{F}_2^m$.
$\mathcal{U} \to \mathcal{S}_1$	:	$(v_1,\dots,v_m)$
$\mathcal{U} \to \mathcal{S}_2$	:	$(w_1,\dots,w_m)$
$\mathcal{U} \leftarrow \mathcal{S}_1$	:	$F(V) \circ 1 \circ c_1(V) \circ \dots \circ c_m(V) \circ v_1 \circ \dots \circ v_m$
$\mathcal{U} \leftarrow \mathcal{S}_2$	:	$1 \circ F(W) \circ w_1 \circ \dots \circ w_m \circ c_1(W) \circ \dots \circ c_m(W)$
$\mathcal{U}$	:	Outputs the dot product of the servers' replies

6.2 The combinatorial scheme of [5]

Unlike the PIR schemes of [3, 19], the scheme of [5] does not explicitly mention low degree multivariate polynomials (or any other functions on groups), therefore it is not immediately clear how to make it bilinear group-based. However it was observed in [3] that in fact this scheme can also be cast in terms of polynomial evaluation. We now sketch the description of the scheme and show that it is essentially identical to the scheme of [3], and therefore can be turned into a bilinear group-based form.

Recall that $m' = n^{1/3}$ is an integer and $\gamma : [n] \to [m'] \times [m'] \times [m']$ is a bijection. For $S \subseteq [m']$ and $j \in [m']$ let

$$S \oplus j = \begin{cases} S \setminus \{j\}, & \text{if } j \in S, \\ S \cup \{j\}, & \text{otherwise.} \end{cases}$$

For $S_1, S_2, S_3 \subseteq [m']$ let

$$T(S_1,S_2,S_3) = \sum_{\{i \mid \forall j \in [3]:\ \gamma(i)_j \in S_j\}} x_i\,.$$

We say that a triple of sets $S'_1, S'_2, S'_3 \subseteq [m']$ is *at distance one* from a triple S_1, S_2, S_3 if there exist unique $j \in [3]$ and $k \in [m']$ such that $S_t = S'_t$ for $t \neq j$ and $S_j = S'_j \oplus k$. Let $B(S_1, S_2, S_3)$ denote the $3m'$ long vector of values of $T(S'_1, S'_2, S'_3)$ at triples S'_1, S'_2, S'_3 that are at distance one from S_1, S_2, S_3. Below is the formal description of the messages exchanged by the user and the servers:

$$\begin{array}{|lcl|}\hline \mathcal{U} & : & \text{Picks } S_1, S_2, S_3 \subseteq [m'] \text{ at random.} \\ \mathcal{U} \to \mathcal{S}_1 & : & S_1, S_2, S_3 \\ \mathcal{U} \to \mathcal{S}_2 & : & S_1 \oplus \gamma(i)_1, S_2 \oplus \gamma(i)_2, S_3 \oplus \gamma(i)_3 \\ \mathcal{U} \leftarrow \mathcal{S}_1 & : & T(S_1, S_2, S_3), B(S_1, S_2, S_3) \\ \mathcal{U} \leftarrow \mathcal{S}_2 & : & T(S_1 \oplus \gamma(i)_1, S_2 \oplus \gamma(i)_2, S_3 \oplus \gamma(i)_3), B(S_1 \oplus \gamma(i)_1, S_2 \oplus \gamma(i)_2, S_3 \oplus \gamma(i)_3) \\ \hline \end{array}$$

Now note that $T(S_1, S_2, S_3) = F(S_1 \circ S_2 \circ S_3)$. Let $P = E(i) \in \mathbb{F}_2^m$. Recall that $e_\ell \in \{0,1\}^m$ denotes a vector whose unique nonzero coordinate is ℓ. We rewrite the protocol above in a different notation:

$$\begin{array}{|lcl|}\hline \mathcal{U} & : & \text{Represents } P \text{ as a random sum } P = V + W \text{ for } V, W \in \mathbb{F}_2^m. \\ \mathcal{U} \to \mathcal{S}_1 & : & (v_1, \ldots, v_m) \\ \mathcal{U} \to \mathcal{S}_2 & : & (w_1, \ldots, w_m) \\ \mathcal{U} \leftarrow \mathcal{S}_1 & : & F(V), F(V + e_1), \ldots, F(V + e_m) \\ \mathcal{U} \leftarrow \mathcal{S}_2 & : & F(W), F(W + e_1), \ldots, F(W + e_m) \\ \hline \end{array}$$

Let c_ℓ denote the polynomial that has been previously used in formula (6.1). It is not hard to verify that

$$c_l(V) = F(V + e_l) + F(V). \tag{6.3}$$

Taking formula (6.3) into account we conclude that the combinatorial scheme above is essentially identical to the scheme from the previous subsection. Thus it can also be turned into a bilinear group-based form.

6.3 The partial derivatives scheme of [19]

An important difference of this scheme is that it requires the field size to be larger than 2. Fix two distinct nonzero elements $\lambda_1, \lambda_2 \in \mathbb{F}_q$. Let $f(\lambda) \in \mathbb{F}_q[\lambda]$ be a univariate cubic polynomial. Note that

$$f(0) = c_1 f(\lambda_1) + c_2 f'(\lambda_1) + c_3 f(\lambda_2) + c_4 f'(\lambda_2),$$

for some constants c_i that are independent of f.

Protocol description : We use standard mathematical notation $\left.\frac{\partial F}{\partial z_\ell}\right|_W$ to denote the value of the partial derivative of F with respect to z_ℓ at the point W. Let $P = E(i)$. The user wishes to retrieve $F(P)$.

$$\begin{array}{|lcl|}\hline \mathcal{U} & : & \text{Picks } V \in \mathbb{F}_q^m \text{ uniformly at random.} \\ \mathcal{U} \to \mathcal{S}_1 & : & P + \lambda_1 V \\ \mathcal{U} \to \mathcal{S}_2 & : & P + \lambda_2 V \\ \mathcal{U} \leftarrow \mathcal{S}_1 & : & F(P + \lambda_1 V), \left.\frac{\partial F}{\partial z_1}\right|_{P+\lambda_1 V}, \ldots, \left.\frac{\partial F}{\partial z_m}\right|_{P+\lambda_1 V} \\ \mathcal{U} \leftarrow \mathcal{S}_2 & : & F(P + \lambda_2 V), \left.\frac{\partial F}{\partial z_1}\right|_{P+\lambda_2 V}, \ldots, \left.\frac{\partial F}{\partial z_m}\right|_{P+\lambda_2 V} \\ \mathcal{U} & : & c_1 F(P + \lambda_1 V) + c_2 \sum_{\ell=1}^{m} \left.\frac{\partial F}{\partial z_\ell}\right|_{P+\lambda_1 V} V_\ell + c_3 F(P + \lambda_2 V) + c_4 \sum_{\ell=1}^{m} \left.\frac{\partial F}{\partial z_\ell}\right|_{P+\lambda_2 V} V_\ell \\ \hline \end{array}$$

Note that in the protocol above the servers represent the database by a function $F:\mathbb{F}_q^m \to \mathbb{F}_q$ on a group and the user can retrieve $F(P)$ for arbitrary element $P\in\mathbb{F}_q^m$. However, the protocol is not bilinear group-based, since the user does not secret share according to the group law (i. e., the difference of shares is different from P), and the user does not output the dot product of the servers' responses. It is not hard to modify the protocol to achieve the desired properties.

Bilinear group-based form:

$$\begin{array}{lcl}
\mathcal{U} & : & \text{Picks } V\in\mathbb{F}_q^m \text{ uniformly at random.}\\
\mathcal{U}\to\mathcal{S}_1 & : & (P+\lambda_1 V)\lambda_2/(\lambda_2-\lambda_1)\\
\mathcal{U}\to\mathcal{S}_2 & : & (P+\lambda_2 V)\lambda_1/(\lambda_2-\lambda_1)\\
\mathcal{U}\leftarrow\mathcal{S}_1 & : & F(P+\lambda_1 V)\circ c_3\circ\left[\frac{c_2}{\lambda_1-\lambda_2}\sum_{\ell=1}^{m}\frac{\partial F}{\partial z_\ell}\Big|_{P+\lambda_1 V}(P+\lambda_1 V)_\ell\right]\circ\frac{\partial F}{\partial z_1}\Big|_{P+\lambda_1 V}\circ\ldots\circ\frac{\partial F}{\partial z_m}\Big|_{P+\lambda_1 V}\circ\\
 & & \circ 1\circ\frac{-c_4}{\lambda_2-\lambda_1}(P+\lambda_1 V)_1\circ\ldots\circ\frac{-c_4}{\lambda_2-\lambda_1}(P+\lambda_1 V)_m\\
\mathcal{U}\leftarrow\mathcal{S}_2 & : & c_1\circ F(P+\lambda_2 V)\circ 1\circ\frac{-c_2}{\lambda_1-\lambda_2}(P+\lambda_2 V)_1\circ\ldots\circ\frac{-c_2}{\lambda_1-\lambda_2}(P+\lambda_2 V)_m\circ\\
 & & \circ\left[\frac{c_4}{\lambda_2-\lambda_1}\sum_{\ell=1}^{m}\frac{\partial F}{\partial z_l}\Big|_{P+\lambda_2 V}(P+\lambda_2 V)_\ell\right]\circ\frac{\partial F}{\partial z_1}\Big|_{P+\lambda_2 V}\circ\ldots\circ\frac{\partial F}{\partial z_m}\Big|_{P+\lambda_2 V}\\
\mathcal{U} & : & \text{Outputs the dot product of the servers' replies}
\end{array}$$

Acknowledgement

We would like to thank Noga Alon, Swastik Kopparty, Madhu Sudan and Avi Wigderson for many helpful discussions concerning this work.

References

[1] * ANDRIS AMBAINIS: Upper bound on the communication complexity of private information retrieval. In *Proc. 32nd Intern. Colloquium on Automata, Languages and Programming (ICALP'97)*, volume 1256 of *Lecture Notes in Computer Science*, pp. 401–407. Springer, 1997. [ICALP:j210805656376051]. 1, 6

[2] * RICHARD BEIGEL, LANCE FORTNOW, AND WILLIAM GASARCH: A tight lower bound for restricted PIR protocols. *Computational Complexity*, 15:82–91, 2006. [10.1007s00037-006-0208-3]. 1.2

[3] * AMOS BEIMEL, YUVAL ISHAI, AND EYAL KUSHILEVITZ: General constructions for information-theoretic private information retrieval. *J. Computer and System Sciences*, 71:213–247, 2005. [JCSS:10.1016/j.jcss.2005.03.002]. 1, 6, 6.1, 6.2

[4] * AMOS BEIMEL, YUVAL ISHAI, EYAL KUSHILEVITZ, AND JEAN-FRANCIOS RAYMOND: Breaking the $O\left(n^{1/(2k-1)}\right)$ barrier for information-theoretic private information retrieval. In *Proc. 43rd FOCS*, pp. 261–270. IEEE Computer Society Press, 2002. [FOCS:10.1109/SFCS.2002.1181949]. 1, 2, 6

[5] * BENNY CHOR, ODED GOLDREICH, EYAL KUSHILEVITZ, AND MADHU SUDAN: Private information retrieval. *J. ACM*, 45:965–981, 1998. [JACM:10.1145/293347.293350]. 1, 5, 6, 6.2

[6] * WILLIAM GASARCH: A survey on private information retrieval. *The Bulletin of the EATCS*, 82:72–107, 2004. 1

[7] * ODED GOLDREICH, HOWARD KARLOFF, LEONARD SCHULMAN, AND LUCA TREVISAN: Lower bounds for linear locally decodable codes and private information retrieval. In *Proc. 17th Computational Complexity Conf. (CCC'02)*, pp. 175–183. IEEE Computer Society Press, 2002. [CCC:10.1109/CCC.2002.1004353]. 1.2, 2.2

[8] * I. MARTIN ISAACS: *Character theory of finite groups*. Academic Press, 1976. 4.1

[9] * TOSHIYA ITOH: Efficient private information retrieval. *IEICE Trans. Fund. of Electronics, Commun. and Comp. Sci.*, E82-A:11–20, 1999. 6

[10] * TOSHIYA ITOH: On lower bounds for the communication complexity of private information retrieval. *IEICE Trans. Fund. of Electronics, Commun. and Comp. Sci.*, E82-A:157–164, 2001. 1.2

[11] * JONATHAN KATZ AND LUCA TREVISAN: On the efficiency of local decoding procedures for error-correcting codes. In *Proc. 32nd STOC*, pp. 80–86. ACM Press, 2000. [STOC:10.1145/335305.335315]. 1

[12] * KIRAN S. KEDLAYA AND SERGEY YEKHANIN: Locally decodable codes from nice subsets of finite fields and prime factors of Mersenne numbers. Electronic Colloquium on Computational Complexity (ECCC) TR07-040, 2007. [ECCC:TR07-040]. 1

[13] * IORDANIS KERENIDIS AND RONALD DE WOLF: Exponential lower bound for 2-query locally decodable codes via a quantum argument. *J. of Computer and System Sciences*, 69:395–420, 2004. [JCSS:10.1016/j.jcss.2004.04.007]. 1

[14] * ERAN MANN: Private access to distributed information. Master's thesis, Technion - Israel Institute of Technology, Haifa, 1998. 1

[15] * ALEXANDER RAZBOROV AND SERGEY YEKHANIN: An $\Omega(n^{1/3})$ lower bound for bilinear group based private information retrieval. In *Proc. 47th FOCS*, pp. 739–748. IEEE Computer Society Press, 2006. [FOCS:10.1109/FOCS.2006.10]. *

[16] * B. L. VAN DER WAERDEN: *Algebra*. Springer, 2003. 4.1

[17] * STEPHANIE WEHNER AND RONALD DE WOLF: Improved lower bounds for locally decodable codes and private information retrieval. In *Proc. 32nd Intern. Colloquium on Automata, Languages and Programming (ICALP'05)*, volume 3580 of *Lecture Notes in Computer Science*, pp. 1424–1436. Springer, 2005. [ICALP:1utwmhj4me3lr582]. 1, 1.2

[18] * STEVEN H. WEINTRAUB: Representation theory of finite groups: algebra and arithmetic. volume 59 of *Graduate Studies in Mathematics*. AMS, 2003. 4.1

[19] * DAVID WOODRUFF AND SERGEY YEKHANIN: A geometric approach to information-theoretic private information retrieval. In *Proc. 20th IEEE Computational Complexity Conf. (CCC'05)*, pp. 275–284. IEEE Computer Society Press, 2005. [CCC:10.1109/CCC.2005.2]. 1, 6, 6.2, 6.3

[20] * SERGEY YEKHANIN: *Locally decodable codes and private information retrieval schemes*. PhD thesis, Massachusetts Institute of Technology, 2007. 1

[21] * SERGEY YEKHANIN: Towards 3-query locally decodable codes of subexponential length. In *Proc. 39th STOC*, pp. 266–274. ACM Press, 2007. [STOC:10.1145/1250790.1250830]. 1

AUTHORS

Alexander A. Razborov
Institute for Advanced Study, Steklov Mathematical Institute
razborov@ias.edu
http://www.mi.ras.ru/~razborov/

Sergey Yekhanin
Institute for Advanced Study
yekhanin@ias.edu
http://math.ias.edu/~yekhanin/

ABOUT THE AUTHORS

ALEXANDER RAZBOROV graduated from the Steklov Mathematical Institute in 1987 under the supervision of Sergei I. Adian. The title of his dissertation was "On systems of equations in a free group" (in Russian). He is interested in theoretical computer science (especially complexity theory of any kind) and mathematics, more often discrete than not.

SERGEY YEKHANIN obtained his doctoral degree from MIT in 2007 under the supervision of Madhu Sudan. Sergey is currently a postdoc at the School of Mathematics of the Institute for Advanced Study. His research interests are in computational complexity theory, cryptography, and the theory of error-correcting codes.