Electronic Colloquium on Computational Complexity, Report No. 50 (2006)

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

Alexander A. Razborov[*]
IAS, Steklov Mathematical Institute
razborov@ias.edu

Sergey Yekhanin[†]
MIT
yekhanin@mit.edu

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 between two servers while each server individually learns no information about i. The main parameter of interest in a PIR scheme is its communication complexity, namely the number of bits exchanged by the user and the servers. A large amount of effort has been invested by researchers over the last decade in search for efficient PIR schemes. A number of different schemes [6, 4, 19] have been proposed, however all of them ended up with the same communication complexity of $O(n^{1/3})$. The best known lower bound to date is $5 \log n$ by [17]. 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.

Our lower bound applies to bilinear group based PIR schemes. A bilinear PIR scheme is a one round PIR scheme, where user computes the dot product of servers' responses to obtain the desired value of the i-th bit. Every linear scheme can be turned into a bilinear one. A group based PIR scheme, is a PIR scheme, that involves servers representing database by a function on a certain finite group G, and allows user to retrieve the value of this function at any group element using the natural secret sharing scheme based on G. Our proof relies on some basic notions of representation theory of finite groups. We also discuss the approaches one may take to obtain a general lower bound for bilinear PIR.

1 Introduction

Private information retrieval (PIR) was introduced in a seminal paper by Chor, Goldreich, Kuzhelevitz and Sudan [6]. 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 that, in a 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 [6] that in case information-theoretic privacy is required the above trivial solution is in fact optimal. To go around this Chor *et al.* suggested replicating the database among $k > 1$ non-communicating servers.

For the case of two servers [6] 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. For general k [6] achieved

[*] Supported by the Charles Simonyi Endowment and the NSF grant ITR-0324906.
[†] Supported by the NSF grant CCR 0219218.

ISSN 1433-8092

the complexity of $O(n^{1/k})$. Their bound was later improved by Ambainis [1] to $O(n^{1/(2k-1)})$. Finally in a breakthrough result [5] Beimel *et al.* achieved the communication complexity of $n^{O\left(\frac{\log\log k}{k\log k}\right)}$.

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 [15]. Later it was improved to $4.4\log n$ by Kerenidis and de Wolf [13] using the results of Katz and Trevisan [14]. 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.

To date PIR literature is extensive. There is 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 [7].

In the current paper we study 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 [6, 4, 19] all of them have the same communication complexity of $O(n^{1/3})$. Secondly, the work of [5] implies that any improvement of the upper bound for two server PIR, yields 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 essentially is a problem regarding the minimal size of an *induced universal graph* for a family of graphs with 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 user computes the dot product of servers' responses to obtain the desired value of the i-th bit. A group based PIR scheme, is a PIR scheme, that involves servers representing database by a function on a certain finite group G, and allows 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 turns out that communication complexity of bilinear group based PIR 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 some basic notions of representation theory of finite groups.

1.2 Related work

Apart from the work on general lower bounds for PIR protocols that we surveyed above, there has been some effort to establish (stronger) lower bounds for various restricted models of PIR. In particular Itoh [12] obtained polynomial lower bounds on communication complexity of one round PIR, under the assumption that each server returns a multilinear or affine function of its input. Goldreich *et. al.* [8] 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 servers' answers

[1] We actually prefer to use language of matrices rather then 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.

in order to compute x_i. Goldreich *et. al.* obtained polynomial lower bounds for communication complexity of two server linear PIR schemes whose probe complexity is constant. Later, their results were extended by Wehner and de Wolf [17] who showed that the restriction of linearity can in fact be dropped.

It is not easy to match the restricted models surveyed above against one another and against our model, because the restrictions are quite different. We do not impose any restriction on the functions computed by the servers as [12], and do not restrict the user to read only a small number of bits from servers' answers as [8]. We show that our bilinearity restriction is weaker than the linearity restriction of [8], since every linear protocol can be easily turned into a bilinear one. However we insist that the PIR scheme should employ group based secrete 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 current paper. We introduce 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 $[s] \stackrel{\mathrm{def}}{=} \{1, \ldots, s\}$. 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 database contains entries from alphabet $[q]$, rather then just a binary alphabet. We also assume some implicit bijection between $[q]$ and $\mathbb{F}_q$. Everywhere $\log$ stands for the $\log$ base q. 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 user $\mathcal{U}$ who knows n and wants 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 [5].

Definition 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 an 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 (que_1, 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 $ans_j = \mathcal{A}(j, x, que_j)$. (We can assume without loss of generality that servers are deterministic; hence, each answer is a function of a query and a database.) Finally, $\mathcal{U}$ computes its output by applying the reconstruction algorithm $\mathcal{C}(ans_1, ans_2, i, r)$. 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. To formalize this let $\mathcal{Q}_j$ denote the of j-th output of $\mathcal{Q}$, $j = 1, 2$. We require that for $j = 1, 2$ and any n, $i_1, i_2 \in [n]$ the distributions $\mathcal{Q}_j(i_1, r)$ and $\mathcal{Q}_j(i_2, r)$ are identical.*

The *communication complexity* of a PIR protocol $\mathcal{P}$, is a 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 *[8] A linear PIR scheme is a PIR scheme, where the answer function $\mathcal{A}(j, x, que_j)$ 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 *A generalized latin square $Q = GLS[n, T]$ is a square matrix 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 GLS$[n, T]$ contains precisely $(T - n)$ stars. We call the ratio n/T the *density* of a generalized latin square. It is easy to see that generalized latin squares of density 1 are simply latin squares.

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

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

Definition 4 *Let $Q = GLS[n, T]$, and $A \in [q]^{l \times l}$. 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 $c(Q_\sigma)$ reduces to A.*

Theorem 5 *The following two implications are valid:*

- *A pair $Q \hookrightarrow A$, where $Q = GLS[n, T]$, $A \in [q]^{l \times l}$, yields a two server PIR protocol with communication $\log T$ from $\mathcal{U}$ to each $\mathcal{S}_j$ and communication $\log l$ from $\mathcal{S}_j$'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 $Q = GLS\left[n, nq^{t(n)+\tau(n)}\right]$, 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 matrix A is known to all paries $\mathcal{U}, \mathcal{S}_1$ and $\mathcal{S}_2$. At the preprocessing stage, servers use the database $x \in [q]^n$, to define the map $\sigma : [n] \to [q]$, setting $\sigma(i) \overset{\text{def}}{=} x_i$. Also, they find an appropriate completion $c(Q_\sigma)$, and obtain maps $\pi_1 : [T] \to [l]$ and $\pi_2 : [T] \to [l]$, such for all j, k $c\left(Q_\sigma\right)_{jk} = A_{\pi_1(j), \pi_2(k)}$. The following protocol is further 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 on k. Correctness follows from the fact that $c(Q_\sigma)$ reduces to A. Total communication is given by $2(\log T + \log l)$.

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 (que_1, 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 $que'_1 = que_1 \circ i_1$, $que'_2 = que_2 \circ i_2$ 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 $ans_j \circ que'_j$.
- Finally, $\mathcal{C}'$ extracts $que_1, que_2, ans_1, ans_2$ 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 $\mathcal{Q}(i, r') = (que_1, que_2)$. $\mathcal{C}'$ runs $\mathcal{C}$ on (ans_1, ans_2, i, r') and returns the answer. 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. Variable que'_j ranges over Q'_j, and variable ans'_j ranges over A'_j. Let $R(que'_j, i)$ denote the set of random strings r that lead to query que'_j to server j on input i. Formally,

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

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

Rows of Q are labelled by possible pairs (que'_1, s), where $s \in [r(que'_1)]$. Similarly columns of Q are labelled by possible pairs (que'_2, s), where $s \in [r(que'_2)]$. We set $Q_{(que'_1, s_1),(que'_2, s_2)} = i$ if there exists a string $r \in R(que'_1, i) \cap R(que'_2, i)$ such that r is the string number s_1 in $R(que'_1, i)$ and the string number s_2 in $R(que'_2, i)$ with respect to lexicographic ordering of these sets; otherwise we set $Q_{(que'_1, s_1),(que'_2, s_2)} = *$.

Consider an arbitrary pair $(i, (que'_1, s_1))$, where $s_1 \in [r(que'_1)]$. Let r be the unique random string that has number s_1 in lexicographic ordering of $R(que'_1, i)$. Let $\mathcal{Q}'(i, r) = (que'_1, que'_2)$, and let s_2 be the number of r in lexicographic ordering of $R(que'_2, i)$. The column of Q labelled (que'_2, s_2) is the unique column such that $Q_{(que'_1, s_1),(que'_2, s_2)} = i$. We demonstrated 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 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 $A_{ans'_1, ans'_2} = \mathcal{C}'(ans'_1, ans'_2)$. The unspecified entries of A are set arbitrarily. Matrix A defined above may not be a 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 a database x, where $x_i = \sigma(i)$. We use 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 $\pi_1(que'_1, s_1) = \mathcal{A}'(1, x, que'_1)$ and $\pi_2(que'_2, s_2) = \mathcal{A}'(2, x, que'_2)$. Correctness property of $\mathcal{P}'$ implies that maps π_1, π_2 reduce certain completion of Q_σ to A. ■

The theorem above presents 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

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 6 *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 dot products of corresponding labels. I.e. $(H_m)_{v_1,v_2} = (v_1, v_2)$.*

Lemma 7 *Let M be a square matrix with entries from $\mathbb{F}_q$; then M reduces to Hadamard matrix H_m if and only if the $\mathbb{F}_q$ rank of M is at most m.*

Proof: Clearly, the $\mathbb{F}_q$ rank of H_m is m therefore the rank of any matrix that reduces to H_m is at most that much. 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 size 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 $v_j(i)u_k(i) = (M_i)_{jk}$. Now the maps $\pi_1 : [t] \rightarrow [q^m]$, $\pi_2 : [t] \rightarrow [q^m]$ 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 to reduce the proof that $Q \hookrightarrow H_m$ for some generalized latin square Q to showing that for every $\sigma : [n] \rightarrow \mathbb{F}_q$, Q_σ can be completed to a low rank matrix.

Definition 8 *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 servers' answers to obtain value of x_i. Next lemma shows that the restriction of bilinearity is weaker than that of linearity.

Lemma 9 *Every linear PIR protocol can be turned into a bilinear PIR protocol with the same asymptotic communication complexity.*

Proof: In a linear PIR protocol user receives two strings ans_1, ans_2 of linear combinations of database bits from servers, and the unit vector corresponding to the i-th bit of the database is guaranteed to be in the joint span of combinations from ans_1 and ans_2. The final output of $\mathcal{U}$ is a sum of two dot products $(c_1, ans_1)+(c_2, ans_2) = x_i$, for some vectors c_1 and c_2 that are computed by user along with queries (que_1, que_2). The idea behind turning a linear protocol into a bilinear one is simple.

After generating (que_1, 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 $que_1 \circ c_{11} \circ c_{21}$ to $\mathcal{S}_1$ and $que_2 \circ c_{12} \circ c_{22}$ to $\mathcal{S}_2$. Each server responds with a string of $2+|ans_1|+|ans_2|$ bits. $\mathcal{S}_1$ sends back $1 \circ (c_{11}, ans_1) \circ c_{21} \circ ans_1$. $\mathcal{S}_2$ sends back $(c_{22}, ans_2) \circ 1 \circ ans_2 \circ c_{12}$. It is easy to see that the dot product of servers answers yields x_i, and that the procedure above increases the overall communication only by a constant factor. ■

3.2 Group based PIR

Finite groups are a natural source of generalized latin squares $Q = \text{GLS}[n, T]$. Consider a finite group $G = \{g_1, \ldots, g_T\}$ of size T, and an ordered subset $S = \{s_1, \ldots, s_n\} \subseteq G$ 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.

In case PIR protocol $Q \hookrightarrow A$ uses a generalized latin square $Q_{G,S}$ we say that such protocol *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 that we later prove a lower bound) is more restrictive. Let $M \in [q]^{T \times T}$ and G be finite group. Assume that 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 10 *We say that 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 $c(Q_\sigma)$ such that $c(Q_\sigma)$ reduces to A and $c(Q_\sigma)$ respects G.*

Stated in other words a PIR scheme is group based if servers represent database by a function on a certain finite group G and the scheme allows 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, query length is $\log |G|$. Let $A(q, G, r)$ 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

$$q^n \leq A(q, G, r), \tag{1}$$

since by lemma 7 every database yields such a matrix and distinct databases yield distinct matrices. In section 4.2 we obtain an equivalent algebraic definition for $A(q, G, r)$, and in section 4.3 we prove an upper bound for $A(q, G, r)$. Our final result is a constraint on the range of possible values of $\log |G|, r$. This constraint implies an $\Omega(n^{1/3})$ lower bound for total communication of any bilinear group based PIR scheme.

4.1 Algebraic preliminaries

Our proof relies on some basic notions of representation theory of finite groups. The standard references for this subject are [18], [9]. For a general algebra background see [16].

Let G be a finite (not necessarily abelian) group, and $g_1, \ldots, g_T$ be all elements of G. General linear group $GL_r(\mathbb{F}_q)$ is a multiplicative group of all non-degenerate r by r matrices over $\mathbb{F}_q$.

- An $\mathbb{F}_q$ *representation* of G of degree r is an homomorphism $\phi : G \to GL_r(\mathbb{F}_q)$.
- A group algebra $\mathbb{F}_q[G]$ of G over a field $\mathbb{F}_q$ is an algebra over $\mathbb{F}_q$ consisting of all possible formal linear combinations $\sum\limits_{i=1}^{T} \alpha_i g_i$, $\alpha_i \in \mathbb{F}_q$. 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 $\mathbb{F}_q[G]$ module M is an $\mathbb{F}_q$ linear space that has left multiplication by the elements of $\mathbb{F}_q[G]$, such that for any $m_1, m_2 \in M$ and any $\alpha, \beta \in \mathbb{F}_q[G]$:

  $$\begin{gathered} \alpha(m_1 + m_2) = \alpha m_1 + \alpha m_2; \\ (\alpha + \beta)m_1 = \alpha m_1 + \beta m_1; \\ (\alpha\beta)m_1 = \alpha(\beta m_1). \end{gathered}$$

  Dimension of a module is its dimension as an $\mathbb{F}_q$ linear space. Two $\mathbb{F}_q[G]$ modules are called isomorphic if there exists an isomorphism between them as linear spaces that preserves multiplication by the elements of $\mathbb{F}_q[G]$.

- There is a one to one correspondence between r dimensional left $\mathbb{F}_q[G]$ modules M considered up to isomorphism and $\mathbb{F}_q$ representations of G of degree r considered up to inner automorphisms of the $GL_r(\mathbb{F}_q)$.

4.2 Algebraic formulation

Let $A = \mathbb{F}_q[G]$. For $\alpha \in A$, let $\text{rk}(\alpha) = \dim(A\alpha)$, 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, $\phi : G \to GL_{|G|}(\mathbb{F}_q)$, defined by

$$(\phi(g))_{g_1,g_2} = \begin{cases} 1, & g_1 g_2^{-1} = g, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

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

$$\text{rk}M = \dim\{M'M \mid M' \in R\}. \tag{3}$$

To verify formula (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 (3) follows. It implies an algebraic definition for $A(q, G, r)$:

$$A(q, G, r) = \#\{\alpha \in \mathbb{F}_q[G] \mid \text{rk}(\alpha) \leq r\}. \tag{4}$$

4.3 Low dimensional principal ideals in group algebras

Let V be an $\mathbb{F}_q$ linear subspace of A. Left annihilator of V is defined by $Ann_L(V) \overset{\text{def}}{=} \{\beta \in A \mid \beta V = 0\}$. Similarly, right annihilator $Ann_R(V) \overset{\text{def}}{=} \{\beta \in A \mid V\beta = 0\}$. Clearly, $Ann_L(V)$ is a left ideal in A and $Ann_R(V)$ is a right ideal in A. Let M be a left A module. Kernel of M is defined by $Ker(M) \overset{\text{def}}{=} \{\beta \in A \mid \beta M = 0\}$. It is straightforward to verify that $Ker(M)$ is a two sided ideal. Our main technical result is given by

Theorem 11 *For arbitrary finite group G and arbitrary values of q and r*

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

Lemma 12 *The number of r dimensional left A modules counted up to isomorphism is at most $q^{\log|G| r^2}$.*

Proof: The fourth bullet from subsection 4.1, implies that it suffices to count $\mathbb{F}_q$ representations of G of degree r. Let $g_1, \ldots, g_s$ be the set of generators for G, where $s \leq \log|G|$. Note that every representation $\phi : G \to GL_r(\mathbb{F}_q)$ is uniquely specified by s matrices $\phi(g_1), \ldots, \phi(g_s)$ each of size r by r. ■

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

Lemma 13 *Let M be an r dimensional left A module; then the dimension of $Ker(M)$ as an $\mathbb{F}_q$ linear space is at least $|G| - r^2$.*

Proof: Note that multiplication by an element of A induces a linear transformation of M. Such transformation can be expressed by an r by r matrix. Multiplication by a linear combination of elements of A corresponds to linear combination of corresponding matrices. Therefore $\dim Ker(M) \geq |G| - r^2$. ■

Lemma 14 *Suppose V is an $\mathbb{F}_q$ linear subspace of A; then $\dim(Ann_R(V)) \leq |G| - \dim(V)$.*

Proof: Consider a bilinear map $l : A \otimes A \to \mathbb{F}_q$, setting $l(x \otimes y)$ equal to the coefficient of 1 in the expansion of xy in the group basis. Clearly, l has full rank (since in the group basis l is defined by an identity matrix up to a permutation of columns). However $l(V \otimes Ann_R(V)) = 0$. Thus $\dim(Ann_R(V)) \leq |G| - \dim(V)$. ■

Proof of theorem 11: Let $\alpha \in A$ be such that $\text{rk}(\alpha) \leq r$. Consider $A\alpha$ as a left A module. $Ker(A\alpha)$ is a two-sided ideal $I = Ann_L(A\alpha)$. Note that $\alpha \in Ann_R(I)$. By lemma 12 every A module of dimension up to r has its kernel coming from a family of at most $rq^{\log |G| r^2}$ ideals. Also by lemmas 13 and 14 there are at most q^{r^2} elements in $Ann_R(I)$ for every I. ■

Combining equation (1) with theorem 11 we obtain our main result.

Theorem 15 *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*

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

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

5 Conclusion

We introduced a novel, though quite natural combinatorial view of the two server PIR problem, and obtained a lower bound for communication complexity of PIR in a restricted model. Stated informally, our main result is that as long as servers represent database by a function on a finite group, protocol allows user to retrieve the value of this function at any group element, and user computes the dot product of servers responses to obtain the final answer communication complexity has to be $\Omega(n^{1/3})$. Clearly, our result admits two interpretations. On the one had it can be viewed as a witness in support of conjecture of Chor *et. al.* from [6] 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 servers' responses. Bilinearity is a weaker assumption then linearity, therefore if one believes that linear PIRs come close to optimal, so do bilinear.

2. We restricted $\mathcal{U}$ to toss linearly many coins in the length of his queries. Although this restriction seems a technicality, so far we do not know how to go around it. The only justification that we have is that it would seem quite surprising if indeed optimal PIR schemes require very large amount of randomness. If one accepts restrictions 1-2; then 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 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 terms this restriction states that $\mathcal{U}$ employs a group based secret sharing scheme to share 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 an extra constraint that $c(Q_\sigma)$ 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 communication complexity of any PIR scheme that satisfies restrictions 1-4 is $\Omega(n^{1/3})$. We leave reader to choose 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 true communication complexity of private information retrieval. In particular the following problem is intriguing:

Open problem: Let $Q = \mathrm{GLS}[n, n^\delta]$ be a generalized latin square 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 well-known *matrix rigidity* problem.

Acknowledgement

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

References

[1] A. Ambainis, “Upper bound on the communication complexity of private information retrieval,” *Proc. of 32th ICALP, LNCS* 1256, pp. 401-407, 1997.

[2] D. Beaver and J. Feigenbaum, “Hiding Instances in Multioracle queries,” In *Proc. of the 7th Annual Symp. on Theoretical Aspects of Computer Science (STACS),* vol. 415 of LNCS, pp. 37-48, 1990.

[3] R. Beigel, L. Fortnow, and W. Gasarch, “A nearly tight lower bound for private information retrieval protocols,” Technical Report TR03-087, *Electronic Colloquim on Computational Complexity* (ECCC), 2003.

[4] A. Beimel and Y. Ishai. “Information-Theoretic Private Information Retrieval: A Unified Construction,” Technical Report TR01-15, *Electronic Colloquim on Computational Complexity* (ECCC), 2001. Extended abstract in: ICALP 2001, vol. 2076 of LNCS, pp. 89-98, 2001.

[5] A. Beimel, Y. Ishai, E. Kushilevitz, and J. F. Raymond. “Breaking the Barrier for Information-Theoretic Private Information Retrieval,” In *Proc. of the 43rd IEEE Symposium on Foundations of Computer Science (FOCS),* pp. 261-270, 2002.

[6] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan. “Private information retrieval,” In *Proc. of the 36rd IEEE Symposium on Foundations of Computer Science (FOCS),* pp. 41-50, 1995. Also, in *Journal of the ACM,* 45, 1998.

[7] B. Gasarch, A Webpage on Private Information Retrieval, http://www.cs.umd.edu/ gasarch/pir/pir.html

[8] O. Goldreich, H. Karloff, L. Schulman, L. Trevisan "Lower bounds for locally decodable codes and private information retrieval," In *Proc. of the 17th IEEE Computational Complexity Conference (CCC)*, pp. 175-183, 2002.

[9] I.M. Isaacs, Character theory of finite groups, Academic Press, 1976.

[10] Y. Ishai and E. Kushilevitz "Improved upper bounds on information-theoretic private information retrieval," In *Proc. of the 31th ACM Sym. on Theory of Computing (STOC)*, pp. 79-88, 1999.

[11] T. Itoh, "Efficient private information retrieval," *IEICE Trans. Fund. of Electronics, Commun and Comp. Sci.,* E82-A(1):11-20, 1999.

[12] T. Itoh, "On lower bounds for the communication complexity of private information retrieval," *IEICE Trans. Fund. of Electronics, Commun. and Comp. Sci.,* E84-A(1):157-164, 2001.

[13] I. Kerendis, R. deWolf, "Exponential Lower Bound for 2-query locally decodable codes via a quantum argument," *Journal of Computer and System Sciences,* 69(3), pp. 395-420. Earlier version in STOC'03. quant-ph/0208062.

[14] J. Katz and L. Trevisan, "On the efficeincy of local decoding procedures for error-correcting codes," In *Proc. of the 32th ACM Sym. on Theory of Computing (STOC)*, pp. 80-86, 2000.

[15] E. Mann, Private access to distributed information. Master's thesis, Technion - Israel Institute of Technology, Haifa, 1998.

[16] B.L. Van der Waerden, Alegbra, Springer, 2003.

[17] S. Wehner and R. de Wolf, "Improved Lower Bounds for Locally Decodable Codes and Private Information Retrieval, " preprint available from the LANL quant-ph archive 0403140, 2004.

[18] S.H. Weintraub, Representation theory of finite groups: algebra and arithmetic, AMS, Graduate studies in mathematics, Vol. 59, 2003.

[19] D. Woodruff and S. Yekhanin, "A Geometric Approach to Information Theoretic Private Information Retrieval," In *Proc. of the 20th IEEE Computational Complexity Conference (CCC)*, pp. 275-284, 2005.

6 Appendix: Current PIR schemes are bilinear group based

A number of two server PIR schemes are known to date [6, 1, 10, 4, 11, 5, 19]. The goal of this section is to show that all of them can be easily turned into bilinear group based. We restrict ourselves to schemes from [6, 4, 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.

It was observed in [4] that all known PIR schemes rely (implicitly or explicitly) on the idea of polynomial interpolation. [2] 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 multivariate polynomial $F(z_1, \dots, z_m) \in \mathbb{F}_q[z_1, \dots, z_m]$, held by the servers, on a point $E(i)$, which 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 [6, 4]. 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

$$F(z_1, \dots, z_m) = \sum_{i=1}^{n} x_i \prod_{E(i)_l = 1} z_l,$$

($E(i)_l$ 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 Monomial distribution scheme of [4]

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

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

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

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

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

$$\begin{aligned} \hat{F}_v(v_1, \dots, v_m, w_1, \dots, w_m) &= F(v_1, \dots, v_m) + \sum_{l=1}^{m} c_l(v_1, \dots, v_m) w_l \\ \hat{F}_w(v_1, \dots, v_m, w_1, \dots, w_m) &= F(w_1, \dots, w_m) + \sum_{l=1}^{m} c_l(w_1, \dots, w_m) v_l \end{aligned} \tag{5}$$

The formal description of the scheme is below. Recall that user holds $P \in \mathbb{F}_2^m$ and wants 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 $F(V) + F(W) + (V, (c_1(W), \dots, c_m(W))) + (W, (c_1(V), \dots, c_m(V)))$

[2] This claim remains true although a number of new PIR schemes appeared after [4] was published.

Note that the protocol above is group based, since the user can retrieve $F(P)$ for any $P \in \mathbb{F}_2^m$, and 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.

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

6.2 Combinatorial scheme of [6]

Unlike the PIR schemes of [4, 19] the scheme of [6] 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 [4] 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 [4], and therefore can be turned into a bilinear group based form.

Recall that $m' = n^{1/3}$ is an integer and $\gamma : [n] \rightarrow [m'] \times [m'] \times [m']$ is bijective. 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

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

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:

$\mathcal{U}$	:	Picks $S_1, S_2, S_3 \subseteq [m']$ uniformly at random.
$\mathcal{U} \rightarrow \mathcal{S}_1$	:	S_1, S_2, S_3
$\mathcal{U} \rightarrow \mathcal{S}_2$	:	$S_1 \oplus \phi(i)_1, S_2 \oplus \phi(i)_2, S_3 \oplus \phi(i)_3$
$\mathcal{U} \leftarrow \mathcal{S}_1$	:	$T(S_1, S_2, S_3), B(S_1, S_2, S_3)$
$\mathcal{U} \leftarrow \mathcal{S}_2$	:	$T(S_1 \oplus \phi(i)_1, S_2 \oplus \phi(i)_2, S_3 \oplus \phi(i)_3), B(S_1 \oplus \phi(i)_1, S_2 \oplus \phi(i)_2, S_3 \oplus \phi(i)_3)$

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 $e_l \in \{0,1\}^m$ denotes a vector whose unique nonzero coordinate is l. We rewrite the protocol above in a different notation:

$\mathcal{U}$	:	Represents P as a random sum $P = V + W$ for $V, W \in \mathbb{F}_2^m$.
$\mathcal{U} \rightarrow \mathcal{S}_1$	:	$(v_1, \ldots, v_m)$
$\mathcal{U} \rightarrow \mathcal{S}_2$	:	$(w_1, \ldots, w_m)$
$\mathcal{U} \leftarrow \mathcal{S}_1$	:	$F(V), F(V + e_1), \ldots, F(V + e_m)$
$\mathcal{U} \leftarrow \mathcal{S}_2$	:	$F(W), F(W + e_1), \ldots, F(W + e_m)$

Let c_l denote the polynomial that has been previously used in formula (5). It is not hard to verify that

$$c_l(V) = F(V + e_l) + F(V). \tag{6}$$

Taking formula (6) 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 Partial derivatives scheme of [19]

An important difference of this scheme is that it requires 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 $\left.\frac{\partial F}{\partial z_l}\right|_W$ to denote the value of the partial derivative of F with respect to z_l at point W. Let $P = E(i)$. The user wants to retrieve $F(P)$.

$\mathcal{U}$	:	Picks $V \in \mathbb{F}_q^m$ uniformly at random.
$\mathcal{U} \rightarrow \mathcal{S}_1$	:	$P + \lambda_1 V$
$\mathcal{U} \rightarrow \mathcal{S}_2$	:	$P + \lambda_2 V$
$\mathcal{U} \leftarrow \mathcal{S}_1$	:	$F(P + \lambda_1 V), \left.\frac{\partial F}{\partial z_1}\right\vert_{P+\lambda_1 V}, \dots, \left.\frac{\partial F}{\partial z_m}\right\vert_{P+\lambda_1 V}$
$\mathcal{U} \leftarrow \mathcal{S}_2$	:	$F(P + \lambda_2 V), \left.\frac{\partial F}{\partial z_1}\right\vert_{P+\lambda_2 V}, \dots, \left.\frac{\partial F}{\partial z_m}\right\vert_{P+\lambda_2 V}$
$\mathcal{U}$	:	Outputs $c_1 F(P + \lambda_1 V) + c_2 \sum_{l=1}^{m} \left.\frac{\partial F}{\partial z_l}\right\vert_{P+\lambda_1 V} V_l + c_3 F(P + \lambda_2 V) + c_4 \sum_{l=1}^{m} \left.\frac{\partial F}{\partial z_l}\right\vert_{P+\lambda_2 V} V_l$

Note that in the protocol above servers represent database by a function $F : \mathbb{F}_q^m \rightarrow \mathbb{F}_q$ on a group and 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 servers' responses. It is not hard to modify the protocol to achieve the desired properties.

Bilinear group based form:

$\mathcal{U}$	:	Picks $V \in \mathbb{F}_q^m$ uniformly at random.
$\mathcal{U} \rightarrow \mathcal{S}_1$	:	$(P + \lambda_1 V)\lambda_2/(\lambda_2 - \lambda_1)$
$\mathcal{U} \rightarrow \mathcal{S}_2$	:	$(P + \lambda_2 V)\lambda_1/(\lambda_2 - \lambda_1)$
$\mathcal{U} \leftarrow \mathcal{S}_1$	:	$F(P + \lambda_1 V) \circ c_3 \circ \left[\frac{c_2}{\lambda_1 - \lambda_2} \sum_{l=1}^{m} \left.\frac{\partial F}{\partial z_l}\right\vert_{P+\lambda_1 V} (P + \lambda_1 V)_l\right] \circ \left.\frac{\partial F}{\partial z_1}\right\vert_{P+\lambda_1 V} \circ \dots \circ \left.\frac{\partial F}{\partial z_m}\right\vert_{P+\lambda_1 V} \circ$
		$\circ 1 \circ \frac{-c_4}{\lambda_2 - \lambda_1}(P + \lambda_1 V)_1 \circ \dots \circ \frac{-c_4}{\lambda_2 - \lambda_1}(P + \lambda_1 V)_m$
$\mathcal{U} \leftarrow \mathcal{S}_1$	:	$c_1 \circ F(P + \lambda_2 V) \circ 1 \circ \frac{-c_2}{\lambda_1 - \lambda_2}(P + \lambda_2 V)_1 \circ \dots \circ \frac{-c_2}{\lambda_1 - \lambda_2}(P + \lambda_2 V)_m \circ$
		$\circ \left[\frac{c_4}{\lambda_2 - \lambda_1} \sum_{l=1}^{m} \left.\frac{\partial F}{\partial z_l}\right\vert_{P+\lambda_2 V} (P + \lambda_2 V)_l\right] \circ \left.\frac{\partial F}{\partial z_1}\right\vert_{P+\lambda_2 V} \circ \dots \circ \left.\frac{\partial F}{\partial z_m}\right\vert_{P+\lambda_2 V}$
$\mathcal{U}$	:	Outputs the dot product of servers' responses.