A Geometric Approach to Information-Theoretic Private Information Retrieval

David Woodruff [*]
MIT
dpwood@mit.edu

Sergey Yekhanin [†]
MIT
yekhanin@mit.edu

Abstract

A t-private private information retrieval (PIR) scheme allows a user to retrieve the ith bit of an n-bit string x replicated among k servers, while any coalition of up to t servers learns no information about i. We present a new geometric approach to PIR, and obtain

- *A t-private k-server protocol with communication $O\left(\frac{k^2}{t}\log k\, n^{1/\lfloor(2k-1)/t\rfloor}\right)$, removing the $\binom{k}{t}$ term of previous schemes. This answers an open question of [14].*
- *A 2-server protocol with $O(n^{1/3})$ communication, polynomial preprocessing, and online work $O(n/\log^r n)$ for any constant r. This improves the $O(n/\log^2 n)$ work of [8].*
- *Smaller communication for instance hiding [3, 14], PIR with a polylogarithmic number of servers, robust PIR [9], and PIR with fixed answer sizes [4].*

To illustrate the power of our approach, we also give alternative, geometric proofs of some of the best 1-private upper bounds from [7].

1 Introduction

Private information retrieval (PIR) was introduced in a seminal paper by Chor *et al* [11]. In such a scheme a server holds an n-bit string $x \in \{0,1\}^n$, representing a database, and a user holds an index $i \in [n] \overset{\text{def}}{=} \{1,\ldots,n\}$. At the end of the protocol the user should learn x_i and the server should learn nothing about i. A trivial solution is for the server to send the user x. While private, the *communication complexity* is linear in n. In contrast, 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 really necessary to achieve privacy.

Unfortunately, if information-theoretic privacy is required, then there is no better solution than the trivial one [11]. To get around this, Chor *et al* [11] suggested replicating the database among $k > 1$ non-communicating servers. In this setting, one can do substantially better. Indeed, Chor *et al* [11] give a protocol with complexity $O(n^{1/3})$ for as few at two servers, and an $O(k^2 \log k\, n^{1/k})$ solution for the general case. Ambainis [1] then extended the $O(n^{1/3})$ protocol to achieve $O(2^{k^2} n^{1/(2k-1)})$ complexity for every k. Finally, in [7], building upon [14, 5], Beimel *et al* reduce the communication to $2^{\tilde{O}(k)} n^{\frac{2\log\log k}{k\log k}}$. For constant k, the latter is the best upper bound to date. The best lower bound is a humble $c\log n$ for some small constant $c > 1$ [18]. For a survey, see [12].

A drawback of all of these solutions is that if any two servers communicate, they can completely recover i. This motivates the notion of a *privacy threshold* t, $1 \le t \le k$, which limits the number of servers that might collude in order to get information about i. That is, the joint view of any t servers should be independent of i. The case $t > 1$ was addressed in [11, 14, 5]. Beimel and Ishai [5] give the best upper bound prior to this work: $O\left(\binom{k}{t}\frac{k^2}{t} n^{1/\lfloor(2k-1)/t\rfloor}\right)$. Since this bound grows rapidly with t, in [14] it is asked:

Can one avoid the $\binom{k}{t}$ overhead induced by our use of replication-based secret sharing?

We give a scheme with communication $O\left(\frac{k^2}{t}\log k\, n^{1/\lfloor(2k-1)/t\rfloor}\right)$ for any t, and thus answer this question in the affirmative.

Our upper bound is of considerable interest in the *oracle instance-hiding* scenario [2, 3]. In this setting there is a function $F_m : \{0,1\}^m \rightarrow \{0,1\}$ held by $\frac{m}{c\log m}$ oracles. The user has $P \in \{0,1\}^m$, and wants to privately retrieve $F_m(P)$, even if up to t oracles collude. The user's computation, let alone the total communication, should be polynomial in m. For constant t, running our PIR scheme on the truth table of F_m gives a scheme with total communication $\tilde{O}(m^{ct/2+2})$. This improves the previous bound[1] of $\tilde{O}(m^{ct/2+2+t})$ (see [14]) by a factor of

[*] Supported by an NDSEG fellowship.

[†] Supported in part by NTT Award MIT 2001-04 and NSF grant CCR 0219218.

[1] The best upper bound for 1-private PIR [7] does not apply since it is

m^t. When $m = \log n$, this is exactly the problem of PIR with $k = \Omega(\log n / \log \log n)$, for which we obtain the best known bound.

Another application of our techniques is k-out-of-l *robust* PIR [9]. In this scenario a user should be able to recover x_i even if after sending his queries, up to $l - k$ servers do not respond. Previous bounds for this problem include $O(kn^{1/k} l \log l)$ and $2^{\tilde{O}(k)} n^{\frac{2 \log \log k}{k \log k}} l \log l$ [9]. The first bound is weak for small k, while the second is weak for large k. We improve upon these with a k-out-of-l robust protocol with communication $O(kn^{1/(2k-1)} l \log l)$.

Another concern with the abovementioned solutions is the *time complexity* of the servers per query. Beimel *et al* [8] show, among other things, that if two servers are given polynomial-time preprocessing, then during the online stage they can respond to queries with $O(n/\log^2 n)$ work, while preserving $O(n^{1/3})$ total communication. By combining a balancing technique similar to that in [10] with a specially-designed 2-server protocol in our language, we can reduce the work to $O(n/\log^r n)$ for any constant $r > 0$. It is immediate from our construction that if a server has answers of size a for any $a = O(n^{1/3})$, then there is a 2-server protocol with query size $O(n/a^2)$. This, in particular, resolves an open question of [4].

We note that using techniques similar to those in [6], our 1-private protocols can be modified to achieve the best known *probe complexity*, which measures the number of bits the user needs to read in the server's answers. Moreover, since we improve upon Theorem 6.1 and Corollary 6.3 of [6], our construction also yields minor improvements for PIR schemes with logarithmic query length, yielding efficient locally decodable codes over large alphabets.

Finally, our techniques are of independent interest, and may serve as a tool for obtaining better upper bounds. As an example of the model's power, we give a new geometric proof of the best known upper bound for 1-private k-server PIR protocols of [7] for $k < 26$.

The general idea behind our protocols is the idea of polynomial interpolation. As in previous work, we model the database as a degree-d polynomial $F \in \mathbb{F}_q[z_1, \ldots, z_m]$ with $m = O(dn^{1/d})$. The polynomial F is such that there is an encoding $E : [n] \rightarrow \mathbb{F}_q^m$ for which $F(E(i)) = x_i$ for every $i \in [n]$. The user wants to retrieve the value $F(P)$ for $P = E(i)$ while keeping the identity of P private. To this end the user randomly selects a low-dimensional affine variety (i.e. line, curve, plane, etc.) $\chi \subseteq \mathbb{F}_q^m$ containing the point P and discloses certain subvarieties of χ to the servers. Each server computes and returns the values of F and the values of *partial derivatives* of F at every point on its subvariety. Finally, the user reconstructs the restriction of F to χ. In particular the user obtains the desired value of $F(P)$. The idea of polynomial interpolation has been used previously in the private information retrieval literature [2, 11, 3]; however, we significantly extend and improve upon earlier techniques through the use of derivatives and more general varieties.

Outline: In section 2 we introduce our notation and provide some necessary definitions. In section 3 we describe a non-recursive 1-private PIR protocol on a line. We also discuss the robustness of our protocol. Section 4 deals with t-private PIR protocols for arbitrary t, and discusses applications to instance-hiding. The underlying variety is a curve. In section 5 we present our construction of PIR protocols with preprocessing. Finally, in section 6 we wrap up with a geometric proof of some of the upper bounds of [7]. The underlying variety is a low dimensional affine space.

2 Preliminaries

By default, variables λ_h take values in a finite field $\mathbb{F}_q$ and variables P, V, V^j, Q and Q^j take values in $\mathbb{F}_q^m$. Let W be an element of $\mathbb{F}_q^m$. We use the subscript W_l to denote the l-th component of W.

A k-server PIR protocol involves k servers $\mathcal{S}_1, \ldots, \mathcal{S}_k$, each holding the same n-bit string x (the database), and a user $\mathcal{U}$ who knows n and wants to retrieve some bit x_i, $i \in [n]$, without revealing i. We restrict our attention to *one-round,* information-theoretic PIR protocols.

Definition : [7] A t-private PIR protocol is a triplet of algorithms $\mathcal{P} = (\mathcal{Q}, \mathcal{A}, \mathcal{C})$. At the beginning of the protocol, the user $\mathcal{U}$ invokes $\mathcal{Q}(k, n, i)$ to pick a randomized k-tuple of queries $(q_1, \ldots, q_k)$, along with an auxiliary information string aux. It sends each server $\mathcal{S}_j$ the query q_j and keeps aux for later use. Each server $\mathcal{S}_j$ responds with an answer $a_j = \mathcal{A}(k, j, x, q_j)$. (We can assume without loss of generality that the 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 $\mathcal{C}(k, n, a_1, \ldots, a_k, aux)$. A protocol as above should satisfy the following requirements:

- **Correctness :** For any $k, n,\ x \in \{0,1\}^n$ and $i \in [n]$, the user outputs the correct value of x_i with probability 1 (where the probability is over the randomness of $\mathcal{Q}$).

- t**-Privacy :** Each collusion of up to t servers learns no information about i. Formally, for any $k, n,\ i_1, i_2 \in [n]$, and every $T \subseteq [k]$ of size $|T| \leq t$ the distributions $\mathcal{Q}_T(k, n, i_1)$ and $\mathcal{Q}_T(k, n, i_2)$ are identical, where $\mathcal{Q}_T$ denotes concatenation of j-th outputs of $\mathcal{Q}$ for $j \in T$.

The *communication complexity* of a PIR protocol $\mathcal{P}$, denoted $C_{\mathcal{P}}(n, k)$ is a function of k and n measuring the total number of bits communicated between the user and k

not known how to make it t-private, and in any case, the dependence on k there is $2^{\Omega(k)}$.

servers, maximized over all choices of $x \in \{0,1\}^n$, $i \in [n]$ and random inputs.

In our protocols we represent the database x by a multivariate polynomial $F(z_1, \ldots, z_m)$ over a finite field. The important parameters of the polynomial F are its degree d and the number of variables m. A very similar representation has been used previously in [7]. An important difference of our representation is that we use polynomials over fields larger than $\mathbb{F}_2$. The polynomial F represents x in the following sense: with every $i \in [n]$ we associate a point $E(i) \in \mathbb{F}_q^m$; the polynomial F satisfies:

$$\forall i \in [n], \quad F(E(i)) = x_i.$$

We use the assignment function $E : [n] \to \mathbb{F}_q^m$ from [7]. Let $E(1), \ldots, E(n)$ denote n distinct points of Hamming weight[2] d with coordinate values from the set $\{0,1\} \subset \mathbb{F}_q$. Such points exist if $\binom{m}{d} \geq n$. Therefore $m = O(dn^{1/d})$ variables are sufficient. Define

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

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

Our constructions rely heavily on the notion of a derivative of a polynomial over a finite field. Recall that for $f(\lambda) = a_0 + \sum\limits_{i=1}^{d} a_i \lambda^i \in \mathbb{F}_q[\lambda]$ the derivative is defined by $f'(\lambda) = \sum\limits_{i=1}^{d} i a_i \lambda^{i-1}$.

We conclude the section with two technical lemmas.

Lemma 1 *Let $f \in \mathbb{F}_q[\lambda]$ and $s \leq char\mathbb{F}_q - 1$. Suppose*

$$f(\lambda_0) = f'(\lambda_0) = \ldots = f^{(s)}(\lambda_0) = 0,$$

then $(\lambda - \lambda_0)^{s+1} \mid f$.

Proof: See lemma 6.51 in [15] and note that $s! \neq 0$. ■

Lemma 2 *Suppose $\{\lambda_h\}, \{v_h^0\}, \{v_h^1\}$ are elements of $\mathbb{F}_q$, where $h \in [s]$ and $\{\lambda_h\}$ are distinct; then there exists at most one polynomial $f(\lambda) \in \mathbb{F}_q[\lambda]$ of degree $\leq 2s - 1$ such that $f(\lambda_i) = v_h^0$ and $f'(\lambda_h) = v_h^1$.*

Proof: Assume there exist two such polynomials $f_1(\lambda)$ and $f_2(\lambda)$. Consider their difference $f = f_1 - f_2$. Clearly, $f(\lambda_h) = f'(\lambda_h) = 0$ for all $h \in [s]$. Therefore, by lemma 1

$$\prod_{h=1}^{s} (\lambda - \lambda_h)^2 \;\Big|\; f(\lambda).$$

This divisibility condition implies that $f(\lambda) = 0$ since the degree of f is at most $2s - 1$. ■

3 PIR on the line

We start this section with a PIR protocol of [11]. This protocol has a simple geometric interpretation and has served as the starting point for our work.

Theorem 3 *([11]) There exists a 1-private k-server PIR protocol with communication complexity $O(k^2 \log k \, n^{1/(k-1)})$.*

Protocol description : Consider a finite field $\mathbb{F}_q$, where $k < q \leq 2k$. Let $\lambda_1, \ldots, \lambda_k \in \mathbb{F}_q$ be distinct and nonzero. Set $d = k - 1$. Let $P = E(i)$. The user wants to retrieve $F(P)$.

$\mathcal{U}$	:	Picks $V \in \mathbb{F}_q^m$ uniformly at random.
$\mathcal{U} \to \mathcal{S}_h$	:	$P + \lambda_h V$
$\mathcal{U} \leftarrow \mathcal{S}_h$	:	$F(P + \lambda_h V)$

Privacy : It is immediate to verify that the input $(P + \lambda_h V)$ of each server $\mathcal{S}_i$ is distributed uniformly over $\mathbb{F}_q^m$. Thus the protocol is private.

Correctness : We need to show that values $F(P + \lambda_h V)$ for $h \in [k]$ suffice to reconstruct $F(P)$. Consider the line $L = \{P + \lambda V \mid \lambda \in \mathbb{F}_q\}$ in the space $\mathbb{F}_q^m$. Let $f(\lambda) = F(P + \lambda V)$ be the restriction of F to L. Clearly, $f \in \mathbb{F}_q[\lambda]$ is a univariate polynomial of degree at most $d = k - 1$. Note that $f(\lambda_h) = F(P + \lambda_h V)$. Thus $\mathcal{U}$ knows the values of $f(\lambda)$ at k points and therefore can reconstruct $f(\lambda)$. It remains to note that $F(P) = f(0)$.

Complexity : The user sends each of k servers a length-m vector of values in $\mathbb{F}_q$. Recall that $m = O(dn^{1/d})$ and $k < q \leq 2k$. Thus the total communication from the user to all the servers is $O(k^2 \log k \, n^{1/(k-1)})$. Each $\mathcal{S}_h$ responds with a single value from $\mathbb{F}_q$, which does not affect the asymptotic communication of the protocol.

In the protocol above there is an obvious imbalance between the communication from the user to the servers and vice versa. The next theorem extends the technique of Theorem 3 to fix this imbalance and obtain a better communication complexity.

Theorem 4 *There exists a 1-private k-server PIR protocol with communication complexity $O(k^2 \log k \, n^{1/(2k-1)})$.*

Protocol description : We use the standard mathematical notation $\left.\frac{\partial F}{\partial z_l}\right|_Q$ to denote the value of the partial derivative of F with respect to z_l at point Q. Let $\lambda_1, \ldots, \lambda_k \in \mathbb{F}_q$ be

[2] The Hamming weight of a vector is defined to be the number of nonzero coordinates.

distinct and nonzero. Set $d = 2k - 1$. Let $P = E(i)$. The user wants to retrieve $F(P)$.

$$\begin{array}{lcl} \mathcal{U} & : & \text{Picks } V \in \mathbb{F}_q^m \text{ uniformly at random.} \\ \mathcal{U} \rightarrow \mathcal{S}_h & : & P + \lambda_h V \\ \mathcal{U} \leftarrow \mathcal{S}_h & : & F(P + \lambda_h V), \\ & & \left.\frac{\partial F}{\partial z_1}\right|_{P+\lambda_h V}, \ldots, \left.\frac{\partial F}{\partial z_m}\right|_{P+\lambda_h V} \end{array}$$

Privacy : The proof of privacy is identical to the proof from Theorem 3.

Correctness : Again, consider the line $L = \{P + \lambda V \mid \lambda \in \mathbb{F}_q\}$. Let $f(\lambda) = F(P + \lambda V)$ be the restriction of F to L. Clearly, $f(\lambda_h) = F(P + \lambda_h V)$. Thus the user knows the values $\{f(\lambda_h)\}$ for all $h \in [k]$. However, this time the values $\{f(\lambda_h)\}$ do not suffice to reconstruct the polynomial f, since the degree of f may be up to $2k - 1$. The main observation underlying our protocol is that knowing the values of partial derivatives $\left.\frac{\partial F}{\partial z_1}\right|_{P+\lambda_h V}, \ldots, \left.\frac{\partial F}{\partial z_m}\right|_{P+\lambda_h V}$, the user can reconstruct the value of $f'(\lambda_h)$. The proof is a straightforward application of the chain rule:

$$\left.\frac{\partial f}{\partial \lambda}\right|_{\lambda_h} = \left.\frac{\partial F(P + \lambda V)}{\partial \lambda}\right|_{\lambda_h} = \sum_{l=1}^{m} \left.\frac{\partial F}{\partial z_l}\right|_{P+\lambda_h V} V_l.$$

Thus the user can reconstruct $\{f(\lambda_h)\}$ and $\{f'(\lambda_h)\}$ for all $h \in [k]$. Combining this observation with Lemma 2, we conclude that user can reconstruct f and obtain $F(P) = f(0)$.

Complexity : The user sends each of k servers a length-m vector of values in $\mathbb{F}_q$. Servers respond with length-$(m + 1)$ vectors of values in $\mathbb{F}_q$. Recall that $m = O(dn^{1/d})$ and $q \leq 2k$. Thus the total communication is $O(k^2 \log k\, n^{1/(2k-1)})$.

3.1 Application to Robust PIR

We review the definition of robust PIR [9].

Definition 5 *A k-out-of-l PIR protocol is a PIR protocol with the additional property that the user always computes the correct value of x_i from any k out of l of the answers.*

As noted in [9], robust PIR has applications to servers which may hold different versions of a database, as long as some k have the latest version and there is a way to distinguish these k. Another application is to servers with varying response times. Here we improve the two bounds $2^{\tilde{O}(k)} n^{\frac{2 \log\log k}{k \log k}} l \log l$ and $O(kn^{1/k} l \log l)$ given in [9].

Indeed, in the protocol above, if for l servers we set the field size $q > l$ and the degree $\deg F = 2k - 1$, then from any k servers' answers, we can reconstruct f as before. We conclude

Theorem 6 *There exists a k-out-of-l robust PIR with communication $O(kn^{1/(2k-1)} l \log l)$.*

4 PIR on the curve

Theorem 7 *There exists a t-private k-server PIR protocol with communication complexity $O\left(\frac{k^2}{t} \log k\, n^{1/\lfloor\frac{2k-1}{t}\rfloor}\right)$.*

Protocol description : Again, consider $\mathbb{F}_q$, where $k < q \leq 2k$ and let $\lambda_1, \ldots, \lambda_k \in \mathbb{F}_q$ be distinct and nonzero. Set $d = \lfloor\frac{2k-1}{t}\rfloor$. Let $P = E(i)$. The user wants to retrieve $F(P)$.

$$\begin{array}{lcl} \mathcal{U} & : & \text{Randomly picks } V^1, \ldots, V^t \in \mathbb{F}_q^m. \\ \mathcal{U} \rightarrow \mathcal{S}_h & : & Q^h \overset{\text{def}}{=} P + \lambda_h V^1 + \lambda_h^2 V^2 + \ldots + \lambda_h^t V^t \\ \mathcal{U} \leftarrow \mathcal{S}_h & : & F(Q^h),\ \left.\frac{\partial F}{\partial z_1}\right|_{Q^h}, \ldots, \left.\frac{\partial F}{\partial z_m}\right|_{Q^h} \end{array}$$

Privacy : We need to show that for every $T \subseteq [k]$, where $|T| \leq t$; the collusion of servers $\{\mathcal{S}_h\}_{h \in T}$ learns no information about the point $P = E(i)$. The joint input of servers $\{\mathcal{S}_h\}_{h \in T}$ is $\{P + \lambda_h V^1 + \ldots + \lambda_h^t V^t\}_{h \in T}$. Since the coordinates are shared independently, it suffices to show that for each $l \in [m]$ and $V_l^j \in \mathbb{F}_q$ chosen independently and uniformly at random; the values $\{P_l + \lambda_h V_l^1 + \ldots + \lambda_h^t V_l^t\}_{h \in T}$ disclose no information about P_l. The last statement is implied by the properties of Shamir's secret sharing scheme [17].

Correctness : Consider the curve $\chi = \{P + \lambda V^1 + \ldots + \lambda^t V^t \mid \lambda \in \mathbb{F}_q\}$. Let $f(\lambda) = F(P + \lambda V^1 + \ldots + \lambda^t V^t)$ be the restriction of F to χ. Obviously, f is a univariate polynomial of degree at most $2k - 1$. By definition, we have $f(\lambda_h) = F(Q^h)$; thus $\mathcal{U}$ knows the values $\{f(\lambda_h)\}$ for all $h \in [k]$. Now we shall see how knowing the values of partial derivatives $\left.\frac{\partial F}{\partial z_1}\right|_{Q^h}, \ldots, \left.\frac{\partial F}{\partial z_m}\right|_{Q^h}$ $\mathcal{U}$ reconstructs the value of $f'(\lambda_h)$. Again, the reconstruction is a straightforward application of the chain rule:

$$\begin{aligned} \left.\frac{\partial f}{\partial \lambda}\right|_{\lambda_h} &= \left.\frac{\partial F(P + \lambda V^1 + \ldots + \lambda^t V^t)}{\partial \lambda}\right|_{\lambda_h} \\ &= \sum_{l=1}^{m} \left.\frac{\partial F}{\partial z_l}\right|_{Q^h} \left.\frac{\partial}{\partial \lambda}(P_l + \lambda V_l^1 + \ldots + \lambda^t V_l^t)\right|_{\lambda_h} \end{aligned}$$

Thus $\mathcal{U}$ can reconstruct $\{f(\lambda_h)\}$ and $\{f'(\lambda_h)\}$ for all $h \in [k]$. Combining this observation with Lemma 2, we conclude that the user can reconstruct f and obtain $F(P) = f(0)$.

Complexity : As in the protocol of Theorem 4, $\mathcal{U}$ sends each of k servers a length-m vector of values in $\mathbb{F}_q$ and servers respond with length-$(m + 1)$ vectors of values in $\mathbb{F}_q$. Here $m = O(dn^{1/d})$ and $q \leq 2k$. Thus the total communication is $O\left(\frac{k^2}{t} \log k\, n^{1/\lfloor\frac{2k-1}{t}\rfloor}\right)$.

4.1 Application to Instance Hiding

As noted in the introduction, in the instance-hiding scenario [2, 3] there is a function $F_m : \{0,1\}^m \to \{0,1\}$ held by $\frac{m}{c \log m}$ oracles for some constant c. The user has a point $P \in \{0,1\}^m$ and should learn $F_m(P)$. Further, the view of up to t oracles should be independent of P. We have the following improvement upon the best known $\tilde{O}(m^{ct/2+2+t})$ bound of [14].

Theorem 8 *There exists a t-private non-adaptive oracle instance-hiding scheme with communication and computation $\tilde{O}(m^{ct/2+2})$, where $\tilde{O}(f) \overset{\mathrm{def}}{=} O(f \log^{O(1)} f)$.*

Proof: Using the above protocol on the truth table of F_m, the communication is

$$O\left(\frac{k^2}{t} \log k n^{1/\lfloor (2k-1)/t \rfloor}\right) =$$

$$\tilde{O}\left(m^2 \cdot (2^m)^{(\lfloor (2k-1)/t \rfloor)^{-1}}\right) = \tilde{O}(m^{ct/2+2}).$$

It is also easy to see that $\mathcal{U}$ runs in time which is quasilinear in the communication. ■

5 PIR with preprocessing

To measure the efficiency of an algorithm with preprocessing, we use the definition of *work* in [8] which counts the number of precomputed and database bits that need to be read in order to respond to a query. The goal of this section is to prove the following theorem.

Theorem 9 *There exists a 2-server PIR protocol with $O(n^{1/3})$ communication, $\mathrm{poly}(n)$ preprocessing, and $O(n/\log^r)$ server work for any constant r.*

We need a lemma about preprocessing polynomials $F \in \mathbb{F}_p[z_1, \ldots, z_m]$. We assume the number of variables m is tending to infinity, while the degree of F is always constant. The lemma is similar to Theorem 3.1 of [8]. The main idea is to write the input polynomial F as a sum of $\mathrm{poly}(m)$ different polynomials over disjoint monomials. We do this so that each summand polynomial G involves only a logarithmic number of variables, and thus we can precompute G on all possible assignments to its variables. As the different G are over disjoint monomials, to evaluate $F(V)$ we simply read one precomputed answer for each G, and sum them up.

Lemma 10 *Let F be a homogeneous degree-d polynomial in $\mathbb{F}_p[z_1, \ldots, z_m]$. Using $\mathrm{poly}(m)$ preprocessing time, for all $V \in \mathbb{F}_p^m$, $F(V)$ can be computed with $O(m^d/\log^d m)$ work.*

Proof: Partition $[m]$ into $\alpha = m/\log m$ disjoint sets $D_1, \ldots, D_\alpha$ of size $\log m$. For every sequence $1 \leq t_1, \ldots, t_d \leq \alpha$, let $F_{D_{t_1}, \ldots, D_{t_d}}$ denote the sum of all monomials of F of the form $c z_{i_1} \cdots z_{i_d}$ for some $c \in \mathbb{F}_p$ and $i_1 \in D_{t_1}, \ldots, i_d \in D_{t_d}$. The following is the preprocessing algorithm.

```
Preprocess(F):
1. For each polynomial $F_{D_{t_1},\ldots,D_{t_d}}$,
    (a) Evaluate $F_{D_{t_1},\ldots,D_{t_d}}$ on all $W \in \mathbb{F}_p^m$
        for which $\mathrm{Supp}(W) \in \cup_i D_{t_i}$.
```

Time Complexity: There are $\alpha^d = (m/\log m)^d$ polynomials $F_{D_{t_1}, \ldots, D_{t_d}}$. For each polynomial, there are at most $p^{d \log m} = \mathrm{poly}(m)$ different W whose support is in $\cup_i D_{t_i}$. Thus the algorithm needs only $\mathrm{poly}(m)$ preprocessing time.

For a set $S \subseteq [m]$, let $V|_S$ denote the point $V' \in \mathbb{F}_p^m$ with $V'_j = V_j$ for $j \in S$ and $V'_j = 0$ otherwise. The following describes how to compute $F(V)$.

```
Evaluate(F, V):
1. $\sigma \leftarrow 0$.
2. For each polynomial $F_{D_{t_1},\ldots,D_{t_d}}$,
    (a) $\sigma \leftarrow \sigma + F_{D_{t_1},\ldots,D_{t_d}}(V|_{\cup_i D_{t_i}})$.
3. Output $\sigma$.
```

Correctness: Immediate from

$$F(V) = \sum_{t_1, \ldots, t_d} F_{D_{t_1}, \ldots, D_{t_d}}(V|_{\cup_i D_{t_i}}).$$

Work: The sum is over $\alpha^d = (m/\log m)^d$ polynomials $F_{D_{t_1}, \ldots, D_{t_d}}$, each with a precomputed answer, and thus the total work is $O(m^d/\log^d m)$. ■

5.1 Two server protocol

We start with the intuition underlying our two server preprocessing protocol. Suppose the servers were to represent the database as a degree-d polynomial F in $m = \Theta(n^{1/d})$ variables, where $d = 2r + 1$ is an arbitrary odd constant. Proceeding as in the protocol of section 3, the user sends each server a point on a random line L through his point of interest. To reconstruct $F|_L$, the user needs the evaluation of F on his query points, together with all partial derivatives of F up to order r. The observation is that each partial derivative computed by the servers is a polynomial of degree at least $d - r = r + 1$ in at most m variables, and therefore we can apply Lemma 10 to achieve low server work.

However, while the user is only sending $O(m)$ bits to the servers, the servers' answers are of size $O(m^r)$. To fix this, we use a balancing technique similar to that in [10]. Each server partitions the database into t databases F_j of size n/t, for some parameter t. Each database will be represented as a degree d polynomial in $m = O((n/t)^{1/d})$ variables. The user sends t points to each server, one for each database. Suppose the user wants $F_u(P)$. For the $t-1$ databases F_j, $j \neq u$, that the user doesn't care about, he sends random V^j and $-V^j$ to servers 1 and 2 respectively. On the other hand, for the database F_u that he cares about, he proceeds as in the protocol of section 3. The servers compute the lists of partial derivatives for each database, as before, but instead of sending them back, they send the *sum* of each partial derivative over all t databases. We show this information is sufficient for the user to reconstruct $F_u(P)$. The total work will be $O(n/\log^{r+1} n)$, and by carefully choosing t, we can keep the communication at $O(n^{1/3})$.

Consider a prime[3] field $\mathbb{F}_p$ for some $\max(2,r) < p < 2\max(2,r)$. Such a prime p exists by the Bertrand's Postulate [16]. $\mathcal{S}_1$ and $\mathcal{S}_2$ preprocess as follows.

```
Preprocessing phase($x$):
1. $s \leftarrow \frac{r-1}{3r}$, $t \leftarrow n^s$.
2. Partition $x$ into $t$ databases $\mathrm{DB}_1, \ldots, \mathrm{DB}_t$, each
   containing $n^{1-s}$ elements.
3. Represent $\mathrm{DB}_j$ as a homogeneous polynomial $F_j$
   of degree $d = 2r+1$ with $m = O\left(n^{(1-s)/d}\right)$ vars.
4. For $a = 0, \ldots, r$, for $j \in [t]$, and for $l_1, \ldots, l_a \in [m]$,
   compute Preprocess$\left(\frac{\partial^a F_j}{\partial z_{l_1} \cdots \partial z_{l_a}}\right)$.
```

Let DB_u be the database containing x_i. Assume the user wants $F_u(P)$. Let $\delta_{\alpha,\beta}$ be 1 if $\alpha = \beta$, and 0 otherwise.

$\mathcal{U}$	:	Randomly picks $V^1, \ldots, V^t \in \mathbb{F}_p^m$.
$\mathcal{U} \rightarrow \mathcal{S}_h$	:	For $j \in [t]$, $Q^{h,j} \overset{\text{def}}{=} (-1)^{h+1} V^j + \delta_{j,u} P$
$\mathcal{U} \leftarrow \mathcal{S}_h$	:	$\forall a \in \{0, \ldots, r\}$ and $l_1, \ldots, l_a \in [m]$, $\sum_{j=1}^t \left.\frac{\partial^a F_j}{\partial z_{l_1} \ldots \partial z_{l_a}}\right\|_{Q^{h,j}} =$ $\sum_{j=1}^t$ Evaluate $\left(\frac{\partial^a F_j}{\partial z_{l_1} \ldots \partial z_{l_a}}, Q^{h,j}\right)$

Correctness: Since d is odd, for all V

$$\left.\frac{\partial^a F_j}{\partial z_{l_1} \cdots \partial z_{l_a}}\right|_{-V} = (-1)^{a+1} \left.\frac{\partial^a F_j}{\partial z_{l_1} \cdots \partial z_{l_a}}\right|_{V}$$

It follows that for all a and all $j \neq u$,

$$\sum_{l_1,\ldots,l_a} \left.\frac{\partial^a F_j}{\partial z_{l_1} \cdots \partial z_{l_a}}\right|_{V^j} + \sum_{l_1,\ldots,l_a} \left.\frac{(-1)^a \partial^a F_j}{\partial z_{l_1} \cdots \partial z_{l_a}}\right|_{-V^j} = 0.$$

Put $f(\lambda) = (F_u)|_{P+\lambda V^u}$, and define $g(\lambda) = f(\lambda) + f(-\lambda)$. We have

$$\begin{aligned} &\sum_j \sum_{l_1,\ldots,l_a} \left.\frac{\partial^a F_j}{\partial z_{l_1} \cdots \partial z_{l_a}}\right|_{Q^{1,j}} V^u_{l_1} \cdots V^u_{l_a} \\ &\qquad + \left.\frac{(-1)^a \partial^a F_j}{\partial z_{l_1} \cdots \partial z_{l_a}}\right|_{Q^{2,j}} V^u_{l_1} \cdots V^u_{l_a} \\ = \; &\sum_{l_1,\ldots,l_a} \left.\frac{\partial^a F_u}{\partial z_{l_1} \cdots \partial z_{l_a}}\right|_{P+V^u} V^u_{l_1} \cdots V^u_{l_a} \\ &\qquad + \left.\frac{(-1)^a \partial^a F_u}{\partial z_{l_1} \cdots \partial z_{l_a}}\right|_{P-V^u} V^u_{l_1} \cdots V^u_{l_a} \\ = \; & f^{(a)}(1) + (-1)^a f^{(a)}(-1) = g^{(a)}(1). \end{aligned}$$

Thus $\mathcal{U}$ can compute $g(1), g^{(1)}(1), \ldots, g^{(r)}(1)$ from the answers. Since every monomial of g has even degree, for $\gamma = \lambda^2$ we can define $h(\gamma) = g(\lambda)$ for a degree-r polynomial h. Using that

$$\frac{dg}{d\lambda} = \frac{dh}{d\gamma} \cdot \frac{d\gamma}{d\lambda} = 2\lambda \frac{dh}{d\gamma},$$

a simple induction shows that from $g^{(0)}(1), \ldots, g^{(r)}(1)$, $\mathcal{U}$ can compute $h^{(0)}(1), \ldots, h^{(r)}(1)$. The claim is that these values determine h. Indeed, if $h_1 \neq h_2$ agree on these values, then by lemma 1

$$(\gamma - 1)^{r+1} \mid (h_1 - h_2),$$

which contradicts that $h_1 - h_2$ has degree at most r. Hence the user obtains $h(0) = g(0) = 2f(0) = 2F(P)$, and thus $F(P)$ since the characteristic $p > 2$.

Privacy: Since the V^j are independent and uniformly random, so are the $Q^{1,j}$ and the $Q^{2,j}$. Thus the view of each of $\mathcal{S}_1, \mathcal{S}_2$ is independent of P.

Communication: $\mathcal{U}$ sends $O(tm) = O(n^{s+(1-s)/(2r+1)}) = O(n^{(r-1)/(3r)+1/(3r)}) = O(n^{1/3})$ bits. $\mathcal{S}_1, \mathcal{S}_2$ respond with $O(m + m^2 + \cdots + m^r) = O(m^r) = O(n^{(1-s)r/(2r+1)}) = O(n^{1/3})$ bits.

Server Work: Notice that the work is dominated by the calls to Evaluate. For any $a \in \{0, \ldots, r\}$, any $l_1, \ldots, l_a \in [m]$, and any $j \in [t]$, the polynomial $\frac{\partial^a F_j}{\partial z_{l_1} \cdots \partial z_{l_a}}$ is either 0 or has degree $2r+1-a$, and at most m variables. Thus for any V, Evaluate$(\frac{\partial^a F_j}{\partial z_{l_1} \cdots \partial z_{l_a}}, V)$ can be computed in $O(m^{2r+1-a}/\log^{2r+1-a} m)$ time. As the number of such $\frac{\partial^a F_j}{\partial z_{l_1} \cdots \partial z_{l_a}}$ is $O(m^a)$, it follows that the time for all calls to Evaluate per DB is

$$\sum_a O\left(\frac{m^a m^{2r+1-a}}{\log^{2r+1-a} m}\right) = \frac{O(m^d)}{\log^{r+1} m} = \frac{O(n^{1-s})}{\log^{r+1} n}.$$

Thus the total work over all n^s DBs is $O(n/\log^{r+1} n)$.

[3] In sections 5 and 6 we base our protocols on prime fields $\mathbb{F}_p$ and do not consider general finite fields $\mathbb{F}_q$. We do this to avoid issues related to subtle properties of derivatives of orders greater than one in finite fields of small characteristic. Another possible solution to this problem is to use Hasse derivatives (referred to as hyperderivatives in [15]) instead of usual derivatives. This allows for protocols over arbitrary finite fields.

5.2 Application to PIR with fixed answer sizes

In [4] it is asked:

> *For two-server PIR protocols and for constant b, if the answers have size at most b, can the queries have size less than n/b?*

We answer this with the following theorem.

Theorem 11 *For any $b = O(n^{1/3})$, there exists a 2 server PIR protocol with answer length $O(b)$ and query length $O(n/b^2)$, where the constant in the big-Oh is independent of n and b.*

Proof: Before the protocol begins, $\mathcal{S}_1$ and $\mathcal{S}_2$ partition x into $t = O(n/b^3)$ databases $\mathrm{DB}_1, \ldots, \mathrm{DB}_t$ each of size $O(b^3)$. Each such DB is a degree-3 polynomial in $m = O(b)$ variables. Let DB_u be the database containing x_i. The protocol follows.

$\mathcal{U}$	:	Randomly picks $V^1, \ldots, V^t \in \mathbb{F}_p^m$.
$\mathcal{U} \to \mathcal{S}_h$	:	For $j \in [t]$, $Q^{h,j} \overset{\text{def}}{=} (-1)^{h+1}V^j + \delta_{j,u}P$
$\mathcal{U} \leftarrow \mathcal{S}_h$	:	$\sum_j F_j(Q^{h,j})$, and $\forall l \in [m]$, $\sum_j \left.\frac{\partial F_j}{\partial z_l}\right\|_{Q^{h,j}}$

The correctness follows from the correctness of our preprocessing protocol for $r = 1$. For the communication, $\mathcal{U}$ sends $tm = O(n/b^2)$ bits, and $\mathcal{S}_1$ and $\mathcal{S}_2$ each respond with $O(b)$ bits. ■

6 Recursive PIR in the space

Assume k is constant. The best known upper bound of $n^{O\left(\frac{\log\log k}{k \log k}\right)}$ for the communication complexity of 1-private k server PIR protocols is due to Beimel *et al.* [7]. Although their proof is elementary, it is rather complicated and hard to follow. The key theorem of [7] is:

Theorem 12 *([7] Theorem 3.5) Suppose there is a 1-private PIR protocol $\mathcal{P}$ with communication complexity $C_{\mathcal{P}}(n,k)$. Let d, λ, k' be positive integers (which may depend on k) such that $k' < k$ and $d \le (\lambda+1)k - (\lambda-1)k' + (\lambda - 2)$. Then there is a 1-private PIR protocol $\mathcal{P}'$ with communication complexity*

$$C_{\mathcal{P}'}(n,k) = O\left(n^{1/d} + \sum_{l=k'}^{k} \binom{k}{l} C_{\mathcal{P}}(n^{\lambda l/d}, l)\right).$$

Recursive applications of Theorem 12 starting from a 2-server protocol with communication complexity $O(n^{1/3})$ yield the best known upper bounds for 1-private PIR. In this section we present an alternative geometric proof of the special case of Theorem 12 that corresponds to setting the value of parameter $\lambda = 2$. This case is sufficient to obtain 1-private PIR protocols with communication complexity matching the results of [7] for all values of $k < 26$, where the bound on k was determined experimentally.

Theorem 13 *Suppose there is a 1-private PIR protocol $\mathcal{P}$ with communication complexity $C_{\mathcal{P}}(n,k)$. Let d, k' be positive integers such that $k' < k$ and $d \le 3k - k'$. Then there is a 1-private PIR protocol $\mathcal{P}'$ with communication complexity*

$$C_{\mathcal{P}'}(n,k) = O\left(n^{1/d} + \binom{k}{k'} C_{\mathcal{P}}(n^{2k'/d}, k')\right).$$

It may seem that the bound of Theorem 13 improves upon the bound of Theorem 12 since there are no terms corresponding to values of $l \in [k'+1, k]$. However this is not a real improvement, since the original proof of Theorem 12 can also be modified to eliminate these terms.

We start with a high-level view of our protocol. $\mathcal{U}$ wants to retrieve the value $F(P)$. To this end $\mathcal{U}$ randomly selects a k' dimensional affine subspace[4] $\pi(L)$ containing the point P and sends each server $\mathcal{S}_h$ a $(k'-1)$ dimensional affine subspace[5] $\pi(L_h) \subseteq \pi(L)$. Each $\mathcal{S}_h$ replies with values and derivatives of the polynomial F at every point of $\pi(L_h)$. We assume the subspaces $\pi(L_h)$ are in general position. In particular this implies that for every set T of k' servers there is a unique point $P^T = \bigcap_{h \in T} \pi(L_h)$ that is known to all of them. For each subset T of k' servers $\mathcal{U}$ *runs a separate k'-server 1-private PIR protocol* to obtain the value of a $2k'$-th partial derivative of the function F at point P^T in the direction towards the point P. Finally we demonstrate that the information about F obtained by $\mathcal{U}$ suffices to reconstruct the restriction of F to $\pi(L)$.

6.1 Preliminaries

In what follows we work in a prime field $\mathbb{F}_p$ with $\max(2k', k, d) < p$. We start with some notation. Let $\{\alpha_h\}_{h \in [k]}$ be distinct and nonzero elements of $\mathbb{F}_p$. For $h \in [k]$ let

$$g_h(\lambda_1, \ldots, \lambda_{k'}) \overset{\text{def}}{=} \alpha_h \lambda_1 + \alpha_h^2 \lambda_2 + \ldots + \alpha_h^{k'} \lambda_{k'} - 1.$$

Let $L = \mathbb{F}_p^{k'}$ be a k' dimensional affine space over $\mathbb{F}_p$. Consider the hyperplanes $L_h \subseteq L$:

$$L_h \overset{\text{def}}{=} \{(\lambda_1, \ldots, \lambda_{k'}) \mid g_h(\lambda_1, \ldots, \lambda_{k'}) = 0\}$$

[4] We use the complicated notation $\pi(L)$ for consistency with the actual proof.

[5] In certain degenerate cases the dimensions of both $\pi(L)$ and $\pi(L_h)$ may in fact be smaller than k' and $k' - 1$.

The properties of the Vandermonde matrix imply that for any $T \subseteq [k]$, where $|T| \leq k'$, the hyperplanes $\{L_h\}_{h \in T}$ are in general position, i.e.:

$$\dim \bigcap_{h \in T} L_h = k' - |T|. \tag{1}$$

For $T \subseteq [k]$, such that $|T| = k'$, let Q^T denote the unique intersection point of $\{L_h\}_{h \in T}$. I.e:

$$Q^T \overset{\text{def}}{=} \bigcap_{h \in T} L_h.$$

Consider a certain hyperplane L_h and a vector $v \in \mathbb{F}_p^{k'}$. We say that vector $v = (v_1, \ldots, v_{k'})$ is *off the hyperplane* L_h if $\alpha_h v_1 + \alpha_h^2 v_2 + \ldots + \alpha_h^{k'} v_{k'} \neq 0$. Clearly, for every hyperplane L_h there exists a vector $v \in \mathbb{F}_p^{k'}$ that is off L_h.

Consider the map $\pi : L \rightarrow \mathbb{F}_p^m$ induced by a uniformly random choice of $\{V^j\}_{j \in [k']} \subseteq \mathbb{F}_p^m$ for a fixed $P \in \mathbb{F}_p^m$:

$$\pi(\lambda_1, \ldots, \lambda_{k'}) \overset{\text{def}}{=} P + \lambda_1 V^1 + \ldots + \lambda_{k'} V^{k'}.$$

Let P^T denote the image of Q^T under π, i.e.:

$$P^T \overset{\text{def}}{=} \pi(Q^T).$$

In the remaining part of this subsection we establish two geometric lemmas. The first lemma concerns the non-recursive part of our protocol.

Lemma 14 *Let $f \in \mathbb{F}_p[\lambda_1, \ldots, \lambda_{k'}]$, $\deg f < |\mathbb{F}_p|$ and $h \in [k]$. Suppose $f|_{L_h} = 0$ and $\left.\frac{\partial f}{\partial v}\right|_{L_h} = 0$, where v is off L_h. Then $g_h^2 \mid f$.*

Proof: The fact that $g_h \mid f$ is a direct consequence of Bézout's theorem ([13] p. 53)[6]. To see that g_h divides f twice, let $f = g \cdot g_h$. By the chain rule,

$$\frac{\partial f}{\partial v} = \frac{\partial g}{\partial v}\, g_h + g \sum_i \alpha_h^i v_i,$$

and since v is off of L_h, $\sum_j \alpha_h^j v_j \neq 0$. Restricting both sides to L_h, the premise of the lemma implies $0 = g|_{L_h}$, and another application of Bézout's theorem gives $g_h \mid g$, which proves the lemma. ■

The next lemma concerns the recursive part of our protocol.

Lemma 15 *Let $f \in \mathbb{F}_p[\lambda_1, \ldots, \lambda_{k'}]$. Assume $T \subseteq [k]$, $|T| = k'$. Suppose $f = g \prod_{h \in T} (g_h)^2$ and $v \in \mathbb{F}_p^{k'}$ is off every $\{L_h\}_{h \in T}$; then*

$$\left.\frac{\partial^{2k'} f}{\partial v^{2k'}}\right|_{Q^T} = C \cdot g(Q^T),$$

where $C \neq 0$ is some constant that depends only on $\{g_h\}_{h \in T}$.

Proof: Let $C_h = \frac{\partial g_h}{\partial v} = \sum_j \alpha_h^j v_j$, and observe that $C_h \neq 0$ since v is off of L_h. By repeated application of the chain rule,

$$\left.\frac{\partial^a \left(\prod_{h \in T} (g_h)^2\right)}{\partial v^a}\right|_{Q^T} = \delta_{a,2k'} (2k')! \prod_{h \in T} C_h^2,$$

where $\delta_{\alpha,\beta}$ is 1 if $\alpha = \beta$ and 0 otherwise. Again by the chain rule,

$$\left.\frac{\partial^{2k'} f}{\partial v^{2k'}}\right|_{Q^T} = g(Q^T) \cdot (2k')! \prod_{h \in T} C_h^2.$$

The lemma follows by setting $C = (2k')! \prod_h C_h^2$. ■

6.2 The protocol

Protocol description : As usual the database is represented by a degree d polynomial in $m = O(dn^{1/d})$ variables. Recall that $d \leq 3k - k'$. Therefore we can treat d as a constant. Let $P = E(i)$. The user wants to retrieve $F(P)$. Our protocol is one-round. However (as in the work of [7]) it is convenient to think about the protocol as several executions of PIR protocols that take place in parallel. $\mathcal{U}$ sends servers the affine spaces $\pi(L_h)$. Each server returns the values of F on $\pi(L_h)$ and the values of all first order partial derivatives of F on $\pi(L_h)$. Moreover, $\mathcal{U}$ runs a separate PIR protocol with every group T of k' servers to obtain the value $\left.\frac{\partial^{2k'} F}{\partial (P - P^T)^{2k'}}\right|_{P^T}$. Below is the formal description of the protocol. Here $\mathcal{S}_T$ denotes the set of servers $\{\mathcal{S}_h\}_{h \in T}$.

$$\boxed{\begin{array}{ll}
\mathcal{U} & : \text{ Picks a random } \pi : L \rightarrow \mathbb{F}_p^m, \\
 & \quad \pi(\lambda_1, \ldots, \lambda_{k'}) = P + \lambda_1 V^1 + \ldots + \lambda_{k'} V^{k'} \\
\mathcal{U} \rightarrow \mathcal{S}_h & : \pi(L_h) \\
\mathcal{U} \leftarrow \mathcal{S}_h & : F|_{\pi(L_h)}, \left.\frac{\partial F}{\partial z_1}\right|_{\pi(L_h)}, \ldots, \left.\frac{\partial F}{\partial z_m}\right|_{\pi(L_h)} \\
\mathcal{U} \leftrightarrow \mathcal{S}_T & : \text{A } k'\text{-server PIR subprotocol for} \\
 & \quad \text{retrieving the value of } \left.\frac{\partial^{2k'} F}{\partial (P - P^T)^{2k'}}\right|_{P^T}
\end{array}}$$

To complete the description of the protocol, we need the following lemma.

[6] More formally, we have a polynomial f that vanishes on every $\mathbb{F}_p$-point of a hyperplane L_h. This implies that f vanishes on every $\overline{\mathbb{F}}_p$-point of L_h, since $|\mathbb{F}_p| > \deg f$. Now, once we have passed to the algebraically closed field $\overline{\mathbb{F}}_p$, we can apply Bézout's theorem to conclude that g_h and f have a common factor, and therefore $g_h \mid f$.

Lemma 16 *Let $F(z_1, \ldots, z_m)$ be an m-variate polynomial of degree d, where d is a constant. Assume $P = E(i) \in \mathbb{F}_p^m$ is a point of Hamming weight d. Let $T \subseteq [k]$, $|T| = k'$. Suppose each of the servers $\{S_h\}_{h\in T}$ knows the point P^T; then $\mathcal{U}$ can learn the value of the directional derivative*

$$\left.\frac{\partial^s F}{\partial(P - P^T)^s}\right|_{P^T}$$

privately (with respect to i) with communication complexity $O(C_{\mathcal{P}}(m^s, k'))$.

Proof: We have

$$\left.\tfrac{\partial^s F}{\partial(P-P^T)^s}\right|_{P^T} = \sum_{l_1,\ldots,l_s} \left.\tfrac{\partial^s F}{\partial z_{l_1}\cdots\partial z_{l_s}}\right|_{P^T} (P - P^T)_{l_1} \cdots (P - P^T)_{l_s}, \qquad (2)$$

and since P^T and F are known to all S_h with $h \in T$, these servers can interpret the RHS of equation (2) as an m-variate degree-s polynomial G in the ring $\mathbb{F}_p[P_1, \ldots, P_m]$. Since $\deg G = s$ and the Hamming weight of P is d, at most $2^d = O(1)$ monomials M of G are nonzero on P. Thus, to learn $G(P)$ it is enough for $\mathcal{U}$ to learn the coefficients of these M. To this end, $\mathcal{U}$ and these servers run a PIR protocol on the list of coefficients of monomials $M = P_{i_1} \cdots P_{i_d}$ for $1 \leq i_1, \ldots, i_d \leq m$. The complexity is therefore $2^d \mathcal{C}_{\mathcal{P}}(O(m^s), k') = O(\mathcal{C}_{\mathcal{P}}(m^s, k'))$. ■

We now show the desired properties of our protocol.

Privacy : Since the subprotocols are independent, and $\mathcal{P}$ is private by assumption (recall the condition of theorem 13), in order to show that $\mathcal{P}'$ is private it suffices to show privacy at the top level of recursion. In this level S_h's view is

$$\begin{aligned}\pi(L_h) = \quad & \{P + \lambda_1 V^1 + \ldots + \lambda_{k'} V^{k'} \mid \\ & \lambda_j \in \mathbb{F}_p,\ \alpha_h \lambda_1 + \ldots + \alpha_h^{k'} \lambda_{k'} = 1\}.\end{aligned}$$

Observe that any point in $\pi(L_h)$ is some linear combination (over $\mathbb{F}_p$) of the points

$$P + (\alpha_h)^{-1} V^1, \ldots, P + (\alpha_h^{k'})^{-1} V^{k'} \in \pi(L_h).$$

Thus S_h's view can be generated from these points. But as distributions,

$$\begin{aligned}(P + (\alpha_h)^{-1} V^1, \ldots, P + (\alpha_h^{k'})^{-1} V^{k'}) \equiv \\ (R^1, \ldots, R^{k'}),\end{aligned}$$

where the $R^j \in \mathbb{F}_p^m$ are independent and uniformly random. Thus $\mathcal{S}_h$'s view does not depend P.

Correctness: Let $f \overset{\text{def}}{=} F(\pi(\lambda_1, \ldots, \lambda_{k'}))$ denote the restriction of F to $\pi(L)$. We show the information that $\mathcal{U}$ obtains from $\{\mathcal{S}_h\}_{h\in[k]}$ suffices to reconstruct f.

Information about F translates into information about f:

1. For $h \in [k]$, $f|_{L_h} = F|_{\pi(L_h)}$, so $\mathcal{U}$ can compute the values of f along every L_h.
2. Now let $h \in [k]$. Let $v_h \in \mathbb{F}_p^{k'}$ be a vector that is off the hyperplane L_h. We show how to compute $\left.\frac{\partial f}{\partial v_h}\right|_{L_h}$ from $\left.\frac{\partial F}{\partial z_1}\right|_{\pi(L_h)}, \ldots, \left.\frac{\partial F}{\partial z_m}\right|_{\pi(L_h)}$. From the chain rule

   $$\left.\tfrac{\partial f}{\partial v_h}\right|_{L_h} = \left.\tfrac{\partial F(\pi(\lambda_1,\ldots,\lambda_{k'}))}{\partial v_h}\right|_{L_h} = \sum_{l=1}^{m} \left.\tfrac{\partial F}{\partial z_l}\right|_{\pi(L_h)} \tfrac{\partial}{\partial v_h}(P_l + \lambda_1 V_l^1 + \ldots + \lambda_{k'} V_l^{k'})\Big|_{L_h}.$$

   Thus for every $h \in [k]$, $\mathcal{U}$ can compute values of $\frac{\partial f}{\partial v_h}$ at every point of L_h.
3. Finally, let $T \subseteq [k]$ be such that $|T| = k'$. Let $\pi_l(\lambda_1, \ldots, \lambda_{k'})$ denote $P_l + \lambda_1 V_l^1 + \ldots + \lambda_{k'} V_l^{k'}$ for $l \in [m]$. We have

   $$\left.\tfrac{\partial^{2k'} f}{\partial(-Q^T)^{2k'}}\right|_{Q^T} = \left.\tfrac{\partial^{2k'} F(\pi(\lambda_1,\ldots,\lambda_{k'}))}{\partial(-Q^T)^{2k'}}\right|_{Q^T} = \sum_{l_1,\ldots,l_{2k'}} \left.\tfrac{\partial^{2k'} F}{\partial z_{l_1}\ldots\partial z_{l_{2k'}}}\right|_{P^T} \textstyle\prod_{j=1}^{2k'}(P_{l_j} - P_{l_j}^T) = \left.\tfrac{\partial^{2k'} F}{\partial(P-P^T)^{2k'}}\right|_{P^T},$$

   where we use that $\left.\frac{\partial \pi_l(\lambda_1,\ldots,\lambda_{k'})}{\partial(-Q^T)}\right|_{Q^T} = P_l - P_l^T$, and that $P_l - P_l^T$ is constant. Thus for every $T \subseteq [k]$, where $|T| = k'$, $\mathcal{U}$ can reconstruct $\left.\frac{\partial^{2k'} f}{\partial(-Q^T)^{2k'}}\right|_{Q^T}$.

Reconstructing f: It suffices to show the above information is sufficient to reconstruct f. Assume there are two functions $f_1 \neq f_2 \in \mathbb{F}_p[\lambda_1, \ldots, \lambda_{k'}]$ that agree on all of the constraints above. Consider their difference $f = f_1 - f_2$. We shall prove that f is identically zero. By Lemma 14, f can be written as

$$f = g \prod_{h=1}^{k} g_h^2,$$

for some $g \in \mathbb{F}_p[\lambda_1, \ldots, \lambda_{k'}]$ with $\deg g \leq d - 2k$.

We induct downwards on r, starting with $r = k'$, to show $g|_{\cap_{h\in T} L_h} = 0$ for every set T of size r. It will follow for $r = 0$ that $g|_L = 0$, and thus $g = 0$. For $r = k'$, since $-Q^T$ is off L_h for every $h \in T$, by Lemma 15 and the above, $g(Q^T) = 0$ for every $T \subset [k]$ with $|T| = r$.

Let $r < k'$ and assume inductively that $g|_{\cap_{h\in T} L_h} = 0$ for every set T of size greater than r. Let $M = \cap_{h\in T} L_h$ for an arbitrary set T of size r. Then $\dim(M) = k' - r$ (recall equation (1)). Consider the $(k'-r-1)$-dimensional spaces of the form $M' = \cap_{h \in T \cup \{j\}} L_h$ for some $j \in [k] \setminus T$. There are $k-r$ of them. Then in the space M, the M' are distinct hyperplanes and can therefore be described as solutions to $\rho_{M'} = 0$ for degree-1 polynomials $\rho_{M'}$. Applying Bézout's theorem,

$$\prod_{M'} \rho_{M'} \;\Bigg|\; g|_M \, .$$

The degree of $g|_M$ is at most $d-2k$ since M is an affine space, while $\deg\left(\prod_{M'} \rho_{M'}\right) = k-r$. But since $d \leq 3k-k'$ by assumption and $r < k'$ by induction, we have $d-2k < k-r$, which means that $g|_M = 0$. By induction, $f = g = 0$, which completes the proof.

Complexity : In the non-recursive steps, $\mathcal{U}$ sends each S_h the space $\pi(L_h)$ described by k' vectors in $\mathbb{F}_p^m$. S_h responds with $F_{\pi(L_h)}$ and $\frac{\partial F}{\partial z_1}\Big|_{\pi(L_h)}, \ldots, \frac{\partial F}{\partial z_m}\Big|_{\pi(L_h)}$, which is just a list of $(m+1)p^{k'} = O(1)$ values[7] in $\mathbb{F}_p$. In the recursive steps, by Lemma 16 the total communication is $\binom{k}{k'} O\left(\mathcal{C}_{\mathcal{P}}(m^{2k'}, k')\right)$. Since $m = O(n^{1/d})$, the total communication of our protocol is

$$C_{\mathcal{P}'}(n,k) = O\left(n^{1/d} + \binom{k}{k'} \mathcal{C}_{\mathcal{P}}(n^{2k'/d}, k')\right).$$

Acknowledgement

S. Yekhanin would like to express his deep gratitude to M. Sudan for introducing the problem to him and many helpful discussions during 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. *Proc. of STACS '90, LNCS* 415, pp. 37–48, 1990.

[3] D. Beaver, J. Feigenbaum, J. Kilian and P. Rogaway. Locally Random Reductions: Improvements and Applications. In *Journal of Cryptology*, 10(1), pp. 17–36, 1997.

[4] 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.

[5] A. Beimel and Y. Ishai. Information-Theoretic Private Information Retrieval: A Unified Construction. In *28th ICALP, LNCS* 2076, pp. 912-926, 2001.

[6] A. Beimel, Y. Ishai, and E. Kushilevitz. General Constructions for Information-Theoretic Private Information Retrieval. Unpublished manuscript available at www.cs.bgu.ac.il/beimel/pub.html

[7] 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.

[8] A. Beimel, Y. Ishai, and T. Malkin. Reducing the servers' computation in private information retrieval: Pir with preprocessing. In *Crypto' 2000, LNCS* 1880, pp. 56-74. 2000.

[9] A. Beimel and Y. Stahl. Robust information-theoretic private information retrieval. In *Proceedings of the 3rd conference on security in Communications networks (SCN)*, pp. 326–341, 2002.

[10] B. Chor and N. Gilboa. Computationally private information retrieval. In *Proc. of the 32th ACM Sym. on Theory of Computing (STOC)*, pp. 304-313, 2000.

[11] 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.

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

[13] R. Hartshorne, *Algebraic geometry*. New York : Springer, 1977.

[14] 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.

[15] R. Lidl and H. Niederreiter, Finite Fields. Cambridge: Cambridge University Press, 1985.

[16] T. Nagell. Introduction to Number Theory. New York: Wiley, 1951.

[17] A. Shamir. How to share a secret. *Communications of ACM,* 22:612-613, 1979.

[18] 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.

[7] Note that we did not attempt to optimize this constant.