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 $O \left( \frac{k^t \log k}{t} n^{1/(2k-1)/t} \right)$, removing the $(k^t)$ term of previous schemes. This answers an open question of [12].
- 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 [7].
- Smaller communication for instance hiding [3, 12], PIR with a polylogarithmic number of servers, robust PIR [8], 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 [6].

1 Introduction

Private information retrieval (PIR) was introduced in a seminal paper by Chor et al [10]. 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 $i \in [n] \seteq \{1, \ldots, n\}$. 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 [10]. To get around this, Chor et al [10] suggested replicating the database among $k > 1$ non-communicating servers. In this setting, one can do substantially better. Indeed, Chor et al [10] give a protocol with complexity $O(n^{1/3})$ for as few as 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 $O(2^k n^{1/(2k-1)})$ complexity for every k. Finally, in [6], building upon [12, 5], Beimel et al reduce the communication to $2^O(k) n^{\frac{2k-1}{2k}}$. 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 $c > 1$ [13, 16].

*Supported by an NDSEG fellowship.
†Supported in part by NTT Award MIT 2001-04 and NSF grant CCR 0219218.
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 $t$, $1 \leq t \leq k$, 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 [10, 12, 5]. Beimel and Ishai [5] give the best upper bound prior to this work: $O\left(\binom{k}{t} \frac{k^2}{t} n^{1/(2k-1)} \right)$. Since this bound grows rapidly with $t$, in [12] 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 $O\left(\frac{k^2}{t} \log k \ n^{1/(2k-1)} \right)$ 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 $\tilde{O}(m^{t/2}+2)$. This improves the previous bound1 of $O(m^{t/2}+2) \log t$ (see [12]) by a factor of $m^t$. When $m = \log n$, this is exactly the problem of PIR with $k = \Theta(\log n / \log \log n)$, for which we obtain the best known bound.

Another application of our techniques is $k$-out-of-$l$ robust PIR [8]. 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 $O(kn^{1/k} \log l)$ and $2^{\tilde{O}(k)n^{2/k} \log \log l \log l}$ [8]. 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 $O(kn^{1/(2k-1)} \log l)$.

Another concern with the abovementioned solutions is the time complexity of the servers per query. Beimel et al. [7] 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 [9] 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 $a$, then there is a 2-server protocol with query size $O(n/a^2)$. This resolves an open question of [4].

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 [6] 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 $F \in \mathbb{F}_p[z_1, \ldots, z_m]$ with $m = O(dn^{1/d})$. The polynomial $F$ is such that there is an encoding $E : [n] \rightarrow \mathbb{F}_p^m$ 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.) $\chi \subseteq \mathbb{F}_p^m$ containing the point $P$ and discloses certain subvarieties of $\chi$ 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 $\chi$. In particular the user obtains the desired value of $F(P)$. The idea of polynomial interpolation

---

1The best upper bound for 1-private PIR [6] does not apply since it is not known how to make it $t$-private, and in any case, the dependence on $k$ there is $2^{\text{th}(k)}$. 

---
has been used previously in the private information retrieval literature [2, 10, 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 [6]. The underlying variety is a low dimensional affine space.

2 Preliminaries

By default, variables $\lambda_k$ take values in a finite field $\mathbb{F}_p$ and variables $P, V, V^j, Q$ and $Q^j$ take values in $\mathbb{F}_p^n$. Let $W$ be an element of $\mathbb{F}_p^n$. We use the subscript $W_i$ to denote the i-th component of $W$.

A k-server PIR protocol involves k servers $S_1, \ldots, S_k$, each holding the same n-bit string $x$ (the database), and a user $U$ 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: [6] A t-private PIR protocol is a triplet of algorithms $\mathcal{P} = (Q, A, C)$. At the beginning of the protocol, the user $U$ invokes $Q(k, n, i)$ to pick a randomized k-tuple of queries $(q_1, \ldots, q_k)$, along with an auxiliary information string $aux$. It sends each server $S_j$ the query $q_j$ and keeps $aux$ for later use. Each server $S_j$ responds with an answer $a_j = A(k, j, x, q_j)$. (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, $U$ computes its output by applying the reconstruction algorithm $C(k, n, a_1, \ldots, a_k, aux)$.

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 $Q$).

- **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 $Q_T(k, n, i_1)$ and $Q_T(k, n, i_2)$ are identical, where $Q_T$ denotes concatenation of j-th outputs of $Q$ 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 $k$ 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 [6]. An important difference of our representation is that we use polynomials over non-binary fields. In particular for k-server protocols, we use prime fields $\mathbb{F}_p$, where $p$ is chosen such that $k < p < 2k$. The existence of such primes $p$ is implied by Bertrand's Postulate [14]. The polynomial $F$ represents $x$ in the following sense: with every $i \in [n]$ we associate a point $E(i) \in \mathbb{F}_p^n$; the polynomial $F$ satisfies:

$$\forall i \in [n], \quad F(E(i)) = x_i.$$
We use the assignment function $E : [n] \rightarrow \mathbb{F}_p^m$ from [6]. Let $E(1), \ldots, E(n)$ denote $n$ distinct points of Hamming weight $d$ with coordinate values from the set $\{0, 1\} \subset \mathbb{F}_p$. Such points exist if $\binom{m}{d} \geq n$. Therefore $m = O(dn^{1/d})$ variables are sufficient. Define

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

($E(i)_l$ 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, $F(E(i)) = x_i$. We conclude this section with a simple technical lemma.

**Lemma 1** Suppose $\{\lambda_h\}, \{v_h^0\}, \{v_h^1\}$ are elements of $\mathbb{F}_p$, where $h \in [s]$ and $\{\lambda_h\}$ are pairwise distinct; then there exists at most one polynomial $f(\lambda) \in \mathbb{F}_p[\lambda]$ 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

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

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 [10]. This protocol has a simple geometric interpretation and has served as the starting point for our work.

**Theorem 2 ([10])** There exists a 1-private $k$-server PIR protocol with communication complexity $O(k^2 \log k \cdot n^{1/(k-1)})$.

**Protocol description:** Let $\lambda_1, \ldots, \lambda_k \in \mathbb{F}_p$ be pairwise distinct and nonzero. Set $d = k - 1$. Let $P = E(i)$. The user wants to retrieve $F(P)$.

$$\begin{align*}
\mathcal{U} & : \text{Picks } V \in \mathbb{F}_p^m \text{ uniformly at random}. \\
\mathcal{U} \rightarrow S_h & : P + \lambda_h V \\
\mathcal{U} \leftarrow S_h & : F(P + \lambda_h V)
\end{align*}$$

**Privacy:** It is immediate to verify that the input $(P + \lambda_h V)$ of each server $S_i$ is distributed uniformly over $\mathbb{F}_p^m$. 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 $L = \{P + \lambda V \mid \lambda \in \mathbb{F}_p\}$ in the space $\mathbb{F}_p^m$. Let $f(\lambda) = F(P + \lambda V)$ be the restriction of $F$ to $L$. Clearly, $f \in \mathbb{F}_p[\lambda]$ 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 $\mathbb{F}_p$. Recall that $m = O(dn^{1/d})$ and $k < p < 2k$. Thus the total communication from the user to all the
servers is $O(k^2 \log k \cdot n^{1/(k-1)})$. Each $S_h$ responds with a single value from $\mathbb{F}_p$, 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 2 to fix this imbalance and obtain a better communication complexity.

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

**Protocol description**: We use the standard mathematical notation $\frac{\partial F}{\partial z_i} \bigg|_Q$ to denote the value of the partial derivative of $F$ with respect to $z_i$ at point $Q$. Let $\lambda_1, \ldots, \lambda_k \in \mathbb{F}_p$ be pairwise distinct and nonzero. Set $d = 2k - 1$. Let $P = E(i)$. The user wants to retrieve $F(P)$.

| $U$ | Picks $V \in \mathbb{F}_p^n$ uniformly at random. |
| $U \rightarrow S_h$ | $P + \lambda_h V$ |
| $U \leftarrow S_h$ | $F(P + \lambda_h V), \frac{\partial F}{\partial z_1} \bigg|_{P+\lambda_h V}, \ldots, \frac{\partial F}{\partial z_m} \bigg|_{P+\lambda_h V}$ |

**Privacy**: The proof of privacy is identical to the proof from Theorem 2.

**Correctness**: Again, consider the line $L = \{ P + \lambda V \mid \lambda \in \mathbb{F}_p \}$. 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 underlining our protocol is that knowing the values of partial derivatives $\frac{\partial F}{\partial z_1} \bigg|_{P+\lambda_h V}, \ldots, \frac{\partial F}{\partial z_m} \bigg|_{P+\lambda_h V}$, the user can reconstruct the value of $f'(\lambda_h)$. The proof is a straightforward application of the chain rule:

$$\frac{\partial f}{\partial \lambda} \bigg|_{\lambda_h} = \frac{\partial F(P + \lambda V)}{\partial \lambda} \bigg|_{\lambda_h} = \sum_{i=1}^m \frac{\partial F}{\partial z_i} \bigg|_{P+\lambda_h V} V_i.$$

Thus the user can reconstruct $\{f(\lambda_h)\}$ and $\{f'(\lambda_h)\}$ for all $h \in [k]$. Combining this observation with Lemma 1, 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 $\mathbb{F}_p$. Servers respond with length-$m$ vectors of values in $\mathbb{F}_p$. Recall that $m = O(dm^{1/d})$ and $p < 2k$. Thus the total communication is $O(k^2 \log k \cdot n^{1/(2k-1)})$.

### 3.1 Application to Robust PIR

We review the definition of robust PIR [8].

**Definition 4** 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 [8], 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 $2\tilde{O}(k)n^{2/\log \log k} \log l$ and $O(kn^{1/k} l \log l)$ given in [8].

Indeed, in the protocol above, if for $l$ servers we set the field size $p > l$ and the degree $\deg F = 2k - 1$, then from any $k$ servers’ answers, we can reconstruct $f$ as before. We conclude
Theorem 5 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 6 There exists a $t$-private $k$-server PIR protocol with communication complexity $O\left(\frac{k^2}{t} \log k n^{1/(2k-1)}\right)$.

Protocol description: Again, let $\lambda_1, \ldots, \lambda_k \in \mathbb{F}_p$ be pairwise distinct and nonzero. Set $d = \lceil \frac{2k-1}{t} \rceil$. Let $P = E(i)$. The user wants to retrieve $F(P)$:

\[
\begin{align*}
\mathcal{U} : & \text{ Picks } V^1, \ldots, V^t \in \mathbb{F}_p^m \text{ uniformly at random.} \\
\mathcal{U} \rightarrow \mathcal{S}_h : & \quad \mathcal{Q}^h \equiv P + \lambda_h V^1 + \lambda_h^2 V^2 + \ldots + \lambda_h^t V^t \\
\mathcal{U} \leftarrow \mathcal{S}_h : & \quad F(\mathcal{Q}^h), \frac{\partial F}{\partial \mathcal{Q}^1}_1, \ldots, \frac{\partial F}{\partial \mathcal{Q}^m}_1 \bigg|_{\mathcal{Q}^h}
\end{align*}
\]

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 \{\mathcal{Q}^h \equiv 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 $V_l^j \in \mathbb{F}_p$ chosen independently and uniformly at random; the values \{\mathcal{Q}^h \equiv P + \lambda_h V^1 + \ldots + \lambda_h^t V^t\}_{h \in T}$ disclose no information about $P_l$. The last statement is implied by the properties of Shamir’s secret sharing scheme [15].

Correctness: Consider the curve $\chi = \{P + \lambda V^1 + \ldots + \lambda^t V^t | \lambda \in \mathbb{F}_p\}$. Let $f(\lambda) = \sum_{l=1}^{m} \frac{\partial F}{\partial \mathcal{Q}^l}_l \bigg|_{\mathcal{Q}^h}$.

Thus $\mathcal{U}$ can reconstruct $\{f(\lambda_h)\}$ and $\{f'(\lambda_h)\}$ for all $h \in [k]$. Combining this observation with Lemma 1, we conclude that the user can reconstruct $f$ and obtain $F(P) = f(0)$.

Complexity: As in the protocol of Theorem 3, $\mathcal{U}$ sends each of $k$ servers a length-$m$ vector of values in $\mathbb{F}_p$, and servers respond with length-($m + 1$) vectors of values in $\mathbb{F}_p$. Here $m = O(dn^{1/d})$ and $p < 2k$. Thus the total communication is $O\left(\frac{k^2}{t} \log k n^{1/(2k-1)}\right)$.

4.1 Application to Instance Hiding

As noted in the introduction, in the instance-hiding scenario [2, 3] there is a function $F_m : \{0,1\}^m \rightarrow \{0,1\}$ 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 $\tilde{O}(m^{ct/2} + t)$ bound of [12].

Theorem 7 There exists a $t$-private non-adaptive oracle instance-hiding scheme with communication and computation $\tilde{O}(m^{ct/2} + t)$, where $\tilde{O}(f) \equiv O(f \log^{O(1)} f)$.
Proof: Using the above protocol on the truth table of $F_m$, the communication is

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

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 [7] 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 8** There exists a 2-server PIR protocol with $O(n^{1/3})$ communication, 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 [7].

**Lemma 9** Let $F$ be a homogeneous degree-$d$ polynomial in $\mathbb{F}_p[z_1, \ldots, z_m]$. Using 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 $cz_{t_1} \cdots z_{t_d}$ for some $c \in \mathbb{F}_p$ and $t_1 \in D_{t_1}, \ldots, t_d \in D_{t_d}$. The following is the preprocessing algorithm.

<table>
<thead>
<tr>
<th>Preprocess($F$):</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. For each polynomial $F_{D_{t_1}, \ldots, D_{t_d}}$,</td>
</tr>
<tr>
<td>(a) Evaluate $F_{D_{t_1}, \ldots, D_{t_d}}$ on all $W \in \mathbb{F}<em>p^m$ for which $\text{Supp}(W) \in \cup_i D</em>{t_i}$.</td>
</tr>
</tbody>
</table>

**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 $p^{d \log m} = \text{poly}(m)$ different $W$ whose support is in $\cup_i D_{t_i}$. Thus the algorithm needs only poly($m$) preprocessing time.

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

<table>
<thead>
<tr>
<th>Evaluate($F, V$):</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. $\sigma \leftarrow 0$.</td>
</tr>
<tr>
<td>2. For each polynomial $F_{D_{t_1}, \ldots, D_{t_d}}$,</td>
</tr>
<tr>
<td>(a) $\sigma \leftarrow \sigma + F_{D_{t_1}, \ldots, D_{t_d}}(V</td>
</tr>
<tr>
<td>3. Output $\sigma$.</td>
</tr>
</tbody>
</table>

**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 use a balancing technique similar to that of [9]. Let the field be \( \mathbb{F}_p \) for some \( p > 2 \). \( S_1 \) and \( S_2 \) preprocess as follows.

Preprocessing phase(\( x \)):
1. \( s \leftarrow \frac{r-1}{3}, \ t \leftarrow n^s \).
2. Partition \( x \) into \( t \) databases \( DB_1, \ldots , DB_t \), each containing \( n^{1-s} \) elements.
3. Let \( DB_j \) be represented by a homogeneous polynomial \( F_j \) of degree \( d = 2r + 1 \) with \( m = O(n^{(1-s)/d}) \) variables.
4. For \( a = 0, \ldots , r, \ j \in [t], \) and \( l_1, \ldots , l_a \in [m] \), compute \( \text{Preprocess} \left( \frac{\partial^a F_j}{\partial z_{l_1} \cdots \partial z_{l_a}} \right) \).

Let \( DB_u \) be the database containing \( x_i \). Assume the user wants \( F_u(P) \). Let \( \delta(\alpha, \beta) \) be an indicator function which is 1 if \( \alpha = \beta \), and 0 otherwise.

\[
\begin{align*}
\mathcal{U} & \quad : \text{Picks } V^1, \ldots , V^t \in \mathbb{F}_p^m \text{ uniformly at random.} \\
\mathcal{U} & \rightarrow S_h \quad : \text{For } j \in [t], \ Q^{h,j} \overset{\text{def}}{=} (-1)^{h+1} V^j + \delta(j, u) P \\
\mathcal{U} & \leftarrow S_h \quad : \forall a \in \{0, \ldots , r\} \text{ and } l_1, \ldots , l_a \in [m], \\
& \quad \sum_{j=1}^t \frac{\partial^a F_j}{\partial z_{l_1} \cdots \partial z_{l_a}} \bigg|_{Q^{h,j}} = \sum_{j=1}^t \text{Evaluate} \left( \frac{\partial^a F_j}{\partial z_{l_1} \cdots \partial z_{l_a}} , Q^{h,j} \right)
\end{align*}
\]

Correctness: Since \( d \) is odd,

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

It follows that

\[
\sum_{j \neq u_1, \ldots , u_a} \left| \left. \frac{\partial^a F_j}{\partial z_{l_1} \cdots \partial z_{l_a}} \right|_{V^j} \right| \cdot V_{l_1}^u \cdots V_{l_a}^u + (-1)^a \sum_{j \neq u_1, \ldots , u_a} \left| \left. \frac{\partial^a F_j}{\partial z_{l_1} \cdots \partial z_{l_a}} \right|_{Q^{2,j}} \right| \cdot V_{l_1}^u \cdots V_{l_a}^u = 0.
\]

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

\[
\begin{align*}
\sum_{j} \sum_{l_1, \ldots , l_a} \left( \left| \left. \frac{\partial^a F_u}{\partial z_{l_1} \cdots \partial z_{l_a}} \right|_{Q^{1,j}} \right| \cdot (-1)^a \left| \left. \frac{\partial^a F_j}{\partial z_{l_1} \cdots \partial z_{l_a}} \right|_{Q^{2,j}} \right| \right) V_{l_1}^u \cdots V_{l_a}^u \\
= \sum_{l_1, \ldots , l_a} \left( \left| \left. \frac{\partial^a F_u}{\partial z_{l_1} \cdots \partial z_{l_a}} \right|_{P+V^u} \right| \cdot (-1)^a \left| \left. \frac{\partial^a F_u}{\partial z_{l_1} \cdots \partial z_{l_a}} \right|_{P-V^u} \right| \right) V_{l_1}^u \cdots V_{l_a}^u \\
= f^{(a)}(1) + (-1)^a f^{(a)}(-1) = g^{(a)}(1),
\end{align*}
\]

and 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

\[
(\gamma - 1)^{r+1} \mid (h_1 - h_2),
\]
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 $S_1, S_2$ is independent of $P$.

**Communication:** $U$ sends $O(tm) = O(n^{s+1-1}+(2r+1)) = O(n^{r-1}/(3^r+1)) = O(n^{1/3})$ bits. $S_1, S_2$ respond with $O(m + m^2 + \cdots + m^r) = O(m^r) = O(n^{1-1}/(2r+1)) = O(n^{1/3})$ 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 $V$, $\text{Evaluate}(\frac{\partial^a F_j}{\partial z_{l_1} \cdots \partial z_{l_a}}, V)$ 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

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

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

### 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 10** 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, $S_1$ and $S_2$ partition $x$ into $t = O(n/b^3)$ databases $DB_1, \ldots, DB_t$ each of size $b^3$. Each such DB is a degree-3 polynomial in $m = O(b)$ variables. Again, let $DB_u$ be the database containing $x_i$. The protocol follows.

$$\begin{align*}
U & \quad: \text{Picks } V^1, \ldots, V^t \in \mathbb{F}_p^m \text{ uniformly at random.} \\
U \rightarrow S_h & \quad: \text{For } j \in [t], \ Q^{h,j} \overset{\text{def}}{=} (-1)^h V^j + \delta(j, u) P \\
U \leftarrow S_h & \quad: \text{For } j \in [t], \ Q^{h,j} = \sum_j F_j(Q^{h,j}), \text{ and for all } l \in [m], \sum_j \frac{\partial F_j}{\partial z_l} \bigg|_{Q^{h,j}}
\end{align*}$$

The correctness follows from the correctness of our preprocessing protocol for $r = 1$. For the communication, $U$ sends $tm = O(n/b^2)$ bits, and $S_1$ and $S_2$ each respond with $O(b)$ bits.

### 6 Recursive PIR in the space

Assume $k$ is constant. The best known upper bound of $n^O(\log \log k)$ for the communication complexity of 1-private $k$ server PIR protocols is due to Beimel et al. [6]. Although their proof is elementary, it is rather complicated and hard to follow. The key theorem of [6] is:
Theorem 11 ([6] Theorem 3.5) Suppose there is a 1-private PIR protocol $P$ with communication complexity $C_P(n,k)$. Let $d, k, k'$ be positive integers (which may depend on $k$) such that $k' < k$ and $d \leq (\lambda + 1)k - (\lambda - 1)k' + (\lambda - 2)$. Then there is a 1-private PIR protocol $P'$ with communication complexity

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

Recursive applications of Theorem 11 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 11 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 [6] for all values of $k < 26$, where the bound on $k$ was determined experimentally.

Theorem 12 Suppose there is a 1-private PIR protocol $P$ with communication complexity $C_P(n,k)$. Let $d, k'$ be positive integers such that $k' < k$ and $d \leq 3k - k'$. Then there is a 1-private PIR protocol $P'$ with communication complexity

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

It may seem that the bound of Theorem 12 improves upon the bound of Theorem 11 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 11 can also be modified to eliminate these terms.

We start with a high-level view of our protocol. $U$ wants to retrieve the value $F(P)$. To this end $U$ randomly selects a $k'$ dimensional affine subspace\(^2\) $\pi(L)$ containing the point $P$ and sends each server $S_h$ a $(k' - 1)$ dimensional affine subspace\(^3\) $\pi(L_h) \subseteq \pi(L)$. Each $S_h$ 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 $U$ 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 $U$ 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 pairwise 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 $L_h \subseteq L$:

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

\(^2\)We use the complicated notation $\pi(L)$ for consistency with the actual proof.

\(^3\)In certain degenerate cases the dimensions of both $\pi(L)$ and $\pi(L_h)$ may in fact be smaller than $k'$ and $k' - 1$.\n
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.:

\[
\dim \bigcap_{h \in T} L_h = k' - |T|.
\]  

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 \( \alpha_h v_1 + \alpha^2_h v_2 + \ldots + \alpha^{k'}_h v_{k'} \neq 0 \). 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 \( \pi \), 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 13** 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 ([11] p. 53)\(^4\). To see that \( g_h \) divides \( f \) twice, let \( f = g \cdot g_h \). By the chain rule,

\[
\frac{\partial f}{\partial v} = \frac{\partial g}{\partial v} g_h + g \sum_i \alpha^i_h v_i,
\]

and since \( v \) is off of \( L_h \), \( \sum_j \alpha^j_h v_j \neq 0 \). 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. \( \blacksquare \)

The next lemma concerns the recursive part of our protocol.

**Lemma 14** 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 \( \{g_h\}_{h \in T} \).

---

\(^4\)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 \( |\overline{\mathbb{F}}_p| > \deg f \). 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 \( g_h \mid f \).
Proof: Let $C_h = \frac{\partial F}{\partial z_h} = \sum_j \alpha_j^h v_j$, and observe that $C_h \neq 0$ since $v$ is off of $L_h$. By repeated application of the chain rule,

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

where $\delta(\alpha, \beta) = 1$ if $\alpha = \beta$ and 0 otherwise. Again by the chain rule,

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

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 [6]) 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 $\frac{\partial^{2k'} F}{\partial (P-P_T)^{2k'}}|_{P_T}$. Below is the formal description of the protocol. Here $S_T$ denotes the set of servers $\{S_h\}_{h \in T}$.

$$\begin{align*}
\mathcal{U} & : \text{Picks a random } \pi : L \to \mathbb{F}_p^m, \quad \pi(\lambda_1, \ldots, \lambda_{k'}) = P + \lambda_1 V^1 + \ldots + \lambda_{k'} V^{k'}.
\mathcal{U} \to S_h & : \pi(L_h)
\mathcal{U} \leftarrow S_h & : \quad F|_{\pi(L_h)}, \quad \frac{\partial F}{\partial z_1}|_{\pi(L_h)}, \ldots, \quad \frac{\partial F}{\partial z_m}|_{\pi(L_h)}
\mathcal{U} \leftrightarrow S_T & : \text{A } k'\text{-server PIR subprotocol for retrieving } \frac{\partial^{2k'} F}{\partial (P-P_T)^{2k'}}|_{P_T}
\end{align*}$$

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

Lemma 15 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

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

privately (with respect to $i$) with communication complexity $O(C_T(m^s, k'))$.

Proof: We have

$$\frac{\partial^s F}{\partial (P-P^T)^s}|_{P^T} = \sum_{i_1, \ldots, i_s} \frac{\partial^s F}{\partial z_{i_1} \cdots z_{i_s}}|_{P^T} (P-P^T)_{i_1} \cdots (P-P^T)_{i_s},$$

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 $2^d C_T(O(m^s), k') = O(C_T(m^s, k'))$. ■
We now show the desired properties of our protocol.

**Privacy**: Since the subprotocols are independent, and $P$ is private by assumption (recall the condition of theorem 12), to show that $P'$ is private it suffices to show privacy at the top level of recursion. In this level $S_h$’s view is

$$\pi(L_h) = \{ P + \lambda_1 V^1 + \ldots + \lambda_k V^k \mid \lambda_j \in \mathbb{F}_p, \ \alpha_h \lambda_1 + \ldots + \alpha_h^k \lambda_k^i = 1 \}.$$ 

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

$$P + (\alpha_h)^{-1} V^1, P + (\alpha_h^2)^{-1} V^2, \ldots, P + (\alpha_h^k)^{-1} V^k \in \pi(L_h).$$

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

$$(P + (\alpha_h)^{-1} V^1, P + (\alpha_h^2)^{-1} V^2, \ldots, P + (\alpha_h^k)^{-1} V^k) \equiv (R^1, \ldots, R^k),$$

where the $R^j \in \mathbb{F}_p^m$ are independent and uniformly random. Thus $S_h$’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 $U$ obtains from $\{ S_h \}_{h \in [k]}$ 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 $U$ 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 $\frac{\partial f}{\partial v_h}|_{L_h}$ from $\frac{\partial F}{\partial z_i}|_{\pi(L_h)}, \ldots, \frac{\partial F}{\partial z_m}|_{\pi(L_h)}$. From the chain rule

$$\frac{\partial f}{\partial v_h}|_{L_h} = \frac{\partial F(\pi(\lambda_1, \ldots, \lambda_k))}{\partial v_h}|_{L_h} = \sum_{i=1}^m \frac{\partial F}{\partial z_i}|_{\pi(L_h)} \frac{\partial}{\partial v_h}(P_t + \lambda_1 V^1_t + \ldots + \lambda_k V^k_t)|_{L_h}.$$

Thus for every $h \in [k]$, $U$ 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 $\pi(\lambda_1, \ldots, \lambda_k)$ denote $P_t + \lambda_1 V^1_t + \ldots + \lambda_k V^k_t$ for $l \in [m]$. We have

$$\frac{\partial^{2k'} f}{\partial (-Q^T)^{2k'}}|_{Q^T} = \frac{\partial^{2k'} F(\pi(\lambda_1, \ldots, \lambda_k))}{\partial (-Q^T)^{2k'}}|_{Q^T} = \sum_{i_1, \ldots, i_{2k'}} \frac{\partial^{2k'} F}{\partial z_{i_1} \ldots z_{i_{2k'}}}|_{P^T} \prod_{j=1}^{2k'} (P_{t_j} - P_{i_j}^T),$$

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

**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 13, $f$ 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 14 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,

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

The degree of $g|_M$ is at most $d - 2k$ since $M$ is an affine subspace, 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, $U$ sends each $S_h$ the space $\pi(L_h)$ described by $k'$ vectors in $\mathbb{F}_p^n$. $S_h$ responds with $F_{\pi(L_h)}$ and $\frac{\partial F}{\partial x_{\pi(L_h)}} \cdots \frac{\partial F}{\partial x_{\pi(L_h)}}$, which is just a list of $(m+1)p^{k'} = O(1)$ values in $\mathbb{F}_p$. In the recursive steps, by Lemma 15 the total communication is $\binom{k'}{k} \mathcal{C}_P(m^{2k'}, k')$. Since $m = O(n^{1/d})$, the total communication of our protocol is

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

**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**


---

5Note that we did not attempt to optimize this constant.


