Polylogarithmic Private Approximations and Efficient Matching

Piotr Indyk
MIT
indyk@mit.edu

David Woodruff
MIT
dpwood@mit.edu

Abstract

In [12] a *private approximation* of a function f is defined to be another function F that approximates f in the usual sense, but does not reveal any information about x other than what can be deduced from $f(x)$. We give the first two-party private approximation of the l_2 distance with polylogarithmic communication. This, in particular, resolves the main open question of [12].

We then look at the *private near neighbor* problem in which Alice has a query point in $\{0,1\}^d$ and Bob a set of n points in $\{0,1\}^d$, and Alice should privately learn the point closest to her query. We improve upon existing protocols, resolving open questions of [13, 10]. Then, we relax the problem by defining the *private approximate near neighbor problem*, which requires introducing a notion of secure computation of approximations for functions that return sets of points rather than values. For this problem we give several protocols with sublinear communication.

1 Introduction

Recent years witnessed the explosive growth of the amount of available data. Large data sets, such as transaction data, the web and web access logs, or network traffic data, are in abundance. Much of the data is stored or made accessible in a distributed fashion. This neccessitates the development of efficient protocols that compute or approximate functions over such data (e.g. see [2]).

At the same time, the availability of this data has raised significant privacy concerns. It became apparent that one needs cryptographic techniques in order to control data access and prevent potential misuse. In principle, this task can be achieved using the general results of secure function evaluation (SFE) [32, 18]. However, in most cases the resulting private protocols are much less efficient than their non-private counterparts[1]. Moreover, SFE applies only to algorithms that compute functions *exactly*, while for most massive data sets problems, only efficient *approximation* algorithms are known or are possible. Indeed, while it is true that SFE can be used to privately implement any efficient algorithm, it is of little use applying it to an approximation algorithm when the approximation leaks more information about the input than the solution itself.

In a pioneering paper [12], the authors introduced a framework for secure computation of approximations. They also proposed an $\tilde{O}(\sqrt{n})$-communication[2] two-party protocol for approximating the Hamming distance between two binary vectors. This improves over the linear complexity of computing the distance exactly via SFE, but still does not achieve the polylogarithmic efficiency of a non-private protocol of [24]. Improving the aforementioned bound was one of the main problems left open in [12].

[1] A rare exception is the result of [28], who show how to obtain private and communication-efficient versions of non-private protocols, as long as the communication cost is logarithmic.

[2] We write $f = \tilde{O}(g)$ if $f(n,k) = O\left(g(n,k)\log^{O(1)}(n)\text{poly}(k)\right)$, where k is a security parameter.

In this paper we provide several new results for secure computation of approximations. Our first result is an $\tilde{O}(1)$-communication protocol for approximating the Euclidean (ℓ_2) distance between two vectors. This, in particular, solves the open problem of [12]. Since distance computation is a basic geometric primitive, we believe that our result could lead to other algorithms for secure approximations. Indeed, in [1] the authors show how to approximate the ℓ_2 distance using small space and/or short amount of communication, initiating a rich body of work on streaming algorithms.

In the second part of the paper, we look at secure computation of a *near neighbor* for a query point q (held by Alice) among n data points P (held by Bob) in $\{0,1\}^d$. We improve upon known results [10, 13] for this problem under various distance metrics, including ℓ_2, set difference, and Hamming distance over arbitrary alphabets. Our techniques also result in better communication for the *all-near neighbors* problem, where Alice holds n different query points, resolving an open question of [13], and yield a binary inner product protocol with communication $d + O(k)$ in the common random string model.

Complexity	Problem	Prior work	SFE
$\tilde{O}(n+d)$	near neighbor under l_2, Hamming over $\{0,1\}^d$, set difference	[10]	$\tilde{O}(nd)$
$\tilde{O}(dU+n)$	near neighbor under distances $f(a,b)=\sum_{i=1}^d f_i(a_i,b_i)$, $a_i, b_i \in [U]$	[10]	$\tilde{O}(nd\log U)$
$\lceil \log d \rceil d + O(k)$	Hamming distance	[14]	$O(kd)$
$\tilde{O}(nd^2+n^2)$	all-near neighbors	[13]	$\tilde{O}(n^2 d)$

However, all of our protocols for the near neighbor problem have the drawback of needing $\Omega(n)$ bits of communication, though the dependence on d is often optimal. Thus, we focus on what we term the *approximate near neighbor problem*. For this we introduce a new definition of secure computation of approximations for functions that return points (or sets of points) rather than values.

Approximate privacy. Let $P_t(q)$ be the set of points in P within distance t from q. In the *c-approximate near neighbor* problem, the protocol is required to report a point in $P_{cr}(q)$, as long as $P_r(q)$ is nonempty. We say that a protocol solving this problem is *c'-private* (or just *private* if $c' = c$) if Bob learns nothing, while Alice learns nothing except what can be deduced from the set $P_{c'r}(q)$. In our paper we always set $c' = c$.

We believe this to be a natural definition of privacy in the context of the approximate near neighbor problem. First, observe that if we insist that Alice learns only the set P_r (as opposed to P_{cr}), then the problem degenerates to the *exact* near neighbor problem. Indeed, even though the definition of correctness allows the protocol to output a point $p \in P_{cr} - P_r$, in general Alice cannot simulate this protocol given only the set P_r. Thus, in order to make use of the flexibility provided by the approximate definition of the problem, it seems necessary to relax the definition of privacy as well.

Second, the above relaxation of privacy appears natural in the context of applications of near neighbor algorithms. In most situations, the distance function is only a heuristic approximation of the dis-similarity between objects, and there is no clear rationale for a sharp barrier between objects that can or cannot be revealed (still, it is important that the information leak is limited). Our model formalizes this intuition, and our algorithmic results shows that it is possible to exploit the model to obtain more efficient algorithms.

Specifically, within this framework, we give a c-approximate near neighbor protocol with communication $\tilde{O}(n^{1/2}+d)$ for any constant $c > 1$. The protocol is based on dimensionality reduction technique of [24]. We show how the dependence on d can be made polylogarithmic if Alice just wants a coordinate of a point in P_{cr}. We also give a protocol based on locality-sensitive hashing (LSH) [23], with communication $\tilde{O}(n^{1/2+1/(2c)}+d)$, but significantly less work (though still polynomial).

Finally, proceeding along the lines of [20], we say the protocol *leaks b bits of information* if it can be simulated given b extra bits which may depend arbitrarily on the input. With this definition, we give a protocol with $\tilde{O}(n^{1/3}+d)$ communication leaking only k bits, where k is a security parameter.

General vs specific solutions. As described above, this paper offers solutions to *specific* computational problems. In principle, a general "compiler-like" approach (as in [32, 18]) would be preferable. However, it appears unlikely that a compiler approach can be developed in the context of *approximate* problems. Indeed, there is no general method that, for a given problem, generates an efficient approximation algorithm (even ignoring the privacy issue). This implies that a compiler would have to start from a particular approximation to a given function. Unfortunately, as mentioned earlier, such approximation itself can leak too much information.

This argument leads us to believe that, in context of approximate algorithms, designing efficient private solutions to specific problems is the only possible approach.

2 Preliminaries

Background on homomorphic encryption, oblivious transfer (OT), and secure function evaluation (SFE) can be found in appendix A.

We assume both parties are computationally bounded and semi-honest, meaning they follow the protocol but may keep message histories in an attempt to learn more than is prescribed. In [18, 7, 28], it is shown how to transform a semi-honest protocol into a protocol secure in the malicious model. Further, [28] does this at a communication blowup of at most a factor of $\text{poly}(k)$. Therefore, we assume parties are semi-honest in the remainder of the paper.

We briefly review the semi-honest model, referring the reader to [17, 25] for more details. Let $f : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^* \times \{0,1\}^*$ be a function, the first element denoted $f_1(x_1, x_2)$ and the second $f_2(x_1, x_2)$. Let π be a two-party protocol for computing f. The views of players P_1 and P_2 during an execution of $\pi(x_1, x_2)$, denoted $\texttt{View}_1^\pi(x_1, x_2)$ and $\texttt{View}_2^\pi(x_1, x_2)$ respectively, are:

$$\texttt{View}_1^\pi(x_1, x_2) = (x_1, r_1, m_{1,1}, \ldots, m_{1,t}), \texttt{View}_2^\pi(x_1, x_2) = (x_2, r_2, m_{2,1}, \ldots, m_{2,t}),$$

where r_i is the random input and $m_{i,j}$ the messages received by player i respectively. The outputs of P_1 and P_2 during an execution of $\pi(x_1, x_2)$ are denoted $\texttt{output}_1^\pi(x_1, x_2)$ and $\texttt{output}_2^\pi(x_1, x_2)$. We define $\texttt{output}^\pi(x_1, x_2)$ to be $(\texttt{output}_1^\pi(x_1, x_2), \texttt{output}_2^\pi(x_1, x_2))$. We say that π privately computes a function f if there exist PPT algorithms S_1, S_2 for which for $i \in \{1, 2\}$ we have the following indistinguishability

$$\{S_i(x_i, f_i(x_1, x_2)), f(x_1, x_2)\} \stackrel{c}{\equiv} \{\texttt{View}_i^\pi(x_1, x_2), \texttt{output}^\pi(x_1, x_2)\}.$$

This simplifies to $\{S_i(x_i, f_i(x_1, x_2))\} \stackrel{c}{\equiv} \{\texttt{View}_i^\pi(x_1, x_2)\}$ if either $f_1(x_1 x_2) = f_2(x_1, x_2)$ or if $f(x_1, x_2)$ is deterministic or equals a specific value with probability $1 - \text{negl}(k, n)$, for k a security parameter.

We need a standard composition theorem [17] concerning private subprotocols. An *oracle-aided protocol* (see [25]) is a protocol augmented with a pair of oracle tapes for each party and oracle-call steps. In an oracle-call step parties write to their oracle tape and the oracle responds to the requesting parties. An oracle-aided protocol uses the *oracle-functionality* $f = (f_1, f_2)$ if the oracle responds to query x, y with $(f_1(x, y), f_2(x, y))$, where f_1, f_2 denote first and second party's output respectively. An oracle-aided protocol *privately reduces* g to f if it privately computes g when using oracle-functionality f.

Theorem 1 *[17] If a function g is privately reducible to a function f, then the protocol g' derived from g by replacing oracle calls to f with a protocol for privately computing f, privately computes g.*

We now define the *functional privacy* of an approximation as in [12]. For our approximation protocols we will have $f_1(x, y) = f_2(x, y) = f(x, y)$.

Definition 2 *Let $f(x,y)$ be a function, and let $\hat{f}(x,y)$ be a randomized function. Then $\hat{f}(x,y)$ is functionally private for f if there is an efficient simulator S s.t. for every x,y, we have $\hat{f}(x,y) \stackrel{c}{\equiv} S(f(x,y))$.*

A *private approximation* of f privately computes a randomized function $\hat{f}$ that is functionally private for f.

Finally, we need the notion of a protocol for securely evaluating a circuit *with ROM*. In this setting, the ith party has a table $R_i \in (\{0,1\}^r)^s$ defined by his inputs. The circuit, in addition to the usual gates, is equipped with *lookup gates* which on inputs (i,j), output $R_i[j]$.

Theorem 3 *[28] If C is a circuit with ROM, then it can be securely computed with $\tilde{O}(|C|T(r,s))$ communication, where $T(r,s)$ is the communication of 1-out-of-s OT on words of size r.*

3 Private ℓ_2 Approximation

Here we give a private approximation of the ℓ_2 distance. Alice is given a vector $a \in [M]^n$, and Bob a vector $b \in [M]^n$. Note that $\|a-b\|^2 \leq T_{max} \stackrel{\text{def}}{=} nM^2$. In addition, parameters ϵ, δ and k are specified. For simplicity, we assume that $k = \Omega(\log(nM))$. The goal is for both parties to compute an estimate E such that $|E - \|x\|^2| \leq \epsilon\|x\|^2$ with probability at least $1-\delta$, for $x \stackrel{\text{def}}{=} a-b$. Further, we want E to be a private approximation of $\|x\|$, as defined in section 2. As discussed there, wlog we assume the parties are semi-honest. We set the parameter $B = \Theta(k)$; this notation means $B = ck$ for a large enough constant c independent from $k, n, M, \delta, \epsilon$. In our protocol we make the following cryptographic assumptions.

1. There exists a PRG G stretching $\text{polylog}(n)$ bits to n bits secure against $\text{poly}(n)$-sized circuits.
2. There exists an OT scheme for communicating 1 of n bits with communication $\text{polylog}(n)$.

At the end of the section we discuss the necessity and plausibility of these assumptions. Our protocol relies on the following fact and corollary.

Fact 4 *[26] Let A be a random $n \times n$ orthonormal matrix (i.e., A is picked from a distribution defined by the Haar measure). Then there is $c > 0$ such that for any $x \in \Re^n$, any $i = 1 \ldots n$, and any $t > 1$,*

$$\Pr[|(Ax)_i| \geq \frac{\|x\|}{\sqrt{n}}t] \leq e^{-ct^2}.$$

Corollary 5 *Suppose we sample A as in Fact 4 but instead generate our randomness from G, rounding its entries to the nearest multiple of $2^{-\Theta(B)}$. Then,*

$$\forall x \in \Re^n,\ \Pr[(1-2^{-B})\|x\|^2 \leq \|Ax\|^2 \leq \|x\|^2 \text{ and } \forall_i (Ax)_i^2 < \frac{\|x\|^2}{n}B] > 1 - \text{neg}(k,n)$$

Proof: If there were an infinite sequence of $x \in [M]^n$ for which this did not hold, a circuit with x hardwired would contradict the pseudorandomness of G. ■

Protocol Overview: Before describing our protocol, it is instructive to look at some natural approaches and why they fail. We start with the easier case of approximating the Hamming distance, and suppose the parties share a common random string. Consider the following non-private protocol of [24] discussed in [12]: Alice and Bob agree upon a random $O(\log n) \times n$ binary matrix R where the ith row consists of n i.i.d. Bernoulli(β^i) entries, where β is a constant depending on ϵ. Alice and Bob exchange Ra, Rb, and compute $R(a-b) = Rx$. Then $\|x\|$ can be approximated by observing that $\Pr[(Ra)_i = (Rb)_i] \approx 1/2$ if $\|x\| \gg \beta^{-i}$, and $\Pr[(Ra)_i = (Rb)_i] \approx 1$ if $\|x\| \ll \beta^{-i}$. Let the output be E. The communication is $O(\log n)$, but it is not private since both parties learn Rx. Indeed, as mentioned in [12], if $a = 0$ and $b = e_i$, then Rx equals the ith column of R, which cannot be simulated without knowing i.

However, given only $\|x\|$, it is possible to simulate E. Therefore, as pointed out in [12], one natural approach to try to achieve privacy is to run an SFE with inputs Ra, Rb, and output E. But this also fails, since knowing E *together with the randomness* R may reveal additional information about the inputs. If E is a deterministic function of Ra, Rb, and if $a = 0$ and $b = e_i$, Alice may be able to find i from a and R.

In [12], two private protocols which each have $\Omega(n)$ communication for a worst-case choice of inputs, were cleverly combined to overcome these problems and to achieve $\tilde{O}(\sqrt{n})$ communication. The first protocol, High-Distance Estimator, works when $\|x\| > \sqrt{n}$. The idea is for the parties to obliviously sample random coordinates of x, and use these to estimate $\|x\|$. Since the sampling is oblivious, the views depend only on $\|x\|$, and since it is random, the estimate is good provided we take $\tilde{O}(\sqrt{n})$ samples.

The second protocol, Low-Distance Estimator, works when $\|x\| \leq \sqrt{n}$. Roughly, the idea is for the parties to perfectly hash their vectors into $\tilde{O}(\sqrt{n})$ buckets so that at most one coordinate j for which $a_j \neq b_j$ lies in any given bucket. The parties then run an SFE with their buckets as input, which can compute $\|x\|$ exactly by counting the number of buckets which differ.

Our protocol breaks this $O(\sqrt{n})$ communication barrier as follows. First, Alice and Bob agree upon a random *orthonormal* matrix A in $\mathbb{R}^{n \times n}$, and compute Aa and Ab. The point of this step is to uniformly spread the mass of the difference vector x over the n coordinates, as per Fact 4, while preserving the length. Since we plan to sample random coordinates of Ax to estimate $\|x\|$, it is crucial to spread out the mass of $\|x\|$, as otherwise we could not for instance, distinguish $x = 0$ from $x = e_i$. The matrix multiplication can be seen as an analogue to the perfect hashing in Low-Distance Estimator, and the coordinate sampling as an analogue to that in High-Distance Estimator.

To estimate $\|x\|$ from the samples, we need to be careful of a few things. First, the parties should not learn the sampled values $(Ax)_j$, since these can reveal too much information. Indeed, if $a = 0$, then $(Ax)_j = (Ab)_j$, which is not private. To this end, the parties run a secure circuit with ROM (see section 2) Aa and Ab, which privately obtains the samples.

Second, we need the circuit's output distribution E to depend only on $\|x\|$. It is not enough for $\mathbf{E}[E] = \|x\|^2$, since a polynomial number of samples from E may reveal non-simulatable information about x based on E's higher moments. To this end, the circuit uses the $(Ax)_j$ to independently generate r.v.s z_j from a Bernoulli distribution with success probability depending only on $\|x\|$. Hence, z_j depends only on $\|x\|$.

Third, we need to ensure that the z_j contain enough information to approximate $\|x\|$. We do this by maintaining a loop variable T which at any point in time is guaranteed to be an upper bound on $\|x\|^2$ with overwhelming probability. Using Corollary 5, for all j it holds that $q \stackrel{\text{def}}{=} n(Ax)_j^2/(TB) \leq 1$ for a parameter B, so we can generate the z_j from a Bernoulli(q) distribution. Since T is halved in each iteration, for some iteration $\mathbf{E}[\sum_j z_j]$ will be large enough to ensure that E is tightly concentrated.

We now describe the protocol in detail. Set $\ell = \Theta(B)(1/\epsilon^2 \log(nM)\log(1/\delta) + k)$. In the following, if $q > 1$, then the distribution Bernoulli(q) means Bernoulli(1).

$\underline{\ell_2\text{-}\mathbf{Approx}}$ (a, b):

1. Alice, Bob exchange a seed of G and generate a random A as in Corollary 5
2. Set $T = T_{max}$
3. Repeat:
   (a) $\{$Assertion: $\|x\|^2 \le T$ $\}$
   (b) A secure circuit with ROM Aa, Ab computes the following
      - Generate random coordinates $i_1, \ldots, i_\ell$ and compute $(Ax)^2_{i_1}, \ldots (Ax)^2_{i_\ell}$
      - For $j \in [\ell]$, independently generate z_j from a Bernoulli$\Big(n(Ax)^2_{i_j}/(TB)\Big)$ distribution
   (c) T = T/2
4. Until $\sum_i z_i \ge \frac{\ell}{4B}$ or $T < 1$
5. Output $E = \frac{2TB}{l} \sum_i z_i$ as an estimate of $\|x\|^2$

Note that the protocol can be implemented in $O(1)$ rounds by parallelizing the secure circuit invocations.

Analysis: To show the correctness and privacy of our protocol, we start with the following lemma.

Lemma 6 *The probability that assertion 3a holds in every iteration of step 3 is* $1 - \mathrm{neg}(k, n)$*. Moreover, when the algorithm exits, with probability* $1 - \mathrm{neg}(k, n)$ *it holds that* $\mathbf{E}[\sum_j z_j] \ge \ell/(3B)$.

Proof: By Corollary 5, $\Pr_A[(1-2^{-B})\|x\|^2 \le \|Ax\|^2 \le \|x\|^2$ and $\forall_i (Ax)^2_i < \frac{\|x\|^2}{n} B] = 1 - \mathrm{neg}(k, n)$, so we may condition on this occurring. If $\|x\|^2 = 0$, then $\Pr[Ax = 0] = 1 - \mathrm{neg}(k, n)$, and thus $\Pr[E = 0] = 1 - \mathrm{neg}(k, n)$. Otherwise, $\|x\|^2 \ge 1$. Consider the smallest j for which $T_{max}/2^j < \|x\|^2$. We show for $T = T_{max}/2^{j-1} \ge \|x\|^2 \ge 1$ that $\Pr[\sum_j z_j < \ell/(4B)] = \mathrm{neg}(k, n)$. The assertion holds at the beginning of the jth iteration by our choice of T. Thus, $n(Ax)^2_i \le TB$ for all $i \in [n]$. So for all j, $\Pr[z_j = 1] = \frac{\|Ax\|^2}{TB} \ge (1 - 2^{-B})/(2B)$, and thus $\mathbf{E}[\sum_j z_j] \ge \ell/(3B)$. By a Chernoff bound, $\Pr[\sum_j z_j < \ell/(4B)] = \mathrm{neg}(k, n)$, so if ever $T = T_{max}/2^{j-1}$, then this is the last iteration with overwhelming probability. ■

Correctness: We show $\Pr[|E - \|x\|^2| \le \epsilon] \ge 1 - \delta$. By Lemma 6, when the algorithm exits, with probability $1 - \mathrm{neg}(k, n)$, $\mathbf{E}[\sum_i z_i] > \frac{\ell}{3B}$, so we assume this event occurs. By a Chernoff bound,

$$\Pr\left[\left|\sum_i z_i - E\left[\sum_i z_i\right]\right| \ge \frac{\epsilon}{2} E\left[\sum_i z_i\right] \mid \sum_i z_i \ge \frac{\ell}{4B}\right] \le e^{-\Theta\left(\epsilon^2 \frac{\ell}{B}\right)} < \frac{\delta}{2}$$

By Lemma 6, assertion 3a holds, so that

$$\ell(1 - 2^{-B})\|x\|^2 \ \le \ TB \cdot \mathbf{E}[\sum_i z_i] \ \le \ell\, \|x\|^2$$

Setting $E = \frac{2TB}{\ell} \sum_i z_i$ (recall that T is halved in step 3c) shows that $\Pr[|E - \|x\|^2 \ge \epsilon \|x\|^2] \le \delta$.

Privacy: We replace the secure circuit with ROM in step 3b of ℓ_2-Approx with an oracle (see section 2). We construct a single simulator Sim, which given $\Delta \stackrel{\mathrm{def}}{=} \|x\|^2$, satisfies $\mathsf{Sim}(\Delta) \stackrel{c}{\equiv} \mathtt{View}^{\pi}_{\mathtt{A}}(\mathtt{a}, \mathtt{b})$ and

$\mathsf{Sim}(\Delta) \stackrel{c}{\equiv} \mathtt{View}^{\pi}_{\mathtt{B}}(\mathtt{a}, \mathtt{b})$, where $\mathtt{View}^{\pi}_{\mathtt{A}}(\mathtt{a}, \mathtt{b}), \mathtt{View}^{\pi}_{\mathtt{B}}(\mathtt{a}, \mathtt{b})$ are Alice, Bob's real views respectively. This, in particular, implies functional privacy. It will follow that ℓ_2-Approx is a private approximation of Δ.

Sim (Δ):

1. Generate a random seed of G
2. Set $T = T_{max}$
3. Repeat:
   (a) For $j \in [\ell]$, independently generate z_j from a Bernoulli$(\Delta/(TB))$ distribution
   (b) $T = T/2$
4. Until $\sum_i z_i \geq \frac{\ell}{4B}$ or $T < 1$
5. Output $E = \frac{2TB}{l} \sum_i z_i$

With probability $1 - \text{neg}(k, n)$, the matrix A satisfies the property in Corollary 5, so we assume this event occurs. In each iteration, the random variables z_j are independent in both the simulation and the protocol. Further, the probabilities that $z_j = 1$ in the simulated and real views differ only by a multiplicative factor of $(1 - 2^{-B})$ as long as $T \geq \Delta$. But the probability that, in either view, we encounter $T < \Delta$ is $\text{neg}(k, n)$.

Complexity. Given our cryptographic assumptions, we use $\tilde{O}(1)$ communication and $O(1)$ rounds.

Remark 7 Our cryptographic assumptions are fairly standard, and similar to the ones in [12]. There the authors make the weaker assumptions that PRGs stretching n^γ bits to n bits and OT with n^γ communication exist for any constant γ. In fact, the latter implies the former [21, 15]. If we were to instead use these assumptions, our communication would be $O(n^\gamma)$, still greatly improving upon the $O(n^{1/2+\gamma})$ communication of [12]. A candidate OT scheme satisfying our assumptions can be based on the Φ-Hiding Assumption [6], and can be derived by applying the PIR to OT transformation of [29] to the scheme in that paper.

Remark 8 For the special case of Hamming distance, we have an alternative protocol based on the following idea. Roughly, both parties apply the perfect hashing of the Low-Distance Estimator protocol of [12] for a logarithmic number of levels j, where the jth level contains $\tilde{O}(2^j)$ buckets. To overcome the $\tilde{O}(\sqrt{n})$ barrier of [12], instead of exchanging the buckets, the set of buckets is randomly and obliviously sampled. From the samples, an estimate of $\Delta(a, b)$ is output. For some j, $2^j \approx \Delta(a, b)$, so the estimate will be tightly concentrated, and for reasons similar to ℓ_2-Approx, will be simulatable. We omit the details, but note that two advantages of this alternative protocol are that the time complexity will be $\tilde{O}(n)$ instead of $\tilde{O}(n^2)$, and that we don't need the PRG G, as we may use k-wise independence for the hashing.

4 Private near neighbor and c-approximate near neighbor problems

Here we consider the setting in which Alice has a point q, and Bob a set of n points P.

4.1 Private near neighbor problem

Suppose for some integer U, Alice has $q \in [U]^d$, Bob has $P = p_1, \ldots, p_n \in [U]^d$, and Alice should learn $\min_i f(q, p_i)$, where f is some distance function. In [10] protocols for ℓ_1, ℓ_2, Hamming distance over U-ary alphabets, set difference, and arbitrary distance functions $f(a,b) = \sum_{i=1}^{d} f_i(a_i, b_i)$ were proposed, using an untrusted third party. We improve the communication of these protocols and remove the third party using homomorphic encryption to implement polynomial evaluation as in [13], and various hashing tricks.

In [13], the authors consider the private all-near neighbors problem in which Alice has n queries $q_1, \ldots, q_n \in [U]^d$ and wants all p_i for which $\Delta(p_i, q_j) \leq t < d$ for some j and parameter t. Our techniques improve the $\tilde{O}(n^2 d)$ communication of a generic SFE and the $\tilde{O}(n\binom{d}{t})$ communication of [13] for this problem to $\tilde{O}(nd^2 + n^2)$. Finally, in the common random string model we achieve $\lceil \log d \rceil + O(k)$ communication for the (exact) Hamming distance, and an inner product protocol with $d + O(k)$ communication.

For the details of our schemes, see appendix B. We do not focus on them since they still suffer from an $\Omega(n)$ communication cost. We instead focus on how to privately approximate these problems.

4.2 Private c-approximate near neighbor problem

Suppose $q \in \{0,1\}^d$ and $p_i \in \{0,1\}^d$ for all i. Let $P_t = \{p \in P \mid \Delta(p,q) \leq t\}$, and $c > 1$ be a constant.

Definition 9 *A* **c-approximate NN protocol** *is correct if when $P_r \neq \emptyset$, Alice outputs a point $f(q,P) \in P_{cr}$ with probability $1 - 2^{-\Omega(k)}$. It is private if in the computational sense, Bob learns nothing, while Alice learns nothing except what follows from P_{cr}. Formally, Alice's privacy is implied by an efficient simulator Sim for which $\langle q, P, f(q,P)\rangle \stackrel{c}{\equiv} \langle q, P, Sim(1^n, P_{cr}, q)\rangle$ for $\mathrm{poly}(d,n,k)$-time machines.*

Following [20], we say the protocol *leaks b bits of information* if there is a deterministic "hint" function $h : \{0,1\}^{(n+1)d} \to \{0,1\}^b$ such that the distributions $\langle q, P, f(q,P)\rangle$ and $\langle q, P, Sim(1^n, P_{cr}, q, h(P,q))\rangle$ are indistinguishable. As motivated in section 1, we believe these to be natural extensions of private approximations in [12, 20] from values to sets of values.

We give a private c-approximate NN protocol with communication $\tilde{O}(\sqrt{n} + d)$ and a c-approximate NN protocol with communication $\tilde{O}(n^{1/3} + d)$ which leaks k bits of information. Both protocols are based on dimensionality reduction in the hypercube [24]. There it is shown that for an $O(\log n) \times d$ matrix A with entries i.i.d. Bernoulli($1/d$), there is an $\tau = \tau(r, cr)$ such that for all $p, q \in \{0,1\}^d$, the following event holds with probability at least $1 - 1/\mathrm{poly}(n)$

$$\text{If } \Delta(p,q) \leq r, \text{ then } \Delta(Ap, Aq) \leq \tau, \text{ and if } \Delta(p,q) \geq cr, \text{ then } \Delta(Ap, Aq) > \tau.$$

Here, arithmetic occurs in $\mathbb{Z}_2$. We use this idea in the following helper protocol $\mathsf{DimReduce}(\tau, B, q, P)$. Let A be a random matrix as described above. Let $S = \{p \in P \mid \Delta(Ap, Aq) \leq \tau\}$. If $|S| > B$, replace S with the lexicographically first B elements of S. $\mathsf{DimReduce}$ outputs random shares of S.

DimReduce (τ, B, q, P):

1. Bob performs the following computation
   - Generate a matrix A as above, and initialize L to an empty list.
   - For each $v \in \{0,1\}^{O(\log n)}$, let $L(v)$ be the first B p_i for which $\Delta(Ap_i, v) \leq \tau$.
2. A secure circuit with ROM L performs the following computation on input (q, A),
   - Compute Aq.
   - Lookup Aq in L to obtain S. If $|S| < B$, pad S so that all S have the same length.
   - Output random shares (S^1, S^2) of S so that $S = S^1 \oplus S^2$.

It is an easy exercise to show the correctness and privacy of DimReduce.

Remark 10 As stated, the communication is $\tilde{O}(dB)$. The dependence on d can be improved to $\tilde{O}(d+B)$ using homomorphic encryption. Roughly, Alice sends $E(q_1), \ldots, E(q_d)$ to Bob, who sets $L(v)$ to be the first B different $E(\Delta(p_i, q))$ for which $\Delta(Ap_i, v) \leq \tau$. Note that $E(\Delta(p_i, q))$ is efficiently computable, and has size $\tilde{O}(1) \ll d$.

It will be useful to define the following event $\mathcal{H}(r_1, r_2, P)$ with $r_1 < r_2$. Suppose we run DimReduce independently k times with matrices A_i. Then $\mathcal{H}(r_1, r_2, P)$ is the event that at least $k/2$ different i satisfy

$$\forall p \in P_{r_1},\ \Delta(A_i p, A_i q) \leq \tau(r_1, r_2) \text{ and } \forall p \in P \setminus P_{r_2},\ \Delta(A_i p, A_i q) > \tau(r_1, r_2).$$

The next lemma follows from the properties of the A_i and standard Chernoff bounds:

Lemma 11 $\Pr[\mathcal{H}(r_1, r_2, P)] = 1 - 2^{-\Omega(k)}$.

4.3 c-approximate NN protocol

Protocol Overview: Our protocol is based on the following intuition. When $|P_{cr}|$ is large, a simple solution is to run a secure function evaluation with Alice's point q as input, together with a random sample P' of roughly a $k/|P_{cr}|$ fraction of Bob's points P. The circuit returns a random point of $P' \cap P_{cr}$, which is non-empty with overwhelming probability. The communication is $\tilde{O}(n/|P_{cr}|)$.

On the other hand, when $|P_{cr}|$ is small, if Alice and Bob run DimReduce$(\tau(r, cr), |P_{cr}|, q, P)$ independently k times, then with overwhelming probability $P_r \subseteq \cup_i S_i$, where S_i denotes the (randomly shared) output in the ith execution. A secure function evaluation can then take in the random shares of the S_i and output a random point of P_r. The communication of this scheme is $\tilde{O}(|P_{cr}|)$.

Our protocol combines these two protocols to achieve $\tilde{O}(\sqrt{n})$ communication, by sampling roughly an $n^{-1/2}$ fraction of Bob's points in the first protocol, and by invoking DimReduce with parameter $B = \tilde{O}(\sqrt{n})$ in the second protocol. This approach is similar in spirit to the "high distance / low distance" approach used to privately approximate the Hamming distance in [12].

<u>c-**Approx** (q, P):</u>

1. Set $B = \tilde{O}(\sqrt{n})$.
2. Independently run DimReduce$(\tau(r, cr), B, q, P)$ k times, generating shares (S_i^1, S_i^2).
3. Bob finds a random subset P' of P of size B.
4. A secure circuit performs the following computation on inputs q, S_i^1, S_i^2, P'.
   - Compute $S_i = S_i^1 \oplus S_i^2$ for all i.
   - Let $f(q, P)$ be a random point from $P_{cr} \cap P' \neq \emptyset$ if it is non-empty,
   - Else let $f(q, P)$ be a random point from $P_r \cap \cup_i S_i$ if it is non-empty, else set $f(q, P) = \emptyset$.
   - Output $(f(q, P), \mathsf{null})$.

Using the ideas in Remark 10, the communication is $\tilde{O}(d + B)$, since the SFE has size $\tilde{O}(B)$. Let $\mathcal{F}$ be the event that $P' \cap P_{cr} \neq \emptyset$, and put $\mathcal{H} = \mathcal{H}(r, cr, P)$.

Correctness: Suppose P_r is nonempty. The probability s of correctness is just the probability we don't output $\emptyset$. Thus $s \geq \Pr[\mathcal{F}] + \Pr[\neg\mathcal{F}]\Pr[f(q,P) \neq \emptyset \mid \neg\mathcal{F}]$.

Case $|P_{cr}| \geq \sqrt{n}$: For sufficiently large B, we have $s \geq \Pr[\mathcal{F}] = 1 - 2^{-\Omega(k)}$.

Case $|P_{cr}| < \sqrt{n}$: It suffices to show $\Pr[f(q,P) \neq \emptyset \mid \neg\mathcal{F}] = 1 - 2^{-\Omega(k)}$. But this probability is at least $\Pr[f(q,P) \neq \emptyset \mid \mathcal{H}, \neg\mathcal{F}]\Pr[\mathcal{H}]$, and if $\mathcal{H}$ occurs, then $f(q,P) \neq \emptyset$. By Lemma 11, $\Pr[\mathcal{H}] = 1 - 2^{-\Omega(k)}$.

Privacy Note that Bob gets no output, so Alice's privacy follows from the composition of of DimReduce and the secure circuit protocol of step 5. Similarly, if we can construct a simulator Sim with inputs $1^n, P_{cr}, q$ so that the distributions $\langle q, P, f(q,P)\rangle$ and $\langle q, P, Sim(1^n, P_{cr}, q)\rangle$ are statistically close, Bob's privacy will follow by that of DimReduce and the secure circuit protocol of step 5.

Sim $(1^n, P_{cr}, q)$:

1. Set $B = \tilde{O}(n^{1/2})$.
2. With probability $1 - \binom{n-|P_{cr}|}{B}\binom{n}{B}^{-1}$, output a random element of P_{cr},
3. Else output a random element of P_r.

Let X denote the output of $Sim(1^n, P_{cr}, q)$. It suffices to show that for each $p \in P$, $|\Pr[f(q,P) = p] - \Pr[X = p]| = 2^{-\Omega(k)}$, since this also implies $|\Pr[f(q,P) = \emptyset] - \Pr[X = \emptyset]| = 2^{-\Omega(k)}$. We have

$$\begin{aligned}
\Pr\left[f(q,P) = p\right] &= \Pr\left[f(q,P) = p, \mathcal{F}\right] + \Pr\left[f(q,P) = p, \neg\mathcal{F}\right] \\
&= \Pr\left[f(q,P) = p, \mathcal{F}\right] + \Pr\left[f(q,P) = p, \neg\mathcal{F} \mid \mathcal{H}\right] \pm 2^{-\Omega(k)} \\
&= \Pr\left[\mathcal{F}\right]|P_{cr}|^{-1} + \Pr[\neg\mathcal{F}]\Pr[f(q,P) = p \mid \mathcal{H}, \neg\mathcal{F}] \pm 2^{-\Omega(k)},
\end{aligned}$$

where we have used Lemma 11. Since $\Pr[\mathcal{F}] = 1 - \binom{n-|P_{cr}|}{B}\binom{n}{B}^{-1}$, we have

$$|\Pr[f(q,P) = p] - \Pr[X = p]| \leq \Pr[\neg\mathcal{F}]\left|\Pr[f(q,P) = p \mid \mathcal{H}, \neg\mathcal{F}] - \delta(p \in P_r)|P_r|^{-1}\right| + 2^{-\Omega(k)}.$$

If $|P_{cr}| \geq \sqrt{n}$, then $\Pr[\neg\mathcal{F}] = 2^{-\Omega(k)}$. If $|P_{cr}| < \sqrt{n}$, then $\Pr[f(q,P) = p \mid \mathcal{H}, \neg\mathcal{F}] = \delta(p \in P_r)|P_r|^{-1}$.

Extensions: The way the current problem is stated, there is an $\Omega(d)$ lower bound. In appendix C we sketch how, if Alice just wants to learn some coordinate of an element of P_{cr}, this dependence can be made polylogarithmic. We also have a similar protocol based on locality-sensitive hashing (LSH), which only achieves $\tilde{O}(n^{1/2+1/(2c)} + d)$ communication, but has much smaller time complexity (though still polynomial). More precisely, the work of the LSH scheme is $n^{O(1)}$, whereas the work of c-**Approx** is $n^{O(1/(c-1)^2)}$, which is polynomial only for constant c. See Appendix D for the details.

4.4 c-approximate NN protocol leaking k bits

Protocol Overview: We consider three balls $P_r \subseteq P_{br} \subseteq P_{cr}$, where $c - b, b - 1 \in \Theta(1)$. We start by trying to use dimensionality reduction to separate P_r from $P \setminus P_{br}$, and to output a random point of P_r. If this fails, we try to sample and output a random point of P_{cr}. If this also fails, then it will likely hold that $n^{1/3} \leq |P_{br}| \leq |P_{cr}| \leq n^{2/3}$. We then sample down the pointset P by a factor of $n^{-1/3}$, obtaining $\tilde{P}$ with survivors $\tilde{P}_{br}, \tilde{P}_{cr}$ of P_{br}, P_{cr} respectively. It will now likely hold that we can use dimensionality reduction to separate $\tilde{P}_{br}$ from $\tilde{P} \setminus \tilde{P}_{cr}$ to obtain and output a random point of $\tilde{P}_{br}$. The hint function will encode the probability, to the nearest multiple of 2^{-k}, that the first dimensionality reduction fails, which may be a non-negligible function of $P \setminus P_{cr}$. This hint will be enough to simulate the entire protocol.

c-**ApproxWithHelp** (q, P):

1. Set $B = \tilde{O}(n^{1/3})$.
2. Independently run $\mathsf{DimReduce}(\tau(r, br), B, q, P)$ k times, generating shares (S_i^1, S_i^2).
3. Bob finds random subsets $P', \tilde{P}$ of P of respective sizes B and $n^{2/3}$.
4. Independently run $\mathsf{DimReduce}(\tau(br, cr), B, q, \tilde{P})$ k times, generating shares $(\tilde{S}_i^1, \tilde{S}_i^2)$.
5. A secure circuit performs the following computation on inputs $q, S_i^1, S_i^2, P', \tilde{S}_i^1, \tilde{S}_i^2$.
   - Compute $S_i = S_i^1 \oplus S_i^2$ and $\tilde{S}_i = \tilde{S}_i^1 \oplus \tilde{S}_i^2$ for all i.
   - If for most i, $|S_i| < B$, let $f(q, P)$ be a random point in $P_r \cap \cup_i S_i$, or $\emptyset$ if it is empty.
   - Else if $P_{cr} \cap P' \neq \emptyset$, let $f(q, P)$ be a random point in $P_{cr} \cap P'$.
   - Else let $f(q, P)$ be a random point in $P_{br} \cap \cup_i \tilde{S}_i$ if it is non-empty, otherwise set $f(q, P) = \emptyset$.
   - Output $(f(q, P), \mathsf{null})$.

The protocol can be implemented in polynomial time with communication $\tilde{O}(B + d) = \tilde{O}(n^{1/3} + d)$.

To prove correctness and privacy, we introduce some notation. Let $\mathcal{E}_1$ be the event that the majority of the $|S_i|$ are less than B, and $\mathcal{E}_2$ the event that $P_r \subseteq \cup_i S_i$. Let $\mathcal{F}$ be the event that $P' \cap P_{cr} \neq \emptyset$. Let $\mathcal{G}_1$ be the event that $1 \leq \tilde{P}_{br} \leq \tilde{P}_{cr} \leq B$ and $\mathcal{G}_2$ the event that $\tilde{P}_{br} \subseteq \cup_i \tilde{S}_i$. Finally, let $\mathcal{H}_1 = \mathcal{H}(r, br, P)$ and $\mathcal{H}_2 = \mathcal{H}(br, cr, \tilde{P})$. Note that $\Pr[\mathcal{H}_1], \Pr[\mathcal{H}_2]$ are $1 - 2^{-\Omega(k)}$ by Lemma 11. We need two lemmas:

Lemma 12 $\Pr[\mathcal{E}_2 \mid \mathcal{E}_1] = 1 - 2^{-\Omega(k)}$.

Proof: If $\mathcal{H}_1$ and $\mathcal{E}_1$ occur, then there is an i for which $P_r \subseteq S_i$, so $\mathcal{E}_2$ occurs. ■

Lemma 13 $\Pr[\mathcal{G}_2 \mid \mathcal{G}_1] = 1 - 2^{-\Omega(k)}$.

Proof: If $\mathcal{H}_2$ and $\mathcal{E}_2$ occur, then the majority of the $\tilde{S}_i$ contain $\tilde{P}_{br}$, so $\mathcal{G}_2$ occurs. ■

Correctness: We may assume $P_r \neq \emptyset$. The probability s of correctness is just the probability the algorithm doesn't return $\emptyset$. Since $\mathcal{F}, \mathcal{E}_1$, and $\mathcal{G}_1$ are independent,

$$s \geq \Pr[\mathcal{E}_1] \Pr[\mathcal{E}_2 \mid \mathcal{E}_1] + \Pr[\neg\mathcal{E}_1](\Pr[\mathcal{F}] + \Pr[\neg\mathcal{F}] \Pr[\mathcal{G}_1] \Pr[\mathcal{G}_2 \mid \mathcal{G}_1]).$$

Case $|P_{br}| < B$: $\mathcal{H}_1$ implies $\mathcal{E}_1$ since $|P_{br}| < B$, and using Lemma 12, $s \geq \Pr[\mathcal{E}_1] \Pr[\mathcal{E}_2 \mid \mathcal{E}_1] = 1 - 2^{-\Omega(k)}$.

Case $|P_{br}| \geq B$: Since $\Pr[\mathcal{E}_2 \mid \mathcal{E}_1] = 1 - 2^{-\Omega(k)}$ by Lemma 12, we just need to show that $\Pr[\mathcal{F}] + \Pr[\neg\mathcal{F}] \Pr[\mathcal{G}_1] \Pr[\mathcal{G}_2 \mid \mathcal{G}_1] = 1 - 2^{-\Omega(k)}$. If $|P_{cr}| > n^{2/3}$, it suffices to show $\Pr[\mathcal{F}] = 1 - 2^{-\Omega(k)}$. This holds for large enough $B = \tilde{O}(n^{1/3})$. Otherwise, if $|P_{cr}| \le n^{2/3}$, then it suffices to show $\Pr[\mathcal{G}_1]\Pr[\mathcal{G}_2 \mid \mathcal{G}_1] = 1-2^{-\Omega(k)}$. By assumption, $B \le |P_{br}| \le |P_{cr}| \le n^{2/3}$. Therefore, for large enough B, $\Pr[\mathcal{G}_1] = 1-2^{-\Omega(k)}$, and thus by Lemma 13, $\Pr[\mathcal{G}_1]\Pr[\mathcal{G}_2 \mid \mathcal{G}_1] = 1-2^{-\Omega(k)}$.

Privacy: Note that Bob gets no output, so Alice's privacy follows from the composition of DimReduce and the secure circuit protocol of step 5. Similarly, if we can construct a simulator Sim with inputs $1^n, P_{cr}, q, h(P_{cr}, q)$ so that the distributions $\langle q, P, f(q,P)\rangle$ and $\langle q, P, Sim(1^n, P_{cr}, q, h(P_{cr}, q))\rangle$ are statistically close, Bob's privacy will follow by that of DimReduce and the secure circuit of step 5.

We define the hint function $h(P_{cr}, q)$ to output the nearest multiple of 2^{-k} to $\Pr[\mathcal{E}_1]$. In the analysis we may assume that Sim knows $\Pr[\mathcal{E}_1]$ exactly, since its output distribution in this case will be statistically close to its real output distribution.

Sim $(1^n, P_{cr}, q, \Pr[\mathcal{E}_1])$:

1. Set $B = \tilde{O}(n^{1/3})$.
2. With probabiity $\Pr[\mathcal{E}_1]$, output a random element of P_r, or output $\emptyset$ if $P_r = \emptyset$.
3. Else with probability $1 - \binom{n-|P_{cr}|}{B}\binom{n}{B}^{-1}$, output a random element of P_{cr},
4. Else output a random element of P_{br}.

Let X denote the output of $Sim(1^n, P_{cr}, q, \Pr[\mathcal{E}_1])$. It suffices to show that for each $p \in P$,

$$|\Pr[f(q,P) = p] - \Pr[X = p]| = 2^{-\Omega(k)},$$

since then we have $|\Pr[f(q,P) = \emptyset] - \Pr[X = \emptyset]| = 2^{-\Omega(k)}$. Using the independence of $\mathcal{F}, \mathcal{E}_1, \mathcal{G}_1$, and Lemmas 12, 13, we bound $\Pr[f(q,P) = p]$ as follows

$$\begin{aligned}
& \Pr[f(q,P)=p] = \Pr[\mathcal{E}_1, f(q,P)=p] + \Pr[\neg\mathcal{E}_1, f(q,P)=p] \\
= \; & \Pr[\mathcal{E}_1]\Pr[f(q,P)=p \mid \mathcal{E}_2\mathcal{E}_1] \pm 2^{-\Omega(k)} + \Pr[\neg\mathcal{E}_1]\Pr[\mathcal{F}]\Pr[f(q,P)=p \mid F, \neg\mathcal{E}_1] \\
+ \; & \Pr[\neg\mathcal{E}_1]\Pr[\neg\mathcal{F}]\Pr[f(q,P)=p \mid \neg F, \neg\mathcal{E}_1] \\
= \; & \Pr[\mathcal{E}_1]|P_r|^{-1}\delta(p \in P_r) \pm 2^{-\Omega(k)} + \Pr[\neg\mathcal{E}_1]\Pr[\mathcal{F}]|P_{cr}|^{-1} \\
+ \; & \Pr[\neg\mathcal{E}_1]\Pr[\neg\mathcal{F}]\Pr[\mathcal{G}_1]\Pr[f(q,P)=p \mid \mathcal{G}_1\mathcal{G}_2\neg\mathcal{F}\neg\mathcal{E}_1] \pm 2^{-\Omega(k)} \\
+ \; & \Pr[\neg\mathcal{E}_1]\Pr[\neg\mathcal{F}]\Pr[\neg\mathcal{G}_1]\Pr[f(q,P)=p \mid \neg\mathcal{G}_1\neg\mathcal{F}\neg\mathcal{E}_1] \\
= \; & \Pr[\mathcal{E}_1]|P_r|^{-1}\delta(p \in P_r) + \Pr[\neg\mathcal{E}_1]\Pr[\mathcal{F}]|P_{cr}|^{-1} + \Pr[\neg\mathcal{E}_1]\Pr[\neg\mathcal{F}]\Pr[\mathcal{G}_1]|P_{br}|^{-1}\delta(p \in P_{br}) \\
+ \; & \Pr[\neg\mathcal{E}_1]\Pr[\neg\mathcal{F}]\Pr[\neg\mathcal{G}_1]\Pr[f(q,P)=p \mid \neg\mathcal{E}_1\neg\mathcal{F}\neg\mathcal{G}_1] \pm 2^{-\Omega(k)}.
\end{aligned}$$

On the other hand, since $\Pr[\mathcal{F}] = 1 - \binom{n-|P_{cr}|}{B}\binom{n}{B}^{-1}$, we have

$$\Pr[X=p] = \Pr[\mathcal{E}_1]|P_r|^{-1}\delta(p \in P_r) + \Pr[\neg\mathcal{E}_1]\Pr[\mathcal{F}]|P_{cr}|^{-1} + \Pr[\neg\mathcal{E}_1]\Pr[\neg\mathcal{F}]|P_{br}|^{-1}\delta(p \in P_{br}),$$

so that

$$|\Pr[f(q,P)=p] - \Pr[X=p]| \le \Pr[\neg\mathcal{E}_1]\Pr[\neg\mathcal{F}]\Pr[\neg\mathcal{G}_1]\Pr[f(q,P)=p \mid \neg\mathcal{E}_1\neg\mathcal{F}\neg\mathcal{G}_1] + 2^{-\Omega(k)}.$$

If $|P_{br}| < B$, $\Pr[\neg\mathcal{E}_1] = 2^{-\Omega(k)}$. If $|P_{cr}| \ge n^{2/3}$, $\Pr[\neg\mathcal{F}] = 2^{-\Omega(k)}$. Otherwise $B \le |P_{br}| \le |P_{cr}| \le n^{2/3}$, and as shown for correctness, $\Pr[\neg\mathcal{G}_1] = 2^{-\Omega(k)}$, which shows $|\Pr[f(q,P)=p] - \Pr[X=p]| = 2^{-\Omega(k)}$.

References

[1] N. Alon, Y. Matias, and M. Szegedy. The space complexity of approximating the frequency moments. In *Proceedings of the 28th Annual ACM Symposum on the Theory of Computing*, p. 20-29, 1996.

[2] K. Bharat and A. Broder. *Estimating the Relative Size and Overlap of Public Web Search Engines*. Proc. WWW 7, 1998.

[3] A. Beimel, Y. Ishai, T. Malkin. *Reducing the Servers Computation in Private Information Retrieval: PIR with Preprocessing.* Proc. of the 20th Annual IACR Crypto conference (CRYPTO '00).

[4] J. D. C. Benaloh, *Verifiable Secret-Ballot Elections.* PhD thesis, Yale University, 1987.

[5] C. Cachin, J. Camenisch, J. Kilian and J. Müller. *One-round secure computation and secure autonomous mobile agents.* In Ugo Montanari, Jos P. Rolim, and Emo Welzl, editors, Proc. 27th International Colloquium on Automata, Languages and Programming (ICALP), volume 1853 of Lecture Notes in Computer Science, pages 512-523. Springer, 2000.

[6] C. Cachin, S. Micali and M. Stadler. *Computationally private information retrieval with polylogarithmic communication.* In Advances in Cryptology – Eurocrypt '99.

[7] R. Canetti, Y. Lindell, R. Ostrovsky, and A. Sahai. *Universally Composable Two-party Computation.* In STOC, 2002.

[8] B. Chor, N. Gilboa and M. Naor, *Private Information Retrieval by Keywords*, TR CS0917, Department of Computer Science, Technion, 1997.

[9] B. Chor, O. Goldreich, E. Kushilevitz and M. Sudan. *Private information retrieval.* In proceedings of FOCS '95.

[10] W. Du and M. J. Attalah. Protocols for Secure Remote Database Access with Approximate Matching. In *the 7th ACM CCS, The First Workshop on Security and Privacy in E-commerce*, 2000.

[11] S. Even, O. Goldreich and A. Lempel. *A randomized protocol for signing contracts.* In Communications of the ACM, 1985.

[12] J. Feigenbaum, Y. Ishai, T. Malkin, K. Nissim, M. Strauss, and R. Wright. Secure Multiparty Computation of Approximations. Proc. of the 28th International Colloquium on Automata, Languages and Programming (ICALP '01).

[13] M. Freedman, K. Nissim and B. Pinkas. Efficient Private Matching and Set Intersection. In *Advances in Cryptology – Eurocrypt '2004 Proceedings*, LNCS 3027, Springer-Verlag, pp. 1-19, May 2004.

[14] B. Goethals, S. Laur, H. Lipmaa, and T. Mielikainen. On Secure Scalar Product Computation for Privacy-Preserving Data Mining. In proceedings of ICISC, 2004.

[15] J. Hastad, R. Impagliazzo, L. A. Levin, and M. Luby. Construction of a pseudo-random generator from any one-way function. Technical Report TR-91-068, International Computer Science Institute, 1991.

[16] Y. Gertner, Y. Ishai, E. Kushilevitz and T. Malkin. *Protecting data privacy in private information retrieval schemes.* In proceedings of STOC '98.

[17] O. Goldreich. *Secure Multi-Party Computation*, 1998. Available at http://philby.ucsd.edu/

[18] O. Goldreich, S. Micali, and A. Wigderson. *How to Play Any Mental Game.* In proceedings of 19th STOC, pp. 218-229, 1987.

[19] S. Goldwasser and S. Micali. *Probabilistic encryption.* JCSS, pp.270-299, 1984.

[20] S. Halevi, R. Krauthgamer, E. Kushilevitz, and K. Nissim. Private approximation of NP-hard functions. Proc of STOC '01.

[21] R. Impagliazzo and M. Luby. One-way functions are essential for complexity-based cryptography.

[22] P. Indyk. *High-dimensional computational geometry*. PhD Thesis, Stanford University, 2000.

[23] P. Indyk and R. Motwani. *Approximate Nearest Neighbors: Towards Removing the Curse of Dimensionality*, In proceedings of STOC '98.

[24] E. Kushilevitz, R. Ostrovsky and Y. Rabani. *Efficient search for approximate nearest neighbor in high dimensional spaces*, In proceedings of STOC'98.

[25] Y. Lindell and B. Pinkas. Privacy Preserving Data Mining. In *Advances in Cryptology – Crypto '2000 Proceedings*, LNCS 1880, Springer-Verlag, pp. 20-34, August 2000.

[26] V.D. Milman and G. Schechtman, *Asymptotic Theory of Finite Dimensional Normed Spaced*, Lecture Notes in Mathematics, **1200**, Springer Verlag, 1986.

[27] D. Naccache and J. Stern. *A new public key cryptosystem.* In Advances in Cryptology – Eurocrypt 1997, pp.27-36.

[28] M. Naor and K. Nissim. *Communication Complexity and Secure Function Evaluation.* In proceedings of STOC 2001.

[29] M. Naor and B. Pinkas. *Oblivious transfer and polynomial evaluation.* In proceedings of STOC 1999.

[30] P. Paillier. *Public-key cryptosystems based on composite degree residuaosity classes.* In Advances in Cryptology – Eucrocrypt 1999, pp. 223-238.

[31] M. Rabin. *How to exchange secrets by oblivious transfer.* Tech report TR 81, Aiken Computation Lab, 1981.

[32] A. C. Yao. *Protocols for secure computations.* In proc. of 23rd FOCS, 1982, pp. 160-164. 16.

A Cryptographic Tools

We write $\mathrm{negl}(k, n)$ to denote an arbitrary negligible function of k, n, that is a function which shrinks faster than any inverse polynomial in n, k.

A.1 Homomorphic Encryption

An encryption scheme, $E : (G_1, +) \to (G_2, \cdot)$ is homomorphic if for all $a, b \in G_1$, $E(a+b) = E(a) \cdot E(b)$. For more background on this primitive see, for example, [19, 27].

We make use of the Paillier homomorphic encryption scheme [30] in some of our protocols and so we briefly repeat it here:

1. **Initialize:** Choose two primes, p and q and set $N = p \cdot q$. Let $\lambda = lcm(p-1, q-1)$. Let the public key $PK = (N, g)$ where the order of g is a multiple of N. Let the secret key, $SK = \lambda$.
2. **Encrypt:** Given a message $M \in Z_N$, choose a random value $x \in Z_N^*$. The encryption of M is, $E(M) = g^M x^N modN^2$.
3. **Decrypt:** Let $L(u) = \frac{(u-1)}{N}$, where u is congruent to 1 modulo N.To recover M from $E(M)$ calculate, $\frac{L(E(M)^\lambda mod\, N^2)}{L(g^\lambda mod\, N^2)} mod\, N$.

In [30] it's shown that the Paillier encryption scheme's semantic security is equivalent to the Decisional Composite Residuosity Assumption. The following shows homomorphy:

$$E(M_1) \cdot E(M_2) = (g^{M_1} {x_1}^N\, modN^2) \cdot (g^{M_2} {x_2}^N\, modN^2) = g^{M_1+M_2}(x_1 x_2)^N\, modN^2 = E(M_1 + M_2).$$

A.2 Oblivious Transfer and SPIR

Oblivious transfer is equivalent to the notion of symmetrically-private information retrieval (SPIR), where the latter usually refers to communication-efficient implementations of the former. SPIR was introduced in [16]. With each invocation of a SPIR protocol a user learns exactly one bit of a binary database while giving the server no information about which bit was learned. We rely on single-server SPIR schemes in our protocols. Such schemes necessarily offer computational, rather than unconditional, security [9]. Applying the transformation of [29] to the PIR scheme of [6] give SPIR constructions with $\tilde{O}(n)$ server work and $\tilde{O}(1)$ communication.

One issue is that in some of our schemes, we actually perform OT on *records* rather than on bits. It is a simple matter to convert a binary OT scheme into an OT scheme on records by running r invocations of the binary scheme in parallel, where r is the record size. This gives us a 1-round, $\tilde{O}(r)$ communication, $\tilde{O}(nr)$ server work OT protocol on records of size r. The dependence on r can be improved using techniques of [8].

A.3 Secure Function Evaluation

In [18, 32] it is shown how two parties holdings inputs x and y can privately evaluate any circuit C with communication $O(k(|C| + |x| + |y|))$, where k is a security parameter. In [5] it is shown how to do this in one round for the semi-honest case we consider. The time complexity is the same as the communication. We use such protocols as black boxes in our protocols.

B Private Near Neighbor and All-Near Neighbors

B.1 Private near neighbor for ℓ_2 and Hamming distance

Alice has $q \in [U]^d$, and Bob a set of points $P = p_1, \ldots, p_n$ in $[U]^d$. Alice should output argmin$_i \sum_j |p_{i,j} - q_j|^2$. The protocol is easily modified to return the p_i realizing the minimum. We assume a semantically secure homomorphic encryption scheme E such as Paillier encryption (see appendix A), that the message domain is isomorphic to $\mathbb{Z}_m$ for some m, and that m is large enough so that arithmetic is actually over $\mathbb{Z}$.

Exact-$\ell_2(q, P)$:

1. Alice generates (PK, SK) for E and sends PK, $E(q_1), \ldots, E(q_d)$ to Bob
2. For all i, Bob computes (by himself) $z_i = E(\langle q, p_i \rangle)$ and $v_i = \|p_i\|^2$
3. A secure circuit with inputs $q, SK, \{z_i\}_i$, and $\{v_i\}_i$ computes
    - $\langle q, p_i \rangle = D_{SK}(z_i)$ for all i
    - Return $\text{argmin}_i(v_i - 2\langle q, p_i \rangle)$

Using the homomorphy of E and the $\tilde{O}(n)$-sized circuit in step 3, we make the communication $\tilde{O}(n + d)$ rather than the $\tilde{O}(nd)$ of a generic SFE. The correctness is easy to verify. Using theorem 1 and the semantic security of E, privacy is just as easy to show. We note a natural extension to ℓ_p distances: Alice sends

$$\{E(q_{i_1})\}, \{E(q_{i_1} q_{i_2})\}, \ldots, \{E(q_{i_1} \cdots q_{i_{p-1}})\},$$

where $i_1, \ldots, i_{p-1}$ range over all of $[d]$. The communication is $\tilde{O}(n + d^{p-1})$, which is interesting for $d = O(n^{1/(p-2)})$.

B.2 Private near neighbor for generic distance functions

Now Alice wants $\min_i f(q, p_i)$ for an arbitrary $f(a, b) = \sum_{i=1}^{d} f_i(a_i, b_i)$. We use homomorphic encryption to implement polynomial evaluation as in [13].

Exact-Generic(q, P):

1. Alice creates d degree-$(U-1)$ polynomials s_j by interpolating from $s_j(u) = f_j(p_j, u)$ for all $u \in [U]$
2. Alice generates (PK, SK) for E and sends the encrypted coefficients of the s_j and PK to Bob
3. Bob computes (by himself) $z_i = E(\sum_j s_j(p_{i,j})) = E(f(q, p_i))$ for all i
4. A secure circuit with inputs $SK, \{z_i\}_i$ outputs $\text{argmin}_i D_{SK}(z_i)$

The proofs are similar to those of the previous section and are omitted. The communication here is $\tilde{O}(dU + n)$, improving the $O(ndU)$ communication of [10]. A special case of the result in section B.4 improves this to $\tilde{O}(d^2 + n)$ in case $f(a, b)$ is Hamming distance and $U > d$.

B.3 Private near neighbor for $n = 1$

We now show how Alice, holding $q \in \{0, 1\}^d$, and Bob, holding $p \in \{0, 1\}^d$ for some prime d, can privately compute $\Delta(q, p)$ with communication $d\lceil \log d \rceil + O(k)$. This extends to solve the private near neighbor problem for $n = 1$ with communication $2d\lceil \log d \rceil + \tilde{O}(k)$. The communication outperforms the $\Theta(dk)$ communication of SFE.

We assume both parties have access to the same uniformly random string. We need a homomorphic encryption whose message domain can be decoupled from its security parameter. Recall in Paillier encryption that if encryptions are k bits long, messages are about $k/2$ bits long. For low communication we want the domain to be very small, that is, roughly d elements instead of $2^{k/2}$. To do this, we use a Benaloh encryption scheme E [4], which is homomorphic and semantically secure assuming the prime residuousity assumption. The message domain is $\mathbb{Z}_d$ while encryptions are of size k.

Exact-1(q, p):

1. Alice generate (PK, SK) for E, and sends PK to Bob
2. Both parties interpret [3] the common random string R as d encryptions $E(z_i)$
3. Alice obtains the z_i by decrypting, and sends Bob $s_i = q_i - z_i \bmod d$ for all i
4. Bob computes (by himself) $E(z_i + q_i) = E(q_i)$ and $E(\sum_{i=1}^{d}(p_i + (-1)^{p_i} q_i)) = E(\Delta(p, q))$
5. Bob rerandomizes the $E(\Delta(p, q))$
6. Alice outputs $D_{SK}(E(\Delta(p, q))) = \Delta(x, y)$

The correctness of the protocol is straightforward. The key property for security is that if R is uniformly random, then for any PK, SK, the $E(z_1), \ldots, E(z_d)$ are independent uniformly random encryptions of random elements $z_1, \ldots, z_d \in [d]$.

To see complexity $d\lceil \log d \rceil + o(d)$, the list of s_i's that Alice sends has length $d\lceil \log d \rceil$. Also, $E(\Delta(q, p))$ has length k, the security parameter, which can be set to d^{ϵ} for any $\epsilon > 0$. Similar techniques give $d + O(k)$ communication for private inner product, using GM-encryption [19].

B.4 Private All-Near Neighbors

We consider the setting of [13], in which Alice and Bob have $Q = q_1, \ldots, q_n \in [U]^d$ and $P = p_1, \ldots, p_n \in [U]^d$ respectively, and Alice wants all p_j for which $\Delta(q_i, p_j) \leq t < d$ for some $i \in [n]$ and parameter t. We assume a semantically secure homomorphic encryption scheme E and OT with $\mathrm{polylog}(n)$ communication.

All-Near(Q, P):

1. The parties randomly permute their points
2. Alice generates parameters (PK, SK) of E and sends Bob PK
3. For $l = 1, \ldots, k$,
   - The parties choose a pairwise independent hash function $h : [U] \to [2d]$
   - For $i \in [n]$, Alice computes $\tilde{x}_i = h(x_i)$, where h is applied coordinate-wise
   - Replace each entry j of each $\tilde{x}_i$ with a length $2d$ unit vector with rth bit 1 iff $\tilde{x}_{i,j} = r$
   - Bob forms $\hat{y}_i$ similarly
   - Alice sends the coordinate-wise encryption of each vector for each coordinate of each $\tilde{x}_i$
   - Bob computes (by himself) $Z_{i,j,l} = E(\Delta(\tilde{x}_i, \tilde{y}_j))$ for all $i, j \in [n]$
4. A secure circuit with inputs $SK, Z_{i,j,l}$ computes
   - $Z_{i,j} = \min_l D_{SK}(Z_{i,j,l})$
   - Output $Z = \{j \mid \exists i \text{ s.t. } Z_{i,j} \geq d - t\}$ to Alice
5. Perform OT on records of size d for Alice to retrieve $Y = \{y_j \mid j \in Z\}$

Theorem 14 *The above is a private all-near neighbors protocol with communication $\tilde{O}(nd^2 + n^2)$.*

Proof: We first argue correctness, which means showing $\Pr[Y = \{y_j \mid \exists i \text{ s.t. } \Delta(q_i, p_j) \leq t\}] = 1 - 2^{-\Omega(k)}$. We show for $i, j \in [n]$, $\Pr[\Delta(q_i, p_j) = n - Z_{i,j}] = 1 - 2^{-\Omega(k)}$. By a union bound, for any h,

$$\Pr[D(Z_{i,j}) = n - \Delta(q_i, p_j)] \geq T/2T = 1/2.$$

But $D(Z_{i,j}) \geq n - \Delta(q_i, p_j)$ since hashing only increases the number of agreements. Thus, $\Pr[\min_l D(Z_{i,j,l}) > n - \Delta(q_i, p_j)] < 2^{-\Omega(k)}$, so that $Z_{i,j} = n - \Delta(q_i, p_j)$ with the required probability.

For privacy, since the output assumes a specific value with probability $1 - 2^{-\Omega(k)}$, we just need to show each party's view is simulatable. As usual, we replace the SFE and OT by oracles. Alice's output from the SFE is a list of random indices, and her output from the OT is her protocol output. Hence, her simulator just outputs a list of $|Y|$ random indices. Bob's simulator chooses k random hash functions and $2d^2nk$ encryptions of 0 under E. By the semantic security of E and theorem 1, the protocol is secure.

To see that the communication is $\tilde{O}(nd^2 + n^2)$, in each of k executions, Alice sends $O(nd^2)$ encryptions. Bob then inputs $O(n^2)$ encryptions to the SFE, which can be implemented with a circuit of size $\tilde{O}(n^2)$. Step 5 of the protocol can be done with $\tilde{O}(nd)$ communication using the best OT schemes (see [8, 6]). ■

Remark 15 A simple modification of the protocol gives the promised $\tilde{O}(d^2 + n)$ communication for Hamming distance in the setting of [10] for any U.

Remark 16 The protocol can be adapted to give $\tilde{O}(d + n)$ communication for set difference. In this case Alice has a single vector q. The idea is that Alice, Bob can hash their entries down to $2d$ values using h as in the protocol, and now Alice can homomorphically encrypt and send the coefficients of a degree-$(2d - 1)$ polynomial pol, where pol is such that $pol(t) = 0$ if $t \in \{r \mid \exists i \text{ s.t. } r = h(q_i)\}$ and $pol(t) = 1$

otherwise. Bob can evaluate pol on each (hashed) coordinate of each p_i and use E's homomorphy to compute $E(f(\tilde{q}, \tilde{p}_i))$, f denoting set difference. We then repeat this k times over different h and take a maximum in the SFE. Since coordinate order is immaterial for set difference, we achieve $\tilde{O}(n + d)$ instead of $\tilde{O}(n + d^2)$ communication.

Although we have improved the communication of [13], one may worry about the work the parties need to perform. We have the following optimization:

Theorem 17 *The protocol can be implemented with total work $\tilde{O}(n^2 d^{2c-4})$, where $c \approx 2.376$ is the exponent of matrix multiplication.*

Proof: The work is dominated by step 3, in which Bob needs to compute encryptions of all pairwise Hamming distances. To reduce the work, we think of what Alice sends as an encrypted $n \times d^2$ matrix M_1, and that Bob has a $d^2 \times n$ matrix M_2 and needs an encrypted $M_1 M_2$. It is shown in [3] that even the best known matrix multiplication algorithm still works if one of the matrices is homomorphically encrypted. Thus Bob can perform $(n/d^2)^2$ fast multiplications of $d^2 \times d^2$ matrices, requiring $\tilde{O}((n/d^2)^2 (d^2)^r) = \tilde{O}(n^2 d^{2r-4})$ work, which improves upon the $\tilde{O}(n^2 d^2)$ work of a naive implementation. ■

C Reducing the dependence on d for private c-approximate NN

Here we sketch how the communication of the protocol of section 4.3 can be reduced to $\tilde{O}(n^{1/2} + \text{polylog}(d))$ if Alice just wants to privately learn some coordinate of some element of P_{cr}.

Proof Sketch: The idea is to perform an approximation to the Hamming distance instead of using the $E(\Delta(p_i, q))$ in the current protocol (see, e.g., DimReduce, and the following remark). The approximation we use is that given in [24], namely, the parties will agree upon random matrices A_i for some subset of i in $[n]$, and from the $A_i p_i$ and $A_i q$ will determine $(1 \pm \epsilon)$ approximations to the $\Delta(p_i, q)$ with probability $1 - 2^{-k}$. We don't need private approximations since the parties will not learn these values, but rather, they will input the $A_i p_i, A_i q$ into a secure circuit which makes decisions based on these approximations.

More precisely, Bob samples B of his vectors p_i, and in parallel agrees upon B matrices A_i and feeds the $A_i p_i$ into a secure circuit. Alice feeds in the $A_i q$. Let $c \geq 1 + 8\epsilon$. The circuit looks for an approximation of at most $r(1 + 6\epsilon)$. If such a value exists, the circuit gives Alice the corresponding index. Observe that if $|P_{r(1+4\epsilon)}| > \sqrt{n}$, then with probability $1 - 2^{-k}$ an index is returned to an element in P_{cr}, and that this distribution is simulatable. So assume $|P_{r(1+4\epsilon)}| \leq \sqrt{n}$.

The parties proceed by performing a variant of $\text{DimReduce}(\tau(r, r(1+4\epsilon)), B, q, P)$, with the important difference being that the output no longer consists of shares of the $E(\Delta(p_i, q))$. Instead, for each entry $L(v)$, Bob pretends he is running the approximation of [24] with Alice's point q. That is, the parties agree on B different matrices A_i and Bob computes $A_i p$ for each $p \in L(v)$. A secure circuit obtains these products, and computes the approximations. It outputs an index to a random element with approximation at most $r(1 + 2\epsilon)$. If P_r is nonempty, such an index will exist with probability $1 - 2^{-k}$. Also, the probability that an index to an element outside of $P_{r(1+4\epsilon)}$ is returned is less than 2^{-k}, and so the distribution of the index returned is simulatable.

Finally, given the index of some element in P_{cr}, the parties perform OT and Alice obtains the desired coordinate, The communication is now $\tilde{O}(\sqrt{n})$. □

D Private c-approximate NN based on locality sensitive hashing

We give an alternative private c-approximate NN protocol, with slightly more communication than that in section 4.2, but less work (though still polynomial). It is based on locality sensitive hashing (LSH) [23]. The fact we need is that there is a family of functions $\mathcal{G} : \{0,1\}^d \to \{0,1\}^{\tilde{O}(1)}$ such that each $g \in \mathcal{G}$ has description size $\tilde{O}(1)$, and $\mathcal{G}$ is such that for all $p, q \in \{0,1\}^d$,

$$\Pr_{g \in \mathcal{G}}[g(p) = g(q)] = \Theta\left(n^{-\Delta(p,q)/cr}\right)$$

Recall that Alice has a point $q \in \{0,1\}^d$ and Bob has n points $P \subseteq \{0,1\}^d$. For correctness, Alice should learn a point of P_{cr} provided $P_r \neq \emptyset$. For privacy, her view should be simulatable given only P_{cr}.

Our protocol is similar to that in section 4.2. When $|P_{cr}|$ is large, one can run a secure function evaluation with Alice's point q as input, together with a random sample P' of roughly a $k/|P_{cr}|$ fraction of Bob's points P. The circuit returns a random point of $P' \cap P_{cr}$ which is non-empty with probabiity $1 - 2^{-\Omega(k)}$. The communication is $\tilde{O}(n/|P_{cr}|)$.

On the other hand, when $|P_{cr}|$ is small, if Alice and Bob exchange functions g_i independently $\tilde{O}(n^{1/c})$ times, then with overwhelming probability $P_r \subseteq \cup_i S_i$, where S_i denotes the subset of Bob's points p with $g_i(p) = g_i(q)$. Using a secure ciruit with ROM, we can obtain these sets S_i, and output a random point of P_r. The communication is $\tilde{O}(n^{1/c}|P_{cr}|)$.

Our protocol balances these approaches to achieve $\tilde{O}(n^{1/2+1/(2c)})$ communication.

There are a few technicalities dodged by this intuition. First, even though the parties exchange $\tilde{O}(n^{1/c})$ different g_i, and can thus guarantee that each p is in some S_i with probability $1 - 2^{-\Omega(k)}$, it may be that whenever $p \in S_i$, many points from $P \setminus P_{cr}$ also land in S_i, so that S_i is very large. Even though we only expect $|P \setminus P_{cr}|O(1/n) = O(1)$ points from $P \setminus P_{cr}$ in S_i, since $\Pr[p \in S_i] = \Theta(n^{-1/c})$ is small, p may only be in S_i when S_i is large. Because the size of the S_i affects the communication of our protocol, we cannot always afford for the ROM to receive the whole S_i (sometimes we will truncate it). However, in the analysis, we show that the average S_i is small, and this will be enough to get by with low communication.

Second, we need to extend the notion of a lookup gate given in section 2. Instead of just mapping inputs (i, j) to output $R_i[j]$, the jth entry in the ith party's ROM, we also allow j to be a key, so that the output is the record in R_i keyed by j. This can be done efficiently using [8], and Theorem 3 is unchanged, assuming the length of the keys is $\tilde{O}(1)$.

LSH (q, P):

1. Set $B = \tilde{O}(n^{1/2+1/(2c)})$ and $C = \tilde{O}(n^{1/c})$.
2. Bob finds a random subset P' of P of size B .
3. For $i = 1$ to k,
   (a) Alice and Bob agree upon C random $g_{i,j} \in \mathcal{G}$.
   (b) Bob creates a ROM L with entries $L(v)$ containing the points p for which $g(p) = v$.
   (c) A secure circuit with ROM L performs the following computation on input $(q, \{g_{i,j}\})$,
      - Compute $v_{i,j} = g_{i,j}(q)$ for each j.
      - Lookup the $L(v_{i,j})$ one by one for the different $v_{i,j}$ until the communication exceeds dB. If it is less, make dummy queries so that it is exactly dB.
      - Output shares S_i^1, S_i^2 so that $S_i^1 \oplus S_i^2$ is the (possibly truncated) set of sets $L(v_j)$.
4. A secure circuit with inputs P', S_i^1, S_i^2,
   - Compute the set $S_i = S_i^1 \oplus S_i^2 = \cup_j L(v_j)$ for all i.
   - Let $f(q, P)$ be random in $P_{cr} \cap P'$ if it is non-empty.
   - Else let $f(q, P)$ be random in $P_r \cap \cup_i S_i$ if it is non-empty, else set $f(q, P) = \emptyset$.
   - Output $(f(q, P), \mathsf{null})$.

The communication is $\tilde{O}(dB)$. By using homomorphic encryption, one can reduce the dependence on d, as per remark 10. Let $\mathcal{E}$ be the event that $P_r \subseteq \cup_i S_i$, and let $\mathcal{F}$ be the event that $P_{cr} \cap P'$ is non-empty.

Correctness: Suppose $P_r \neq \emptyset$. The probability s of correctness is just the probability we don't output $\emptyset$. Thus $s \geq \Pr[\mathcal{F}] + \Pr[\neg\mathcal{F}] \Pr[f(q, P) \neq \emptyset \mid \neg\mathcal{F}]$.

Case $|P_{cr}| \geq n^{1/2-1/(2c)}$: For sufficiently large B, we have $s \geq \Pr[\mathcal{F}] = 1 - 2^{-\Omega(k)}$.

Case $|P_{cr}| < n^{1/2-1/(2c)}$: It is enough to show $\Pr[f(q, P) \neq \emptyset \mid \neg\mathcal{F}] = 1 - 2^{-\Omega(k)}$. Fix i. Put $Y = \sum_j |L(v_{i,j})|$, where $|L(v_{i,j})|$ denotes the number of points in $L(v_{i,j})$. The expected number of points in $P \setminus P_{cr}$ that are in $L(v_{i,j})$ is at most $n \cdot O(1/n) = O(1)$. Since $|P_{cr}| < n^{1/2-1/(2c)}$, $\mathbf{E}[L(v_{i,j})] < n^{1/2-1/(2c)} + O(1)$. Thus $\mathbf{E}[Y] \leq B/3$ for large enough B, so $\Pr[Y > B] \leq 1/3$ by Markov's inequality. Thus, with probability $1 - 2^{-\Omega(k)}$, for at least half of the i, S_i is not truncated in step 3c. Moreover, for large enough B, any i, and any $p \in P_r$, $\Pr[p \in S_i] = 1 - 2^{-\Omega(k)}$ for large enough C. By a few union bounds then, $\Pr[P_r \subseteq \cup_i S_i] = \Pr[\mathcal{E}] = 1 - 2^{-\Omega(k)}$. Thus,

$$\Pr[f(q, P) \neq \emptyset \mid \neg\mathcal{F}] \geq \Pr[f(q, P) \neq \emptyset,\ \mathcal{E} \mid \neg\mathcal{F}] = \Pr[f(q, P) \neq \emptyset \mid \mathcal{E},\ \neg\mathcal{F}] \Pr[\mathcal{E}] \geq 1 - 2^{-\Omega(k)}.$$

Privacy: Note that Bob gets no output, so Alice's privacy follows from that of the secure circuit protocol. We construct a simulator $Sim(1^n, P_{cr}, q)$ so that the distributions $\langle q, P, f(q, P)\rangle$ and $\langle q, P, Sim(1^n, P_{cr}, q)\rangle$ are statistically close. Bob's privacy then follows by the composition with the secure circuit protocol.

Sim $(1^n, P_{cr}, q)$:

1. Set $B = \tilde{O}(n^{1/2+1/(2c)})$.
2. With probabiity $1 - \binom{n-|P_{cr}|}{B}\binom{n}{B}^{-1}$, output a random element of P_{cr}.
3. Else output a random element of P_r.

Let X denote the output of $Sim(1^n, P_{cr}, q)$. It suffices to show that for each $p \in P$, $|\Pr[f(q,P) = p] - \Pr[X = p]| = 2^{-\Omega(k)}$, since this also implies $|\Pr[f(q,P) = \emptyset] - \Pr[X = \emptyset]| = 2^{-\Omega(k)}$. We have

$$\begin{aligned} \Pr\left[f(q,P) = p\right] &= \Pr\left[f(q,P) = p, \mathcal{F}\right] + \Pr\left[f(q,P) = p, \neg\mathcal{F}\right] \\ &= \Pr\left[\mathcal{F}\right]|P_{cr}|^{-1} + \Pr\left[f(q,P) = p, \neg\mathcal{F}\right] \end{aligned}$$

Note that $\Pr[\mathcal{F}] = 1 - \binom{n-|P_{cr}|}{B}\binom{n}{B}^{-1}$. Therefore,

$$|\Pr[f(q,P) = p] - \Pr[X = p]| = \Pr[\neg\mathcal{F}]|\Pr\left[f(q,P) = p \mid \neg\mathcal{F}\right] - \delta(p \in P_r)|P_r|^{-1}|.$$

If $|P_{cr}| \geq n^{1/2-1/(2c)}$, this is $2^{-\Omega(k)}$, since then $\Pr[\neg\mathcal{F}] = 2^{-\Omega(k)}$. Otherwise, $|P_{cr}| < n^{1/2-1/(2c)}$, and as shown in the proof of correctness, we have $\Pr[\mathcal{E}] = \Pr[P_r \subseteq \cup_i S_i] = 1 - 2^{-\Omega(k)}$. Thus

$$\Pr[f(q,P) = p \mid \neg\mathcal{F}] = \Pr[f(q,P) = p \mid \mathcal{E},\ \neg\mathcal{F}]\Pr[\mathcal{E}] \pm 2^{-\Omega(k)} = \delta(p \in P_r)|P_r|^{-1} \pm 2^{-\Omega(k)},$$

which completes the proof.