Limits of Practical Sublinear Secure Computation

Elette Boyle [⋆] Yuval Ishai [⋆⋆] Antigoni Polychroniadou [⋆⋆⋆]

Abstract. Secure computations on big data call for protocols that have sublinear communication complexity in the input length. While fully homomorphic encryption (FHE) provides a general solution to the problem, employing it on a large scale is currently quite far from being practical. This is also the case for secure computation tasks that reduce to weaker forms of FHE such as "somewhat homomorphic encryption" or single-server private information retrieval (PIR).

Quite unexpectedly, Aggarwal, Mishra, and Pinkas (Eurocrypt 2004), Brickell and Shmatikov (Asiacrypt 2005), and shelat and Venkitasubramaniam (Asiacrypt 2015) have shown that in several natural instances of secure computation on big data, there are practical sublinear communication protocols that only require sublinear local computation and minimize the use of expensive public-key operations. This raises the question of whether similar protocols exist for other natural problems.

In this paper we put forward a framework for separating "practical" sublinear protocols from "impractical" ones, and establish a methodology for identifying "provably hard" big-data problems that do not admit practical protocols. This is akin to the use of NP-completeness to separate hard algorithmic problems from easy ones. We show that while the previous protocols of Aggarwal et al., Brickell and Shmatikov, and shelat and Venkitasubramaniam are indeed classified as being "practical" in this framework, slight variations of the problems they solve and other natural computational problems on big data are hard.

Our negative results are established by showing that the problem at hand is "PIR-hard" in the sense that *any* secure protocol for the problem implies PIR on a large database. This imposes a barrier on the local computational cost of secure protocols for the problem. We also identify a new natural relaxation of PIR that we call *semi-PIR,* which is useful for establishing "intermediate hardness" of several practically motivated secure computation tasks. We show that semi-PIR implies *slightly* sublinear PIR via an adaptive black-box reduction and that ruling out a stronger black-box reduction would imply a major breakthrough in complexity theory. We also establish information-theoretic separations between semi-PIR and PIR, showing that some problems that we prove to be semi-PIR-hard are not PIR-hard.

Keywords: Secure Computation, Private Information Retrieval, Sublinear Communication, Locally Decodable Codes

[⋆] IDC Herzliya, email: eboyle@alum.mit.edu.

[⋆⋆] Technion, email: yuvali@cs.technion.ac.il.

[⋆⋆⋆] Cornell Tech and University of Rochester, email: antigoni@cornell.edu.

1 Introduction

Protocols for *secure multi-party computation* (MPC) enable mutually distrusting parties to jointly evaluate a function on their private inputs, without revealing any information beyond the prescribed function outputs [Yao82, GMW87, BGW88, CCD88].

An important efficiency metric of MPC protocols is the required communication between parties. A great deal of research focus has gone towards minimizing the asymptotic communication complexity of MPC, as well as improving the practical efficiency of MPC. Our work proposes a theoretical framework for capturing the intersection. This framework can be used to provide a crude distinction between tasks that admit "practical" sublinear-communication protocols and ones that do not, akin to the use of NP-completeness to separate hard algorithmic problems from easy ones.

Secure computation on big data calls for MPC protocols that have sublinear communication complexity in the input size. The line of work on sublinear-communication MPC started with works on private information retrieval (PIR) [CKGS98, KO97] and related primitives, and culminated in the constructions of fully homomorphic encryption (FHE) [Gen09] schemes. FHE gives a general solution to the problem in that it essentially closes the gap between secure and insecure communication complexity.

The main concrete bottleneck of current FHE schemes, which makes them slow in practice, is their computational complexity. Even in the case of PIR, which can be viewed as the simplest instance of "somewhat-homomorphic encryption," local computation on the server side is by far the most significant cost. Indeed, PIR protocols without preprocessing provably require linear computational complexity, and all known PIR protocols on a database of length N require at least N "public-key operations" (comparable to the amortized cost of encrypting a bit in an underlying public-key encryption scheme). Consequently, the computational cost of such protocols is significantly higher than that of protocols that process a similar amount of information using only symmetric encryption. Moreover, unlike the case of OT-based protocols, little can be done for amortizing the cost of PIR or for pushing it to an input-independent preprocessing phase. Despite recent advances on the concrete cost of PIR [MBFK16] and the asymptotic cost of PIR with preprocessing [BIPW17, CHR17], performing a single instance of PIR on an N-bit database is more expensive in terms of local computation than, say, securely evaluating a boolean circuit of size N by relying on efficient OT extension techniques.

Notable exceptions to the above state of affairs are the work of Aggarwal et al. [AMP10] on medians, the work of Brickell and Shmatikov [BS05] on certain graph problems, and the work of shelat and Venkitasubramaniam [SV15] that vastly generalizes them. These works show that for certain natural and practically motivated problems, including several central combinatorial optimization problems, one can enjoy the best of both worlds: sublinear communication complexity with low computational overhead, both asymptotically and concretely. These works leave several interesting open questions. In particular, it is not clear how robust the positive results are to natural variations of the functionality and whether they extend to other optimization problems.

1.1 Our Results

Towards addressing the above open questions in a systematic way, we propose a clean formal framework to capture the feature of the protocols of [AMP10, BS05, SV15] that distinguishes them from more generic alternatives based on FHE or PIR. Concretely, we consider a model of secure two-party computation with input-independent preprocessing in the form of correlated randomness. (The latter can be used to implement oblivious transfer unconditionally.) We distinguish between:

- "Easy" problems, namely ones admitting sublinear-communication secure protocols that may rely on input-independent correlated randomness and oblivious transfer, and
- "PIR-hard" problems, for which any sublinear-communication protocol implies a nontrivial PIR protocol on a large database.

Given the current state of the art, PIR-hard problems are unlikely to be shown "easy," and any protocol for such problems is likely to have poor concrete efficiency.

PIR-Hardness of Combinatorial Problems. We then revisit a class of combinatorial optimization problems for which "easy" protocols have been demonstrated. (In particular, all of the protocols from [AMP10, BS05, SV15] are easy in the above sense.) We show that, while the original formulation of the problem yields lightweight protocols, certain natural and useful variants of the same problems are in fact PIR-hard. We first demonstrate this for the case of *one-sided* variants—in which only one party learns the function output—for an assortment of combinatorial problems with different structures:

- **Median.** The *median* functionality accepts a list of numerical inputs from each party and outputs the median of the combined list.
- **Convex Hull.** The 2-dimensional *convex hull* functionality accepts a set of points in 2-space from each party and outputs the subset of those points on the convex hull of the combined set.
- **Single-Source Shortest Distance.** The *SSSD* functionality accepts a set of weighted edges from each party (on n fixed vertices) with distinguished vertex v^* and outputs the lengths of $n-1$ shortest paths from v^* to each $v \neq v^*$ in the combined graph (taking parallel edges).
- **Approximate Set Cover.** The *approx set cover* functionality refers to the output of the polynomial-time greedy algorithm for polynomial time approximation of set cover (which iteratively selects the set that contains the largest number of uncovered elements).

We prove that the one-sided variant of each of the above problems is PIR-hard. This further implies that any *secret-shared* variant of the problems, in which the parties compute secret shares of the corresponding functionality output, is additionally PIR-hard. (Indeed, existence of an "easy" protocol for the latter directly yields an analogous protocol for the former, by having party 2 send his secret share to party 1). This may indicate that lightweight protocols for these problems cannot be effectively used "within" other larger MPC computations.

We remark that the previous "easy" protocol constructions for the above combinatorial problems frequently provide security within a promise setting, where certain restrictions are assumed to hold on parties' inputs (e.g., that parties' inputs are disjoint). Our negative results are each within the respective promise settings.

Our PIR-hardness results can be interpreted as imposing a barrier on the local computational cost of secure protocols for the problem. This barrier applies both asymptotically (linear computation is necessary without preprocessing) and concretely (achieving sublinear communication comes at a high computational cost given the current state of the art on PIR).

Semi-PIR-Hardness. We then identify a new natural relaxation of PIR that we call *semi-PIR,* which is useful for establishing "intermediate hardness" of several practically motivated secure computation tasks. Semi-PIR is defined analogously to PIR, except that the privacy requirement is relaxed to guarantee privacy of the client's query index $i \in [n]$ *only* if it holds that the corresponding database value z_i is equal to 1. This notion may be independently motivated by settings where there is natural asymmetry in privacy concerns for 0 versus 1 values (e.g., database of patients with and without some disease).

We show that semi-PIR with polylogarithmic communication complexity implies *slightly* sublinear PIR via an adaptive black-box reduction. Thus, semi-PIR-hardness can have a similar interpretation as PIR-hardness from a crude asymptotic perspective. Our reduction from PIR to semi-PIR makes use of query-efficient locally decodable codes (LDC). Correspondingly, ruling out a stronger black-box reduction would imply a major breakthrough in complexity theory, concerning existence of LDCs with polynomial rate and low query complexity.

Theorem 1 (Informal). *Suppose there is an efficient q-query LDC $C : \{0,1\}^n \to \{0,1\}^N$. Then, there exists a protocol that implements PIR on a database $z \in \{0,1\}^n$ by using an expected $O(2^q)$ (adaptive) calls to semi-PIR on a database $z' \in \{0,1\}^N$ and no additional interaction.*

The reduction effectively attempts to reconstruct the desired database value z_i, $i \in [n]$ by accessing positions $j_1, \ldots, j_q$ in the encoded database $j_\ell \in [N]$, each time to either the direct bit value or a negated version (so that the read value will be 0 with probability 1/2). At any point in which the queried location stores a 0, this query index is no longer hidden, and the reduction will restart with a freshly sampled set of q-queries. The smoothness of the LDC guarantees that revealing any *single* query index reveals nothing about the ultimate desired index i. Note the inherent adaptivity of this approach.

We also establish information-theoretic separations between semi-PIR and PIR. These imply that some problems that we prove to be semi-PIR-hard are provably *not* PIR-hard in a strong sense, suggesting that semi-PIR captures the true complexity of some natural secure computation tasks rather than being an artifact of our proof techniques.

Our semi-PIR-hardness results apply to natural two-sided functions, whose output is revealed to both parties. A broad class of such examples is "optimal selection from a short-list," where a Receiver has a small list of candidate indices, and both parties learn the identity of the candidate with the maximum/minimum/most desired value. Situations of secure computation of such problems can be motivated by real-life scenarios in which the identity of the winning candidate (selected job applicant, purchased item, travel destination) is public information that cannot be hidden, yet one is interested in hiding the runner-ups or the choice criteria.

One such concrete problem is the Two-Sided Nearest Neighbor problem. Here a server holds a large database of points (x_i, y_i) in the Euclidean plane, say a list of restaurant locations, and a client holds a point (x, y), say representing its own location. The output of both parties is the point (x_i, y_i) which is closest to (x, y). As discussed above, the reason we consider here a two-sided output is that the selected restaurant can be publicly observed. And while this output may reveal a lot of partial information about the client's input, it is easy to imagine situations in which the client may wish to hide the exact location (x, y) from which the search has been conducted.

Two-sided versus one-sided functionalities. Unlike secure protocols realizing two-sided functionalities, secure protocols for one-sided functionalities must reveal no information about the output to one of the parties. This rules out iterative approaches in which partial information about the output is gradually revealed to both parties, allowing them to minimize the local computation by accessing only relevant portions of the input. However, as we show in this work, some natural two-sided functionalities exhibit an intermediate form of hardness captured by Semi-PIR. In such cases, both parties get the output, but one of the parties receives additional information only if some condition on the output is met.

Hardness	**Combinatorial Problems**
Easy	Two-Sided Locally Compressible Minimum Spanning Tree Two-Sided Locally Compressible High-Order Median Predicates Protocols from [AMP10, BS05, SV15]
Semi-PIR Hard	Two-Sided Single Source Single Destination Shortest Path Two-Sided Nearest Neighbor Two-Sided Closest Destination Problem Two-Sided Short-List Selection
PIR Hard	One-Sided Median One-Sided Approximate Set Cover One-Sided Convex Hull One-Sided Single Source Shortest Distances Two-sided Median Predicate

Table 1. Sample of our hardness results for combinatorial problems

Local Compressibility. On the positive side, we identify a generic *local compressibility* property of combinatorial problems that directly permits efficient secure protocols for the problem, as well as any sufficiently "close" variant.

Loosely speaking, we say that a functionality $F : \{0,1\}^N \times \{0,1\}^N \to \{0,1\}^m \times \{0,1\}^m$ is locally compressible if there exists a preprocessing function $\mathsf{Pre} : \{0,1\}^N \to \{0,1\}^n$ for some $n \ll N$, for which it holds that $F(X,Y) = F(\mathsf{Pre}(X), \mathsf{Pre}(Y))$. In such a case, an "easy" sublinear protocol for securely computing F can be achieved by first performing the local preprocessing, and then executing an arbitrary MPC for the circuit/program on the compressed inputs. This generality allows us to extend beyond the core functionality F itself, to provide an "easy" sublinear protocol for any composed function $G \circ F$ for which the circuit size of G is not too complex. This includes, for example, one-sided variants.

We demonstrate this local compressibility property in two example settings:

- **Minimum Spanning Tree.** The *MST* functionality accepts a set of weighted edges from each party (on n fixed vertices) and outputs the minimum spanning tree of the combined graph.

  A lightweight protocol for MST was given by [SV15] for the promise setting where all edge weights are distinct, as the corresponding MST promise problem falls within their "greedy-compatible" protocol framework. We observe that, within a similar promise setting, the MST of the combined graph is preserved when parties compute the MST of their local graphs first and then submit the resulting tree as their input to the MST functionality (i.e., $\mathsf{Pre}(X) = MST(X)$). Our approach thus yields "easy" protocols with sublinear communication for MST and related variants.

- **"High-Order" Median Predicates.** For any predicate function P that depends only on the highest-order bits of its input, we show that the median predicate functionality $P \circ Med$ is locally compressible. More specifically, consider the median problem for n inputs, and suppose P depends only on the $\ell \in o(\log n)$ most-significant bits of its input. Then a party's list of n input values can be compressed to a succinct $2^\ell \in o(n)$-size *count vector* corresponding to the number of occurrences of each length-ℓ prefix within the list. Since the high-order prefix of the median is equal to the median of the corresponding high-order prefixes, this short count vector carries sufficient information to evaluate the desired functionality.

1.2 Organization of the Paper

Section 2 contains useful preliminaries. In section 3 we present our formal notion of PIR-hardness, and PIR-hardness results for various combinatorial problem variants. Section 4 contains the definition and results pertaining to the notion of semi-PIR. Section 5 contains our positive local-compressibility results.

2 Preliminaries

Notation. We denote the security parameter by κ. We say that a function $\mu : \mathbb{N} \to \mathbb{N}$ is *negligible* if for every positive polynomial $p(\cdot)$ and all sufficiently large κ's it holds that $\mu(\kappa) < \frac{1}{p(\kappa)}$. We often use $[n]$ to denote the set $\{1, \ldots, n\}$. Moreover, we use $d \leftarrow \mathcal{D}$ to denote the process of sampling d from the distribution $\mathcal{D}$ or, if $\mathcal{D}$ is a set, a uniform choice from it. If $\mathcal{D}_1$ and $\mathcal{D}_2$ are two distributions, then we denote that they are statistically close by $\mathcal{D}_1 \approx_{\mathrm{s}} \mathcal{D}_2$; we denote that they are computationally indistinguishable by $\mathcal{D}_1 \approx_{\mathrm{c}} \mathcal{D}_2$; and we denote that they are identical by $\mathcal{D}_1 \equiv \mathcal{D}_2$.

Two-Party Computation. We assume familiarity with standard cryptographic primitives. For notational purposes, we recall here the basic working definitions. We refer to e.g. [Can01] for the formal definitions. A two-party protocol is cast by specifying a random process that maps pairs of inputs to pairs of outputs (one for each party). We refer to such a process as a functionality and denote it by $F : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^* \times \{0,1\}^*$ where $F = (F_1, F_2)$. That is, for every pair of inputs (x, y), the output-pair is a random variable $(F_1(x,y), F_2(x,y))$ ranging over pairs of strings. The first party (with input x) wishes to obtain $F_1(x,y)$ and the second party (with input y) wishes to obtain $F_2(x,y)$. The aim of a secure two-party protocol is to protect an honest party against dishonest behavior by the other party. In this paper, we consider semi-honest static adversaries which strengthens our impossibility results.

The security of a protocol is analyzed by comparing what an adversary can do in the protocol to what it can do in an ideal scenario that is secure by definition. This is formalized by considering an ideal computation involving an incorruptible trusted third party to whom the parties send their inputs. The trusted party computes the functionality on the inputs and returns to each party its respective output. Loosely speaking, a protocol is secure if any adversary interacting in the real protocol (where no trusted third party exists) can do no more harm than if it was involved in the above-described ideal computation.

Protocols in the Preprocessing or Correlated Randomness Model. We will also consider protocols for the preprocessing model. In the preprocessing model, the specification of a protocol also includes a joint distribution $P_{R_1 \cdots R_n}$ over $\mathcal{R}_1 \times \ldots \times \mathcal{R}_n$, where the $\mathcal{R}_i$'s are finite randomness domains. This distribution is used for sampling correlated random inputs $(r_1, \ldots, r_n) \leftarrow P_{R_1 \cdots R_n}$ received by the parties before the execution of the protocol. Therefore, the preprocessing is independent of the inputs. The actions of a party P_i in a given round may in this case depend on the private random input r_i received by P_i from the distribution $P_{R_1 \cdots R_n}$ and on its input x_i and the messages received in previous rounds. In addition, the action might depend on the statistical security paramenter κ which is given as input to all parties along with x_i and r_i. Using the standard terminology of secure computation, the preprocessing model can be thought of as a hybrid model where the parties have one-time access to an ideal randomized functionality P (with no inputs) providing them with correlated, private random inputs r_i.

2.1 Private Information Retrieval

A (single-server) Private Information Retrieval (PIR) [CKGS98, KO97] protocol allows a client to retrieve a data item from a database held by a server while hiding which item it is after. More specifically, the database is modeled as an n-bit string z out of which the client retrieves the i-th bit z_i, while giving the server no information about the index i. The communication complexity of such a protocol is denoted by $c(n)$. A trivial PIR protocol would have the server sending the entire data string to the client (i.e. $c(n) = n$), thus satisfying the PIR privacy requirement in an information-theoretic way. We assume by default that any PIR protocol should be nontrivial in the sense that $c(n) < n$, and only consider computational security against semi-honest (passive) servers. We denote by $\mathsf{View}_S(z, i))$ the view of the PIR server in its interaction with the client on local inputs z, i and public input $n = |z|$, and by $\mathsf{Out}_C(z, i)$ the output of the client. Our definition treats the database size n as a public parameter that is also used as a security parameter.

Definition 1 (PIR). *Let (S, C) be an interactive protocol between a server S and a client C, where both S and C are PPT algorithms. We say that (S, C) is a* private information retrieval (PIR) *protocol if there exists a negligible function $\nu(n)$ such that:*

- **Correctness:** *For every $n \in \mathbb{N}$, $i \in [n]$, and $z = (z_1, \ldots, z_n) \in \{0,1\}^n$,*
$$\Pr\left[\mathsf{Out}_C(z, i) = z_i\right] \geq 1 - \nu(n).$$
- **Security:** *For every non-uniform polynomial time distinguisher D, $n \in \mathbb{N}$, $i, j \in [n]$, and $z = (z_1, \ldots, z_n) \in \{0,1\}^n$, it holds that $|p_i - p_j| \leq \nu(n)$, where*
$$p_i := \Pr\left[D(1^n, \mathsf{View}_S(z, i)) = 1\right],$$
$$p_j := \Pr\left[D(1^n, \mathsf{View}_S(z, j)) = 1\right].$$
- **Efficiency:** *The communication complexity $c(n)$ on a database $z \in \{0,1\}^n$ is always required to be at most $n - 1$. We say that PIR protocol is* slightly sublinear *if $c(n) = O(n / \log^\gamma n)$ for every positive integer γ, and that it is* polylogarithmic *if $c(n) = O(\log^\gamma n)$ for some positive integer γ.*

We note that polylogarithmic single-server PIR protocols exist under (subexponential versions of) standard cryptographic assumptions [CMS99, Lip05, BV14]. On the other hand, PIR provably requires linear server computation in the database size [BIM04], and all known protocols make an intensive use of public key cryptography. Even in the fastest existing implementations of PIR [MBFK16], maximizing the speed of server (which is still at least an order of magnitude slower than a symmetric encryption of the entire database) has a high cost in communication.

Additional evidence for the hardness of PIR comes from the impossibility of realizing PIR information-theoretically in the OT-hybrid model or even using general correlated randomness [IKM$^+$13]. This gives evidence against the possibility of using input-independent preprocessing or fast OT extension techniques [IKNP03] for amortizing the cost of PIR-based protocols, and should be contrasted with the fact that without the sublinear communication requirement, information-theoretic protocols exist in these models.

3 The PIR-Hardness Framework

We put forth a framework for separating "practical" sublinear computation protocols from "impractical" ones, by means of a notion of *PIR hardness.* PIR serves as an appealing benchmark metric for measuring protocol computation complexity in the sublinear communication regime: The functionality is natural and convenient to reduce to. And, since all known constructions make use of heavy public-key computations, this gives an indication that for any functionality which reduces to it, an analogous level of computation may be required. The high-level interpretation is thus that (given the current state of the art on PIR) saying that f is PIR-hard implies that evaluating f with a low communication complexity has a high computational cost. Even further, this computational cost cannot be amortized or moved to an input-independent preprocessing phase.

Definition 2 (PIR Hardness). *Let* $f : \{0,1\}^N \times \{0,1\}^N \to \{0,1\}^{m(N)} \times \{0,1\}^{m(N)}$ *be a two-party functionality.*

- *We say that* f *is* $(n(N), \tau(N))$-PIR-hard *if there is a single-server PIR protocol that makes* $\tau(N)$ *(expected) oracle calls to* f *on inputs of length* N*, where the PIR database size is* $n(N)$ *and, in addition to the oracle calls there is no additional communication.*
- *We say that* f *is* non-interactively $n(N)$-PIR-hard *if it is* $(n(N), 1)$*-PIR-hard, and that* f *is PIR-hard if it is non-interactively* $n(N)$*-PIR-hard for some* $n(N) = \tilde{\Omega}(N) = N/polylog(N)$*.*

Most (but not all) of the PIR hardness results obtained in this paper are of the simpler non-interactive type, namely the PIR protocol only applies a local mapping to the input of each party and then makes a single invocation of f with no additional interaction. The parameter $n(N)$ and $\tau(N)$ should be interpreted as a lower bound on the amount of expensive computation (which cannot be amortized or moved to a preprocessing phase) that is required for a sublinear-communication secure computation of f.

More concretely, we have the following easy corollary of PIR-hardness.

Claim. Suppose $f : \{0,1\}^N \times \{0,1\}^N \to \{0,1\}^{m(N)} \times \{0,1\}^{m(N)}$ is PIR-hard and has a protocol Π with $O(N^\beta)$ bits of communication for some $\beta < 1$. Then there is a nontrivial PIR protocol which on a database of size n makes a single invocation of Π on inputs of length $N = O(n)$ and uses no further interaction or assumptions.

The following remarks on our notion of PIR-hardness are in place:

Remark 1.

1. The above definition can be extended to allow extra sublinear communication beyond the f-oracle calls; however, our PIR-hardness results do not use this extension.
2. In the case of combinatorial problems involving graphs or other natural objects, the parameter N denotes the bit-length of a binary representation of the input for f. For example, in the case of a graph on ℓ nodes with polynomially bounded edge weights, we have $N = O(\ell^2 \log \ell)$. The polylogarithmic slackness in our default notion of PIR-hardness is meant to reduce the sensitivity of this notion to the way inputs are represented.

In the remainder of this section, we explore a general condition on functionalities which imply PIR hardness. We first consider functionalities f with *one-sided output*, i.e. where $f : \{0,1\}^N \times \{0,1\}^N \to \{\perp\} \times \{0,1\}^{m(N)}$ delivers output only to one of the two parties. We observe that in this setting, PIR hardness is tightly related to a combinatorial VC-dimension-style measure of complexity. We then extend this to demonstrate a sufficient condition for PIR-hardness of *two-sided predicate* functionalities.

3.1 VC-Dimension and Non-Interactive PIR-Hardness

In the case of one-sided output functionalities, where only one of the two parties receives output, the *privacy* property of PIR can be obtained immediately (namely, the server will play the role of the party who receives no output). PIR hardness of such a functionality then translates to a sufficient "combinatorial richness," capturing that the input-output behavior of the functionality is enough to encode the information of an entire database. We draw a connection between this property and a form of "efficient VC-dimension."

VC Dimension. We next define the Vapnik Chervonenkis (VC) dimension of a class of functions $\mathcal{F}$. The VC dimension gives a measure for the 'richness' of $\mathcal{F}$, which is useful in learning theory and computational complexity. We assume in the following that all functions in $\mathcal{F}$ are defined over the same input domain.

Definition 3 (VC-Dimension [VC71]). *Let $\mathcal{F}$ be a class of functions from some input domain D to $\{0,1\}$. We say that $\mathcal{F}$ shatters a point set $I \subset D$, if for every function $g : I \to \{0,1\}$, there is a function $f \in \mathcal{F}$ which agrees with g on I. The VC-dimension of $\mathcal{F}$, denoted by VC($\mathcal{F}$), is the size of the largest point set I, that is shattered by $\mathcal{F}$.*

The VC-dimension can be extended to a class $\mathcal{F}$ of non-boolean functions from D to E. In this case, the set I is shattered if there exists a universal boolean (single-bit output) decoder $\gamma : E \to \{0,1\}$ such that I is shattered in the above sense by $\mathcal{F}' = \{\gamma(f(\cdot)) : f \in \mathcal{F}\}$.

For the generalization of VC dimension to functions with multi-bit outputs, a number of notions have been considered in the literature (e.g., [NAT89, BIKO12]). In this work, we handle multi-bit outputs applying a universal boolean decoder on the output of non-boolean functions, as was previously suggested in [BIKO12]. The work of [BIKO12] uses the relation between PIR and VC dimension to construct PIR protocols. We further develop this relation and use it to establish PIR-harndess.

Essentially, the VC-dimension of a multi-bit output function class is the maximum VC-dimension of the boolean function class $\gamma \circ \mathcal{F}$ over the choice of the boolean "decoder" function γ.

We observe that for a one-sided functionality $f : \{0,1\}^N \times \{0,1\}^N \to \{\perp\} \times \{0,1\}^{m(N)}$, non-interactive PIR hardness of f coincides directly with the following notion of *efficiently computable* VC-dimension of the induced function class $\mathcal{F} = \{f(x,\cdot)\}_{x \in \{0,1\}^N}$. Explicitly, a non-interactive construction of PIR of database size n from f corresponds directly to efficient procedures for: identifying a shattering set $I \subseteq \{0,1\}^N$ for $\mathcal{F}$ (dictating how the client maps his query index i to an input y to f), finding the appropriate function $f(x,\cdot)$ to yield the desired output string on the n inputs in I (dictating how the server maps his n-size database to an input x to f), and determining and evaluating the universal decoder γ (for converting the output of f to an output of the PIR query). Privacy of the resulting PIR scheme follows immediately, since the functionality does not output anything to the first party (server). Correctness of the PIR holds because this gives a mapping from $x \in \{0,1\}^N$ to a function $f(x,\cdot)$ and $i \in [N]$ to an input y for which $f(x,y) = x_i$.

Below is a proof of equivalence for non-interactive reductions for the case of boolean functionalities.

Theorem 2. *Let $f : \{0,1\}^N \times \{0,1\}^N \to \{\perp\} \times \{0,1\}$ be a one-sided functionality with inputs $x, y \in \{0,1\}^\kappa$ and a bit output. Let $\mathcal{F}_\kappa = \{f_\kappa(x,\cdot)\}$ for $x \in \{0,1\}^\kappa$. Then the set $\mathcal{S} = \{\mathcal{F}_\kappa\}_{\kappa \in \mathbb{N}}$ has* efficiently computable *VC-dimension h, where $h(\kappa) = \kappa$, if and only if f_κ is $(\kappa, 1)$-PIR-hard.*

Proof. If $\mathrm{VC}(\mathcal{F}_\kappa) \geq h(\kappa)$ then for every κ both parties in the PIR protocol have access to shattered set I.[1] Then, the server given the database, which is an assignment of I, computes x for the function f_κ that satisfies the assignment. More specifically, if $\mathrm{VC}(\mathcal{F}_\kappa) \geq h(\kappa)$ then $\exists y = (y_1, \ldots, y_\kappa)$ such that for every assignment $(b_1, \ldots, b_\kappa)$ of y, $\exists\ x$ such that for every $i, f_\kappa(x, y_i) = b_i$. It is easy to see that both parties can run the PIR protocol on input x and y_i from the server and the client, respectively, such that only the client receives the i-th bit of the database b_i.

For the other direction, we need to show that given a PIR-hard function f_κ then $\mathrm{VC}(\mathcal{F}_\kappa) \geq h(\kappa)$. Based on the fact that the deterministic reduction of PIR on a κ-bit database is non-interactive and that the client has no information about the database the claim follows. In particular, since the client has no access to the database then for the i-th bit of the database the client will use the same y_i and the server will use the same x based on the database acting as the assignment. Therefore, $\mathrm{VC}(\mathcal{F}_\kappa) \geq h(\kappa)$ since PIR holds for all databases/assignments.

Two-Sided Predicates. The above additionally gives an approach for showing PIR-hardness of *two-sided predicate* functionalities, as we now describe.

[1] The shattered set must be exactly the same between the client and the server of the PIR protocol.

Theorem 3. *Suppose the one-sided functionality $f : \{0,1\}^N \times \{0,1\}^N \to \{\perp\} \times \{0,1\}$ is $(n(N), 1)$-PIR-hard. Then the corresponding two-sided functionality $f' : \{0,1\}^N \times \{0,1\}^N \to \{0,1\} \times \{0,1\}$ that delivers the same output predicate f to both parties is $(\lfloor n(N)/2 \rfloor, 1)$-PIR-hard.*

Proof. By definition of $(n(N), 1)$-PIR-hardness, there exists an efficient non-interactive construction of single-server PIR on a $n(N)$-size database using a single execution of f. This corresponds to three efficient algorithms: (1) a mapping $C : [n] \to \{0,1\}^N$ taking the client's index $i \in [n]$ to some input $x \in \{0,1\}^N$ to submit to f, (2) a mapping $S : \{0,1\}^n \to \{0,1\}^N$ taking the server's database $z \in \{0,1\}^n$ to some input $y \in \{0,1\}^N$ to submit to f, and (3) a reconstruction procedure R (which may depend on state from the execution of C) translating the output bit of f to the queried value z_i.

Note that by the correctness of the existing PIR scheme for any database z (in particular, for a randomly chosen z), it must be that the output bit of f on inputs $(C(i), S(z))$ provides a full bit of entropy of information about the value of z_i. That is, the output bit of f must *be* either the value z_i or its negation, and the choice of which cannot be dependent on x (as this is unknown to the client).

We provide a construction of PIR on a $\lfloor n(N)/2 \rfloor$-size database using a single execution of f', corresponding to (C', S', R'). For notational simplicity, assume $\lfloor n/2 \rfloor = n/2$. The transformation is as follows:

- C': The client encodes his input $i \in [n/2]$ as follows. First, sample a random bit $b \leftarrow \{0,1\}$. Then execute $C(i + b \cdot n/2)$.
- S': The server encodes his database $z \in \{0,1\}^{n/2}$ by executing $S(z||\bar{z})$; i.e., on the n-bit value formed by concatenating z with the bitwise negation of z.
- R': Given output w from the execution of f', output $b \oplus w$.

Correctness follows directly from the correctness of the underlying PIR (C, S, R). Security holds because the output bit w is distributed uniformly given the view of the server (i.e., given x).

A general version of Theorem 3 that applies to functionalities f with very short (sub-logarithmic length) outputs appears in the full version.

3.2 PIR-Hardness of Natural Combinatorial Problems

We demonstrate that in many cases even close variants of problems which admit practical sublinear protocols can be PIR-hard. In the following subsections, we consider variants of the Median, Convex Hull, Single-Source Shortest Path, and Approximate Set Cover problems.

Each of our reductions follows the approach and notation of the "efficient" VC-dimension connection described above, including the identification of shattered set I of the client's input space, and a universal decoder γ for converting the (possibly multi-bit) output of f to the output of the PIR. For each case, the corresponding mappings will indeed be efficiently computable, as required.

Revisiting the Median Protocol. For a subset $S \subset U$ of a totally ordered universe set U, the ρth-ranked element is the value $x \in S$ that is ranked ρ when the set S is sorted in increasing order. The median is the element with rank $\rho = \lceil |S|/2 \rceil$. Given two parties A and B with input sets $X_A, Y_B \subset U$, respectively, we consider the problem of privately computing the ρth-ranked element of $X_A \cup Y_B$. Aggarwal et al. [AMP10] described protocols for the *median* function with sublinear communication and computation overhead. Specifically, in the two-party case, let the size of U be polynomial in N (so that elements are described by polylog(N) bits), and let $|X_A|, |Y_B| = N$ be the total number of the input elements. Then, the protocol of Aggarwal et al. [AMP10] for securely evaluating the *median* entails a communication cost of $\tilde{\mathcal{O}}(\log N)$. We remark that the protocol of Aggarwal et al. [AMP10] finds the median on simplified input instances X_A and Y_B where $X_A \cap Y_B = \emptyset$ and $|X_A| = |Y_B|$.

The median two-party and multi-party protocols of [AMP10] are in the two-sided model, where both parties receive an output. Moreover, the security of their protocols relies on the fact that partial information is only leaked via the function output. We now show that secure protocols for the *one-sided* setting cannot enjoy such efficient sublinear-communication properties: namely, the one-sided median functionality is PIR-hard.

One-sided Median Functionality In this one-sided model, given two parties A and B, only the first party A receives the output of the function while party B should not learn any information about the input of party A.

Definition 4 (One-sided Med functionality). *Let $N \in \mathbb{N}$. We define the two-party functionality* $\textsc{Med} : \{0,1\}^{\tilde{\mathcal{O}}(N)} \times \{0,1\}^{\tilde{\mathcal{O}}(N)} \to \{\perp\} \times \{0,1\}^{\tilde{\mathcal{O}}(\log N)}$ *by* $(X, Y) \mapsto (\perp, median(X \cup Y))$ *which on input two sets* $X, Y \subset \mathbb{Z}_{poly(N)}$*, from the sender and the receiver, respectively, outputs* $\perp$ *to the sender and the median of* $X \cup Y$ *to the receiver.*

Theorem 4. *The one-sided functionality* $\textsc{Med} : \{0,1\}^{\tilde{\mathcal{O}}(N)} \times \{0,1\}^{\tilde{\mathcal{O}}(N)} \to \{\perp\} \times \{0,1\}^{\tilde{\mathcal{O}}(\log N)}$ *is PIR-hard.*

Proof. We define a universal encoder γ that on input a bit-string outputs its Least-Significant Bit (LSB). We are going to find the point set I of size N, that is shattered by $\mathcal{F}'_{\textsc{Med}} = \{\gamma(\textsc{Med}(X, \cdot))\}_{X \in \mathbb{Z}_{poly(N)}}$.

Let (max, min) denote the maximum and the minimum element of $\mathbb{Z}_{poly(N)}$, respectively. Moreover, for each $i \in [N]$ let MIN_i (respectively, MAX_i) denote the multiset of size $|MIN_i| = i$ (resp, $|MAX_i| = i$) where each entry is equal to min (resp., max), respectively. Define $I = \{Y_1, \ldots, Y_N\}$ such that $Y_i = \{MIN_{N-i} \cup MAX_i\}$ for all $i \in [N]$. We will show that $\mathcal{F}'_{\textsc{Med}}$ shatters I. In particular, for each $g : I \to \{0,1\}$ we will show that $\exists X$ such that for every $Y_i \in I$, $\gamma(\textsc{Med}(X, Y_i)) = g(Y_i)$.

Let $g : I \to \{0,1\}$. Define $X = \{x_1, \ldots, x_N\}$ such that $x_i = (i)_2 || 10 \cdots 0 || g(Y_i) \in \mathbb{Z}_{poly(N)}$ where $(i)_2$ denotes the bit representation of i. More specifically, x_i is defined by concatenating a unique $\log N$-length prefix to each bit of $g(Y_i)$ to ensure that the resulting elements are sorted, and appending the binary representation of $N+1$ (i.e., $\log N + 1$ bits) to ensure the existence of N distinct integers smaller than all the resulting values.[2]

It holds that $\forall i \in [N]$, $\gamma(\textsc{Med}(X, Y_i)) = g(Y_i)$ since $\textsc{Med}(X, Y_i) = x_i$ and $\gamma(\textsc{Med}(X, Y_i)) = LSB(x_i) = g(Y_i)$. That said, it follows that $VC(\mathcal{F}'_{\textsc{Med}}) \geq N$. Since all mappings are efficiently computable, it follows that $\textsc{Med}$ is PIR-hard.

[2] For the case where the set Y has to be distinct then $MIN_j = \{min, min + 1, \ldots, min_{j-1}\}$, $MAX_j = \{max, max + 1, \ldots, max_{j-1}\}$. Furthermore, in such a case $\forall I \in [N]$ compute $x_i = (i)_2 || min || 10 \cdots 0 || g(Y_i) \in \mathbb{Z}_{2^\ell}$

Revisiting the Convex Hull Protocol. In the convex hull algorithm, two parties securely compute the convex hull M of the union of their input sets of points G_A and G_B in an euclidian plane. Each element consists of two integers that represent the X and Y coordinates of the point. We are interested in cases where the convex hull has description size that is sublinear in the input size (as otherwise sublinear communication protocols are unachievable). We thus consider a promise problem variant of the functionality CH, defined as follows:

Definition 5 (One-sided CH functionality). *Let* $N \in \mathbb{N}$. *Define the two-party (promise problem) convex hull functionality* $\mathrm{CH} : \{0,1\}^{\tilde{O}(N)} \times \{0,1\}^{\tilde{O}(N)} \rightarrow \{\perp\} \times \{0,1\}^{o(\log N)}$ *by* $\mathrm{CH}(G_A, G_B) = (\perp, convexhull(G_A \cup G_B))$*, which on input two sets* G_A, G_B *of* N *points on the* 2*-dimensional euclidean plane, from party* A *and party* B*, respectively, outputs* $\perp$ *to party* A *and the convex hull of* $G_A \cup G_B$ *to party* B*.*

An efficient sublinear-communication protocol for the *two-sided* convex hull promise problem was given by [SV15] (as it fits into their "greedy compatible" framework), assuming slight additional promise restrictions on the inputs (namely, no two points have the same X or Y coordinate and no three-points are collinear). We prove that the one-sided convex hull problem is PIR-hard.

Theorem 5. *The one-sided functionality* $\mathrm{CH} : \{0,1\}^{\tilde{O}(N)} \times \{0,1\}^{\tilde{O}(N)} \rightarrow \{\perp\} \times \{0,1\}^{\tilde{O}(\log N)}$ *is PIR-hard.*

Proof. We define a universal encoder γ that on input a convex hull of four nodes identifies the longest edge and rotates it such that: (1) the longest edge is parallel to the X axes and (2) the shortest edge is above the longest edge. The encoder outputs 0 if the node of the shortest edge, which is closer to the longest edge, is the left one, otherwise output 1. We are going to find the point set I of size N, that is shattered by $\mathcal{F}'_{\mathrm{CH}} = \{\gamma(\mathrm{CH}(G, \cdot))\}_{G \subset S}$.

Let Cr be a circle with center the origin of the axes (with arbitrary radius) on the euclidian plane. Set $\psi = 2\pi/2N = \pi/N$. Let us define $I = \{Y_1, \ldots, Y_N\}$ for all $i \in [N]$. Consider the two points on the circle with angle $\phi_i = (2i) \cdot \psi$ and angle $\overline{\phi}_i = (2i+1) \cdot \psi$. Then, define by τ_1 and τ_2 the tangents of these two points, respectively. Tangents τ_1 and τ_2 intersect at point P_i. Consider the line e_i passing through the center of the circle and the point P_i. Denote the intersection points of the line e_i with the circle by Q_i, Q'_i such that point Q_i is closer to point P_i. Next, consider the tangent τ_3 of the point Q'_i and define by R_i, S_i the points created by the intersection of τ_1, τ_3 and τ_2, τ_3, respectively. The set Y_i includes points Q_i, R_i, S_i.

We will show that $\mathcal{F}'_{\mathrm{CH}}$ shatters I. In particular, for each $g : I \rightarrow \{0,1\}$ we will show that $\exists G$ such that for every $Y_i \in I$, $\gamma(\mathrm{CH}(G, Y_i)) = g(Y_i)$. Let $g : I \rightarrow \{0,1\}$ then for all $i \in \{0, \ldots, N-1\}$, assign each $g(Y_i)$ to the point T_i with angle $\phi_i = (2i + g(Y_i)) \cdot \psi$. Then, G consists for all points assigned to $g(Y_i)$.

It holds that $\forall i \in [N]$, $\gamma(\mathrm{CH}(G, Y_i)) = g(Y_i)$. In particular, the convex hull in each case contains the points (Q_i, R_i, S_i, T_i). By construction, the longest edge is drawn by nodes R_i, S_i and the shortest edge by nodes T_i, Q_i and point Q_i is closer to the longest edge. If $g(Y_i)$ is 0 then Q_i is closer to R_i and γ outputs 0. Thus, since each of the above mappings is efficiently computable, it follows that CH is PIR-hard.

□

Revisiting the Single-Source Shortest Distance Protocol. In the Single Source Shortest Distance (SSSD) protocol, two parties securely compute the shortest path distances from a source vertex s to all other vertices in a joint weighted graph. More specifically, let G_A and G_B be the two parties' respective weighted graphs. Assume that $G_A = (V_A, E_A, w_A)$ and $G_B = (V_B, E_B, w_B)$ are complete graphs on the same set of vertices. Let $w_A(e)$ and $w_B(e)$ represent the weight of edge e in G_A and G_B, respectively.[3] The goal is to output the list M which contains the shortest path distances from the source vertex s to all other vertices. If the input graphs (which may have quadratically many edges) are describable in $\tilde{\mathcal{O}}(N)$ bits, the output (which must have at most linearly many items) can be described by $\tilde{\mathcal{O}}(\sqrt{N})$ bits.

Definition 6 (One-sided SSSP functionality). *Define the two-party functionality* $\mathrm{SSSP} : \{0,1\}^{\tilde{\mathcal{O}}(N)} \times \{0,1\}^{\tilde{\mathcal{O}}(N)} \to \{\perp\} \times \{0,1\}^{\tilde{\mathcal{O}}(\sqrt{N})}$ *by* $\mathrm{SSSP}(G_A, G_B) = (\perp, \mathit{shortestpaths}(G_A, G_B))$ *which takes as input from A and B two complete, weighted graphs G_A, G_B respectively, on the same set of vertices. Then, it outputs $\perp$ to A and the list of shortest path distances from a source vertex s to all other vertices in the joint weighted directed graph to B.*

An efficient sublinear-communication protocol was given by [SV15] for the two-sided version of a related problem, of single-source all-destinations (SSAD), which outputs the list of shortest paths from s to each other node, as opposed to just the distance of these paths. (This follows from their "greedy compatible" framework, via Dijkstra's algorithm.)

We prove the one-sided SSSP problem is PIR-hard. As the information of one-sided SSSP can be directly inferred from the information of one-sided SSAD, this further implies PIR-hardness of the one-sided SSAD problem.

Theorem 6. *The one-sided functionality* $\mathrm{SSSP} : \{0,1\}^{\tilde{\mathcal{O}}(N)} \times \{0,1\}^{\tilde{\mathcal{O}}(N)} \to \{\perp\} \times \{0,1\}^{\tilde{\mathcal{O}}(\sqrt{N})}$ *is PIR-hard.*

Proof. We define a universal encoder γ that on input N integers and an index i outputs 0 if the ith integer is even, or 1 otherwise. We are going to define a set I of size $N(N-1)/2$, that is shattered by $\mathcal{F}'_{\mathrm{SSSP}} = \{\gamma(\mathrm{SSSP}(G, \cdot))\}_G$.

Let us define $I = \{Y_1, \ldots, Y_N\}$ for all $i \in [N]$. For each edge $i = (u, v)$ in the graph Y_i proceed as follows. The edge between the starting note s to u is set to the minimum weight i.e. $w_{Y_i}\big((s,u)\big) = 0$ and there is no weight assignment for (s, v). For every other edge $w \neq \{u, v\}$ connected to s, the weight on the edge (s, w) is assigned to N^2 i.e. $w_{Y_i}\big((s,w)\big) = N^2$.

We will show that $\mathcal{F}'_{\mathrm{SSSP}}$ shatters I. In particular, for each $g : I \to \{0,1\}$ we will show that $\exists G$ such that for every $Y_i \in I$, $\gamma(\mathrm{SSSP}(G, Y_i)) = g(Y_i)$. Let $g : I \to \{0,1\}$ then enumerate all the nodes from 1 up to N and for every edge $j \in \binom{N}{2}$ in the graph G assign each weight to $2N^2 + 2j + g(Y_i)$ (For the special case where the edge includes the starting point s there is no weight assignment).

It holds that $\forall i \in [N]$, $\gamma(\mathrm{SSSP}(G, Y_i)) = g(Y_i)$. By construction the distance from the starting point s to v for $i = (u, v)$ is equal to $w_{Y_i}\big(u, v\big)$ which is equal to $2N^2 + 2j + g(Y_i)$. If $g(Y_i) = 0$ then w_{Y_i} is even. □

[3] Note that we can also consider incomplete graphs and graphs that include disjoint edges by setting appropriate special values of $w(e)$ for the given edges e.

Revisiting the Approximate Set Cover Protocol. Given a collection S of sets over a universe U, a set cover $C \subseteq S$ is a subcollection of the sets whose union is U. The set cover problem allows two parties A and B to securely find a minimum-cardinality set cover given S_A and S_B. While this problem is NP hard to solve exactly, it yields a natural greedy approximation algorithm. Namely, in each iteration, the algorithm takes the set of those remaining which contains the largest number of uncovered elements.

In what follows, the "Approximate Set Cover" functionality will refer to the output generated by running this greedy algorithm. As with previous problems, we will restrict our attention to a promise version of the problem, where the description size of the output set cover is sublinear in the input description size (as otherwise sublinear-communication protocols will not be possible).

Definition 7 (One-sided Approximate Set Cover SC functionality). *Let $N \in \mathbb{N}$. Given a universe U, we define the two-party functionality* $\mathrm{SC} : \{0,1\}^{\tilde{O}(N)} \times \{0,1\}^{\tilde{O}(N)} \rightarrow \{\bot\} \times \{0,1\}^{o(N)}$ *by* $\mathrm{SC}(S_A, S_B) = (\bot, C)$ *which on input finite sets $S_A \subseteq U$ and $S_B \subseteq U$ from party A and party B, respectively, outputs the result $C \subseteq S_A \cup S_B$ of the greedy set cover algorithm to party B.*

An efficient sublinear-communication protocol for the *two-sided* greedy approximate set cover promise problem was given by [SV15], following their "greedy compatible" framework. We prove the corresponding one-sided problem is PIR-hard.

Theorem 7. *The one-sided functionality* $\mathrm{SC} : \{0,1\}^{\tilde{O}(N)} \times \{0,1\}^{\tilde{O}(N)} \rightarrow \{\bot\} \times \{0,1\}^{o(N)}$ *is PIR-hard.*

Proof. We define a universal encoder γ that on input two sets outputs the minimum element that resides in both sets. We are going to define a set I of size $\Theta(N)$, that is shattered by $\mathcal{F}'_{\mathrm{SC}} = \{\gamma(\mathrm{SC}(S, \cdot))\}_S$.

Let $|U| = \ell + 2$ where $\binom{\ell}{\ell/2} \geq N$. In particular, let $U = \{0, 1, u_1, \ldots, u_\ell\}$. Let $V = \{\{0,1\}^\ell\}^N$ be a vector with all bit-strings of length ℓ with hamming weight 1/2 in lexicographical order. Denote by $V_{i,j}$ the bit of the j-th position of the i-th element of V.Define $I = \{Y_1, \ldots, Y_N\}$ such that $Y_i = \{0,1\} \cup_{j \in \ell} \{u_j \ |V_{i,j} = 0\}$ for all $i \in [N]$.

We will show that $\mathcal{F}'_{\mathrm{SC}}$ shatters I. In particular, for each $g : I \rightarrow \{0,1\}$ we will show that $\exists S = \{S_1, \ldots, S_N\}$ such that for every $Y_i \in I$, $\gamma(\mathrm{SC}(S, Y_i)) = g(Y_i)$. Let $g : I \rightarrow \{0,1\}$ then for all $i \in [N]$, set $S_i = \{g(Y_i)\} \cup_{j \in \ell} \{u_j \ |V_{i,j} = 1\}$.

It holds that $\forall i \in [N]$, $\gamma(\mathrm{SC}(S, Y_i)) = g(Y_i)$. By construction the output collection consists of two sets, i.e., S_i and Y_i. If $g(Y_i) = 0$ then the common minimum element in both sets is 0. □

4 Intermediate Hardness via Semi-PIR

There are natural two-sided functionalities that are provably not PIR-hard, but which instead imply the following notion of *semi-PIR*. Intuitively, semi-PIR is a relaxed version of PIR where the server is allowed to learn the output z_i and can furthermore learn the client's actual selection i only if $z_i = 0$. Note that a semi-PIR protocol with only two messages is necessarily a PIR protocol, but it is easy to convert any 2-message PIR protocol into (an artificial) 3-message semi-PIR protocol which is not a PIR protocol by having the client send i to the server if and only if $z_i = 0$.

The semi-PIR primitive is formally defined by making the following small change in the security requirement of PIR from Definition 1: instead of requiring indistinguishability between any $i, j \in [n]$, the requirement is only made for i, j such that $z_i = z_j = 1$.

One can roughly think of a semi-PIR protocol as a low-communication (passively) secure protocol for the functionality $\frac{1}{2}\mathsf{PIR}$ that maps (z, i) to (y, z_i), where $y = i$ if $z_i = 0$ and $y = \perp$ otherwise. Indeed, any semi-PIR protocol as above can be converted into a protocol for this functionality by having the client send y to the server in the end of the protocol.

4.1 Does Semi-PIR Imply PIR?

In this section we study the relation between semi-PIR and PIR. We show that a strong form of semi-PIR implies a weak form of PIR. Interestingly, this result is shown via an inherently *adaptive* reduction, which also exhibits some unusual tradeoffs between communication and computation. We then show that the semi-PIR functionality does not satisfy the default notion of PIR-hardness from Definition 2. In other words, one cannot construct a PIR protocol via a single non-interactive call to $\frac{1}{2}\mathsf{PIR}$. While we leave open the possibility of constructing polylogarithmic PIR from polylogarithmic semi-PIR, we show that ruling out such a construction would imply a breakthrough in the achievable complexity of locally decodable codes.

Obtaining weak PIR from semi-PIR. We start by showing how to use a single invocation of semi-PIR to build a probabilistic PIR functionality that (on every selection i) leaks i to the server with probability $1/2$ (and lets the client know that leakage occurred), but otherwise reveals nothing to the server. We denote this probabilistic functionality by $\mathsf{Rand}\frac{1}{2}\mathsf{PIR}$.

Lemma 1. *There exists a protocol for* $\mathsf{Rand}\frac{1}{2}\mathsf{PIR}$ *that, on a database* $z \in \{0,1\}^n$, *uses a single invocation of* $\frac{1}{2}\mathsf{PIR}$ *on a database* $z' \in \{0,1\}^{2n}$ *and no additional interaction.*

Proof. The $\mathsf{Rand}\frac{1}{2}\mathsf{PIR}$ protocol proceeds as follows. The server maps z to $z' = (z, \bar{z})$. The client picks a random mask $r \in \{0,1\}$ and maps i to $i' = i + rn$. The parties then invoke the $\frac{1}{2}\mathsf{PIR}$ oracle on inputs (z', i'). The client's output in the $\mathsf{Rand}\frac{1}{2}\mathsf{PIR}$ protocol is $z'_{i'} \oplus r$, where $z'_{i'}$ is the output of the $\frac{1}{2}\mathsf{PIR}$. It is easy to check that the output is correct, and that the server learns nothing about i if $z'_{i'} = 0$, which happens with probability $1/2$ and is detectable by the client. □

Given Lemma 1, it suffices to reduce PIR to $\mathsf{Rand}\frac{1}{2}\mathsf{PIR}$. Our reduction relies on the following strong form of locally decodable codes (LDCs), which can be viewed as 1-round multi-server PIR protocols with uniform queries of logarithmic size and a single answer bit. Using a general transformation of LDC to multi-server PIR from [KT00], such codes are implied by standard LDCs by allowing a small decoding error probability. For simplicity, we define here only the perfect notion which is satisfied by the best known LDC constructions.

Definition 8 (Perfect LDC). *We say that an encoding function* $C : \{0,1\}^n \to \{0,1\}^N$ *is a* q-query perfect LDC, *if there exists a probabilistic decoder algorithm* $D(i)$ *which probes* q *bits of the encoding such that the following properties hold:*

- **Correctness:** *For every* $z \in \{0,1\}^n$ *and* $i \in [n]$, *we have* $\Pr[D^{C(z)}(i) = z_i] = 1$.

- **Uniform queries:** *Letting* $(i_1, \ldots, i_q) \in [N]^q$ *be the sequence of indices read by* $D(i)$*, it holds that for every* $j \in [q]$ *the index* i_j *is uniformly distributed over* $[N]$*.*

Our construction of PIR from $\frac{1}{2}\mathsf{PIR}$ encodes the PIR database using a perfect LDC, and applies a "cautious" decoding strategy by repeatedly (and adaptively) using $\mathsf{Rand}\frac{1}{2}\mathsf{PIR}$ to simulate the LDC decoder while ensuring that at most one query from each decoding attempt is leaked. This strategy yields the following theorem.

Theorem 8. *Let* $n(N)$ *and* $q(N)$ *be functions such that there is a* $q(N)$*-query perfect LDC* $C : \{0,1\}^{n(N)} \to \{0,1\}^N$ *in which both the encoder and the decoder can be implemented in time poly*(N)*. Then, there exists a protocol that, given a parameter* N*, implements in time poly*(N) *PIR on a database* $z \in \{0,1\}^{n(N)}$ *by using an expected* $O(q(N) \cdot 2^{q(N)})$ *(adaptive) calls to* $\frac{1}{2}\mathsf{PIR}$ *on a database* $z' \in \{0,1\}^N$ *and no additional interaction.*

Proof. Let $q = q(N)$. The PIR protocol will make at most $q \cdot 2^q$ expected calls to $\mathsf{Rand}\frac{1}{2}\mathsf{PIR}$, which using Lemma 1 can be implemented using $q \cdot 2^{q+1}$ expected calls to $\frac{1}{2}\mathsf{PIR}$. The protocol starts with the server encoding the PIR database $z \in \{0,1\}^n$ into a codeword $Z \in \{0,1\}^N$. The client and the server then repeatedly apply the following procedure until z_i is successfully recovered.

1. The client invokes the LDC decoder $D(i)$ to generate query indices $(i_1, \ldots, i_q)$.
2. For $j = 1, \ldots, q$ (sequentially), the client and the server invoke $\mathsf{Rand}\frac{1}{2}\mathsf{PIR}$ with client input i_j and server input Z. The protocol restarts at Step 1 if i_j leaks (which occurs with probability 1/2), otherwise it continues to the next j. If all indices Z_{i_j} have been successfully retrieved, the client invokes D to recover z_i.

Since the leakage events in different invocations of $\mathsf{Rand}\frac{1}{2}\mathsf{PIR}$ are independent, the expected number of attempts until decoding is fully successful is 2^q, and so the expected number of $\mathsf{Rand}\frac{1}{2}\mathsf{PIR}$ invocations is $q \cdot 2^q$. The (perfect) security of the protocol follows from the fact that in any invocation of D, at most a single index i_j is leaked. By the definition of perfect LDC, this index is uniformly distributed independently of i. □

Alternatively, one can implement a *worst-case* variant of the above reduction that runs σ copies in parallel, each with a constant failure probability. This results in a PIR to semi-PIR reduction that has $2^{-\Omega(\sigma)}$ error probability and makes $O(q(N) \cdot 2^{q(N)})$ rounds of calls to $\frac{1}{2}\mathsf{PIR}$ with a total number of $O(\sigma \cdot q(N) \cdot 2^{q(N)})$ of $\frac{1}{2}\mathsf{PIR}$ calls.

One can instantiate Theorem 8 by using known LDC constructions in several ways. In particular, using Reed-Muller LDCs with $q(N) = \Theta(\log N)$, one gets PIR with good communication complexity but super-polynomial computational complexity. To get slightly sublinear PIR with polynomial computational complexity, we rely on best constant-query LDC constructions from [Efr09].

Corollary 1 (polylogarithmic semi-PIR ⇒ slightly sublinear PIR). *The existence of a polylogarithmic semi-PIR protocol implies the existence of a slightly sublinear PIR protocol. Moreover, if the semi-PIR protocol has constant round complexity then so does the PIR protocol.*

Proof. The LDC construction from [Efr09] is in fact a perfect LDC according to our definition, with the following parameters. For any positive integer α, there is a constant $q = q(\alpha)$, such that there is a q-query perfect LDC with $N(n) = \exp(\exp(\log^{1/\alpha} n))$, or $n(N) = \exp((\log\log N)^{\alpha}))$. Note that $n(N)$ is bigger than any polylogarithmic function in N. A slightly sublinear PIR is obtained by chopping a database of size N into blocks of size $n(N)$ and running the protocol guaranteed by Theorem 8 on each block. □

We note that the existence of “dream LDC” with $q = O(1)$ queries and polynomial length $N(n)$ would imply a stronger reduction that constructs polylogarithmic PIR from polylogarithmic semi-PIR. Thus, ruling out such a reduction would imply ruling out such dream LDC, which would be considered a breakthrough in complexity theory.

Separating semi-PIR from PIR. On the other hand, we show that semi-PIR is *not* PIR hard. More broadly, we demonstrate limitations in the possibility of *non-adaptive* reductions from PIR to semi-PIR.

We begin by showing that with a single call to semi-PIR one cannot achieve secure PIR even with small non-trivial correctness.

Theorem 9. *There cannot exist any reduction from n-bit PIR to $\frac{1}{2}\mathsf{PIR}$ with correctness better than* 0.6 *which makes a single call to $\frac{1}{2}\mathsf{PIR}$.*

Proof. Suppose towards a contradiction that there exists a reduction from PIR to $\frac{1}{2}\mathsf{PIR}$ via a single call with correctness 0.6. This corresponds to a (randomized) encoding E_{DB} from $x \in \{0,1\}^n$ to $\hat{x} \in \{0,1\}^{\hat{n}}$ and E_{index} from $i \in [n]$ to $j \in [\hat{n}]$, where the client learns $\hat{x}_j$ and the server learns j iff $\hat{x}_j = 1$ via $\frac{1}{2}\mathsf{PIR}$. Since correctness is 0.6, there must exist $i \neq i' \in [n]$ for which the distributions $\{j \leftarrow E_{\mathsf{index}}(i)\}$ and $\{j' \leftarrow E_{\mathsf{index}}(i')\}$ are statistically far. By the privacy requirement, this means the resulting index j or j' cannot be revealed except with negligible probability. In turn, this implies $\hat{x}_j = 0$ except with negligible probability over $E_{\mathsf{DB}}, E_{\mathsf{index}}$. However, this implies that on a random database x the client has a negligible advantage in guessing x_i, yielding a contradiction. □

We next build atop this result to further rule out the possibility of a reduction making *two* non-adaptive calls.

Theorem 10. *There cannot exist any reduction from PIR to $\frac{1}{2}\mathsf{PIR}$ which makes two parallel calls to $\frac{1}{2}\mathsf{PIR}$.*

Proof. Consider any reduction achieving n-bit PIR, making 2 parallel calls to $\frac{1}{2}\mathsf{PIR}$. This corresponds to a (randomized) encoding E_{DB} from $x \in \{0,1\}^n$ to $\hat{x} \in \{0,1\}^{\hat{n}}$ and E_{index} from $i \in [n]$ to $(i_1, i_2) \in [\hat{n}]^2$. By correctness, for every $i \in [n]$ there exists $i' \in [n]$ for which the distributions $\{(i_1, i_2) \leftarrow E_{\mathsf{index}}(i)\}$ and $\{(i'_1, i'_2) \leftarrow E_{\mathsf{index}}(i')\}$ are statistically far. Because of this, for each index i, it must be that the read values $(\hat{x}_{i_1}, \hat{x}_{i_2})$ take value $(1,1)$ with negligible probability over $E_{\mathsf{DB}}, E_{\mathsf{index}}$. Correctness of the final scheme implies that the values of $(\hat{x}_{i_1}, \hat{x}_{i_2})$ must have a full bit of entropy over a random database x; in particular, the value $(0,0)$ can occur with probability at most $1/2$. Then either $\hat{x}_{i_1}$ or $\hat{x}_{i_2}$ must equal 1 with probability at least $1/4$, without loss of generality say $\hat{x}_{i_1}$.

Consider, then, the following reduction which makes a single call to $\frac{1}{2}\mathsf{PIR}$ and achieves correctness $1/2 + 1/8 - \mathsf{negl}(n)$.

1. The server samples $\hat{x} \leftarrow E_{\mathsf{DB}}(x)$ and submits $\hat{x}$ to $\frac{1}{2}\mathsf{PIR}$.
2. The client samples $(i_1, i_2) \leftarrow E_{\mathsf{index}}(i)$ and submits i_1 to $\frac{1}{2}\mathsf{PIR}$.
3. The $\frac{1}{2}\mathsf{PIR}$ execution outputs $\hat{x}_{i_1}$ to the client and i_1 or $\perp$ to the server (depending on $\hat{x}_{i_1}$).
4. If $\hat{x}_{i_1} = 1$ then the client executes the decoding procedure for the original reduction on input $(1,0)$. Otherwise, he outputs a random bit.

Privacy of this construction follows from privacy of the original reduction. In the case that $\hat{x}_{i_1} = 1$, then with overwhelming probability we know that $\hat{x}_{i_2} = 0$, and thus the client computes the correct output. This means correctness of the overall scheme will hold with probability at least $1/4 + 3/4 \cdot 1/2 - \mathsf{negl}(n)$, contradicting Theorem 9. □

Because of the degradation in parameters, extending this separation to additional parallel queries will seem to require new ideas (e.g., for three queries ruling out $(1, 1, 1)$ gives a smaller boost in correctness when reducing to the two query case, which is insufficient to directly derive a contradiction). However, as a final note, we return to the $\mathsf{Rand}\frac{1}{2}\mathsf{PIR}$ functionality (used as an intermediate step in the earlier construction of PIR from $\frac{1}{2}\mathsf{PIR}$), in which the input index is revealed with probability $1/2$. This setting yields a direct analysis, and we observe that even $O(\log n)$ parallel calls to $\mathsf{Rand}\frac{1}{2}\mathsf{PIR}$ cannot yield PIR.

Proposition 1. *There cannot exist any reduction from PIR to* $\mathsf{Rand}\frac{1}{2}\mathsf{PIR}$ *making* $c \in O(\log n)$ *parallel calls to* $\mathsf{Rand}\frac{1}{2}\mathsf{PIR}$ *with negligible correctness error.*

Proof. Consider any reduction achieving n-bit PIR, making $c \in O(\log n)$ parallel calls to $\mathsf{Rand}\frac{1}{2}\mathsf{PIR}$. This corresponds to a (randomized) encoding E_{DB} from $x \in \{0,1\}^n$ to $\hat{x} \in \{0,1\}^{\hat{n}}$ and E_{index} from $i \in [n]$ to $(i_1, \ldots, i_c) \in [\hat{n}]^c$. By correctness, there exists $i \neq i' \in [n]$ for which the distributions $\{(i_1, \ldots, i_c) \leftarrow E_{\mathsf{index}}(i)\}$ and $\{(i'_1, \ldots, i'_c) \leftarrow E_{\mathsf{index}}(i')\}$ are statistically far. However, with noticeable probability $2^{-c} \in n^{-O(1)}$, all executions of $\mathsf{Rand}\frac{1}{2}\mathsf{PIR}$ will reveal the queried index, thus revealing the entire vector query $(i_1, \ldots, i_c)$, violating privacy of the scheme. □

4.2 Examples of Semi-PIR-Hard Problems

In this section we provide several natural examples for (two-sided) semi-PIR hard functionalities. The results of the previous section imply that any polylogarithmic protocol for these functionalities would imply a slightly sublinear PIR.

Definition 9 (Two-sided Single Source Single Destination Shortest Path). *Let* $N \in \mathbb{N}$. *Define the two-party functionality* $\mathrm{SSSD}_{s,t} : \{0,1\}^{\widetilde{O}(N^2)} \times \{0,1\}^{\widetilde{O}(N^2)} \rightarrow \{0,1\}^{\log N} \times \{0,1\}^{\log N}$ *by* $\mathrm{SSSD}_{s,t}(G_A, G_B) = (\mathit{shortestpath}(G_A, G_B))$ *that expects as input from* A *and* B *two directed, complete, weighted graphs* G_A, G_B *respectively, on the same set of* N *vertices where each weight is in* $\mathbb{N}$. *The functionality outputs the shortest path from the source vertex* s *to the destination vertex* t *in the joint weighted directed graph to both* A *and* B.

Theorem 11. *Let* $N \in \mathbb{N}$. *The two-sided Single Source Single Destination shortest Path function* $\mathrm{SSSD}_{s,t} : \{0,1\}^{\widetilde{O}(N^2)} \times \{0,1\}^{\widetilde{O}(N^2)} \rightarrow \{0,1\}^{\log N} \times \{0,1\}^{\log N}$ *is semi-PIR hard.*

Proof. We construct a semi-PIR protocol Π_{SSSD}, by calling functionality SSSD. Let Z be the server's input set of size $\widetilde{O}(N^2)$ where each element is a bit, and let i be the client's index. Moreover, let (G_A, G_B) be the two weighted input graphs provided to the SSSD functionality by the server and the client, respectively. Protocol $\Pi_{\mathrm{SSSD}}(Z, i)$ proceeds as follows:

Input Phase:

1. We split the nodes of G_A into two sets and the weight of each edge within each set is equal to infinity. Essentially, we form a complete bipartite graph with two extra vertices s, t. The source vertex s is connected to the vertices on the left side and a target vertex t connected to the vertices on the right side of the bipartite graph. We also consider an edge connecting s and t. The Server encodes the database Z on $\mathcal{O}(N^2)$ edges of the bipartite graph in G_A. In particular, the server assigns to edge j the weight $2Z_j$. The weight of the edge connecting s and t is set to 1.
2. The client sets up his graph G_B such that for the edge of interest $i = (u, v)$ the weight is set to $w_B(i) = 2$ and $w_B\big((s, u)\big) = 0$ and $w_B\big((v, t)\big) = 0$. The weights of all other edges are set to infinity.

Evaluation and Output Phase:

Invoke the two-sided SSSD functionality $\Pi_{\text{SSSD}}(G_A, G_B)$ that outputs the shortest path. If the shortest path contains the edge connecting s and t then $Z_i = 1$, otherwise $Z_i = 0$.

If $Z_i = 1$ then the i'th edge weight is 2, and shortest path will consist of the single edge connecting s and t, hiding the identity of i. If $Z_i = 0$, the shortest path contains edge i revealing the index i to the Server.

Definition 10 (Two-sided Closest Destination Problem). *Let $N \in \mathbb{N}$. Define the two-party functionality* $\text{CDP} : \{0,1\}^{\widetilde{\mathcal{O}}(N)} \times \{0,1\}^{\widetilde{\mathcal{O}}(\log N)} \to \{0,1\}^{\log N} \times \{0,1\}^{\log N}$ *by* $\text{CDP}(G, (s, t_1, t_2)) = (\mathit{ClosestDest}(G, (s, t_1, t_2)))$ *that expects as input a (sparse) graph G_A with size $\widetilde{\mathcal{O}}(N)$ description from party A and a source vertex s along with two target vertices t_1, t_2 with description size $\widetilde{\mathcal{O}}(\log N)$ from party B. Then, it outputs the identity of the closest destination from s to the one-out-of-two target vertices $\mathit{dist}(s, t_b) \leq \mathit{dist}(s, t_{1-b})$ to both A and B while t_{1-b} remains hidden.*

Theorem 12. *The two-sided Closest Destination Problem function* $\text{CDP} : \{0,1\}^{\widetilde{\mathcal{O}}(N)} \times \{0,1\}^{\widetilde{\mathcal{O}}(\log N)} \to \{0,1\}^{\log N} \times \{0,1\}^{\log N}$ *is semi-PIR hard.*

Proof. We construct a semi-PIR protocol Π_{CDP}, calling functionality $f_{G_A,(s,t_1,t_2)_B}$. Let Z be the server's input set of size $\widetilde{\mathcal{O}}(N)$ where each element is a bit, and let i be the client's index. Moreover, let $G_A, (s, t_1, t_2)_B$ be the inputs to the CDP functionality by the server and the client, respectively. Π_{CDP} proceeds as follows:

Input Phase:

1. Without loss of generality the Server encodes the database Z on $\mathcal{O}(N)$ edges of a star graph G_A with $N + 2$ vertices where the node s is connected to the other $N + 1$ vertices. The server enumerates all these $N + 1$ vertices from 1 up to $N + 1$ and for $j \in [N]$ assigns the weight of the edge connecting s and j to $2Z_j$ and the edge connecting s to $N + 1$ to 1.
2. The client chooses vertices s, i and $N + 1$.

Evaluation and Output Phase:

Invoke the two-sided protocol Π_{CDP} that outputs a target destination. If the target is vertex i then $Z_i = 0$ and if the target is vertex $N + 1$ then $Z_i = 1$.

If $Z_i = 1$ then the output is independent of the index i and thus the identity of i is hidden.

Definition 11 (Two-sided Nearest Neighbor Problem). *Define the two-party functionality* $\text{NN} : \{0,1\}^{\widetilde{\mathcal{O}}(N)} \times \{0,1\}^{\widetilde{\mathcal{O}}(\log N)} \to \{0,1\}^{\mathcal{O}(\log N)} \times \{0,1\}^{\mathcal{O}(\log N)}$ *by* $\text{NN}(D, \mathsf{loc})$

that expects as input a list D (of size N) of locations on the 2-dimensional euclidean plane from party A and a single location loc on the same plane from party B. Then, it outputs to both parties the location (x, y) in D that is nearest to location loc^A.

Theorem 13. *The two-sided Nearest Neighbor function* $\mathrm{NN} : \{0,1\}^{\widetilde{\mathcal{O}}(N)} \times \{0,1\}^{\widetilde{\mathcal{O}}(\log N)} \rightarrow \{0,1\}^{\mathcal{O}(\log N)} \times \{0,1\}^{\mathcal{O}(\log N)}$ *is semi-PIR hard.*

Proof. We construct a semi-PIR protocol Π_{NN}, calling functionality NN. Let $Z \in \{0,1\}^N$ be the server's input database, and let i be the client's index. Moreover, let (D, loc) be the inputs to the NN functionality by the server and the client, respectively. Protocol Π_{NN} proceeds as follows:

Input Phase:

1. For $j \in [N]$, let $(a,b)_j$ be evenly spaced points on a circle with center c and radius r in the Euclidean plane. The Server generates his input D to NN with respect to these points in the following way. If $Z_j = 0$ then set the jth location $(x,y)_j = (a,b)_j$. If $Z_j = 1$ set the location $(x,y)_j$ arbitrary outside the circle. In addition, he includes the center point c.
2. The client outputs the location loc that intersects the line crossing from the centre c and location $(a,b)_i$ and the circle with center c and radius $r/2$.

Evaluation and Output Phase: Invoke the two-sided protocol Π_{NN} that outputs the nearest location to loc. If $Z_i = 0$ then the output is $(a,b)_i$. If $Z_i = 1$ then the output is the centre c which is independent of the index i. That said, in this case the identity of i is not leaked.

Definition 12 (Two-sided Short-List Selection). *Define the two-party functionality* $\mathrm{SLS} : \{0,1\}^{\widetilde{\mathcal{O}}(N)} \times \{0,1\}^{\widetilde{\mathcal{O}}(\log N)} \rightarrow \{0,1\}^{2\log N} \times \{0,1\}^{2\log N}$ *by* $\mathrm{SLS}(L, (\mathsf{idx}_0, \mathsf{idx}_1))$ *that expects as input a list L of size N and input domain $[N]$ from party A and two indices $(\mathsf{idx}_0, \mathsf{idx}_1)$ from party B. The output is idx_0 if $L_{\mathsf{idx}_0} < L_{\mathsf{idx}_1}$, idx_1 if $L_{\mathsf{idx}_0} > L_{\mathsf{idx}_1}$ or both $\mathsf{idx}_0, \mathsf{idx}_1$ if $L_{\mathsf{idx}_0} = L_{\mathsf{idx}_1}$.*

Theorem 14. *The two-sided Short-List Selection function* $\mathrm{SLS} : \{0,1\}^{\widetilde{\mathcal{O}}(N)} \times \{0,1\}^{\widetilde{\mathcal{O}}(\log N)} \rightarrow \{0,1\}^{2\log N} \times \{0,1\}^{2\log N}$ *is semi-PIR hard.*

Proof. We construct a semi-PIR protocol Π_{SLS}, calling functionality SLS. Let Z be the server's input set of size N where each element is a bit, and let i be the client's index. Moreover, let $(L, \mathsf{idx}_0, \mathsf{idx}_1)$ be the inputs to the SLS functionality. Protocol Π_{SLS} proceeds as follows:

Input Phase:

1. The Server generates the list L of size $N+1$ as follows. For $j \in [N]$, $L_j = Z_j$ and $L_{N+1} = 0$.
2. The client chooses indices i and $N+1$.

Evaluation and Output Phase:
Invoke Π_{SLS} that outputs the index of the smallest entry or both indices in case of ties. If $Z_i = 0$ then the indices i and $N+1$ are revealed. If $Z_i = 1$ only the $N+1$ index is revealed which is independent of i.

Next, we observe that this problem is *not* PIR-hard by demonstrating it is implied by $\frac{1}{2}\mathsf{PIR}$ (which is separated from PIR in the above results).

Theorem 15. *If there exists a Semi-PIR protocol for a database of size $\mathcal{O}(N)$ that runs in k-rounds, then for every constant $c > 0$ there exists a protocol for the two-sided Short-List Selection function* $\mathrm{SLS} : \{0, \dots, c\}^{\widetilde{\mathcal{O}}(N)} \times \{0,1\}^{\widetilde{\mathcal{O}}(\log N)} \to \{0,1\}^{2\log N} \times \{0,1\}^{2\log N}$ *that runs in $\mathcal{O}(c \cdot k)$ rounds.*

Proof. Let $\mathit{\Pi}$ be the Semi-PIR protocol. Let L be the input list of f_{SLS}. Modify L such that each element is in its unary representation with $c+1$ bits. In particular, a number $n < c$ in the database is represented in unary by n ones. The rest of the $c + 1 - n$ most significant bits are set to 0. Construct the semi-PIR database D by storing all $N(c+1)$ bits of L in such a way that the element with index (idx, ℓ) is the ℓ-th bit of the element of L with index idx. Let $(\mathsf{idx}_0, \mathsf{idx}_1)$ be the input indices of party B to SLS. Then party A and party B run at most c sequential rounds, each one consisting of two parallel calls to $\mathit{\Pi}$. In the ℓ-th round, where $\ell \in [1, c]$, party B makes the following two parallel queries for every $b \in \{0,1\}$.

$$\mathit{\Pi}\big(D, (\mathsf{idx}_b, \ell)\big) = \left\{ \begin{array}{ll} \Big(\bot, \big((\mathsf{idx}_b, \ell), D_{\mathsf{idx}_b,\ell}\big)\Big), & \text{if } D_{\mathsf{idx}_b,\ell} = 1 \\ \Big((\mathsf{idx}_b, \ell), \big((\mathsf{idx}_b, \ell), D_{\mathsf{idx}_b,\ell}\big)\Big), & \text{if } D_{\mathsf{idx}_b,\ell} = 0 \end{array} \right\}$$

If for some ℓ, b, $D_{\mathsf{idx}_b,\ell} = 0$ the protocol completes and there are no more adaptive calls. For the case where $D_{\mathsf{idx}_b,\ell} \neq D_{\mathsf{idx}_{1-b},\ell}$ and $D_{\mathsf{idx}_b,\ell} = 0$ then $L_{\mathsf{idx}_{1-b}} > L_{\mathsf{idx}_b}$ and both parties receive idx_b. If $D_{\mathsf{idx}_b,\ell} = D_{\mathsf{idx}_{1-b},\ell}$ then $L_{\mathsf{idx}_{1-b}} = L_{\mathsf{idx}_b}$ and both parties receive $(\mathsf{idx}_b, \mathsf{idx}_{1-b})$.

Combining Theorem 14 with Theorem 15, we obtain the following corollary:

Corollary 2 (Short-List Selection is not PIR-hard). *The two-sided Short-List Selection function* $\mathrm{SLS} : \{0,1\}^{\widetilde{\mathcal{O}}(N)} \times \{0,1\}^{\widetilde{\mathcal{O}}(\log N)} \to \{0,1\}^{2\log N} \times \{0,1\}^{2\log N}$ *is not PIR-hard.*

5 Low Communication Locally Compressible Problems

In this section, we show that it is actually possible to achieve semi-honest security for one-sided problems and beyond if the problem satisfies the following notion of input compressibility.

Definition 13 (Locally Compressible Inputs). *We say that a functionality F : $\{0,1\}^N \times \{0,1\}^N \to \{0,1\}^m \times \{0,1\}^m$ has* locally compressible inputs *if there exists a preprocessing function* $\mathsf{Pre} : \{0,1\}^N \to \{0,1\}^{N^\alpha}$ *with $\alpha < 1$ for which $F(X, Y) = F(\mathsf{Pre}(X), \mathsf{Pre}(Y))$.*

Local compressibility of the inputs can yield semi-honest secure non PIR-hard ("easy") protocols with reduced communication complexity by first executing the local preprocessing and then calling a generic two-party protocol on the preprocessed input data.

In the following section we show that two optimization problems that satisfy the above property admit low communication complexity and are *not* PIR-hard. The first problem is the minimum spanning tree and the second one is the median protocol for a certain predicate on the output specified in Section 5.2.

5.1 Revisiting the Minimum Spanning Tree Protocol

A Minimum Spanning Tree (MST) of an edge-weighted graph is a spanning tree whose weight is no larger than the weight of any other spanning tree. More formally, given a connected, undirected graph $G = (V, E)$, a *spanning tree* is an acyclic subset of edges $T \subseteq E$ that connects all the vertices together. Assuming that each edge e=(u,v) of G has a numeric weight or cost, $w(e)$, we define the cost of a spanning tree T to be the sum of edges in the spanning tree

$$w(T) = \sum_{(u,v)\in T} w(u, v).$$

MST is a spanning tree of minimum weight. Note that the MST may not in general be unique, but it is true that if all the edge weights are distinct, then the MST will be unique.

Definition 14 (MST functionality). *Let $N \in \mathbb{N}$. We define the two-party functionality $f_{\mathrm{MST}_N}(G_A, G_B) = (T, T)$ which on input two connected, unidirected graphs $G_A = (V_A, E_A, w_A)$ and $G_B = (V_B, E_B, w_B)$ of size N with distinct edges where $V_A = V_B$ and $w_A(e), w_B(e)$ represent the weight of edge e in G_A and G_B, outputs a subset of edges $T \subseteq E_A \cup E_B$ that connect all the vertices together with the minimum weight $w(T) = \sum_{(e)\in T} w(e)$.*

An efficient sublinear-communication protocol for two-sided MST was given in [SV15].

Two-Sided Locally Compressible MST. In the sequel, we show that the MST protocol has locally compressible inputs and admits "easy" low communication secure protocols. Beyond the results of [SV15], this approach enables such protocols for secure computation of *functions of* the MST (whereas the [SV15] protocol only supports MST itself).

Theorem 16. *Let $n \in \mathbb{N}$, and let $\{0,1\}^\ell$ be the input domain of edge weights. Then for any function $g : \{0,1\}^{2\ell\cdot n} \to \{0,1\}^{n'}$ with circuit size $o(N)$, there exists a secure two-party computation protocol Π_{MST} for the functionality $g \circ f_{\mathrm{MST}_{n^2}} : \{0,1\}^{\ell\cdot n^2} \times \{0,1\}^{\ell\cdot n^2} \to \{0,1\}^{n'}$ which achieves statistical security in the preprocessing model, with communication complexity $\widetilde{\mathcal{O}}(n) \in o(N)$ (where $N = \ell \cdot n^2$).*

Proof. We proceed by constructing an MST protocol Π_{MST}, as per Definition 14, calling the preprocessing function $\mathsf{Pre} : \{0,1\}^{n^2} \to \{0,1\}^n$ as per definition 13. Let (G_A, G_B) be the connected, unidirected graphs provided to the Π_{MST} protocol by party A and party B, respectively.
Protocol $\Pi_{\mathrm{MST}}(G_A, G_B)$:

Input Phase:
The preprocessing function Pre on input a graph G outputs its MST, denoted by MST(G). In this phase each party locally computes $\mathsf{Pre}(G_A)$ and $\mathsf{Pre}(G_B)$ to obtain MST(G_A) and MST(G_B), respectively.

Evaluation and Output Phase:
Given two graphs $G_1 = (V_1, E_1, w_1)$ and $G_2 = (V_2, E_2, w_2)$ we denote by G_1 & G_2 the graph $G = (V, E, w)$ with $V = V_1, E = E_1 \cup E_2$ and for each edge $e \in E$, $w(e) = min(w_1(e), w_2(e))$.

Let $\mathit{\Pi}$ denote a generic two-party protocol in the RAM computation model. Such a protocol is run in order to compute and output $\mathrm{MST}((\mathrm{MST}(G_A)\ \&\ \mathrm{MST}(G_B))$ to both parties.

In order to prove correctness of the above protocol Π_{MST}, we need to prove that the local compressibility does not alter the final output. More specifically, we need to show that $\forall e \in \mathrm{MST}(G_A\ \&\ G_B)$ it is implied that $e \in \mathrm{MST}((\mathrm{MST}(G_A)\ \&\ \mathrm{MST}(G_B))$.

Suppose for contradiction that there is an edge e in G_B that is in the $\mathrm{MST}(G_A\ \&\ G_B)$ but not in $\mathrm{MST}(G_B)$. Consider the cut C of vertices (created by drawing a line that intersects the middle of the edge e), that contains *only* the edge e of $\mathrm{MST}(G_A\ \&\ G_B)$ (it exists since by definition there no cycles in the MST). It must be the case that e is the lightest edge of $G_A\ \&\ G_B$ in this cut C, otherwise we can swap it out with a lighter edge and contradict the minimality of $\mathrm{MST}(G_A\ \&\ G_B)$. A swap is defined by adding in e, forming a cycle in the graph, therefore removing the other edge in this cut and cycle, which is by assumption strictly heavier.

However, all edge weights in $G_A\ \&\ G_B$ are smaller or equal to the weights in G_B, since we take the minimum weight at every edge. This means that e must also be the lightest edge of G_B in this cut. But this contradicts minimality of $\mathrm{MST}(G_B)$ since we could always swap some edge of $\mathrm{MST}(G_B)$ in this cut with e to get a strictly cheaper MST. Finally, since without loss of generality we can consider disjoint edges and connected graphs the edge e must also be included in the final tree $\mathrm{MST}((\mathrm{MST}(G_A)\ \&\ \mathrm{MST}(G_B))$. This concludes the proof.

Security of the protocol Π_{MST} follows immediately from the security of the $\mathit{\Pi}$ protocol. Furthermore, it is clear that the communication complexity of the $\Pi_{\mathrm{MST}}(G_A, G_B)$ in the RAM model is $\widetilde{\mathcal{O}}(n)$ since after the local compressibility the input size to the generic two-party protocol $\mathit{\Pi}$ is reduced to $\mathcal{O}(n)$, making use of a generic statistically secure ORAM-based protocol.

□

5.2 Revisiting the Two-Sided Median Predicates Protocol

In the sequel, we focus on the case of predicates on the output of the median protocol and in particular on the high-order bits of their input.

Theorem 17. *Let $N \in \mathbb{N}$ and let $\{0,1\}^\ell$ be the input domain. For any predicate $\mathcal{P} : \{0,1\}^\ell \to \{0,1\}$ which depends only on the $o(\log N)$ most significant bits of the input, there exists a secure two-party computation protocol $\Pi_{\mathrm{pMED}_{\mathrm{N}}^{\mathcal{P}\ell}}$ for the functionality $\mathrm{pMED}_N^{\mathcal{P}}$ which achieves statistical security in the preprocessing model, with communication complexity $o(N)$.*

Proof. We proceed by constructing the $\Pi_{\mathrm{pMED}_{\mathrm{N}}^{\mathcal{P}\ell}}$ protocol, calling the preprocessing function $\mathsf{Pre} : \{0,1\}^N \to \{0,1\}^{o(N)}$ as per definition 13. Let $X, Y \subset (\{0,1\}^\ell)^N$ be two input sets from party A and party B, respectively, sorted in increasing order such that $|X \cup Y| = 2N$. Protocol $\Pi_{\mathrm{pMED}_{\mathrm{N}}^{\mathcal{P}\ell}}(X, Y)$ proceeds as follows:

Input Phase:
The preprocessing function Pre on input a set S outputs a compressed output $\mathsf{Pre}(\mathrm{S})$ of $2^\ell \in o(N)$-size, denoted by ℓ'-size, *count vector* corresponding to the number of occurrences of each length-ℓ' prefix within the elements of the set. More specifically,

since there are $2^{\ell'}$ different representations for the ℓ' most significant bits, party A computes a counter vector $\mathbf{c}^A = (c_1^A, \dots c_{2^{\ell'}}^A)$ counting the appearance of each possible representation in the most significant bits of each element in the set X. Respectively, party B computes his counter vector $\mathbf{c}^B = (c_1^B, \dots c_{2^{\ell'}}^B)$.

Evaluation and Output Phase:

Let $\mathit{\Pi}$ denote a generic two-party protocol $\mathit{\Pi}(\mathbf{c}^A, \mathbf{c}^B)$ which on input the sets $\mathbf{c}^A, \mathbf{c}^B$, outputs to both parties the predicate result. For our purposes, protocol $\mathit{\Pi}$ is computing the median of the prefixes as encoded by the counter vectors $\mathbf{c}^A, \mathbf{c}^B$.

Correctness of the above protocol $\Pi_{\mathrm{pMED}_{\mathrm{N}}^{\mathcal{P}^\ell}}$ follows from the correctness of the $\mathit{\Pi}$, which does output the correct output predicate guaranteed by the structure of $\mathbf{c}^A, \mathbf{c}^B$. More specifically, since the high-order prefix of the median is equal to the median of the corresponding high-order prefixes, this short count vector carries sufficient information to evaluate the desired output predicate. Security follows immediately from the security of the protocol $\mathit{\Pi}$. The communication complexity of the $\Pi_{\mathrm{pMED}_{\mathrm{N}}^{\mathcal{P}^\ell}}(X, Y)$ protocol is $o(N)$ since after the local compressibility the input size to the generic two-party protocol $\mathit{\Pi}$ is set to $o(N)$. □

6 Concluding Remarks and Open Problems

Our work initiates an effort to design a rigorous complexity framework for identifying "hard" tasks, to which previous techniques for low-complexity sublinear MPC cannot possibly apply, making the first broad strokes of classifying natural problems as "hard" or "potentially easy." The framework we propose is not perfect, and indeed, problems that are "potentially easy" are not necessarily easy. This is also the case for the theory of NP-completeness, where some problems that are conjectured not to be NP-hard (such as integer factorization) are also conjectured to be not easy. However, again like NP-completeness, our framework does provide meaningful and useful separations between different flavors of natural problems that would otherwise look very similar. This can help understand and guide MPC solutions over big data.

There are many questions left to be studied. Whereas for one-sided functionalities, VC-dimension gives a good combinatorial characterization for PIR-hardness (restricted to deterministic, non-interactive reductions), the situation for two-sided functionalities is not as well understood unless the output is very short. Is there a natural analogue of VC-dimension that captures PIR-hardness and semi-PIR-hardness of two-sided functionalities? What about multi-party functionalities, or two-party functionalities that deliver different outputs to the two parties? What about extending our framework to the setting of security against malicious parties?

The relation between semi-PIR to PIR is also only partially understood. While we show that strong semi-PIR implies weak (but nontrivial) PIR, it is not clear that our reduction is the best possible. In particular, our reduction makes use of non-trivial machinery of locally decodable codes, it requires multiple rounds of calls to the semi-PIR oracle, and exhibits a tradeoff between communication and local computation. Are these nonstandard features inherent? For instance, can we rule out parallel reductions of this type, or prove that any reduction that makes few (sequential) calls to the semi-PIR oracle implies a locally decodable code with related parameters?

As discussed above, problems that escape our notions of hardness are not necessarily easy. It would be interesting to identify natural candidate problems of this kind and try to refine our hardness notions to capture them.

Finally, is there a useful hierarchy of hardness classes beyond PIR-hardness and Semi-PIR-hardness? For instance, one could try to capture different levels of "somewhat homomorphic encryption" that are more expensive to implement than PIR, say, corresponding to the circuit depth or algebraic degree.

Acknowledgements. The first author was supported in part by ISF grant 1861/16, AFOSR Award FA9550-17-1-0069, and ERC Grant no. 307952. The second author was supported in part by ERC grant 742754, ISF grant 1709/14, NSF-BSF grant 2015782, and a grant from the Ministry of Science and Technology, Israel and Department of Science and Technology, Government of India. The third author was supported by NSF grants 1617676, 1526377 and 1618884, IBM under Agreement 4915013672 and the Packard Foundation under Grant 2015-63124.

References

AMP10. Gagan Aggarwal, Nina Mishra, and Benny Pinkas. Secure computation of the median (and other elements of specified ranks). *J. Cryptology*, 23(3):373–401, 2010.

BGW88. Michael Or Ben, Shafi Goldwasser, and Avi Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation (extended abstract). pages 1–10, 1988.

BIKO12. Amos Beimel, Yuval Ishai, Eyal Kushilevitz, and Ilan Orlov. Share conversion and private information retrieval. In *Proceedings of the 27th Conference on Computational Complexity, CCC 2012, Porto, Portugal, June 26-29, 2012*, pages 258–268, 2012.

BIM04. Amos Beimel, Yuval Ishai, and Tal Malkin. Reducing the servers' computation in private information retrieval: PIR with preprocessing. *J. Cryptology*, 17(2):125–151, 2004.

BIPW17. Elette Boyle, Yuval Ishai, Rafael Pass, and Mary Wootters. Can we access a database both locally and privately? In *Theory of Cryptography - 15th International Conference, TCC 2017, Baltimore, MD, USA, November 12-15, 2017, Proceedings, Part II*, pages 662–693, 2017.

BS05. Justin Brickell and Vitaly Shmatikov. Privacy-preserving graph algorithms in the semi-honest model. In *Advances in Cryptology - ASIACRYPT 2005, 11th International Conference on the Theory and Application of Cryptology and Information Security, Chennai, India, December 4-8, 2005, Proceedings*, pages 236–252, 2005.

BV14. Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) \$\mathsf{LWE}\$. *SIAM J. Comput.*, 43(2):831–871, 2014.

Can01. Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *42nd Annual Symposium on Foundations of Computer Science, FOCS 2001, 14-17 October 2001, Las Vegas, Nevada, USA*, pages 136–145, 2001.

CCD88. David Chaum, Claude Crépeau, and Ivan Damgård. Multiparty unconditionally secure protocols (extended abstract). In *Proceedings of the 20th Annual ACM Symposium on Theory of Computing, May 2-4, 1988, Chicago, Illinois, USA*, pages 11–19, 1988.

CHR17. Ran Canetti, Justin Holmgren, and Silas Richelson. Towards doubly efficient private information retrieval. In *Theory of Cryptography - 15th International Conference, TCC 2017, Baltimore, MD, USA, November 12-15, 2017, Proceedings, Part II*, pages 694–726, 2017.

CKGS98. Benny Chor, Eyal Kushilevitz, Oded Goldreich, and Madhu Sudan. Private information retrieval. *J. ACM*, 45(6):965–981, 1998.

CMS99. Christian Cachin, Silvio Micali, and Markus Stadler. Computationally private information retrieval with polylogarithmic communication. In *Advances in Cryptology - EUROCRYPT '99, International Conference on the Theory and Application of Cryptographic Techniques, Prague, Czech Republic, May 2-6, 1999, Proceeding*, pages 402–414, 1999.

Efr09. Klim Efremenko. 3-query locally decodable codes of subexponential length. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 39–44, 2009.

Gen09. Craig Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 169–178, 2009.

GMW87. Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game or A completeness theorem for protocols with honest majority. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computing, 1987, New York, New York, USA*, pages 218–229, 1987.

IKM+13. Yuval Ishai, Eyal Kushilevitz, Sigurd Meldgaard, Claudio Orlandi, and Anat Paskin-Cherniavsky. On the power of correlated randomness in secure computation. In *Theory of Cryptography - 10th Theory of Cryptography Conference, TCC 2013, Tokyo, Japan, March 3-6, 2013. Proceedings*, pages 600–620, 2013.

IKNP03. Yuval Ishai, Joe Kilian, Kobbi Nissim, and Erez Petrank. Extending oblivious transfers efficiently. In *Advances in Cryptology - CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, Proceedings*, pages 145–161, 2003.

KO97. Eyal Kushilevitz and Rafail Ostrovsky. Replication is NOT needed: SINGLE database, computationally-private information retrieval. In *38th Annual Symposium on Foundations of Computer Science, FOCS '97, Miami Beach, Florida, USA, October 19-22, 1997*, pages 364–373, 1997.

KT00. Jonathan Katz and Luca Trevisan. On the efficiency of local decoding procedures for error-correcting codes. In *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing, May 21-23, 2000, Portland, OR, USA*, pages 80–86, 2000.

Lip05. Helger Lipmaa. An oblivious transfer protocol with log-squared communication. In *Information Security, 8th International Conference, ISC 2005, Singapore, September 20-23, 2005, Proceedings*, pages 314–328, 2005.

MBFK16. Carlos Aguilar Melchor, Joris Barrier, Laurent Fousse, and Marc-Olivier Killijian. XPIR : Private information retrieval for everyone. *PoPETs*, 2016(2):155–174, 2016.

NAT89. B. K. NATARAJAN. On learning sets and functions. *Machine Learning*, 4:67–97, 1989.

SV15. Abhi Shelat and Muthuramakrishnan Venkitasubramaniam. Secure computation from millionaire. In *Advances in Cryptology - ASIACRYPT 2015*

- *21st International Conference on the Theory and Application of Cryptology and Information Security, Auckland, New Zealand, November 29 - December 3, 2015, Proceedings, Part I*, pages 736–757, 2015.

VC71. V. N. Vapnik and A. Ya. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16(2):264–280, 1971.

Yao82. Andrew Chi-Chih Yao. Protocols for secure computations (extended abstract). In *FOCS*, pages 160–164, 1982.