Lower-Bounds on Public-Key Operations in PIR

Jesko Dujmovic[1] and Mohammad Hajiabadi[2]

[1]Helmholtz Center for Information Security (CISPA)
[2]University of Waterloo

Abstract

Private information retrieval (PIR) is a fundamental cryptographic primitive that allows a user to fetch a database entry without revealing to the server which database entry it learns. PIR becomes non-trivial if the server communication is less than the database size. We show that building (even) very weak forms of single-server PIR protocols, without pre-processing, requires the number of public-key operations to scale linearly in the database size. This holds irrespective of the number of symmetric-key operations performed by the parties.

We then use this bound to examine the related problem of communication efficient oblivious transfer (OT) extension.

Oblivious transfer is a crucial building block in secure multi-party computation (MPC). In most MPC protocols, OT invocations are the main bottleneck in terms of computation and communication. OT extension techniques allow one to minimize the number of public-key operations in MPC protocols. One drawback of all existing OT extension protocols is their communication overhead. In particular, the sender's communication is roughly double what is information-theoretically optimal.

We show that OT extension with close to optimal sender communication is impossible, illustrating that the communication overhead is inherent. Our techniques go much further; we can show many lower bounds on communication-efficient MPC. E.g., we prove that to build high-rate string OT from generic groups, the sender needs to do linearly many group operations.

1 Introduction

Secure Multi-Party Computation (MPC), allows two parties to jointly evaluate a function f while leaking nothing about their input to the other party beyond the output of f. A central goal of modern cryptography is to construct efficient MPC protocols. This goal is important not only from a theoretical but also from a practical viewpoint. The computational efficiency of all MPC protocols is typically bottlenecked by public-key operations (e.g., group operations, oblivious transfers (OT)). OT extension is a technique toward reducing public-key operations [Bea96, IKNP03], allowing one to get the results of many OTs at the cost of performing only a few OTs and some symmetric-key operations. This technique has revolutionized the practical development of MPC, leading to protocols which employ a small number of public-key operations for sophisticated tasks.

A significant limitation of existing OT extension techniques is their high communication cost: for performing ℓ 1-out-of-2 bit OTs, the sender communicates at least 2ℓ bits. This severely limits the use of OT extension in MPC settings where a low amount of communication is required 'by design' (e.g., Private-Information Retrieval (PIR)). The overarching goal of our paper is to understand communication-computation tradeoffs in such MPC settings. We show that in many such situations, performing many public-key operations is provably unavoidable. To put our results in context, let us illustrate how communication efficient OT would impact private information retrieval.

Private-Information Retrieval (PIR). Private information retrieval [CGKS95, KO97] is a fundamental cryptographic primitive that allows a user to fetch a database entry without revealing to the server which database entry it learns. PIR becomes non-trivial if the server-to-user communication is strictly less than the database size (and ideally growing sub-linearly or even polylogarithmically in the database size). In some applications, one may need extra properties, such as an overall (as opposed to server-to-user) sub-linear communication or server privacy. Throughout the paper, we require neither of these unless otherwise stated. Since we prove lower bounds, this makes our results stronger. A truly efficient PIR protocol has significant real-world applications such as private certificate retrieval or private DNS lookups. By now, we know how to build PIR with communication complexity polylogarithmic in n from a wide range of assumptions [CMS99, IP07, DGI$^+$19, CGH$^+$21, HHC$^+$22]. While the amount of communication is attractively low, the computation overhead leaves much to be desired.

Computational complexity of PIR. In (single-server) PIR protocols, the running time of the server cannot be sub-linear in n, the database size, without preprocessing [BIM00]. If it was sub-linear the server could not read all the entries, leaking information about the user's index i. Faced with this lower-bound, and the fact that PIR requires public-key assumptions [DMO00], one may wish to settle for the next best thing: making the number of public-key operations independent of n. Somehow curiously, in all existing PIR protocols based on Diffie-Hellman or OT related assumptions [DGI$^+$19, CGH$^+$21], not only the server's running time, but the number of public-key operations performed by the server grows at least linearly with n. There is no evidence, however, if this is inherent, and in fact, it has remained an open problem whether one can build a PIR protocol where the number of public-key operations is sub-linear in n.

Is it possible construct a single-server non-preprocessing PIR with a sub-linear amount of public-key operations and an arbitrarily large number of symmetric-key operations?[1]

OT Extension. A major tool used for minimizing computation is OT extension. Existing OT extension techniques induce at least a linear amount of communication for the sender, making them unsuitable for PIR applications. Specifically, under existing constructions, an extended OT sender needs to communicate at least as many bits as its total input length. Beaver's seminal construction [Bea96] works by encoding all the sender messages into a garbled circuit, which the receiver can evaluate only on the labels that correspond to her choice bits; the IKNP protocol [IKNP03] establishes correlated randomness between the sender and the receiver, allowing the sender to XOR his messages with the corresponding masks such that the receiver can only de-mask the correct messages. In both these protocols, the sender's outgoing protocol messages information-theoretically determine the entire sender's input, causing the communication overhead. This state of affairs raises the following natural question.

Is it possible construct OT extension where the sender communication is close to optimal?

In the above question, by 'optimal sender communication' we mean the best information-theoretically achievable communication: which is ℓ bits for the sender for performing ℓ 1-out-2 single-bit OTs. Since OT extension is crucially used in many MPC protocols, understanding its communication complexity is of both practical and theoretical value.

Having optimal sender communication for OT extension is reminiscent of rate-1 string OT: building 1-out-of-2 string OTs for a pair of ℓ-bit strings, where (roughly speaking) the sender communication grows as $\ell + \lambda$ (as opposed to $2\ell + \lambda$), where λ is the security parameter. Two-round rate-1 OT has found a number of applications, notably in the construction of PIR protocols with polylogarithmic communication [IP07, DGI$^+$19, GHO20, CGH$^+$21, ADD$^+$22, BBDP22]. We know how to build rate-1 OT from a wide variety of assumptions [IP07, DGI$^+$19, GHO20, CGH$^+$21, ADD$^+$22], but all these constructions make at least a linear number of public-key operations. In particular, computational efficiency (e.g., a sub-linear number of public-key operations) and communication efficiency (e.g., sub-linear communication) seem to have largely been in conflict with each other — for reasons we have not been to justify so far. The goal of our paper is to elucidate this conflicting situation.

[1] Throughout this paper, when we say PIR, we mean a single-server non-preprocessing PIR.

1.1 Our Results

We answer both of the above questions, and several other related ones, negatively. In particular, we give a lower-bound on the number of public-key operations that need to be performed by servers in PIR protocols, and use this lower-bound to derive similar results for related primitives. Our core idea is based on a compilation technique that allows one to remove public-key operation queries from a PIR protocol at the cost of proportionally increasing the communication complexity in the public-key operation free protocol. As applications of our main theorem, we obtain results that settle several open problems in MPC.

In the statement below, by an SO oracle we mean a *simulatable oracle*: roughly speaking, one that can be simulated via lazy sampling (aka, on the fly). Examples of such oracles include generic-group oracles, public-key encryption oracles, etc. See Section 2 for more details. Also, we use the term a "party's SO bit complexity" to indicate the total bit size of all SO queries made by the party.

Theorem 1 (Informal Main Theorem)**.** *If there exists a PIR for n-bit databases (denoted as n-bit PIR) with oracle access to simulatable oracle SO, arbitrary oracle O, server communication of $\eta < cn$ for $c < 1$, $r \in o(n)$ rounds of interaction with the user, and $q \in o(n)$ bits of communication with the SO oracle, then there exists a PIR with oracle access to O, server communication $\overline{\eta} \leq \overline{c}n$ for $\overline{c} < 1$, and no calls to SO.*

We derive the following corollary.

Corollary 1. *There exists no n-bit PIR protocol built solely[2] from a simulatable oracle SO and a random oracle O with $o(n)$ round complexity, with $o(n)$ server's SO bit complexity and with $\eta \leq cn$ server's communication for $c < 1$.*

For example, letting SO be a generic group oracle (GGM), we rule out all n-bit PIR protocols that have $o(n)$ rounds and where the server's communication and the server's total number of GGM queries are, respectively, cn and $o(n)$ for $c < 1$. This holds irrespective of the number of RO queries the protocol is allowed to make. This closely matches the known upper-bounds, as [OS07, DGI$^+$19] give n-bit PIR protocols based on the DDH assumption with server communication of $O(\lambda)$ and with the sever making $O(n)$ group operations.

The strength of the main theorem lies in its flexibility in instantiating the oracle SO: for example, one may let SO be an FHE oracle, and obtain similar results as long as the amount of server's communication with the FHE oracle respects the bounds. The work of [OS07] shows how to obtain PIR generically from (additively) homomorphic encryption, where the sever performs $O(n)$ homomorphic additions. Our work shows that this is close to optimal.

We will show that our computational lower-bounds for PIRs give rise to communication lower-bounds for OT extension.

Corollary 2 (OT Extension: Communication Lower-Bounds)**.** *There exists no ℓ-batch k-bit OT extension protocol with round complexity $r \in o(k\ell)$ and with server communication $\eta < c2k\ell$ for $c < 1$.*

In the above corollary, by ℓ-batch k-bit OT we mean performing ℓ OTs for pairs of k-bit strings. The IKNP protocol [IKNP03] in the ℓ single-bit OT case achieves sender communication of $> 2\ell$. Our result shows that the IKNP's sender communication complexity is close to optimal.

Finally, we relate PIR to other MPC protocols such as rate-1 OT to arrive at the following corollary. For brevity, we describe the statements when O is the GGM oracle, and only for rate-1 OT. In fact, we can show that achieving any rate strictly greater than 1/2 (measured as the information-theoretically optimal sender communication size divided by the sender's communication size in the actual protocol) requires making an almost linear number of group operations.

Corollary 3 (String OT Corollary)**.** *There exists no ℓ-bit string OT protocol in the GGM+RO model with sender communication of $\eta \leq c2\ell$ and $o(\ell)$ calls to the generic group for $c < 1$.*

Similar results can be proven for unbalanced PSI and in general for MPC with unbalanced inputs (aka, asymmetric MPC); see Section 5 for details.

[2] By "solely" we mean that the parties also have access to a PSPACE-complete oracle. This stops the party from using any hardness assumptions other than the ones provided by the oracles.

Pre-Processing PIR. Our techniques also apply to single-server pre-processing PIR (e.g., offline/online [CK20], doubly efficient PIR [LMW23a]) in the sense that by merging the offline and online phases together, we will get a PIR without pre-processing. So, our results will imply non-trivial lower-bounds in the pre-processing setting as well.

In particular, for any constant $c < 1$ and for n-bit databases, our results rule out pre-processing PIRs with $O(n^c)$ "total" public-key operations for the server and a total cn server-side communication. Here by total we mean offline+online together. Again this holds irrespective of the number of symmetric-key operation calls. Such pre-processing PIRs will imply single-server PIR without pre-processing and with the same properties, which we have already ruled out.

For example, the pre-processing single-server PIR of Corrigan-Gibbs and Kogan [CK20, Theorem 20] induces $O(n^{2/3})$ total server communication, with the server performing, respectively, $O(n)$ and zero public-key operations in the offline and online phases. Our lower-bounds show that the number of public-key operations of [CK20] is close to optimal.

2 Technical Overview

First, we will give a quick example of how to simulate a generic group efficiently to illustrate the concept of simulatable oracles. Then, we sketch the proof of the main theorem. We proceed by showing why the main theorem is useful by demonstrating a few MPC protocols which imply non-trivial PIR, allowing us to apply our lower-bounds to them.

2.1 Generic Group Model

A generic group of order p is the group $\mathbb{Z}_p$ together with a random injective encoding function $\sigma\colon \mathbb{Z}_p \to S$, where $S = \{0, \ldots, p-1\}$. The algorithms can access this group via the oracle Add which decodes two encoded elements, computes a linear combination of them and gives back the encoded result. More formally, $\mathsf{Add} : \mathbb{Z}_p^2 \times S^2 \to S$, $(a_1, a_2, \ell_1, \ell_2) \mapsto \sigma(a_1 v_1 + a_2 v_2)$, where $v_i = \sigma^{-1}(\ell_i)$ for $i \in \{1, 2\}$.[3] This way the algorithms interacting with the oracle can manipulate the encodings of group elements via the oracle Add.

To simulate a GGM oracle efficiently, the simulator dynamically generates the encoding function σ. More specifically, it maintains a partial set L of $\mathbb{Z}_p$-label pairs sampled by the simulator so far. (It is initially empty.) Whenever a query $(a_1, a_2, \ell_1, \ell_2)$ is made, the simulator checks if $(*, \ell_1) \in L$ (meaning that if for some v_1, $(v_1, \ell_1) \in L$); if not, the simulator samples a random v_1 from $\mathbb{Z}_p$ subject to $(v_1, *) \notin L$, and adds (v_1, ℓ_1) to L. The simulator does the same thing for ℓ_2. Now assuming $(v_1, \ell_1) \in L$ and $(v_2, \ell_2) \in L$, letting $a_3 = a_1 v_1 + a_2 v_2$ if $(a_3, \ell_3) \in L$ for some ℓ_3, the simulator responds to the query with ℓ_3; else, the simulator samples a random ℓ_3 subject to $(*, \ell_3) \notin L$, adds (a_3, ℓ_3) to L, and responds to the query with ℓ_3.

Other simulatable oracles that are useful in our main theorem are black-box oblivious transfer [GKM+00], black-box public-key encryption [GKM+00], and ideal obfuscation [JLLW23].

2.2 Proof Sketch of Main Theorem

The observation that leads to the main theorem is that in PIR protocols the server does not need to have privacy. This means the protocol would still be secure if the user learned all of the server's oracle queries. Also, the user knows all of its own oracle queries. Therefore, the user can just simulate the oracle for both of the parties. For this the server just has to send all of its queries to the user. This modification increases the server communication roughly by the amount that the server would have communicated to the oracle. We will prove that this transformation preserves user security. Since we require the server's 'oracle communication' to be $o(n)$ and the server's protocol communication to be $< cn$ for some $c < 1$, the modified protocol will have no oracle queries and the server communication will be $< \bar{c}n$ for $\bar{c} < 1$. Moreover, in the actual compiled protocol, to enable the compiled user to distinguish query messages from normal protocol

[3] One may set S to be a random subset of size p of a larger set $\{0,1\}^u$ for $u > \log p$. Our analysis will remain unchanged, so we simply assume $S = \{0, \ldots, p-1\}$. See [Zha22] for differences between various models.

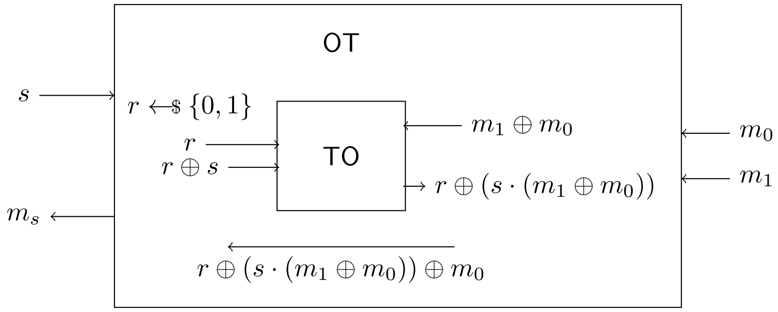
OT
s
$r \leftarrow\!\$ \{0,1\}$
TO
r
$r \oplus s$
$m_1 \oplus m_0$
$r \oplus (s \cdot (m_1 \oplus m_0))$
m_0
m_1
m_s
$r \oplus (s \cdot (m_1 \oplus m_0)) \oplus m_0$

Figure 1: Similar to [WW06] a visual representation of how to build an oblivious transfer OT from an oblivious transfer TO that goes in the opposite direction.

messages, we append a flag bit to the end of each server's protocol messages — causing a dependency on r, the number of rounds, in Theorem 1.

2.3 PIR Related Protocols

Now we exhibit a few protocols which imply non-trivial PIR, allowing us to apply our PIR impossibility results to these protocols.

Low Sender-Communication OT We show that an ℓ-batch k-bit OT protocol with sender communication $< c2k\ell$ for $c < 1$ implies a PIR. The folklore transformation works as follows: suppose w.l.o.g the database size is $2k\ell$. The server runs the OT protocol and encodes the first half of the database into the messages $(m_i^{(0)})_{i\in[\ell]}$ and the second half into $(m_i^{(1)})_{i\in[\ell]}$. Now, if a user wants to look up the j-th element of the database, it acquires $(m_i^{(0)})_{i\in[\ell]}$ if $j \leq k\ell$ and $(m_i^{(1)})_{i\in[\ell]}$ otherwise. The database entry that the client wants to learn is contained in the OT output. The server communication is $< c2k\ell$ and the user's input j is hidden from the server by the OT's receiver security.

Low Total Communication OT. We next attempt to prove lower-bounds for the case where the OT protocol has low total communication (as opposed to low sender communication). For convenience we focus on ℓ-batch single-bit OT. By low total communication we mean an amount that is close to the information-theoretically optimal communication, which is 2ℓ bits. The techniques for low sender-communication as above do not apply outright because the sender might cause the bulk of communication itself, an amount close to 2ℓ bits (e.g., suppose the sender's communication is $2\ell + \lambda$ bits and the receiver's communication is λ bits). We get around this issue via the following intuitive idea: When an OT protocol has low total communication it must have either low sender communication, from which we already showed how to obtain a non-trivial PIR, or it must have low receiver communication. In the latter case, we swap the roles of the two different parties with a role-flipping trick, implicitly used in [IKNP03] and explicitly in [WW06]. This role flipping trick turns the low receiver communication into low sender communication, which we can then turn into a non-trivial PIR. Figure 1 depicts the construction for $\ell = 1$. In essence, we show how to turn a communication-efficient OT into a sender-communication-efficient OT by introducing an additioal round.

OT Extension. The above results immediately imply communication lower bounds for OT extension: showing that performing OT extension for ℓ-batch k-bit OTs with $c2k\ell$ bits of sender communication for $c < 1$, and with an $O(\lambda)$ (and even $o(k\ell)$) number of public-key operations is impossible.

Unbalanced PSI. Private set intersection (PSI) is an MPC protocol between two parties each holding a set and the party called the receiver learns the intersection of the two sets. No other information should be revealed to is to any of the parties. In Unbalanced PSI, a special case of PSI, the receiver set is much smaller than the sender set and the communication should only scale with the receiver set. To build a non-trivial PIR from such a protocol, for a client index i, the client sets $x := i$ (padding it out if necessary), and for a database DB, the server forms the set $\{i \mid \mathsf{DB}[i] = 1\}$. An answer to $x \in^? S$ reveals $\mathsf{DB}[i]$. This observation allows us to prove that in unbalanced PSI with sub-linear communication, the receiver should perform close to linear public-key operations. This shows that the large number of public-key operations used in unbalanced PSI protocols of [DGI+19, GHO20, CGH+21] is inherent.

Non-Trivial PIR implies Oblivious Transfer We know that non-trivial PIR implies oblivious transfer [DMO00], and this is used to get our final impossibility results. The transformation utilizes the user security of the PIR protocol and deploys a compression argument to argue information loss. The entropy garnered from the information loss is then fed into a randomness extractor, the output of which can be used to guarantee sender security in the resulting oblivious transfer protocol.

2.4 Oracles

Notice that all these transformations only make black-box use of the protocols they transform. This means that if the starting protocol uses some oracle other than the one we want to remove, say the random oracle [BR93], then the resulting protocol will also use the random oracle even if we remove other oracle queries.

3 Related Work

Techniques Our 'compilation-out' techniques bear some similarities to ideas used by Gennaro and Trevisan [GT00] for giving lower-bounds on the query complexity of PRGs from OWPs. Essentially, they showed that if the number of queries is 'small', they can be encoded as part of the input, hence getting rid of OWP calls in the construction of a PRG. Gennaro et al. [GGK03] built on that idea to give lower-bounds on the efficiency of various cryptographic primitives. These works mostly deal with non-interactive primitives. Our techniques are used in a different way in that we leverage the lack of security requirements for a party to get rid of oracle calls of an interactive protocol.

Private-Information Retrieval. In all but this section of the paper we talk about non-trivial single-server private information retrieval, which is why we will sometimes leave out the descriptor "single-server". Traditionally, PIR [CGKS95, KO00] is a protocol between one user and possibly multiple servers. Just like in the non-trivial single-server case the user with an index $i \in [n]$ learns the i-th element of a database $\mathsf{DB} \in \{0,1\}^n$ held by all the servers without disclosing i to the servers. If the caveat of non-triviality is not made, then not only the server communication needs to be sub-linear in n, but also the total communication. Thus, a non-trivial PIR (requiring only the server-to-user communication to be small, as ruled out in our lower-bounds) is a weaker primitive than PIR. Multi-server PIR protocols assume some kind of non-collusion between the servers, which allows them to achieve statistical security as opposed to computational security.

By now PIR is a well studied primitive; here we focus on the single server setting. We know how to build PIR with communication complexity polylogarithmic in n from a wide range of assumptions [CMS99, IP07, DGI+19, CGH+21]. In the last few years, we have also made progress towards practically efficient PIR [CK20, KC21, SACM21, CHK22, ZLTS23, HHC+23a, ZPSZ23, HHC+23b] and asymptotically efficient PIR [CHR17, BIPW17, LMW23b] when the server and the client (or sometimes only the server) are allowed to preprocess the database. We even know some lower bounds for different preprocessing settings [BIM04, CK20, CHK22, PY22, Yeo23].

OT Extension. The intuition behind OT extension is that it only uses very few calls to an OT functionality to implement many more OTs. An equivalent description is that an OT extension protocol is an OT protocol that can make calls to an OT functionality. The protocol becomes valuable if the number of OT calls in the protocol is much less than the 'size' of the OT being implemented. OT calls are typically modelled as oracle calls to an OT functionality or the OT hybrid model.

Beaver [Bea96] constructed the first OT extension protocol, which makes non-black-box use of pseudorandom generators and which has two rounds. Ishai et al. [IKNP03] give the first OT extension protocol only making black-box use of symmetric-key cryptography while increasing the rounds to three. Garg et al. [GMMM18] show that three rounds are necessary in the OT hybrid model when only making black-box use of symmetric-key cryptography.

Rate-1 String OT. The notion of rate-1 OT has applications beyond the construction of PIR with polylog communication. In particular, a generalization of this notion, called trapdoor hash, has been used as a building block to build non-interactive zero knowledge for NP [BKM20]. This has made the notion of rate-1 OT appealing from both a theoretical and practical points of view.

Unbalanced Private-Set Intersection (PSI). Private keyword search allows a receiver, with a single element x, to learn whether x is a member of a large set S held by a sender, or sometimes called PIR for keywords [CGN98]. This is an instance of the so-called unbalanced PSI problem, defined earlier. A desirable feature of such unbalanced PSI protocols is sub-linear communication: the total amount of communication must be sub-linear the larger set size. We have protocols, from a wide variety of cryptographic assumptions, for unbalanced PSI whose communication complexity grows only polylogarithmically with the larger set size [IP07, DGI$^+$19, GHO20, CGH$^+$21].

Again, the Diffie-Hellman-based protocols come with a high sender computation cost: the number of group operations grows at least linearly in the bigger set size. As in PIR, one can prove that the strict running time of the sender in unbalanced PSI cannot be sub-linear in $|S_1|$, but that does not mean the number of public-key operations must also grow with n — especially, if the sender is allowed to make an arbitrarily-large number of symmetric-key operations. In fact, while the protocols in [DGI$^+$19, GHO20, CGH$^+$21] induce little communication, the large number of public-key operations involved is a major bottleneck.

In the absence of the sub-linear communication requirement, one may use oblivious-transfer (OT) extension techniques [Bea96, IKNP03] to design unbalanced PSI protocols with a number of public-key operations independent of $|S_1|$. These protocols can be made concretely efficient as well (e.g., [CM20]). However, all these OT-extension-based protocols fail to achieve sub-linear communication, and our lower-bound results explain this situation.

4 Preliminaries

We denote the security parameter by λ. We say a function negl is negligible if for any polynomial poly we have $\mathsf{negl}(\lambda) \in o(\frac{1}{\mathsf{poly}(\lambda)})$. For two integers i and i', we define $[i, i'] := \{i, i+1, \ldots, i'\}$. We let $[n] := \{1, \ldots, n\}$.

For $i \in \{r, s\}$, denoting receiver (r) and sender (s), we let $\mathsf{view}_i^{\Pi}(1^\lambda, x, y)$ denote the view of Party i in an honest execution of the protocol Π on 1^λ and on the parties' respective inputs, where the view contains the private input and the random coins of the respective party, the protocol's transcript, and the transcript of oracle queries and their responses. We may omit the security parameter 1^λ whenever it is clear from the context.

4.1 Oblivious Transfer

Definition 1 (Oblivious Transfer (OT))**.** *An ℓ-batch k-bit string OT protocol* OT *is a protocol between two interactive PPT programs* $(\mathsf{OTR}, \mathsf{OTS})$*, where* OTR *and* OTS *denote, respectively, the receiver and the sender.*

$\mathsf{OTR}(1^\lambda, 1^\ell, 1^k, s)$: *An interactive algorithm that takes in a security parameter 1^λ, batching parameter 1^ℓ, message parameter 1^k, and choice vector $s \in \{0,1\}^\ell$, and outputs $m \in \{0,1\}^\ell$.*

$\mathsf{OTS}(1^\lambda, 1^\ell, 1^k, m^{(0)}, m^{(1)})$: *An interactive algorithm that takes in a security parameter* 1^λ*, batching parameter* 1^ℓ*, message parameter* 1^k *and two message vectors* $m^{(0)}, m^{(1)} \in \mathsf{M}^\ell$*, for* $\mathsf{M} = \{0,1\}^k$*, and outputs* $\perp$

We require the following.

Correctness. OT *is* $\alpha(\cdot)$*-correct if for any* λ*,* $s \in \{0,1\}^\ell$*,* $(m_i^{(0)}, m_i^{(1)})_{i\in[\ell]} \in (\mathsf{M} \times \mathsf{M})^\ell$*, the probability over an honest interaction between* $\mathsf{OTR}(1^\lambda, 1^k, 1^\ell, s)$ *and* $\mathsf{OTS}(1^\lambda, 1^k, 1^\ell, m^{(0)}, m^{(1)})$ *that* OTR *outputs* $(m_1^{(s_1)} \dots m_\ell^{(s_\ell)})$ *is* $\geq \alpha(\lambda)$*. The protocol is perfectly correct if* $\alpha = 1$*. By default we require prefect correctness.*

Semi-Honest Receiver Security. *For any strings* $s_0, s_1 \in \{0,1\}^\ell$*,* $m^{(0)}, m^{(1)} \in \mathsf{M}^\ell$ *we have that* $\mathsf{view}_s^{\mathsf{OT}}(s_0, (m^{(0)}, m^{(1)}))$ *and* $\mathsf{view}_s^{\mathsf{OT}}(s_1, (m^{(0)}, m^{(1)}))$ *are computationally indistinguishable.*

Semi-Honest Sender Security. *For any* $s \in \{0,1\}^\ell$ *and* $m^{(0)}, m^{(1)}, z^{(0)}, z^{(1)} \in \mathsf{M}^\ell$ *such that* $\{(m_i^{s_i})\} = \{(z_i^{s_i})\}$*, we have that the two views* $\mathsf{view}_r^{\mathsf{OT}}(s, (m^{(0)}, m^{(1)}))$ *and* $\mathsf{view}_r^{\mathsf{OT}}(s, (z^{(0)}, z^{(1)}))$ *are computationally indistinguishable.*

OT Terminologies. We may sometimes refer to an ℓ-batch single-bit OT as an ℓ-batch OT. Also, whenever we say a k-bit string OT we mean $\ell = 1$.

We define notions of *rate* as asymptotic ratios between the actual communication under a given protocol and the best achievable communication under a (possibly) insecure protocol; i.e., for ℓ-batch single-bit OT the sender must communicate at least ℓ bits to the receiver, if perfect correctness is required. Therefore, the optimal download communication is ℓ. Similarly, the optimal total communication is 2ℓ.

Expected Download Rate. An ℓ-batch single-bit OT protocol has expected download rate c if for all λ, s, $m^{(0)}$, $m^{(1)}$, and all but finitely many ℓ

$$\frac{\ell}{d(\lambda, \ell)} \geq c,$$

where $d(\lambda, \ell)$ is expected communication from $\mathsf{OTS}(1^\lambda, 1^\ell, m^{(0)}, m^{(1)})$ to $\mathsf{OTR}(1^\lambda, 1^\ell, s)$.

Expected (Overall) Rate. An ℓ-batch single-bit OT protocol has expected (overall) rate c if for all λ, s, $m^{(0)}$, $m^{(1)}$, and all but finitely many ℓ

$$\frac{2\ell}{t(\lambda, \ell)} \geq c,$$

where $t(\lambda, \ell)$ is the expected total communication.

We now define the notion of OT extension in the black-box OT model, which is stronger than the OT-hybrid model.

Definition 2 (Black-Box OT Extension)**.** *A black-box OT extension* $\mathsf{OTExt}^{\mathsf{OT}} = (\mathsf{OTRExt}^{\mathsf{OT}}, \mathsf{OTSExt}^{\mathsf{OT}})$ *is an* ℓ*-batch* k*-bit OT protocol that for a fixed polynomial* poly*, independent of* ℓ*, makes at most* $\mathsf{poly}(\lambda)$ *calls to the base single-bit OT oracle* $\mathsf{OT} = (\mathsf{OTR}, \mathsf{OTS})$.

4.2 Private-Information Retrieval (PIR)

Definition 3 (Non-Trivial PIR)**.** *A non-trivial (single-server) private information retrieval* ntPIR *is an interactive protocol between two interactive PPT programs* $(\mathsf{PIRU}, \mathsf{PIRS})$*, where* PIRU *and* PIRS *denote, respectively, the client (user) and the server.*

$\mathsf{PIRU}(1^\lambda, 1^n, i)$ *: An interactive algorithm that takes in a security parameter* 1^λ*, the database size* n*, and a choice index* $i \in [n]$*, and at the end of the interaction outputs* $y \in \{0,1\}$*.*

$\mathsf{PIRS}(1^\lambda, 1^n, \mathsf{DB})$ *: An interactive algorithm that takes in a security parameter 1^λ, database size n and a database $\mathsf{DB} \in \{0,1\}^n$, and outputs $\perp$.*

We require the following properties.

Correctness. *The PIR protocol is $\alpha(\cdot)$-correct if for any λ, n, $i \in [n]$ and $\mathsf{DB} \in \{0,1\}^n$, the probability over an honest interaction between $\mathsf{PIRU}(1^\lambda, 1^n, i)$ and $\mathsf{PIRS}(1^\lambda, 1^n, \mathsf{DB})$ that PIRU outputs DB_i is $\geq \alpha(\lambda)$. The protocol is perfectly correct if $\alpha = 1$. By default we require perfect correctness.*

Semi-Honest Client Security. *For any n, $i, i' \in [n]$, $\mathsf{DB} \in \{0,1\}^n$, $\mathsf{view}_s^{\mathsf{ntPIR}}(i, \mathsf{DB})$ and $\mathsf{view}_s^{\mathsf{ntPIR}}(i', \mathsf{DB})$ are computationally indistinguishable.*

Non-Trivial Expected Download Communication. *There exists a polynomial poly such that for all sufficiently large λ, for all $n \geq \mathsf{poly}(\lambda)$, for all $i \in [n]$, and $\mathsf{DB} \in \{0,1\}^n$, the expected communication from $\mathsf{PIRS}(1^\lambda, 1^n, i)$ to $\mathsf{PIRU}(1^\lambda, 1^n, \mathsf{DB})$ is $d(\lambda, n) < n$.*

The following result shows that non-trivial PIR implies public-key cryptography in a black-box way. We use this theorem for our lower-bound results.

Theorem 2 ([DMO00])**.** *There exists a black-box construction of OT from a non-trivial PIR protocol.*

5 Protocols that Imply Non-Trivial PIR

We substantiate the relevance of non-trivial PIR by showing that communication-efficient versions of some popular MPC protocols can be transformed into non-trivial PIR in a black-box manner. These transformations later let us transfer the lower-bounds regarding PIR to these protocols.

In the following, we focus on different variants of oblivious transfer and unbalanced private set intersection to demonstrate the concept. The same ideas apply to many other protocols such as vector oblivious linear evaluation and oblivious polynomial evaluation.

5.1 Oblivious Transfer

We show how to transform a protocol for k-bit string oblivious transfer $\mathsf{OT} = (\mathsf{OTR}, \mathsf{OTS})$ that makes calls to an oracle $\mathcal{O}$ into a PIR protocol $(\mathsf{PIRU}, \mathsf{PIRS})$ with database size $n = 2k$ that makes calls to the same oracle $\mathcal{O}$ in a black-box manner. The construction is folklore and works by splitting the database in half, using each half as one of the two strings, and choosing the OT choice bit based on the PIR client's index accordingly.

- $\mathsf{PIRU}^{\mathcal{O}}(1^\lambda, 1^n, i)$: For $n = 2k$, set the choice bit $b \leftarrow \lfloor (i-1)/k \rfloor$. Run the OT receiver $m_b \leftarrow \mathsf{OTR}^{\mathcal{O}}(1^\lambda, b)$ to get the chosen string m_b. Return $m_b[i - kb]$
- $\mathsf{PIRS}^{\mathcal{O}}(1^\lambda, 1^n, \mathsf{DB})$: Let strings $m_0 \leftarrow \mathsf{DB}[1, \ldots, k]$ and $m_1 \leftarrow \mathsf{DB}[k+1, \ldots, 2k]$. Run the OT sender $\mathsf{OTS}^{\mathcal{O}}(1^\lambda, m_0, m_1)$.

Lemma 1 (Folklore)**.** *The PIR protocol $(\mathsf{PIRU}, \mathsf{PIRS})$ has the same correctness error, the same sender/receiver query complexity, and the same sender/receiver communication as those of the OT protocol OT.*

That means if the expected sender communication in OT is less than $n = 2k$, then $(\mathsf{PIRU}, \mathsf{PIRS})$ is a non-trivial PIR protocol.

Proof of Correctness. Correctness follows from the correctness of the OT protocol and for $b = \lfloor (i-1)/k \rfloor$ we have $m_b[i - kb] = \mathsf{DB}[kb+1, \ldots, kb+k][i - kb] = \mathsf{DB}[i]$. □

Proof of Client Security. Suppose there exists $i, i' \in [N]$, $\mathsf{DB} \in \{0,1\}^N$ such that an adversary $\mathcal{A}$ can distinguish $\mathsf{view}_s^{\mathsf{PIR}}(i, \mathsf{DB})$ from $\mathsf{view}_s^{\mathsf{PIR}}(i', \mathsf{DB})$ with non-negligible probability. Then $\lfloor (i-1)/k \rfloor \neq \lfloor (i'-1)/k \rfloor$ else $\mathsf{view}_s^{\mathsf{PIR}}(i, \mathsf{DB})$ and $\mathsf{view}_s^{\mathsf{PIR}}(i', \mathsf{DB})$ follow the exact same distribution. The same adversary $\mathcal{A}$ distinguishes between $\mathsf{view}_s^{\mathsf{OT}}(\lfloor (i-1)/k \rfloor, (\mathsf{DB}[1, \dots, k], \mathsf{DB}[k+1, \dots, 2k]))$ and $\mathsf{view}_s^{\mathsf{OT}}(\lfloor (i'-1)/k \rfloor, (\mathsf{DB}[1, \dots, k], \mathsf{DB}[k+1, \dots, 2k]))$ with the same non-negligible probability since the views are exactly the same as $\mathsf{view}_s^{\mathsf{PIR}}(i, \mathsf{DB})$ and $\mathsf{view}_s^{\mathsf{PIR}}(i', \mathsf{DB})$ respectively. □

Remark 1. *One can transform any ℓ-batch k-bit OT protocol into a $k\ell$-bit string OT protocol by reusing the same choice bit across all the ℓ batches. This works without any issues because we only talk about semi-honest security.*

OT With Low Total Communications. Using the symmetric nature of OT we transform an OT protocol with low communication (not just low sender communication) into a low sender communication OT protocol in a black-box manner. This allows us to apply our PIR lower-bounds to OT with low expected communication. Our transformation works by noting that every communication efficient OT protocol has either low sender or low receiver communication; if the receiver communication is low, our transformation will swap the roles of the sender and receiver, to obtain an OT protocol with low sender communication, as desired.

The following transformation was implicitly used in [IKNP03] and explicitly in [WW06]. The transformation works as follows: Let $\mathsf{OT} = (\mathsf{OTR}, \mathsf{OTS})$ be an ℓ-batch single-bit OT with expected total communication $t(\lambda, \ell)$, expected download communication $d(\lambda, \ell)$, expected upload communication $u(\lambda, \ell)$, and oracle accesses to $\mathcal{O}$. We define a $\mathsf{OT}' = (\mathsf{OTR}', \mathsf{OTS}')$ as follows

```
$\mathsf{OTR}'^{\mathcal{O}}(1^\lambda, 1^\ell, s)$ :
    1. If the expected download communication of OT is $d(\lambda, \ell) < u(\lambda, \ell) + \ell$:
        (a) Run $(m'_1, \dots, m'_\ell) \leftarrow \mathsf{OTR}^{\mathcal{O}}(1^\lambda, 1^\ell, s)$, the $\ell$-batch OT receiver on the choice string $s$.
        (b) Return $(m'_1, \dots, m'_\ell)$
    2. Else:
        (a) Sample $r \xleftarrow{\$} \{0,1\}^\ell$ uniformly at random.
        (b) Run $\mathsf{OTS}^{\mathcal{O}}(1^\lambda, 1^\ell, r, s \oplus r)$, the $\ell$-batch OT sender on messages $m_0 = r$ and $m_1 = r \oplus s$.
        (c) Receive $v$ in the round after OTS is done.
        (d) Return $v \oplus r$

$\mathsf{OTS}'^{\mathcal{O}}(1^\lambda, 1^\ell, m^{(0)}, m^{(1)})$ :
    1. If the expected download communication of OT is $d(\lambda, \ell) < u(\lambda, \ell) + \ell$:
        (a) Run $\mathsf{OTS}^{\mathcal{O}}(1^\lambda, 1^\ell, m^{(0)}, m^{(1)})$, the $\ell$-batch OT sender on messages $m_0 = m^{(0)}$ and $m_1 = m^{(1)}$.
        (b) Return
    2. Else:
        (a) Run $z \leftarrow \mathsf{OTR}^{\mathcal{O}}(1^\lambda, 1^\ell, m^{(1)} \oplus m^{(0)})$, the $\ell$-batch OT receiver on the choice string $m^{(1)} \oplus m^{(0)}$
            to receive the string $z$.
        (b) Send $z \oplus m^{(0)}$ in the round after OTR is done
        (c) Return
```

Lemma 2. *The constructed OT protocol* $\mathsf{OT}' = (\mathsf{OTR}', \mathsf{OTS}')$ *has the same correctness as the base OT* $\mathsf{OT} = (\mathsf{OTR}, \mathsf{OTS})$. *Moreover,* $\mathsf{OT}' = (\mathsf{OTR}', \mathsf{OTS}')$ *is secure if* $\mathsf{OT} = (\mathsf{OTR}, \mathsf{OTS})$ *is secure.*

Assuming $\mathsf{OT} = (\mathsf{OTR}, \mathsf{OTS})$ *has expected overall rate $r > 2/3$, the constructed OT has expected download rate $w > 1/2$.*

Correctness. If $d(\lambda, \ell) < u(\lambda, \ell) + \ell$ then both parties behave exactly like OT and therefore correctness is inherited.

If $d(\lambda, \ell) \geq u(\lambda, \ell) + \ell$ the sender OTS' learns

$$\left(r_1 \oplus s_1 \cdot (m_1^{(1)} \oplus m_1^{(0)}), \ldots, r_\ell \oplus s_\ell \cdot (m_\ell^{(1)} \oplus m_\ell^{(0)})\right)$$

it then sends back

$$\left(r_1 \oplus s_1 \cdot (m_1^{(1)} \oplus m_1^{(0)}) \oplus m_1^{(0)}, \ldots, r_\ell \oplus s_\ell \cdot (m_\ell^{(1)} \oplus m_\ell^{(0)}) \oplus m_\ell^{(0)}\right)$$

then the receiver OTR' computes

$$\begin{aligned}&\left(s_1 \cdot (m_1^{(1)} \oplus m_1^{(0)}) \oplus m_1^{(0)}, \ldots, s_\ell \cdot (m_\ell^{(1)} \oplus m_\ell^{(0)}) \oplus m_\ell^{(0)}\right)\\ =&\left(m_1^{(s_1)}, \ldots, m_\ell^{(s_\ell)}\right)\end{aligned}$$

□

Security. The security in the case that $d(\lambda, \ell) < u(\lambda, \ell) + \ell$ directly follows from the security of OT.

In the other case it follows from the security of OT and the work of [WW06] which proves that this exact construction is secure. For receiver security we have that the sender (according to sender security of OT) only learns z. Each bit z_i is either r_i or $s_i \oplus r_i$, in both cases it is uniformly random because r is uniformly random. Sender security of OT' follows because the execution of OT leaks nothing to the receiver (according to the receiver security of OT). That means all the receiver learns is $z \oplus m^{(0)}$. By correctness of OT' this is exactly $(m_1^{(s_1)} \oplus r_1, \ldots, m_\ell^{(s_\ell)} \oplus r_\ell)$ and therefore contains no information about $(m_1^{(1-s_1)}, \ldots, m_\ell^{(1-s_\ell)})$. □

Expected Download Communication. Let r be the rate, $t(\lambda, n)$ be the expected total communication which is the sum of the expected receiver-to-sender communication $u(\lambda, \ell)$ and the expected sender-to-receiver communication $d(\lambda, \ell)$. Then for all but finitely many ℓ we have $t(\lambda, \ell) < \frac{2\ell}{r}$. In the following, the expected sender-to-receiver communication of OT' will be called $d'(\lambda, \ell)$.

If $d(\lambda, \ell) < u(\lambda, \ell) + \ell$ then

$$\begin{aligned}d'(\lambda, \ell) = d(\lambda, \ell) = t(\lambda, \ell) - u(\lambda, \ell) \leq t(\lambda, \ell) - d(\lambda, \ell) + \ell \qquad &\Leftrightarrow\\ d'(\lambda, \ell) \leq \frac{t(\lambda, \ell) + \ell}{2}&\end{aligned}$$

Else the new expected sender-to-receiver communication is

$$\begin{aligned}d'(\lambda, \ell) = u(\lambda, \ell) + \ell = t(\lambda, \ell) - d(\lambda, \ell) + \ell \leq t(\lambda, \ell) - u(\lambda, \ell) - \ell + \ell \qquad &\Leftrightarrow\\ d'(\lambda, \ell) \leq \frac{t(\lambda, \ell) + \ell}{2}&\end{aligned}$$

Either way, $d'(\lambda, \ell) \leq \frac{t(\lambda,\ell)+\ell}{2}$ which means that for all but finitely many ℓ we have $d'(\lambda, \ell) < (\frac{1}{r} + \frac{1}{2})\ell$. Therefore, the expected download rate is $\frac{2}{(\frac{1}{r}+\frac{1}{2})}$ which is $> 1/2$ for $r > 2/3$. □

5.2 Unbalanced Private-Set Intersection

In unbalanced private set intersection we have a set A of n λ-bit messages, held by a sender $\mathsf{PSIS}(1^\lambda, 1^n, A)$, and a singleton set B, held by a receiver $\mathsf{PSIR}(1^\lambda, 1^n, B)$. The goal is for the receiver to learn $A \cap B$ while the sender should learn nothing. Semi-honest receiver security can be defined along the lines of receiver (client) security of PIR (Definition 3).

We show how to transform a protocol for unbalanced private-set intersection $\mathsf{PSI} = (\mathsf{PSIS}, \mathsf{PSIR})$ that makes calls to oracle $\mathcal{O}$ into a PIR protocol $(\mathsf{PIRU}, \mathsf{PIRS})$ that makes calls to the same oracle $\mathcal{O}$ in a black-box manner. We do this by simply encoding the PIR-database and the PIR-query as sets.

- $\mathsf{PIRU}^{\mathcal{O}}(1^\lambda, 1^n, i)$: Set $A := \{i\}$. Run the PSI receiver $I \leftarrow \mathsf{PSIR}^{O}O(1^\lambda, A)$ to get the intersection I. Return 1 if $\{i\} = I$ and 0 otherwise.
- $\mathsf{PIRS}^{\mathcal{O}}(1^\lambda, 1^n, \mathsf{DB})$: Form the set $B := \{x \mid \mathsf{DB}[x] = 1\}$. Run the PSI sender $\mathsf{PSIS}^{\mathcal{O}}(1^\lambda, B)$.

Lemma 3 (Folklore)**.** *The PIR protocol* $(\mathsf{PIRU}, \mathsf{PIRS})$ *has the same correctness error, sender and receiver communication and sender and receiver query complexity as those of* PSI*. Moreover, the resulting PIR protocol has client security if* PSI *provides receiver security.*

That means if the sender communication in PSI *is less than* n*, then* $(\mathsf{PIRU}, \mathsf{PIRS})$ *is a non-trivial PIR protocol.*

Proof of Correctness. Correctness follows from the correctness of the PSI protocol and the intersection of $\{i\}$ and B being $\{i\}$ if $i \in B \Leftrightarrow \mathsf{DB}[i] = 1$ and $\emptyset$ otherwise.

Proof of Client Security. Suppose there exists $i, i' \in [n]$, $\mathsf{DB} \in \{0,1\}^n$ such that an adversary $\mathcal{A}$ can distinguish $\mathsf{view}_s^{\mathsf{PIR}}(i, \mathsf{DB})$ from $\mathsf{view}_s^{\mathsf{PIR}}(i', \mathsf{DB})$ with non-negligible probability. The same adversary $\mathcal{A}$ distinguishes between $\mathsf{view}_s^{\mathsf{PSI}}(\{i\}, B)$ and $\mathsf{view}_s^{\mathsf{PSI}}(\{i'\}, B)$ with the same non-negligible probability since the views are exactly the same. □

6 Lower-Bounds on the Number Oracle Queries in PIR

In this section, we show how to transform a private information retrieval (PIR) protocol with access to some *simulatable* oracle SO into one that does not query that oracle. To have something concrete in mind one may imagine SO being the generic group model, though the technique is much more general. We will later go into common instantiations of the oracle. This transformation allows us to transfer lower-bounds from PIR without oracle access to PIR with oracle access.

Simulatable Oracles. A simulatable oracle is an oracle SO which can efficiently be simulated by a stateful simulator Sim. More formally, a computationally unbounded adversary $\mathcal{A}$ cannot win the following game with a non-negligible advantage in polynomially many rounds r, where Sim is a PPT algorithm:

1. Sample random bit $b \xleftarrow{\$} \{0,1\}$.
2. Initialize the state of the oracle as $\mathsf{st} \leftarrow \perp$.
3. Initialize the state of the adversary $\mathsf{ast} \leftarrow \perp$.
4. The adversary produces a first query qu.
5. For $i \in [r]$:
   (a) If $b = 0$:
      - Let the response be $\mathsf{resp} \leftarrow \mathsf{SO}(\mathsf{qu})$

   (b) Else:
      - Let response and new oracle state be $(\mathsf{resp}, \mathsf{st}) \leftarrow \mathsf{Sim}(\mathsf{qu}, \mathsf{st})$

   (c) Let new query and adversary state be $(\mathsf{qu}, \mathsf{ast}) \leftarrow \mathcal{A}(\mathsf{ast}, \mathsf{resp})$
6. Let the adversary output its guess $b' \leftarrow \mathcal{A}(\mathsf{ast})$.
7. The adversary wins if $b = b'$.

Typical examples of simulatable oracles include the random oracle and the generic group oracle.

Construction 1. *Let* $\mathsf{PIR} := (\mathsf{PIRU}^{\mathsf{SO},\mathsf{O}}, \mathsf{PIRS}^{\mathsf{SO},\mathsf{O}})$ *be a bit PIR protocol that uses a simulatable oracle* SO *and another oracle* O*. We show how to compile out the* SO*-calls of* $(\mathsf{PIRU}^{\mathsf{SO},\mathsf{O}}, \mathsf{PIRS}^{\mathsf{SO},\mathsf{O}})$*, obtaining an* SO*-free PIR protocol* $\overline{\mathsf{PIR}} := (\overline{\mathsf{PIRU}}^{\mathsf{O}}, \overline{\mathsf{PIRS}}^{\mathsf{O}})$*.*

For notational convenience, in the following whenever calling PIRU *or* PIRS*, we omit the private-state part of the input.*

The protocol messages sent from $\overline{\mathsf{PIRU}}$ *to* $\overline{\mathsf{PIRS}}$ *are tagged with either 'protocol' (or bit zero) signifying a normal protocol message, or with 'query' (or bit 1) signifying a query message.*

$\overline{\mathsf{PIRU}}^{\mathsf{O}}(1^\lambda, 1^n, i)$ *:*

- *Initialize the state of the simulatable oracle* $\mathsf{st} \leftarrow \perp$*.*
- *Run the interactive PPT* $\mathsf{PIRU}^{\mathsf{SO},\mathsf{O}}$ *with the following interactions:*
  1. *When* $\mathsf{PIRU}^{\mathsf{SO},\mathsf{O}}$ *calls* O *on a query* qu *forward the query to* O *and respond with the received response.*
  2. *When* $\mathsf{PIRU}^{\mathsf{SO},\mathsf{O}}$ *calls* SO *on a query* qu*, simulate the response and update the oracle simulators state* $(\mathsf{resp}, \mathsf{st}) \leftarrow \mathsf{Sim}(\mathsf{qu}, \mathsf{st})$*.*
  3. *Upon* $\overline{\mathsf{PIRU}}$ *receiving a message of the form* $(\text{'}query', \mathsf{msgs})$*, interpret* msgs *as a query* qu*, simulate the oracle response and update the oracle simulators state as* $(\mathsf{resp}, \mathsf{st}) \leftarrow \mathsf{Sim}(\mathsf{qu}, \mathsf{st})$ *and return* resp *to the sender* $\overline{\mathsf{PIRS}}$*. If the message has the form* $(\text{'}protocol', \mathsf{msgs})$*, run* $\mathsf{PIRU}^{\mathsf{SO},\mathsf{O}}$ *on the protocol message* msgs *until it produces the next message* msgr *and send that to* $\overline{\mathsf{PIRS}}$*. The oracle queries are handled as described above.*

$\overline{\mathsf{PIRS}}^{\mathsf{O}}(1^\lambda, 1^n, \mathsf{DB})$ *:*

- *Run the interactive PPT* $\mathsf{PIRS}^{\mathsf{SO},\mathsf{O}}$ *with the following interactions:*
  1. *When* $\mathsf{PIRS}^{\mathsf{SO},\mathsf{O}}$ *calls* O *on a query* qu*, forward the query to* O *and respond with the received response.*
  2. *When* $\mathsf{PIRS}^{\mathsf{SO},\mathsf{O}}$ *calls* SO *on a query* qu*, send a tagged query pair* $(\text{'}query', \mathsf{qu})$ *to* $\overline{\mathsf{PIRU}}$ *and use the response* msgr *as a query response for* qu *to* PIRS*.*
  3. *Else, run* $\mathsf{PIRS}^{\mathsf{SO},\mathsf{O}}$ *until it produces a message* msgs *and send the tagged message* $(\text{'}protocol', \mathsf{msgs})$ *to* $\overline{\mathsf{PIRU}}$*, then wait for the response* msgr *and continue.*

Theorem 3. *If* PIR *is a non-trivial private information retrieval with server communication of* $\eta < cn$ *for* $c < 1$*,* $r \in o(n)$ *rounds of interaction with the user, and* $q \in o(n)$ *bits of communication with the* SO *oracle then* $\overline{\mathsf{PIR}}$ *is a non-trivial private information retrieval with server communication* $\overline{\eta} \leq \overline{c}n$ *for* $\overline{c} < 1$ *and no calls to* SO*.*

Server communication. The server's additional communication overhead includes 1 bit per round as well as a total of $O(q)$ bits. Since the number rounds is $o(n)$, the total server communication complexity becomes $cn + o(n)$, which is less than $\overline{c}n$ for some $\overline{c} < 1$. □

Correctness. Notice that the above protocol will have different output from an execution of PIR either

1. if a with Sim simulated oracle behave differently from the real oracle behaviour or
2. if a message in the execution of PIR happens to start with t.

Both of these events happen with negligible probability. Therefore, if PIR has statistical correctness then so does $\overline{\mathsf{PIR}}$. □

Client Security. Suppose there exists $i, i' \in [n]$, $\mathsf{DB} \in \{0,1\}^n$ such that an adversary $\overline{\mathcal{A}}$ can distinguish $\mathsf{view}_s^{\overline{\mathsf{PIR}}}(i, \mathsf{DB})$ from $\mathsf{view}_s^{\overline{\mathsf{PIR}}}(i', \mathsf{DB})$ with non-negligible probability. We construct a new adversary $\mathcal{A}$ to distinguish between $\mathsf{view}_s^{\mathsf{PIR}}(i, \mathsf{DB})$ and $\mathsf{view}_s^{\mathsf{PIR}}(i', \mathsf{DB})$. The new adversary $\mathcal{A}$ gets as input a view v either from $\mathsf{view}_s^{\mathsf{PIR}}(i, \mathsf{DB})$ or $\mathsf{view}_s^{\mathsf{PIR}}(i', \mathsf{DB})$ and does the following:

1. Generate an empty view $\overline{v}$.
2. Copy all O-oracle calls from v to $\overline{v}$.
3. Run PIRS on the randomness and DB as defined in the view v and simulate its interaction as follows:
   (a) For PIRS's calls to the SO oracle with query qu and gets response resp enter $('query', \mathsf{qu})$ as a server message into $\overline{v}$ and resp as a user message.
   (b) For PIRS's messages msgs enter $('protocol, \mathsf{msgs})$ in the transcript $\overline{v}$ as a server message and enter the users response msgr as a users message.
4. Run $b \leftarrow \overline{\mathcal{A}}(\overline{v})$, on the view $\overline{v}$ produced by $\overline{\mathsf{PIRS}}$
5. Return b

$\mathcal{A}$ will distinguish $\mathsf{view}_s^{\mathsf{PIR}}(i, \mathsf{DB})$ and $\mathsf{view}_s^{\mathsf{PIR}}(i', \mathsf{DB})$ with negligibly close to the probability as $\overline{\mathcal{A}}$ can distinguish $\mathsf{view}_s^{\overline{\mathsf{PIR}}}(i, \mathsf{DB})$ from $\mathsf{view}_s^{\overline{\mathsf{PIR}}}(i', \mathsf{DB})$. This is because $\overline{v}$ follows the same distribution as $\mathsf{view}_s^{\overline{\mathsf{PIR}}}(i, \mathsf{DB})$ (except that the SO queries are produced by the real oracle, not the simulator) if v was from $\mathsf{view}_s^{\overline{\mathsf{PIR}}}(i, \mathsf{DB})$ and $\overline{v}$ follows the same distribution as $\mathsf{view}_s^{\overline{\mathsf{PIR}}}(i', \mathsf{DB})$ (same caveat here) if v was from $\mathsf{view}_s^{\mathsf{PIR}}(i', \mathsf{DB})$. If the adversary could notice the simulation of SO then it would break its simulatability. □

Remark 2. *Theorem 3 is applicable to any two-PC protocol with one-sided receiver security. Of course, in the absence of further restrictions, such protocols are trivial to realize (e.g., by the sender sending its input in the clear to the receiver). One restriction that makes the problem non-trivial is to require the sender-to-receiver communication to be sub-linear in the sender's input size, as in PIR.*

The utility of Theorem 3, beyond PIR itself, becomes apparent when one considers other protocols that imply non-trivial PIR while instantiating their underlying oracles via ideal forms of powerful primitives. We first discuss the implications of the theorem in terms of particular instantiations of the oracle, and in the next section we consider protocols that imply PIR.

Theorem 3 allows us to also rule out powerful non-black-box techniques for building PIR. We demonstrate this by letting SO include an OT oracle and an ideal obfuscation oracle that can obfuscate circuits with generic OT gates and random oracle gates. (See [AS15, GHMM18] for capturing similar non-black-box techniques via oracle-aided circuits.)

Corollary 4. *For any constants $c < 1$, there exists no n-bit PIR protocol with server communication $\eta \leq cn$, round complexity $r \in o(n)$, and with oracle access to a PSPACE-complete oracle, a random oracle, a generic OT oracle, and an obfuscation oracle for circuits with OT and random oracle gates, and where the server only communicates $q \in o(n)$ bits to the ideal obfuscation and OT oracles.*

Proof. In Lemma 1 we show an OT protocol with the above mentioned characteristics implies a non-trivial PIR. Let SO consist of an OT oracle [GKM$^+$00] and an ideal obfuscation oracle [AS15, JLLW23] for obfuscating circuits with OT/RO gates. (Such an SO oracle is simulatable.) By invoking Theorem 3 one gets a non-trivial PIR with oracle access to the random oracle and an PSPACE-complete oracle. This in turn can be transformed into an OT protocol (while retaining the O oracles) via [DMO00]. The existence of such an object however was ruled out by [GKM$^+$00]. □ □

Back to the black-box setting, other illustrative examples include the use of GGMs for building non-trivial PIR.

Corollary 5. *For any constants $c < 1$, a non-trivial n-bit PIR protocol with server communication of cn, round complexity $r \in o(n)$ and where the server makes sublinear in n many generic group queries requires MPC-hard assumptions, beyond the generic group.*

Proof. In Theorem 3, if one instantiates SO by a generic group [Sho97] and let the O oracle be empty, then one gets a non-trivial PIR without any oracle calls. This in turn can be transformed into an OT protocol without any oracles via [DMO00]. OT is an MPC-complete protocol. □ □

The above corollary is almost tight as there exists GGM-based PIR protocols with a linear number of GGM queries.

Lemma 4 ([DGI+19])**.** *Based on the DDH assumption, there exists a non-trivial n-bit PIR protocol with server communication of $O(\lambda)$ and with the sever making $O(n)$ group operations.*

Finally, we may derive a statement for FHE oracles.

Corollary 6. *For any constants $c < 1$, a non-trivial n-bit PIR protocol with server communication of cn, round complexity $r \in o(n)$ where the server makes $q \in o(n)$ black-box use of fully homomorphic encryption*[4] *requires MPC-hard assumptions, even beyond the fully homomorphic encryption.*

Proof. Let SO be an FHE oracle [GMM17] (defined similarly to a generic PKE oracle of [GKM+00]). Let the O oracle be empty. Invoking Theorem 3 one gets a non-trivial PIR without any oracle calls. This in turn can be transformed into an OT protocol without oracles via [DMO00]. OT is an MPC-complete protocol. □ □

7 Communication Lower-Bounds for OT Extension

Theorem 3 provides lower-bounds on the computational complexity of PIR protocols. In this section, we show that these computational lower-bounds give rise to communication lower-bounds for OT extension (i.e., the number of bits that an extended OT sender needs to communicate). The result of this section implies that the communication complexity of the sender in the IKNP OT extension protocol [IKNP03] is close to optimal.

Corollary 7 (OT Extension: Sender Communication Lower-Bound)**.** *For any constants $c < 1$, there exist no ℓ-batch k-bit OT extension protocol with sender communication $\eta < c2k\ell$, round complexity $r \in o(k\ell)$, and with the sender making $q \in o(k\ell)$ OT calls.*

Proof. OT extension is just an OT protocol that makes use of only a black-box OT and a random oracle. An ℓ-batch k-bit OT naturally gives rise to a $k\ell$-bit string OT. In Lemma 1 we show that such an OT protocol implies a non-trivial PIR for databases of $2\ell k$ bits. Under the resulting PIR protocol, the server communication is $\eta < c2k\ell$, round complexity $r \in o(k\ell)$, and the server communicates a total of $o(k\ell)$ bits with the OT oracle. Invoking Theorem 3, by instantiating SO with a generic OT oracle [GKM+00] and O with the random oracle [BR93] and by also including a PSPACE-complete oracle, we get a non-trivial PIR with oracle access to the random oracle and a PSPACE-complete oracle. This in turn can be transformed into an OT protocol (while retaining the O oracles) via [DMO00]. The existence of such an object however was ruled out by [GKM+00]. □ □

Corollary 8 (OT Extension: Total Communication Lower-Bound)**.** *For any constants $c < 1$, there exist no ℓ-batch k-bit OT extension protocol with total communication $\eta < \frac{3}{2}c(\ell + k\ell)$, round complexity $r \in o(k\ell)$, and with the sender making $q \in o(k\ell)$ OT calls.*

Proof. Follows from Lemma 2 and Corollary 7. □ □

[4] This means the server communicates at most q bits to the FHE oracle

Acknowledgements

We thank the anonymous reviewers for their valuable comments and suggestions.

J.D.: Funded by the European Union (ERC, LACONIC, 101041207). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them.

M. Hajiabadi supported in part by an NSERC Discovery Grant 03270, and a Meta Research Award.

References

[ADD$^+$22] Divesh Aggarwal, Nico Döttling, Jesko Dujmovic, Mohammad Hajiabadi, Giulio Malavolta, and Maciej Obremski. Algebraic restriction codes and their applications. In Mark Braverman, editor, *13th Innovations in Theoretical Computer Science Conference, ITCS 2022, January 31 - February 3, 2022, Berkeley, CA, USA*, volume 215 of *LIPIcs*, pages 2:1–2:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. 2

[AS15] Gilad Asharov and Gil Segev. Limits on the power of indistinguishability obfuscation and functional encryption. In Venkatesan Guruswami, editor, *56th Annual Symposium on Foundations of Computer Science*, pages 191–209. IEEE Computer Society Press, October 2015. 14

[BBDP22] Zvika Brakerski, Pedro Branco, Nico Döttling, and Sihang Pu. Batch-OT with optimal rate. In Orr Dunkelman and Stefan Dziembowski, editors, *Advances in Cryptology – EUROCRYPT 2022, Part II*, volume 13276 of *Lecture Notes in Computer Science*, pages 157–186. Springer, Heidelberg, May / June 2022. 2

[Bea96] Donald Beaver. Correlated pseudorandomness and the complexity of private computations. In *28th Annual ACM Symposium on Theory of Computing*, pages 479–488. ACM Press, May 1996. 1, 2, 7

[BIM00] Amos Beimel, Yuval Ishai, and Tal Malkin. Reducing the servers computation in private information retrieval: PIR with preprocessing. In Mihir Bellare, editor, *Advances in Cryptology – CRYPTO 2000*, volume 1880 of *Lecture Notes in Computer Science*, pages 55–73. Springer, Heidelberg, August 2000. 2

[BIM04] Amos Beimel, Yuval Ishai, and Tal Malkin. Reducing the servers' computation in private information retrieval: PIR with preprocessing. *Journal of Cryptology*, 17(2):125–151, March 2004. 6

[BIPW17] Elette Boyle, Yuval Ishai, Rafael Pass, and Mary Wootters. Can we access a database both locally and privately? In Yael Kalai and Leonid Reyzin, editors, *TCC 2017: 15th Theory of Cryptography Conference, Part II*, volume 10678 of *Lecture Notes in Computer Science*, pages 662–693. Springer, Heidelberg, November 2017. 6

[BKM20] Zvika Brakerski, Venkata Koppula, and Tamer Mour. NIZK from LPN and trapdoor hash via correlation intractability for approximable relations. In Daniele Micciancio and Thomas Ristenpart, editors, *Advances in Cryptology – CRYPTO 2020, Part III*, volume 12172 of *Lecture Notes in Computer Science*, pages 738–767. Springer, Heidelberg, August 2020. 7

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In Dorothy E. Denning, Raymond Pyle, Ravi Ganesan, Ravi S. Sandhu, and Victoria Ashby, editors, *ACM CCS 93: 1st Conference on Computer and Communications Security*, pages 62–73. ACM Press, November 1993. 6, 15

[CGH+21] Melissa Chase, Sanjam Garg, Mohammad Hajiabadi, Jialin Li, and Peihan Miao. Amortizing rate-1 OT and applications to PIR and PSI. In Kobbi Nissim and Brent Waters, editors, *TCC 2021: 19th Theory of Cryptography Conference, Part III*, volume 13044 of *Lecture Notes in Computer Science*, pages 126–156. Springer, Heidelberg, November 2021. 2, 6, 7

[CGKS95] Benny Chor, Oded Goldreich, Eyal Kushilevitz, and Madhu Sudan. Private information retrieval. In *36th Annual Symposium on Foundations of Computer Science*, pages 41–50. IEEE Computer Society Press, October 1995. 2, 6

[CGN98] Benny Chor, Niv Gilboa, and Moni Naor. Private information retrieval by keywords. Cryptology ePrint Archive, Report 1998/003, 1998. https://eprint.iacr.org/1998/003. 7

[CHK22] Henry Corrigan-Gibbs, Alexandra Henzinger, and Dmitry Kogan. Single-server private information retrieval with sublinear amortized time. In Orr Dunkelman and Stefan Dziembowski, editors, *Advances in Cryptology – EUROCRYPT 2022, Part II*, volume 13276 of *Lecture Notes in Computer Science*, pages 3–33. Springer, Heidelberg, May / June 2022. 6

[CHR17] Ran Canetti, Justin Holmgren, and Silas Richelson. Towards doubly efficient private information retrieval. In Yael Kalai and Leonid Reyzin, editors, *TCC 2017: 15th Theory of Cryptography Conference, Part II*, volume 10678 of *Lecture Notes in Computer Science*, pages 694–726. Springer, Heidelberg, November 2017. 6

[CK20] Henry Corrigan-Gibbs and Dmitry Kogan. Private information retrieval with sublinear online time. In Anne Canteaut and Yuval Ishai, editors, *Advances in Cryptology – EUROCRYPT 2020, Part I*, volume 12105 of *Lecture Notes in Computer Science*, pages 44–75. Springer, Heidelberg, May 2020. 4, 6

[CM20] Melissa Chase and Peihan Miao. Private set intersection in the internet setting from lightweight oblivious PRF. In Daniele Micciancio and Thomas Ristenpart, editors, *Advances in Cryptology – CRYPTO 2020, Part III*, volume 12172 of *Lecture Notes in Computer Science*, pages 34–63. Springer, Heidelberg, August 2020. 7

[CMS99] Christian Cachin, Silvio Micali, and Markus Stadler. Computationally private information retrieval with polylogarithmic communication. In Jacques Stern, editor, *Advances in Cryptology – EUROCRYPT'99*, volume 1592 of *Lecture Notes in Computer Science*, pages 402–414. Springer, Heidelberg, May 1999. 2, 6

[DGI+19] Nico Döttling, Sanjam Garg, Yuval Ishai, Giulio Malavolta, Tamer Mour, and Rafail Ostrovsky. Trapdoor hash functions and their applications. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology – CRYPTO 2019, Part III*, volume 11694 of *Lecture Notes in Computer Science*, pages 3–32. Springer, Heidelberg, August 2019. 2, 3, 6, 7, 15

[DMO00] Giovanni Di Crescenzo, Tal Malkin, and Rafail Ostrovsky. Single database private information retrieval implies oblivious transfer. In Bart Preneel, editor, *Advances in Cryptology – EUROCRYPT 2000*, volume 1807 of *Lecture Notes in Computer Science*, pages 122–138. Springer, Heidelberg, May 2000. 2, 6, 9, 14, 15

[GGK03] Rosario Gennaro, Yael Gertner, and Jonathan Katz. Lower bounds on the efficiency of encryption and digital signature schemes. In *35th Annual ACM Symposium on Theory of Computing*, pages 417–425. ACM Press, June 2003. 6

[GHMM18] Sanjam Garg, Mohammad Hajiabadi, Mohammad Mahmoody, and Ameer Mohammed. Limits on the power of garbling techniques for public-key encryption. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology – CRYPTO 2018, Part III*, volume 10993 of *Lecture Notes in Computer Science*, pages 335–364. Springer, Heidelberg, August 2018. 14

[GHO20] Sanjam Garg, Mohammad Hajiabadi, and Rafail Ostrovsky. Efficient range-trapdoor functions and applications: Rate-1 OT and more. In Rafael Pass and Krzysztof Pietrzak, editors, *TCC 2020: 18th Theory of Cryptography Conference, Part I*, volume 12550 of *Lecture Notes in Computer Science*, pages 88–116. Springer, Heidelberg, November 2020. 2, 6, 7

[GKM$^+$00] Yael Gertner, Sampath Kannan, Tal Malkin, Omer Reingold, and Mahesh Viswanathan. The relationship between public key encryption and oblivious transfer. In *41st Annual Symposium on Foundations of Computer Science*, pages 325–335. IEEE Computer Society Press, November 2000. 4, 14, 15

[GMM17] Sanjam Garg, Mohammad Mahmoody, and Ameer Mohammed. Lower bounds on obfuscation from all-or-nothing encryption primitives. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology – CRYPTO 2017, Part I*, volume 10401 of *Lecture Notes in Computer Science*, pages 661–695. Springer, Heidelberg, August 2017. 15

[GMMM18] Sanjam Garg, Mohammad Mahmoody, Daniel Masny, and Izaak Meckler. On the round complexity of OT extension. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology – CRYPTO 2018, Part III*, volume 10993 of *Lecture Notes in Computer Science*, pages 545–574. Springer, Heidelberg, August 2018. 7

[GT00] Rosario Gennaro and Luca Trevisan. Lower bounds on the efficiency of generic cryptographic constructions. In *41st Annual Symposium on Foundations of Computer Science*, pages 305–313. IEEE Computer Society Press, November 2000. 6

[HHC$^+$22] Alexandra Henzinger, Matthew M. Hong, Henry Corrigan-Gibbs, Sarah Meiklejohn, and Vinod Vaikuntanathan. One server for the price of two: Simple and fast single-server private information retrieval. Cryptology ePrint Archive, Report 2022/949, 2022. https://eprint.iacr.org/2022/949. 2

[HHC$^+$23a] Alexandra Henzinger, Matthew M. Hong, Henry Corrigan-Gibbs, Sarah Meiklejohn, and Vinod Vaikuntanathan. One server for the price of two: Simple and fast single-server private information retrieval. In Joseph A. Calandrino and Carmela Troncoso, editors, *32nd USENIX Security Symposium, USENIX Security 2023, Anaheim, CA, USA, August 9-11, 2023*. USENIX Association, 2023. 6

[HHC$^+$23b] Alexandra Henzinger, Matthew M. Hong, Henry Corrigan-Gibbs, Sarah Meiklejohn, and Vinod Vaikuntanathan. One server for the price of two: Simple and fast single-server private information retrieval. In Joseph A. Calandrino and Carmela Troncoso, editors, *32nd USENIX Security Symposium, USENIX Security 2023, Anaheim, CA, USA, August 9-11, 2023*. USENIX Association, 2023. 6

[IKNP03] Yuval Ishai, Joe Kilian, Kobbi Nissim, and Erez Petrank. Extending oblivious transfers efficiently. In Dan Boneh, editor, *Advances in Cryptology – CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 145–161. Springer, Heidelberg, August 2003. 1, 2, 3, 5, 7, 10, 15

[IP07] Yuval Ishai and Anat Paskin. Evaluating branching programs on encrypted data. In Salil P. Vadhan, editor, *TCC 2007: 4th Theory of Cryptography Conference*, volume 4392 of *Lecture Notes in Computer Science*, pages 575–594. Springer, Heidelberg, February 2007. 2, 6, 7

[JLLW23] Aayush Jain, Huijia Lin, Ji Luo, and Daniel Wichs. The pseudorandom oracle model and ideal obfuscation. In Helena Handschuh and Anna Lysyanskaya, editors, *Advances in Cryptology - CRYPTO 2023 - 43rd Annual International Cryptology Conference, CRYPTO 2023, Santa Barbara, CA, USA, August 20-24, 2023, Proceedings, Part IV*, volume 14084 of *Lecture Notes in Computer Science*, pages 233–262. Springer, 2023. 4, 14

[KC21] Dmitry Kogan and Henry Corrigan-Gibbs. Private blocklist lookups with checklist. In Michael Bailey and Rachel Greenstadt, editors, *USENIX Security 2021: 30th USENIX Security Symposium*, pages 875–892. USENIX Association, August 2021. 6

[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*, pages 364–373. IEEE Computer Society Press, October 1997. 2

[KO00] Eyal Kushilevitz and Rafail Ostrovsky. One-way trapdoor permutations are sufficient for non-trivial single-server private information retrieval. In Bart Preneel, editor, *Advances in Cryptology – EUROCRYPT 2000*, volume 1807 of *Lecture Notes in Computer Science*, pages 104–121. Springer, Heidelberg, May 2000. 6

[LMW23a] Wei-Kai Lin, Ethan Mook, and Daniel Wichs. Doubly efficient private information retrieval and fully homomorphic RAM computation from ring LWE. In Barna Saha and Rocco A. Servedio, editors, *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023, Orlando, FL, USA, June 20-23, 2023*, pages 595–608. ACM, 2023. 4

[LMW23b] Wei-Kai Lin, Ethan Mook, and Daniel Wichs. Doubly efficient private information retrieval and fully homomorphic RAM computation from ring LWE. In Barna Saha and Rocco A. Servedio, editors, *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023, Orlando, FL, USA, June 20-23, 2023*, pages 595–608. ACM, 2023. 6

[OS07] Rafail Ostrovsky and William E. Skeith III. A survey of single-database private information retrieval: Techniques and applications (invited talk). In Tatsuaki Okamoto and Xiaoyun Wang, editors, *PKC 2007: 10th International Conference on Theory and Practice of Public Key Cryptography*, volume 4450 of *Lecture Notes in Computer Science*, pages 393–411. Springer, Heidelberg, April 2007. 3

[PY22] Giuseppe Persiano and Kevin Yeo. Limits of preprocessing for single-server PIR. In Joseph (Seffi) Naor and Niv Buchbinder, editors, *Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms, SODA 2022, Virtual Conference / Alexandria, VA, USA, January 9 - 12, 2022*, pages 2522–2548. SIAM, 2022. 6

[SACM21] Elaine Shi, Waqar Aqeel, Balakrishnan Chandrasekaran, and Bruce M. Maggs. Puncturable pseudorandom sets and private information retrieval with near-optimal online bandwidth and time. In Tal Malkin and Chris Peikert, editors, *Advances in Cryptology – CRYPTO 2021, Part IV*, volume 12828 of *Lecture Notes in Computer Science*, pages 641–669, Virtual Event, August 2021. Springer, Heidelberg. 6

[Sho97] Victor Shoup. Lower bounds for discrete logarithms and related problems. In Walter Fumy, editor, *Advances in Cryptology – EUROCRYPT'97*, volume 1233 of *Lecture Notes in Computer Science*, pages 256–266. Springer, Heidelberg, May 1997. 15

[WW06] Stefan Wolf and Jürg Wullschleger. Oblivious transfer is symmetric. In Serge Vaudenay, editor, *Advances in Cryptology – EUROCRYPT 2006*, volume 4004 of *Lecture Notes in Computer Science*, pages 222–232. Springer, Heidelberg, May / June 2006. 5, 10, 11

[Yeo23] Kevin Yeo. Lower bounds for (batch) PIR with private preprocessing. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology – EUROCRYPT 2023, Part I*, volume 14004 of *Lecture Notes in Computer Science*, pages 518–550. Springer, Heidelberg, April 2023. 6

[Zha22] Mark Zhandry. To label, or not to label (in generic groups). In Yevgeniy Dodis and Thomas Shrimpton, editors, *Advances in Cryptology – CRYPTO 2022, Part III*, volume 13509 of *Lecture Notes in Computer Science*, pages 66–96. Springer, Heidelberg, August 2022. 4

[ZLTS23] Mingxun Zhou, Wei-Kai Lin, Yiannis Tselekounis, and Elaine Shi. Optimal single-server private information retrieval. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology – EUROCRYPT 2023, Part I*, volume 14004 of *Lecture Notes in Computer Science*, pages 395–425. Springer, Heidelberg, April 2023. 6

[ZPSZ23] Mingxun Zhou, Andrew Park, Elaine Shi, and Wenting Zheng. Piano: Extremely simple, single-server PIR with sublinear server computation. *IACR Cryptol. ePrint Arch.*, page 452, 2023. 6