The Feasibility of Outsourced Database Search in the Plain Model

Carmit Hazay* Hila Zarosim†

Abstract

The problem of securely outsourcing computation to an *untrusted* server gained momentum with the recent penetration of *cloud computing* services. The ultimate goal in this setting is to design efficient protocols that minimize the computational overhead of the clients and instead rely on the extended resources of the server. In this paper, we focus on the *outsourced database search* problem which is highly motivated in the context of delegatable computing since it offers storage alternatives for massive databases, that may contain confidential data. This functionality is described in two phases: (1) setup phase and (2) query phase. The main goal is to minimize the parties workload in the query phase so that it is proportional to the query size and its corresponding response.

We study whether a trusted setup or a random oracle are *necessary* for protocols with *minimal interaction* that meet the optimal communication and computation bounds in the query phase. We answer this question positively and demonstrate a lower bound on the communication or the computational overhead in this phase.

Keywords: Outsourced Computation, Database Search Functionalities, Lower Bound, Communication and Computational Complexities, Minimal Interaction

*Faculty of Engineering, Bar-Ilan University, Israel. Email: carmit.hazay@biu.ac.il. Research partially supported by a grant from the Israel Ministry of Science and Technology (grant No. 3-10883), by the European Research Council under the ERC consolidators grant agreement n. 615172 (HIPS).

†Department of Computer Science, Bar-Ilan University, Israel. Email: zarosih@cs.biu.ac.il. The author is grateful to the Azrieli Foundation for the Azrieli Fellowship award.

1 Introduction

Background on outsourced secure computation. The problem of securely outsourcing computation to an *untrusted* server gained momentum with the recent penetration of *cloud computing* services, where clients can lease computing services on demand rather than maintaining their own infrastructure. The ultimate goal in this setting is to design efficient protocols that minimize the computational overhead of the clients and instead rely on the extended resources of the server. Of course, the amount of work invested by the client in order to verify the correctness of the computation needs to be *substantially* smaller than running the computation by itself. Another ambitious challenge of delegatable computation is to design protocols that minimize the *communication* between the cloud and the client. This becomes of particular importance with the proliferation of smartphone technology and mobile broadband internet connections, as for mobile devices communication and data connectivity is often the more severe bottleneck.

The study of delagatable computation was initiated with the study of a restricted scenario where a *single client* outsources its computation to an external server. Two main approaches are examined in this context. In the first setting there is only one phase of interaction between the client and the server such that the overall amount of work performed by the client is smaller than performing the computation on its own. Correctness in this setting is achieved by succinct zero-knowledge proofs [GLR11, BCCT12, DFH12] with the aim of minimizing the number of rounds between the client and the server. In the *amortized setting* [GGP10, AIK10] the computational complexity of the client is analyzed in an amortized sense. Namely, the client can perform some expensive preprocessing (also known as the offline phase). After this phase is completed, it is required to run very efficient computations in the online phase.

Recent results also study an extended setting with *multiple r clients* that mutually distrust each other and wish to securely outsource a joint computation on their inputs with reduced costs [KMR11, KMR12, LATV12, AJLA$^+$12, CKKC13]. In particular, it is required that the communication between the clients and the server, as well as the communication between the clients, will be sufficiently smaller than running a secure computation in the standard setting. This more complex setting is strictly harder than the single client setting since one must handle potential corruptions of any (proper) subset of the clients, that might collude with the server. It is worth noting that in case only correctness is required then security in the multi clients setting is reduced to security in the single client setting. This is due to the fact that we can consider a protocol where $r-1$ clients send their inputs to the rth client, that communicates with the server using all inputs. It then forwards the server's proof to the other clients who can verify its correctness. Generally speaking, outsourced secure computation in the presence of collusion between any t clients and the server implies secure computation in the standard setting with $r-t+1$ parties. Thus, the problem of delegatable computation with multiple clients focuses on achieving privacy (with or without imposing correctness).

Modeling outsourced database search. We consider an outsourced database search functionality where one client has a database, and another set of clients search the database using a sequence of queries. To simplify the presentation we denote the data owner by the sender and the other set of clients by the receiver (for simplicity, we focus on a single receiver asking for multiple queries). The input of the sender is a database of size n.[1] The input queries of the receiver $\{q_i\}_{i\in[t]}$ are picked from a predefined set Q_n where Q_n is a set of queries that correspond to a database of size n. This functionality can be described in two phases. In the *setup phase* the sender uploads a function of its database to the server. This phase is run only once, where the sender's state after this phase is independent of n. Next, in the *query phase* the receiver picks a search query and obtains from the server the answer to this query. To restrict the number of queries, the sender must approve each query by providing a trapdoor that depends on the content of the query.

[1] We remark that the internal structure of the database is not important for our proofs.

This functionality is highly motivated in the context of outsourced computation since it offers storage alternatives for massive databases that may contain confidential data (e.g., health related data about patient history). Our formalization captures a large class of search functionalities such as oblivious transfer (OT) with adaptive queries [NP99, GH11], keyword search [FIPR05], pattern matching [HT10] and the indexing problem [Goh03]. The former two functionalities are part of a class for which a query's response size is bounded by an a priori fixed length, whereas for the latter two functionalities a response is unbounded and might be $O(n)$. Consequently, secure implementations of such functionalities are more involved. Moreover, our infeasibility results are more meaningful for this class.

Security is formalized via the standard ideal/real paradigm where in the ideal setting the three parties: sender, receiver and server, communicate with an ideal functionality that first obtains the preprocessed database from the sender and later answers search queries made by the receiver, while leaking some information about the responses to the server.[2] Our modeling also captures collusion between the server and the receiver. In order to take some advantage from this modeling, we would like the setup phase to imply $O(n)$ workload, yet the overall cost of issuing a query should only *grow linearly with the size of the query's response* (which is as optimal as one can obtain). For functionalities that do not imply a fixed bound on the query responses, this optimization comes at the price of revealing some leakage about the database (for instance, in pattern matching the server learns the number of matches of some hidden query).

Another important complexity measure of secure computation that has been extensively studied in literature, is the *round-complexity* of secure protocols. In the *stand-alone* setting, Katz and Ostrovsky [KO04] determined that the exact round complexity of achieving a secure two-party computation protocol is five (and four if only one of the parties receives an output). More recently, Ostrovsky, Richelson and Scafuro [ORS15] strengthened this construction by demonstrating a five-round protocol where the underlying cryptographic primitives are used only in a "black-box" way. Both the results also provide a four-round protocol for single-output functionalities. In the multi-party setting, the recent work by Garg et al. [GMPP16] studies the exact round-complexity of multi-party computation (MPC) protocols in the plain model and shows that at least four rounds are necessary for realizing general functionalities.

In this work we study the feasibility of protocols with *minimal interaction* in the outsourced setting with no trusted setup, where in the setup phase the sender sends a single message to the server, whereas in the query phase the sender and the receiver exchange only two messages (one in each direction), and then one message in each direction between the receiver and the server.[3] Specifically, we focus on semi-honest security and study the *feasibility* of the outsourced database search functionality in the plain model with *minimal interaction* and using *minimal resources* of communication and computation. Security in this model implies sender's privacy against, potentially colluding, server and receiver. Whereas receiver's privacy is ensured against either corrupted sender or server. We address the following question,

Does there exist a private protocol with minimal interaction for the outsourced search functionality in the plain model, that meets the optimal communication/computation bounds in the query phase?

We prove that the answer for this question is negative and that there exists a large class of search functionalities that *cannot* be realized privately with optimal resources in the query phase.

[2] Our formalization considers the minimal leakage of the length of the queries responses, yet our proofs follow for any type of leakage as the preprocessed database is computed independently of that leakage.

[3] We prove that if the order of communication between the receiver and the sender/server is swapped then our lower bounds follow more easily. We further note that our lower bounds are not restricted to a minimal interaction between the server and the receiver.

1.1 Our Results

We prove that using a trusted setup or a random oracle is essential in order to reduce the resources of the receiver within protocols with minimal interaction, *even* if the sender's state in $o(n)$ and the number of rounds between the server and the receiver is arbitrary. This result has the consequence that for certain search functionalities (e.g, pattern matching and all its variants, and the indexing problem), either the communication complexity or the running time of the receiver must be as large as the size of the database. In this paper we examine both non-private and private channels scenarios (where in the latter setting corrupted parties do not see the communication between the honest parties), and prove that our lower bound holds in both settings, where our proof in the non-private setting relies on a weaker adversary.

More formally, let $\mathcal{ANS}_{n,q}$ denote the set of *all potential responses* for the query q when ranging over all databases T of size n, and let $H_{n,Q} = \max_{q \in Q_n} \log |\mathcal{ANS}_{n,q}|$ (intuitively, $H_{n,Q}$ is the logarithm of the number of potential query responses when ranging over all databases of size n and all queries in Q_n; see Definition 3.2). Then we prove the following theorem,

Theorem 1.1 *(informal) For any protocol with minimal interaction that securely implements the outsourced database search functionality in the presence of semi-honest adversaries, one of the following holds:*

1. *The communication complexity in the query phase is $\Omega(H_{n,Q})$.*
2. *The number of random bits used by the receiver is $\Omega(H_{n,Q})$.*

Our proof follows a similar intuition of the proof from [Nie02] when showing the impossibility of constructing non-interactive non-committing encryption schemes in the plain model. Nevertheless, the formalization is more challenging since the number of involved parties is three. One consequence that we need to take into account is the order of rounds of which the receiver interacts with the other parties. This is because for our proof in the private channels setting we need to consider an adversary that corrupts both the server and the receiver. In this case, the view of the adversary contains both the randomness of the receiver and the server, as well as the messages sent from the sender. We further must distinguish between the randomness of the receiver and that of the server and rely on the fact that the random tape of the server is uniformly independent of the receiver's view. Specifically, we need to show that when we fix a partial view of the receiver, then for almost all random tapes of the server, the receiver outputs the correct value. Note that if the receiver communicates with the server first then this independence no longer holds since the communication between the receiver and the sender may depend on the random tape of the server (as it may depend on the messages from the server). On the other hand, if the receiver communicates with the sender first then independence follows, as semi-honest adversaries cannot pick their randomness arbitrarily. This subtlety is in contrast to the proof in [Nie02] that relies on the correctness of the non-interactive decryption algorithm of the underlying encryption scheme.

We consider the two potential orders of rounds in the query phase. If the receiver *communicates with the server first*, we show that the communication complexity of the protocol must be large. This is intuitively because at the time the receiver communicates with the server, the server does not know which information to send back and essentially must send as much information as the maximal amount of information sent within any response to query q (when ranging over all databases of size n). On the other hand, if the receiver *communicates with the sender first* then recall that in the simulation of the setup phase the simulator must commit to a setup message independently of the sender's database, where this message is fixed and cannot be later changed. Then in the query phase, the simulator has to simulate the view for the corrupted parties so that it yields the correct output for the receiver. We show that this means that for every possible answer there must exist a view $(r_{\mathtt{Rec}}, \mathsf{m}_2)$ for the receiver (where $r_{\mathtt{Rec}}$ is the random tape of the receiver and m_2 the message from the sender to the receiver) such that with a high probability (over the random coins of the

server), the receiver outputs the correct query response. This implies that the number of views $(r_{\mathtt{Rec}}, \mathsf{m}_2)$ must be proportional to the number of potential query responses when ranging over all databases (and hence the length of $(r_{\mathtt{Rec}}, \mathsf{m}_2)$ is linear in $H_{n,Q}$), which can be as large as the size of the database for certain search functionalities *even when the receiver's output size is small*.

It is important to note that our lower bounds hold for *any* protocol with minimal interaction. Therefore, we can always focus on a protocol that makes use of a minimal number of random coins. Saying differently, our lower bounds consider the *effective* number of bits used by the receiver and even cover scenarios where the receiver's random tape is very large, for which the receiver ignores some portion of it. The reason for this is that for every such protocol, we can consider an equivalent protocol where the receiver's random tape does not contain any unused bits and apply our lower bounds to the new protocol. This further implies a lower bound on the running time of the receiver since these random bits must be incorporated in the computation of the receiver. Moreover, our lower bounds hold *even if the receiver maintains no privacy* since they follow from the non-committing property that we require in the simulation.[4] Importantly, any attempt to replace the uniform random bits of the receiver by an output of a pseudorandom generator (PRG) in order to strengthen our lower bounds fails since it requires finding a preimage relative to the PRG; see more details in Section 4.3.

Finally, we note that our lower bounds also apply in the two-party setting for reactive search functionalities (with a preprocessing phase), which implies the infeasibility of private reactive pattern matching with optimal query response and minimal interaction in the plain model. This is in contrast to the non-private setting, where suffix trees [Wei73] (a data structure that solves pattern matching and related problems on unencrypted data), are useful to store the text in a way that allows fast string operations. In particular, it illustrates that private pattern matching cannot be optimized in the preprocessing setting.

1.2 Prior Work

In [FHV13], Faust et al. use novel ideas to solve pattern matching in the cloud based on a reduction to the subset sum problem, which do not rely on the hardness of the problem but rather require instances that are solvable in polynomial-time. This paper presents the first concrete protocols for this problem where the receiver wishes to learn the positions at which a pattern of length m matches the text (and nothing beyond that). Their constructions offer simulation-based security in the presence of semi-honest and malicious adversaries and limit the communication in the query phase to $O(m)$ bits plus the number of occurrences (where the semi-honest protocol is with minimal interaction). Nevertheless, Faust et al. rely heavily on the programmability property of the random oracle, and use it to equivocate a fake text. In [CS14], Chase and Shen solve outsourced pattern matching by constructing a so called queryable encryption, which is an encryption scheme that supports search queries between a client and a server with three rounds of communication. Their construction is based on suffix trees.

In [CK10] Chase and Kamara informally discuss (without providing a proof) a lower bound on the token length for structured encryption scheme, that encrypts a structured data in a way that allows to privately query the data. Their intuition says that the length of the token for a given query grows with the number of potential answers when working with a simulation-based definition. Our proofs formalize this intuition for settings with multiple clients for which the data owner is a different entity than the receiver.

Another related line of works regarding symmetric searchable encryption (SSE) allows a (single) client to store data on an untrusted server and later perform keyword searches. This primitive has been widely studied recently; see [CGKO11, KPR12, KP13, JJK$^+$13, CGPR15, ANSS16] for just few examples. The standard security definition for SSE schemes follows the ideal/real simulation paradigm and comes with

[4] We note that when privacy is not considered, we prove that there exists a query for which our lower bounds hold. For private protocols this implies that these lower bounds hold for all queries or else some information about the query leaks.

two flavours of static and adaptive searches (where in the latter modeling a keyword may be determined as a function of the previous tokens/responses). We note that our results also hold for non-interactive SSE for which the tokens maintain the keyword privacy, and thus can be transferred via a 2PC protocol to a different client than the data owner (denoted by the receiver in our paper). This scenario is considered in [JJK$^+$13] yet their security definition is weaker in the sense that the receiver cannot collude with the server.

2 Preliminaries

Basic notations. We denote the security parameter by κ and by $\mathcal{U}_n$ the uniform distribution over strings of length n. 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 it holds that $\mu(\kappa) < \frac{1}{p(\kappa)}$. We use the abbreviation PPT to denote probabilistic polynomial-time and the notation $[n]$ to denote the set of integers $\{1, \ldots, n\}$.

We specify the definition of computational indistinguishability.

Definition 2.1 (Computational indistinguishability by circuits) *Let $X = \{X(a,\kappa)\}_{a\in\{0,1\}^*,\kappa\in\mathbb{N}}$ and $Y = \{Y(a,\kappa)\}_{a\in\{0,1\}^*,\kappa\in\mathbb{N}}$ be two distribution ensembles. We say that X and Y are* computationally indistinguishable*, denoted $X \overset{\text{c}}{\approx} Y$, if for every PPT machine D, every $a \in \{0,1\}^*$, every positive polynomial $p(\cdot)$ and all sufficiently large κ:*

$$\left|\Pr\left[D(X(a,\kappa),1^\kappa)=1\right] - \Pr\left[D(Y(a,\kappa),1^\kappa)=1\right]\right| < \frac{1}{p(\kappa)}.$$

3 Our Modeling

In this section we model the reactive database search functionality where one client has a database, and another set of clients search the database using a sequence of queries. To simplify the presentation we denote the former client by the sender and the other set of clients by the receiver. (For simplicity, we focus on a single receiver asking for multiple queries).

Inputs and outputs. The input of the sender is a database T of size n bits. The input queries of the receiver $\{q_i\}_{i\in[t]}$ are picked from a predefined set Q_n of binary strings, where Q_n is a set of queries that correspond to a database of size n. Specifically, we let the set of queries $\{Q_n\}_{n\in\mathbb{N}}$ depend on the database size. This formalization captures search functionalities where Q_n changes with the database size, such as in oblivious transfer with adaptive queries. It further covers search functionalities where the same set of queries is used for databases of different sizes by fixing the same set of queries for all n, such as in pattern matching, (see Section 3.2 for the formal definitions of these functionalities).

The queries made by the receiver are determined adaptively by a PPT algorithm M that takes the receiver's initial input and the outputs of prior search results. Whenever we say that the honest receiver picks a search query $q_i \in Q_n$, we assume that the receiver applies its input selection algorithm M as specified above. Queries that do not have a suitable answer in the database will be responded with a *"no match"* message whenever queried by the receiver. Finally, we assume that $|q| \leq m$ for all $q \in Q_n$ and some fixed parameter $m = m(\kappa)$. We further assume that n is polynomial in the security parameter κ.

We let T_q denote the response of the functionality on database T and query $q \in Q_n$. As above, this formalization is general enough and allows to capture different search functionalities with different output structure (for instance, when the query outcome contains a single vs. a set of records).

Functionality $\mathcal{F}_{\text{ODBS}}$

Let $m, t \in \mathbb{N}$ and $Q = \{Q_n\}_n$. Functionality $\mathcal{F}_{\text{ODBS}}$ sets a table $\mathcal{B}$ initially to be empty and proceeds as follows, running with sender Sen, receiver Rec, server Ser and ideal adversary $\mathcal{SIM}$.

1. Upon receiving a message (DB, T, m) from Sen, send $(\mathsf{preprocess}, |T|, m)$ to Ser and $\mathcal{SIM}$, and record (DB, T) and $n = |T|$.
2. Upon receiving a message (query, q_i) from Rec (for $i \in [t]$), where message $(\mathsf{DB}, \cdot)$ has been recorded, $|q_i| \leq m$ and $q_i \in Q_n$, check if the table $\mathcal{B}$ already contains an entry of the form $(q_i, \cdot)$. If not, then pick the next available identifier id from $\{0,1\}^*$ and add (q_i, id) to $\mathcal{B}$. Send $(\mathsf{query}, \mathtt{Rec})$ to Sen and $\mathcal{SIM}$.
   (a) Upon receiving $(\mathsf{approve}, \mathtt{Rec})$ from Sen send $(\mathsf{response}, \mathtt{Rec}, |T_{q_i}|, \mathsf{id})$ to server Ser. Otherwise, if no $(\mathsf{approve}, \mathtt{Rec})$ message has been received from Sen, send $\perp$ to Rec and abort.
   (b) Send $(\mathsf{response}, q_i, T_{q_i}, \mathsf{id})$ to Rec.

Figure 1: The outsourced database search functionality

The reactive search functionality. The reactive search functionality is described in two phases.

1. In the *setup phase* the sender sends a message $a(T)$ to the server, where $a(\cdot)$ is some polynomial-time algorithm. This phase is run only once, such that the size of the sender's state s upon completion is bounded by $\mathsf{poly}(\kappa)$.[5]
2. In the *query phase* the receiver picks a search query and obtains from the server the answer to this query. Note that this definition is meaningful only if we restrict the number of queries made by the receiver. Otherwise, no notion of privacy is guaranteed for the sender, since the receiver (or even the server) can potentially search the database for as many queries as they wish. This requirement is formalized by asking the sender's "permission" whenever a query is made, and is an important feature of payment-based search applications where the receiver pays per search. Looking ahead, we implement this restriction using a secure protocol between the sender and the receiver that allows the receiver to learn the answer to its search query while maintaining the privacy of its query.

The formal definition of outsourced database search functionality appears in Figure 1.

Communication model. Our result are introduced in the plain model in two different settings: (1) in the private channels case where corrupted parties do not see the communication between the honest parties. (2) In the non-private channels case. In the later setting the adversary can observe the messages between the honest parties. We note that any infeasibility result in the private setting implies the same result in the non-private setting. Nevertheless, we reprove our theorem for the latter setting as well, assuming a weaker type of adversary. Concretely, our infeasibility result in the private setting requires a collusion between the server and receiver, whereas the analogue proof relies on an adversary that corrupts only the receiver.

Complexities. In order to take some advantage from this modeling, we would like the setup phase to require $O(n)$ workload, yet the overall cost of issuing a query should only *grow linearly with the size of the query's response* (which is as optimal as one can obtain). As mentioned before, for some search functionalities, where there is no fixed bound on the query's response, this optimization comes with the price

[5]For this to be meaningful, we requite that the size of the sender's state is strictly less than n. This is formalized by assuming the existence of two polynomials $p_1(\cdot)$ and $p_2(\cdot)$ such that $n \leq p_1(\kappa)$, $s \leq p_2(\kappa)$ and $s \in o(n)$.

of revealing some leakage about the database. We further allow leaking the search pattern, where the server recognizes whether the same query already asked before. Finally, we require that the round complexity of any protocol implemented in this setting is minimal. I.e., in the setup phase we require a single message sent from the sender to the server, whereas in the query phase we require the receiver exchange only two messages (one in each direction) with each of the other clients.

Security definition. Security is formalized using the ideal/real paradigm, considering the server as a separate entity that does not contribute any input to the computation. As in the standard two-party modeling a corrupted party is either semi-honest or malicious, where in the semi-honest setting the attacker follows the protocol's instructions but tries to gain additional information about the honest parties' inputs, whereas in the malicious setting the attacker follows an arbitrary efficient strategy. This modeling also captures collusion between some of the parties, when the adversary corrupts more than one party and the corrupted parties share a joint state. In this work we only consider collusion between the server and the receiver.[6] We say that a protocol is secure in the presence of (P_1/P_2)-*collusion* if security holds against collusion between parties P_1 and P_2 (in addition to individual corruptions).

Formally, denote by $\mathbf{IDEAL}_{\mathcal{F}_{\mathrm{ODBS}},\mathcal{SIM}(z)}(\kappa,(-,T,(q_1,\ldots,q_t)))$ the output of an ideal adversary $\mathcal{SIM}$, server $\mathtt{Ser}$, receiver $\mathtt{Rec}$ and sender $\mathtt{Sen}$ in the above ideal execution of $\mathcal{F}_{\mathrm{ODBS}}$ upon given the respective inputs $(-,T,(q_1,\ldots,q_t))$. Functionality $\mathcal{F}_{\mathrm{ODBS}}$ is implemented via a protocol $\pi=(\pi_{\mathsf{Pre}},\pi_{\mathsf{Query}})$ consisting of a pair of protocols specified as follows. A two-party protocol π_{Pre} that is carried out in the setup phase by $\mathtt{Sen}$, that preprocesses database T and forwards the outcome $a(T)$ to $\mathtt{Ser}$. During the query phase protocol π_{Query} is carried out between $\mathtt{Rec}$ (holding a query q) and $\mathtt{Sen}$, $\mathtt{Ser}$, where $\mathtt{Rec}$ communicates with each party separately. We denote by $\mathbf{REAL}_{\pi,\mathcal{A}(z)}(\kappa,(-,T,(q_1,\ldots,q_t)))$ the output of a non-uniform PPT adversary $\mathcal{A}$, server $\mathtt{Ser}$, sender $\mathtt{Sen}$ and receiver $\mathtt{Rec}$ in a real execution of $\pi=(\pi_{\mathsf{Pre}},\pi_{\mathsf{Query}})$ upon given the respective inputs $(-,T,(q_1,\ldots,q_t))$.

Definition 3.1 (Security of outsourced database search) *We say that* π securely implements $\mathcal{F}_{\mathrm{ODBS}}$ *with respect to queries* $Q=\{Q_n\}_{n\in\mathbb{N}}$ *in the presence of* $(\mathtt{Ser}/\mathtt{Rec})$*-collusion and semi-honest (respectively, malicious) adversaries, if for any PPT semi-honest (respectively, malicious) adversary* $\mathcal{A}$ *there exists a PPT semi-honest (respectively, malicious) simulator* $\mathcal{SIM}$ *such that for any tuple of inputs* $(T,(q_1,\ldots,q_t))$ *such that* $q_1,\ldots,q_t\in Q_{|T|}$*, and auxiliary input* z*,*

$$\{\mathbf{IDEAL}_{\mathcal{F}_{\mathrm{ODBS}},\mathcal{SIM}(z)}(\kappa,(-,T,(q_1,\ldots,q_t)))\}_{\kappa\in\mathbb{N}}\overset{\mathrm{c}}{\approx}\{\mathbf{REAL}_{\pi,\mathcal{A}(z)}(\kappa,(-,T,(q_1,\ldots,q_t)))\}_{\kappa\in\mathbb{N}}.$$

3.1 Useful Notations

Let n be a natural number denoting the size of the database and let $Q=\{Q_n\}_{n\in\mathbb{N}}$ be such that Q_n is a set of appropriate queries for databases of size n bits.[7] We introduce important notations next.

Definition 3.2 *For every* $q\in Q_n$*, we let* $\mathcal{ANS}_{n,q}$ *denote the set of all potential responses* T_q *for the query* q *when ranging over all databases* T *of size* n*. Formally,* $\mathcal{ANS}_{n,q}=\{T_q \mid T\in\{0,1\}^n\}$*. Furthermore, let* $H_{n,Q}=\max_{q\in Q_n}\log|\mathcal{ANS}_{n,q}|$*, which intuitively captures the maximal amount of information that a response for* any *query* $q\in Q_n$ *provides.*

For instance, consider the oblivious transfer with adaptive queries functionality where every entry in the database is of size ℓ. In this case, $\mathcal{ANS}_{n,q}$ is the set of all ℓ-length binary strings.

[6] Notably, our lower bounds also apply to settings where all type of collusion are allowed since this only strengthens the model.

[7] We emphasize that the infeasibility proof holds for any database of length n (regardless of its internal structure).

Definition 3.3 *We specify the following definitions:*

1. *Denote by* $\mathsf{cc}^{n,q}_{\mathtt{Ser}}(\kappa) = \mathsf{cc}_{\mathtt{Ser}}(\kappa, n, q)$ *the communication complexity of the interaction between* `Rec` *and* `Ser` *within* π_{Query} *such that the receiver's input is the query* q *and the database is of size* n*. Namely, the number of bits being transferred between the receiver and the server in the query phase with parameters* κ *and* q*.*

2. *Analogously, denote by* $\mathsf{cc}^{n,q}_{\mathtt{Sen}}(\kappa) = \mathsf{cc}_{\mathtt{Sen}}(\kappa, n, q)$ *the communication complexity of the interaction between the receiver and the sender within* π_{Query} *such that the receiver's input is the query* q *and the database is of size* n*.*

3. *Denote by* $\mathsf{cc}^{n,q}(\kappa) = \mathsf{cc}(\kappa, n, q)$ *the overall communication complexity within* π_{Query}*. Namely, the overall number of bits being transferred during the execution of* π_{Query} *such that the receiver's input is the query* q *and the database is of size* n*.*

4. *Finally, denote by* $\mathsf{rand}^{n,q}_{\mathtt{Rec}}(\kappa) = \mathsf{rand}_{\mathtt{Rec}}(\kappa, n, q)$ *the size of the receiver's random tape within* π_{Query} *such that the receiver's input is the query* q *and the database is of size* n*.*

3.2 Concrete Functionalities

We specify the description of two important functionalities in the context of database search.

3.2.1 Outsourced Oblivious Transfer with Adaptive Queries

The basic t-out-of-n oblivious transfer functionality between a sender and a receiver is denoted by $\mathcal{F}_{\mathrm{OT}} : ((x_1, \ldots, x_n), (q_1, \ldots, q_t)) \mapsto (-, (x_{q_1}, \ldots, x_{q_t}))$, where $x_i \in \{0,1\}^\ell$ for all $i \in [n]$, and $\ell = \ell(\kappa), n = n(\kappa)$ are polynomials in κ. Namely, the receiver learns t elements from the input vector of the sender while the sender learns nothing. Note that by definition the receiver decides on the elements it wishes to obtain in advance. Alternatively, we can modify the description of $\mathcal{F}_{\mathrm{OT}}$ so that the receiver picks its input adaptively. This functionality is denoted by oblivious transfer with *adaptive* queries, where the queries are indices from $[n]$ and the outcome is the record in the ith database entry. The outsourced variant of this problem is defined by having the sender uploading its database to an external server.

3.2.2 Outsourced Pattern Matching

The inputs for the basic pattern matching problem are a text T of length n and a pattern p (i.e., keyword) of length m; the goal is to find all the text locations in which the pattern matches the text. A private distributed variant of this problem is defined in the two-party setting, where party P_1 holds a text T and party P_2 holds a pattern p. The goal of P_2 is to learn the positions in which p matches the text without revealing anything about the pattern to P_1; at the same time, P_2 should not learn anything else about the text. The outsourced variant of the problem which is specified in two phases, following the notation of Faust et al. [FHV13]. In the setup phase the sender uploads a (preprocessed) text $a(T)$ to an external server `Ser`. In the query phase the receiver queries the text by searching patterns and learns the matched text locations. The reader can think of each record in the pattern matching database as a sequence of indices from $[n]$. In comparison with oblivious transfer, implementing this functionality is much more involved, since the database records are strongly related. This makes simulation (for the case the receiver is corrupted) much more challenging.

4 Infeasibility of Outsourced Database Search in the Plain Model

In this section we introduce our infeasibility result of outsourced database search in the plain model. We introduce our lower bound in two settings: (1) In Section 4.1 we prove the private channels case where corrupted parties do not see the communication between the honest parties. (2) In Section 4.2 we prove a similar theorem in the non-private channels case. In the later proof the adversary can observe the messages between the honest parties, which implies that a corrupted receiver observes the setup message. This simplifies our proof since the simulator does not need to generate the internal state of the server. The proof in the former setting holds only for protocols secure against $(\mathtt{Ser}/\mathtt{Rec})$-collusion and is slightly more involved.

4.1 The Private Channels Case

Our proof is shown in the presence of collusion between the receiver and the server and crucially relies on the assumption that the receiver communicates with the sender first. This ordering enables to split the randomness of an adversary controlling these parties into two distinct and independent sets. In Theorem 4.1 we show that this ordering in necessary, proving that if this order of rounds is modified then the communication complexity between the server and the receiver must be proportional to $H_{n,Q}$, that might be as large as the database size for some functionalities (see Lemma 4.6). Informally, this statement follows since at the time the receiver communicates with the server, the server does not know anything about the database. It therefore does not know the correct response to the receiver's query, and essentially must send as much information as the maximal amount of information sent within any response to query q (with respect to all possible databases of size n). Recall that we assume that the receiver communicates with each party only once. Formally,

Theorem 4.1 *Fix n and m, and let $\pi = (\pi_{\mathsf{Pre}}, \pi_{\mathsf{Query}})$ be a protocol with minimal interaction that securely implements $\mathcal{F}_{\mathrm{ODBS}}$ with respect to queries $Q = \{Q_n\}_n$ in the presence of $(\mathtt{Ser}/\mathtt{Rec})$-collusion and semi-honest adversaries. Then, if π_{Query} is defined such that* Rec *communicates with* Ser *first, for every n there exists $q \in Q_n$ such that it holds that $\mathtt{cc}^{n,q}_{\mathtt{Ser}}(\kappa) \geq H_{n,Q} - s$.*[8]

Proof: Fix n and assume by contradiction that $\mathtt{cc}^{n,q'}_{\mathtt{Ser}}(\kappa) < H_{n,Q} - s$ for every $q' \in Q_n$ and consider the case that only the server is corrupted. Note first whenever the receiver communicates with the server *first*, then for every query q the length of the server's response must be the same for all databases of size n. Otherwise the server can distinguish between two different databases of size n at the end of the setup phase (we recall that the server communicates with each party only once). More formally, assume that for some query q' and n, there exist two databases T_1 and T_2 of size n, such that the length of the server's response to the receiver is different for the following executions: (1) The input of the sender is T_1 and the input of the receiver is q'. (2) The input of the sender is T_2 and the input of the receiver is q'. Next, consider a corrupted server that obtains q' as part of its auxiliary input. Now, since the receiver communicates with the server first, the server can emulate this interaction by its own at the end of the setup phase and distinguish between the case where the sender's input is T_1 and the case where the sender's input is T_2 by observing the length of the emulated response to the receiver, which violates the sender's privacy. Note that this attack does not work in the case that the receiver talks to the sender first because the receiver's message to the server cannot be emulated by the server.

This implies that the *for every* fixed query $q \in Q_n$, the server must send the same number of bits for any database T. Specifically, this holds for the case that the receiver's input is the query q^*, where q^* is such that $\log |\mathcal{ANS}_{n,q^*}| = H_{n,Q}$. Now, since the number of potential answers for q^* is $|\mathcal{ANS}_{n,q^*}|$, the receiver must eventually learn $\log |\mathcal{ANS}_{n,q^*}| = H_{n,Q}$ bits. Nevertheless, since the sender can only send at most s

[8] Recall that s denotes the size of the sender's state in the query phase and that $s \in o(n)$.

bits to the receiver, we conclude that there exists a query q^* such that the server must send at least $H_{n,Q} - s$ bits to the receiver for all databases.

■

We stress that for every q, $|\mathcal{ANS}_{n,q}|$ is *independent* of the actual size of T_q for a concrete T, since it counts the number of potential responses when ranging over *all databases of length* n. Thus, the above lower bound is meaningful in the sense that it shows that the communication complexity might be large even if $|T_q|$ is small for some concrete T.

We are now ready to prove the following theorem.

Theorem 4.2 *Fix n and m, and let $\pi = (\pi_{\mathsf{Pre}}, \pi_{\mathsf{Query}})$ be a protocol with minimal interaction that securely implements $\mathcal{F}_{\mathrm{ODBS}}$ with respect to queries $Q = \{Q_n\}_n$ in the presence of $(\mathtt{Ser}/\mathtt{Rec})$-collusion and semi-honest adversaries in the private channels setting, such that $\mathtt{Rec}$ communicates with $\mathtt{Sen}$ first. Then one of the following holds:*

1. For every query $q \in Q_n$ *the communication complexity* $\mathsf{cc}^{n,q}_{\mathtt{Sen}}(\kappa) \geq \frac{H_{n,Q}-3}{2}$ *or*
2. There exists a query $q \in Q_n$ *such that* $\mathsf{rand}^{n,q}_{\mathtt{Rec}}(\kappa) \geq \frac{H_{n,Q}-3}{2}$.

Proof: Let $\pi = (\pi_{\mathsf{Pre}}, \pi_{\mathsf{Query}})$ be as in Theorem 4.2, let $\mathcal{A}_{\mathtt{Ser},\mathtt{Rec}}$ be a real-world semi-honest adversary controlling the server and the receiver, and let $\mathcal{SIM}_{\mathtt{Ser},\mathtt{Rec}}$ be an ideal-world adversary guaranteed to exist by the security of $\pi = (\pi_{\mathsf{Pre}}, \pi_{\mathsf{Query}})$. By definition, upon given a message $(\mathsf{preprocess}, |T|, m)$ in the setup phase $\mathcal{SIM}_{\mathtt{Ser},\mathtt{Rec}}$ outputs a string a_{Sim}. Moreover, upon given a message $(\mathsf{response}, q, T_q, \mathsf{id})$ in the query phase it outputs a valid view for $\mathcal{A}_{\mathtt{Ser},\mathtt{Rec}}$ (recall that T_q represents the correct output for query q with respect to database T). This view is a triple $(r_{\mathtt{Rec}}, \mathsf{m}_2, r_{\mathtt{Ser}})$, where $r_{\mathtt{Rec}}$ and $r_{\mathtt{Ser}}$ are the respective random tapes of $\mathtt{Rec}$ and $\mathtt{Ser}$ and m_2 is a simulated message from $\mathtt{Sen}$ to $\mathtt{Rec}$.

For a security parameter κ and a pair of query/response (q, T_q), let $\Pr_{\mathcal{SIM}_{\mathtt{Ser},\mathtt{Rec}},\kappa}[a_{\mathsf{Sim}}]$ denote the probability distribution over the simulated message of π_{Pre} and let $\Pr_{\mathcal{SIM}_{\mathtt{Ser},\mathtt{Rec}},\kappa,q,T_q}[a_{\mathsf{Sim}}, r_{\mathtt{Rec}}, \mathsf{m}_2, r^*_{\mathtt{Ser}}]$ denote the probability distribution on the values $(a_{\mathsf{Sim}}, r_{\mathtt{Rec}}, \mathsf{m}_2, r^*_{\mathtt{Ser}})$ where a_{Sim} is generated by $\mathcal{SIM}_{\mathtt{Ser},\mathtt{Rec}}$ in the simulation of π_{Pre}, $(r_{\mathtt{Rec}}, \mathsf{m}_2)$ are generated by $\mathcal{SIM}_{\mathtt{Ser},\mathtt{Rec}}$ in the simulation of π_{Query} and $r^*_{\mathtt{Ser}}$ is a uniformly random string. Moreover, let $\Pr_{\pi,\mathcal{A}_{\mathtt{Ser},\mathtt{Rec}},\kappa,T,q}[a(T), r_{\mathtt{Rec}}, \mathsf{m}_2, r_{\mathtt{Ser}}]$ denote the probability distribution on the values $(a(T), r_{\mathtt{Rec}}, \mathsf{m}_2, r_{\mathtt{Ser}})$ that are generated in a real execution of π with $\mathcal{A}_{\mathtt{Ser},\mathtt{Rec}}$, on inputs T for the sender and q of the receiver. We further denote by $\pi_{\mathsf{Output}}(a_{\mathsf{Sim}}, r_{\mathtt{Rec}}, \mathsf{m}_2, r_{\mathtt{Ser}})$ the output of the receiver in an execution of π with a message a_{Sim} from $\mathtt{Sen}$ to $\mathtt{Ser}$ in π_{Pre}, and a message m_2 from $\mathtt{Sen}$ to $\mathtt{Rec}$ in π_{Query}, where $r_{\mathtt{Rec}}$ and $r_{\mathtt{Ser}}$ denote the respective random tapes of the receiver and the server.

We begin with a claim that states that whenever $(a_{\mathsf{Sim}}, r_{\mathtt{Rec}}, \mathsf{m}_2, r^*_{\mathtt{Ser}})$ are sampled according to the distribution $\Pr_{\mathcal{SIM}_{\mathtt{Ser},\mathtt{Rec}},\kappa,q,T_q}[a_{\mathsf{Sim}}, r_{\mathtt{Rec}}, \mathsf{m}_2, r^*_{\mathtt{Ser}}]$, then the receiver outputs the correct output with probability at least $3/4$. Intuitively, this claim follows by the correctness of the real protocol and the indistinguishability of the ideal and real executions. That is, by the correctness of the protocol it holds that most of the real views $(r_{\mathtt{Ser}}, \mathsf{m}_2)$ yield the correct output, when $r_{\mathtt{Ser}}$ is randomly chosen (recall that by the order of the rounds, $r_{\mathtt{Ser}}$ is independent of $(r_{\mathtt{Rec}}, \mathsf{m}_2)$ in the real protocol). By the security of the protocol this must also hold in the simulation. Therefore, the simulated views must have the property that with a high probability the receiver returns the correct output when $r^*_{\mathtt{Ser}}$ is picked at random.

Claim 4.3 *There exists a κ_0 such that for all $\kappa > \kappa_0$, $T \in \{0,1\}^n$ and $q \in Q_n$,*

$$\Pr_{\mathcal{SIM}_{\mathtt{Ser},\mathtt{Rec}},\kappa,q,T_q}[\pi_{\mathsf{Output}}(a_{\mathsf{Sim}}, r_{\mathtt{Rec}}, \mathsf{m}_2, r^*_{\mathtt{Ser}}) = T_q] \geq \frac{3}{4}. \tag{1}$$

Proof Sketch: Assume that for infinitely many κ's there exists $T \in \{0,1\}^n$ and $q \in Q_n$ such that

$$\Pr_{\mathcal{SIM}_{\mathtt{Ser},\mathtt{Rec}},\kappa,q,T^q}[\pi_{\mathsf{Output}}(a_{\mathsf{Sim}}, r_{\mathtt{Rec}}, \mathsf{m}_2, r^*_{\mathtt{Ser}}) = T^q] < \frac{3}{4}. \tag{2}$$

By the correctness of π, we are guaranteed that for all sufficiently large κ, every $T \in \{0,1\}^n$ and every $q \in Q_n$, there exists a negligible function $\mathsf{negl}(\cdot)$ such that

$$\Pr_{\pi,\mathcal{A}_{\mathtt{Ser},\mathtt{Rec}},\kappa,T,q}[\pi_{\mathsf{Output}}(a(T), r_{\mathtt{Rec}}, \mathsf{m}_2, r_{\mathtt{Ser}}) = T^q] > 1 - \mathsf{negl}(\kappa). \tag{3}$$

Therefore, we can construct a PPT distinguisher D that distinguishes between a real execution of π with $\mathcal{A}_{\mathtt{Ser},\mathtt{Rec}}$ and an ideal execution of $\mathcal{F}_{\mathrm{ODBS}}$ with $\mathcal{SIM}_{\mathtt{Ser},\mathtt{Rec}}$ as follows. Given input T, q and a view $(a, r_{\mathtt{Rec}}, \mathsf{m}_2, r_{\mathtt{Ser}})$ that is either generated by $\mathcal{SIM}_{\mathtt{Ser},\mathtt{Rec}}$ or by the honest parties in a real execution of π, D chooses a uniform random string $r^*_{\mathtt{Ser}}$ and outputs 1 if and only if $\pi_{\mathsf{Output}}(a, r_{\mathtt{Rec}}, \mathsf{m}_2, r^*_{\mathtt{Ser}}) = T^q$.

It is easy to see that if $(a_{\mathsf{Sim}}, r_{\mathtt{Rec}}, \mathsf{m}_2, r_{\mathtt{Ser}})$ were generated by $\mathcal{SIM}_{\mathtt{Ser},\mathtt{Rec}}$, then D outputs 1 with probability $\Pr_{\mathcal{SIM}_{\mathtt{Ser},\mathtt{Rec}},\kappa,q,T^q}[\pi_{\mathsf{Output}}(a_{\mathsf{Sim}}, r_{\mathtt{Rec}}, \mathsf{m}_2, r^*_{\mathtt{Ser}}) = T^q]$, whereas if $(a_{\mathsf{Sim}}, r_{\mathtt{Rec}}, \mathsf{m}_2, r_{\mathtt{Ser}})$ were generated in a real execution of π with $\mathcal{A}_{\mathtt{Ser},\mathtt{Rec}}$, then D outputs 1 with probability $\Pr_{\pi,\mathcal{A}_{\mathtt{Ser},\mathtt{Rec}},\kappa,T,q}[\pi_{\mathsf{Output}}(a(T), r_{\mathtt{Rec}}, \mathsf{m}_2, r_{\mathtt{Ser}}$ $T^q]$. Hence, by Equations. (3) and (2), D distinguishes the views with overwhelming probability. ■

To this end, we fix κ and q. Then, for every a_{Sim} and T_q let

$$\mathsf{GoodView}(a_{\mathsf{Sim}}, T_q) = \left\{ (r_{\mathtt{Rec}}, \mathsf{m}_2) \Big|_{\mathcal{SIM}_{\mathtt{Ser},\mathtt{Rec}},\kappa,q,T_q} \Pr[\pi_{\mathsf{Output}}(a_{\mathsf{Sim}}, r_{\mathtt{Rec}}, \mathsf{m}_2, r^*_{\mathtt{Ser}}) = T_q \mid a_{\mathsf{Sim}}, r_{\mathtt{Rec}}, \mathsf{m}_2] > \frac{1}{2} \right\}.$$

Note that the above probability is only taken over the choice of $r^*_{\mathtt{Ser}}$ which is a uniformly random string. Next, for a fixed T_q we let $\mathcal{E}(T_q)$ denote the expected value of $|\mathsf{GoodView}(a_{\mathsf{Sim}}, T_q)|$ when a_{Sim} is generated by $\mathcal{SIM}_{\mathtt{Ser},\mathtt{Rec}}$ in the simulation of π_{Pre}. That is,

$$\mathcal{E}(T_q) = \mathbb{E}_{a_{\mathsf{Sim}}}\left[|\mathsf{GoodView}(a_{\mathsf{Sim}}, T_q)|\right] = \sum_{a_{\mathsf{Sim}}} \Pr_{\mathcal{SIM}_{\mathtt{Ser},\mathtt{Rec}},\kappa}[a_{\mathsf{Sim}}] \cdot |\mathsf{GoodView}(a_{\mathsf{Sim}}, T_q)|\,.$$

Then, we prove the following claim,

Claim 4.4 *For every T_q, it holds that $\mathcal{E}(T_q) \geq \frac{1}{4}$.*

Proof: Let T_q be such that $\mathcal{E}(T_q) < 1/4$, we show that this contradicts Claim 4.3. First, recall that $\mathcal{E}(T_q) = \mathbb{E}_{a_{\mathsf{Sim}}}\left[|\mathsf{GoodView}(a_{\mathsf{Sim}}, T_q)|\right]$. By the Markov inequality it holds that

$$\Pr_{\mathcal{SIM}_{\mathtt{Ser},\mathtt{Rec}},\kappa}\left[|\mathsf{GoodView}(a_{\mathsf{Sim}}, T_q)| \geq 1\right] < \frac{1}{4}. \tag{4}$$

Then, by the total probability theorem it holds that

$$\begin{aligned}
&\Pr_{\mathcal{SIM}_{\mathtt{Ser},\mathtt{Rec}},n,q,T_q}[\pi_{\mathsf{Output}}(a_{\mathsf{Sim}}, r_{\mathtt{Rec}}, \mathsf{m}_2, r^*_{\mathtt{Ser}}) = T_q] \\
&= \Pr\left[\pi_{\mathsf{Output}}(a_{\mathsf{Sim}}, r_{\mathtt{Rec}}, \mathsf{m}_2, r^*_{\mathtt{Ser}}) = T_q \,\middle|\, |\mathsf{GoodView}(a_{\mathsf{Sim}}, T_q)| \geq 1\right] \cdot \Pr\left[|\mathsf{GoodView}(a_{\mathsf{Sim}}, T_q)| \geq 1\right] \\
&\quad + \Pr\left[\pi_{\mathsf{Output}}(a_{\mathsf{Sim}}, r_{\mathtt{Rec}}, \mathsf{m}_2, r^*_{\mathtt{Ser}}) = T_q \,\middle|\, |\mathsf{GoodView}(a_{\mathsf{Sim}}, T_q)| = 0\right] \cdot \Pr\left[|\mathsf{GoodView}(a_{\mathsf{Sim}}, T_q)| = 0\right] \\
&\leq \Pr\left[|\mathsf{GoodView}(a_{\mathsf{Sim}}, T_q)| \geq 1\right] + \Pr\left[\pi_{\mathsf{Output}}(a_{\mathsf{Sim}}, r_{\mathtt{Rec}}, \mathsf{m}_2, r^*_{\mathtt{Ser}}) = T_q \,\middle|\, |\mathsf{GoodView}(a_{\mathsf{Sim}}, T_q)| = 0\right] \\
&< \frac{1}{4} + \frac{1}{2} = \frac{3}{4}.
\end{aligned}$$

The last inequality is due to Eq. (4) and the definition of $\mathsf{GoodView}(a_{\mathsf{Sim}}, T_q)$. This contradicts Eq. (1). ■

Let $X_{n,q}$ denote the sum of the expected value $\mathcal{E}(T_q)$ when ranging over all possible T_q's. Then, by Claim 4.4 it holds that $X_{n,q} \geq \frac{1}{4} \cdot |\mathcal{ANS}_{n,q}|$. Moreover, it holds that

$$\begin{aligned} X_{n,q} = \sum_{T_q \in \mathcal{ANS}_{n,q}} \mathcal{E}(T_q) &= \sum_{T_q \in \mathcal{ANS}_{n,q}} \sum_{a_{\mathsf{Sim}}} \Pr_{\mathcal{SIM}_{\mathtt{Ser},\mathtt{Rec}}}[a_{\mathsf{Sim}}] \cdot |\mathsf{GoodView}(a_{\mathsf{Sim}}, T_q)| \\ &= \sum_{a_{\mathsf{Sim}}} \Pr_{\mathcal{SIM}_{\mathtt{Ser},\mathtt{Rec}}}[a_{\mathsf{Sim}}] \cdot \sum_{T_q \in \mathcal{ANS}_{n,q}} |\mathsf{GoodView}(a_{\mathsf{Sim}}, T_q)| \,. \end{aligned}$$

Note that for a fixed a_{Sim}, every pair $(r_{\mathtt{Rec}}, \mathsf{m}_2)$ belongs to only one set $\mathsf{GoodView}(a_{\mathsf{Sim}}, T_q)$. This is due to the fact that if $(r_{\mathtt{Rec}}, \mathsf{m}_2) \in \mathsf{GoodView}(a_{\mathsf{Sim}}, T_q)$ for some T_q then by definition the following probability $\Pr[\pi_{\mathsf{Output}}(a_{\mathsf{Sim}}, r_{\mathtt{Rec}}, \mathsf{m}_2, r^*_{\mathtt{Ser}}) = T_q \mid a_{\mathsf{Sim}}, r_{\mathtt{Rec}}, \mathsf{m}_2] > \frac{1}{2}$. This implies that if a pair $(r_{\mathtt{Rec}}, \mathsf{m}_2)$ belongs to two distinct sets $T^0_q \neq T^1_q$, then $\Pr[\pi_{\mathsf{Output}}(a_{\mathsf{Sim}}, r_{\mathtt{Rec}}, \mathsf{m}_2, r^*_{\mathtt{Ser}}) \in \{T^0_q, T^1_q\} \mid a_{\mathsf{Sim}}, r_{\mathtt{Rec}}, \mathsf{m}_2] > 1$. Therefore, for every a_{Sim} the sum $\sum_{T_q} |\mathsf{GoodView}(a_{\mathsf{Sim}}, T_q)|$ is over disjoint sets. We conclude that

$$\sum_{T_q \in \mathcal{ANS}_{n,q}} |\mathsf{GoodView}(a_{\mathsf{Sim}}, T_q)| \leq |\{(r_{\mathtt{Rec}}, \mathsf{m}_2)\}| = \sum_{i \leq \mathsf{cc}^{n,q}_{\mathtt{Sen}}(\kappa) + \mathsf{rand}^{n,q}_{\mathtt{Rec}}(\kappa)} 2^i = 2^{\mathsf{cc}^{n,q}_{\mathtt{Sen}}(\kappa) + \mathsf{rand}^{n,q}_{\mathtt{Rec}}(\kappa) + 1} - 1$$

where the second to the last equality is implied by the fact that $\mathsf{cc}^{n,q}_{\mathtt{Sen}}(\kappa)$ is a bound on the length of m_2 and $\mathsf{rand}^{n,q}_{\mathtt{Rec}}(\kappa)$ is a bound on the length of $r_{\mathtt{Rec}}$. We therefore conclude that

$$X_{n,q} \leq \sum_{a_{\mathsf{Sim}}} \Pr_{\mathcal{SIM}_{\mathtt{Ser},\mathtt{Rec}}}[a_{\mathsf{Sim}}] \cdot \left(2^{\mathsf{cc}^{n,q}_{\mathtt{Sen}}(\kappa) + \mathsf{rand}^{n,q}_{\mathtt{Rec}}(\kappa) + 1} - 1\right) \leq 2^{\mathsf{cc}^{n,q}_{\mathtt{Sen}}(\kappa) + \mathsf{rand}^{n,q}_{\mathtt{Rec}}(\kappa) + 1} - 1.$$

Combining this with the observation that $X_{n,q} \geq \frac{1}{4} \cdot |\mathcal{ANS}_{n,q}|$, we obtain $2^{\mathsf{cc}^{n,q}_{\mathtt{Sen}}(\kappa) + \mathsf{rand}^{n,q}_{\mathtt{Rec}}(\kappa) + 1} - 1 \geq \frac{1}{4} \cdot |\mathcal{ANS}_{n,q}|$ and hence for every query q,

$$\mathsf{cc}^{n,q}_{\mathtt{Sen}}(\kappa) + \mathsf{rand}^{n,q}_{\mathtt{Rec}}(\kappa) \geq \log\left(\frac{1}{4}\,|\mathcal{ANS}_{n,q}|\right) - 1 = \log|\mathcal{ANS}_{n,q}| - 3.$$

Therefore for every query q, it either holds that $\mathsf{cc}^{n,q}_{\mathtt{Sen}}(\kappa) \geq \frac{\log|\mathcal{ANS}_{n,q}|-3}{2}$ or $\mathsf{rand}^{n,q}_{\mathtt{Rec}}(\kappa) \geq \frac{\log|\mathcal{ANS}_{n,q}|-3}{2}$. Recall that $H_{n,Q} = \max_{q \in Q_n} \log|\mathcal{ANS}_{n,q}|$. We conclude that there exists a query $q \in Q_n$ for which either $\mathsf{cc}^{n,q}_{\mathtt{Sen}}(\kappa) \geq \frac{H_{n,Q}-3}{2}$ or $\mathsf{rand}^{n,q}_{\mathtt{Rec}}(\kappa) \geq \frac{H_{n,Q}-3}{2}$. Note that if the former inequality holds, then by the security of π the communication complexity is at least $\frac{H_{n,Q}-3}{2}$ *for all queries* $q \in Q_n$ (otherwise, the sender can learn the receiver's input by just looking at the length of the messages sent in π_{Query}, thus breaking privacy). This concludes the proof of Theorem 4.2.

■

Lemma 4.6 below demonstrates that for the pattern matching functionality there exists a family of queries Q such that $H_{n,Q} = n$ for every n. Combining this with Theorems 4.1-4.2, the following holds,

Corollary 4.5 *There exists a family of queries $Q = \{Q_n\}_n$ such that for* any *protocol with minimal interaction that implements the outsourced pattern matching functionality securely with respect to Q in the private channels setting, for every n one of the following holds:*

1. *There exists $q \in Q_n$ such that the communication complexity in the query phase is at least $\frac{n-3}{2} - s$;*
2. *There exists $q \in Q_n$ such that the length of the receiver's random tape is at least $\frac{n-3}{2} - s$.*

A bound on $H_{n,Q}$ for pattern matching. We prove the following simple observation relative to the pattern matching functionality; see Section 3.1 for the definition of this functionality.

Lemma 4.6 *For the pattern matching functionality there exists a family of queries Q such that $H_{n,Q} = n$ for every n.*

Proof: We prove the existence of a family of queries $Q = \{Q_n\}_n$ such that $H_{n,Q} = n$ for every n. Fix n and let $Q_n = \{0\}$ denote the single-bit pattern $q = 0$. In addition, recall that $H_{n,Q} = \max_{q \in Q_n} \log |\mathcal{ANS}_{n,q}|$ where $\mathcal{ANS}_{n,q} = \{T^q \mid T \in \{0,1\}^n\}$. Note that $\mathcal{ANS}_{n,q=0}$ includes all subsets of $[n]$ and thus, $|\mathcal{ANS}_{n,q=0}| = 2^n$ and $\log |\mathcal{ANS}_{n,q=0}| = n$, implying that $H_{n,Q} \geq \log |\mathcal{ANS}_{n,q=0}| = n$. ■

4.2 The Non-Private Channels Case

In this setting a corrupted party observes the communication between the honest parties. In our context this implies that a corrupted receiver sees the setup message sent from the sender to the server. Consequently, we only need to consider the corruption of the receiver where the order of communication in the query phase does not matter as in the private channels case. We continue with our main theorem for this section.

Theorem 4.7 *Fix n and m, and let $\pi = (\pi_{\mathsf{Pre}}, \pi_{\mathsf{Query}})$ be a protocol with minimal interaction that securely implements $\mathcal{F}_{\text{ODBS}}$ with respect to queries $Q = \{Q_n\}_n$ in the presence of semi-honest adversaries in the non-private channels setting. Then one of the following holds:*

1. For every query $q \in Q_n$ *the communication complexity* $\mathsf{cc}^{n,q} \geq \frac{H_{n,Q}-2}{2}$ *or*
2. There exists a query $q \in Q_n$ *such that* $\mathsf{rand}^{n,q}_{\mathtt{Rec}}(\kappa) \geq \frac{H_{n,Q}-2}{2}$.

Proof: The proof of Theorem 4.7 is very similar to the proof of Theorem 4.2. We present the outline of the proof. Let $\pi = (\pi_{\mathsf{Pre}}, \pi_{\mathsf{Query}})$ be as in Theorem 4.7, let $\mathcal{A}_{\mathtt{Rec}}$ be a real-world semi-honest adversary controlling the receiver (note that since we do not assume private channels, $\mathcal{A}_{\mathtt{Rec}}$ sees all communication between the honest parties and in particular the message within π_{Pre}), and let $\mathcal{SIM}_{\mathtt{Rec}}$ be an ideal-world adversary guaranteed to exist by the security of $\pi = (\pi_{\mathsf{Pre}}, \pi_{\mathsf{Query}})$. By definition, upon given a message $(\mathsf{preprocess}, |T|, m)$ in the setup phase $\mathcal{SIM}_{\mathtt{Ser},\mathtt{Rec}}$ outputs a string a_{Sim}. Moreover, upon given a message $(\mathsf{response}, q, T_q, \mathsf{id})$ in the query phase it outputs a valid view for $\mathcal{A}_{\mathtt{Ser}}$which consists of a triple $(r_{\mathtt{Rec}}, \mathsf{m}_2, \mathsf{m}_4)$, where $r_{\mathtt{Rec}}$ is the random tape of $\mathtt{Rec}$, m_2 is a simulated message from $\mathtt{Sen}$ to $\mathtt{Rec}$ and m_4 is a simulated message from $\mathtt{Ser}$ to $\mathtt{Rec}$.

For a security parameter κ and a pair of query/response (q, T_q), let $\Pr_{\mathcal{SIM}_{\mathtt{Rec}},\kappa}[a_{\mathsf{Sim}}]$ denote the probability distribution over the simulated message of π_{Pre} and let $\Pr_{\mathcal{SIM}_{\mathtt{Rec}},\kappa,q,T_q}[a_{\mathsf{Sim}}, r_{\mathtt{Rec}}, \mathsf{m}_2, \mathsf{m}_4]$ denote the probability distribution on the values $(a_{\mathsf{Sim}}, r_{\mathtt{Rec}}, \mathsf{m}_2, \mathsf{m}_4)$ where a_{Sim} is generated in the simulation of π_{Pre}, and $r_{\mathtt{Rec}}, \mathsf{m}_2, \mathsf{m}_4$ are generated in the simulation of π_{Query}. Moreover, let $\Pr_{\pi,\mathcal{A}_{\mathtt{Rec}},\kappa,T,q}[a(T), r_{\mathtt{Rec}}, \mathsf{m}_2, \mathsf{m}_4]$ denote the probability distribution over the values $(a(T), r_{\mathtt{Rec}}, \mathsf{m}_2, \mathsf{m}_4)$ that are generated in a real execution of π with $\mathcal{A}_{\mathtt{Rec}}$, on inputs T for the sender and q for the receiver. We further denote by $\pi_{\mathsf{Output}}(a_{\mathsf{Sim}}, r_{\mathtt{Rec}}, \mathsf{m}_2, \mathsf{m}_4)$ the output of the receiver in an execution of π with a message a_{Sim} from $\mathtt{Sen}$ to $\mathtt{Ser}$ in π_{Pre}, a message m_2 from $\mathtt{Sen}$ to $\mathtt{Rec}$ in π_{Query} and a message m_4 from $\mathtt{Ser}$ to $\mathtt{Rec}$, where $r_{\mathtt{Rec}}$ denotes the random tape of the receiver.

We continue with the following claim,

Claim 4.8 *There exists a κ_0 such that for all $\kappa > \kappa_0$ and $T \in \{0,1\}^n$, $q \in Q_n$,*

$$\Pr_{\mathcal{SIM}_{\mathtt{Rec}},\kappa,q,T_q}[\pi_{\mathsf{Output}}(a_{\mathsf{Sim}}, r_{\mathtt{Rec}}, \mathsf{m}_2, \mathsf{m}_4) = T_q] \geq \frac{1}{2}. \tag{5}$$

Proof Sketch: Assume that for infinitely many κ's there exists $T \in \{0,1\}^n$ and $q \in Q_n$ such that

$$\Pr_{\mathcal{SIM}_{\mathtt{Rec}},\kappa,q,T_q}[\pi_{\mathsf{Output}}(a_{\mathsf{Sim}}, r_{\mathtt{Rec}}, \mathsf{m}_2, \mathsf{m}_4) = T_q] < \frac{1}{2}.$$

By the correctness of protocol π, it is guaranteed that for all sufficiently large κ, every $T \in \{0,1\}^n$ and every $q \in Q_n$, there exists a negligible function $\mathsf{negl}(\cdot)$ such that

$$\Pr_{\pi,\mathcal{A}_{\mathtt{Rec}},\kappa,T,q}[\pi_{\mathsf{Output}}(a(T), r_{\mathtt{Rec}}, \mathsf{m}_2, \mathsf{m}_4) = T_q] > 1 - \mathsf{negl}(\kappa)$$

Therefore we can construct a PPT distinguisher D that distinguishes a real execution of π with $\mathcal{A}_{\mathtt{Rec}}$ and an ideal execution of $\mathcal{F}_{\mathrm{ODBS}}$ with $\mathcal{SIM}_{\mathtt{Rec}}$ as follows. Given input T, q and view $a, r_{\mathtt{Rec}}, \mathsf{m}_2, \mathsf{m}_4$, output 1 if and only if the receiver's output is T_q. It is easy to verify that there is a non-negligible gap relative to the real and the simulated views, and thus D distinguishes the executions with this gap. ■

To this end, we fix κ and q. Then, for every a_{Sim} and T_q let

$$\mathsf{GoodView}(a_{\mathsf{Sim}}, T_q) = \{(r_{\mathtt{Rec}}, \mathsf{m}_2, \mathsf{m}_4) \mid \pi_{\mathsf{Output}}(a_{\mathsf{Sim}}, r_{\mathtt{Rec}}, \mathsf{m}_2, \mathsf{m}_4) = T_q\}.$$

For a fixed T_q, we let $\mathcal{E}(T_q)$ denote the expected value of $|\mathsf{GoodView}(a_{\mathsf{Sim}}, T_q)|$ when a_{Sim} is generated by $\mathcal{SIM}_{\mathtt{Rec}}$ in the simulation of π_{Pre}. The following claim is proved similarly to the proof of Claim 4.4:

Claim 4.9 *For every T_q, it holds that*

$$\mathcal{E}(T_q) \geq \frac{1}{2}.$$

Let $X_{n,q}$ denote the sum of the expected value $\mathcal{E}(T_q)$ when ranging over all possible T_q's. We have that

$$X_{n,q} = \sum_{a_{\mathsf{Sim}}} \Pr_{\mathcal{SIM}_{\mathtt{Rec}}}[a_{\mathsf{Sim}}] \cdot \sum_{T_q \in \mathcal{AMS}_{n,q}} |\mathsf{GoodView}(a_{\mathsf{Sim}}, T_q)|.$$

Then by Claim 4.9, we have that $X_{n,q} \geq \frac{1}{2} \cdot |\mathcal{AMS}_{n,q}|$.

Note that for a fixed a_{Sim}, every triple $(r_{\mathtt{Rec}}, \mathsf{m}_2, \mathsf{m}_4)$ belongs to only one set $\mathsf{GoodView}(a_{\mathsf{Sim}}, T_q)$. This is due to the fact that a triple $(r_{\mathtt{Rec}}, \mathsf{m}_2, \mathsf{m}_4)$ fixes the output of $\mathtt{Rec}$. Therefore, for every a_{Sim} the sum $\sum_{T_q} |\mathsf{GoodView}(a_{\mathsf{Sim}}, T_q)|$ is of disjoint sets. We conclude that

$$\sum_{T_q \in \mathcal{AMS}_{n,q}} |\mathsf{GoodView}(a_{\mathsf{Sim}}, T_q)| \leq |\{(r_{\mathtt{Rec}}, \mathsf{m}_2, \mathsf{m}_4)\}| \leq \sum_{i \leq \mathsf{cc}^{n,q}(\kappa) + \mathsf{rand}^{n,q}_{\mathtt{Rec}}(\kappa)} 2^i = 2^{\mathsf{cc}^{n,q}(\kappa) + \mathsf{rand}^{n,q}_{\mathtt{Rec}}(\kappa) + 1} - 1$$

where the second to the last inequality is implied by the fact that $\mathsf{cc}^{n,q}(\kappa)$ is a bound on the overall communication complexity in π_{Query} and $\mathsf{rand}^{n,q}_{\mathtt{Rec}}(\kappa)$ is a bound on the length of $r_{\mathtt{Rec}}$. We therefore conclude that

$$X_{n,q} \leq \sum_{a_{\mathsf{Sim}}} \Pr_{\mathcal{SIM}_{\mathtt{Rec}}}[a_{\mathsf{Sim}}] \cdot \left(2^{\mathsf{cc}^{n,q}(\kappa) + \mathsf{rand}^{n,q}_{\mathtt{Rec}}(\kappa) + 1} - 1\right) \leq 2^{\mathsf{cc}^{n,q}(\kappa) + \mathsf{rand}^{n,q}_{\mathtt{Rec}}(\kappa) + 1} - 1.$$

Combining this with the observation that $X_{n,q} \geq \frac{1}{2} \cdot |\mathcal{AMS}_{n,q}|$, we obtain

$$2^{\mathsf{cc}^{n,q}(\kappa) + \mathsf{rand}^{n,q}_{\mathtt{Rec}}(\kappa) + 1} - 1 \geq \frac{1}{2} \cdot |\mathcal{AMS}_{n,q}|$$

and hence for every query q,

$$\mathsf{cc}^{n,q}(\kappa) + \mathsf{rand}^{n,q}_{\mathtt{Rec}}(\kappa) \geq \log\left(\frac{1}{2}|\mathcal{AMS}_{n,q}|\right) - 1 = \log|\mathcal{AMS}_{n,q}| - 2.$$

We conclude the proof of Theorem 4.7 similarly to the proof of Theorem 4.2. ■

Applying Lemma 4.6 we obtain he following corollary,

Corollary 4.10 *There exists a family of queries $Q = \{Q_n\}_n$ such that for* any *protocol with minimal interaction that securely implements the outsourced pattern matching functionality with respect to Q in the non-private channels setting and for every n one of the following holds:*

1. *The communication complexity between the sender and the receiver in π_{Query} for any $q \in Q_n$ is at least $(n-2)/2$;*
2. *There exists $q \in Q_n$ such that the length of the receiver's random tape is at least $(n-2)/2$.*

4.3 Difficulties with Proving a Communication Complexity Lower Bound

Recall that our infeasibility result provides a lower bound on either the communication complexity of an outsourced protocol or the size of the receiver's random tape. Clearly, it would be preferable if we could give a strict lower bound on each of these complexities separately. Towards achieving this goal, it seems very appealing to use a pseudorandom generator G that shortens the length of the receiver's random tape. Namely, replace the uniform randomness of the receiver in an outsourced protocol π by an output of a pseudorandom generator, computed on a shorter seed of length κ; thus obtaining a new protocol π' where the length of the random tape of the receiver is bounded by κ. It is simple to observe that the communication complexity of π' is exactly the same as the communication complexity of π. We can then apply our lower bound on π' in order to claim that either the random tape of $\mathtt{Rec}'$ in π' is large or the communication complexity of π' is large. Now, since we already know that the random tape of $\mathtt{Rec}'$ is of length κ, we conclude that the communication complexity of π' must be large; hence obtaining that the communication complexity of π is large as well.

Unfortunately, this intuition fails when trying to formalize it. We demonstrate why it fails as follows. Let $\pi = (\pi_{\mathsf{Pre}}, \pi_{\mathsf{Query}})$ be a protocol for securely computing $\mathcal{F}_{\mathrm{ODBS}}$ in the presence of $(\mathtt{Ser}/\mathtt{Rec})$-collusion and semi-honest adversaries, and let π' be a protocol obtained from π by having the receiver $\mathtt{Rec}'$ pick a random seed $s \in \{0,1\}^\kappa$ and invoke $\mathtt{Rec}$ with randomness $G(s)$. Our goal is to show that π' is also secure in the presence of $(\mathtt{Ser}'/\mathtt{Rec}')$-collusion and semi-honest adversaries by reducing its security to the security of π. Namely, we need to simulate the view of the corrupted parties in π' using the simulators constructed in the security proof of π. Consider the corruption case of the receiver $\mathtt{Rec}$ in π. Then, in order to construct a simulator $\mathcal{SIM}'$ for the corrupted receiver $\mathtt{Rec}'$ in π' we need to invoke simulator $\mathcal{SIM}$ and use its output in order to produce a simulated view for $\mathtt{Rec}'$.

Recall that a valid view of $\mathtt{Rec}$ consists of a pair $(r_{\mathtt{Rec}}, \mathsf{trans})$, where $r_{\mathtt{Rec}}$ is a random string of length $\mathsf{rand}^{n,q}_{\mathtt{Rec}}(\kappa)$ and trans are the incoming messages that $\mathtt{Rec}$ observes during the execution of π_{Query} with randomness $r_{\mathtt{Rec}}$, whereas a valid view for $\mathtt{Rec}'$ consists of a pair (s, trans) where s is a random seed of length κ and trans are the incoming message that $\mathtt{Rec}'$ observes during the execution of π_{Query} with randomness $G(s)$. Then, it is not clear how to use the output $(r_{\mathtt{Rec}}, \mathsf{trans})$ of $\mathcal{SIM}$ in order to construct a simulated view (s, trans) for $\mathtt{Rec}'$ within π'. Specifically, the difficulty is mainly because it might be that $\mathcal{SIM}$ outputs only views for which $r_{\mathtt{Rec}}$ is not in the range of G, and hence obtaining a corresponding s (that is part of $\mathcal{SIM}'$'s output) is not even possible.

Finally, we remark that any attempt to relax the security definition in a way that forces $\mathcal{SIM}$ to only output strings $r_{\mathtt{Rec}}$ that have preimages relative to G, fails as well. This is because in this case the real and the ideal ensembles that correspond to $\mathtt{Rec}'$'s view must consist of the seed s to the pseudorandom generator. This implies that the security argument cannot be based on the indistinguishability of $G(s)$ from a random string of the appropriate length.

References

[AIK10] Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. From secrecy to soundness: Efficient verification via secure computation. In *ICALP (1)*, pages 152–163, 2010.

[AJLA+12] Gilad Asharov, Abhishek Jain, Adriana López-Alt, Eran Tromer, Vinod Vaikuntanathan, and Daniel Wichs. Multiparty computation with low communication, computation and interaction via threshold fhe. In *EUROCRYPT*, pages 483–501, 2012.

[ANSS16] Gilad Asharov, Moni Naor, Gil Segev, and Ido Shahaf. Searchable symmetric encryption: Optimal locality in linear space via two-dimensional balanced allocations. *IACR Cryptology ePrint Archive*, 2016:251, 2016.

[BCCT12] Nir Bitansky, Ran Canetti, Alessandro Chiesa, and Eran Tromer. From extractable collision resistance to succinct non-interactive arguments of knowledge, and back again. In *ITCS*, pages 326–349, 2012.

[Can00] Ran Canetti. Security and composition of multi-party cryptographic protocols. *Journal of Cryptology*, 13:143–202, 2000.

[CGKO11] Reza Curtmola, Juan A. Garay, Seny Kamara, and Rafail Ostrovsky. Searchable symmetric encryption: Improved definitions and efficient constructions. *Journal of Computer Security*, 19(5):895–934, 2011.

[CGPR15] David Cash, Paul Grubbs, Jason Perry, and Thomas Ristenpart. Leakage-abuse attacks against searchable encryption. In *CCS*, pages 668–679, 2015.

[CK10] Melissa Chase and Seny Kamara. Structured encryption and controlled disclosure. In *ASIACRYPT*, pages 577–594, 2010.

[CKKC13] Seung Geol Choi, Jonathan Katz, Ranjit Kumaresan, and Carlos Cid. Multi-client non-interactive verifiable computation. In *TCC*, pages 499–518, 2013.

[CS14] Melissa Chase and Emily Shen. Pattern matching encryption. *IACR Cryptology ePrint Archive*, 2014:638, 2014.

[DFH12] Ivan Damgård, Sebastian Faust, and Carmit Hazay. Secure two-party computation with low communication. In *TCC*, pages 54–74, 2012.

[EGL85] Shimon Even, Oded Goldreich, and Abraham Lempel. A randomized protocol for signing contracts. *Commun. ACM*, 28(6):637–647, 1985.

[FHV13] Sebastian Faust, Carmit Hazay, and Daniele Venturi. Outsourced pattern matching. In *ICALP*, 2013.

[FIPR05] Michael J. Freedman, Yuval Ishai, Benny Pinkas, and Omer Reingold. Keyword search and oblivious pseudorandom functions. In *TCC*, pages 303–324, 2005.

[GGP10] Rosario Gennaro, Craig Gentry, and Bryan Parno. Non-interactive verifiable computing: Outsourcing computation to untrusted workers. In *CRYPTO*, pages 465–482, 2010.

[GH11] Matthew Green and Susan Hohenberger. Practical adaptive oblivious transfer from simple assumptions. In *TCC*, pages 347–363, 2011.

[GLR11] Shafi Goldwasser, Huijia Lin, and Aviad Rubinstein. Delegation of computation without rejection problem from designated verifier cs-proofs. *IACR Cryptology ePrint Archive*, 2011:456, 2011.

[GMPP16] Sanjam Garg, Pratyay Mukherjee, Omkant Pandey, and Antigoni Polychroniadou. The exact round complexity of secure computation. *To appear at Eurocrypt*, 2016.

[Goh03] Eu-Jin Goh. Secure indexes. *IACR Cryptology ePrint Archive*, 2003:216, 2003.

[HT10] Carmit Hazay and Tomas Toft. Computationally secure pattern matching in the presence of malicious adversaries. In *ASIACRYPT*, pages 195–212, 2010.

[JJK+13] Stanislaw Jarecki, Charanjit S. Jutla, Hugo Krawczyk, Marcel-Catalin Rosu, and Michael Steiner. Outsourced symmetric private information retrieval. In *ACM Conference on Computer and Communications Security*, pages 875–888, 2013.

[KMR11] Seny Kamara, Payman Mohassel, and Mariana Raykova. Outsourcing multi-party computation. *IACR Cryptology ePrint Archive*, 2011:272, 2011.

[KMR12] Seny Kamara, Payman Mohassel, and Ben Riva. Salus: a system for server-aided secure function evaluation. In *ACM Conference on Computer and Communications Security*, pages 797–808, 2012.

[KO04] Jonathan Katz and Rafail Ostrovsky. Round-optimal secure two-party computation. In *CRYPTO*, pages 335–354, 2004.

[KP13] Seny Kamara and Charalampos Papamanthou. Parallel and dynamic searchable symmetric encryption. In *Financial Cryptography*, pages 258–274, 2013.

[KPR12] Seny Kamara, Charalampos Papamanthou, and Tom Roeder. Dynamic searchable symmetric encryption. In *ACM Conference on Computer and Communications Security*, pages 965–976, 2012.

[LATV12] Adriana López-Alt, Eran Tromer, and Vinod Vaikuntanathan. On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption. In *STOC*, pages 1219–1234, 2012.

[Nie02] Jesper Buus Nielsen. Separating random oracle proofs from complexity theoretic proofs: The non-committing encryption case. In *CRYPTO*, pages 111–126, 2002.

[NP99] Moni Naor and Benny Pinkas. Oblivious transfer with adaptive queries. In *CRYPTO*, pages 573–590, 1999.

[NR97] Moni Naor and Omer Reingold. Number-theoretic constructions of efficient pseudo-random functions. In *FOCS*, pages 458–467, 1997.

[ORS15] Rafail Ostrovsky, Silas Richelson, and Alessandra Scafuro. Round-optimal black-box two-party computation. In *CRYPTO*, pages 339–358, 2015.

[Wei73] Peter Weiner. Linear pattern matching algorithms. In *SWAT (FOCS)*, pages 1–11, 1973.

[Yao86] Andrew Chi-Chih Yao. How to generate and exchange secrets (extended abstract). In *FOCS*, pages 162–167, 1986.