On the Power of Random Oracles

Iftach Haitner[*†] Eran Omri[‡§†] Hila Zarosim[¶ §]

October 9, 2012

Abstract

In the *random oracle* model, the parties are given oracle access to a random member of a (typically huge) function family, and are assumed to have *unbounded* computational power (though they can only make a bounded number of oracle queries). This model provides powerful properties that allow *proving* the security of many protocols, even such that cannot be proved secure in the standard model (under *any* hardness assumption). The random oracle model is also used to show that a given cryptographic primitive *cannot* be used in a black-box way to construct another primitive; in their seminal work, Impagliazzo and Rudich [STOC '89] showed that in the *random function* model – when the function family is the set of *all* functions – it is impossible to construct (secure) key-agreement protocols, yielding that key-agreement cannot be *black-box* reduced to one-way functions. Their work has a long line of followup works (Simon [EC '98], Gertner et al. [STOC '00] and Gennaro et al. [SICOMP '05], to name a few), showing that given oracle access to a certain type of function family (e.g., the family that "implements" public-key encryption) is not sufficient for building a given cryptographic primitive (e.g., oblivious transfer). Yet, in the more general sense, the following fundamental question remained open:

> *What is the exact power of the random oracle model, and more specifically, of the random function model?*

We make progress towards answering the above question, showing that *any* (no private input) semi-honest two-party functionality that can be securely implemented in the random function model, *can* be securely implemented information theoretically (where parties are assumed to be all powerful, and no oracle is given). We further generalize the above result to function families that provide some natural combinatorial property.

To exhibit the power of our result, we use the recent *information theoretic* impossibility result of McGregor et al. [FOCS '10], to show the existence of functionalities (e.g., inner product) that *cannot* be computed both accurately and in a differentially private manner in the random function model; yielding that protocols for computing these functionalities cannot be black-box reduced to the existence of one-way functions.

Keywords: random oracles; black-box separations; one-way functions; differential privacy; key agreement

[*]School of Computer Science, Tel Aviv University. E-mail: iftachh@cs.tau.ac.il.

[†]Supported by the Israeli Centers of Research Excellence (I-CORE) program (Center No. 4/11)

[‡]Department of Mathematics and Computer Science, Ariel University Center. E-mail: omrier@gmail.com. Research was done while Eran Omri was at Bar Ilan University.

[§]This work was supported by the ISRAEL SCIENCE FOUNDATION (grant No. 189/11)

[¶]Department of Computer Science, Bar Ilan University. E-mail: zarosih@cs.biu.ac.il.

1 Introduction

In the *random-oracle* model, the parties are given oracle access to a random member of a (typically huge) function family, and are assumed to have unbounded computational power (though they can only make a bounded number of oracle queries). Instantiations of this model with a given function family are typically very powerful, and can be used to implement many cryptographic primitives, such as one-way functions and cryptographic hash functions, with extremely strong security. More importantly, it can even be used for implementing secure protocols for tasks that are hard to implement in the standard model, and sometimes even completely unachievable; a well known example is the work of Fiat and Shamir [7], showing how to convert three-message identification schemes to a highly efficient (non interactive) signature scheme. Their scheme is provably secure in the *random-function* model – an instantiation of the random oracle-model with the function family being the set of *all* functions [20].[1] Under standard cryptographic assumptions, however, there exist protocols (with no oracle access) whose Fiat-Shamir variant (secure in the random-function model), cannot be proven secure under any implementation of the function family, and any hardness assumption [11, 3].

On a different route, the random-oracle model was used to show that one cryptographic primitive *cannot* be used in a black-box way to construct another primitive. In their seminal work, Impagliazzo and Rudich [13] showed that in the random-function model, it is impossible to construct (secure) key-agreement protocols, yielding that key-agreement cannot be black-box reduced to one-way functions. Their work has initiated a long line of follow up works (Simon [24], Gertner et al. [10], and Gennaro et al. [9], to name a few) showing that given oracle access to a certain type of function family (e.g., the family that "implements" public-key encryption) is not sufficient for building a given cryptographic primitive (e.g., oblivious transfer). Yet, in the more general sense, the following fundamental question remained open:

> *What is the exact power of the random-oracle model, and more specifically, of the random-function model?*

Apart from being aesthetic mathematically, answers to these question are very likely to enrich our understanding of (the limitations of) black-box reductions in cryptography.

It is well known that for malicious adversaries, there exist functionalities that cannot be achieved in the *information-theoretic model*, i.e., where all entities are assumed to be unbounded (with no oracle access), yet can be securely computed in the random-function model (e.g., commitment schemes, coin-tossing protocols and, non-trivial zero-knowledge proofs). All of these functionalities, however, are blatantly trivial when considering semi-honest adversaries, which is the focus of this work.

1.1 Our Result

We make progress towards answering the above question, showing that *any* (no private input) semi-honest, two-party computation that can be securely implemented in the *random-function* model, *can* be securely implemented in the *information-theoretic* model.

[1] It is common to identify the random-oracle model with the (more specific) random-function model defined above. Here, however, we distinguish between the two.

Theorem 1 (Main theorem, informal)**.** *Let π be a no private-input, m-round, ℓ-query, oracle-aided two-party protocol. Then for any $\varepsilon > 0$, there exists an $O(\ell^2/\varepsilon^2)$-query oracle-aided function Map, and a stateless, no oracle, m-round protocol $\tilde{\pi}$ such that:*

$$\mathrm{SD}\left(\left(\mathrm{out}_{\mathsf{A}}, \mathrm{out}_{\mathsf{B}}, \mathsf{Map}^f(\bar{t})\right)_{f \leftarrow \mathcal{F}_{\mathrm{AF}}, (\mathrm{out}_{\mathsf{A}}, \mathrm{out}_{\mathsf{B}}, \bar{t}) \leftarrow \langle \mathsf{A}^f, \mathsf{B}^f \rangle}, \left\langle \tilde{\mathsf{A}}, \tilde{\mathsf{B}} \right\rangle\right) \in O(\varepsilon),$$

where $\mathcal{F}_{\mathrm{AF}}$ is the all functions family, and $\langle X, Y \rangle$ stands for a random execution of the protocol (X, Y), resulting in the parties' private outputs and the common transcript.

Namely, the distributions induced by a random execution of π^f (for a random $f \leftarrow \mathcal{F}_{\mathrm{AF}}$) on the parties' private outputs and the common transcript, is almost the same as that induced by a random execution of the (no oracle) protocol $\tilde{\pi}$, where the only difference is that one needs to apply an efficient procedure Map to π's transcript.

Theorem 1 generalizes to all function families with the property that answers for *distinct* queries, induced by drawing a random member from the family, are independent.

1.1.1 Applications

Informally speaking, Theorem 1 yields the following meta result.

Corollary 2 (very informal)**.** *Assume that π securely realizes a no-input functionality P in the random-function model, then the security of P against semi-honest adversaries is* trivial *— can be realized in the information-theoretic model.*

We give two concrete instantiations of the above informal claim. The first example (reproving [13, 1]) concerns the existence of key-agreement protocols in the random-function model. Recall that key-agreement protocols cannot be realized in the information-theoretic model. Namely, for any (no oracle) protocol π, there exists a passive (i.e., semi-honest) adversary that extracts the key from the protocol's transcript. Hence, Theorem 1 yields that key-agreement protocols cannot be realized in the random-function model, and thus key-agreement protocols cannot be black-box reduced to one-way functions, reproving the result of [13, 1]. The actual parameters achieved by applying Theorem 1, match the optimal bound given in Barak and Mahmoody [1].

As a second more detailed example, we prove that in the random-function model, it is impossible for two parties to approximately compute the inner-product function in a *differentially private* manner. Namely, in a way that very little information is leaked about *any* single bit of the input of each party, to the other party. A recent result of McGregor et al. [17] shows that in the information-theoretic model, it is impossible to approximately compute the inner product function in a differentially private manner. Combining their result with Theorem 1, we obtain the following fact.[2]

Theorem 3. *Any ℓ-query, (ℓ^2, α, γ)-differentially private protocol, errs (with constant probability) with magnitude at least $\frac{\sqrt{n}}{\log(n) \cdot e^{\alpha}}$, in computing the inner product of two n-bits strings.*

[2]We mention that the result of [17] is stated for protocol with inputs, where Theorem 1 is only applicable to no-input protocols. Indeed, a fair amount of work was needed to derive an impossibility result for no-input protocols, from the work of [17].

Very informally, a protocol is (k, α, γ)-differentially private, if no party making at most k queries to the oracle, learns more then ε information about one of the other party's input bits, except with some small probability γ.

The above result yields the impossibility of fully black-box reducing differentially private protocols for (well) approximating two-party inner-product to the existence of one-way functions. Roughly speaking, such a fully black-box reduction is a pair of efficient oracle-aided algorithms (Q, R) such that the following hold: (1) Q^f is a good approximation protocol of the inner-product for any function f, and (2) $\mathsf{R}^{f,\mathcal{A}}$ inverts f, for any adversary $\mathcal{A}$ that learns too much about the input of one of the parties in Q^f. Since a random sample from the all-function family is hard to invert (cf., [13, 9]), the existence of such a reduction yields that Q^f is differentially-private with respect to poly-query adversaries, when f is chosen at random from the set of all functions.[3] Hence, Theorem 3 yields the following result.

Corollary 4. *There exists* no *fully black-box reduction from (α, γ)-differentially private protocol computing the inner product of two n-bit strings with error magnitude less than $\frac{\sqrt{n}}{\log(n) \cdot e^{\alpha}}$, to one-way functions.*

Finally, following an observation made by McGregor et al. [17], Theorem 3 and Corollary 4 imply similar results for two-party differentially private protocols for the Hamming distance functionality.[4]

1.2 Our Technique

When using a no-oracle protocol to emulate an oracle-aided protocol π, having oracle access to a random member of the all-function family, the crucial issue is to find all *common* information the parties share at a given point. The clear obstacle are the oracle calls: the parties might share information without explicitly communicating it, say by making the same oracle call.

Here comes into play the *Dependency Finder* of Impagliazzo and Rudich, and Barak and Mahmoody (algorithm *Eve*, in their terminology). This oracle-aided algorithm (Finder, hereafter) gets as input a communication transcript $\bar{t}$ of a random execution of π, and an oracle access to f, the "oracle" used by the parties in this execution. Algorithm Finder outputs a list of query/answer pairs to f that, with high probability, contains *all* oracle queries that are *common* to both parties (and possibly also additional ones). Moreover, with high probability Finder is guaranteed not to make "too many" oracle queries.

Equipped with Finder, we give the following definition for the mapping procedure Map and the stateless (no oracle) protocol $\tilde{\pi} = (\tilde{\mathsf{A}}, \tilde{\mathsf{B}})$: on a communication transcript $\bar{t}$, the oracle-aided algorithm Map^f outputs $\big((\bar{t}_1, \mathcal{I}_1 = \mathsf{Finder}^f(\bar{t}_1)), (\bar{t}_{1,2}, \mathcal{I}_2 = \mathsf{Finder}^f(\bar{t}_{1,2})) \ldots, (\bar{t}, \mathcal{I}_m = \mathsf{Finder}^f(\bar{t}))\big)$. Namely, Map invokes Finder on each prefix of the transcript, and outputs the result. The no-oracle protocol $\tilde{\pi} = (\tilde{\mathsf{A}}, \tilde{\mathsf{B}})$ is defined as follows: assume that $\tilde{\mathsf{A}}$ speaks at round $(i+1)$, and that the i'th message is $((\bar{t}_1, \mathcal{I}_1), \ldots, (\bar{t}_{1,\ldots,i}, \mathcal{I}_i))$. The stateless, plain-model $\tilde{\mathsf{A}}$ samples random values for $f \in \mathcal{F}_{\text{AF}}$ and the random coins of A, conditioned on $(\bar{t}_{1,\ldots,i}, \mathcal{I}_i)$ being the protocol's transcript. It then lets t_{i+1} be the next message of A induced by the above choice of f and random coins,

[3] Assume towards a contradiction the existence of a poly-query adversary $\mathcal{A}$ for Q^f, then the poly-query $\mathsf{R}^{f,\mathcal{A}}$ would successfully invert a random f.

[4] The inner product between two bit strings x, y can be expressed as $\mathsf{IP}(x, y) = w(x) + w(y) + H_d(x, y)$, where the weight $w(z)$ is number of 1-bits in z. Thus, a differentially private protocol for estimating the Hamming distance $H_d(x, y)$ can be turned into one for the inner product by having the parties send differentially private approximations of the weights of their inputs.

and sends $\left(\bar{t}' = (\bar{t}_{1,\ldots,i}, t_{i+1}), \mathsf{Finder}^f(\bar{t}')\right)$ back to $\tilde{\mathsf{B}}$. In case this is the last round of interaction, $\tilde{\mathsf{A}}$ locally outputs the (local) output of A induced by this choice of f and random coins. In other words, $\tilde{\mathsf{A}}$ selects a random view (including the oracle itself) for A that is consistent with the information contained in the plain-model protocol augmented transcript (i.e., the transcript of the oracle protocol and the oracle calls), and then acts as A would.

The fact that $\tilde{\mathsf{A}}$ perfectly emulates A (and that $\tilde{\mathsf{B}}$ perfectly emulates B) trivially holds for information theoretic reasons. For the same reason, it also holds that the transcript generated by applying Map^f to a random transcript of π^f, where $f \leftarrow \mathcal{F}_{\mathrm{AF}}$, generates *exactly* the same transcript as a random execution of $\tilde{\pi}$ does (actually, the above facts hold for any definition of Finder[5] and for any function family). The interesting part is arguing that the *joint output* of the plain model protocol has similar distribution to that of the oracle aided protocol. To see that this is not trivial, assume that in the last round both oracle parties make the *same* oracle query q and output the query/answer pair $(q, f(q))$. If it happens that $(q, \cdot) \notin \mathcal{I}$, where $\mathcal{I} = \mathsf{Finder}(\bar{t})$ is the query/answer pairs made by the final call to Finder on transcript $\bar{t}$, then the answer that each of the plain-model parties compute for the query q might be different. In this case, the joint output of the plain model protocol does not look like the joint output of the oracle protocol. Luckily, the guarantee of Finder yields that the above scenario is unlikely to happen; with high probability $\mathcal{I}$ contains *all* common queries the two parties made, yielding that the joint output of the plain-model protocol has similar distribution to that of the oracle protocol. It turns out that the above example generalizes to any possible protocol, showing that the above mapping and plain-model protocol are indeed the desired ones.

1.3 Related Work

In their seminal work, Impagliazzo and Rudich [13] showed that there are no key-agreement protocols in the random-function model, and deduce that key-agreement protocols cannot be black-box reduced to one-way functions. This result was later improved by Barak and Mahmoody [1], showing there are no ℓ-query key-agreement protocols in the random-function model, secure against adversary making $O(\ell^2)$ queries. Thus, matching the upper bound of Merkle [18]. On the technical level, Impagliazzo and Rudich [13] implemented a dependency finder for the all-function family (see Section 1.2), and then showed that no key-agreement protocol exists relative to a function family that has such a dependency finder. The latter observation was the starting point of the research we describe here.

Parts of the proof of our main technical lemma (Lemma 16), and in particular the first part of the proof of Lemma 22, follow the footsteps of [1] (which in turn, follows those of [13]). Yet, the proof we give for these parts presents several improvements over those of [13, 1]. First, our proof holds with respect to a large set of function families, and not only for the all-function family considered in [13, 1]. Second, our proof respects the round structure of the original protocol (the works of [13, 1] first transform the protocol in hand, into a normal form; one round per query, which induces a blow up in the protocol's round complexity). Finally, we find our proof more modular and with a simpler structure comparing to the previous proofs. In particular, it separates the generic parts of the proof from those building upon the specific properties of a function family in hand.

In an independent work, Mahmoody et al. [16] show that the all-function family (and thus one-way functions) are useless for *secure function evaluation* of deterministic, polynomial input-domain,

[5] Whose output contains all queries made to the oracle.

two-party functionalities. In other words, deterministic, bounded input domain, functionalities that *can* be securely computed in the random-function model, are the *trivial* ones — functionalities that can be securely computed unconditionally. The comparison to the result stated here is that [16] handle *with input* functionalities, but *only deterministic with polynomial input domain*, where here we handle *input-less* functionalities, but *including randomized ones*. Putting the two results together, gives a partial characterization of the power of the random-function model for (semi-honest) two-party computation. It is still open, however, whether the random-function model is useful for securely computing randomized functionalities *with* inputs, or functionalities of super-polynomial input domain.

1.3.1 Additional Black-box Separations

Following [13], the method of black-box separation was subsequently used in many other works: [22] shows that there does not exist a black-box reduction from a k-pass secret key agreements to $(k-1)$-pass secret key agreements; [24] shows that there exist no black-box reductions from collision-free hash functions to one-way permutations; [14] shows that there exists no construction of one-way permutations based on one-way functions. Other works using this paradigm contain [15], [8], [9], [4], [10], [12] and [25].

1.3.2 Differential Privacy

Distributed differential privacy was considered by Beimel et al. [2], who studied the setting of multiparty differentially private computation (where an n-bit database is shared between n parties). They gave a separation between the information theoretic and computational differential privacy in the distributed setting. The notion of computational differential privacy was considered in Mironov et al. [19]. They presented several definitions of computational differential privacy, studied the relationships between these definitions, and constructed efficient two-party computational differentially private protocols for approximating the Hamming-distance between two vectors. Two-party differential privacy (where an n-bit database is shared between two parties) was considered by McGregor et al. [17]. They prove a lower-bound on the accuracy of two party differentially private protocols, in the information theoretic model, for computing the inner-product between two n-bit strings (and, consequently for protocols for computing the Hamming distance). Hence, proving a separation between information theoretic and computational two-party differentially private computation. In this paper, we extend the lower-bound of [17] to the random-oracle model.

1.4 Open Problems

As mentioned above, the main open problem is the full characterization of the power of the random-function model with respect to semi-honest adversaries. Specifically, is it possible to come up with a similar mapping from any (also with inputs) oracle-aided protocol to an equivalent one in the no-oracle model. Another interesting problem is to use our mapping (or a variant of it) to show that the random-function model is also useless for protocols (say, input-less) that are secure against fail-stop adversaries. An immediate implication of such a result would be that optimally-fair coin tossing are impossible to achieve in the random function model.[6]

[6] We mention that Dachman-Soled et al. [5] showed such an impossibility result for $O(n/\log n)$-round protocols, where n being the random function input length.

Paper Organization

Formal definitions are given in Section 2. We state our main result in Section 3, and prove the main technical lemma in Section 4. The different applications of our main result are given in Section 5.

2 Preliminaries

2.1 Notations

Let poly be the set of all polynomials, and let PPTM stand for probabilistic polynomial-time algorithm. A function $\mu\colon \mathbb{N} \to [0,1]$ is *negligible*, denoted $\mu(n) = \mathrm{neg}(n)$, if $\mu(n) = n^{-\omega(1)}$. For $m \in \mathbb{N}$, let $[m] = \{1, \ldots, m\}$. For a finite set $\mathcal{S}$, we let $\chi_{\mathcal{S}}$ stands for its characteristic function, and write $x \leftarrow \mathcal{S}$ to denote that x is selected according to the uniform distribution over $\mathcal{S}$. Similarly, for a random variable X, we write $x \leftarrow X$ to denote that x is chosen according to X. The support of a distribution D over a finite set $\mathcal{U}$, denoted $\mathrm{Supp}(D)$, is defined as $\{u \in \mathcal{U} : D(u) > 0\}$. The statistical distance between two distributions P and Q over a finite set $\mathcal{U}$, denoted $\mathrm{SD}(P,Q)$, is defined as $\frac{1}{2}\sum_{u\in\mathcal{U}} |P(u) - Q(u)|$, and is known to be equal to $\max_{\mathcal{S}\subseteq\mathcal{U}} (\Pr_P[\mathcal{S}] - \Pr_Q[\mathcal{S}])$. We say that two distributions P and Q are δ-close if $\mathrm{SD}(P,Q) \leq \delta$.

2.2 Interactive Protocols

A two-party protocol $\pi = (\mathsf{A}, \mathsf{B})$ (with no oracle access) is a pair of probabilistic interactive Turing machines. The communication between the Turing machines A and B is carried out in rounds, where in each round one of the parties is active and the other party is idle. In the j'th round of the protocol, the currently active party P acts according to its partial view, writing some value on its output tape, and then sending a message to the other party (i.e., writing the message on the common tape). A communication transcript $\bar{t}$ (i.e., the "transcript") is the list of messages exchanged between the parties in an execution of the protocol, where $\bar{t}_{1,\ldots,j}$ denotes the first j messages in $\bar{t}$. A view of a party contains its input, its random tape and the messages exchanged by the parties during the execution. Specifically, A's view is a tuple $v_{\mathsf{A}} = (i_{\mathsf{A}}, r_{\mathsf{A}}, \bar{t})$, where i_{A} is A's input, r_{A} are A's coins, and $\bar{t}$ is the transcript of the execution. We let $(v_{\mathsf{A}})_j$ denote the partial view of A in the first j rounds of the execution described by v_{A}, namely, $(v_{\mathsf{A}})_j = (i_{\mathsf{A}}, r_{\mathsf{A}}, \bar{t}_{1,\ldots,j})$; we define v_{B} analogously. We call $v = (v_{\mathsf{A}}, v_{\mathsf{B}})$ the *joint view* of A and B, and let $v_j = ((v_{\mathsf{A}})_j, (v_{\mathsf{B}})_j)$. Given a distribution (or a set) $\mathcal{D}$ on the joint views of A and B, we let $\mathcal{D}_{\mathsf{A}}$ be the projection of $\mathcal{D}$ on A's view (i.e., $\Pr_{\mathcal{D}_{\mathsf{A}}}[v_{\mathsf{A}}] = \Pr_{(v_{\mathsf{A}},\cdot)\leftarrow\mathcal{D}}[v_{\mathsf{A}}]$), and define $\mathcal{D}_{\mathsf{B}}$ analogously. Finally, we sometimes refer to a well structured tuple v as a "view" of π, even though v happens with zero probability. When we wish to stress that we consider a view that has non-zero probability, we call it a *valid* view.

We call π an m-round protocol, if for *every* possible random tapes for the parties, the number of rounds is *exactly* m. Given a joint view v (containing the views of both parties) of an execution of (A, B) and $\mathsf{P} \in \{\mathsf{A}, \mathsf{B}\}$, let v_{P} denote P's part in v and let $\mathsf{trans}(v)$ denote the communication transcript in v. For $j \in [m]$, let $\mathrm{out}^{\mathsf{P}}_j(v) = \mathrm{out}^{\mathsf{P}}_j(v_{\mathsf{P}})$ denote the output of the party P at the end of j'th round of v (i.e., the string written on P's output tape), where $\mathrm{out}^{\mathsf{P}}_j(v) = \mathrm{out}^{\mathsf{P}}_{j-1}(v)$, in case P is inactive in the j'th round of v.

We sometimes consider *stateless* protocols – the parties hold no state, and in each round act on the message received in the previous round with freshly sampled random coins. Throughout this paper we almost solely consider *no-private input* protocols – the parties' only input is the common

input (the only exception to that is in Section 5.2, additional required notations introduced therein). Given a no-input two-party protocol π, let $\langle\pi\rangle$ be the distribution over the joint views of the parties in a random execution of π.

2.2.1 Oracle-Aided Protocols

An oracle-aided, two-party protocol $\pi = (\mathsf{A}, \mathsf{B})$ is a pair of interactive Turing machines, where each party has an additional tape called the *oracle tape*; the Turing machine can make a query to the oracle by writing a string q on its tape. It then receives a string ans (denoting the answer for this query) on the oracle tape.

For simplicity, we only consider function families whose inputs and outputs are binary strings. For an oracle-aided, no-input, two-party protocol $\pi = (\mathsf{A}, \mathsf{B})$ and a function family $\mathcal{F}$, we let $\Omega^{\mathcal{F},\pi}$ be the set of all triplets $(r_{\mathsf{A}}, r_{\mathsf{B}}, f)$, where r_{A} and r_{B} are possible random coins for A and B, and $f \in \mathcal{F}$ (henceforth, we typically omit the superscript $(\mathcal{F}, \pi)$ from the notation, whenever their values are clear from the context). For $f \in \mathcal{F}$, the distribution $\left\langle \pi^f = (\mathsf{A}^f, \mathsf{B}^f) \right\rangle$, is defined analogously to $\langle\pi\rangle = \langle \mathsf{A}, \mathsf{B} \rangle$, i.e., it is the distribution over the joint views of parties in a random execution of π with access to f. Given some information inf about some element of Ω (e.g., a set of query/answer pairs, or a view), let $\Pr_\Omega[\text{inf}] = \Pr_{\omega \leftarrow \Omega}[\omega \text{ is consistent with inf}]$, and let $\Pr_{\Omega|\text{inf}'}[\text{inf}]$ be this probability conditioned that ω is consistent with inf$'$ (set to zero, in case $\Pr_\Omega[\text{inf}'] = 0$).

Given a (possibly partial) execution of π^f, the views of the parties contain additional lists of query/answer pairs made to the oracle throughout the execution of the protocol. Specifically, A's view is a tuple $v_{\mathsf{A}} = (r_{\mathsf{A}}, \bar{t}, f_{\mathsf{A}})$, where r_{A} are A's coins, $\bar{t}$ is the transcript of the execution, and f_{A} are the oracle answers to A's queries. By convention, the active party in round j first makes all its queries to the oracle for this round, and then writes a value to its output tape and send a message to the other party. We denote by $(f_{\mathsf{P}})_j$ the oracle answers to the queries that party P makes during the first j rounds. As above, we let $(v_{\mathsf{A}})_j$ denote the partial view of A in the first j rounds of the execution described by v_{A}, namely, $(v_{\mathsf{A}})_j = (r_{\mathsf{A}}, \bar{t}_{1,\ldots,j}, (f_{\mathsf{A}})_j)$. We define v_{B} analogously.

For $\omega \in \Omega$, we let view(ω) be the full view of the parties determined by ω. We say that a "view" v is *consistent* with $(\mathcal{F}, \pi)$, if $\Pr_{\Omega^{\mathcal{F},\pi}}[v] > 0$.

We assume without loss of generality that in each round, the active party sends exactly one message to the other party (furthermore, the party acting in the last round outputs a final message, after which none of the parties makes any action). Therefore, a partial transcript $\bar{t}$ of the protocol uniquely determines the length of the partial execution that generated it (i.e., the number of rounds of π played), which we denote by $|\bar{t}|$. We consider the following distributions.

Definition 5 ($\Omega(\bar{t}, \mathcal{I})$ and $\mathcal{VIEW}(\bar{t}, \mathcal{I})$)**.** *Given a partial transcript $\bar{t}$ and a set of query/answer pairs $\mathcal{I}$, let $\Omega(\bar{t}, \mathcal{I}) = \Omega^{\mathcal{F},\pi}(\bar{t}, \mathcal{I})$ be the set of all tuples $(r_{\mathsf{A}}, r_{\mathsf{B}}, f) \in \Omega = \Omega^{\mathcal{F},\pi}$, in which f is consistent with $\mathcal{I}$, and $\bar{t}$ is a prefix of the transcript induced by $\left\langle \mathsf{A}^f(r_{\mathsf{A}}), \mathsf{B}^f(r_{\mathsf{B}}) \right\rangle$. Given a set $\mathcal{P} \subseteq \Omega$, let $\Omega_{\mathcal{P}}(\bar{t}, \mathcal{I}) = \Omega(\bar{t}, \mathcal{I}) \cap \mathcal{P}$.*

Let $\mathcal{VIEW}(\bar{t}, \mathcal{I}) = \mathcal{VIEW}^{\mathcal{F},\pi}(\bar{t}, \mathcal{I})$ be the value of $\text{view}(\omega)_{|\bar{t}|}$ for $\omega \leftarrow \Omega(\bar{t}, \mathcal{I})$, and define $\mathcal{VIEW}^{\mathcal{F},\pi}_{\mathcal{P}}(\bar{t}, \mathcal{I})$ analogously.

We note that since we consider the uniform distribution over Ω, we have that for any partial transcript $\bar{t}$, set of query/answer pairs $\mathcal{I}$, set $\mathcal{P} \subseteq \Omega$, and information inf about some element of Ω it holds that $\Pr_{\Omega_{\mathcal{P}}(\bar{t},\mathcal{I})}[\text{inf}] = \Pr_{\Omega|\bar{t},\mathcal{I},\mathcal{P}}[\text{inf}]$.

3 Mapping Oracle-Aided Protocols to No-Oracle Protocols

In this section we prove our main result, a mapping from protocols in the random-oracle model to (inefficient) no-oracle protocols.

3.1 Dependent Views

In the following we fix an m-round oracle-aided protocol π and a function family $\mathcal{F}$. We would like to restrict $\mathcal{VIEW}(\bar{t},\mathcal{I})$ to those views for which $\mathcal{I}$ contains all the joint information of the parties about f. We start by formally defining what it means for $\mathcal{I}$ to contain all the joint information.

Definition 6. *Let v_{A} be a j_{A}-round view for A and v_{B} be a j_{B}-round view for B, for some $j_{\mathsf{A}}, j_{\mathsf{B}} \in [m]$. For $i \in [j_{\mathsf{A}}]$, let ${\mathcal{I}_{\mathsf{A}}}^i$ be the set of query/answer pairs that A makes in the i'th round of v_{A} (where ${\mathcal{I}_{\mathsf{A}}}^i = \emptyset$, if A is idle in round i), and define ${\mathcal{I}_{\mathsf{B}}}^i$ analogously. Given a set $\mathcal{I}$ of query/answer pairs, we define*

1. $\alpha^{\mathcal{I}}_{v_{\mathsf{A}}} = \prod_{i\in[j_{\mathsf{A}}]} \Pr_{\Omega \mid \mathcal{I},{\mathcal{I}_{\mathsf{A}}}^1,\ldots,{\mathcal{I}_{\mathsf{A}}}^{i-1}}\left[{\mathcal{I}_{\mathsf{A}}}^i\right]$ *and*
2. $\alpha^{\mathcal{I}}_{v_{\mathsf{A}}|v_{\mathsf{B}}} = \prod_{i\in[j_{\mathsf{A}}]} \Pr_{\Omega \mid \mathcal{I},{\mathcal{I}_{\mathsf{A}}}^1,{\mathcal{I}_{\mathsf{B}}}^1,\ldots,{\mathcal{I}_{\mathsf{A}}}^{i-1},{\mathcal{I}_{\mathsf{B}}}^{i-1}}\left[{\mathcal{I}_{\mathsf{A}}}^i\right]$,

and define $\alpha^{\mathcal{I}}_{v_{\mathsf{B}}|v_{\mathsf{A}}}$ and $\alpha^{\mathcal{I}}_{v_{\mathsf{B}}}$ analogously.

Intuitively, $\alpha^{\mathcal{I}}_{v_{\mathsf{A}}}$ is the probability of A's view of f given $\mathcal{I}$, and $\alpha^{\mathcal{I}}_{v_{\mathsf{A}}|v_{\mathsf{B}}}$ is this probability when conditioning also on B's view. We will focus on those views with $\alpha^{\mathcal{I}}_{v_{\mathsf{A}}} = \alpha^{\mathcal{I}}_{v_{\mathsf{A}}|v_{\mathsf{B}}}$ and $\alpha^{\mathcal{I}}_{v_{\mathsf{B}}} = \alpha^{\mathcal{I}}_{v_{\mathsf{B}}|v_{\mathsf{A}}}$.

Definition 7 (dependent views)**.** *Let $v = (v_{\mathsf{A}}, v_{\mathsf{B}})$ be a pair of (possibly partial) valid views.*[7] *We say that v_{A} and v_{B} are* dependent *with respect to a set of query/answer pairs $\mathcal{I}$ and a function family $\mathcal{F}$, denoted* $\mathrm{Dependent}^{\mathcal{F}}_{\mathcal{I}}(v) = 1$*, if $\alpha^{\mathcal{I}}_{v_{\mathsf{A}}} \neq \alpha^{\mathcal{I}}_{v_{\mathsf{A}}|v_{\mathsf{B}}}$ or $\alpha^{\mathcal{I}}_{v_{\mathsf{B}}} \neq \alpha^{\mathcal{I}}_{v_{\mathsf{B}}|v_{\mathsf{A}}}$.*[8]

A pair of views $v = (v_{\mathsf{A}}, v_{\mathsf{B}})$ with $\mathrm{Dependent}_{\mathcal{I}}(v) = 0$ *is called* independent*. We let* $\mathrm{Ind}^{\mathcal{F},\pi}(\bar{t},\mathcal{I}) = \{\omega \in \Omega(\bar{t},\mathcal{I})\colon \mathrm{Dependent}^{\mathcal{F}}_{\mathcal{I}}(\mathrm{view}(\omega)_{|\bar{t}|}) = 0\}$ *and let $\mathcal{VIEW}^{\mathcal{F},\pi}_{\mathrm{Ind}}(y)$ stand for $\mathcal{VIEW}^{\mathcal{F},\pi}_{\mathrm{Ind}^{\mathcal{F},\pi}(y)}(y)$.*

The following observation (generalizing a similar observation made in [1]) plays a crucial role in the proof of our main result (stated in Section 3.3). It shows how to express the probability of a given view v, using $\alpha^{\mathcal{I}}_{v_{\mathsf{A}}|v_{\mathsf{B}}}$ and $\alpha^{\mathcal{I}}_{v_{\mathsf{B}}|v_{\mathsf{A}}}$. In particular, it implies that for an *independent* pair of views $v = (v_{\mathsf{B}}, v_{\mathsf{A}})$ and any set $\mathcal{P} \subseteq \Omega$, the probability that $\mathcal{VIEW}_{\mathcal{P}}(\bar{t},\mathcal{I}) = v$ can be written as a product of a term determined solely by v_{A} and $(\bar{t},\mathcal{I},\mathcal{P})$, and a term determined solely by v_{B} and $(\bar{t},\mathcal{I},\mathcal{P})$.

Proposition 8. *Let $\bar{t}$ be a transcript, let $\mathcal{I}$ be a list of query/answer pairs, and let $\mathcal{P} \subseteq \Omega$. Then, for every view $v = (r_{\mathsf{A}}, r_{\mathsf{B}}, \cdot) \in \mathrm{Supp}\left(\mathcal{VIEW}\left(\bar{t},\mathcal{I}\right)\right)$ with $\Pr_{\Omega}[v,\mathcal{I},\bar{t},\mathcal{P}] = \Pr_{\Omega}[v,\mathcal{I}]$, it holds that*

$$\Pr[\mathcal{VIEW}_{\mathcal{P}}(\bar{t},\mathcal{I}) = v] \triangleq \Pr_{\Omega_{\mathcal{P}}(\bar{t},\mathcal{I})}[v] = \frac{\Pr_{\Omega}[r_{\mathsf{A}}, r_{\mathsf{B}}] \cdot \alpha^{\mathcal{I}}_{v_{\mathsf{A}}|v_{\mathsf{B}}} \cdot \alpha^{\mathcal{I}}_{v_{\mathsf{B}}|v_{\mathsf{A}}}}{\Pr_{\Omega|\mathcal{I}}[\bar{t},\mathcal{P}]}.$$

[7] While properly defined for any pair of views $(v_{\mathsf{A}}, v_{\mathsf{B}})$, we will typically only consider the following notions for pairs with $\mathsf{trans}(v_{\mathsf{A}}) = \mathsf{trans}(v_{\mathsf{B}})$ (i.e., both views induce the same transcript).

[8] One can verify that $\mathrm{Dependent}^{\mathcal{F}}_{\mathcal{I}}(v) = 1$, in case v is inconsistent with $\mathcal{F}$, namely, $\Pr_{\mathcal{F}}[\mathcal{I}_{\mathsf{A}}, \mathcal{I}_{\mathsf{B}}] = 0$, where $\mathcal{I}_{\mathsf{A}}$ and $\mathcal{I}_{\mathsf{B}}$, are the lists of query/answer pairs appear in v_{A} and v_{B} respectively.

Proof. Note that

$$\Pr_{\Omega_{\mathcal{P}}(\bar{t},\mathcal{I})}[v] = \frac{\Pr_{\Omega}[v,\bar{t},\mathcal{I},\mathcal{P}]}{\Pr_{\Omega}[\bar{t},\mathcal{I},\mathcal{P}]} = \frac{\Pr_{\Omega}[v,\mathcal{I}]}{\Pr_{\Omega}[\bar{t},\mathcal{I},\mathcal{P}]} = \frac{\Pr_{\Omega}[r_{\mathsf{A}},r_{\mathsf{B}},\mathcal{I}]\cdot\Pr_{\Omega|r_{\mathsf{A}},r_{\mathsf{B}},\mathcal{I}}[v]}{\Pr_{\Omega}[\bar{t},\mathcal{I},\mathcal{P}]},$$

where the second equality holds since by assumption, $\Pr_{\Omega}[v,\mathcal{P},\mathcal{I},\bar{t}] = \Pr_{\Omega}[v,\mathcal{I}]$. Since the choice of random coins is independent of the choice of f, we can write

$$\Pr_{\Omega_{\mathcal{P}}(\bar{t},\mathcal{I})}[v] = \frac{\Pr_{\Omega}[r_{\mathsf{A}},r_{\mathsf{B}}]\cdot\Pr_{\Omega}[\mathcal{I}]\cdot\Pr_{\Omega\,|\,r_{\mathsf{A}},r_{\mathsf{B}},\mathcal{I}}[v]}{\Pr_{\Omega}[\mathcal{I}]\cdot\Pr_{\Omega\,|\,\mathcal{I}}[\bar{t},\mathcal{P}]} = \frac{\Pr_{\Omega}[r_{\mathsf{A}},r_{\mathsf{B}}]\cdot\Pr_{\Omega\,|\,r_{\mathsf{A}},r_{\mathsf{B}},\mathcal{I}}[v]}{\Pr_{\Omega\,|\,\mathcal{I}}[\bar{t},\mathcal{P}]} \tag{1}$$

We next analyze the term $\Pr_{\Omega|r_{\mathsf{A}},r_{\mathsf{B}},\mathcal{I}}[\mathcal{I}_{\mathsf{A}},\mathcal{I}_{\mathsf{B}},\bar{t}]$, where $\mathcal{I}_{\mathsf{A}}$ is the set of query/answer pairs that A sees according to v_{A} ($\mathcal{I}_{\mathsf{B}}$ is defined analogously). Note that this term is equal to $\Pr_{\Omega|r_{\mathsf{A}},r_{\mathsf{B}},\mathcal{I}}[v]$, since given $(r_{\mathsf{A}},r_{\mathsf{B}},\mathcal{I})$, the values of $(\mathcal{I}_{\mathsf{A}},\mathcal{I}_{\mathsf{B}},\bar{t})$ and v are implied by each other.

Let j be the number of rounds in v, and for $i\in[j]$ recall that ${\mathcal{I}_{\mathsf{A}}}^i$ is the set of query/answer pairs that A sees in the i'th round of the execution according to v_{A} (${\mathcal{I}_{\mathsf{B}}}^i$ is defined analogously). Since at any point through the execution of π^f, the next query of the acting party is determined by its partial view, it follows that

$$\begin{aligned}
\Pr_{\Omega|r_{\mathsf{A}},r_{\mathsf{B}},\mathcal{I}}[\mathcal{I}_{\mathsf{A}},\mathcal{I}_{\mathsf{B}},\bar{t}] &= \prod_{i\in[j]} \Pr_{\Omega|r_{\mathsf{A}},r_{\mathsf{B}},\mathcal{I},{\mathcal{I}_{\mathsf{A}}}^1,{\mathcal{I}_{\mathsf{B}}}^1,\ldots,{\mathcal{I}_{\mathsf{A}}}^{i-1},{\mathcal{I}_{\mathsf{B}}}^{i-1},t_1,\ldots,t_{i-1}} \left[{\mathcal{I}_{\mathsf{A}}}^i,{\mathcal{I}_{\mathsf{B}}}^i,t_i\right] \\
&= \prod_{i\in[j]} \Pr_{\Omega|r_{\mathsf{A}},r_{\mathsf{B}},\mathcal{I},{\mathcal{I}_{\mathsf{A}}}^1,{\mathcal{I}_{\mathsf{B}}}^1,\ldots,{\mathcal{I}_{\mathsf{A}}}^{i-1},{\mathcal{I}_{\mathsf{B}}}^{i-1},t_1,\ldots,t_{i-1}} \left[{\mathcal{I}_{\mathsf{A}}}^i,{\mathcal{I}_{\mathsf{B}}}^i\right] \\
&= \prod_{i\in[j]} \Pr_{\Omega|\mathcal{I},{\mathcal{I}_{\mathsf{A}}}^1,{\mathcal{I}_{\mathsf{B}}}^1,\ldots,{\mathcal{I}_{\mathsf{A}}}^{i-1},{\mathcal{I}_{\mathsf{B}}}^{i-1}} \left[{\mathcal{I}_{\mathsf{A}}}^i,{\mathcal{I}_{\mathsf{B}}}^i\right] \\
&= \left(\prod_{i\in[j]} \Pr_{\Omega|\mathcal{I},{\mathcal{I}_{\mathsf{A}}}^1,{\mathcal{I}_{\mathsf{B}}}^1,\ldots,{\mathcal{I}_{\mathsf{A}}}^{i-1},{\mathcal{I}_{\mathsf{B}}}^{i-1}} \left[{\mathcal{I}_{\mathsf{A}}}^i\right]\right)\cdot\left(\prod_{i\in[j]} \Pr_{\Omega|\mathcal{I},{\mathcal{I}_{\mathsf{A}}}^1,{\mathcal{I}_{\mathsf{B}}}^1,\ldots,{\mathcal{I}_{\mathsf{A}}}^{i-1},{\mathcal{I}_{\mathsf{B}}}^{i-1}} \left[{\mathcal{I}_{\mathsf{B}}}^i\right]\right) \\
&= \alpha^{\mathcal{I}}_{v_{\mathsf{A}}|v_{\mathsf{B}}}\cdot\alpha^{\mathcal{I}}_{v_{\mathsf{B}}|v_{\mathsf{A}}},
\end{aligned}$$

where the second equation holds since the i'th message is (deterministically) determined by the randomness of the parties, the oracle answers and the transcript till now. The third one holds since the distribution on the oracle answers at each point during the execution is a function of $\mathcal{I}$ and the previous queries made by the parties (recall that $\Pr_{\Omega|\inf'}\left[{\mathcal{I}_{\mathsf{A}}}^i,{\mathcal{I}_{\mathsf{B}}}^i\right] = \Pr_{\omega\leftarrow\Omega|\inf'}[\omega \text{ is consistent with } {\mathcal{I}_{\mathsf{A}}}^i,{\mathcal{I}_{\mathsf{B}}}^i]$ and that the first $i-1$ messages are determined by the randomness of the parties and the oracle answers to the queries made till round $i-1$). Finally, the fourth one holds since only one party is active in each round (hence, for every i either ${\mathcal{I}_{\mathsf{A}}}^i$ or ${\mathcal{I}_{\mathsf{B}}}^i$ is empty). □

3.2 Intersecting Views

A special case of dependent views is when the two paries share a *common* oracle query not in $\mathcal{I}$.

Definition 9 (intersecting views)**.** *A (possibly partial) pair of views $v=(v_{\mathsf{A}},v_{\mathsf{B}})$ are* intersecting *with respect to a set of query/answer pairs $\mathcal{I}$, denoted* $\text{Intersect}_{\mathcal{I}}(v)=1$*, if v_{A} and v_{B} share a common query q not in $\mathcal{I}$ (i.e., $(q,\cdot)\notin\mathcal{I}$).*

For most function families, an intersection implies being dependent (with respect to the same list of query/answer pairs). In this paper we limit our attention to "simple" function families for which also the other direction holds, namely dependency implies intersection.

Definition 10 (simple function families)**.** *A function family $\mathcal{F}$ is* simple*, if for any oracle-aided protocol π, list $\mathcal{I}$ of query/answer pairs that is consistent with some $f \in \mathcal{F}$, and a (possibly partial) pair of views $v = (v_{\mathsf{A}}, v_{\mathsf{B}})$ consistent with $\mathcal{I}$, it holds that* $\text{Dependent}^{\mathcal{F}}_{\mathcal{I}}(v) = 1$ *iff* $\text{Intersect}_{\mathcal{I}}(v) = 1$.

Example 11. *It is not hard to verify (see Section 5.3) that the "all-function family" (i.e., the set of all functions from $\{0,1\}^n$ to $\{0,1\}^n$) is simple.*

3.3 Oracle-Aided to No-Oracle Protocol Mapping

The following theorem shows that an execution of an oracle-aided protocol with oracle access to a random $f \in \mathcal{F}$, where $\mathcal{F}$ is a simple function family, can be mapped to an execution of a related protocol with no oracle access. In Section 5 we use this result to prove limitations on the power of oracle-aided protocols in achieving specific cryptographic tasks.

Definition 12 (oracle-aided to no-oracle mapping)**.** *A pair of a function family $\mathcal{F}$ and a no-input, m-round, oracle-aided protocol $\pi = (\mathsf{A}, \mathsf{B})$, has a* (T, ε)-mapping*, if there exists a deterministic, oracle-aided T-query algorithm* Map *and a stateless, m-round, no-input (and no-oracle) protocol $(\tilde{\mathsf{A}}, \tilde{\mathsf{B}})$, such that the following hold:*

1. $\text{SD}\left(\mathcal{D}_{\mathcal{F}}, \mathcal{D}_P\right) \leq \varepsilon$ *for every $j \in [m]$, where*

$$\mathcal{D}_{\mathcal{F}} = \left(\text{out}^{\mathsf{A}}_j(v), \text{out}^{\mathsf{B}}_j(v), \mathsf{Map}^f(\mathsf{trans}(v)_{1,\ldots,j})\right)_{f \leftarrow \mathcal{F}, v \leftarrow \langle \mathsf{A}^f, \mathsf{B}^f \rangle} \qquad \textit{and,}$$

$$\mathcal{D}_P = \left(\text{out}^{\tilde{\mathsf{A}}}_j(v), \text{out}^{\tilde{\mathsf{B}}}_j(v), \mathsf{trans}(v)_{1,\ldots,j}\right)_{v \leftarrow \langle \tilde{\mathsf{A}}, \tilde{\mathsf{B}} \rangle}.$$ [9]

   Furthermore, $\mathcal{D}_P[1,3] \equiv \mathcal{D}_{\mathcal{F}}[1,3]$ and $\mathcal{D}_P[2,3] \equiv \mathcal{D}_{\mathcal{F}}[2,3]$. [10]

2. *For every $f \in \mathcal{F}$, an m-round transcript $\bar{t}$ and $j \in [m]$, it holds that $\mathsf{Map}^f(\bar{t}_{1,\ldots,j}) = \mathsf{Map}^f(\bar{t})_{1,\ldots,j}$. Furthermore, the oracle calls made in $\mathsf{Map}^f(\bar{t}_{1,\ldots,j})$ are a subset of those made in $\mathsf{Map}^f(\bar{t})$.*

Theorem 13. *Let $\mathcal{F}$ be a simple function family and let $\pi = (\mathsf{A}, \mathsf{B})$ be an ℓ-query, oracle-aided, no-input protocol, then $(\mathcal{F}, \pi)$ has an $(256 \cdot \ell^2/\varepsilon^2, \varepsilon)$-mapping for any $0 < \varepsilon \leq 1$.*

Remark 14 (Round complexity of the no-oracle protocol)**.** *The proof of Theorem 13 can be easily modified to yield a one-message no-oracle protocol (in this case, $\mathcal{D}_{\mathcal{F}}$ and $\mathcal{D}_P$ should be modified to reflect the transcript and outputs at the end of the executions). The roles of $\tilde{\mathsf{A}}$ and $\tilde{\mathsf{B}}$ in the resulting protocol however, cannot reflect as closely the roles of A and B, as done in the many-round, no-oracle protocol stated above.*

The heart of the proof of Theorem 13 lies in the following lemma, proof of which is given in Section 4.

[10] I.e., the projections of $\mathcal{D}_P$ and $\mathcal{D}_{\mathcal{F}}$ to their transcript part and the output of one of the parties, are identically distributed.

Definition 15 (DependencyFinder)**.** *Let $\mathcal{F}$ be a function family and let $\pi = (\mathsf{A}, \mathsf{B})$ be an m-round oracle-aided protocol. A deterministic oracle-aided algorithm* Finder *is a (T, ε)-*DependencyFinder *for $(\mathcal{F}, \pi)$ if the following holds for any $j \in [m]$: consider the following random process $\mathsf{CF} = \mathsf{CF}(\mathcal{F}, \pi, \mathsf{Finder})$:*

1. *Choose $(r_{\mathsf{A}}, r_{\mathsf{B}}, f) \leftarrow \Omega^{\mathcal{F},\pi}$ and let $\bar{t}$ be the j-round transcript of π induced by $(r_{\mathsf{A}}, r_{\mathsf{B}}, f)$.*
2. *For $i = 1$ to j set $\mathcal{I}_i = \mathcal{I}_{i-1} \cup \mathsf{Finder}^f(\bar{t}_{1,\ldots,i}, \mathcal{I}_{i-1})$ (letting $\mathcal{I}_0 = \emptyset$), where $\mathsf{Finder}^f(x)$ is the set of queries/answers made by $\mathsf{Finder}^f(x)$ to f.*
3. *Output $\left(\bar{t}, \mathcal{I}_j\right)$.*

Then

1. $\mathrm{E}_{d \leftarrow \mathsf{CF}}\left[\mathrm{SD}\left(\mathcal{VIEW}^{\mathcal{F},\pi}(d), (\mathcal{VIEW}^{\mathcal{F},\pi}(d)_{\mathsf{A}}, \mathcal{VIEW}^{\mathcal{F},\pi}(d)_{\mathsf{B}})\right)\right] \leq \varepsilon$, *and*
2. $\Pr[\#$ *of f-calls made in* $\mathsf{CF} > T] \leq \varepsilon$.

That is, conditioned on a random transcript of $\pi^{\mathcal{F}}$ and the oracle queries made by a (T, δ)-DependencyFinder, the parties' views are close to being in a product distribution.

Lemma 16. *Let $\mathcal{F}$ be a simple function family and let $\pi = (\mathsf{A}, \mathsf{B})$ be an ℓ-query oracle-aided protocol, then $(\mathcal{F}, \pi)$ has a $(64/\delta^2, \ell\delta)$-*DependencyFinder *for any $0 < \delta \leq 1/\ell$.*

Before proving Lemma 16, we first use it to prove Theorem 13.

3.3.1 Proving Theorem 13

Fix a simple function family $\mathcal{F}$ and a no-input, m-round, ℓ-query oracle-aided protocol π. Fix $0 < \varepsilon \leq 1$ and let Finder be the $(T = 256 \cdot \ell^2/\varepsilon^2, \varepsilon/2)$-DependencyFinder guaranteed by Lemma 16 for $(\mathcal{F}, \pi)$ (taking $\delta = \varepsilon/2\ell$). We start by defining the mapping algorithm and then we define a protocol with no oracle access.

Algorithm 17 (Map)**.**

Oracle: $f \in \mathcal{F}$.

Input: j-round transcript $\bar{t}$ of π.

Operation:

1. *For $i = 1$ to j set $\mathcal{I}_i = \mathcal{I}_{i-1} \cup \mathsf{Finder}^f(\bar{t}_{1,\ldots,i}, \mathcal{I}_{i-1})$ (letting $\mathcal{I}_0 = \emptyset$).*
2. *If in some round i^* the* overall *number of f calls (made by* Finder*) is T, halt the above procedure and set $\mathcal{I}_{i^*}$ to be the set of T query/answer pairs obtained so far,*[11] *and set $\mathcal{I}_i = \mathcal{I}_{i^*}$ for all $i^* < i \leq j$.*
3. *Output $\left(\bar{t}_1, \mathcal{I}_1\right), \left(\bar{t}_{1,2}, \mathcal{I}_2\right), \ldots, \left(\bar{t}, \mathcal{I}_j\right)$.*

[11] I.e., augmenting $\mathcal{I}_{i^*-1}$ with the queries/answers made in round i^* before halting.

The no-oracle protocol. Our stateless, no-oracle protocol $\tilde{\pi} = (\tilde{\mathsf{A}}, \tilde{\mathsf{B}})$, emulates the oracle-aided protocol π by keeping the "important" oracle queries as part of the transcript, and selecting the rest of the oracle at random (independently in each round). In particular, $\tilde{\mathsf{A}}$ is active in $\tilde{\pi}$ in the same rounds that A is in π (same for $\tilde{\mathsf{B}}$ and B). The definition of $\tilde{\mathsf{A}}$ is given below ($\tilde{\mathsf{B}}$ is analogously defined).

Algorithm 18 ($\tilde{\mathsf{A}}$)**.**

Input: A pair $(\bar{t}, \mathcal{I})$, where $\bar{t}$ is a transcript of length j and $\mathcal{I}$ is a set of query/answer pairs.

Operation:

1. *Sample $(r_{\mathsf{A}}, r_{\mathsf{B}}, f) \leftarrow \Omega(\bar{t}, \mathcal{I})$, and let out_{j+1} and t_{j+1} denote A's output and message respectively, in the $(j+1)$ round of $\langle \mathsf{A}^f(r_{\mathsf{A}}), \mathsf{B}^f(r_{\mathsf{B}}) \rangle$.*
2. *Output out_{j+1}.*
3. *Compute the value of $\mathcal{I}_{j+1}$ output by $\mathsf{Map}^f(\overline{t_{j+1}})$ for $\overline{t_{j+1}} = (\bar{t}, t_{j+1})$.*
4. *Send $(\overline{t_{j+1}}, \mathcal{I}_{j+1})$ to $\tilde{\mathsf{B}}$.*

Proof of Theorem 13. We prove that algorithm Map (17) and protocol $\tilde{\pi} = (\tilde{\mathsf{A}}, \tilde{\mathsf{B}})$ (18), form an (T, ε)-mapping for $(\mathcal{F}, \pi)$. By construction, algorithm Map is deterministic (since Finder is deterministic), makes at most T queries and fulfills the second item of Definition 12. Towards showing that $(\mathsf{Map}, \tilde{\pi})$ fulfills also the first property of Definition 12 with respect to stated parameter, we prove the following claim:

Claim 19. $\left(\mathsf{Map}^f(\mathsf{trans}(v)_{1,\ldots,j})\right)_{f \leftarrow \mathcal{F}, v \leftarrow \langle \mathsf{A}^f, \mathsf{B}^f \rangle} \equiv (\mathsf{trans}(v)_{1,\ldots,j})_{v \leftarrow \langle \tilde{\mathsf{A}}, \tilde{\mathsf{B}} \rangle}$ *for every $j \in [m]$.*

Proof. We prove the claim by induction on j; by the induction hypothesis and the fact that $\mathsf{Map}^f(\mathsf{trans}(v)_{1,\ldots,j})_{1,\ldots,j-1} = \mathsf{Map}^f(\mathsf{trans}(v)_{1,\ldots,j-1})$ (since Map fulfills the second item of Definition 12), it suffices to prove that the distributions in the claim are the same, conditioned that their $(j-1)$-"round" prefix is fixed to some value $\left(\ldots, \left(\bar{t}_{1,\ldots,j-1}, \mathcal{I}_{j-1}\right)\right)$. Since $\mathcal{I}_{j-1}$ is the set of queries/answers made by $\mathsf{Map}^f(\mathsf{trans}(v)_{1,\ldots,j-1})$ to f, the value of the right-hand-side distribution under this conditioning is $\mathsf{Map}^f(\bar{t}')$, where f and $\bar{t}'$ are the function and the j-round transcript of π respectively, induced by $\omega \leftarrow \Omega\left(\bar{t}_{1,\ldots,j-1}, \mathcal{I}_{j-1}\right)$. It is easy to verify that, under this conditioning, the latter process also describes the left-hand-side distribution. □

We next note that Claim 19 yields that $\mathcal{D}_P[1,3] \equiv \mathcal{D}_{\mathcal{F}}[1,3]$ (and similarly that $\mathcal{D}_P[2,3] \equiv \mathcal{D}_{\mathcal{F}}[2,3]$); indeed, conditioned on $\mathcal{D}_P[3] = \mathcal{D}_{\mathcal{F}}[3] = \left(\ldots, \left(\bar{t}_{1,\ldots,j}, \mathcal{I}_j\right)\right)$, the values of both $\mathcal{D}_P[1]$ and of $\mathcal{D}_{\mathcal{F}}[1]$ (i.e., A's output), are obtained by the following random process: sample $\omega \leftarrow \Omega\left(\bar{t}_{1,\ldots,j}, \mathcal{I}_j\right)$ and output A's output in the j'th round induced by ω.

Finally, the definition of $\tilde{\pi} = (\tilde{\mathsf{A}}, \tilde{\mathsf{B}})$ yields that

$$\begin{aligned} \mathcal{D}_P &:= \left(\mathrm{out}_j^{\tilde{\mathsf{A}}}(v), \mathrm{out}_j^{\tilde{\mathsf{B}}}(v), \mathsf{trans}(v)_{1,\ldots,j}\right)_{v \leftarrow \langle \tilde{\mathsf{A}}, \tilde{\mathsf{B}} \rangle} \\ &\equiv \left(\mathrm{out}_j^{\mathsf{A}}(v_{\mathsf{A}}), \mathrm{out}_j^{\mathsf{B}}(v_{\mathsf{B}}), \tilde{t}_{1,\ldots,j}\right)_{\tilde{t} \leftarrow \mathsf{trans}(\langle \tilde{\mathsf{A}}, \tilde{\mathsf{B}} \rangle), v_{\mathsf{A}} \leftarrow \mathcal{VIEW}(\tilde{t}_j)_{\mathsf{A}}, v_{\mathsf{B}} \leftarrow \mathcal{VIEW}(\tilde{t}_j)_{\mathsf{A}}} \end{aligned} \tag{2}$$

where we recall that $\tilde{t}_j$ consists of a pair $(\overline{t_j}, \mathcal{I}_j)$. It is easy to verify that

$$\begin{aligned}\mathcal{D}_{\mathcal{F}} &:= \left(\mathrm{out}_j^{\mathsf{A}}(v), \mathrm{out}_j^{\mathsf{B}}(v), \mathsf{Map}^f(\mathsf{trans}(v)_{1,\ldots,j})\right)_{f\leftarrow\mathcal{F}, v\leftarrow\langle \mathsf{A}^f, \mathsf{B}^f\rangle} \\ &\equiv \left(\mathrm{out}_j^{\mathsf{A}}(v_{\mathsf{A}}), \mathrm{out}_j^{\mathsf{B}}(v_{\mathsf{B}}), \tilde{t}_{1,\ldots,j}\right)_{f\leftarrow\mathcal{F}, \bar{t}\leftarrow\mathsf{trans}(\langle \mathsf{A}^f, \mathsf{B}^f\rangle), \tilde{t}\leftarrow\mathsf{Map}^f(\bar{t}), v\leftarrow\mathcal{VIEW}(\tilde{t}_j)}\end{aligned}$$

and therefore Claim 19 yields that

$$\mathcal{D}_{\mathcal{F}} \equiv \left(\mathrm{out}_j^{\mathsf{A}}(v_{\mathsf{A}}), \mathrm{out}_j^{\mathsf{B}}(v_{\mathsf{B}}), \tilde{t}_{1,\ldots,j}\right)_{\tilde{t}\leftarrow\mathsf{trans}(\langle \tilde{\mathsf{A}}, \tilde{\mathsf{B}}\rangle), v\leftarrow\mathcal{VIEW}(\tilde{t}_j)} \tag{3}$$

We conclude the proof using the fact that Finder is a $(T, \varepsilon/2)$-DependencyFinder for $(\mathcal{F}, \pi)$. The issue to note is that Process CF (described in Definition 15) may make arbitrary number of oracle queries, while Map is restricted to at most T queries. Let $\mathcal{S}$ be the set of pairs $d = \left(\bar{t}_{1,\ldots,j}, \mathcal{I}_j\right)$ in the support of the Process CF with $|\mathcal{I}_j| \leq T$. Note that the probability that CF outputs $d \in \mathcal{S}$, is exactly the probability of the transcript part being of the form $(\ldots, d)$ according to distribution $\mathcal{D}_{\mathcal{F}}$, where by Claim 19 this is also the probability of the this event according to $\mathcal{D}_P$. We bound the statistical distance between $\mathcal{D}_{\mathcal{F}}$ and $\mathcal{D}_P$, by separately bounding the part contributed by transcripts $(\ldots, d)$ with $d \in \mathcal{S}$ and by transcripts $(\ldots, d)$ with $d \notin \mathcal{S}$.

The fact that Finder is a $(T, \varepsilon/2)$-DependencyFinder for $(\mathcal{F}, \pi)$, yields a bound of $\varepsilon/2$ on the contribution of elements whose transcripts is *inside* $\mathcal{S}$ to the statistical distance between $\mathcal{D}_{\mathcal{F}}$ and $\mathcal{D}_P$. It also bounds by $\varepsilon/2$ the probability that CF outputs elements whose transcripts is outside $\mathcal{S}$; yielding the same bound on the contribution of such elements to the statistical distance between $\mathcal{D}_{\mathcal{F}}$ and $\mathcal{D}_P$. We conclude that $\mathrm{SD}\left(\mathcal{D}_{\mathcal{F}}, \mathcal{D}_P\right) \leq \varepsilon/2 + \varepsilon/2 = \varepsilon$.

□

4 Proving Lemma 16

We start by proving Lemma 16 for normal-form protocols, defined below. In Section 4.1 we extend the proof to arbitrary protocols. In the following, for a view v describing an execution of an oracle-aided protocol π, we let $\ell_i(v)$ be the number of queries made (by the non-idle party) in round i according to v.

Definition 20 (normal-form protocols)**.** *An oracle-aided protocol π is in* normal-form *if $\ell_i(v) \leq 1$ for every possible view v and every round i, that is, if a party makes* at most *a single query to the oracle in each communication round.*

Fix a normal-form, oracle-aided protocol π, a simple function family $\mathcal{F}$, and $0 < \delta \leq 1/\ell$. Define the following oracle-aided algorithm.

Algorithm 21 (Finder)**.**

Input: a transcript $\bar{t}$ and a list $\mathcal{I}$ of query/answer pairs.

Oracle: $f \in \mathcal{F}$.

Operation: While there exists a query q with $(q, \cdot) \notin \mathcal{I}$ and $p_q > \delta/32$, where p_q is the probability that q was asked either by A or B in a random sample from $\mathcal{VIEW}_{\mathrm{Ind}}^{\mathcal{F},\pi}(\bar{t}, \mathcal{I})$:[12]

[12] Recall that the distribution $\mathcal{VIEW}_{\mathrm{Ind}}^{\mathcal{F},\pi}(\bar{t}, \mathcal{I})$ stands for a random, independent joint view of π, which is consistent with the transcript $\bar{t}$ and the query/answer list $\mathcal{I}$.

Add $(q, f(q))$ to $\mathcal{I}$ (choose the lexicographically first q if there are more than one).

Namely, algorithm Finder considers executions of π that are consistent with the given partial transcript and a set of "known" query/answer pairs, in which the views of the parties are independent (since $\mathcal{F}$ is a simple family, it means that the parties share no *intersecting query* — a common query whose answer in not described in the given information). The goal of Finder is to find all "heavy queries": those queries that have substantial probability of being asked by one of the parties in a random such execution.

We prove Lemma 16 (for the case of normal-form protocols) by showing that Finder is a $(64/\delta^2, \ell\delta)$-$\mathsf{DependencyFinder}$ for $(\mathcal{F}, \pi)$. The heart of the proof is in the following lemma, yielding that in a random execution of $\mathsf{CF} = \mathsf{CF}(\mathcal{F}, \pi, \mathsf{Finder})$, the probability that the views of the parties "become dependent", in any given round, is small.

Lemma 22. *Let $\bar{t}$ be a (possibly partial) transcript of (the normal-form protocol) π, let $\mathcal{I}$ be a set of query/answer pairs, let $\mathrm{Ind} = \mathrm{Ind}^{\pi,\mathcal{F}}(\bar{t}, \mathcal{I})$, let $\mathcal{D} = \mathcal{VIEW}_{\mathrm{Ind}}(\bar{t}, \mathcal{I})$, and let $\delta > 0$ be such that $\Pr_{v \leftarrow \mathcal{D}}[q \in v \wedge (q, \cdot) \notin \mathcal{I}] \leq \delta \leq \frac{1}{4\ell}$ for every query q. Then, the following hold:*

1. *There exists a* product *distribution $\mathcal{C}$ over $\mathrm{Supp}(\mathcal{D}_{\mathsf{A}}) \times \mathrm{Supp}(\mathcal{D}_{\mathsf{B}})$ with $\mathrm{SD}(\mathcal{D}, \mathcal{C}) \leq 2\ell\delta$.*
2. *Let $\mathcal{D}^+$ be the value of $\mathrm{view}(\omega)_{|\bar{t}|+1}$ for $\omega \leftarrow \Omega_{\mathrm{Ind}}(\bar{t}, \mathcal{I})$, then $\Pr_{v \leftarrow \mathcal{D}^+}[\mathrm{Dependent}_{\mathcal{I}}(v)] \leq 4\delta \cdot \Pr_{v \leftarrow \mathcal{D}^+}\left[\ell_{|\bar{t}|+1}(v) = 1\right]$.*

We prove Lemma 22 in Section 4.2, but first use it for proving Lemma 16. We start by showing that Algorithm Finder (see 21) is a good $\mathsf{DependencyFinder}$ for $(\mathcal{F}, \pi)$, proving Lemma 16 for normal-form protocols.

Claim 23. *Algorithm Finder is a $(64/\delta^2, \ell\delta)$-$\mathsf{DependencyFinder}$ for $(\mathcal{F}, \pi)$.*

Proof. The following random variables are defined with respect to a random execution of $\mathsf{CF} = \mathsf{CF}(\mathcal{F}, \pi, \mathsf{Finder})$. Let $V = \mathrm{view}(\omega)_j$, i.e., the j long prefix of the full view implied by the triplet $\omega \in \Omega$ chosen in the first step of CF. We let $\overline{T}$ denote the (j-round) transcript in V. For $i \in [j]$ let I_i denote the value of $\mathcal{I}_i$ computed in CF, and for $2 \leq i \leq j$ let $\mathrm{FirstDep}_i = \mathrm{Dependent}_{I_{i-1}}(V_i) \wedge \neg\,\mathrm{Dependent}_{I_{i-2}}(V_{i-1})$ (where $I_0 = \emptyset$). We start by bounding the probability of $\mathrm{Dependent}_{I_j}(V)$, which yields the first property required for being a $(\cdot, \ell\delta)$-$\mathsf{DependencyFinder}$ for $(\mathcal{F}, \pi)$, and then complete the proof by bounding the number of queries made in CF.

Bounding the probability of dependencies. The following claim bounds the probability that a single "round" of CF causes dependencies.

Claim 24. $\Pr[\mathrm{FirstDep}_i] \leq \frac{\delta}{8} \cdot \mathrm{E}[\ell_i(V)]$ *for every* $2 \leq i \leq j$.

Proof. In the following we fix $2 \leq i \leq j$ and a value for $(\overline{T}_{1,\ldots,i-1}, I_{i-1})$, and prove (the slightly stronger fact) that the claim holds even under any such fixing. We next bound the probability that (under this fixing) the i'th round of π causes a collision. Recall that by Process CF, we obtained $I_{i-1} = I_{i-2} \cup \mathsf{Finder}^f(\overline{T}_{1,\ldots,i-1}, I_{i-2})$. The definition of Finder yields that

$$\Pr[q \in V_{i-1} \wedge (q, \cdot) \notin I_{i-1} \mid \neg\,\mathrm{Dependent}_{I_{i-1}}(V_{i-1})] \leq \delta/32 \tag{4}$$

for any query q. We can hence apply Lemma 22(2) with respect to the function family $\mathcal{F}$, protocol π, transcript $\overline{T}_{1,\ldots,i-1}$ and query/answer list I_{i-1}, yielding that

$$\Pr\left[\text{Dependent}_{I_{i-1}}(V_i) \mid \neg\,\text{Dependent}_{I_{i-1}}(V_{i-1})\right] \leq \frac{\delta}{8} \cdot \Pr\left[\ell_i(V) = 1 \mid \neg\,\text{Dependent}_{I_{i-1}}(V_{i-1})\right] \quad (5)$$

We conclude that

$$\begin{aligned}
&\Pr\left[\text{FirstDep}_i\right] \qquad (6)\\
&= \Pr[\text{Dependent}_{I_{i-1}}(V_i) \wedge \neg\,\text{Dependent}_{I_{i-2}}(V_{i-1})]\\
&\leq \Pr[\text{Dependent}_{I_{i-1}}(V_i) \wedge \neg\,\text{Dependent}_{I_{i-1}}(V_{i-1})]\\
&= \Pr[\neg\,\text{Dependent}_{I_{i-1}}(V_{i-1})] \cdot \Pr[\text{Dependent}_{I_{i-1}}(V_i) \mid \neg\,\text{Dependent}_{I_{i-1}}(V_{i-1})]\\
&\leq \Pr[\neg\,\text{Dependent}_{I_{i-1}}(V_{i-1})] \cdot \frac{\delta}{8} \cdot \Pr\left[\ell_i(V) = 1 \mid \neg\,\text{Dependent}_{I_{i-1}}(V_{i-1})\right]\\
&\leq \frac{\delta}{8} \cdot \mathrm{E}[\ell_i(V)].
\end{aligned}$$

The first inequality holds since $\mathcal{F}$ is a simple family (hence, $\text{Dependent}_{\mathcal{I}}(v) = 0$ implies $\text{Dependent}_{\mathcal{I}'}(v) = 0$, for every view v and lists $\mathcal{I}' \supseteq \mathcal{I}$) and the second one follows from Equation (5). Finally, since Equation (6) holds conditioned on any fixing of $(\overline{T}_{1,\ldots,i-1}, I_{i-1})$, it also holds without this conditioning and the claim follows. □

Continuing the proof, Claim 24 yields that

$$\begin{aligned}
\Pr\left[\text{Dependent}_{I_j}(V)\right] &\leq \Pr\left[\bigvee_{2\leq i\leq j} \text{FirstDep}_i\right] \leq \sum_{2\leq i\leq j} \frac{\delta}{8} \cdot \mathrm{E}\left[\ell_i(V)\right] \qquad (7)\\
&= \frac{\delta}{8}\,\mathrm{E}\left[\sum_{2\leq i\leq j} \ell_i(V)\right] \leq \frac{\ell\delta}{8}.
\end{aligned}$$

The first inequality holds since $\mathcal{F}$ is a simple family and since a first round view is never dependent (the inactive party made no queries). The third inequality holds since ℓ bounds the overall number of queries made in any execution of π. Since for any $d = (\bar{t}, \mathcal{I}) \in \text{Supp}(\overline{T}, I_j)$ it holds that $\text{SD}\left(\mathcal{VIEW}(d), \mathcal{VIEW}_{\text{Ind}}(d)\right) = \Pr_{\omega\leftarrow\Omega(d)}[w \notin \text{Ind}(d)] = \Pr_{v\leftarrow\mathcal{VIEW}(d)}[\text{Dependent}_{\mathcal{I}}(v)]$, it follows that

$$\begin{aligned}
\mathop{\mathrm{E}}_{d\leftarrow(\overline{T},I_j)}\left[\text{SD}\left(\mathcal{VIEW}(d), \mathcal{VIEW}_{\text{Ind}}(d)\right)\right] &= \mathop{\mathrm{E}}_{d\leftarrow(\overline{T},I_j)}\left[\Pr_{v\leftarrow\mathcal{VIEW}(d)}[\text{Dependent}_{\mathcal{I}}(v)]\right] \qquad (8)\\
&= \Pr[\text{Dependent}_{I_j}(V)]\\
&\leq \frac{\ell\delta}{8},
\end{aligned}$$

where the inequality hold by Equation (7). We complete the proof of this part by showing that $\mathcal{VIEW}_{\text{Ind}}(d)$ is close to some product distribution over the parties' views. The definition of Finder yields that

$$\Pr[q \in V \wedge (q,\cdot) \notin I_j \mid \neg\,\text{Dependent}_{I_j}(V), (\overline{T}, I_j) = d] \leq \frac{\ell\delta}{32} \quad (9)$$

for any possible query q and $d \in \mathrm{Supp}(\overline{T}, I_j)$. Therefore, Lemma 22(1) yields that

$$\mathrm{SD}\left(\mathcal{VIEW}_{\mathrm{Ind}}(d), \mathcal{C}(d)\right) \leq \frac{\ell\delta}{16}, \tag{10}$$

for any $d \in \mathrm{Supp}(\overline{T}, I_j)$, where $\mathcal{C}(d)$ is a *product* distribution over the parties' views. It follows (using the triangle inequality) that $\mathrm{SD}\left(\mathcal{VIEW}_{\mathrm{Ind}}(d), \left(\mathcal{VIEW}_{\mathrm{Ind}}(d)_{\mathsf{A}}, \mathcal{VIEW}_{\mathrm{Ind}}(d)_{\mathsf{B}}\right)\right) \leq \frac{3}{16} \cdot \ell\delta$ for any $d \in \mathrm{Supp}(\overline{T}, I_j)$, and therefore

$$\mathop{\mathrm{E}}_{d \leftarrow (\overline{T}, I_j)} \left[\mathrm{SD}\left(\mathcal{VIEW}_{\mathrm{Ind}}(d), \left(\mathcal{VIEW}_{\mathrm{Ind})}(d)_{\mathsf{A}}, \mathcal{VIEW}_{\mathrm{Ind})}(d)_{\mathsf{B}}\right)\right)\right] \leq \frac{3}{16} \cdot \ell\delta. \tag{11}$$

Combining Equations (8) and (11) and the triangle inequality, yields that

$$\mathop{\mathrm{E}}_{d \leftarrow (\overline{T}, I_j)} \left[\mathrm{SD}\left(\mathcal{VIEW}(d), \left(\mathcal{VIEW}(d)_{\mathsf{A}}, \mathcal{VIEW}(d)_{\mathsf{B}}\right)\right)\right] \leq \frac{9}{16} \cdot \ell\delta \leq \ell\delta.$$

Bounding the query complexity. We complete the proof by bounding the probability that CF makes too many oracle queries. For $i \in [j]$ let $Q_i = \{q \colon (q, \cdot) \in I_i \setminus I_{i-1}\}$ and let $Q = \bigcup_{i \in [j]} Q_i$ (i.e., Q is the set of queries appearing in a query/answer pair of I_j). The heart of the proof is in the following claim.

Claim 25. *For every query q it holds that*

$$\sum_{i \in [j]} \Pr[q \in Q_i \wedge \neg\, \mathrm{Dependent}_{I_{i-1}}(V_i)] \leq \frac{32}{\delta} \cdot \Pr[q \in V].$$

Namely, Claim 25 relates the probability that a query is asked by Finder, to the probability that this query is asked by one of the parties in V.

Proof. Fix $i \in [j]$ for a moment, and assume that during the i'th call to Finder on input $(\overline{t}^*, \cdot)$, algorithm Finder is about to ask a query q and let $\mathcal{I}^*$ be the value of $\mathcal{I}$ at this moment. The definition of Finder tells us that

$$\Pr[q \in V_i \mid \mathcal{I}^*, \overline{t}^*, \neg\, \mathrm{Dependent}_{\mathcal{I}^*}(V_i)] \geq \delta/32, \tag{12}$$

where the conditioning on $(\mathcal{I}^*, \overline{t}^*)$ means that V_i is consistent with $\mathcal{I}^*$ and $\overline{t}^*$. Applying a simple Bayes' rule, it follows that

$$\begin{aligned} \Pr[q \in V_i \mid \mathcal{I}^*, \overline{t}^*] &\geq \Pr[q \in V_i \wedge \neg\, \mathrm{Dependent}_{\mathcal{I}^*}(V_i) \mid \mathcal{I}^*, \overline{t}^*] \\ &= \Pr[\neg\, \mathrm{Dependent}_{\mathcal{I}^*}(V_i) \mid \mathcal{I}^*, \overline{t}^*] \cdot \Pr[q \in V_i \mid \mathcal{I}^*, \overline{t}^*, \neg\, \mathrm{Dependent}_{\mathcal{I}^*}(V_i)] \\ &\geq \Pr[\neg\, \mathrm{Dependent}_{\mathcal{I}^*}(V_i) \mid \mathcal{I}^*, \overline{t}^*] \cdot \frac{\delta}{32}. \end{aligned} \tag{13}$$

For $i \in [j]$ let $\mathcal{S}_i(q)$ be the set of $(\mathcal{I}^*, \overline{t}^*)$ pairs that cause Finder to ask the query q in the i'th round of CF — assuming that $w \in \Omega$ (chosen in CF) is consistent with $(\mathcal{I}^*, \overline{t}^*)$, then Finder asks the

query q in the i'th round of CF, and the value of $\mathcal{I}$ before it does so is $\mathcal{I}^*$. It follows that

$$\begin{aligned}\Pr[q \in V] &\geq \sum_{i\in[j]} \sum_{(\mathcal{I}^*,\bar{t}^*)\in\mathcal{S}_i(q)} \Pr[w \text{ is consistent with } (\mathcal{I}^*,\bar{t}^*)] \cdot \Pr[q \in V \mid \mathcal{I}^*,\bar{t}^*] \\ &\geq \sum_{i\in[j]} \sum_{(\mathcal{I}^*,\bar{t}^*)\in\mathcal{S}_i(q)} \Pr[w \text{ is consistent with } (\mathcal{I}^*,\bar{t}^*)] \cdot \frac{\delta}{32} \cdot \Pr[\neg \text{Dependent}_{\mathcal{I}^*}(V_i) \mid \mathcal{I}^*,\bar{t}^*] \\ &\geq \frac{\delta}{32} \cdot \sum_{i\in[j]} \Pr[q \in Q_i \wedge \neg \text{Dependent}_{I_{i-1}}(V_i)].\end{aligned}$$

The first inequality holds since a query is asked at most once in CF (and hence we are summing over disjoint events), the second one by Equation (13) and the last one holds since $\mathcal{F}$ is a simple family. □

Let $\widetilde{\mathsf{CF}}$ be the variant of CF that aborts in case $\text{Dependent}_{I_{i-1}}(V_i) = 1$ for some $i \in [j]$ (i.e., $\widetilde{\mathsf{CF}}$ aborts right after computing I_{i-1}). Let $\widetilde{Q}_i$ be the respective analogs of Q_i defined with respect to a random execution of $\widetilde{\mathsf{CF}}$, and let $\widetilde{Q} = \bigcup_{i\in[j]} \widetilde{Q}_i$ (i.e., $\widetilde{Q}$ denote all queries asked by Finder in $\widetilde{\mathsf{CF}}$). The same calculation done in Equation (7) yields that

$$SD(Q, \widetilde{Q}) \leq \ell\delta/8 \tag{14}$$

In the following we bound the number of queries made by Finder in $\widetilde{\mathsf{CF}}$, and derive a similar bound on CF.

A simple argument yields that $\Pr[q \in \widetilde{Q}_i] \leq \Pr[q \in Q_i \wedge \neg \text{Dependent}_{I_{i-1}}(V_i)]$, for every $i \in [j]$ and every query q. Thus, Claim 25

$$\sum_{i\in[j]} \Pr[q \in Q_i \wedge \neg \text{Dependent}_{I_{i-1}}(V_i)] \leq \frac{32}{\delta} \cdot \Pr[q \in V]$$

yields that

$$\Pr[q \in \widetilde{Q}] = \sum_{i\in[j]} \Pr[q \in \widetilde{Q}_i] \leq \frac{32}{\delta} \cdot \Pr[q \in V] \tag{15}$$

for every query q. It follows that

$$\mathrm{E}\left[\left|\widetilde{Q}\right|\right] = \mathrm{E}\left[\sum_q \chi_{\widetilde{Q}}(q)\right] \leq \frac{32}{\delta} \cdot \mathrm{E}\left[\sum_q \chi_V(q)\right] \leq 32\ell/\delta. \tag{16}$$

The first inequality holds by Equation (15) and linearity of expectation and the last one since at most ℓ queries are asked in V (where $\chi_x(q) = 1$ if $q \in x$ and $\chi_x(q) = 0$ otherwise).

A Markov argument yields that $\Pr\left[\left|\widetilde{Q}\right| > 64/\delta^2\right] \leq \ell\delta/2$. Hence, Equation (14) yields that $\Pr\left[|Q| > 64/\delta^2\right] \leq \ell\delta/2 + \ell\delta/8 < \ell\delta$. □

4.1 Handling Non Normal-Form Protocols

In this section we show how to construct a DependencyFinder for every simple function family and every oracle-aided protocol (possibly not in normal form). We do this by showing how to use a DependencyFinder for the normal form variant of a protocol, defined below, to construct a DependencyFinder (of the same quality) for the original protocol.

Definition 26 (the normal-form variant of a protocol)**.** *Given an ℓ-query oracle-aided protocol π, we define its* normal form variant *π_N as follows: the parties of π_N act as in π while sending additional "dummy" messages; following each oracle query made through the execution, the parties interact in a "dummy round" — the active party sends $\perp$ to the other party who answers with $\perp$. In addition, before sending the next message of π, the parties interact in $(\ell - k)$ consecutive dummy rounds, where k is the number of oracle queries made by the active party in the current round.*[13]

Lemma 27. *Let $\mathcal{F}$ be a function family, let π be an oracle-aided protocol and let π_N be its normal form variant. Assume $(\mathcal{F}, \pi_N)$ has a (T, ε)-*DependencyFinder*, then $(\mathcal{F}, \pi)$ has a (T, ε)-*DependencyFinder*.*

The straightforward proof of Lemma 27 is given below, but first let us use it for concluding the proof of Lemma 16.

Proof of Lemma 16. Let $\mathcal{F}$ be a simple function family, let $\pi = (\mathsf{A}, \mathsf{B})$ be an ℓ-query oracle-aided protocol and let π_{N} be its normal-form variant. Since π_{N} is in normal form according to Definition 20, Claim 23 yields that $(\mathcal{F}, \pi_{\mathrm{N}})$ has a $(64/\delta^2, \ell\delta)$-DependencyFinder for any $\delta \leq 1/\ell$. Hence, Lemma 27 yields that the same holds for $(\mathcal{F}, \pi)$. □

Proof of Lemma 27. Let $\mathcal{F}$, π and π_{N} be as in the statement of the lemma, and let $\mathsf{Finder}_{\mathrm{N}}$ be a (T, ε)-DependencyFinder for $(\mathcal{F}, \pi_{\mathrm{N}})$. We define the DependencyFinder for $(\mathcal{F}, \pi)$ as follows:

Algorithm 28 (Finder)**.**

Input: a transcript $\bar{t}$ of π and a list $\mathcal{I}$ of query/answer pairs.

Oracle: $f \in \mathcal{F}$.

Operation:

1. *Create the transcript $\bar{t}_N$ from $\bar{t}$ by inserting 2ℓ strings '$\perp$,', following each but the last message in $\bar{t}$.*
2. *For $k = 1$ to ℓ do:*
   (a) *Set $\mathcal{I} = \mathcal{I} \cup \mathsf{Finder}_N(\mathcal{I}, \bar{t}_N)$.*
   (b) *Set $\bar{t}_N = \bar{t}_N, \perp, \perp$*

It is easy to verify that a random output of CF_{N} can be sampled by applying a (deterministic) *injective* function M to a random output of CF, where M preserves the number of queries in the input (specifically, M is simply the padding function described in the first line of 28). This observation

[13] Note that each round in the original protocol is replaced by ℓ rounds in its normal-form variant, hence, concealing the number of actual oracle-queries made in each round.

immediately yields the required bound on the number of oracle queries done in CF, since these outputs determine the number of queries made to the oracle. To prove that the first property required by Definition 15 also holds (see the equation below), we also note that a random sample of $\mathcal{VIEW}^{\mathcal{F},\pi_{\mathrm{N}}}(d_{\mathrm{N}})$, for $d_{\mathrm{N}} \in \mathrm{Supp}(\mathsf{CF}_{\mathrm{N}})$, can be sampled by applying a (deterministic) *injective* function to $\mathcal{VIEW}^{\mathcal{F},\pi}(\mathsf{M}^{-1}(d_{\mathrm{N}}))$. It follows that

$$\begin{aligned}
&\mathop{\mathrm{E}}_{d\leftarrow\mathsf{CF}}\left[\mathrm{SD}\left(\mathcal{VIEW}^{\mathcal{F},\pi}(d),(\mathcal{VIEW}^{\mathcal{F},\pi}(d)_{\mathsf{A}},\mathcal{VIEW}^{\mathcal{F},\pi}(d)_{\mathsf{B}})\right)\right]\\
&=\mathop{\mathrm{E}}_{d\leftarrow\mathsf{CF}_{\mathrm{N}}}\left[\mathrm{SD}\left(\mathcal{VIEW}^{\mathcal{F},\pi_{\mathrm{N}}}(d),(\mathcal{VIEW}^{\mathcal{F},\pi_{\mathrm{N}}}(d)_{\mathsf{A}},\mathcal{VIEW}^{\mathcal{F},\pi_{\mathrm{N}}}(d)_{\mathsf{B}})\right)\right]\\
&\leq\varepsilon,
\end{aligned}$$

concluding the proof of the lemma. □

4.2 Proving Lemma 22

The following discussion is with respect to fixed values of $\bar{t}$ and $\mathcal{I}$, where Ind, $\mathcal{D}$ and $\mathcal{D}^{+}$ are defined with respect to these values as in the statement of Lemma 22.

Towards proving Lemma 22, we make the following observations: in Claim 29 we show that $\mathcal{D}$ is distributed as some *product distribution*, under the independence condition. In Claim 31, we use the first observation to express $\mathcal{D}$ as a uniform sampled edge of a *dense* bipartite graph.[14]

4.2.1 Product Characterization

Claim 29. *There exist two distributions $\mathcal{A}$ and $\mathcal{B}$ with $\mathcal{D} = (\mathcal{A}\times\mathcal{B}) \mid \mathrm{Ind}$.*

Proof. We show that we can write $\mathcal{D}(v) = \gamma_{\mathsf{A}}(v_{\mathsf{A}})\cdot\gamma_{\mathsf{B}}(v_{\mathsf{B}})\cdot c$ for every $v = (v_{\mathsf{A}}, v_{\mathsf{B}}) \in \mathrm{Supp}(\mathcal{D})$, where γ_{A} and γ_{B} are appropriate functions, and c is a global constant. This would imply the claim, letting $\mathcal{A}$ be the distribution over $\mathrm{Supp}(\mathcal{D}_{\mathsf{A}})$ with $\mathcal{A}(v_{\mathsf{A}}) = c_{\mathcal{A}}\cdot\gamma_{\mathsf{A}}(v_{\mathsf{A}})$, and $\mathcal{B}$ be the distribution over $\mathrm{Supp}(\mathcal{D}_{\mathsf{B}})$ with $\mathcal{B}(v_{\mathsf{B}}) = c_{\mathcal{B}}\cdot\gamma_{\mathsf{B}}(v_{\mathsf{B}})$, for the appropriate constants $c_{\mathcal{A}}$ and $c_{\mathcal{B}}$.

Proposition 8 yields that

$$\mathcal{D}(v) = \frac{\Pr_{\Omega}[r_{\mathsf{A}}, r_{\mathsf{B}}]\cdot\alpha^{\mathcal{I}}_{v_{\mathsf{A}}|v_{\mathsf{B}}}\cdot\alpha^{\mathcal{I}}_{v_{\mathsf{B}}|v_{\mathsf{A}}}}{\Pr_{\Omega|\mathcal{I}}[\bar{t},\mathrm{Ind}]}, \tag{17}$$

for every $v = (r_{\mathsf{A}}, r_{\mathsf{B}}, \cdot) \in \mathrm{Supp}(\mathcal{D})$. Since the random coins of the parties are chosen independently, it holds that $\Pr_{\Omega}[r_{\mathsf{A}}, r_{\mathsf{B}}] = \Pr_{\Omega}[r_{\mathsf{A}}]\cdot\Pr_{\Omega}[r_{\mathsf{B}}]$. In addition, the definition of $\mathcal{D}$ yields that $\alpha^{\mathcal{I}}_{v_{\mathsf{A}}|v_{\mathsf{B}}}\alpha^{\mathcal{I}}_{v_{\mathsf{A}}}$ and that $\alpha^{\mathcal{I}}_{v_{\mathsf{B}}|v_{\mathsf{A}}} = \alpha^{\mathcal{I}}_{v_{\mathsf{B}}}$, for every $v \in \mathrm{Supp}(\mathcal{D})$. Taking $\gamma_{\mathsf{A}}(v_{\mathsf{A}}) := \Pr_{\Omega}[r_{\mathsf{A}}]\cdot\alpha^{\mathcal{I}}_{v_{\mathsf{A}}}$ and $\gamma_{\mathsf{B}}(v_{\mathsf{B}}) := \Pr_{\Omega}[r_{\mathsf{B}}]\cdot\alpha^{\mathcal{I}}_{v_{\mathsf{B}}}$, we obtain the desired result. □

4.2.2 Graph Characterization

We use the above product characterization to show that we can think of $\mathcal{D}$ as a distribution over random edges of some bipartite graph G. We show that $\mathcal{D}$ is close to being a product distribution by showing that the graph is dense. Specifically, we show that every vertex in G is connected to most of the vertices on the other side.

[14] Observations of similar spirit were done in [1]. Much the following text is taken verbatim from [1].

Definition 30 (The graph G)**.** *Define the bipartite graph $G = (V_{\mathcal{A}}, V_{\mathcal{B}}; E)$ as follows: let $\mathcal{A}$ and $\mathcal{B}$ be as the distribution guaranteed by Claim 29. Every node $a \in V_{\mathcal{A}}$ has a corresponding view $\text{view}_{\mathsf{A}}(a)$ of A in the support of $\mathcal{A}$; we let the number of nodes corresponding to a view v_{A} be proportional to $\mathcal{A}(v_{\mathsf{A}})$, meaning that $\mathcal{A}$ corresponds to the uniform distribution over the left-side vertices $V_{\mathcal{A}}$. Similarly, every node $b \in V_{\mathcal{B}}$ has a corresponding view $\text{view}_{\mathsf{B}}(b)$ of B, such that $\mathcal{B}$ corresponds to the uniform distribution over $V_{\mathcal{B}}$. Finally, let $E = \{(a,b) \in (V_{\mathcal{A}} \times V_{\mathcal{B}}) \cap \text{Ind}\}$.*

Claim 29 yields that $\mathcal{D}$ corresponds to the distribution obtained by letting $(a,b) \leftarrow E$, and choosing $(\text{view}_{\mathcal{A}}(a), \text{view}_{\mathcal{B}}(b))$. We next show that this graph must be dense.

Claim 31. *Let $G = (V_{\mathcal{A}}, V_{\mathcal{B}}; E)$ be as in Definition 30. Then $d(a) \geq |V_{\mathcal{B}}| \cdot (1 - 2\ell\delta)$ for every $a \in V_{\mathcal{A}}$, and $d(b) \geq |V_{\mathcal{A}}| \cdot (1 - 2\ell\delta)$ for every $b \in V_{\mathcal{B}}$, where $d(x)$ stands for the degree of the vertex x.*

Proof. Since, by assumption, $\Pr_{v \leftarrow \mathcal{D}}[q \in v \wedge (q, \cdot) \notin \mathcal{I}] \leq \delta$ for every query q, since ℓ bounds the query complexity of π, and finally since $\mathcal{F}$ is simple, we have that

$$\Pr_{v_{\mathsf{A}} \leftarrow \mathcal{D}_{\mathsf{A}}} [\text{Dependent}_{\mathcal{I}}(v_{\mathsf{A}}, v_{\mathsf{B}})] \leq \ell\delta \tag{18}$$

for every fixed $v_{\mathsf{B}} \in \text{Supp}(\mathcal{D}_{\mathsf{B}})$, and the analogous condition for every fixed $v_{\mathsf{A}} \in \text{Supp}(\mathcal{D}_{\mathsf{A}})$. For a vertex $a \in V_{\mathcal{A}}$, let $\tilde{\tilde{E}}(a) = \{b \in V_{\mathcal{B}} : (a,b) \notin E\}$. We next show that $\sum_{b \in \tilde{\tilde{E}}(a)} d(b) \leq \ell\delta \cdot |E|$ for every $a \in V_{\mathcal{A}}$. Note that the probability of a vertex x being chosen when selecting a random edge in E is $\frac{d(x)}{|E|}$. Assuming that $\sum_{b \in \tilde{\tilde{E}}(a)} \frac{d(b)}{|E|} > \ell\delta$, then $\Pr_{v_{\mathsf{B}} \leftarrow \mathcal{D}_{\mathsf{B}}} [\text{Dependent}_{\mathcal{I}}(\text{view}_{\mathsf{A}}(a), v_{\mathsf{B}})] > \ell\delta$, contradicting Equation (18). An analogous argument shows that $\sum_{a \in \tilde{\tilde{E}}(b)} d(a) \leq \ell\delta \cdot |E|$ for every $b \in V_{\mathcal{B}}$, and we conclude the proof using the following claim:

Fact 32. *[1, claim 4.6] Let $G = (V_{\mathcal{A}}, V_{\mathcal{B}}; E)$ be a nonempty bipartite graph. Assume there exists $\gamma \leq 1/2$ such that $|\tilde{\tilde{E}}(v)| \leq \gamma |E|$ for every vertex $v \in (V_{\mathcal{A}} \cup V_{\mathcal{B}})$), then $d(a) \geq |V_{\mathcal{B}}| \cdot (1 - 2\gamma)$ for every $a \in V_{\mathcal{A}}$, and $d(b) \geq |V_{\mathcal{A}}| \cdot (1 - 2\gamma)$ for every $b \in V_{\mathcal{B}}$.*

□

We now use the above claims to prove Lemma 22.

Proof of Lemma 22. The first part of the lemma immediately follows from Claim 31. We prove the second part of the lemma by showing that

$$\Pr_{v \leftarrow \mathcal{D}^+ | v_{\mathsf{B}} = v_{\mathsf{B}}^+} \left[\text{Dependent}_{\mathcal{I}}(v_{\mathsf{A}}, v_{\mathsf{B}}^+)\right] \leq 4\delta \cdot \ell_{|\bar{t}|+1}(v_{\mathsf{B}}^+) \tag{19}$$

for every $v_{\mathsf{B}}^+ \in \text{Supp}(\mathcal{D}_{\mathsf{B}}^+)$, where we assume for concreteness that B is active in the $(|\bar{t}|+1)$ round of π. Since Equation (19) trivially holds in case $\ell_{|\bar{t}|+1}(v_{\mathsf{B}}^+) = 0$, in the following we prove it for $\ell_{|\bar{t}|+1}(v_{\mathsf{B}}^+) = 1$ (recall that, by definition, $\ell_{|\bar{t}|+1}(v_{\mathsf{B}}^+) \leq 1$).

Fix such view $v_{\mathsf{B}}^+ \in \text{Supp}(\mathcal{D}_{\mathsf{B}}^+)$, let $v'_{\mathsf{B}} \in \text{Supp}(\mathcal{D}_{\mathsf{B}})$ be its $|\bar{t}|$-round prefix and let q' be the query asked in its $(|\bar{t}|+1)$ round. We assume without loss of generality that $(q', \cdot) \notin \mathcal{I}$, as otherwise the proof is trivial. Fix further $b \in V_{\mathsf{B}}$ with $\text{view}_{\mathsf{B}}(b) = v'_{\mathsf{B}}$, let $N(b)$ be b's neighbors in G and let $\mathcal{S} = \{a \in N(b) \colon q' \in \text{view}_{\mathsf{A}}(a)\}$. Since $\Pr_{v \leftarrow \mathcal{D}}[q \in v \wedge (q, \cdot) \notin \mathcal{I}] \leq \delta$, we have that

$$\frac{\sum_{a \in \mathcal{S}} d(a)}{|E|} \leq \delta \tag{20}$$

and therefore

$$\Pr_{a\leftarrow N(b)}[a\in\mathcal{S}]=\frac{|\mathcal{S}|}{d(b)}\leq\frac{|\mathcal{S}|}{(1-2\ell\delta)|V_{\mathcal{A}}|}\leq\frac{|\mathcal{S}||V_{\mathcal{B}}|}{(1-2\ell\delta)|E|}\leq\frac{\sum_{a\in\mathcal{S}}d(a)}{(1-2\ell\delta)^2|E|}\leq\frac{\delta}{(1-2\ell\delta)^2}\leq 4\delta \quad (21)$$

The first and third inequalities hold by Claim 31, the second since $|E|\leq|V_{\mathcal{A}}||V_{\mathcal{B}}|$, and the fourth by Equation (20). Since Equation (21) holds for every $b\in V_{\mathsf{B}}$ with $\text{view}_{\mathsf{B}}(b)=v'_{\mathsf{B}}$, it follows that

$$\Pr_{v\leftarrow\mathcal{D}^+|v_{\mathsf{B}}=v^+_{\mathsf{B}}}\left[q'\in v_{\mathsf{A}}\right]\leq 4\delta \quad (22)$$

Finally, since $\mathcal{F}$ is a simple family, Equation (22) yields that $\Pr_{v\leftarrow\mathcal{D}^+|v_{\mathsf{B}}=v^+_{\mathsf{B}}}\left[\text{Dependent}_{\mathcal{I}}(v_{\mathsf{A}},v^+_{\mathsf{B}})\right]\leq 4\delta$, concluding the proof of Equation (19), and thus the proof of the lemma. □

5 Applications

In this section we use our main result (i.e., the oracle-aided to no-oracle protocol mapping for simple function families) from Section 3 to derive the impossibility of realizing two cryptographic tasks, with respect to simple function families (implying the same result with respect to the all function family, which we later show to be simple). In Section 5.1 we re-establish the result of [13], showing that key-agreement protocols cannot be realized with respect to simple function families. Then, in Section 5.2, we extend the lower-bound of [17] on the accuracy of two-party differentially private no-oracle protocols, to show it also holds (with a slight loss in parameters) for oracle-aided differentially private protocols (with respect to this class of function families). In Section 5.3, we use the fact that the all-function family is simple to prove the impossibility of reducing each of the above primitives to the hardness of one-way functions in a black-box manner.

Remark 33 (definitions for no-oracle primitives)**.** *Throughout this section we only give formal definitions (of the security and correctness) of primitives with respect to oracle-aided protocols. Deriving formal definitions for their no-oracle counterparts can be easily done by considering the trivial function family (i.e., a singleton family, whose only member returns $\perp$ on any query).*

5.1 Key Agreement Protocols

In a key-agreement protocol two parties wish to agree on a common secret in a secure way — an observer (adversary) seeing the communication transcript, cannot find the secret. Below we prove that with respect to a certain class of function families, non-trivial key-agreement cannot be achieved. We start by formally defining the notion of key agreement. We then recall the known fact that in the no-oracle model, an adversary can reveal any secret agreement between two parties in the strongest possible sense (i.e., with the same probability that the parties themselves agree). Combining this fact with the mapping from oracle-aided to no-oracle protocols, described in Section 3, yields a similar result for oracle-aided protocols.

We remark that the results presented in this section yield very little conceptual added-value to what was already shown by [13, 1]. We do, however, present them here to demonstrate how they are easily derived from our main result (Theorem 13), and as a warm-up before moving on to the other application of our main result, described in Section 5.2.

5.1.1 Standard Definitions and Known Facts

Recall (see Section 2.2) that for a joint view $v \in \mathrm{Supp}\left(\langle \pi^f \rangle\right)$, we let $\mathsf{trans}(v)$ denote the communication transcript in v, and $\mathrm{out}_i^{\mathsf{P}}(v)$ denote the output of the party P at the i'th round. In the following we let $\mathrm{out}^{\mathsf{P}}(v) = \mathrm{out}_m^{\mathsf{P}}(v)$, where m is the last round in v.

Definition 34 (key agreement protocol)**.** *Let $0 \leq \gamma,\ \alpha \leq 1$ and $k \in \mathbb{N}$. A two-party, oracle-aided protocol $\pi = (\mathsf{A}, \mathsf{B})$ is a (k, α, γ)-key-agreement protocol with respect to a function family $\mathcal{F}$, if the following hold:*

Consistency: *π is* $(1-\alpha)$-consistent *with respect to $\mathcal{F}$. Namely,*

$$\Pr_{f \leftarrow \mathcal{F}, v \leftarrow \langle \pi^f \rangle} \left[\mathrm{out}^{\mathsf{A}}(v) = \mathrm{out}^{\mathsf{B}}(v) \right] \geq 1 - \alpha. \tag{23}$$

Security: *For every $P \in \{\mathsf{A}, \mathsf{B}\}$ and any k-query adversary* Eve*,*

$$\Pr_{f \leftarrow \mathcal{F}, v \leftarrow \langle \pi^f \rangle} \left[\mathsf{Eve}^f(\mathsf{trans}(v)) = \mathrm{out}^P(v) \right] \leq \gamma. \tag{24}$$

A protocol π is an (α, γ)-key-agreement protocol, if it is a $(\cdot, \alpha, \gamma)$-key-agreement protocol with respect to the trivial function family.[15]

In the no-oracle model, all correlation between the parties is implied by the transcript. Hence, an adversary that on a given transcript $\bar{t}$ samples a random view for A that is consistent with $\bar{t}$ and outputs whatever A would upon this view, agrees with B with the same probability as does A. This simple argument yields the following fact.

Fact 35. *Let $0 \leq \alpha \leq 1$ and let $\pi = (\mathsf{A}, \mathsf{B})$ be a no-oracle, two-party, no-input protocol. Assume that the probability that in a random execution of π both parties output the same value is $1 - \alpha$. Then there exists an adversary that, given the transcript of a random execution of π, outputs the same value as does B with probability $1 - \alpha$.*

An immediate implication of Fact 35 is that there does not exist any no-oracle, two-party, (α, γ)-key-agreement protocol for any $0 \leq \gamma < 1 - \alpha$. We next use our main result from Section 3 to prove a similar result for oracle-aided protocols.

5.1.2 Limits on Oracle-Aided Key-Agreement Protocols

Theorem 36. *Let $\mathcal{F}$ be a function family and let π be an oracle-aided protocol. Assume that the pair $(\mathcal{F}, \pi)$ has a* (T, ε)-mapping*, then π is* not *a (T, α, γ)-key-agreement with respect to $\mathcal{F}$ for any $0 \leq \gamma < 1 - (\alpha + \varepsilon)$.*

Proof. Assume to the contrary that π is a (T, α, γ)-key-agreement with respect to $\mathcal{F}$ for some $0 \leq \gamma < 1 - (\alpha + \varepsilon)$. Let $\tilde{\pi} = (\tilde{\mathsf{A}}, \tilde{\mathsf{B}})$ and Map be the no-input no-oracle protocol and oracle-aided algorithm, guaranteed by the assumption of the theorem. The first item in Definition 12 yields that

$$\mathrm{SD}\left(\left(\mathrm{out}^{\tilde{\mathsf{A}}}(v), \mathrm{out}^{\tilde{\mathsf{B}}}(v)\right)_{v \leftarrow \langle \tilde{\pi} \rangle}, \left(\mathrm{out}^{\mathsf{A}}(v), \mathrm{out}^{\mathsf{B}}(v)\right)_{f \leftarrow \mathcal{F}, v \leftarrow \langle \pi^f \rangle} \right) \leq \varepsilon. \tag{25}$$

[15] A stronger definition may require consistency to hold for every $f \in \mathcal{F}$, rather than for a random f. Our impossibility result (as well the results of [13, 1]), however, holds even with respect to the weaker definition given here.

Hence, the $(1-\alpha)$-consistency of π yields that

$$\tau := \Pr_{v \leftarrow \langle \tilde{\pi} \rangle} \left[\mathrm{out}^{\tilde{\mathsf{A}}}(v) = \mathrm{out}^{\tilde{\mathsf{B}}}(v) \right] \geq 1 - (\alpha + \varepsilon). \tag{26}$$

Fact 35 yields an adversary $\widetilde{\mathsf{Eve}}$ that given the transcript of a random execution of $\tilde{\pi}$, outputs the same value as does B with probability τ. Let Eve be an adversary for π that upon a transcript $\bar{t}$ (of an execution of π with access to f) applies $\widetilde{\mathsf{Eve}}$ to $\mathsf{Map}^f(\bar{t})$ and outputs whatever $\widetilde{\mathsf{Eve}}$ does. Note that by Definition 12, Eve makes at most T oracle calls. The definition of Eve yields that

$$\begin{aligned} \Pr_{f \leftarrow \mathcal{F}, v \leftarrow \langle \pi^f \rangle} \left[\mathsf{Eve}^f(\mathsf{trans}(v)) = \mathrm{out}^{\mathsf{B}}(v) \right] &= \Pr_{f \leftarrow \mathcal{F}, v \leftarrow \langle \pi^f \rangle} \left[\tilde{\mathsf{Eve}}\left(\mathsf{Map}^f(\mathsf{trans}(v)) \right) = \mathrm{out}^{\mathsf{B}}(v) \right] \\ &= \Pr_{\tilde{v} \leftarrow \langle \tilde{\pi} \rangle} \left[\tilde{\mathsf{Eve}}(\mathsf{trans}(\tilde{v})) = \mathrm{out}^{\tilde{\mathsf{B}}}(\tilde{v}) \right] \\ &= \tau \geq 1 - (\alpha + \varepsilon), \end{aligned} \tag{27}$$

where the second equality follows from the furthermore statement of the first item in Definition 12, stating that $(\mathsf{Map}^f(\mathsf{trans}(v)), \mathrm{out}^{\mathsf{B}}(v))$ is identically distributed as $(\mathsf{trans}(\tilde{v}), \mathrm{out}^{\tilde{\mathsf{B}}}(\tilde{v}))$, where f, v, and $\tilde{v}$ are sampled as in Equation (27). □

Combining Theorems 13 and 36 yields the following corollary.

Corollary 37. *Let $\mathcal{F}$ be a simple function family. For parameters $k, \ell \in \mathbb{N}$ and $\alpha, \gamma \in \mathbb{R}$ with $k \geq 2^{10} \cdot \left(\frac{\ell}{1-\alpha-\gamma} \right)^2$ and $1 - \alpha > \gamma \geq 0$, there exists no ℓ-query oracle-aided protocol, that is (k, α, γ)-key-agreement with respect to $\mathcal{F}$.*

Proof. Let $\mathcal{F}$ be a simple function family and let π be an ℓ-query oracle-aided protocol. For $\varepsilon = \frac{1-\alpha-\gamma}{2}$, Theorem 13 yields that $(\mathcal{F}, \pi)$ has a (T, ε)-mapping for $T = 256 \cdot \left(\frac{\ell}{\varepsilon} \right)^2 = 2^{10} \cdot \left(\frac{\ell}{1-\alpha-\gamma} \right)^2$. Since $0 \leq \gamma < 1 - (\alpha + \varepsilon)$ and $k \geq T$, Theorem 36 yields that π is not a (k, α, γ)-key-agreement protocol with respect to $\mathcal{F}$. □

5.2 Differentially Private Two-Party Computation

In this section we apply our main result to extend the lower-bound of McGregor et al. [17] to oracle-aided protocols equipped with simple function families. Specifically, we show that when given access to a random member of a simple function family (e.g., the all-function family), any two-party, differentially private, oracle-aided protocol computing the inner product of two s-bit strings, exhibits error magnitude of roughly $\Omega(\sqrt{s}/\log s)$. This fact is later used in Section 5.3.4 to show that differentially private accurate computation of the inner product, *cannot* be reduced to one-way functions in a black-box way.

Unlike the case of key-agreement protocols discussed in Section 5.1, here we consider a setting where the parties *do* hold private inputs. Since our main result (Theorem 13) only handles no input protocols, in order to apply it to differentially private protocols we need first to reduce the question in hand to such no-input protocols. Indeed, much of the following text deals with this transformation.

We begin (Section 5.2.2) by using Theorem 13 together with (an "information theoretic") result by [17], to show that a "sampled-input" protocol cannot be both differentially private and a good

approximation for the inner product of two strings. In a sampled-input protocol, the no-input parties choose the inputs to the functionality (in our case, the inner product function) by themselves. We then (Section 5.2.3) derive the same limitation on protocols with inputs, but where the correctness and privacy are measured with respect to *uniformly chosen* inputs. Finally (Section 5.2.4), we use the latter result to show the same limitation for *fixed* inputs, hence proving the required result. Before starting with the aforementioned plan, let us first recall the formal definition of differential privacy, and cite the result of [17].

5.2.1 Standard Definitions and Known Facts

We start by recalling the standard definition of differential privacy for mechanisms (in a centralized model, where the mechanism has access to all the data). Let Σ be some alphabet. For strings $x, x' \in \Sigma^s$, let $H_d(x, x') = |\{i \in [s] \colon x_i \neq x'_i\}|$ denote the Hamming distance between x and x'. A *randomized mechanism* operating on s-long strings (databases) is a randomized algorithm that given input in Σ^s, outputs a value in the range $\mathcal{R}$.

Definition 38 ((α, γ)-differential privacy [6] (in the centralized model))**.** *A randomized mechanism* M *over* Σ^s *is* (α, γ)-$\mathsf{differentially\ private}$*, if for every distinguisher* D *and every* $x, x' \in \Sigma^s$ *with* $H_d(x, x') = 1$*, it holds that*

$$\Pr[\mathsf{D}(\mathsf{M}(x)) = 1] \leq e^{\alpha} \cdot \Pr[\mathsf{D}(\mathsf{M}(x')) = 1] + \gamma.$$

If M *satisfies* (α, γ)*-differential privacy with* $\gamma = 0$*, then* M *is just* α*-differentially private.*[16]

Differential privacy extends naturally to the setting of two-party (semi-honest) protocols by requiring that the view of each party satisfies differential privacy with respect to the other party's private input. In this work we use a relaxed definition (and hence potentially easier to achieve) that only requires that the communication transcript (rather than the whole view of a party) is differentially private with respect to each party's input. Such a requirement is easily implied by the above requirement on views, since any distinguisher that breaks the privacy seeing only the transcript can break the privacy seeing the whole view of a party (by simply disregarding everything in the view but the transcript part). We next define differential privacy for protocols using similar definitions to those given in [2, 17, 19]. Indeed, our definitions are close in spirit to the definition of IND-CDP from [19] (which they showed to be implied by all other definitions that they considered for computational differential privacy).

In the following, when we say protocol, we mean a two-party protocol. We focus on protocols where each party holds an s-bit string as its private input, and call such protocols *s-bit input* protocols. We adapt the notations from Section 2.2 (defined for no-input protocols) to protocols with inputs, with the understanding that the view of a party also includes its s-bit private input. Specifically, given an oracle-aided protocol $\pi = (\mathsf{A}, \mathsf{B})$, a function f, and $x, y \in \{0,1\}^*$, we define $\langle \pi^f(x, y) \rangle$ to be $\langle (\mathsf{A}^f(x), \mathsf{B}^f(y)) \rangle$ (i.e., the distribution over the joint views of parties in a random execution of π with access to f, where the private input of A is x and the private input of B is y). Recall that for $v \in \mathrm{Supp}(\langle \pi^f(x, y) \rangle)$, we let $\mathsf{trans}(v)$ denote the communication transcript in v, and we let $\mathrm{out}_i^{\mathsf{P}}(v)$ denote the output of the party P at the i'th round. In the following we let $\mathrm{out}^{\mathsf{P}}(v)$ denote the output of the party P at the last round of v.

[16] Throughout this section, we assume $\alpha, \gamma \geq 0$.

Definition 39 (differential privacy for oracle-aided protocols)**.** *Let $\mathcal{F}$ be a function family and let $\pi = (\mathsf{A}, \mathsf{B})$ be an s-bit input, oracle-aided protocol. The protocol π is* (k, α, γ)-differentially private with respect to $\mathcal{F}$ and A*, if for every k-query, oracle-aided distinguisher D and every $x, x', y \in \{0,1\}^s$ with $H_d(x, x') = 1$, it holds that*

$$\Pr_{f \leftarrow \mathcal{F}, v \leftarrow \langle \pi^f(x,y) \rangle} \left[\mathsf{D}^f(\mathsf{trans}(v)) = 1 \right] \leq e^{\alpha} \cdot \Pr_{f \leftarrow \mathcal{F}, v \leftarrow \langle \pi^f(x',y) \rangle} \left[\mathsf{D}^f(\mathsf{trans}(v)) = 1 \right] + \gamma.$$

Being (k, α, γ)-differentially private with respect to $\mathcal{F}$ and B, is analogously defined. If π is (k, α, γ)-differentially private with respect to $\mathcal{F}$ and both parties, then it is (k, α, γ)-differentially private with respect to $\mathcal{F}$*.*
Finally, π is (α, γ)-differentially private, if it is $(\cdot, \alpha, \gamma)$-differentially private with respect to the trivial function family.

Note that for no-oracle protocols, the above definition of (α, γ)-differentially private matches the standard (no-oracle) definition (slightly relaxed, as we only require the transcript to preserve the privacy of the parties). Our impossibility results, given below, apply to privacy parameter α being smaller than some constant.

Since differentially private mechanisms cannot be deterministic, for any deterministic (non-constant) function g of the input, one can only hope for the output of the mechanism being a good approximation for g. We next define a notion of accuracy for differentially private protocols.

Definition 40 (good approximations)**.** *Let $g : \{0,1\}^s \times \{0,1\}^s \mapsto \mathbb{R}$ be a deterministic function and let $\pi = (\mathsf{A}, \mathsf{B})$ be an s-bit input, oracle-aided protocol. The protocol π is a (β, d)-approximation for g with respect to a function family $\mathcal{F}$, if for every $x, y \in \{0,1\}^s$ and $\mathsf{P} \in \{\mathsf{A}, \mathsf{B}\}$, it holds that*

$$\Pr_{f \leftarrow \mathcal{F}, v \leftarrow \langle \pi^f(x,y) \rangle} \left[\left| g(x,y) - \mathrm{out}^{\mathsf{P}}(v) \right| > d \right] < \beta. \tag{28}$$

Namely, we require that the output of both parties is within distance d from $g(x, y)$ with probability at least β.

For two s-bit strings x and y, let $\mathsf{IP}(x, y)$ denote the inner product of x and y; that is $\mathsf{IP}(x, y) = \sum_{i \in [s]} x_i \cdot y_i$. McGregor et al. [17] showed that for small enough γ, no two-party, no-oracle, (α, γ)-differentially private protocol for computing the inner product of two s-bit databases can be a $(0.01, d)$-approximation for $d \in o(\sqrt{s}/\log s)$. This follows from the following general theorem.

Theorem 41 ([17, Theorem A.5])**.** *Let $\pi = (\mathsf{A}, \mathsf{B})$ be an s-bit (no-oracle) protocol, let X_{In} and Y_{In} be the inputs of A and B, respectively, and let X_{out} and Y_{out} be the outputs of A and B, respectively, induced by a random execution of π. Assume that both X_{In} and Y_{In} are independently and uniformly chosen from $\{0,1\}^s$, and that π is (α, γ)-differentially private, then*

$$\Pr\left[|Y_{\mathrm{out}} - \mathsf{IP}(X_{\mathrm{In}}, Y_{\mathrm{In}})| < \Delta := \Omega\left(\frac{\sqrt{s}}{\log s} \cdot \frac{\tau}{e^{\alpha}} \right) \right] \leq \tau$$

for every $1 \geq \tau \geq 48s\gamma$. [17] *The same holds for X_{out}.*

[17] This constraint implies that γ should be smaller than the inverse of some polynomial in s, however, this is how we typically think of γ.

In the next section we use similar arguments to the ones used by McGregor et al. [17], to prove a variant of Theorem 41 for (no-oracle) no-input protocols (which we call here *sampled-input* protocols). For that we recall a few definitions and results from [17].

Lemma 42 ([17, Lemma A.3])**.** *Let* M *be an* (α,γ)*-differentially private mechanism over* $\{0,1\}^s$*. Then for every* $\nu > 0$ *and every* $x, x' \in \{0,1\}^s$ *with* $H_d(x,x') = 1$*, it holds that*

$$\Pr_{m \leftarrow \mathsf{M}(x)}\left[\frac{\Pr[\mathsf{M}(x)=m]}{\Pr[\mathsf{M}(x')=m]} \notin \left[e^{-(\nu+\alpha)}, e^{(\nu+\alpha)}\right]\right] < \gamma \cdot \frac{1+e^{-(\nu+\alpha)}}{1-e^{-\nu}}. \tag{29}$$

Unpredictability of bit sources. The model of random sources introduced by Santha and Vazirani [23] is one where each bit is somewhat unpredictable given the previous ones. An unpredictable s-bit source is a random variable over $\{0,1\}^s$ with the property that given any prefix of it, it is hard to guess the value of the next bit.

Definition 43 ((η,γ)-unpredictable bit source)**.** *For* $\eta \in [0,1]$ *a random variable* $X = (X_1,\ldots,X_s)$ *taking values in* $\{0,1\}^s$ *is an* (η,γ)-unpredictable bit source*, if with probability at least* $1-\gamma$ *over* $i \leftarrow [s]$ *and over* $(x_1,\ldots,x_{i-1}) \leftarrow (X_1,\ldots,X_{i-1})$*, it holds that*

$$\eta \leq \frac{\Pr\left[X_i = 0 \mid X_1 = x_1, \ldots X_{i-1} = x_{i-1}\right]}{\Pr\left[X_i = 1 \mid X_1 = x_1, \ldots X_{i-1} = x_{i-1}\right]} \leq 1/\eta.$$

A variable X *is* η*-unpredictable, if it is* $(\eta,0)$*-unpredictable.*

A random variable $X = (X_1,\ldots,X_s)$ *taking values in* $\{0,1\}^s$ *is an* (η,γ)-strongly-unpredictable bit source*, if with probability at least* $1-\gamma$ *over* $i \leftarrow [s]$ *and over* $(x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_s) \leftarrow (X_1,\ldots,X_{i-1},X_{i+1},\ldots,X_s)$*, it holds that*

$$\eta \leq \frac{\Pr\left[X_i = 0 \mid X_1 = x_1, \ldots X_{i-1} = x_{i-1}, X_{i+1} = x_{i+1}, \ldots, X_s = x_s\right]}{\Pr\left[X_i = 1 \mid X_1 = x_1, \ldots X_{i-1} = x_{i-1}, X_{i+1} = x_{i+1}, \ldots, X_s = x_s\right]} \leq 1/\eta.$$

Note that if X is η-unpredictable for $\eta = 1$, then it is uniform. More generally, the larger η is, the more "randomness" is the source guaranteed to have. Specifically, an unpredictable source has high min-entropy.

Fact 44. *Let* $X = (X_1,\ldots,X_s)$ *be an* η*-unpredictable source, then the* min-entropy of X*, defined as* $\mathrm{H}_\infty(X) = \min_{x \in \mathrm{Supp}(X)} \log \frac{1}{\Pr[X=x]}$*, is at least* βs *for* $\beta = \log(1+\eta)$*.*

Proof. Fix $(x_1,\ldots,x_s) \in \mathrm{Supp}(X)$, $i \in [s]$ and $b \in \{0,1\}$. Definition 43 yields that $\Pr\left[X_i = b \mid X_1 = x_1, \ldots X_{i-1} = x_{i-1}\right] \geq \eta \cdot \Pr\left[X_i = 1-b \mid X_1 = x_1, \ldots X_{i-1} = x_{i-1}\right]$.[18] Since $\Pr\left[X_i = b \mid X_1 = x_1, \ldots X_{i-1} = x_{i-1}\right] + \Pr\left[X_i = 1-b \mid X_1 = x_1, \ldots X_{i-1} = x_{i-1}\right] = 1$, it follows that $(1+\eta) \cdot \Pr\left[X_i = b \mid X_1 = x_1, \ldots X_{i-1} = x_{i-1}\right] \leq 1$, and therefore $\Pr[X = (x_1,\ldots,x_s)] \leq \left(\frac{1}{1+\eta}\right)^s$. □

We will make use of the following results from [17].

Lemma 45 ([17, Lemma A.2])**.** *Let* $X = (X_1,\ldots,X_s)$ *be an* (η,γ)*-strongly-unpredictable bit source, then, for every* $\nu > 0$*, it is* $\frac{s\gamma}{\nu}$*-close to some* $\hat{\eta}$*-unpredictable bit source, where* $\hat{\eta} = \eta \cdot \frac{1-\nu}{1+\nu}$*.*

[18] For $b = 1$, this is implied by the right hand side inequality in the condition of Definition 43.

Corollary 46. *Let $X = (X_1, \ldots, X_s)$ be an (η, γ)-strongly-unpredictable bit source, then it is $2s\gamma$-close to some $\eta/3$-unpredictable bit source.*

Proof. Apply Lemma 45 with $\nu = 1/2$. □

Theorem 47 ([17, Theorem 3.4])**.** *Let X and Y be s-bit independent bit sources, where X is η-unpredictable and Y has min-entropy at least βs, and let $Z = \mathsf{IP}(X, Y) \bmod m$ for some $m \in \mathbb{N}$. Assume that $s \geq c \cdot \frac{m^2}{\eta\beta} \cdot \log\left(\frac{m}{\beta}\right) \cdot \log\left(\frac{m}{\gamma}\right)$ for some $\gamma \in [0, 1]$, where c is a universal constant, then $\mathrm{SD}\left((Y, Z), (Y, U_m)\right) \leq \gamma$, where U_m is uniform on $\mathbb{Z}_m$ and independent of Y.*

5.2.2 Limits on Sampled-Input Protocols

In this section we give a lower-bound on the accuracy of no-input, two-party, differentially private protocols, where the inputs for the functionally are derived from the parties' private coins (while preserving differential privacy with respect to these inputs). We do so by combining a result from [17] (stated here as Theorem 41) and our main result from Section 3 (Theorem 13).

Definition 48 (sampled-input protocols)**.** *A no-input protocol $\pi = (\mathsf{A}, \mathsf{B})$ is an s-*bit sampled-input protocol*, if the output of party A in any execution of π is of the form (x, a) and the output of party B is of the form (y, b), where $x, y \in \{0,1\}^s$. We call x [resp., y] the* sampled input *of A [resp., B], and a [resp., b] the* actual output *of A [resp., B].*

For $v \in \mathrm{Supp}\left(\langle \pi^f \rangle\right)$, let $\mathrm{SInp}^{\mathsf{P}}(v)$ denote the sampled input of party P in v, and $\mathrm{AOut}^{\mathsf{P}}(v)$ denote the actual output of the party P.[19]

We next extend the notion of good-approximations to sampled-input protocols. Intuitively, we require the output of both parties to be within distance d from the value of g applied to the sampled inputs of the parties, except with probability β.

Definition 49 (sampled-input good approximations)**.** *Let $g : \{0,1\}^s \times \{0,1\}^s \mapsto \mathbb{R}$ be a deterministic function, and let $\pi = (\mathsf{A}, \mathsf{B})$ be an oracle-aided, s-bit sampled-input protocol. The protocol π is a (β, d)-*SI-approximation for g with respect to a function family $\mathcal{F}$ and $\mathsf{P} \in \{\mathsf{A}, \mathsf{B}\}$*, if*

$$\Pr_{f \leftarrow \mathcal{F}, v \leftarrow \langle \pi^f \rangle} \left[\left| g\left(\mathrm{SInp}^{\mathsf{A}}(v), \mathrm{SInp}^{\mathsf{B}}(v) \right) - \mathrm{AOut}^{\mathsf{P}}(v) \right| > d \right] < \beta. \tag{30}$$

*Protocol π is a (β, d)-*SI-approximation for g with respect to $\mathcal{F}$*, if it is a (β, d)-SI-approximation for g with respect to $\mathcal{F}$ and both parties.*

We also extend the notion of differential privacy to sampled-input protocols.

Definition 50 (differential privacy sampled-input protocols)**.** *Let $\mathcal{F}$ be a function family and let $\pi = (\mathsf{A}, \mathsf{B})$ be an oracle-aided, s-bit sampled-input protocol. The protocol π is (k, α, γ)-*differentially private with respect to $\mathcal{F}$ and A*, if for every k-query, oracle-aided distinguisher D and every $x, x' \in \{0,1\}^s$ with $H_d(x, x') = 1$, it holds that*

$$\Pr_{f \leftarrow \mathcal{F}, v \leftarrow \langle \pi^f \rangle} \left[\mathsf{D}^f\left(\mathsf{trans}(v)\right) \mid \mathrm{SInp}^{\mathsf{A}}(v) = x \right] \leq e^{\alpha} \cdot \Pr_{f \leftarrow \mathcal{F}, v \leftarrow \langle \pi^f \rangle} \left[\mathsf{D}^f\left(\mathsf{trans}(v)\right) \mid \mathrm{SInp}^{\mathsf{A}}(v) = x' \right] + \gamma.$$

The differential privacy of π with respect to ($\mathcal{F}$ and) B is defined analogously.
*The protocol π is (k, α, γ)-*differentially private with respect to $\mathcal{F}$*, if it is (k, α, γ)-differentially private with respect to $\mathcal{F}$ and both parties.*

[19] Namely, $\mathrm{SInp}^{\mathsf{P}}(v) = \mathrm{out}^{\mathsf{P}}(v)_{1,\ldots,s}$ and $\mathrm{AOut}^{\mathsf{P}}(v) = \mathrm{out}^{\mathsf{P}}(v)_{s+1,\ldots}$.

Lower bound for no-oracle sampled-input protocols. The following proposition is a variant of Theorem 41, suited for no-oracle, sampled-input protocols.

Proposition 51. *For numbers $\nu > 0$ and $\alpha \geq 0$, there exist numbers $\lambda > 0$ and $z \in \mathbb{N}$ such that the following holds. Let $\pi = (\mathsf{A}, \mathsf{B})$ be a no-oracle, s-bit sampled-input protocol, let X_{In} and Y_{In} be the sampled inputs of A and B, respectively, and let X_{out} and Y_{out} be the actual outputs of A and B, respectively, induced by a random execution of π. Assume that both X_{In} and Y_{In} are uniformly distributed over $\{0,1\}^s$, and that π is (α, γ)-differentially private. If $s \geq z$, then*

$$\Pr\left[|Y_{\text{out}} - \mathsf{IP}(X_{\text{In}}, Y_{\text{In}})| \leq \Delta := \lambda \cdot \frac{\sqrt{s}}{\log s} \cdot \tau\right] \leq \tau$$

for every $1 \geq \tau \geq \max\{48s\gamma, \nu\}$.[17] *The same holds for X_{out}.*

The main difference between Proposition 51 and Theorem 41, is that Proposition 51 allows X_{In} and Y_{In} to be chosen during the protocol (and hence not necessarily be independent), where Theorem 41 assumes that the inputs are selected by an external entity (hence, needing to require independence of inputs). We observe that the proof of Theorem 41 given in [17], does not require a priori independence between X_{In} and Y_{In}, but only that they are independent given any transcript of the protocol. The latter holds, however, for any joint distribution for $(X_{\text{In}}, Y_{\text{In}})$, since the views of the parties (in the no-oracle model with no inputs) are always independent of each other, *given* the transcript. Indeed, Proposition 51 easily follows by slight adaptation to the proof of Theorem 41, given in [17]. For completeness, however, we include a proof (much of which, taken verbatim from [17]).

Let us first describe the outline of the proof given in [17] for Theorem 41 (Theorem A.5 in [17]), which is in turn the scheme of our proof. Their proof is twofold. The first part of it is a result about unpredictable bit sources, showing that it is possible to extract a uniform element in $\mathbb{Z}_m$ from the inner product between two independent unpredictable s-bit variables (even given one of these variables), provided that m is somewhat less than $\sqrt{s}$. We can use this result without reproving it (for the formal statement see Theorem 47). The second part of the proof deals with executions of (α, γ)-differentially private protocols, where the inputs of the parties are selected uniformly at random. It is shown that the input of each party in such executions, given the transcript of the execution, is close to an unpredictable bit-source. We reprove this part, with respect to sampled-input protocols. Finally, combining the above two results yields that every two-party differentially-private protocol for approximating the inner product function must incur an error of roughly $m \approx \sqrt{s}$. Indeed, if a significantly better approximation could be computed given the transcript (and one party's input), then the inner product would be concentrated in an interval of size significantly smaller than m, contradicting the fact that it reduces to an almost-uniform element of $\mathbb{Z}_m$.

Proof of Proposition 51. Let $\pi = (\mathsf{A}, \mathsf{B})$ be a no-oracle, s-bit sampled-input protocol, let X_{In} and Y_{In} be the sampled inputs of A and B, respectively, and let X_{out} and Y_{out} be the actual outputs of A and B, respectively, induced by a random execution of π. Let $\overline{T}$ be the communication transcript in a random execution of π, and for $\bar{t} \in \text{Supp}\left(\overline{T}\right)$ let $X_{\text{In} \mid \bar{t}}$ [resp., $Y_{\text{In} \mid \bar{t}}$] be the value of X_{In} [resp., Y_{In}] in such a random execution, conditioned on $\overline{T} = \bar{t}$. Assume that both X_{In} and Y_{In} are uniformly distributed over $\{0,1\}^s$ and that π is (α, γ)-differentially private. Let $\eta = e^{-(1.1+\alpha)}/3$ and $\beta = \log(1+\eta))$. Finally, fix $\nu > 0$ and $\tau \geq \max\{48s\gamma, \nu\}$.

The proof is carried via the following claims (proofs given below). In Claim 52 we show that by the differential privacy of π, it holds that $X_{\text{In} \mid \bar{t}}$ and $Y_{\text{In} \mid \bar{t}}$ are, on average, close to being unpredictable. In Claim 53 we define the constants λ and z so that we can apply Theorem 47 with respect to such sources.

Claim 52. *There are numbers $\{\gamma_{\bar{t}}\}_{\bar{t} \in \text{Supp}(\overline{T})}$ with $\mathrm{E}\left[\gamma_{\overline{T}}\right] \leq 4\gamma$, such that the following holds for every $\bar{t} \in \text{Supp}\left(\overline{T}\right)$: the random variable $X_{\text{In} \mid \bar{t}}$ [resp., $Y_{\text{In} \mid \bar{t}}$] is $2s\gamma_{\bar{t}}$-close to some η-unpredictable bit-source $\widehat{X}_{\bar{t}}$ [resp., $\widehat{Y}_{\bar{t}}$].*

Claim 53. *There are numbers $\lambda > 0$ and $z \in \mathbb{N}$, functions of α and ν, such that the following holds. Let $\Delta = \lambda \cdot \frac{\sqrt{s}}{\log s} \cdot \tau$, let $m = 6 \cdot \Delta / \tau$ and let c be the constant from Theorem 47. Assuming $s \geq z$, then $s \geq c \cdot \frac{m^2}{\eta\beta} \cdot \log\left(\frac{m}{\beta}\right) \cdot \log\left(\frac{m}{\tau/3}\right)$.*

We use the above claims for proving the proposition for Y_{out}, where the proof for X_{out} is analogous. Fix for a moment $\bar{t} \in \text{Supp}\left(\overline{T}\right)$, and note that $X_{\text{In} \mid \bar{t}}$ and $Y_{\text{In} \mid \bar{t}}$ are independent (since π is a no-input, no-oracle protocol). Let $\{\gamma_{\bar{t}}\}_{\bar{t} \in \text{Supp}(\overline{T})}$, λ, z, Δ and m, be the numbers from Claims 52 and 53. Claim 52 yields that

$$\text{SD}\left(\left(Y_{\text{In} \mid \bar{t}}, \mathsf{IP}(X_{\text{In} \mid \bar{t}}, Y_{\text{In} \mid \bar{t}}) \bmod m\right), \left(\widehat{Y}_{\bar{t}}, \mathsf{IP}(\widehat{X}_{\bar{t}}, \widehat{Y}_{\bar{t}}) \bmod m\right)\right) \leq 4s\gamma_{\bar{t}} \tag{31}$$

for some two (independent) η-unpredictable bit sources $\widehat{X}_{\bar{t}}$ and $\widehat{Y}_{\bar{t}}$. Note that by Fact 44, both $\widehat{X}_{\bar{t}}$ and $\widehat{Y}_{\bar{t}}$ have min-entropy βs.

Assume $s \geq z$. Since by Claim 53 $s \geq c \cdot \frac{m^2}{\eta\beta} \cdot \log\left(\frac{m}{\beta}\right) \cdot \log\left(\frac{m}{\tau/3}\right)$, Theorem 47 yields that

$$\text{SD}\left(\left(\widehat{Y}_{\bar{t}}, \mathsf{IP}(\widehat{X}_{\bar{t}}, \widehat{Y}_{\bar{t}}) \bmod m\right), \left(\widehat{Y}_{\bar{t}}, U_m\right)\right) \leq \tau/3, \tag{32}$$

where U_m is independently and uniformly distributed over $\mathbb{Z}_m$. Finally, combining Equations (31) and (32) yields that

$$\text{SD}\left(\left(Y_{\text{In} \mid \bar{t}}, \mathsf{IP}(X_{\text{In} \mid \bar{t}}, Y_{\text{In} \mid \bar{t}}) \bmod m\right), \left(\widehat{Y}_{\bar{t}}, U_m\right)\right) \leq 4s\gamma_{\bar{t}} + \tau/3 \tag{33}$$

for every $\bar{t} \in \text{Supp}\left(\overline{T}\right)$.

In the following we assume without loss of generality that B's output is a *deterministic* function f_{B} of $\left(Y_{\text{In}}, \overline{T}\right)$.[20]

Let $\mathcal{S} = \left\{\left(y, \bar{t}, z\right) \in \text{Supp}\left(Y_{\text{In}}, \overline{T}\right) \times \mathbb{R} \colon \left(f_{\mathsf{B}}\left(y, \bar{t}\right) - z \bmod m\right) \in \{m - \Delta, \ldots, 0, \ldots, \Delta\}\right\}$. It

[20] For an arbitrary function f_{B}, consider its variant f'_{B} that applies f_{B} on a random view that is consistent with $\left(Y_{\text{In}}, \overline{T}\right)$. Clearly, f'_{B} computes the inner product correctly with the same probability as f_{B} does, and its output is a randomized function of (only) $\left(Y_{\text{In}}, \overline{T}\right)$. Finally, the deterministic function f''_{B} that applies f'_{B} with the best choice of random coins, computes the inner product correctly no worse than f'_{B} does, and thus no worse than f_{B}.

follows that

$$\begin{aligned}
&\Pr\left[|Y_{\text{out}} - \mathsf{IP}(X_{\text{In}}, Y_{\text{In}})| \leq \Delta\right] \\
&\leq \Pr\left[(Y_{\text{In}}, \overline{T}, \mathsf{IP}(X_{\text{In}}, Y_{\text{In}})) \in \mathcal{S}\right] \\
&= \Pr\left[(Y_{\text{In}}, \overline{T}, \mathsf{IP}(X_{\text{In}}, Y_{\text{In}}) \bmod m) \in \mathcal{S}\right] \\
&\leq \Pr\left[(\widehat{Y}_{\overline{T}}, \overline{T}, U_m) \in \mathcal{S}\right] + \text{SD}\left((Y_{\text{In}}, \overline{T}, \mathsf{IP}(X_{\text{In}}, Y_{\text{In}}) \bmod m), (\widehat{Y}_{\overline{T}}, \overline{T}, U_m)\right) \\
&\leq \Pr\left[(\widehat{Y}_{\overline{T}}, \overline{T}, U_m) \in \mathcal{S}\right] + \text{E}[4s\gamma_{\overline{T}} + \tau/3] \\
&\leq 2\Delta/m + 16s\gamma + \tau/3 \\
&\leq \tau.
\end{aligned}$$

The second inequality holds by Equation (33), the third one since $\text{E}[\gamma_{\overline{T}}] \leq 4\gamma$, and the last one since $\frac{\Delta}{m} = \gamma/6$ and since, by assumption, $\tau \geq 48s\gamma$. □

Proof of Claim 52. Let X_j denote the j'th bit in X_{In}. For $i \in [s]$ and $\left(x, \bar{t}\right) \in \text{Supp}\left(X_{\text{In}}, \overline{T}\right)$, define

$$\begin{aligned}
\rho_X\left(i, x, \bar{t}\right) &:= \frac{\Pr\left[X_i = 0 \mid X_1 = x_1, \ldots X_{i-1} = x_{i-1}, X_{i+1} = x_{i+1}, \ldots, X_s = x_s, \overline{T} = \bar{t}\right]}{\Pr\left[X_i = 1 \mid X_1 = x_1, \ldots X_{i-1} = x_{i-1}, X_{i+1} = x_{i+1}, \ldots, X_s = x_s, \overline{T} = \bar{t}\right]} \\
&= \frac{\Pr\left[\overline{T} = \bar{t} \mid X_1 = x_1, \ldots X_{i-1} = x_{i-1}, X_i = 0, X_{i+1} = x_{i+1}, \ldots, X_s = x_s\right]}{\Pr\left[\overline{T} = \bar{t} \mid X_1 = x_1, \ldots X_{i-1} = x_{i-1}, X_i = 1, X_{i+1} = x_{i+1}, \ldots, X_s = x_s\right]},
\end{aligned} \tag{34}$$

where the equality holds by the uniformity of X_{In} (using Bayes' Rule), and let $\mathcal{S}_X = \left\{\left(i, x, \bar{t}\right) : \rho_X\left(i, x, \bar{t}\right) \notin \left[e^{-(1.1+\alpha)}, e^{(1.1+\alpha)}\right]\right\}$. Define $\mathcal{S}_Y$ analogously for Y_{In}. For $\bar{t} \in \text{Supp}\left(\overline{T}\right)$, set $\gamma_{\bar{t}} := \max\left\{\Pr_{i \leftarrow [s], x \leftarrow X_{\text{In} \mid \bar{t}}}\left[\left(i, x, \bar{t}\right) \in \mathcal{S}_X\right], \Pr_{i \leftarrow [s], y \leftarrow Y_{\text{In} \mid \bar{t}}}\left[\left(i, y, \bar{t}\right) \in \mathcal{S}_Y\right]\right\}$. It follows that $X_{\text{In} \mid \bar{t}}$ and $Y_{\text{In} \mid \bar{t}}$ are $(e^{-(1.1+\alpha)}, \gamma_{\bar{t}})$-strongly-unpredictable bit sources, and hence, Corollary 46 yields that both $X_{\text{In} \mid \bar{t}}$ and $Y_{\text{In} \mid \bar{t}}$ are $2s\gamma_{\bar{t}}$-close to some $(e^{-(1.1+\alpha)}/3)$-unpredictable bit sources, yielding the first requirement of the claim.

For the second requirement of the claim, applying Lemma 42 with $\nu = 1.1$ yields that

$$\begin{aligned}
\max\left\{\Pr_{(x,\bar{t}) \leftarrow \left(X_{\text{In}}, \overline{T}\right)}\left[\left(i, x, \bar{t}\right) \in \mathcal{S}_X\right], \Pr_{(y,\bar{t}) \leftarrow \left(Y_{\text{In}}, \overline{T}\right)}\left[\left(i, y, \bar{t}\right) \in \mathcal{S}_Y\right]\right\} &\leq \gamma \cdot \frac{1 + e^{-(1.1+\alpha)}}{1 - e^{-1.1}} \\
&< 2\gamma
\end{aligned} \tag{35}$$

for every $i \in [s]$,[21] and we conclude that

$$\begin{aligned}
\text{E}\left[\gamma_{\overline{T}}\right] &\leq \text{E}_{\bar{t} \leftarrow \overline{T}}\left[\Pr_{i \leftarrow [s], x \leftarrow X_{\text{In} \mid \bar{t}}}\left[\left(i, x, \bar{t}\right) \in \mathcal{S}_X\right] + \Pr_{i \leftarrow [s], y \leftarrow Y_{\text{In} \mid \bar{t}}}\left[\left(i, y, \bar{t}\right) \in \mathcal{S}_Y\right]\right] \\
&\leq \text{E}\left[2 \cdot \max\left\{\Pr_{(x,\bar{t}) \leftarrow \left(X_{\text{In}}, \overline{T}\right)}\left[\left(i, x, \bar{t}\right) \in \mathcal{S}_X\right], \Pr_{(y,\bar{t}) \leftarrow \left(Y_{\text{In}}, \overline{T}\right)}\left[\left(i, y, \bar{t}\right) \in \mathcal{S}_Y\right] : i \in [s]\right\}\right] \\
&< 4\gamma.
\end{aligned} \tag{36}$$

□

[21] We note that Lemma 42 is stated for differentially private mechanisms. Nevertheless, its proof for sampled-input protocols readily follows from the original proof.

Proof of Claim 53. Let $\lambda_1 = \lambda/\eta > 0$, where $\lambda \in (0,1)$ is determined later, and note that $s = \left(\frac{\Delta \cdot \log s}{\lambda_1 \cdot \tau \cdot \eta}\right)^2$. Let $z_1 = z_1(\lambda, \alpha)$ be such that $s \geq z_1$ implies $\log s \geq 1/\lambda_1$. Note that

$$\begin{aligned} s &\geq \frac{1}{\lambda_1^2} \cdot \left(\frac{\Delta}{\tau\eta}\right)^2 \cdot \left(\log\left(\frac{\Delta}{\tau\eta}\right)\right)^2 \\ &= \frac{1}{\lambda_1^2} \cdot \frac{m^2}{36 \cdot \eta^2} \cdot \left(\log\left(\frac{m}{6\eta}\right)\right)^2 \\ &\geq \frac{1}{36 \cdot \lambda_1^2} \cdot \frac{m^2}{\eta\beta} \cdot \left(\log\left(\frac{m}{6\eta}\right)\right)^2 \end{aligned} \tag{37}$$

for every $s \geq z_1$, where the last inequality holds since, by inspection, $\beta \geq \eta$.

Let $z_2 = z_2(\lambda, \alpha)$ be such that $s \geq z_2$ implies $\frac{m}{6\cdot\eta} \geq 2$, and let $z = \max\{z_1, z_2\}$. Let $\kappa = \kappa(\alpha, \nu) = 1 + \max\left\{\log\left(\frac{6\eta}{\beta}\right), \log\left(\frac{6\eta}{\nu/3}\right)\right\}$ and let $\lambda = \lambda(\alpha, \nu) \in (0,1)$ be such that $\frac{1}{36\cdot\lambda_1^2} \geq c \cdot \kappa^2$. Fix $s \geq z$. Since $\left(\frac{m}{6\eta}\right)^\kappa \geq \max\left\{\frac{m}{\beta}, \frac{m}{\nu/3}\right\}$, Equation (37) yields that

$$\begin{aligned} s &\geq c \cdot \kappa^2 \cdot \frac{m^2}{\eta\beta} \cdot \left(\log\left(\frac{m}{6\eta}\right)\right)^2 \\ &\geq c \cdot \frac{m^2}{\eta\beta} \cdot \log\left(\frac{m}{\beta}\right) \cdot \log\left(\frac{m}{\tau/3}\right), \end{aligned}$$

concluding the claim's proof. □

Lower bound for oracle-aided, sampled-input protocols. We now use Theorem 13 to give variant of Proposition 51 for (sampled input) *oracle-aided* protocols. We start by showing that the existence of differentially private, oracle-aided, sampled-input protocols implies the existence of no-oracle, sampled-input protocols, incurring no loss in privacy, and a minor loss in accuracy.

Lemma 54. *Let $\mathcal{F}$ be a function family, let π be an oracle-aided, s-bit sampled-input protocol, and let $g : \{0,1\}^s \times \{0,1\}^s \mapsto \mathbb{R}$ be a deterministic function. Assume that the pair $(\mathcal{F}, \pi)$ has a (T, ε)-*mapping*, and assume that π is a (β, d)-SI-approximation for g with respect to $\mathcal{F}$ and party P, and satisfies (T, α, γ)-differential privacy with respect to $\mathcal{F}$. Then, the no-oracle, s-bit sampled-input protocol $\tilde{\pi} = (\tilde{\mathsf{A}}, \tilde{\mathsf{B}})$, guaranteed by the (T, ε)-*mapping*, is a $(\beta + \varepsilon, d)$-SI-approximation for g with respect to party P, and is (α, γ)-differentially private.*

Furthermore, the sampled input of party A (resp. B) and the sampled input of party $\tilde{\mathsf{A}}$ (resp. $\tilde{\mathsf{B}}$) are identically distributed.

Proof. Let $\tilde{\pi} = (\tilde{\mathsf{A}}, \tilde{\mathsf{B}})$ and Map be the no-input, no-oracle protocol and oracle-aided algorithm guaranteed by Definition 12 with respect to π and $\mathcal{F}$. We first argue that $\tilde{\pi}$ satisfies (α, γ)-differential privacy. Assume to the contrary that this is not the case. Specifically, assume without loss of generality that there exists a (no-oracle) adversary $\tilde{\mathsf{D}}$, such that

$$\Pr_{\tilde{v} \leftarrow \langle \tilde{\pi} \rangle}\left[\tilde{\mathsf{D}}\left(\mathsf{trans}\left(\tilde{v}\right)\right) \mid \mathrm{SInp}^{\tilde{\mathsf{A}}}\left(\tilde{v}\right) = x\right] > e^{\alpha} \cdot \Pr_{\tilde{v} \leftarrow \langle \tilde{\pi} \rangle}\left[\tilde{\mathsf{D}}\left(\mathsf{trans}\left(\tilde{v}\right)\right) \mid \mathrm{SInp}^{\tilde{\mathsf{A}}}\left(\tilde{v}\right) = x'\right] + \gamma \tag{38}$$

for some $x, x' \in \{0,1\}^s$ with $H_d(x,x') = 1$. Consider the adversary D for π that on a given transcript $\bar{t}$ (of an execution of π with access to f) applies $\tilde{\mathsf{D}}$ to $\mathsf{Map}^f(\bar{t})$. We claim that

$$\Pr_{f\leftarrow\mathcal{F},v\leftarrow\langle\pi^f\rangle}\left[\mathsf{D}^f(\mathsf{trans}(v)) \mid \mathrm{SInp}^{\mathsf{A}}(v) = x\right] > e^{\alpha}\cdot\Pr_{f\leftarrow\mathcal{F},v\leftarrow\langle\pi^f\rangle}\left[\mathsf{D}^f(\mathsf{trans}(v)) \mid \mathrm{SInp}^{\mathsf{A}}(v) = x'\right] + \gamma \tag{39}$$

To see that Equation (39) holds, note that by the furthermore statement of the first item in Definition 12, the transcript together with the sampled input of $\tilde{\mathsf{A}}$ in a random execution of $\tilde{\pi}$ (i.e., $\left(\mathsf{trans}(\tilde{v}), \mathrm{SInp}^{\tilde{\mathsf{A}}}(\tilde{v})\right)$, where $\tilde{v} \leftarrow \langle\tilde{\pi}\rangle$), are (jointly) identically distributed as the value of Map applied to the transcript and the sampled input of A in a random execution of π (i.e., $\left(\mathsf{Map}^f(\mathsf{trans}(v)), \mathrm{SInp}^{\mathsf{A}}(v)\right)$, where $v \leftarrow \langle\pi^f\rangle$ for $f \leftarrow \mathcal{F}$). In addition, by Definition 12, D makes at most T oracle calls. Hence, we obtain a contradiction to the (T,α,γ)-differential privacy of π, yielding that the protocol $\tilde{\pi}$ must be (α,γ)-differentially private.

We conclude the proof by showing that $\tilde{\pi}$ is a good approximation for g with respect to any party $\mathsf{P} \in \left\{\tilde{\mathsf{A}}, \tilde{\mathsf{B}}\right\}$. Specifically, we show that

$$\Pr_{\tilde{v}\leftarrow\langle\tilde{\pi}\rangle}\left[\left|g\left(\mathrm{SInp}^{\tilde{\mathsf{A}}}(\tilde{v}), \mathrm{SInp}^{\tilde{\mathsf{B}}}(\tilde{v})\right) - \mathrm{AOut}^{\mathsf{P}}(\tilde{v})\right| \geq d\right] < \beta + \varepsilon. \tag{40}$$

By the first item in Definition 12, we have that the (actual) joint outputs of the parties in a random execution of π, are in statistical distance at most ε from the (actual) joint outputs of the parties in a random execution of $\tilde{\pi}$. Formally, if we let $\mathcal{D}_{\tilde{\pi}} = \left(\mathrm{SInp}^{\tilde{\mathsf{A}}}(\tilde{v}), \mathrm{SInp}^{\tilde{\mathsf{B}}}(\tilde{v}), \mathrm{AOut}^{\mathsf{P}}(\tilde{v})\right)_{\tilde{v}\leftarrow\langle\tilde{\pi}\rangle}$ and $\mathcal{D}_{\pi} = \left(\mathrm{SInp}^{\mathsf{A}}(v), \mathrm{SInp}^{\mathsf{B}}(v), \mathrm{AOut}^{\mathsf{P}}(v)\right)_{f\leftarrow\mathcal{F},v\leftarrow\langle\pi^f\rangle}$, then we have that $\mathrm{SD}(\mathcal{D}_{\tilde{\pi}}, \mathcal{D}_{\pi}) \leq \varepsilon$. Hence, Equation (40) follows from the accuracy of π, i.e., since we have that

$$\Pr_{f\leftarrow\mathcal{F},v\leftarrow\langle\pi^f\rangle}\left[\left|g\left(\mathrm{SInp}^{\mathsf{A}}(v), \mathrm{SInp}^{\mathsf{B}}(v)\right) - \mathrm{AOut}^{\mathsf{P}}(v)\right| \geq d\right] < \beta \tag{41}$$

To verify this, let $\mathcal{S} = \{(x,y,w) \in \mathrm{Supp}\,\mathcal{D}_{\tilde{\pi}} \colon |g(x,y) - w| \geq d\}$ and observe that the probability of falling into S according to $\mathcal{D}_{\tilde{\pi}}$ can be larger than the probability of falling into S according to $\mathcal{D}_{\pi}$ (which is bounded by β), by at most the statistical distance between $\mathcal{D}_{\tilde{\pi}}$ and $\mathcal{D}_{\pi}$.

The furthermore statement follows from the furthermore statement of the first item in Definition 12. □

We now combine Lemma 54 and Proposition 51 to prove a lower bound on the accuracy of oracle-aided, sampled-input protocols that are differentially private.

Proposition 55. *For numbers $\nu > 0$ and $\alpha \geq 0$, there exist numbers $\lambda > 0$ and $z \in \mathbb{N}$ such that the following holds. Let $\mathcal{F}$ be a function family and let $\pi = (\mathsf{A},\mathsf{B})$ be an oracle-aided, s-bit sample-input protocol, let X_{In} and Y_{In} be the sampled inputs of A and B, respectively, and let X_{out} and Y_{out} be the actual outputs of A and B, respectively, induced by a random execution of π. Assume that both X_{In} and Y_{In} are uniformly distributed over $\{0,1\}^s$.*

*Assume that π is (T,α,γ)-differentially private with respect to $\mathcal{F}$, and that the pair $(\mathcal{F},\pi)$ has a (T,ε)-*mapping*. If $s \geq z$, then*

$$\Pr\left[|Y_{\mathrm{out}} - \mathsf{IP}(X_{\mathrm{In}}, Y_{\mathrm{In}})| \leq \Delta := \lambda\cdot\frac{\sqrt{s}}{\log s}\cdot(\tau-\varepsilon)\right] \leq \tau$$

for every $1 \geq \tau$ such that $\tau - \varepsilon \geq \max\{48s\gamma, \nu\}$.[17] *The same holds for X_{out}.*

Proof. Given values for ν and α, set λ and z to be as in Proposition 51. Let $\mathcal{F}$ and π be as in the statement of the proposition. Since π is assumed to be (T, α, γ)-differentially private with respect to $\mathcal{F}$, and since the pair $(\mathcal{F}, \pi)$ is assumed to have a (T, ε)-mapping, it follows from Lemma 54 that the no-oracle, sampled-input protocol $\tilde{\pi}$ (guaranteed by this mapping) is (α, γ)-differentially private.

Since X_{In} and Y_{In} are uniformly distributed over $\{0,1\}^s$, it follows from the furthermore statement of Lemma 54 that the same holds for the sampled inputs of both parties in $\tilde{\pi}$. Hence, Proposition 51 yields that for $s \geq z$ and τ such that $\tau' := \tau - \varepsilon \geq \max\{48s\gamma, \nu\}$, it holds that $\tilde{\pi}$ is not a $(1-\tau', \Delta)$-SI-approximation for $\Delta = \lambda \cdot \frac{\sqrt{s}}{\log s} \cdot \tau'$. By Lemma 54, π is not a $(1-\tau'+\varepsilon, \Delta)$-SI-approximation, namely, $\Pr\left[|Y_{\text{out}} - \mathsf{IP}(X_{\text{In}}, Y_{\text{In}})| \leq \Delta\right] \leq \tau' - \varepsilon = \tau$. □

5.2.3 Limits on Uniform-Input Executions

The focus of this section is on executions of differentially private protocols, where the inputs of the parties are chosen uniformly at random. Towards proving a lower-bound on the accuracy of approximating the inner-product function in such executions, we next define a mapping from uniform-input executions to sampled-input protocols. Roughly speaking, we let the parties in the sampled-input protocol sample their inputs at random.

Notation 56 (the sampled-input variant $\mu(\pi)$)**.** *Given an s-bit input, (possibly, oracle-aided) protocol $\pi = (\mathsf{A}, \mathsf{B})$, let $\mu(\pi) = (\mu(\mathsf{A}), \mu(\mathsf{B}))$ denote the following s-bit sampled-input protocol: The parties $\mu(\mathsf{A})$ and $\mu(\mathsf{B})$ interact in an execution of $(\mathsf{A}(x_{\mathsf{A}}; r_{\mathsf{A}}), \mathsf{B}(x_{\mathsf{B}}; r_{\mathsf{B}}))$, taking the roles of A and B respectively, where x_{A} [resp., x_{B}] is the first s bits of $\mu(\mathsf{A})$'s [resp., $\mu(\mathsf{B})$'s] coins, and r_{A} [resp., r_{B}] is the rest of $\mu(\mathsf{A})$'s [resp., $\mu(\mathsf{B})$'s] coins. Let a and b be the outputs of A and B, respectively, in this execution, then the outputs of $\mu(\mathsf{A})$ and $\mu(\mathsf{B})$ will be (x_{A}, a) and (x_{B}, b), respectively.*

We next define what it means for a protocol to approximate a function with good probability when the inputs of the parties are uniformly selected.

Definition 57 (good random-approximations)**.** *Let $g : \{0,1\}^s \times \{0,1\}^s \mapsto \mathbb{R}$ be a deterministic function, and let $\pi = (\mathsf{A}, \mathsf{B})$ be an oracle-aided, s-bit input protocol. Protocol π is a (β, d)-*random-approximation for g with respect to a function family $\mathcal{F}$ and $\mathsf{P} \in \{\mathsf{A}, \mathsf{B}\}$*, if*

$$\Pr_{f \leftarrow \mathcal{F}, x, y \leftarrow \{0,1\}^s, v \leftarrow \langle \pi^f(x,y) \rangle} \left[\left| g(x,y) - \text{out}^{\mathsf{P}}(v) \right| > d \right] < \beta. \tag{42}$$

*Protocol π is a (β, d)-*random-approximation for g with respect to $\mathcal{F}$*, if it is a (β, d)-random-approximation for g with respect to $\mathcal{F}$ and both parties.*

The following observation allows us to use the lower bound stated in Lemma 54, to derive a similar bound for with-input protocols, when the inputs of the parties are chosen uniformly at random.

Lemma 58. *Let $g : \{0,1\}^s \times \{0,1\}^s \mapsto \mathbb{R}$ be a deterministic function and let $\mathcal{F}$ be some oracle family. Assume that there exists an oracle-aided, s-bit input protocol $\pi = (\mathsf{A}, \mathsf{B})$ that is a (β, d)-random-approximation for g with respect to $\mathcal{F}$ and party P, and is (k, α, γ)-differential privacy with respect to $\mathcal{F}$. Then $\mu(\pi)$ is a (β, d)-SI-approximation for g with respect to $\mathcal{F}$ and P, and (k, α, γ)-differentially private with respect to $\mathcal{F}$.*

Proof. Immediate by definition. □

Combining Proposition 55 and Lemma 58 yields the following result.

Proposition 59. *For numbers $\nu > 0$ and $\alpha \geq 0$, there exist numbers $\lambda > 0$ and $z \in \mathbb{N}$ such that the following holds. Let $\mathcal{F}$ be a function family and let $\pi = (\mathsf{A}, \mathsf{B})$ be an oracle-aided, s-bit input protocol. Let X_{In} and Y_{In} be the inputs of A and B, respectively, and let X_{out} and Y_{out} be the outputs of A and B, respectively, induced by a random execution of π. Assume that both X_{In} and Y_{In} are independently and uniformly chosen from $\{0,1\}^s$.*

*Assume that π is (T, α, γ)-differentially private with respect to $\mathcal{F}$, and that the pair $(\mathcal{F}, \mu(\pi))$ has a (T, ε)-*mapping *(where $\mu(\pi)$ is sampled-input variant of π). If $s \geq z$, then*

$$\Pr\left[|Y_{\text{out}} - \mathsf{IP}(X_{\text{In}}, Y_{\text{In}})| \leq \Delta := \lambda \cdot \frac{\sqrt{s}}{\log s} \cdot (\tau - \varepsilon)\right] \leq \tau$$

for every $1 \geq \tau$ such that $\tau - \varepsilon \geq \max\{48s\gamma, \nu\}$.[17] *The same holds for X_{out}.*

Proof. Given values for ν and α, set λ and z to be as in Proposition 55. Let $\mathcal{F}$ and π be as in the statement of the proposition. Let $\mu(\pi)$ be the (oracle-aided) sampled-input variant of π (see Notation 56). By construction, the sampled inputs of both parties in $\mu(\pi)$ are uniformly distributed. Since π is assumed to be (T, α, γ)-differentially private with respect to $\mathcal{F}$, it follows by Lemma 58 that $\mu(\pi)$ is also (T, α, γ)-differentially private with respect to $\mathcal{F}$. Since the pair $(\mathcal{F}, \mu(\pi))$ is assumed to have a (T, ε)-mapping, it follows from Proposition 55 that for $s \geq z$ and τ such that $\tau' := \tau - \varepsilon \geq \max\{48s\gamma, \nu\}$, the protocol $\mu(\pi)$ is not a $(1 - \tau' + \varepsilon, \Delta)$-SI-approximation for $\Delta = \lambda \cdot \frac{\sqrt{s}}{\log s} \cdot \tau'$. Hence, by Lemma 58, π is not a $(1 - \tau' + \varepsilon, \Delta)$-random-approximation, namely, $\Pr[|Y_{\text{out}} - \mathsf{IP}(X_{\text{In}}, Y_{\text{In}})| \leq \Delta] \leq \tau' - \varepsilon = \tau$. □

5.2.4 Limits on Arbitrary Protocols

The results presented above give a lower-bound on the accuracy of differentially private protocols, with respect to executions where inputs are selected uniformly at random. Indeed, such a lower-bound easily implies a similar lower bound for arbitrary executions of such protocols. Intuitively, this is because if a protocol errs with probability β on uniform inputs, then there must be a specific choice of inputs for the parties on which the protocol errs with probability at least β. We next give a formal statement.

Proposition 60. *For numbers $\nu > 0$ and $\alpha \geq 0$, there exist numbers $\lambda > 0$ and $z \in \mathbb{N}$ such that the following holds. Let $\mathcal{F}$ be a function family and let $\pi = (\mathsf{A}, \mathsf{B})$ be an oracle-aided, s-bit input protocol. Assume that π is (T, α, γ)-differentially private with respect to $\mathcal{F}$, and that the pair $(\mathcal{F}, \mu(\pi))$ has a (T, ε)-*mapping *(where $\mu(\pi)$ is sampled-input variant of π).*

If $s \geq z$, then for every $\mathsf{P} \in \{\mathsf{A}, \mathsf{B}\}$ there exist $x, y \in \{0,1\}^s$, such that

$$\Pr_{f \leftarrow \mathcal{F}, v \leftarrow \langle \pi^f(x,y) \rangle}\left[\left|\text{out}^{\mathsf{P}}(v) - \mathsf{IP}(x,y)\right| \leq \Delta := \lambda \cdot \frac{\sqrt{s}}{\log s} \cdot (\tau - \varepsilon)\right] \leq \tau \tag{43}$$

for every $\tau \leq 1$ such that $\tau - \varepsilon \geq \max\{48s\gamma, \nu\}$.[17] *The same holds for X_{out}.*

Proof. Immediate, by taking x and y that maximize the probability in Equation (43), and using Proposition 59 to bound this probability from below. □

Combining Proposition 60 and theorem 13 yields the following corollary.

Corollary 61. *Let $\mathcal{F}$ be a simple function family. For numbers $0 < \nu < 1$ and $\alpha \geq 0$, there exist numbers $\lambda > 0$ and $z \in \mathbb{N}$ such that, for $s \geq z$, the following holds. Assume that π is an s-bit input, ℓ-query oracle-aided protocol that is (k, α, γ)-differentially private with respect to $\mathcal{F}$, with $k > 2^{10} \cdot \left(\frac{\ell}{1-\nu}\right)^2$ and $\gamma \leq \frac{\nu}{48 \cdot s}$. Then, π is* not *a (β, d)-approximation with respect to $\mathcal{F}$ for the inner-product function, with $\beta < \frac{1-\nu}{2}$ and $d \leq \lambda \cdot \nu \cdot \frac{\sqrt{s}}{\log s}$.*

Proof. For numbers $0 < \nu < 1$ and $\alpha \geq 0$, let λ and z be as in Proposition 60. Let $\mathcal{F}$ be a simple function family and let π be an s-bit input, ℓ-query oracle-aided protocol. Let $\mu(\pi)$ be the (oracle-aided) sampled-input variant of π (see Notation 56). By construction, $\mu(\pi)$ is an ℓ-query, oracle-aided, no-input protocol. Finally, let $\varepsilon = \frac{1-\nu}{2}$.

Theorem 13 yields that $(\mathcal{F}, \mu(\pi))$ has a (T, ε)-mapping for $T = 256 \cdot \left(\frac{\ell}{\varepsilon}\right)^2 = 2^{10} \cdot \left(\frac{\ell}{1-\nu}\right)^2$.

Let γ be such that $\gamma \leq \frac{\nu}{48 \cdot s}$. Taking $\tau = \nu + \varepsilon$, it follows that $\tau - \varepsilon \geq \max\{48s\gamma, \nu\}$. Hence for $k \geq T$, Proposition 60 yields that if π is (k, α, γ)-differentially private with respect to $\mathcal{F}$, then it is not a (β, d)-approximation for the inner-product function with respect to $\mathcal{F}$, whenever $d \leq \lambda \cdot \frac{\sqrt{s}}{\log s} \cdot (\tau - \varepsilon) = \lambda \cdot \nu \cdot \frac{\sqrt{s}}{\log s}$ and $\beta \leq 1 - \tau = 1 - \nu - \varepsilon$. Plugging in the value of ε, the latter holds whenever $\beta \leq \frac{1-\nu}{2}$. □

5.3 Applications to Random Functions, and Black-Box Reductions to One-way Functions

In this section we show that the all-function family is simple. and therefore Theorem 13 holds with respect to this family. We then use this fact to give limits on black-box reductions to one-way functions.

5.3.1 Standard Definitions and Known Facts

One-way functions and the all-function family. An efficiently computable function is one-way, if it is hard to invert on a random output.

Definition 62 (one-way functions)**.** *A polynomially-time computable function $f : \{0,1\}^* \mapsto \{0,1\}^*$ is* one-way*, if*

$$\Pr_{x \leftarrow \{0,1\}^n}[\mathsf{A}(1^n, f(x)) \in f^{-1}(f(x))] = \mathrm{neg}(n)$$

for any PPTM A.

We define the all-function family over a given input length, as the set of all length-preserving functions over this input length.

Definition 63 (the all-function family)**.** *For $n \in \mathbb{N}$, let $\mathcal{F}_{\mathrm{AF}_n}$ be the family of all functions from n-bit strings to n-bit strings.*

It is well known (cf., [13, 9]) that random members of the all-function family are "one-way". Specifically, we use the following fact.

Fact 64. *For any $(2^{n/3}-1)$-query oracle-aided algorithm A, it holds that*

$$\Pr_{f\leftarrow\mathcal{F}_{\mathrm{AF}n}}\left[\Pr_{x\leftarrow\{0,1\}^n}[\mathsf{A}^f(f(x))\in f^{-1}(f(x))]>2^{-n/3}\right]\leq 2^{-n/3}.$$

Proof. It is easy to verify that

$$\Pr_{f\leftarrow\mathcal{F}_{\mathrm{AF}n},;x\leftarrow\{0,1\}^n}\left[\mathsf{A}^f(f(x))\in f^{-1}(f(x))\right]\leq 2^{n/3}/2^{-n}=2^{-2n/3},$$

and the proof follows by a straightforward averaging argument. □

Black-box reductions. Loosely speaking, a fully black-box reduction from a primitive Q (e.g., key agreement protocol) to a primitive P (e.g., one-way function) is: (1) a construction of Q out of P that "ignores" the structure of the implementation of P (i.e., uses it as a "black box"), and (2) a *generic* reduction from the security of P to that of Q. In more details, such a reduction consists of a PPTM pair (Q,R) such that the following holds. (1) for every correct implementation P of P, it holds that Q^{P} is a correct implementation of Q, and (2) for every adversary A that breaks (the security of) Q^{P}, it holds that $\mathsf{R}^{\mathsf{P},\mathsf{A}}$ breaks P. See [21] for a more formal discussion.

Cryptographic primitives are typically parameterized by the so called security parameter, which determines their security and functionality (e.g., the key length of the key-agreement protocol). For such primitives we consider a restricted form of black-box reductions that requires the reduction, and in particular the security proof R, to work for *every* choice of the security parameter n, e.g., an algorithm that guesses the agreed key of the key-agrement protocol "too well" on security parameter n, can be used by the reduction to invert the one-way function on inputs of length n. See Definitions 67 and 69 for concrete examples.[22]

5.3.2 The All-Function Family is Simple

Claim 65. *For every $n\in\mathbb{N}$, the family $\mathcal{F}_{\mathrm{AF}n}$ is simple.*

In the following we fix n and let $\mathcal{F}_{\mathrm{AF}}=\mathcal{F}_{\mathrm{AF}n}$. Claim 65 is proved via the following claim:

Claim 66. *Let $\pi=(\mathsf{A},\mathsf{B})$ be a no-input, oracle-aided protocol, let $\mathcal{I}\in\{0,1\}^n\times\{0,1\}^n$ be a set query/answer pairs, let $(v_{\mathsf{A}},\cdot),(\cdot,v_{\mathsf{B}})\in\mathrm{Supp}(\mathcal{VIEW}^{\mathcal{F}_{\mathrm{AF}},\pi}(\cdot,\mathcal{I}))$ and let $\mathcal{I}_{\mathsf{A}}$ [resp., $\mathcal{I}_{\mathsf{B}}$] be the set of query/answer pairs that A makes in v_{A} [resp., B makes in v_{B}]. The following hold for every $(q,a)\in\mathcal{I}_{\mathsf{A}}$ (an analogous observation holds for $(q,a)\in\mathcal{I}_{\mathsf{B}}$):*

1. *If $(v_{\mathsf{A}},v_{\mathsf{B}})$ is* consistent *(i.e., $\Pr_\Omega[v_{\mathsf{A}},v_{\mathsf{B}}]>0$), then*

$$\Pr_{f\leftarrow\left(\mathcal{F}_{\mathrm{AF}}|\mathcal{I}_{\mathsf{A}}^q\cup\mathcal{I}_{\mathsf{B}}^q\cup\mathcal{I}\right)}[f(q)=a]\geq\Pr_{f\leftarrow\left(\mathcal{F}_{\mathrm{AF}}|\mathcal{I}_{\mathsf{A}}^q\cup\mathcal{I}\right)}[f(q)=a],$$

   where $\mathcal{I}_{\mathsf{A}}^q$ [resp., $\mathcal{I}_{\mathsf{B}}^q$] is the set of queries/answers asked in v_{A} [resp., v_{B}] before q is asked in v_{A}.

[22] We choose to focus on this simpler form of black-box reductions as it simplifies the proofs given in Sections 5.3.3 and 5.3.4, and still seems to capture the same set of known reductions captured by the standard notion of black-box reductions.

2. *If* $\text{Intersect}_{\mathcal{I}}(v_{\mathsf{A}}, v_{\mathsf{B}}) = 0$, *then*

$$\Pr_{f \leftarrow (\mathcal{F}_{\text{AF}} | \mathcal{I}_{\mathsf{A}}^q \cup \mathcal{I}_{\mathsf{B}}^q \cup \mathcal{I})} [f(q) = a] = \Pr_{f \leftarrow (\mathcal{F}_{\text{AF}} | \mathcal{I}_{\mathsf{A}}^q \cup \mathcal{I})} [f(q) = a] .$$

Proof. The proof of Item 2 is immediate. For proving Item 1, assume first that $(q, \cdot) \in \mathcal{I}_{\mathsf{B}}^q$. Since the views are consistent, it follows that $\Pr_{f \leftarrow (\mathcal{F}_{\text{AF}} | \mathcal{I}_{\mathsf{A}}^q \cup \mathcal{I}_{\mathsf{B}}^q \cup \mathcal{I})} [f(q) = a] = 1$ and the claim follows for this case. For the other case ($(q, \cdot) \notin \mathcal{I}_{\mathsf{B}}^q$), it is easy to verify that under this assumption $\Pr_{f \leftarrow (\mathcal{F}_{\text{AF}} | \mathcal{I}_{\mathsf{A}}^q \cup \mathcal{I}_{\mathsf{B}}^q \cup \mathcal{I})} [f(q) = a] = \Pr_{f \leftarrow (\mathcal{F}_{\text{AF}} | \mathcal{I}_{\mathsf{A}}^q \cup \mathcal{I})} [f(q) = a]$, concluding the claim's proof. □

Proof of Claim 65. Claim 66(2) immediately yields that $\text{Intersect}_{\mathcal{I}}(v_{\mathsf{A}}, v_{\mathsf{B}}) = 0$ implies $\text{Dependent}_{\mathcal{I}}(v_{\mathsf{A}}, v_{\mathsf{B}}) = 0$. For the other direction ($\text{Intersect}_{\mathcal{I}}(v_{\mathsf{A}}, v_{\mathsf{B}}) = 1$ implies $\text{Dependent}_{\mathcal{I}}(v_{\mathsf{A}}, v_{\mathsf{B}}) = 1$), we assume $\text{Intersect}_{\mathcal{I}}(v_{\mathsf{A}}, v_{\mathsf{B}}) = 1$ and show that either $\alpha^{\mathcal{I}}_{v_{\mathsf{A}} | v_{\mathsf{B}}} \neq \alpha^{\mathcal{I}}_{v_{\mathsf{A}}}$ or $\alpha^{\mathcal{I}}_{v_{\mathsf{B}} | v_{\mathsf{A}}} \neq \alpha^{\mathcal{I}}_{v_{\mathsf{B}}}$.

We distinguish between two cases: assuming that $(v_{\mathsf{A}}, v_{\mathsf{B}})$ are *not* consistent, it follows that $\Pr_{f \leftarrow \mathcal{F}_{\text{AF}}} [\mathcal{I}, \mathcal{I}_{\mathsf{A}}, \mathcal{I}_{\mathsf{B}}] = 0$ and therefore $\alpha^{\mathcal{I}}_{v_{\mathsf{A}} | v_{\mathsf{B}}} = 0$. But since v_{A} is consistent with $\mathcal{I}$, it holds that $\alpha^{\mathcal{I}}_{v_{\mathsf{A}}} \neq 0$, yielding that $\text{Dependent}_{\mathcal{I}}(v_{\mathsf{A}}, v_{\mathsf{B}}) = 1$.

We complete the proof by considering the case that $(v_{\mathsf{A}}, v_{\mathsf{B}})$ are consistent. Let $(q, a) \in (\mathcal{I}_{\mathsf{A}} \cap \mathcal{I}_{\mathsf{B}}) \setminus \mathcal{I}$ be a query/answer pair whose existence guaranteed by the intersection assumption, and assume without loss of generality that q was asked first by B. It follows that

1. $\Pr_{f \leftarrow (\mathcal{F}_{\text{AF}} | \mathcal{I}_{\mathsf{A}}^q \cup \mathcal{I}_{\mathsf{B}}^q \cup \mathcal{I})} [f(q) = a] = 1$, and
2. $\Pr_{f \leftarrow (\mathcal{F}_{\text{AF}} | \mathcal{I}_{\mathsf{A}}^q \cup \mathcal{I})} [f(q) = a] = 2^{-n}$.

Applying Claim 66(1) yields that $\alpha^{\mathcal{I}}_{v_{\mathsf{A}} | v_{\mathsf{B}}} \geq 2^n \cdot \alpha^{\mathcal{I}}_{v_{\mathsf{A}}}$, and therefore $\text{Dependent}_{\mathcal{I}}(v_{\mathsf{A}}, v_{\mathsf{B}}) = 1$. □

5.3.3 Key Agreement Protocols

Following Definition 34 and the discussion in Section 5.3.1, we define fully black-box reduction from key-agreement protocols to one-way functions as follows.

Definition 67 (fully black-box reduction from key agreement to one-way functions)**.** *A* PPTM *triplet* $(\mathsf{A}, \mathsf{B}, \mathsf{R})$ *is a fully black-box reduction from an* (α, γ)*-key-agreement protocol to one-way functions, where* α *and* γ *are functions over* $\mathbb{N}$*, if the following holds for every* $n \in \mathbb{N}$ *and every function* f *over* $\{0,1\}^n$*.*

1. $\Pr_{v \leftarrow \langle (\mathsf{A}^f, \mathsf{B}^f)(1^n) \rangle} \left[\text{out}^{\mathsf{A}}(v) = \text{out}^{\mathsf{B}}(v) \right] \geq 1 - \alpha(n)$,
2. *For every algorithm* D *and* $\delta > 0$ *such that* $\Pr_{v \leftarrow \langle (\mathsf{A}^f, \mathsf{B}^f)(1^n) \rangle} \left[\mathsf{D}(\mathsf{trans}(v)) = \text{out}^P(v) \right] \geq \gamma + \delta$ *for some* $P \in \{\mathsf{A}, \mathsf{B}\}$*, algorithm* $\mathsf{R}^{\mathsf{D}, f}$ *inverts* f *with probability at least* $p(\delta)$*, where* $p \in \text{poly}$ *is universal.*

Combining Corollary 37, Fact 64, and Claim 65, yields the following result.

Corollary 68. *There exists no fully black-box reduction from an* (α, γ)*-key-agreement protocol to one-way functions, with* $1 - \alpha(n) - \gamma(n) > 1/\text{poly}(n)$*.*

Proof's sketch. Assume there exists a fully black-box reduction $(\mathsf{A},\mathsf{B},\mathsf{R})$ from an (α,γ)-key-agreement protocol to one-way functions, with $1-\alpha(n)-\gamma(n) > 1/\operatorname{poly}(n)$. By Corollary 37 and Claim 65 and a simple average argument there exists a $\operatorname{poly}(n)$-query algorithm D such that

$$\Pr_{f\leftarrow\mathcal{F}_{\mathrm{AF}\,n}}\left[\Pr_{v\leftarrow\langle(\mathsf{A}^f,\mathsf{B}^f)(1^n)\rangle}\left[\mathsf{D}^f(\mathsf{trans}(v))=\mathrm{out}^P(v)\right]\geq\gamma(n)+1/\operatorname{poly}(n)\right]\geq 1/\operatorname{poly}(n) \tag{44}$$

It follows that $\Pr_{f\leftarrow\mathcal{F}_{\mathrm{AF}\,n}}\left[\Pr_{x\leftarrow\{0,1\}^n}[\mathsf{R}^{\mathsf{D}^f,f}(f(x))\in f^{-1}(f(x))]>1/\operatorname{poly}(n)\right]>1/\operatorname{poly}(n)$, in contradiction to Fact 64. □

5.3.4 Differentially Private Two-Party Computation

Following Definitions 39 and 40 and the discussion in Section 5.3.1, we define fully black-box reduction from differentially private protocols to one-way functions as follows.

Definition 69 (fully black-box reduction from differentially private protocols to one-way functions)**.** *A* PPTM *triplet* $(\mathsf{A},\mathsf{B},\mathsf{R})$ *is a fully black-box reduction from an* $(s,\beta,d,\alpha,\gamma)$*-differentially private protocol for a function* g *to one-way functions, where* s*,* β*,* d*,* α *and* γ *are functions over* $\mathbb{N}$*, if the following holds for every* $n\in\mathbb{N}$ *and every function* f *over* $\{0,1\}^n$*.*

1. *For every* $x,y\in\{0,1\}^{s(n)}$ *and* $\mathsf{P}\in\{\mathsf{A},\mathsf{B}\}$*, it holds that*

$$\Pr_{v\leftarrow\langle(\mathsf{A}^f(x),\mathsf{B}^f(y))\rangle}\left[\left|g(x,y)-\mathrm{out}^{\mathsf{P}}(v)\right|>d(n)\right]<\beta(n).$$

2. *For every algorithm* D *and* $\delta>0$ *such that*

$$\Pr_{v\leftarrow\langle(\mathsf{A}^f(x),\mathsf{B}^f(y))\rangle}[\mathsf{D}(\mathsf{trans}(v))=1]\geq e^{\alpha(n)}\cdot\Pr_{v\leftarrow\langle(\mathsf{A}^f(x'),\mathsf{B}^f(y))\rangle}[\mathsf{D}(\mathsf{trans}(v))=1]+\gamma(n)+\delta$$

for some $x,x',y\in\{0,1\}^{s(n)}$ *with* $H_d(x,x')=1$*, or the analogue condition holds for some* $y,y',x\in\{0,1\}^{s(n)}$ *with* $H_d(y,y')=1$*, algorithm* $\mathsf{R}^{\mathsf{D},f}$ *inverts* f *with probability at least* $p(\delta)$*, where* $p\in\operatorname{poly}$ *is universal.*

Combining Corollary 61, Fact 64, and Claim 65 yields the following result.

Corollary 70. *For constants* $\nu\in(0,1)$ *and* $\eta\geq 0$*, there exist* $\lambda>0$ *such that the following holds. There exists no fully black-box reduction from an* $(s,\beta,d,\alpha,\gamma)$*-differentially private protocol for inner product to one-way functions, with* $\alpha(n)\leq\eta$*,* $\gamma(n)\leq\frac{\nu}{48\cdot s(n)}-1/\operatorname{poly}(n)$*,* $\beta(n)\leq\frac{1-\nu}{2}$ *and* $d(n)\leq\lambda\cdot\nu\cdot\frac{\sqrt{s(n)}}{\log s(n)}$ *for infinitely many* n*'s.*

References

[1] B. Barak and M. Mahmoody. Merkle puzzles are optimal - an $O(n^2)$-query attack on any key exchange from a random oracle. In *Advances in Cryptology – CRYPTO '09*, pages 374–390, 2009.

[2] A. Beimel, K. Nissim, and E. Omri. Distributed private data analysis: On simultaneously solving how and what. *CoRR*, abs/1103.2626, 2011.

[3] R. Canetti, O. Goldreich, and S. Halevi. On the random-oracle methodology as applied to length-restricted signature schemes. In *Theory of Cryptography, First Theory of Cryptography Conference, TCC 2004*, 2004.

[4] Y.-C. Chang, C.-Y. Hsiao, and C.-J. Lu. On the impossibilities of basing one-way permutations on central cryptographic primitives. In *Advances in Cryptology – CRYPTO '02*, pages 110–124, 2002.

[5] D. Dachman-Soled, Y. Lindell, M. Mahmoody, and T. Malkin. On the black-box complexity of optimally-fair coin tossing. In *tcc11*, pages 450–467, 2011.

[6] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006*, pages 265–284, 2006.

[7] A. Fiat and A. Shamir. How to prove yourself: practical solutions to identification and signature problems. In *Advances in Cryptology – CRYPTO '86*, pages 186–194, 1987.

[8] R. Gennaro and L. Trevisan. Lower bounds on the efficiency of generic cryptographic constructions. In *Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, pages 305–313, 2000.

[9] R. Gennaro, Y. Gertner, J. Katz, and L. Trevisan. Bounds on the efficiency of generic cryptographic constructions. *SIAM Journal on Computing*, 35(1):217–246, 2005.

[10] Y. Gertner, S. Kannan, T. Malkin, O. Reingold, and M. Viswanathan. The relationship between public key encryption and oblivious transfer. In *Proceedings of the 32nd Annual ACM Symposium on Theory of Computing (STOC)*, 2000.

[11] S. Goldwasser and Y. Tauman-Kalai. On the (in)security of the fiat-shamir paradigm. In *Proceedings of the 44th Annual Symposium on Foundations of Computer Science (FOCS)*, 2003.

[12] I. Haitner, J. J. Hoch, O. Reingold, and G. Segev. Finding collisions in interactive protocols – A tight lower bound on the round complexity of statistically-hiding commitments. In *Proceedings of the 48th Annual Symposium on Foundations of Computer Science (FOCS)*, 2007.

[13] R. Impagliazzo and S. Rudich. Limits on the provable consequences of one-way permutations. In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing (STOC)*, pages 44–61. ACM Press, 1989.

[14] J. Kahn, M. Saks, and C. Smyth. A dual version of reimer's inequality and a proof of rudich's conjecture. In *Computational Complexity, 2000. Proceedings. 15th Annual IEEE Conference on*, pages 98 –103, 2000.

[15] J. H. Kim, D. Simon, and P. Tetali. Limits on the efficiency of one-way permutation-based hash functions. In *Foundations of Computer Science, 1999. 40th Annual Symposium on*, pages 535 –542, 1999.

[16] M. Mahmoody, H. K. Maji, and M. Prabhakaran. Limits of random oracles in secure computation. Technical Report 1205.3554v1, arXiv, 2012. arXiv:1205.3554v1.

[17] A. McGregor, I. Mironov, T. Pitassi, O. Reingold, K. Talwar, and S. P. Vadhan. The limits of two-party differential privacy. *Electronic Colloquium on Computational Complexity (ECCC)*, page 106, 2011. Preliminary version in *FOCS'10*.

[18] R. C. Merkle. Secure communications over insecure channels. In *SIMMONS: Secure Communications and Asymmetric Cryptosystems*, 1982.

[19] I. Mironov, O. Pandey, O. Reingold, and S. P. Vadhan. Computational differential privacy. In *Advances in Cryptology – CRYPTO '09*, pages 126–142, 2009.

[20] D. Pointcheval and J. Stern. Security proofs for signature schemes. In *Advances in Cryptology – EUROCRYPT '96*, pages 387–398, 1996.

[21] O. Reingold, L. Trevisan, and S. P. Vadhan. Notions of reducibility between cryptographic primitives. In *Theory of Cryptography, First Theory of Cryptography Conference, TCC 2004*, volume 2951 of *Lecture Notes in Computer Science*, pages 1–20. Springer, 2004.

[22] S. Rudich. The use of interaction in public cryptosystems. In *Proceedings of the 11th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '91, pages 242–251, 1992.

[23] M. Santha and U. V. Vazirani. Generating quasi-random sequences from semi-random sources. *J. Comput. Syst. Sci.*, 33(1):75–87, 1986.

[24] D. Simon. Finding collisions on a one-way street: Can secure hash functions be based on general assumptions? In *Advances in Cryptology – EUROCRYPT '98*, pages 334–345, 1998.

[25] H. Wee. One-way permutations, interactive hashing and statistically hiding commitments. In *Theory of Cryptography, First Theory of Cryptography Conference, TCC 2004*, pages 419–433, 2007.