J Cryptol (2016) 29:283–335
DOI: 10.1007/s00145-014-9194-9

Journal of CRYPTOLOGY

Limits on the Usefulness of Random Oracles*

Iftach Haitner†
School of Computer Science, Tel Aviv University, Tel Aviv, Israel
iftachh@cs.tau.ac.il

Eran Omri‡
Department of Computer Science and Mathematics, Ariel University, Ariel, Israel
omrier@ariel.ac.il

Hila Zarosim§
Department of Computer Science, Bar Ilan University, Ramat Gan, Israel
zarosih@cs.biu.ac.il

Communicated by Jonathan Katz.

Received 14 January 2013
Online publication 24 December 2014

Abstract. In their seminal work, Impagliazzo and Rudich (STOC'89) showed that no key-agreement protocol exists in the random-oracle model, yielding that key agreement cannot be black-box reduced to one-way functions. In this work, we generalize their result, showing that, to a large extent, no-private-input, semi-honest, two-party functionalities that can be securely implemented in the random oracle model can be securely implemented *information theoretically* (where parties are assumed to be all powerful, and no oracle is given). Using a recent information-theoretic impossibility result by McGregor et al. (FOCS'10), our result yields that certain functionalities (e.g. inner product) cannot be computed both in an accurately and in a differentially private manner in the random oracle model, implying 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.

* A preliminary version appeared in [17].

† Supported by the Israeli Centers of Research Excellence (I-CORE) program (Center No. 4/11), Israel Science Foundation (Grant No. 1076/11).

‡ Research was done while Eran Omri was at Bar Ilan University. Supported by the Israel Science Foundation (Grant No. 189/11).

§ Supported by the Israel Science Foundation (Grant No. 189/11). Hila Zarosim is grateful to the Azrieli Foundation for the award of an Azrieli Fellowship.

1. Introduction

In the *random-oracle* model, the parties are given oracle access to a random function (i.e. a uniformly chosen function from the set of all functions of a given input and output length—the *all-function* family) and are assumed to have unbounded computational power (though they can only make a bounded number of oracle queries). Many cryptographic primitives are known to exist in this model, such as (exponentially hard) collision-resistant hash functions. More importantly, in this model, it is possible to implement 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 [10], showing how to convert three-message identification schemes to a highly efficient (non interactive) signature scheme. In the random-oracle model, their methodology preserves the security of the original scheme [25], but (for some schemes) does not do so in the standard model [4,14].

Random oracles, however, are not all powerful. In their seminal work, [18] showed that key-agreement protocols cannot be realized in the random-oracle model. Still, characterizing the functionalities that can be (securely) realized in this model remained an open question.

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-oracle model (e.g. commitment schemes, coin-tossing protocols and zero-knowledge proofs). All of these functionalities, however, are blatantly trivial when considering semi-honest adversaries, which are the focus of this work.

1.1. *Our Result*

We make progress towards answering the above question, showing that, to a large extent, any no-private input,[1] two-party computation that can be securely implemented in the random-oracle model in the presence of semi-honest adversaries *can* be securely implemented in the *information-theoretic* model in the presence of semi-honest adversaries.

Theorem 1.1. (Main theorem, informal). *Let π be a no-private-input, m-round, ℓ-query, oracle-aided two-party protocol and let $\langle X, Y\rangle$ stand for a random execution of the protocol (X, Y), resulting in the parties' private outputs and the common transcript. 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 $\widetilde{\pi} = (\widetilde{\mathsf{A}}, \widetilde{\mathsf{B}})$ such that:*

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

where $\mathcal{F}_{\mathrm{AF}}$ is the all-functions family.[2]

[1] The parties' only input, if any, is a common value drawn from some arbitrary distribution.

[2] We emphasize that the protocol $\widetilde{\pi}$ and the mapping function Map are typically inefficient.

Furthermore, the projections of the above distributions to their first and third coordinates, or to their second and third coordinates (i.e. the transcripts concatenated with the outputs of one of the parties) are identically *distributed.*

Intuitively, Theorem 1.1 states that for any oracle-aided protocol π that is executed with access to a random function, there exists a no-oracle protocol $\widetilde{\pi}$ in the information-theoretic setting that is almost as correct and almost as secure as π. This is formalized by the requirement that the distributions induced by the private outputs of the parties and the common transcript in a random execution of π^f (for $f \leftarrow \mathcal{F}_{\mathrm{AF}}$), being almost the same as that induced by a random execution of the (no-oracle) protocol $\widetilde{\pi}$, where the only difference is that one needs to apply a query-efficient procedure Map to the transcript in the execution of π. Correctness follows directly from the statistical closeness of the outputs, where security is implied by the fact that anything that can be learnt from the transcript of $\widetilde{\pi}$ can also be learnt from the transcript of π by applying the query-efficient function Map on it. Theorem 1.1 generalizes to all-function families that are finite and have the property that answers for *distinct* queries, induced by drawing a random member from the family, are independent.

The main power of Theorem 1.1 is that it facilitates proving the *inexistence* of certain *random-oracle* model primitives, by proving the inexistence of their *no-oracle* analogues; given a random-oracle model protocol π implementing a certain functionality, say key agreement (where the parties wish to secretly agree on a common key), consider its no-oracle variant $\widetilde{\pi}$ guaranteed by Theorem 1.1. Since no information-theoretically secure key-agreement protocol exists, there exists a passive (i.e. semi-honest) adversary $\widetilde{\mathsf{Eve}}$ that "extracts" the common key agreed by the parties of $\widetilde{\pi}$ from the protocol's transcript. Theorem 1.1 yields that by invoking $\widetilde{\mathsf{Eve}}$ on the output of Map applied to the transcript of the random-oracle protocol π, one finds the key agreed by the parties of π with high probability. Since we considered an arbitrary protocol π, the above yields the inexistence of key-agreement protocols in the random-oracle model, reproving [1,18].[3]

A major ingredient of the proof of Theorem 1.1 is the *dependency finder* algorithm presented by [1], refining a similar algorithm by [18] (see Sect. 1.2). While we could have based the proof of Theorem 1.1 on a combination of several results from [1] (or alternatively, to get a theorem with worse parameters, by basing the proof on a followup result of [7, Lemma 5] or of [21, Lemma A.1]), we chose to give a new proof also for this part (modulo clearly marked parts taken from [1]). The new proof (given as part of the proof of Lemma 3.10) holds with respect to a larger set of function families. More significantly, it is more modular and introduces several simplifications compared with the previous proofs.

1.1.1. *Applications*

We demonstrate the usefulness of Theorem 1.1 via the following two examples. The first example shows that in the random-oracle model, it is impossible for two parties

[3] The actual parameters achieved by applying Theorem 1.1 (see Sect. 4.1) match the optimal bound given in [1]. As in the case of [1,18], the above yields that key-agreement protocols cannot be black-box reduced to one-way functions.

to accurately approximate 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 one party to the other party. A recent result of [22] 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 1.1, we obtain the following fact.[4]

Theorem 1.2. (informal). *In the random-oracle model, any ℓ-query (ℓ^2, α, γ)-differentially private oracle-aided protocol for computing the inner product of two n-bits strings, errs by $\frac{\sqrt{n}}{\log(n)\cdot e^{\alpha}}$ with a constant probability.*

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

The above result yields the impossibility of fully black-box reducing differentially private protocols for (well) approximating the 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., [12,18]), 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.[5] Hence, Theorem 1.2 yields the following result.

Corollary 1.3. *(informal). There exists* no *fully black-box reduction from (α, γ)-differentially private protocol computing with error $o(\frac{\sqrt{n}}{\log(n)\cdot e^{\alpha}})$ the inner product of two n-bit strings, to one-way functions.*

We mention that, following an observation made by [22], Theorem 1.2 and Corollary 1.3 imply similar results for two-party differentially private protocols for the Hamming distance functionality.[6]

The second example we give is for secure sampling. Let $G = (G_{\mathsf{A}}, G_{\mathsf{B}})$ be a distribution over $\mathcal{A} \times \mathcal{B}$, where G_{A} and G_{B} denote its marginal distributions over $\mathcal{A}$ and $\mathcal{B}$, respectively. A protocol $\pi = (\mathsf{A}, \mathsf{B})$ is an *information-theoretically* δ-secure implementation of G, if it is a δ-correct (no-oracle) implementation of G (i.e. the local outputs of the parties induced by a random execution of π are δ-close to G) and is δ-private

[4] We mention that the result of [22] is stated for protocol with inputs, where Theorem 1.1 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 [22].

[5] 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.

[6] 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.

according to the simulation paradigm (against all-powerful distinguishers). Specifically, there exists an algorithm $\mathrm{Sim}_{\mathsf{A}}$ (a simulator for A) such that $(x, y, \mathrm{Sim}_{\mathsf{A}}(x))_{(x,y)\leftarrow G}$ is δ-close to the distribution of the parties outputs and A's view in a random execution of π. (Similarly, there exists such a simulator $\mathrm{Sim}_{\mathsf{B}}$ for B). A protocol π is a (T, δ)-secure *random-oracle* implementation of G, if it is a δ-correct random-model implementation of G, and it is δ-private according to the simulation paradigm, against T-query, all-powerful distinguishers. Theorem 1.1 yields the following result.

Theorem 1.4. (informal). *Let π be an ℓ-query oracle-aided protocol that is an $(O(\ell^2/\delta^2), \delta)$-secure implementation of a distribution G in the random-oracle model, then G has an information-theoretically $O(\delta)$-secure implementation.*

We note that Theorem 1.4 does not seem to imply the aforementioned differential privacy result, since the notion of differential privacy cannot be realized via the real/ideal paradigm.

Limitations of Theorem 1.1. By definition, applications of Theorem 1.1 are restricted to no-private input, semi-honest protocols (for obvious reasons though, inexistence of semi-honest security yields inexistence of the full security). In addition, since the distributions described in Theorem 1.1 are only $O(\varepsilon)$ close to each other, the theorem seems to only be useful for showing the impossibility of *robust* primitives: ones that remain information-theoretically unachievable, even if one slightly changes the primitive correctness or security requirement (e.g. the parties agree on the same key with slightly smaller probability). We are unaware, however, of any natural primitive for which the above robustness condition does not hold.

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 is 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 [18], and [1] (algorithm *Eve*, in their terminology, and *Independence Learner* in the terminology of [7,21]). 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 $\widetilde{\pi} = (\widetilde{\mathsf{A}}, \widetilde{\mathsf{B}})$: 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 $\widetilde{\pi} = (\widetilde{\mathsf{A}}, \widetilde{\mathsf{B}})$ is defined as follows:

assume that $\widetilde{\mathsf{A}}$ speaks in 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, no-oracle $\widetilde{\mathsf{A}}$ samples random values for $f \in \mathcal{F}_{\mathrm{AF}}$ and the random coins of A conditioned on $\bar{t}_{1,\ldots,i}$ being the protocol's transcript, and f being consistent with $\mathcal{I}_i$. It then lets t_{i+1} be the next message of A induced by the above choice of f and random coins, and sends $\left(\bar{t}' = (\bar{t}_{1,\ldots,i}, t_{i+1}), \mathsf{Finder}^f(\bar{t}')\right)$ back to $\widetilde{\mathsf{B}}$. In case this is the last round of interaction, $\widetilde{\mathsf{A}}$ locally outputs the (local) output of A induced by this choice of f and random coins. In other words, $\widetilde{\mathsf{A}}$ selects a random view (including the oracle itself) for A that is consistent with the information contained in the no-oracle protocol augmented transcript (i.e. the transcript of the oracle protocol and the oracle calls) and then acts as A would.

The fact that $\widetilde{\mathsf{A}}$ perfectly emulates A (and that $\widetilde{\mathsf{B}}$ 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 $\widetilde{\pi}$ does (actually, the above facts hold for any reasonable definition of Finder,[7] and for any function family). The interesting part is arguing that the *joint output* of the no-oracle 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 no-oracle parties compute for the query q might be different. In this case, the joint output of the no-oracle protocol does not look like the joint output of the oracle protocol. Luckily, the above scenario is unlikely to happen due to the guarantee of Finder; with high probability, $\mathcal{I}$ contains *all* common queries that the two parties made, yielding that the joint output of the no-oracle protocol has similar distribution to that of the oracle protocol. It turns out that the above example generalizes to any possible protocol, yielding that the above mapping and no-oracle protocol are indeed the desired ones.

1.3. *Related Work*

In their seminal work, [18] showed that there are no key-agreement protocols in the random-oracle model and deduced that key-agreement protocols cannot be black-box reduced to one-way functions. This result was later improved by [1], showing there are no ℓ-query key-agreement protocols in the random-oracle model, secure against adversaries making $O(\ell^2)$ queries, thus matching the upper bound of [23]. [21] show that the all-function family (and thus one-way functions) are useless for *secure function evaluation* of deterministic, polynomial input-domain, two-party functionalities. In other words, deterministic, bounded input-domain functionalities that *can* be securely computed in the random-oracle model are the *trivial* ones—functionalities that can be securely computed unconditionally. The comparison to the result stated here is that [21] handle *with polynomial input-domain* functionalities, but only *deterministic* ones, where here we handle *input-less* functionalities, but including *randomized ones*. Putting the above re-

[7] It is only required Finder's output contains *all* queries it made to the oracle.

sults together gives a partial characterization of the power of the random-oracle model for two-party computation secure in the presence of semi-honest adversaries. It is still open, however, whether the random-oracle model is useful for securely computing randomized functionalities *with* inputs or functionalities of super-polynomial input domain. [7] give a random-oracle to no-oracle equivalence in a similar flavour to that of Theorem 1.1. The result of [7] holds for general *with-input* protocols, but is *restricted* to $o(n/\log n)$-round protocols, where n being the random function input length. In addition, the (implicit) mapping procedure given in [7] might make *sub-exponential* number of oracle calls (even for a constant ε). Finally, a long line of research, starting with the work of [6], gives a partial characterization of those functionalities that can be securely computed information theoretically.

1.3.1. *Additional Black-Box Separations*

Following [18], the method of black-box separation was subsequently used in many other works: [27] shows that there exists no black-box reduction from a k-round secret key agreements to $(k-1)$-round secret key agreements; [29] shows that there exists no black-box reductions from collision-free hash functions to one-way permutations; [19] shows that there exists no construction of one-way permutations based on one-way functions. Other works using this paradigm contain [5,11–13,16,20,30], to name a few.

1.3.2. *Differential Privacy*

Distributed differential privacy was considered by [3], who studied the setting of multiparty differentially private computation (where an n-bit database is shared between n parties). They gave a separation between information-theoretic and computational differential privacy in the distributed setting. The notion of computational differential privacy was considered in [24]. 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 a $2n$-bit database is shared between two parties, each holding half of the bits) was considered by [22]. 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 [22] to the random-oracle model. For the client–server setting (where the server holds the entire database and the client may issue queries to it), [15] considered the limitations of computational differentially private computation in the centralized setting (where a single entity holds the entire database). For this setting, they gave a black-box separation of computational differentially private computation from a large range of cryptographic primitives such as trapdoor permutations, collision-resistant hash functions and random oracles.

1.4. *Open Problems*

As mentioned above, the main open problem is a full characterization of the power of the random-oracle model with respect to semi-honest adversaries. Specifically, is it possible to come up with a random-oracle to no-oracle mapping that works for *any* (also with inputs) oracle-aided protocol?[8] Note that it is still open whether the mapping described in Theorem 1.1 can be used to rule out *any* non-trivial, input-less semi-honest random-oracle protocol, or only those that are "far" enough from being trivial (see discussion at the end of Sect. 1.1.1).

Finally, an interesting question is to come up with a random-oracle to no-oracle mapping that is applicable to protocols secure against *fail-stop adversaries*. We failed to use the mapping described in Theorem 1.1 for these settings. Loosely speaking, the reason that our approach fails is that the party that is active at the i'th round might first compute the i'th message and only then decide whether to abort or not based on this message. Now, if this is the case, then the probability that a party aborts in the oracle-aided protocol might be correlated with the view of the other party because of the queries that Finder makes while computing the i'th message; hence, the joint distribution of the oracle-aided protocol is no longer close to product distribution, unlike the plain model protocol. A potential implication of such a result for fail-stop adversaries is that optimally fair coin tossing is impossible to achieve in the random function model.[9]

Paper Organization

Formal definitions and notation used throughout the paper are given in Sect. 2. Our main result is stated and proved in Sect. 3, and several applications of this result are given in Sect. 4. Our new proof for the main technical lemma of [1] is given in "Appendix".

2. Preliminaries

2.1. *Notations*

We use calligraphic letters to denote sets, uppercase for random variables and lowercase for values. Let poly be the set of all polynomials, let PPT stand for probabilistic polynomial time, and PPTM stands for PPT algorithm (machine). 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 $[m] = \{1, \ldots, m\}$.

[8] For *with input* protocols, the approach described in Sect. 1.2 miserably fails. The reason is that a no-oracle party might choose a function (oracle) f that is *inconsistent* with the other party (already chosen) input, yielding a wrong emulation of the oracle-aided protocol. This issue does not arise in the case of no-input protocols, since the distribution induced by the random choice of f done by the no-oracle party can be shown to yield the *right* distribution for the parties (yet to be chosen) outputs.

[9] The work of [7] mentioned in Sect. 1.3 shows such an impossibility result for $o(n/\log n)$-round protocols, where n being the random function input length. A recent result of [8] shows the impossibility of optimally fair coin tossing in the random function model for protocols that are "function-oblivious" which means that the output of the protocol does not depend on the specific instantiation of the random oracle, but only on the random coins of the parties.

For a finite set $\mathcal{S}$, let $\chi_{\mathcal{S}}$ stand for its characteristic function, and let $x \leftarrow \mathcal{S}$ to denote that x is selected according to the uniform distribution over $\mathcal{S}$. Similarly, for a random variable X, let $x \leftarrow X$ to denote that x is chosen according to X. The support of the distribution D, denoted $\mathrm{Supp}(D)$, is defined as $\{u \in \mathcal{U} : \Pr_D[u] > 0\}$. The statistical distance between two distributions P and Q over a finite set $\mathcal{U}$, denoted $\mathrm{SD}(P, Q)$, is defined as $\frac{1}{2}\sum_{u\in\mathcal{U}} \left|\Pr_P[u] - \Pr_Q[u]\right|$, and is known to be equal to $\max_{\mathcal{S}\subseteq\mathcal{U}} \left(\Pr_P[\mathcal{S}] - \Pr_Q[\mathcal{S}]\right)$. 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).

The communication transcript (henceforth, the transcript) of a given execution of the protocol $\pi = (\mathsf{A}, \mathsf{B})$ is the list of messages $\bar{t}$ 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 random coins, and $\bar{t}$ is the transcript of the execution. Let $(v_{\mathsf{A}})_j$ denote the partial view of A in the first j rounds of the execution described by v_{A}, namely $(v_{\mathsf{A}})_j = (i_{\mathsf{A}}, r_{\mathsf{A}}, \bar{t}_{1,\ldots,j})$; the view v_{B} of B is defined analogously. Let $v = (v_{\mathsf{A}}, v_{\mathsf{B}})$ 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, 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.

A protocol π has m rounds, 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 $\mathrm{out}_j^{\mathsf{P}}(v) = \mathrm{out}_j^{\mathsf{P}}(v_{\mathsf{P}})$ denote the output of party P at the end of the j'th round of v (i.e. the string written on P's output tape), where $\mathrm{out}_j^{\mathsf{P}}(v) = \mathrm{out}_{j-1}^{\mathsf{P}}(v)$, in case P is inactive in the j'th round of v.

In a *stateless* protocol, 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 only input of the parties is their common input (the only exception to that is in Sect. 4.2, 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. Without loss of generality, all oracle function families considered map binary strings to binary strings.

For an oracle-aided, no-input two-party protocol $\pi = (\mathsf{A}, \mathsf{B})$ and a function family $\mathcal{F}$, 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, if its value is clear from the context, we sometimes omit the superscript pair $(\mathcal{F}, \pi)$). For $f \in \mathcal{F}$, the distribution $\langle \pi^f = (\mathsf{A}^f, \mathsf{B}^f) \rangle$ 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, let $(v_{\mathsf{A}})_j$ denote the partial view of A in the first j rounds of the execution described by v_{A}, namely $(v_{\mathsf{A}})_j = (r_{\mathsf{A}}, \bar{t}_{1,\ldots,j}, (f_{\mathsf{A}})_j)$. The view v_{B} is analogously defined.

For $\omega \in \Omega$, let $\mathsf{view}(\omega)$ 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 the party acting in the last round outputs a final message. 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 is denoted by $|\bar{t}|$. Consider the following distributions.

Definition 2.1. ($\Omega(\bar{t}, \mathcal{I})$ and $\mathcal{VIEW}(\bar{t}, \mathcal{I})$). Let $\mathcal{F}$ be a function family and let π be an oracle-aided protocol. 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 $\langle \mathsf{A}^f(r_{\mathsf{A}}), \mathsf{B}^f(r_{\mathsf{B}}) \rangle$. Given a set $\mathcal{P} \subseteq \Omega$, let $\Omega_{\mathcal{P}}(\bar{t}, \mathcal{I}) = \Omega(\bar{t}, \mathcal{I}) \cap \mathcal{P}$.

Define the random variable $\mathcal{VIEW}^{\mathcal{F},\pi}(\bar{t}, \mathcal{I})$ as the value of $\mathsf{view}(\omega)_{|\bar{t}|}$ for $\omega \leftarrow \Omega(\bar{t}, \mathcal{I})$, and define $\mathcal{VIEW}^{\mathcal{F},\pi}_{\mathcal{P}}(\bar{t}, \mathcal{I})$ analogously.

Since the above definition considers the uniform distribution over Ω, 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 state and prove our main result, a mapping from protocols in the random-oracle model to (inefficient) no-oracle protocols.

3.1. *Dependent Views*

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 joint information of the parties about f. We start by formally defining what it means for $\mathcal{I}$ to contain all joint information.

Definition 3.1. Let v_{A} be a j_{A}-round view for A and v_{B} be a j_{B}-round view for B. 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, let

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 3.2. (*Dependent views*) Let $v = (v_{\mathsf{A}}, v_{\mathsf{B}})$ be a pair of (possibly partial) valid views. The views v_{A} and v_{B} are dependent with respect to a set of query/answer pairs $\mathcal{I}$ and a function family $\mathcal{F}$, if $\alpha^{\mathcal{I}}_{v_{\mathsf{A}}} \neq \alpha^{\mathcal{I}}_{v_{\mathsf{A}}|v_{\mathsf{B}}}$ or $\alpha^{\mathcal{I}}_{v_{\mathsf{B}}} \neq \alpha^{\mathcal{I}}_{v_{\mathsf{B}}|v_{\mathsf{A}}}$. Otherwise, v_{A} and v_{B} are independent with respect to $\mathcal{I}$ and $\mathcal{F}$.

The following observation (generalizing a similar observation made in [1]) plays a crucial role in the proof of our main result (stated in Sect. 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 3.3. *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 $\Pr_{\Omega}[v, \mathcal{I}, \mathcal{P}] = \Pr_{\Omega}[v, \mathcal{I}]$, it holds that*

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

Proof. Note that

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

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

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

Note that $\Pr_{\Omega|r_\mathsf{A},r_\mathsf{B},\mathcal{I}}[v] = \Pr_{\Omega|r_\mathsf{A},r_\mathsf{B},\mathcal{I}}[\mathcal{I}_\mathsf{A}, \mathcal{I}_\mathsf{B}, \bar{t}]$, 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). The reason for this is that 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. It follows that

$$\Pr_{\Omega_{\mathrm{P}}(\bar{t},\mathcal{I})}[v] = \frac{\Pr_\Omega[r_\mathsf{A}, r_\mathsf{B}] \cdot \Pr_{\Omega|r_\mathsf{A},r_\mathsf{B},\mathcal{I}}[\mathcal{I}_\mathsf{A}, \mathcal{I}_\mathsf{B}, \bar{t}]}{\Pr_{\Omega \mid \mathcal{I}}[\bar{t}, \mathrm{P}]} \tag{2}$$

We next analyse the term $\Pr_{\Omega|r_\mathsf{A},r_\mathsf{B},\mathcal{I}}[\mathcal{I}_\mathsf{A}, \mathcal{I}_\mathsf{B}, \bar{t}]$. 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

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

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 $\Pr_{\Omega|\mathrm{inf}'}\left[{\mathcal{I}_\mathsf{A}}^i, {\mathcal{I}_\mathsf{B}}^i\right] = \Pr_{\omega\leftarrow\Omega|\mathrm{inf}'}[\omega \text{ is consistent with } {\mathcal{I}_\mathsf{A}}^i, {\mathcal{I}_\mathsf{B}}^i]$ 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 3.4. (*Intersecting views*) A (possibly partial) pair of views $v = (v_{\mathsf{A}}, v_{\mathsf{B}})$ is 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 a typical function family, a view with an intersection is 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 3.5. (*Simple function families*) A function family $\mathcal{F}$ is simple, if it is finite, and $\text{Dependent}^{\mathcal{F}}_{\mathcal{I}}(v) \implies \text{Intersect}_{\mathcal{I}}(v)$ 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 consistent with $\mathcal{I}$.

It is immediate that the all-function family i.e. the set of "random functions" (see formal definition in Sect. 4.4) 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 Sect. 4, we use this result to prove limitations on the power of oracle-aided protocols in achieving specific cryptographic tasks.

Definition 3.6. (*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 $(\widetilde{\mathsf{A}}, \widetilde{\mathsf{B}})$, such that the following holds.[10]

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

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

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

Furthermore, $\mathcal{D}_P[1, 3] \equiv \mathcal{D}_{\mathcal{F}}[1, 3]$ and $\mathcal{D}_P[2, 3] \equiv \mathcal{D}_{\mathcal{F}}[2, 3]$.[11]

[10] Recall that $\langle X, Y \rangle$ stands for a random execution of the protocol (X, Y) that $\mathsf{trans}(v)$ denotes the transcript part in v and that $\text{out}^X_j(v)$ denotes the output of party X in the j'th round of v.

[11] Ie 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.

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 set of oracle calls made in $\mathsf{Map}^f(\bar{t}_{1,\ldots,j})$ is a subset of those made in $\mathsf{Map}^f(\bar{t})$.

Theorem 3.7. *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 a $(2^{10} \cdot \ell^2/\varepsilon^2, \varepsilon)$-*mapping *for any $0 < \varepsilon \leq 1$.*

Remark 3.8. (Round complexity of the no-oracle protocol) The proof of Theorem 3.7 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 $\widetilde{\mathsf{A}}$ and $\widetilde{\mathsf{B}}$ 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 proof of Theorem 3.7 immediately follows by the next two lemmata.

Definition 3.9. ($\mathsf{DependencyFinder}$) 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)-$\mathsf{DependencyFinder}$ for $(\mathcal{F}, \pi)$, if the following holds for any $j \in [m]$.

Let $\mathsf{CF} = \mathsf{CF}(\mathcal{F}, \pi, \mathsf{Finder})$ be the following random process:

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. $\Pr[\#\text{ of } f - \text{calls made in } \mathsf{CF} > T] \leq \varepsilon$.

That is, conditioned on a random transcript of $\pi^{\mathcal{F}}$ and the oracle queries made by a (T, δ)-$\mathsf{DependencyFinder}$, the views of the parties are close to being in a product distribution.

Lemma 3.10. *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 $(64/\delta^2, \ell\delta)$-*$\mathsf{DependencyFinder}$ *for any $0 < \delta \leq \frac{1}{4\ell}$.*

Lemma 3.11. *Any pair of function family and protocol that has a (T, ε)-*$\mathsf{DependencyFinder}$ *has a $(T, 2\varepsilon)$-*mapping.

The proof of Lemma 3.11 is given in Sect. 3.3.2, where the proof of Lemma 3.10 is given in "Appendix".[12]

[12] As mentioned in the introduction, the proof of Lemma 3.10 could have been derived by combining several statements appearing in [1]. A somewhat weaker variant of the lemma can be directly proved using the followup result of [7, Lemma 5] or of [21, Lemma A.1].

3.3.1. *Proving Theorem 3.7*

Proof of Theorem 3.7. Immediately follows from Lemmas 3.10 and 3.11, taking $\delta = \varepsilon/4\ell$. □

3.3.2. *Proving Lemma 3.11*

Proof. Let $(\mathcal{F}, \pi)$ be a pair of a function family and an m-round oracle-aided protocol that have a (T, ε)-DependencyFinder algorithm Finder. We start by defining the mapping algorithm and then define the no-oracle protocol. □

Algorithm 3.12. (Map).

Oracle: $f \in \mathcal{F}$.
Input: *A* 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* $\mathcal{I}_0 = \emptyset$*).*
   If in some round i *the* overall *number of* f *calls (made by* Finder*) is* T*, halt the above loop, and for all* $i \le i' \le j$ *set* $\mathcal{I}_{i'}$ *to be the set of* T *query/answer pairs obtained so far.*
2. *Output* $\left(\bar{t}_1, \mathcal{I}_1\right), \left(\bar{t}_{1,2}, \mathcal{I}_2\right), \ldots, \left(\bar{t}, \mathcal{I}_j\right)$.

The no-oracle protocol. Our stateless, no-oracle protocol $\widetilde{\pi} = (\widetilde{\mathsf{A}}, \widetilde{\mathsf{B}})$ 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, $\widetilde{\mathsf{A}}$ is active in $\widetilde{\pi}$ in the same rounds that A is in π (same for $\widetilde{\mathsf{B}}$ and B). The definition of $\widetilde{\mathsf{A}}$ is given below ($\widetilde{\mathsf{B}}$ is analogously defined).

Algorithm 3.13. *(*$\widetilde{\mathsf{A}}$*).*

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* $\widetilde{\mathsf{B}}$.

We prove that algorithm Map (Algorithm 3.12) and protocol $\widetilde{\pi} = (\widetilde{\mathsf{A}}, \widetilde{\mathsf{B}})$ (Algorithm 3.13) form a $(T, 2\varepsilon)$-mapping for $(\mathcal{F}, \pi)$. By construction, algorithm Map is deterministic (since Finder is deterministic), makes at most T queries and fulfils the second item of Definition 3.6. Towards showing that $(\mathsf{Map}, \widetilde{\pi})$ fulfils also the first property of Definition 3.6 with respect to the stated parameter, we prove the following claim:

Claim 3.14. *for every* $j \in [m]$: $\left(\mathsf{Map}^f(\mathsf{trans}(v)_{1,\ldots,j})\right)_{f\leftarrow\mathcal{F},v\leftarrow\langle \mathsf{A}^f,\mathsf{B}^f\rangle} \equiv (\mathsf{trans}(\tilde{v})_{1,\ldots,j})_{\tilde{v}\leftarrow\langle\widetilde{\mathsf{A}},\widetilde{\mathsf{B}}\rangle}$.

Proof. The claim trivially holds for $j = 0$, where the proof for a larger value of j is done by induction. 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 fulfils the second item of Definition 3.6), 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 3.14 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 $\mathcal{D}_{\mathcal{F}}[1]$ (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 $\widetilde{\pi} = (\widetilde{\mathsf{A}},\widetilde{\mathsf{B}})$ yields that

$$\begin{aligned}\mathcal{D}_P &:= \left(\mathrm{out}_j^{\widetilde{\mathsf{A}}}(\tilde{v}),\mathrm{out}_j^{\widetilde{\mathsf{B}}}(\tilde{v}),\mathsf{trans}(\tilde{v})_{1,\ldots,j}\right)_{\tilde{v}\leftarrow\langle\widetilde{\mathsf{A}},\widetilde{\mathsf{B}}\rangle}\\ &\equiv \left(\mathrm{out}_j^{\mathsf{A}}(v_{\mathsf{A}}),\mathrm{out}_j^{\mathsf{B}}(v_{\mathsf{B}}),\tilde{t}_{1,\ldots,j}\right)_{\tilde{t}\leftarrow\mathsf{trans}(\langle\widetilde{\mathsf{A}},\widetilde{\mathsf{B}}\rangle),v_{\mathsf{A}}\leftarrow\mathcal{VIEW}(\tilde{t}_j)_{\mathsf{A}},v_{\mathsf{B}}\leftarrow\mathcal{VIEW}(\tilde{t}_j)_{\mathsf{A}}}\end{aligned} \tag{3}$$

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

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

and therefore, Claim 3.14 yields that

$$\mathcal{D}_{\mathcal{F}} \equiv \left(\mathrm{out}_j^{\mathsf{A}}(v_{\mathsf{A}}),\mathrm{out}_j^{\mathsf{B}}(v_{\mathsf{B}}),\tilde{t}_{1,\ldots,j}\right)_{\tilde{t}\leftarrow\mathsf{trans}(\langle\widetilde{\mathsf{A}},\widetilde{\mathsf{B}}\rangle),v\leftarrow\mathcal{VIEW}(\tilde{t}_j)} \tag{4}$$

We conclude the proof using the fact that Finder is a (T,ε)-$\mathsf{DependencyFinder}$ for $(\mathcal{F},\pi)$. The issue to note here is that Process CF (described in Definition 3.9) 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 $\left|\mathcal{I}_j\right| \le T$. 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 3.14 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, ε)-DependencyFinder for $(\mathcal{F}, \pi)$ yields a bound of ε on the contribution of elements whose transcripts are *inside* $\mathcal{S}$ to the statistical distance between $\mathcal{D}_{\mathcal{F}}$ and $\mathcal{D}_P$. It also bounds by ε the probability that CF outputs elements whose transcripts are 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 $\mathrm{SD}(\mathcal{D}_{\mathcal{F}}, \mathcal{D}_P) \leq \varepsilon + \varepsilon = 2\varepsilon$.

4. Applications

In this section, we use the oracle-aided to no-oracle protocol mapping from Sect. 3, to derive the impossibility of realizing three cryptographic tasks relative to simple function families. In Sect. 4.1, we re-establish the result of [18], showing that key-agreement protocols cannot be realized relative to simple function families. In Sect. 4.2, we extend the lower bound of [22] on the accuracy of two-party differentially private no-oracle protocols, to show it also holds for relative to simple function families. In Sect. 4.3, we show that a distribution that cannot be securely sampled in the information-theoretic model, cannot be securely sampled relative to simple function families. Finally, in Sect. 4.4, we use that the all-function family is simple, to prove the impossibility of reducing the above first two primitives to the hardness of one-way functions in a black-box manner.

We emphasize that all adversaries considered in Sects. 4.1 to 4.3 (and most of those considered in Sect. 4.4) are computationally *unbounded*, but typically can only make bounded number of oracle queries.

Throughout this section, we sometimes only define the security and correctness of the primitives in consideration for oracle-aided implementations. Their no-oracle counterparts are derived by considering these definitions with respect to the trivial function family (i.e. the singleton family, whose only member returns $\perp$ on every query).

4.1. *Key-Agreement Protocols*

In a key-agreement protocol, two parties wish to agree on a common secret in a secure manner—an adversary (observer) seeing the communication transcript cannot find the secret. Below, we prove that non-trivial key-agreement cannot be achieved relative to simple function families. We start by formally defining the notion of key agreement and then recall the known fact that in the information-theoretic 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 Sect. 3, yields a similar result for oracle-aided protocols relative to simple function families.

We remark that the results presented in this section yield very little conceptual added value to what was already shown by [1,18]. We do, however, present them here to demonstrate how they are easily derived from our main result (Theorem 3.7) and as a warm-up before moving on to the other applications of our main result, described in Sects. 4.2 and 4.3.

4.1.1. *Standard Definitions and Known Facts*

Let π be a two-party protocol and let v be the parties' joint view in an interaction of π (i.e. $v \in \text{Supp}\left(\langle \pi^f \rangle\right)$). Recall (see Sect. 2.2) that $\mathsf{trans}(v)$ denotes the communication transcript in v, and $\text{out}_i^{\mathsf{P}}(v)$ denotes the output of the party P at the i'th round. In the following, let $\text{out}^{\mathsf{P}}(v) = \text{out}_m^{\mathsf{P}}(v)$, where m is the last round in v.

Definition 4.1. (*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 relative to a function family $\mathcal{F}$, if the following hold:

Consistency: π is $(1-\alpha)$-*consistent* relative to $\mathcal{F}$. Namely, for every $f \in \mathcal{F}$,

$$\Pr_{v \leftarrow \langle \pi^f \rangle}\left[\text{out}^{\mathsf{A}}(v) = \text{out}^{\mathsf{B}}(v)\right] \geq 1 - \alpha \tag{5}$$

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

$$\Pr_{f \leftarrow \mathcal{F}, v \leftarrow \langle \pi^f \rangle}\left[\mathsf{Eve}^f(\mathsf{trans}(v)) = \text{out}^P(v)\right] \leq \gamma \tag{6}$$

A protocol π is an (α, γ)-key-agreement protocol, if it is a $(\cdot, \alpha, \gamma)$-key-agreement protocol relative to the trivial function family.[13]

In the information-theoretic 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 4.2. *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 a adversary that given the transcript of a random execution of π, outputs the same value as B does with probability $1-\alpha$.*

An immediate implication of Fact 4.2 is that there does not exist a no-oracle, two-party, (α, γ)-key-agreement protocol for any $0 \leq \gamma < 1 - \alpha$. We next use our main result from Sect. 3 to prove a similar result for oracle-aided protocols.

4.1.2. *Limits on Oracle-Aided Key-Agreement Protocols*

In the language of the above definition, our main result is stated as follows.

Theorem 4.3. *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*, and then, π is* not *a (T, α, γ)-key-agreement relative to $\mathcal{F}$ for any $0 \leq \gamma < 1 - (\alpha + \varepsilon)$.*

[13] We remark that our impossibility result (as well the results of [1,18]) would also hold with respect to a weaker definition, requiring consistency to hold for a random f, rather than for every $f \in \mathcal{F}$.

Proof. Assume to the contrary that π is a (T, α, γ)-key-agreement relative to $\mathcal{F}$ for some $0 \leq \gamma < 1 - (\alpha + \varepsilon)$. Let $\widetilde{\pi} = (\widetilde{\mathsf{A}}, \widetilde{\mathsf{B}})$ and Map be $\widetilde{\pi} = (\widetilde{\mathsf{A}}, \widetilde{\mathsf{B}})$ and Map be a (T, δ)-mapping for $(\mathcal{F}, \pi)$. It follows (see Definition 3.6:1) that

$$\mathrm{SD}\left(\left(\mathrm{out}^{\widetilde{\mathsf{A}}}(v), \mathrm{out}^{\widetilde{\mathsf{B}}}(v)\right)_{v \leftarrow \langle \widetilde{\pi} \rangle}, \left(\mathrm{out}^{\mathsf{A}}(v), \mathrm{out}^{\mathsf{B}}(v)\right)_{f \leftarrow \mathcal{F}, v \leftarrow \langle \pi^f \rangle}\right) \leq \varepsilon \tag{7}$$

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

$$\tau := \Pr_{v \leftarrow \langle \widetilde{\pi} \rangle}\left[\mathrm{out}^{\widetilde{\mathsf{A}}}(v) = \mathrm{out}^{\widetilde{\mathsf{B}}}(v)\right] \geq 1 - (\alpha + \varepsilon) \tag{8}$$

Fact 4.2 yields an adversary $\widetilde{\mathsf{Eve}}$ that given the transcript of a random execution of $\widetilde{\pi}$, 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 Eve makes at most T oracle calls. It follows that

$$\begin{aligned}
\Pr_{f \leftarrow \mathcal{F}, v \leftarrow \langle \pi^f \rangle}\left[\mathsf{Eve}^f(\mathsf{trans}(v)) = \mathrm{out}^{\mathsf{B}}(v)\right] &= \Pr_{f \leftarrow \mathcal{F}, v \leftarrow \langle \pi^f \rangle}\left[\widetilde{\mathsf{Eve}}\left(\mathsf{Map}^f(\mathsf{trans}(v))\right) = \mathrm{out}^{\mathsf{B}}(v)\right] \\
&= \Pr_{\tilde{v} \leftarrow \langle \widetilde{\pi} \rangle}\left[\widetilde{\mathsf{Eve}}(\mathsf{trans}(\tilde{v})) = \mathrm{out}^{\widetilde{\mathsf{B}}}(\tilde{v})\right] \\
&= \tau \geq 1 - (\alpha + \varepsilon),
\end{aligned} \tag{9}$$

where the second equality follows since $(\mathsf{Map}^f(\mathsf{trans}(v)), \mathrm{out}^{\mathsf{B}}(v))$ is identically distributed as $(\mathsf{trans}(\tilde{v}), \mathrm{out}^{\widetilde{\mathsf{B}}}(\tilde{v}))$, where f, v and $\tilde{v}$ are sampled as in Eq. (9) (follows from the second property of the protocol/mapping pair, see Definition 3.6). □

Combining Theorems 3.7 and 4.3 yields the following result.

Theorem 4.4. *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 relative 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 3.7 yields that $(\mathcal{F}, \pi)$ has a (T, ε)-mapping for $T = 2^{10} \cdot \left(\frac{\ell}{\varepsilon}\right)^2 = 2^{10} \cdot \left(\frac{\ell}{1-\alpha-\gamma}\right)^2$. Since $0 \leq \gamma < 1 - (\alpha + \varepsilon)$ and $k \geq T$, Theorem 4.3 yields that π is not a (k, α, γ)-key-agreement protocol relative to $\mathcal{F}$. □

4.2. *Differentially Private Two-Party Computation*

In this section, we apply our main result to extend the lower bound of [22] 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 $\Omega\left(\sqrt{s}/\log s\right)$ (see Sect. 4.2.2 for the formal statement). This fact is later used in Sect. 4.4.3 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 Sect. 4.1, here we consider a setting where the parties *do* hold private inputs. Since our main result (Theorem 3.7) 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.

In proving the results of this section, we begin (Sect. 4.2.3) by using Theorem 3.7 together with an ("information theoretic") result by [22], 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 (Sect. 4.2.4) derive the same limitation on protocols with inputs, but where the correctness and privacy are measured with respect to *uniformly chosen* inputs. Finally, Sect. 4.2.5, we use the latter result to show the same limitation for *fixed* inputs, hence proving our main result. Before starting with the aforementioned plan, we first recall the formal definition of differential privacy, cite the result of [22] (Sect. 4.2.1) and formally state our main results (Sect. 4.2.2).

4.2.1. *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 $H_d\left(x, x'\right) = \left|\left\{i \in [s] \colon x_i \neq x'_i\right\}\right|$ 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 4.5. ((α, γ)-differential privacy [9] (in the centralized model)). A randomized mechanism M over Σ^s is (α, γ)-differentially private, if for every distinguisher D and every $x, x' \in \Sigma^s$ with $H_d\left(x, x'\right) = 1$, it holds that

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

If M satisfies (α, γ)-differential privacy with $\gamma = 0$, then M is just α-differentially private.[14]

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

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

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 [3,22,24]. Indeed, our definitions are close in spirit to the definition of IND-CDP from [24] (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 Sect. 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 $v \in \mathrm{Supp}\left(\langle \pi^f(x,y) \rangle\right)$, 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.

Definition 4.6. (*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 relative to $\mathcal{F}$ and B is analogously defined. If π is (k, α, γ)-differentially private relative to $\mathcal{F}$ and both parties, then it is (k, α, γ)-differentially private relative to $\mathcal{F}$.

Finally, π is (α, γ)-differentially private, if it is $(\cdot, \alpha, \gamma)$-differentially private relative 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 the 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 4.7. (*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 relative to a function family $\mathcal{F}$, if for every $f \in \mathcal{F}$, for every $x, y \in \{0,1\}^s$ and $\mathsf{P} \in \{\mathsf{A}, \mathsf{B}\}$, it holds that

$$\Pr_{v \leftarrow \langle \pi^f(x,y) \rangle} \left[\left| \text{out}^{\mathsf{P}}(v) - g(x,y) \right| > d \right] < \beta. \tag{10}$$

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 $\mathsf{IP}(x, y) = \sum_{i \in [s]} x_i \cdot y_i$. [22] Showed that for a 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 4.8. ([22, 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*

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

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

In the next section, we use similar arguments to the ones used by [22], to prove a variant of Theorem 4.8 for (no-oracle) no-input protocols (which we call here *sampled-input* protocols). For that we recall a few definitions and results from [22].

Lemma 4.9. ([22, 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*

$$\Pr_{m \leftarrow \mathsf{M}(x)} \left[\frac{\Pr[\mathsf{M}(x) = m]}{\Pr[\mathsf{M}(x') = m]} \notin \left[e^{-(\nu+\alpha)}, e^{(\nu+\alpha)} \right] \right] < \gamma \cdot \frac{1 + e^{-(\nu+\alpha)}}{1 - e^{-\nu}}. \tag{11}$$

Unpredictability of Bit Sources. The model of random sources introduced by [28] 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 4.10. ((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 the η is, the more the "randomness" is the source guaranteed to have. Specifically, an unpredictable source has high min-entropy.

Fact 4.11. *Let $X = (X_1, \ldots, X_s)$ be an η-unpredictable source, then the* min-entropy of X, *defined as $H_\infty(X) = \min_{x \in \mathrm{Supp}(X)} \log \frac{1}{\Pr[X=x]}$ 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 4.10 yields that[15]

$$\Pr\left[X_i = b \mid X_1 = x_1, \ldots X_{i-1} = x_{i-1}\right] \geq \eta \cdot \Pr\left[X_i = 1-b \mid X_1 = x_1, \ldots X_{i-1} = x_{i-1}\right].$$

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 [22].

Lemma 4.12. ([22, 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 $\hat{\eta} = \eta \cdot \frac{1-\nu}{1+\nu}$.*

Corollary 4.13. *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 4.12 with $\nu = 1/2$. □

Theorem 4.14. ([22, Theorem3.4]). *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 $Z = \mathsf{IP}(X, Y)$* mod

[15] For $b = 1$, this is implied by the right-hand-side inequality in the condition of Definition 4.10.

r for some $r \in \mathbb{N}$. Assume that $s \geq c \cdot \frac{r^2}{\eta\beta} \cdot \log\left(\frac{r}{\beta}\right) \cdot \log\left(\frac{r}{\gamma}\right)$ for some $\gamma \in [0,1]$, where c is a universal constant, then $\mathrm{SD}((Y,Z),(Y,U_r)) \leq \gamma$, where U_r is uniform on $\mathbb{Z}_r$ and independent of Y.

4.2.2. *Limits on Differentially Private Oracle-Aided Protocols for Computing Inner Product*

In this section, we state our main impossibility results for differentially private, oracle-aided protocols for accurately approximating the inner product of two s-bit strings. We first give (Theorem 4.16) a lower bound on the accuracy of differentially private protocols for approximating the inner product of two s-bit strings relative to general function families (for protocols that have a certain type of mapping to no-oracle protocols). We then give lower bound on the accuracy of *any* differentially private, oracle-aided protocol for approximating the inner product of two s-bit strings relative to simple function families (Theorem 4.17).

Since these results deal with with-input protocols and since our discussion in Sect. 3 only handles no-input protocols, our proof proceeds by reducing the problem of with-input protocols that accurately approximate the inner-product function to a similar problem on no-input protocols. Specifically, for a given with-input protocol, we consider its no-input variant (called the *sampled-input variant*), in which the parties use the first s bits in their random input string as inputs (see the formal definition below).

Definition 4.15. (*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.

Roughly speaking, in Theorem 4.16, stated below, we show that if an oracle-aided protocol π is (T,α,γ)-differentially private relative to a function family $\mathcal{F}$ and if $(\mu(\pi),\mathcal{F})$ has a (T,ε)-mapping, then π is not a good approximation for the inner-product functionality relative to $\mathcal{F}$.

Theorem 4.16. *For $v > 0$ and $\alpha \geq 0$, there exist $\lambda > 0$ and $z \in \mathbb{N}$ such that the following holds. Let $\mathcal{F}$ be a function family, let $s \geq z$, let $\pi = (\mathsf{A},\mathsf{B})$ be an oracle-aided, s-bit input protocol, and let $\mu(\pi)$ be its sampled-input variant. Assume that π is (T,α,γ)-differentially private relative to $\mathcal{F}$ and that $(\mathcal{F},\mu(\pi))$ has a (T,ε)-mapping, then for some $f \in \mathcal{F}$ and every $\mathsf{P} \in \{\mathsf{A},\mathsf{B}\}$, there exist $x, y \in \{0,1\}^s$ such that*

$$\Pr_{v \leftarrow \langle \pi^f(x,y) \rangle}\left[\left|\mathrm{out}^{\mathsf{P}}(v) - \mathsf{IP}(x,y)\right| \leq \Delta := \lambda \cdot \frac{\sqrt{s}}{\log s} \cdot (\tau - \varepsilon)\right] \leq \tau, \tag{12}$$

for every $\tau \leq 1$ with $\tau - \varepsilon \geq \max\{48s\gamma, v\}$. The same holds for X_{out}.

The proof of Theorem 4.16 is given in Sect. 4.2.5. At the end of this subsection, we provide a high-level overview of the steps towards proving Theorem 4.16.

Combining Theorems 3.7 and 4.16 yields an impossibility result for differentially private oracle-aided protocols for approximating the inner-product functionality relative to *simple function* families.

Theorem 4.17. *For a simple function family $\mathcal{F}$ and constants $0 < \nu < 1$ and $\alpha \geq 0$, there exist $\lambda > 0$ and $z \in \mathbb{N}$ such that the following holds. Let $s \geq z$ and let π be an s-bit input, ℓ-query oracle-aided protocol that is (k, α, γ)-differentially private relative to $\mathcal{F}$, for some $k > 2^{10} \cdot \left(\frac{2\ell}{1-\nu}\right)^2$ and $\gamma \leq \frac{\nu}{48 \cdot s}$. Then, for $\beta < \frac{1-\nu}{2}$ and $d \leq \lambda \cdot \nu \cdot \frac{\sqrt{s}}{\log s}$, protocol π is* not *a (β, d)-approximation for the inner-product functionality relative to $\mathcal{F}$.*

Proof. For numbers $0 < \nu < 1$ and $\alpha \geq 0$, let λ and z be as in Theorem 4.16. 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 Definition 4.15). By construction, $\mu(\pi)$ is an ℓ-query, oracle-aided, no-input protocol. Finally, let $\varepsilon = \frac{1-\nu}{2}$.

Theorem 3.7 yields that $(\mathcal{F}, \mu(\pi))$ has a (T, ε)-mapping for $T = 2^{10} \cdot \left(\frac{\ell}{\varepsilon}\right)^2 = 2^{10} \cdot \left(\frac{2\ell}{1-\nu}\right)^2$. Let γ be such that $\gamma \leq \frac{\nu}{48 \cdot s}$. Taking $\tau = \nu + \varepsilon$, it follows that $\tau - \varepsilon \geq \max\{48s\gamma, \nu\}$, as required by Theorem 4.16. Hence, for $k \geq T$, Theorem 4.16 yields that if π is (k, α, γ)-differentially private relative to $\mathcal{F}$, then it is not a (β, d)-approximation for the inner-product functionality relative 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 $\beta \leq \frac{1-\nu}{2}$. □

A High-Level Overview of the Proof of Theorem 4.16. First, in Sect. 4.2.3, we define sampled-input protocols. In such a protocol, the parties have no initial inputs and the output of each party in any execution of the protocol consists of a *sampled input* and an *actual output*. Namely, the parties may sample an input during the execution of the protocol. See Definition 4.18 for a formal definition. Note that this notion is different from the notion of *the sampled-input variant of a protocol* (Definition 4.15): In Definition 4.15, the inputs of the parties are chosen independently at random at the beginning of the protocol, whereas in a sampled-input protocol, the inputs may be chosen during the execution of the protocol and are not necessarily independent or uniform. However, note that for any protocol π, it holds that $\mu(\pi)$ is a sampled-input protocol.

In Theorem 4.21, we provide limitations on the accuracy of a *no-oracle* sampled-input differentially private protocol. The proof of Theorem 4.21 is similar to the proof of Theorem 4.8 with some adaptations that are needed since the sampled inputs of the parties are not necessarily independent.

Then, in Proposition 4.25, we give limitations on the accuracy of an *oracle-aided* sampled-input differentially private protocol π relative to a function family $\mathcal{F}$, such that the pair $(\pi, \mathcal{F})$ has an appropriate mapping to a no-oracle protocol. Proposition 4.25 is proved by combining Definition 3.6 and Proposition 4.21.

In Sect. 4.2.4, we define *uniform-input executions of protocols*, where the correctness and privacy are measured with respect to *uniformly chosen* inputs. In Proposition 4.28, we provide limitations on the accuracy of an *oracle-aided uniform-input* differentially private protocol π relative to a function family $\mathcal{F}$ where $(\mu(\pi), \mathcal{F})$ has an appropriate mapping to a no-oracle protocol. This is proved by the observation that if π is differentially private and a good approximation relative to $\mathcal{F}$, then so is $\mu(\pi)$, since $\mu(\pi)$ works identically to π by Definition 4.15. We then use Proposition 4.25 to obtain that $\mu(\pi)$ cannot be a good approximation for the inner-product functionality relative to $\mathcal{F}$, and hence, we conclude that π cannot be a good approximation as well. One subtle point that we should mention here is that while the inputs of $\mu(\pi)$ are chosen uniformly at random and are independent, the inputs of the *no-oracle* protocol obtained from $\mu(\pi)$ in the transformation applied in the proof of Proposition 4.25 are not necessarily independent (we show in Lemma 4.24 that each of the sampled inputs of the no-oracle protocol is identically distributed as the corresponding sampled input of the oracle-aided protocol, but the joint distribution of the sampled inputs of *both parties* in the no-oracle protocol is not necessarily distributed as the joint distribution in the oracle-aided protocol), and hence, we needed to prove Theorem 4.21 for the stronger case of sampled-input protocols.

Finally, in Sect. 4.2.5, we finish the proof of Theorem 4.16 by showing that a lower bound on the accuracy of a differentially private protocol with respect to uniform inputs implies a similar lower bound for a certain choice of inputs, and hence, we obtain a lower bound on arbitrary protocols.

4.2.3. *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 [22] (stated here as Theorem 4.8) and our main result from Sect. 3 (Theorem 3.7).

Definition 4.18. (*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.[16]

We next extend the notion of good approximations to sampled-input protocols. Intuitively, we require the actual outputs 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.

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

Definition 4.19. (*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-approximationfor g relative to a function family $\mathcal{F}$ and $\mathsf{P} \in \{\mathsf{A}, \mathsf{B}\}$, if for every $f \in \mathcal{F}$, it holds that

$$\Pr_{v \leftarrow \langle \pi^f \rangle} \left[\left| \mathrm{AOut}^{\mathsf{P}}(v) - g\left(\mathrm{SInp}^{\mathsf{A}}(v), \mathrm{SInp}^{\mathsf{B}}(v) \right) \right| > d \right] < \beta. \tag{13}$$

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

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

Definition 4.20. (*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 relative 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

$$\begin{aligned} &\Pr_{f \leftarrow \mathcal{F}, v \leftarrow \langle \pi^f \rangle} \left[\mathsf{D}^f(\mathrm{trans}(v)) = 1 \mid \mathrm{SInp}^{\mathsf{A}}(v) = x \right] \\ &\leq e^{\alpha} \cdot \Pr_{f \leftarrow \mathcal{F}, v \leftarrow \langle \pi^f \rangle} \left[\mathsf{D}^f(\mathrm{trans}(v)) = 1 \mid \mathrm{SInp}^{\mathsf{A}}(v) = x' \right] + \gamma. \end{aligned}$$

The differential privacy of π relative to $\mathcal{F}$ and B is defined analogously.

The protocol π is (k, α, γ)-differentially private relative to $\mathcal{F}$, if it is (k, α, γ)-differentially private relative to $\mathcal{F}$ and both parties.

Lower Bound for No-Oracle Sampled-Input Protocols. The following theorem is a variant of Theorem 4.8, suited for no-oracle, sampled-input protocols.

Theorem 4.21. *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 $\{0,1\}^s$, that π is (α, γ)-differentially private and that $s \geq z$, then

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

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

The main difference between Theorem 4.21 and Theorem 4.8 is that Theorem 4.21 allows X_{In} and Y_{In} to be chosen during the protocol (and hence not necessarily be independent), where Theorem 4.8 assumes that the inputs are selected by an external entity

(hence, needing to require independence of inputs). Note that Theorem 4.21 requires that each of X_{In} and Y_{In} is uniformly distributed over $\{0,1\}^s$, but they are not assumed to be independent. We observe that the proof of Theorem 4.8 given in [22] 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 information-theoretic model with no inputs) are always independent of each other, *given* the transcript. Indeed, Theorem 4.21 easily follows by slight adaptation to the proof of Theorem 4.8, given in [22]. For completeness, however, we include a proof (much of which, taken verbatim from [22]). Let us first describe the outline of the proof given in [22] for Theorem 4.8 (Theorem A.5 in [22]), which is in turn the scheme of our proof. Their proof is twofold:

1. The first part of it is a result about unpredictable bit sources, showing that it is possible to extract a uniform element in $\mathbb{Z}_r$ from the inner product between two independent unpredictable s-bit variables (even given one of these variables), provided that r is somewhat less than $\sqrt{s}$ (for the formal statement see Theorem 4.14).
2. 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.

Finally, combining the above two results yields that every two-party differentially private protocol for approximating the inner-product functionality must incur an error of roughly $r \approx \sqrt{s}$. 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 r, contradicting the fact that it reduces to an almost-uniform element of $\mathbb{Z}_r$.

When proving Theorem 4.21 for the case of sampled-input protocols, we can use the first part of the proof given in [22] for Theorem 4.8, without reproving it, whereas we reprove the second part, with respect to sampled-input protocols, in Claim 4.22.

Proof of Theorem 4.21. 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 4.22, 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 4.23, we define the constants λ and z so that we can apply Theorem 4.14 with respect to such sources. □

Claim 4.22. *There exist numbers* $\{\gamma_{\bar{t}}\}_{\bar{t}\in\mathrm{Supp}(\overline{T})}$ *with* $E\left[\gamma_{\overline{T}}\right] \leq 4\gamma$*, such that the following holds for every* $\bar{t} \in \mathrm{Supp}\left(\overline{T}\right)$*: the random variable* $X_{\mathrm{In}\mid\bar{t}}$ *[resp.,* $Y_{\mathrm{In}\mid\bar{t}}$*] is* $2s\gamma_{\bar{t}}$*-close to some* η*-unpredictable bit source* $\widehat{X}_{\bar{t}}$ *[resp.,* $\widehat{Y}_{\bar{t}}$*].*

Claim 4.23. *There exist 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* $r = 6 \cdot \Delta/\tau$ *and let* c *be the universal constant from Theorem 4.14. Assuming* $s \geq z$*, then* $s \geq c \cdot \frac{r^2}{\eta\beta} \cdot \log\left(\frac{r}{\beta}\right) \cdot \log\left(\frac{r}{\tau/3}\right)$.

We use the above claims for proving Theorem 4.21 Y_{out}, where the proof for X_{out} is analogous. Fix for a moment $\bar{t} \in \mathrm{Supp}\left(\overline{T}\right)$, and note that $X_{\mathrm{In}\mid\bar{t}}$ and $Y_{\mathrm{In}\mid\bar{t}}$ are independent (since π is a no-input, no-oracle protocol). Let $\{\gamma_{\bar{t}}\}_{\bar{t}\in\mathrm{Supp}(\overline{T})}$, λ, z, Δ and r, be the numbers from Claims 4.22 and 4.23. Claim 4.22 yields that

$$\mathrm{SD}\left(\left(Y_{\mathrm{In}\mid\bar{t}}, \mathsf{IP}(X_{\mathrm{In}\mid\bar{t}}, Y_{\mathrm{In}\mid\bar{t}}) \bmod r\right), \left(\widehat{Y}_{\bar{t}}, \mathsf{IP}(\widehat{X}_{\bar{t}}, \widehat{Y}_{\bar{t}}) \bmod r\right)\right) \leq 4s\gamma_{\bar{t}} \tag{14}$$

for some two (independent) η-unpredictable bit sources $\widehat{X}_{\bar{t}}$ and $\widehat{Y}_{\bar{t}}$. Note that by Fact 4.11, both $\widehat{X}_{\bar{t}}$ and $\widehat{Y}_{\bar{t}}$ have min-entropy βs.

Assume $s \geq z$. Since by Claim 4.23 $s \geq c \cdot \frac{r^2}{\eta\beta} \cdot \log\left(\frac{r}{\beta}\right) \cdot \log\left(\frac{r}{\tau/3}\right)$, Theorem 4.14 yields that

$$\mathrm{SD}\left(\left(\widehat{Y}_{\bar{t}}, \mathsf{IP}(\widehat{X}_{\bar{t}}, \widehat{Y}_{\bar{t}}) \bmod r\right), \left(\widehat{Y}_{\bar{t}}, U_r\right)\right) \leq \tau/3, \tag{15}$$

where U_r is independently and uniformly distributed over $\mathbb{Z}_r$. Finally, combining Eqs. (14) and (15) yields that

$$\mathrm{SD}\left(\left(Y_{\mathrm{In}\mid\bar{t}}, \mathsf{IP}(X_{\mathrm{In}\mid\bar{t}}, Y_{\mathrm{In}\mid\bar{t}}) \bmod r\right), \left(\widehat{Y}_{\bar{t}}, U_r\right)\right) \leq 4s\gamma_{\bar{t}} + \tau/3 \tag{16}$$

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

In the following, we assume without loss of generality that B's output is a *deterministic* function f_{B} of $\left(Y_{\mathrm{In}}, \overline{T}\right)$.[17]

[17] For an arbitrary function f_{B}, consider its variant f'_{B} that applies f_{B} on a random view that is consistent with $\left(Y_{\mathrm{In}}, \overline{T}\right)$. 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) $\left(Y_{\mathrm{In}}, \overline{T}\right)$. 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}.

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

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

The second inequality holds by Eq. (16), the third one since $\text{E}[\gamma_{\overline{T}}] \le 4\gamma$, and the last one since $\frac{\Delta}{r} = \gamma/6$ and since, by assumption, $\tau \ge 48s\gamma$.

Proof of Claim 4.22. 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

$$\begin{aligned}
\rho_X\left(i, x, \bar{t}\right) :&= \frac{\Pr\left[X_i = 0 \mid X_1 = x_1, \ldots X_{i-1} = x_{i-1}, X_{i+1} = x_{i+1}, \ldots, X_s = x_s, \overline{T} = \bar{t}\right]}{\Pr\left[X_i = 1 \mid X_1 = x_1, \ldots X_{i-1} = x_{i-1}, X_{i+1} = x_{i+1}, \ldots, X_s = x_s, \overline{T} = \bar{t}\right]} \\
&= \frac{\Pr\left[\overline{T} = \bar{t} \mid X_1 = x_1, \ldots X_{i-1} = x_{i-1}, X_i = 0, X_{i+1} = x_{i+1}, \ldots, X_s = x_s\right]}{\Pr\left[\overline{T} = \bar{t} \mid X_1 = x_1, \ldots X_{i-1} = x_{i-1}, X_i = 1, X_{i+1} = x_{i+1}, \ldots, X_s = x_s\right]},
\end{aligned} \tag{17}$$

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 4.13 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 ($\text{E}\left[\gamma_{\overline{T}}\right] \le 4\gamma$), applying Lemma 4.9 with $\nu = 1.1$ yields that

$$\begin{aligned}
\max\left\{\Pr_{(x,\bar{t}) \leftarrow (X_{\text{In}}, \overline{T})}\left[\left(i, x, \bar{t}\right) \in \mathcal{S}_X\right], \Pr_{(y,\bar{t}) \leftarrow (Y_{\text{In}}, \overline{T})}\left[\left(i, y, \bar{t}\right) \in \mathcal{S}_Y\right]\right\} &\le \gamma \cdot \frac{1 + e^{-(1.1+\alpha)}}{1 - e^{-1.1}} \\
&< 2\gamma
\end{aligned} \tag{18}$$

for every $i \in [s]$,[18] and we conclude that

$$\begin{aligned}
\mathrm{E}\left[\gamma_{\overline{T}}\right] &\leq \mathop{\mathrm{E}}_{\overline{t}\leftarrow\overline{T}}\left[\Pr_{i\leftarrow[s],x\leftarrow X_{\mathrm{In}\mid\overline{t}}}\left[(i,x,\overline{t})\in\mathcal{S}_X\right]+\Pr_{i\leftarrow[s],y\leftarrow Y_{\mathrm{In}\mid\overline{t}}}\left[(i,y,\overline{t})\in\mathcal{S}_Y\right]\right] \\
&\leq \mathrm{E}\left[2\cdot\max\left\{\Pr_{(x,\overline{t})\leftarrow(X_{\mathrm{In}},\overline{T})}\left[(i,x,\overline{t})\in\mathcal{S}_X\right],\right.\right. \\
&\qquad\left.\left.\Pr_{(y,\overline{t})\leftarrow(Y_{\mathrm{In}},\overline{T})}\left[(i,y,\overline{t})\in\mathcal{S}_Y\right] : i\in[s]\right\}\right] \\
&< 4\gamma. \qquad (19)
\end{aligned}$$

□

Proof of Claim 4.23. Let $\lambda_1 = \lambda/\eta > 0$, where $\lambda \in (0,1)$ will be determined later. Since $\Delta = \lambda \cdot \frac{\sqrt{s}}{\log s} \cdot \tau$, we have that $s = \left(\frac{\Delta\cdot\log s}{\lambda\cdot\tau}\right)^2 = \left(\frac{\Delta\cdot\log s}{\lambda_1\cdot\tau\cdot\eta}\right)^2$. Let $z_1 = z_1(\lambda,\alpha)$ be such that $s \geq z_1$ implies $\log s \geq \lambda_1$. By the above, we have that for every $s \geq z_1$, it holds that $s = \left(\frac{\Delta}{\tau\eta}\cdot\frac{\log s}{\lambda_1}\right)^2 \geq \left(\frac{\Delta}{\tau\eta}\right)^2$. Hence, $\log s \geq \log\left(\frac{\Delta}{\tau\eta}\right)^2 \geq \log\left(\frac{\Delta}{\tau\eta}\right)$. Recalling that $r = 6\cdot\Delta/\tau$, it holds that $\frac{\Delta}{\tau\eta} = \frac{r}{6\eta}$ and we obtain that

$$\begin{aligned}
s &\geq \frac{1}{\lambda_1^2}\cdot\left(\frac{\Delta}{\tau\eta}\right)^2\cdot\left(\log\left(\frac{\Delta}{\tau\eta}\right)\right)^2 \qquad (20)\\
&= \frac{1}{\lambda_1^2}\cdot\frac{r^2}{36\cdot\eta^2}\cdot\left(\log\left(\frac{r}{6\eta}\right)\right)^2 \\
&\geq \frac{1}{36\cdot\lambda_1^2}\cdot\frac{r^2}{\eta\beta}\cdot\left(\log\left(\frac{r}{6\eta}\right)\right)^2
\end{aligned}$$

for every $s \geq z_1$, where the last inequality holds since, by inspection, $\beta \geq \eta$ (recall that $\beta = \log(1+\eta)$).

Let $\lambda = \lambda(\alpha,\nu) \in (0,1)$ be such that $\frac{1}{36\cdot\lambda_1^2} \geq 4c$. Let $z_2 = z_2(\lambda,\alpha)$ be such that $s \geq z_2$ implies $r \geq 36\cdot\eta^2\cdot\max\left\{\frac{1}{\beta},\frac{3}{\tau}\right\}$, and let $z = \max\{z_1,z_2\}$. Fix $s \geq z$. By the above, $\left(\frac{r}{6\cdot\eta}\right)^2 \geq \max\left\{\frac{r}{\beta},\frac{r}{\tau/3}\right\}$, and therefore, $2\cdot\log\left(\frac{r}{6\eta}\right) \geq \max\left\{\log\left(\frac{r}{\beta}\right),\log\left(\frac{r}{\tau/3}\right)\right\}$. Thus, Eq. (20) yields that

[18] We note that Lemma 4.9 is stated for differentially private mechanisms. Nevertheless, its proof for sampled-input protocols readily follows from the original proof.

$$\begin{aligned} s &\geq 4 \cdot c \cdot \frac{r^2}{\eta\beta} \cdot \left(\log\left(\frac{r}{6\eta}\right)\right)^2 \\ &\geq 4 \cdot c \cdot \frac{r^2}{\eta\beta} \cdot \frac{1}{2} \cdot \log\left(\frac{r}{\beta}\right) \cdot \frac{1}{2} \cdot \log\left(\frac{r}{\tau/3}\right) \\ &\geq c \cdot \frac{r^2}{\eta\beta} \cdot \log\left(\frac{r}{\beta}\right) \cdot \log\left(\frac{r}{\tau/3}\right), \end{aligned}$$

concluding the claim's proof. □

Lower Bound for Oracle-Aided, Sampled-Input Protocols. We now use Definition 3.6 to give a variant of Theorem 4.21 for (sampled-input) *oracle-aided* protocols with an appropriate mapping to a no-oracle protocol. 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 4.24. *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 relative to $\mathcal{F}$ and party P, and satisfies (T, α, γ)-differential privacy relative to $\mathcal{F}$. Then, the no-oracle, s-bit sampled-input protocol $\widetilde{\pi} = (\widetilde{\mathsf{A}}, \widetilde{\mathsf{B}})$, 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 $\widetilde{\mathsf{A}}$ (resp. $\widetilde{\mathsf{B}}$) are identically distributed.

Proof. Let $\widetilde{\pi} = (\widetilde{\mathsf{A}}, \widetilde{\mathsf{B}})$ and Map be the no-input, no-oracle protocol and oracle-aided algorithm guaranteed by Definition 3.6 with respect to π and $\mathcal{F}$. We first argue that $\widetilde{\pi}$ 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

$$\begin{aligned} &\Pr_{\tilde{v} \leftarrow \langle \widetilde{\pi} \rangle}\left[\tilde{\mathsf{D}}\left(\mathsf{trans}\left(\tilde{v}\right)\right) = 1 \mid \mathrm{SInp}^{\widetilde{\mathsf{A}}}\left(\tilde{v}\right) = x\right] \\ &\quad > e^{\alpha} \cdot \Pr_{\tilde{v} \leftarrow \langle \widetilde{\pi} \rangle}\left[\tilde{\mathsf{D}}\left(\mathsf{trans}\left(\tilde{v}\right)\right) = 1 \mid \mathrm{SInp}^{\widetilde{\mathsf{A}}}\left(\tilde{v}\right) = x'\right] + \gamma \end{aligned} \tag{21}$$

for some $x, x' \in \{0,1\}^s$ with $H_d\left(x, x'\right) = 1$. 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 $\mathsf{Map}^f\left(\bar{t}\right)$. We claim that

$$\begin{aligned} &\Pr_{f \leftarrow \mathcal{F}, v \leftarrow \langle \pi^f \rangle}\left[\mathsf{D}^f\left(\mathsf{trans}\left(v\right)\right) = 1 \mid \mathrm{SInp}^{\mathsf{A}}\left(v\right) = x\right] \\ &\quad > e^{\alpha} \cdot \Pr_{f \leftarrow \mathcal{F}, v \leftarrow \langle \pi^f \rangle}\left[\mathsf{D}^f\left(\mathsf{trans}\left(v\right)\right) = 1 \mid \mathrm{SInp}^{\mathsf{A}}\left(v\right) = x'\right] + \gamma \end{aligned} \tag{22}$$

To see that Eq. (22) holds, note that by the furthermore statement of the first item in Definition 3.6, the transcript together with the sampled input of $\widetilde{\mathsf{A}}$ in a random execution of $\widetilde{\pi}$ (i.e. $\left(\mathsf{trans}\left(\tilde{v}\right), \mathrm{SInp}^{\widetilde{\mathsf{A}}}\left(\tilde{v}\right)\right)$, where $\tilde{v} \leftarrow \langle\widetilde{\pi}\rangle$) is (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. $\left(\mathsf{Map}^{f}\left(\mathsf{trans}\left(v\right)\right), \mathrm{SInp}^{\mathsf{A}}\left(v\right)\right)$, where $v \leftarrow \langle\pi^{f}\rangle$ for $f \leftarrow \mathcal{F}$). In addition, by Definition 3.6, 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 $\widetilde{\pi}$ must be (α, γ)-differentially private.

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

$$\Pr_{\tilde{v} \leftarrow \langle\widetilde{\pi}\rangle}\left[\left|\mathrm{AOut}^{\mathsf{P}}\left(\tilde{v}\right) - g\left(\mathrm{SInp}^{\widetilde{\mathsf{A}}}\left(\tilde{v}\right), \mathrm{SInp}^{\widetilde{\mathsf{B}}}\left(\tilde{v}\right)\right)\right| \geq d\right] < \beta + \varepsilon. \tag{23}$$

By the first item in Definition 3.6, 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 $\widetilde{\pi}$. Formally, if we let $\mathcal{D}_{\widetilde{\pi}} = \left(\mathrm{SInp}^{\widetilde{\mathsf{A}}}\left(\tilde{v}\right), \mathrm{SInp}^{\widetilde{\mathsf{B}}}\left(\tilde{v}\right), \mathrm{AOut}^{\mathsf{P}}\left(\tilde{v}\right)\right)_{\tilde{v} \leftarrow \langle\widetilde{\pi}\rangle}$ and $\mathcal{D}_{\pi} = \left(\mathrm{SInp}^{\mathsf{A}}\left(v\right), \mathrm{SInp}^{\mathsf{B}}\left(v\right), \mathrm{AOut}^{\mathsf{P}}\left(v\right)\right)_{f \leftarrow \mathcal{F}, v \leftarrow \langle\pi^{f}\rangle}$, then we have that $\mathrm{SD}\left(\mathcal{D}_{\widetilde{\pi}}, \mathcal{D}_{\pi}\right) \leq \varepsilon$. Hence, Eq. (23) follows from the accuracy of π, i.e. since we have that

$$\Pr_{f \leftarrow \mathcal{F}, v \leftarrow \langle\pi^{f}\rangle}\left[\left|\mathrm{AOut}^{\mathsf{P}}\left(v\right) - g\left(\mathrm{SInp}^{\mathsf{A}}\left(v\right), \mathrm{SInp}^{\mathsf{B}}\left(v\right)\right)\right| \geq d\right] < \beta \tag{24}$$

To verify this, let $\mathcal{S} = \{(x, y, w) \in \mathrm{Supp}\,\mathcal{D}_{\widetilde{\pi}} \colon |g(x, y) - w| \geq d\}$ and observe that the probability of falling into S according to $\mathcal{D}_{\widetilde{\pi}}$ 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 $\mathcal{D}_{\widetilde{\pi}}$ and $\mathcal{D}_{\pi}$.

The furthermore statement follows from the furthermore statement of the first item in Definition 3.6. □

We now combine Lemma 4.24 and Theorem 4.21 to prove a lower bound on the accuracy of oracle-aided, sampled-input protocols that are differentially private.

Proposition 4.25. *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, let $s \geq z$ and let $\pi = (\mathsf{A}, \mathsf{B})$ be an oracle-aided, s-bit sample-input protocol. For $f \in \mathcal{F}$, let X_{In}^{f} and Y_{In}^{f} be the sampled inputs of A and B, respectively, and let X_{out}^{f} and Y_{out}^{f} be the actual outputs of A and B, respectively, induced by a random execution of π^{f}.*

Assume that both X_{In}^{f} and Y_{In}^{f} are uniformly distributed over $\{0,1\}^{s}$ for every $f \in \mathcal{F}$, that π is (T, α, γ)-differentially private relative to $\mathcal{F}$ and that the pair $(\mathcal{F}, \pi)$ has a (T, ε)-mapping, *then for some $f \in \mathcal{F}$, it holds that*

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

for every $1 \geq \tau$ *with* $\tau - \varepsilon \geq \max\{48s\gamma, \nu\}$. *The same holds for* X_{out}^{f}.

Proof. Given values for ν and α set λ and z to be as in Theorem 4.21. Let $\mathcal{F}$ and π be as in the statement of the proposition. Since π is assumed to be (T, α, γ)-differentially private relative to $\mathcal{F}$, and since the pair $(\mathcal{F}, \pi)$ is assumed to have a (T, ε)-mapping, it follows from Lemma 4.24 that the no-oracle, sampled-input protocol $\widetilde{\pi}$ (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 4.24 that the same holds for the sampled inputs of both parties in $\widetilde{\pi}$. Hence, Theorem 4.21 yields that for $s \geq z$ and τ such that $\tau' := \tau - \varepsilon \geq \max\{48s\gamma, \nu\}$, it holds that $\widetilde{\pi}$ is not a $\left(1 - \tau', \Delta\right)$-SI-approximation for $\Delta = \lambda \cdot \frac{\sqrt{s}}{\log s} \cdot \tau'$. By Lemma 4.24, π is not a $\left(1 - \tau' - \varepsilon, \Delta\right)$-SI-approximation, namely $\Pr\left[|Y_{\text{out}} - \mathsf{IP}(X_{\text{In}}, Y_{\text{In}})| \leq \Delta\right] \leq \tau' + \varepsilon = \tau$. □

4.2.4. *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 map a uniform-input execution of a with-input protocol to the sampled-input variant of this protocol (as defined in Definition 4.15). Roughly speaking, in the sampled-input variant of a protocol, the parties sample their inputs at random at the beginning of an execution.

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 4.26. (*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 relative to function family $\mathcal{F}$ and $\mathsf{P} \in \{\mathsf{A}, \mathsf{B}\}$, if for every $f \in \mathcal{F}$, it holds that

$$\Pr_{x,y \leftarrow \{0,1\}^s, v \leftarrow \langle \pi^f(x,y) \rangle}\left[\left|\text{out}^{\mathsf{P}}(v) - g(x,y)\right| > d\right] < \beta. \tag{25}$$

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

The following observation allows us to use the lower bound stated in Lemma 4.24, to derive a similar bound for with-input protocols, when the inputs of the parties are chosen uniformly at random.

Lemma 4.27. *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 *relative to* $\mathcal{F}$ *and party* P*, and*

is (k, α, γ)-differential privacy relative to $\mathcal{F}$. Then, $\mu(\pi)$ is a (β, d)-SI-approximation for g relative to $\mathcal{F}$ and P, and (k, α, γ)-differentially private relative to $\mathcal{F}$.

Proof. Immediate by definition. □

Combining Proposition 4.25 and Lemma 4.27 yields the following result.

Proposition 4.28. *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, let $s \geq z$ and let $\pi = (\mathsf{A}, \mathsf{B})$ be an oracle-aided, s-bit input protocol. For $f \in \mathcal{F}$, let X_{In}^f and Y_{In}^f be the inputs of A and B, respectively, and let X_{out}^f and ${Y_{\text{out}}}^f$ be the outputs of A and B, respectively, induced by a random execution of ${\pi}^f$.*

*Assume that both X_{In}^f and Y_{In}^f are independently and uniformly chosen from $\{0,1\}^s$ for every $f \in \mathcal{F}$, that π is (T, α, γ)-differentially private relative 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). Then, for some $f \in \mathcal{F}$, it holds that*

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

for every $1 \geq \tau$ with $\tau - \varepsilon \geq \max\{48s\gamma, \nu\}$. The same holds for X_{out}^f.

Proof. Given values for ν and α, set λ and z to be as in Proposition 4.25. 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 Definition 4.15). 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 relative to $\mathcal{F}$, it follows by Lemma 4.27 that $\mu(\pi)$ is also (T, α, γ)-differentially private relative to $\mathcal{F}$. Since the pair $(\mathcal{F}, \mu(\pi))$ is assumed to have a (T, ε)-mapping, it follows from Proposition 4.25 that for $s \geq z$ and τ such that $\tau' := \tau - \varepsilon \geq \max\{48s\gamma, \nu\}$, the protocol $\mu(\pi)$ is not a $(1 - \tau, \Delta)$-SI-approximation for $\Delta = \lambda \cdot \frac{\sqrt{s}}{\log s} \cdot \tau'$. Hence, by Lemma 4.27, π is not a $(1 - \tau, \Delta)$-random approximation, namely $\Pr\left[|Y_{\text{out}} - \mathsf{IP}(X_{\text{In}}, Y_{\text{In}})| \leq \Delta\right] \leq \tau$. □

4.2.5. *Limits on Arbitrary Protocols: Proving Theorem 4.16*

The results presented in the previous section yield the lower bounds of Sect. 4.2.2 in a straightforward manner. That is, the lower bound on the accuracy of differentially private protocols, with respect to executions where inputs are selected uniformly at random, 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.

Proof of Theorem 4.16. Immediate, by taking x and y that maximize the probability in Eq. (12), and using Proposition 4.28 to bound this probability from below. □

4.3. *Secure Sampling*

In this section, we apply our main result to show that when given access to a random member of a simple function family (e.g. the all-function family), no-oracle-aided protocol can securely sample a distribution G that cannot be (almost) securely sampled by a no-oracle protocol.

In semi-honest no-input secure sampling, two parties with no-private inputs wish to compute some (possibly randomized) functionality privately and correctly. Let $G = (G_{\mathsf{A}}, G_{\mathsf{B}})$ be a distribution over $\mathcal{A} \times \mathcal{B}$, where G_{A} and G_{B} denote its marginal distributions over $\mathcal{A}$ and $\mathcal{B}$, respectively. The parties A and B wish to perform a computation, where party A learns $g_{\mathcal{A}}$ and party B learns $g_{\mathcal{B}}$ for $g = (g_{\mathcal{A}}, g_{\mathcal{B}}) \leftarrow G$, but nothing else. Since the parties are semi-honest, they will always follow the prescribed protocol. A corrupted party, however, may try to use its view in the computation to infer additional information after the computation terminates.

4.3.1. *Standard Definitions*

Definition 4.29. (*No-input secure sampling*) Let $\mathcal{A}$ and $\mathcal{B}$ be sets, and let $G = (G_{\mathsf{A}}, G_{\mathsf{B}})$ be a distribution over $\mathcal{A} \times \mathcal{B}$, where G_{A} and G_{B} denote its marginal distributions over $\mathcal{A}$ and $\mathcal{B}$, respectively. A two-party oracle-aided protocol $\pi = (\mathsf{A}, \mathsf{B})$ is a (k, δ)-secure protocol for G relative to a function family $\mathcal{F}$, if the following conditions hold.

Correctness: π is a δ-correct implementation of G relative to $\mathcal{F}$. That is

$$\mathrm{SD}\left(\left(\mathrm{out}^{\mathsf{A}}(v), \mathrm{out}^{\mathsf{B}}(v)\right)_{v \leftarrow \langle \pi^f \rangle}, G\right) \leq \delta$$

for every $f \in \mathcal{F}$.

Privacy: π is a (k, δ)-private implementation of G relative to $\mathcal{F}$: for every $P \in \{\mathsf{A}, \mathsf{B}\}$, there exists an oracle-aided algorithm (simulator) $\mathrm{Sim}_{\mathsf{P}}$ such that:

$$\begin{aligned} \mathop{\mathrm{E}}_{f \leftarrow \mathcal{F}} \Bigg[\Bigg| \Pr\left[\mathsf{D}^f\left(\left(\mathrm{Sim}_{\mathsf{P}}^f(g_{\mathsf{P}}), g_{\mathsf{A}}, g_{\mathsf{B}}\right)_{(g_{\mathsf{A}}, g_{\mathsf{B}}) \leftarrow G_{\mathsf{A}}}\right) = 1\right] \\ - \Pr\left[\mathsf{D}^f\left(\left(v_{\mathsf{P}}, \mathrm{out}^{\mathsf{A}}(v), \mathrm{out}^{\mathsf{B}}(v)\right)_{v \leftarrow \langle \pi^f \rangle}\right) = 1\right] \Bigg| \Bigg] \leq \delta, \end{aligned}$$

for any k-query distinguisher D.[19]

A protocol π is a δ-secure (no-oracle) implementation of G, if it is a $(0, \delta)$-secure implementation of G relative to the trivial function family—the function family whose only member returns $\perp$ on every query. A distribution G is δ-trivial, if G has a δ-secure no-oracle implementation.

[19] A stricter security definition would restrict also the query complexity of the simulator $\mathrm{Sim}_{\mathsf{P}}$ (and not only that of the distinguisher). We use the above more relaxed version since our goal is proving an impossibility result for such secure sampling.

Remark 4.30. (privacy for no-oracle protocols, alternative definition). The success probability of the "best" distinguisher in the information-theoretic model is defined by the statistical distance between the output of a real execution: P's view and the parties local outputs, and the output of the simulation: the simulator's output and the sample from the distribution.

Furthermore, in the information-theoretic model, it suffices to assume that a party's view contains only the communication transcript (i.e. without its random coins). This is because conditioned on the transcript, the parties' views are in a product distribution, and thus, the party's view is a *function* of the transcript and its local output.

The following is an alternative (and by the above, equivalent) definition for δ-privacy of (no-oracle) protocols in the information-theoretic model: a protocol $\widetilde{\pi} = \left(\widetilde{\mathsf{A}}, \widetilde{\mathsf{B}}\right)$ is a δ-private implementation of a distribution G if for every $P \in \left\{\widetilde{\mathsf{A}}, \widetilde{\mathsf{B}}\right\}$ there exists an algorithm (simulator) $\widetilde{\mathrm{Sim}_{\widetilde{\mathsf{P}}}}$ such that,

$$\mathrm{SD}\left(\left(\widetilde{\mathrm{Sim}_{\widetilde{\mathsf{P}}}}\left(g_{\mathsf{P}}\right), g_{\mathsf{A}}, g_{\mathsf{B}}\right)_{(g_{\mathsf{A}}, g_{\mathsf{B}}) \leftarrow G}, \left(\mathsf{trans}\left(v\right), \mathrm{out}^{\mathsf{A}}\left(v\right), \mathrm{out}^{\mathsf{B}}\left(v\right)\right)_{v \leftarrow \langle \widetilde{\pi} \rangle}\right) \leq \delta.$$

4.3.2. *Limits on Oracle-Aided Secure Sampling*

In the language of the above definitions, the main result of this section is stated as follows.

Theorem 4.31. *Let $\mathcal{F}$ be a function family, and let π be an oracle-aided protocol that is a (T, δ)-secure oracle-aided implementation of a distribution G with respect to $\mathcal{F}$. Assume that $(\mathcal{F}, \pi)$ has a* (T, δ')-mapping*, then G is $(\delta + \delta')$-trivial.*

Proof. Let $\mathcal{F}$ and π be as above and let $\widetilde{\pi} = (\widetilde{\mathsf{A}}, \widetilde{\mathsf{B}})$ and Map be the (T, δ')-mapping for $(\mathcal{F}, \pi)$, we will prove that $\widetilde{\pi}$ is a (no oracle) $(\delta + \delta')$-secure implementation of G. Since $(\widetilde{\pi}, \mathsf{Map})$ is a (T, δ')-mapping, it follows (see Definition 3.6:1) that

$$\begin{aligned}\mathrm{SD}\Big(&\left(\mathrm{out}^{\widetilde{\mathsf{A}}}\left(v\right), \mathrm{out}^{\widetilde{\mathsf{B}}}\left(v\right), \mathsf{trans}(v)\right)_{v \leftarrow \langle \widetilde{\pi} \rangle}, \left(\mathrm{out}^{\mathsf{A}}\left(v\right), \mathrm{out}^{\mathsf{B}}\left(v\right),\right.\\ &\left.\left.\mathsf{Map}^{f}\left(\mathsf{trans}(v)\right)\right)_{f \leftarrow \mathcal{F}, v \leftarrow \langle \pi^{f} \rangle}\Big) \leq \delta'\end{aligned} \tag{26}$$

Hence, the δ-correctness of π yields that

$$\mathrm{SD}\left(\left(\mathrm{out}^{\widetilde{\mathsf{A}}}\left(v\right), \mathrm{out}^{\widetilde{\mathsf{B}}}\left(v\right)\right)_{v \leftarrow \langle \widetilde{\pi} \rangle}, G\right) \leq \delta + \delta',$$

and thus, $\widetilde{\pi}$ is $(\delta + \delta')$-correct implementation of G.

For the privacy property, we construct a no-oracle simulator $\widetilde{\mathrm{Sim}_{\widetilde{\mathsf{A}}}}$ for party $\widetilde{\mathsf{A}}$ (a simulator $\widetilde{\mathsf{B}}$ can be constructed analogously). Let $\mathrm{Sim}_{\mathsf{A}}$ be the oracle-aided simulator guaranteed to exist for π and A. Fix a T-query distinguisher D. The assumption that $\mathrm{Sim}_{\mathsf{A}}$ is a good simulator yields that

$$\mathop{\mathrm{E}}_{f\leftarrow\mathcal{F}}\left[\left|\Pr\left[\mathsf{D}^f\left(\left(\mathrm{Sim}_{\mathsf{A}}^f\left(g_{\mathsf{A}}\right),g_{\mathsf{A}},g_{\mathsf{B}}\right)_{(g_{\mathsf{A}},g_{\mathsf{B}})\leftarrow G_{\mathsf{A}}}\right)=1\right]\right.\right.$$
$$\left.\left.-\Pr\left[\mathsf{D}^f\left(\left(v_{\mathsf{A}},\mathrm{out}^{\mathsf{A}}\left(v\right),\mathrm{out}^{\mathsf{B}}\left(v\right)\right)_{v\leftarrow\langle\pi^f\rangle}\right)=1\right]\right|\right]\leq\delta,$$

and therefore,

$$\sum_{f\in\mathcal{F}}\Pr\left[f\right]\cdot\left|\Pr\left[\mathsf{D}^f\left(\left(\mathrm{Sim}_{\mathsf{A}}^f\left(g_{\mathsf{A}}\right),g_{\mathsf{A}},g_{\mathsf{B}}\right)_{(g_{\mathsf{A}},g_{\mathsf{B}})\leftarrow G_{\mathsf{A}}}\right)=1\right]\right.$$
$$\left.-\Pr\left[\mathsf{D}^f\left(\left(v_{\mathsf{A}},\mathrm{out}^{\mathsf{A}}\left(v\right),\mathrm{out}^{\mathsf{B}}\left(v\right)\right)_{v\leftarrow\langle\pi^f\rangle}\right)=1\right]\right|\leq\delta.$$

By the triangle inequality

$$\sum_{f\in\mathcal{F}}\Pr\left[f\right]\cdot\left|\Pr\left[\mathsf{D}^f\left(\left(\mathrm{Sim}_{\mathsf{A}}^f\left(g_{\mathsf{A}}\right),g_{\mathsf{A}},g_{\mathsf{B}}\right)_{(g_{\mathsf{A}},g_{\mathsf{B}})\leftarrow G_{\mathsf{A}}}\right)=1\right]\right.$$
$$\left.-\Pr\left[\mathsf{D}^f\left(\left(v_{\mathsf{A}},\mathrm{out}^{\mathsf{A}}\left(v\right),\mathrm{out}^{\mathsf{B}}\left(v\right)\right)_{v\leftarrow\langle\pi^f\rangle}\right)=1\right]\right|$$
$$\geq\left|\sum_{f\in\mathcal{F}}\Pr[f]\cdot\left(\Pr\left[\mathsf{D}^f\left(\left(\mathrm{Sim}_{\mathsf{A}}^f\left(g_{\mathsf{A}}\right),g_{\mathsf{A}},g_{\mathsf{B}}\right)_{(g_{\mathsf{A}},g_{\mathsf{B}})\leftarrow G_{\mathsf{A}}}\right)=1\right]\right.\right.$$
$$\left.\left.-\Pr\left[\mathsf{D}^f\left(\left(v_{\mathsf{A}},\mathrm{out}^{\mathsf{A}}\left(v\right),\mathrm{out}^{\mathsf{B}}\left(v\right)\right)_{v\leftarrow\langle\pi^f\rangle}\right)=1\right]\right)\right|$$

and we conclude that

$$\left|\Pr_{f\leftarrow\mathcal{F},(g_{\mathsf{A}},g_{\mathsf{B}})\leftarrow G_{\mathsf{A}}}\left[\mathsf{D}^f\left(\left(\mathrm{Sim}_{\mathsf{A}}^f\left(g_{\mathsf{A}}\right),g_{\mathsf{A}},g_{\mathsf{B}}\right)\right)=1\right]\right.$$
$$\left.-\Pr_{f\leftarrow\mathcal{F},v\leftarrow\langle\pi^f\rangle}\left[\mathsf{D}^f\left(\left(v_{\mathsf{A}},\mathrm{out}^{\mathsf{A}}\left(v\right),\mathrm{out}^{\mathsf{B}}\left(v\right)\right)\right)=1\right]\right|\leq\delta\tag{27}$$

Recalling that Map makes at most T-queries, Eq. (27) yields that

$$\mathrm{SD}\left(\left(\mathsf{Map}^f\left(\mathsf{trans}\left(\mathrm{Sim}_{\mathsf{A}}^f\left(g_{\mathsf{A}}\right)\right)\right),g_{\mathsf{A}},g_{\mathsf{B}}\right)_{f\leftarrow\mathcal{F},(g_{\mathsf{A}},g_{\mathsf{B}})\leftarrow G},\right.$$
$$\left.\left(\mathsf{Map}^f\left(\mathsf{trans}\left(v\right)\right),\mathrm{out}^{\mathsf{A}}\left(v\right),\mathrm{out}^{\mathsf{B}}\left(v\right)\right)_{f\leftarrow\mathcal{F},v\leftarrow\langle\pi^f\rangle}\right)\leq\delta\tag{28}$$

□

The no-oracle simulator $\widetilde{\mathrm{Sim}_{\tilde{\mathsf{A}}}}$ is defined as follows.

Algorithm 4.32. $(\widetilde{\mathrm{Sim}_{\widetilde{\mathsf{A}}}})$.

Input: $g_{\mathsf{A}} \in \mathrm{Supp}\,(G_{\mathsf{A}})$

Operation:

1. *Let* $f \leftarrow \mathcal{F}$ *and let* $v_{\mathsf{A}} \leftarrow \mathrm{Sim}^f_{\mathsf{A}}(g_{\mathsf{A}})$.
2. *Output* $\mathsf{Map}^f\,(\mathsf{trans}\,(v_{\mathsf{A}}))$.

Note that

$$\begin{aligned}&\left(\mathsf{Map}^f\left(\mathsf{trans}\left(\mathrm{Sim}^f_{\mathsf{A}}\,(g_{\mathsf{A}})\right)\right), g_{\mathsf{A}}, g_{\mathsf{B}}\right)_{f\leftarrow\mathcal{F},(g_{\mathsf{A}},g_{\mathsf{B}})\leftarrow G}\\ &\quad\equiv\left(\widetilde{\mathrm{Sim}_{\widetilde{\mathsf{A}}}}\,(g_{\mathsf{A}}), g_{\mathsf{A}}, g_{\mathsf{B}}\right)_{(g_{\mathsf{A}},g_{\mathsf{B}})\leftarrow G},\end{aligned}$$

and that by Eq. (26)

$$\begin{aligned}&\mathrm{SD}\left(\left(\mathsf{Map}^f\,(\mathsf{trans}\,(v)), \mathrm{out}^{\mathsf{A}}(v), \mathrm{out}^{\mathsf{B}}(v)\right)_{f\leftarrow\mathcal{F},v\leftarrow\langle\pi^f\rangle},\right.\\ &\left.\left(\mathsf{trans}(v), \mathrm{out}^{\widetilde{\mathsf{A}}}(v), \mathrm{out}^{\widetilde{\mathsf{B}}}(v)\right)_{v\leftarrow\langle\widetilde{\pi}\rangle}\right)\leq\delta'.\end{aligned}$$

We conclude that

$$\begin{aligned}&\mathrm{SD}\left(\left(\widetilde{\mathrm{Sim}_{\widetilde{\mathsf{A}}}}\,(g_{\mathsf{A}}), g_{\mathsf{A}}, g_{\mathsf{B}}\right)_{(g_{\mathsf{A}},g_{\mathsf{B}})\leftarrow G}, \left(\mathsf{trans}(v), \mathrm{out}^{\widetilde{\mathsf{A}}}(v), \mathrm{out}^{\widetilde{\mathsf{B}}}(v)\right)_{v\leftarrow\langle\widetilde{\pi}\rangle}\right)\\ &\leq \mathrm{SD}\left(\left(\mathsf{Map}^f\left(\mathsf{trans}\left(\mathrm{Sim}^f_{\mathsf{A}}\,(g_{\mathsf{A}})\right)\right), g_{\mathsf{A}}, g_{\mathsf{B}}\right)_{f\leftarrow\mathcal{F},(g_{\mathsf{A}},g_{\mathsf{B}})\leftarrow G},\right.\\ &\quad\left.\left(\mathsf{Map}^f\,(\mathsf{trans}\,(v)), \mathrm{out}^{\mathsf{A}}(v), \mathrm{out}^{\mathsf{B}}(v)\right)_{f\leftarrow\mathcal{F},v\leftarrow\langle\pi^f\rangle}\right)+\delta'\\ &\leq \delta+\delta'.\end{aligned}$$

The first inequality is by the above two equations and the triangle inequality, and last one by Eq. (28). It follows, see Remark 4.30, that $\widetilde{\pi}$ is $(\delta+\delta')$-private.

Combining Theorems 4.31 and 3.7 yields the following result.

Theorem 4.33. *Let $\mathcal{F}$ be a simple function family. Let $k, \ell \in \mathbb{N}$, let $\delta \in [0,1]$ be such that $k \geq 2^{10}\cdot\left(\frac{\ell}{\delta}\right)^2$, and let G be a* non *2δ-trivial distribution. Then, G has* no *ℓ-query, (k,δ)-secure oracle-aided implementation relative to $\mathcal{F}$.*

Proof. Let π be an ℓ-query oracle-aided protocol. Theorem 3.7 yields that $(\mathcal{F},\pi)$ has a (T,δ)-mapping for $T = 2^{10}\cdot\left(\frac{\ell}{\delta}\right)^2$. Assume that π is a (T,δ)-secure oracle-aided implementation of G relative to $\mathcal{F}$, then Theorem 4.31 would yield that G is 2δ-trivial. Hence, G has no ℓ-query (k,δ)-secure implementation relative to $\mathcal{F}$. □

4.4. *Applications to the All-Function Family and Black-Box Reductions to One-way Functions*

In this section, we use Theorem 3.7 to derive limits on possible implementations relative to the all-function family (i.e. the set of "random functions"). We then use the above and the fact that a random element of the all-function family is hard to invert, to give limits on fully black-box reductions to one-way functions.

4.4.1. *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 4.34. (*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 PPT 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 4.35. (*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.

The following fact is immediate.

Fact 4.36. *For every $n \in \mathbb{N}$, the family $\mathcal{F}_{\mathrm{AF}_n}$ is simple.*

It is well known (cf., [12,18]) that random members of the all-function family are "one way". Specifically, we use the following fact.

Fact 4.37. *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 pair of PPTM (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 [26] 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 4.38 and 4.40 for concrete examples.[20]

4.4.2. *Key-Agreement Protocols*

Following Definition 4.1 and the discussion in Sect. 4.4.1, we define fully black-box reduction from key-agreement protocols to one-way functions as follows.

Definition 4.38. (*Gully 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, if the following holds for every $n \in \mathbb{N}$.

1. (A, B) is $(1 - \alpha(n))$-consistent with respect to $\mathcal{F}_{\mathrm{AF}n}$ according to Definition 4.1.
2. For every function f over $\{0,1\}^n$, algorithm D and $\delta > 0$ such that $\Pr_{v \leftarrow \langle (\mathsf{A}^f, \mathsf{B}^f)(1^n) \rangle} \left[\mathsf{D}(\mathsf{trans}(v)) = \mathrm{out}^P(v) \right] \geq \gamma + \delta$ for some $P \in \{\mathsf{A}, \mathsf{B}\}$, algorithm $\mathsf{R}^{\mathsf{D},f}$ inverts f with probability at least $p(\delta)$, for some universal $p \in \mathrm{poly}$.

Combining Theorem 4.4 and Facts 4.36 and 4.37 yields the following result.

Corollary 4.39. *There exists no fully black-box reduction from an (α, γ)-key-agreement protocol to one-way functions, with $1 - \alpha(n) - \gamma(n) > 1/\mathrm{poly}(n)$.*[21]

Proof's sketch. Assume that 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 $1-\alpha(n)-\gamma(n) > 1/\mathrm{poly}(n)$. Since $\mathcal{F}_{\mathrm{AF}n}$ is simple (Fact 4.36), by Theorem 4.4 and a simple averaging argument,[22] there exists a $\mathrm{poly}(n)$-query algorithm D such that

[20] We choose to focus on this simpler form of black-box reductions as it simplifies the proofs given in Sects. 4.4.2 and 4.4.3 and still seems to capture the same set of known reductions captured by the standard notion of black-box reductions.

[21] As previously mentioned, the same result, proven using different means, appears in [1].

[22] A similar counting argument was used in [2].

$$\Pr_{f\leftarrow\mathcal{F}_{\mathrm{AF}n}}\left[\Pr_{v\leftarrow\langle(\mathsf{A}^f,\mathsf{B}^f)(1^n)\rangle}\left[\mathsf{D}^f(\mathsf{trans}(v))=\mathrm{out}^P(v)\right]\geq\gamma(n)+1/\operatorname{poly}(n)\right]\geq 1/\operatorname{poly}(n) \tag{29}$$

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 4.37. □

4.4.3. *Differentially Private Two-Party Computation*

Following Definitions 4.6 and 4.7 and the discussion in Sect. 4.4.1, we define fully black-box reduction from differentially private protocols to one-way functions as follows.

Definition 4.40. (*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 functionality g to one-way functions, if the following holds for every $n\in\mathbb{N}$.

1. (A,B) is a $(\beta(n),d(n))$-approximation for g with respect to $\mathcal{F}_{\mathrm{AF}n}$ according to Definition 4.7.
2. For every function f over $\{0,1\}^n$, algorithm D and $\delta>0$ such that

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

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 $p(\delta)$, for some universal $p\in\operatorname{poly}$.

Combining Theorem 4.17 and Facts 4.37 and 4.36 yields the following result.

Corollary 4.41. *For constants $\nu\in(0,1)$ and $\eta\geq 0$, there exist $\lambda>0$ such that the following holds: let $q\in\operatorname{poly}$ and let s,β,d,α and γ be such that $\alpha(n)\leq\eta$, $\gamma(n)\leq\frac{\nu}{48\cdot s(n)}-\frac{1}{q(n)}$, $\beta(n)\leq\frac{1-\nu}{2}$ and $d(n)\leq\lambda\cdot\nu\cdot\frac{\sqrt{s(n)}}{\log s(n)}$ for infinitely many n's. Then, there exists* no *fully black-box reduction from an $(s,\beta,d,\alpha,\gamma)$-differentially private protocol for computing the inner-product functionality, to one-way functions.*

5. Appendix: Proving Lemma 3.10

Definition 5.1. (*Restating Definition 3.9*). 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 $j\in[m]$.

Let $\mathsf{CF}=\mathsf{CF}(\mathcal{F},\pi,\mathsf{Finder})$ be the following random process:

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. $\Pr[\#\text{ of } f - \text{calls made in } \mathsf{CF} > T] \leq \varepsilon$.

Lemma 5.2. (Restating Lemma 3.10). *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 $(64/\delta^2, \ell\delta)$-*$\mathsf{DependencyFinder}$ *for any $0 < \delta \leq \frac{1}{4\ell}$.*

We prove Lemma 3.10 by showing that a simple family $\mathcal{F}$ has an "intersection finder": there exists an algorithm $\mathsf{IntFinder}$ such that

$$\Pr_{d \leftarrow \mathsf{CF}(\mathcal{F},\pi,\mathsf{IntFinder}), v \leftarrow \mathcal{VIEW}^{\mathcal{F},\pi}(d)}[\mathrm{Intersect}_{\mathcal{I}}(v)] \leq \varepsilon.$$

We then use the above and the fact that $\mathcal{F}$ is simple, to prove the lemma.

We start by proving the case of normal-form protocols whose parties make *at most* one query per round.[23] In Sect. 5.1, 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 5.3. (*Normal-form protocols*) An oracle-aided protocol π is in $\mathsf{normal\text{-}form}$ 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.

For defining our dependency finder for such protocols, we use the following definition.

Definition 5.4. Let $\mathcal{F}$ be a function family and let π be an oracle-aided protocol. For protocol transcript $\bar{t}$ and set of query/answer pairs $\mathcal{I}$, let $\mathrm{NoInt}^{\mathcal{F},\pi}(\bar{t}, \mathcal{I}) = \{\omega \in \Omega^{\mathcal{F},\pi}(\bar{t}, \mathcal{I}) \colon \mathrm{Intersect}_{\mathcal{I}}(\mathsf{view}(\omega)_{|\bar{t}|}) = 0\}$ and let $\mathcal{VIEW}^{\mathcal{F},\pi}_{\mathrm{NoInt}}(y = (\bar{t}, \mathcal{I})) = \mathcal{VIEW}_{\mathrm{NoInt}^{\mathcal{F},\pi}(y)}(y)$.

Namely, $\mathcal{VIEW}^{\mathcal{F},\pi}_{\mathrm{NoInt}}(y)$ is a random joint view v of π that is consistent with $(\bar{t}, \mathcal{I})$ and has $\mathrm{Intersect}_{\mathcal{I}}(v) = 0$. The following algorithm is defined with respect to a function family $\mathcal{F}$, protocol π and $\delta > 0$.

Algorithm 5.5. ($\mathsf{IntFinder}_{\mathcal{F},\pi,\delta}$)

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

[23] Called "semi-normal form" in [1].

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 by* B *in a random sample from* $\mathcal{VIEW}^{\mathcal{F},\pi}_{\text{NoInt}}(\bar{t},\mathcal{I})$*:*

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

Namely, algorithm IntFinder considers executions of π in which the views of the parties have no unexposed intersection query (i.e. a joint query that does not appear in the query list) and are consistent with the given partial transcript and set of query/answer pairs.

The goal of IntFinder 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 3.10 (for the case of normal-form protocols) by showing that IntFinder is a $(64/\delta^2, \ell\delta)$-DependencyFinder for $(\mathcal{F},\pi)$. The heart of the proof is in the following lemma, yielding that in a random execution of $\mathsf{CF} = \mathsf{CF}(\mathcal{F},\pi,\mathsf{IntFinder})$, the probability that the views of the parties make an unexposed intersection query, in any given round, is small.

Lemma 5.6. *Let $\mathcal{F}$ be a finite function family and let π be an ℓ-query, normal-form protocol. Let $\bar{t}$ be a (possibly partial) transcript of π, let $\mathcal{I}$ be a set of query/answer pairs, let* $\text{NoInt} = \text{NoInt}^{\mathcal{F},\pi}$*, and let $\mathcal{D} = \mathcal{VIEW}^{\mathcal{F},\pi}_{\text{NoInt}}(\bar{t},\mathcal{I})$. Assume there exists $\delta > 0$ such that $\Pr_{v\leftarrow\mathcal{D}}[q \in v \wedge (q,\cdot) \notin \mathcal{I}] \le \delta \le \frac{1}{4\ell}$ for every query q, then the following hold:*

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

We prove Lemma 5.6 in Sect. 5.2, but first use it for proving Lemma 3.10. We start by showing that algorithm IntFinder defined above is a good DependencyFinder for $(\mathcal{F},\pi)$ (namely, we are proving Lemma 3.10 for normal-form protocols).

Claim 5.7. *Let $(\mathcal{F},\pi)$ be a pair of simple function family and ℓ-query normal-from protocol, and let $0 < \delta \le \frac{1}{4\ell}$. Then, algorithm* $\mathsf{IntFinder} = \mathsf{IntFinder}_{\mathcal{F},\pi,\delta}$ *is a* $(64/\delta^2, \ell\delta)$-DependencyFinder *for* $(\mathcal{F},\pi)$.

Proof. The following random variables are defined with respect to a random execution of $\mathsf{CF} = \mathsf{CF}(\mathcal{F},\pi,\mathsf{IntFinder})$. Let $W = (r_{\mathsf{A}}, r_{\mathsf{B}}, f) \in \Omega$ be the triplet chosen in the first step of CF. Let $V = \mathsf{view}(W)_j$ (i.e. the j'th long prefix of the full view implied by W) and 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 $2 \le i \le j$ let $\text{FirstInt}_i = \text{Intersect}_{I_{i-1}}(V_i) \wedge \neg\,\text{Intersect}_{I_{i-2}}(V_{i-1})$ (where $I_0 = \emptyset$). We start by showing that IntFinder is a good dependencies remover. We do so by bounding the probability of $\text{Intersect}_{I_j}(V)$. Since $\mathcal{F}$ is a simple family, this yields the first property required for being a $(\cdot,\ell\delta)$-DependencyFinder for $(\mathcal{F},\pi)$. We later complete the proof by bounding the number of queries made in CF.

IntFinder is a Good Dependencies Remover. The following claim bounds the probability that a single "round" of CF causes an intersection.

Claim 5.8. $\Pr[\text{FirstInt}_i] \leq \frac{\delta}{8} \cdot E[\ell_i(V)]$ *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 $I_{i-1} = I_{i-2} \cup \mathsf{IntFinder}^f(\overline{T}_{1,\ldots,i-1}, I_{i-2})$. The definition of IntFinder yields that

$$\Pr[q \in V_{i-1} \wedge (q,\cdot) \notin I_{i-1} \mid \neg \text{Intersect}_{I_{i-1}}(V_{i-1})] \leq \delta/32 \tag{30}$$

for any query q. We can hence apply Lemma 5.6(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

$$\begin{aligned} &\Pr\left[\text{Intersect}_{I_{i-1}}(V_i) \mid \neg \text{Intersect}_{I_{i-1}}(V_{i-1})\right] \\ &\quad \leq \frac{\delta}{8} \cdot \Pr\left[\ell_i(V) = 1 \mid \neg \text{Intersect}_{I_{i-1}}(V_{i-1})\right] \end{aligned} \tag{31}$$

We conclude that

$$\begin{aligned} &\Pr[\text{FirstInt}_i] \\ &= \Pr[\text{Intersect}_{I_{i-1}}(V_i) \wedge \neg \text{Intersect}_{I_{i-2}}(V_{i-1})] \\ &\leq \Pr[\text{Intersect}_{I_{i-1}}(V_i) \wedge \neg \text{Intersect}_{I_{i-1}}(V_{i-1})] \\ &= \Pr[\neg \text{Intersect}_{I_{i-1}}(V_{i-1})] \cdot \Pr[\text{Intersect}_{I_{i-1}}(V_i) \mid \neg \text{Intersect}_{I_{i-1}}(V_{i-1})] \\ &\leq \Pr[\neg \text{Intersect}_{I_{i-1}}(V_{i-1})] \cdot \frac{\delta}{8} \cdot \Pr\left[\ell_i(V) = 1 \mid \neg \text{Intersect}_{I_{i-1}}(V_{i-1})\right] \\ &\leq \frac{\delta}{8} \cdot \mathrm{E}[\ell_i(V)]. \end{aligned} \tag{32}$$

The first inequality holds since extending $\mathcal{I}$ never increases the probability of intersection. Since Eq. (32) 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 with the proof (of Claim 5.7), Claim 5.8 yields that

$$\begin{aligned} \Pr\left[\text{Intersect}_{I_j}(V)\right] &\leq \Pr\left[\bigvee_{2 \leq i \leq j} \text{FirstInt}_i\right] \leq \sum_{2 \leq i \leq j} \frac{\delta}{8} \cdot \mathrm{E}\left[\ell_i(V)\right] \\ &= \frac{\delta}{8} \mathrm{E}\left[\sum_{2 \leq i \leq j} \ell_i(V)\right] \leq \frac{\ell\delta}{8}. \end{aligned} \tag{33}$$

The first inequality holds since the first round cannot have an intersection query (and since extending $\mathcal{I}$ never increases the probability of intersection). The third inequality holds since ℓ bounds the overall number of queries made in any execution of π.

Since for any $d = (\overline{t}, \mathcal{I}) \in \mathrm{Supp}(\overline{T}, I_j)$ it holds that $\mathrm{SD}\left(\mathcal{VIEW}(d), \mathcal{VIEW}_{\mathrm{NoInt}}(d)\right) = \Pr_{\omega \leftarrow \Omega(d)}[w \notin \mathrm{NoInt}(d)] = \Pr_{v \leftarrow \mathcal{VIEW}(d)}[\mathrm{Intersect}_{\mathcal{I}}(v)]$, it follows that

$$\begin{aligned}
&\mathrm{E}_{d \leftarrow (\overline{T}, I_j)}\left[\mathrm{SD}\left(\mathcal{VIEW}(d), \mathcal{VIEW}_{\mathrm{NoInt}}(d)\right)\right] \\
&= \mathrm{E}_{d \leftarrow (\overline{T}, I_j)}\left[\Pr_{v \leftarrow \mathcal{VIEW}(d)}[\mathrm{Intersect}_{\mathcal{I}}(v)]\right] \\
&= \Pr[\mathrm{Intersect}_{I_j}(V)] \\
&\leq \frac{\ell\delta}{8},
\end{aligned} \tag{34}$$

where the inequality hold by Eq. (33). We complete the proof of this part by showing that $\mathcal{VIEW}_{\mathrm{NoInt}}(d)$ is close to some product distribution over the views of the parties. The definition of IntFinder yields that

$$\Pr[q \in V \wedge (q, \cdot) \notin I_j \mid \neg\, \mathrm{Intersect}_{I_j}(V), (\overline{T}, I_j) = d] \leq \frac{\delta}{32} \tag{35}$$

for any possible query q and $d \in \mathrm{Supp}(\overline{T}, I_j)$. Therefore, Lemma 5.6(1) yields that

$$\mathrm{SD}\left(\mathcal{VIEW}_{\mathrm{NoInt}}(d), \mathcal{C}(d)\right) \leq \frac{\ell\delta}{16}, \tag{36}$$

for any $d \in \mathrm{Supp}(\overline{T}, I_j)$, where $\mathcal{C}(d)$ is a *product* distribution over the views of the parties. It follows (using the triangle inequality) that $\mathrm{SD}\left(\mathcal{VIEW}_{\mathrm{NoInt}}(d), \left(\mathcal{VIEW}_{\mathrm{NoInt}}(d)_{\mathsf{A}}, \mathcal{VIEW}_{\mathrm{NoInt}}(d)_{\mathsf{B}}\right)\right) \leq \frac{3}{16} \cdot \ell\delta$ for any $d \in \mathrm{Supp}(\overline{T}, I_j)$, and therefore

$$\mathrm{E}_{d \leftarrow (\overline{T}, I_j)}\left[\mathrm{SD}\left(\mathcal{VIEW}_{\mathrm{NoInt}}(d), \left(\mathcal{VIEW}_{\mathrm{NoInt}}(d)_{\mathsf{A}}, \mathcal{VIEW}_{\mathrm{NoInt}}(d)_{\mathsf{B}}\right)\right)\right] \leq \frac{3}{16} \cdot \ell\delta. \tag{37}$$

Combining Eqs. (34) and (37) and the triangle inequality yields that

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

Bounding the query complexity of IntFinder. We complete the proof of Claim 5.7 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 5.9. *For every query q, it holds that*

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

Namely, Claim 5.9 relates the probability that a query is asked by IntFinder, to the probability that this query is asked by one of the parties in V.

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

$$\Pr[\hat{q} \in V_i \mid W \in \Omega(\mathcal{I}^*, \bar{t}^*), \neg \text{Intersect}_{\mathcal{I}^*}(V_i)] \geq \delta/32 \tag{38}$$

Applying a simple Bayes' rule, it follows that

$$\begin{aligned}
\Pr[\hat{q} \in V_i \mid W \in (\mathcal{I}^*, \bar{t}^*)] &\geq \Pr[\hat{q} \in V_i \wedge \neg \text{Intersect}_{\mathcal{I}^*}(V_i) \mid W \in \Omega(\mathcal{I}^*, \bar{t}^*)] \\
&= \Pr[\neg \text{Intersect}_{\mathcal{I}^*}(V_i) \mid W \in (\mathcal{I}^*, \bar{t}^*)] \cdot \Pr[\hat{q} \\
&\quad \in V_i \mid W \in \Omega(\mathcal{I}^*, \bar{t}^*), \neg \text{Intersect}_{\mathcal{I}^*}(V_i)] \\
&\geq \Pr[\neg \text{Intersect}_{\mathcal{I}^*}(V_i) \mid W \in \Omega(\mathcal{I}^*, \bar{t}^*)] \cdot \frac{\delta}{32}. \qquad (39)
\end{aligned}$$

For $i \in [j]$, let $\mathcal{S}_i(q)$ be the set of $(\mathcal{I}^*, \bar{t}^*)$ pairs that cause IntFinder to ask the query q in the i'th round of CF. That is: if $W \in \Omega(\mathcal{I}^*, \bar{t}^*)$, then IntFinder 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

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

The first inequality holds since a query is asked at most once in CF (and hence we are summing over disjoint events) and the second one by Eq. (39).

Let $\widetilde{\text{CF}}$ be the variant of CF that aborts in case $\text{Intersect}_{I_{i-1}}(V_i) = 1$ for some $i \in [j]$ (i.e. $\widetilde{\text{CF}}$ aborts right after computing I_{i-1}). Let $\widetilde{Q}_i$ be the respective analogues of Q_i defined with respect to a random execution of $\widetilde{\text{CF}}$, and let $\widetilde{Q} = \bigcup_{i\in[j]} \widetilde{Q}_i$ (i.e. $\widetilde{Q}$ denote all queries asked by IntFinder in $\widetilde{\text{CF}}$). The same calculation done in Eq. (33) yields that

$$SD(Q, \widetilde{Q}) \leq \ell\delta/8 \tag{40}$$

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

A simple argument yields that $\Pr[q \in \widetilde{Q}_i] \leq \Pr[q \in Q_i \wedge \neg \text{Intersect}_{I_{i-1}}(V_i)]$, for every $i \in [j]$ and every query q. Thus, Claim 5.9 yields that

$$\Pr[q \in \widetilde{Q}] = \sum_{i \in [j]} \Pr[q \in \widetilde{Q}_i] \leq \frac{32}{\delta} \cdot \Pr[q \in V] \tag{41}$$

for every query q. It follows that

$$\mathrm{E}\left[\left|\widetilde{Q}\right|\right] = \mathrm{E}\left[\sum_q \chi_{\widetilde{Q}}(q)\right] \leq \frac{32}{\delta} \cdot \mathrm{E}\left[\sum_q \chi_V(q)\right] \leq 32\ell/\delta \tag{42}$$

(where $\chi_x(q) = 1$ if $q \in x$ and $\chi_x(q) = 0$ otherwise). The first inequality holds by Eq. (41) and linearity of expectation and the last one since at most ℓ queries are asked in V.

A Markov argument yields that $\Pr\left[\left|\widetilde{Q}\right| > 64/\delta^2\right] \leq \ell\delta/2$. Hence, Eq. (40) yields that $\Pr\left[|Q| > 64/\delta^2\right] \leq \ell\delta/2 + \ell\delta/8 < \ell\delta$.

5.1. *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 5.10. (*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.[24]

Lemma 5.11. *Let $\mathcal{F}$ be a function family, let π_{G} be an oracle-aided protocol and let π_{N} be its normal-form variant. Assume $(\mathcal{F}, \pi_{\mathrm{N}})$ has a (T, ε)-*DependencyFinder*, then $(\mathcal{F}, \pi_{\mathrm{G}})$ has a (T, ε)-*DependencyFinder.

The straightforward proof of Lemma 5.11 is given below, but first let us use it for concluding the proof of Lemma 3.10.

[24] 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.

Proof of Lemma 3.10. Let $\mathcal{F}$ be a simple function family, let $\pi_G = (\mathsf{A}, \mathsf{B})$ 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 5.3, Claim 5.7 yields that $(\mathcal{F}, \pi_N)$ has a $(64/\delta^2, \ell\delta)$-DependencyFinder for any $\delta \leq 1/4\ell$. Hence, Lemma 5.11 yields that the same holds for $(\mathcal{F}, \pi_G)$. □

Proof of Lemma 5.11. Let $\mathcal{F}$, π_G and π_N be as in the statement of the lemma, and let Finder_N be a (T, ε)-DependencyFinder for $(\mathcal{F}, \pi_N)$. We define the DependencyFinder for $(\mathcal{F}, \pi_G)$ as follows: □

Algorithm 5.12. (Finder_G)

Input: a transcript $\bar{t}$ of π_G 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 Algorithm 5.12). This observation immediately yields the required bound on the number of oracle queries made in CF, since these outputs determine the number of queries made to the oracle. To prove that the first property required by Definition 3.9 also holds (see the equation below), we also note that a random sample of $\mathcal{VIEW}^{\mathcal{F},\pi_N}(d_N)$, for $d_N \in \mathrm{Supp}(\mathsf{CF}_N)$, can be sampled by applying a (deterministic) *injective* function to $\mathcal{VIEW}^{\mathcal{F},\pi_G}(\mathsf{M}^{-1}(d_N))$. It follows that

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

concluding the proof of the lemma. □

5.2. *Proving Lemma 5.6*

The following discussion is with respect to fixed values of $\bar{t}$ and $\mathcal{I}$, where NoInt, $\mathcal{D}$ and $\mathcal{D}^+$ are defined with respect to these values as in the statement of 5.6. Towards proving 5.6, we make the following observations: in Claim 5.13, we show that under the no intersection condition, $\mathcal{D}$ is distributed as some *product distribution*. In Claim 5.15, we

use the first observation to express $\mathcal{D}$ as a uniform sampled edge of a *dense* bipartite graph.[25]

5.2.1. *Product Characterization*

Claim 5.13. *There exist two distributions $\mathcal{A}$ and $\mathcal{B}$ with $\mathcal{D} = (\mathcal{A} \times \mathcal{B}) \mid \neg \text{Intersect}_{\mathcal{I}}$.*

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 $v = (v_{\mathsf{A}}, v_{\mathsf{B}}) \in \text{Supp}(\mathcal{D})$, 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 $\text{Supp}(\mathcal{D}_{\mathsf{A}})$ with $\mathcal{A}(v_{\mathsf{A}}) = c_{\mathcal{A}} \cdot \gamma_{\mathsf{A}}(v_{\mathsf{A}})$, and $\mathcal{B}$ be the distribution over $\text{Supp}(\mathcal{D}_{\mathsf{B}})$ 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 3.3 yields that

$$\mathcal{D}(v) = \frac{\Pr_{\Omega}[r_{\mathsf{A}}, r_{\mathsf{B}}] \cdot \alpha^{\mathcal{I}}_{v_{\mathsf{A}}|v_{\mathsf{B}}} \cdot \alpha^{\mathcal{I}}_{v_{\mathsf{B}}|v_{\mathsf{A}}}}{\Pr_{\Omega|\mathcal{I}}[\bar{t}, \text{NoInt}]}, \tag{43}$$

for every $v = (r_{\mathsf{A}}, r_{\mathsf{B}}, \cdot) \in \text{Supp}(\mathcal{D})$. Since the random coins of the parties are chosen independently, it holds that $\Pr_{\Omega}[r_{\mathsf{A}}, r_{\mathsf{B}}] = \Pr_{\Omega}[r_{\mathsf{A}}] \cdot \Pr_{\Omega}[r_{\mathsf{B}}]$. Since $\mathcal{F}$ is a finite family, it holds 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 $v \in \text{Supp}(\mathcal{D})$. 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.

5.2.2. *Graph Characterization*

Let $\mathcal{A}$ and $\mathcal{B}$ be the distributions guaranteed by Claim 5.13. This product characterization allows us to think of $\mathcal{D}$ as the uniform distribution over the edges the following bipartite graph $G = (V_{\mathcal{A}}, V_{\mathcal{B}}; E)$.

Definition 5.14. (*The graph G*) A node $a \in V_{\mathcal{A}}$ corresponds to a view $\mathsf{view}_{\mathsf{A}}(a)$ of A in the support of $\mathcal{A}$, and the number of nodes corresponding to a view v_{A} is proportional to $\mathcal{A}(v_{\mathsf{A}})$.[26] A node $b \in V_{\mathcal{B}}$ corresponds to a view $\mathsf{view}_{\mathsf{B}}(b)$ of B in an analogues manner. Let $E = \{(a, b) \in (V_{\mathcal{A}} \times V_{\mathcal{B}}) \colon \text{Intersect}_{\mathcal{I}}(\mathsf{view}_{\mathsf{A}}(a), \mathsf{view}_{\mathsf{B}}(b)) = 0\}$.

Hence, $\mathcal{A}$ and $\mathcal{B}$ correspond to the uniform distribution over $V_{\mathcal{A}}$ and $V_{\mathcal{B}}$, respectively, and Claim 5.13 yields that $\mathcal{D}$ is the distribution of $(\mathsf{view}_{\mathcal{A}}(a), \mathsf{view}_{\mathcal{B}}(b))$ for $(a, b) \leftarrow E$. We show that $\mathcal{D}$ is close to being a product distribution by showing that G is dense. Specifically, we show that every vertex in G is connected to most of the vertices on the other side. For $x \in V_{\mathcal{A}} \cup V_{\mathcal{B}}$, let $d(x)$ denote the degree of x in the graph G.

Claim 5.15. *It holds that $d(a) \geq |V_{\mathcal{B}}| \cdot (1 - 2\ell\delta)$, and $d(b) \geq |V_{\mathcal{A}}| \cdot (1 - 2\ell\delta)$ for every $b \in V_{\mathcal{B}}$*

[25] Observations of similar spirit were done in [1], and parts of the following text are taken verbatim from there.

[26] Since $\mathcal{F}$ is finite, all probabilities in consideration are *rational*, and therefore, the described graph is well defined.

Proof. Since, by assumption, $\Pr_{v \leftarrow \mathcal{D}}[q \in v \wedge (q, \cdot) \notin \mathcal{I}] \leq \delta$ for every query q and since ℓ bounds the query complexity of π, a union bound yields that

$$\Pr_{v_{\mathsf{A}} \leftarrow \mathcal{D}_{\mathsf{A}}}[\text{Intersect}_{\mathcal{I}}(v_{\mathsf{A}}, v_{\mathsf{B}})] \leq \ell\delta \tag{44}$$

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 $\tilde{E}(a) = \{b \in V_{\mathcal{B}} : (a, b) \notin E\}$. We next show that $\sum_{b \in \tilde{E}(a)} d(b) \leq \ell\delta \cdot |E|$ 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 $\sum_{b \in \tilde{E}(a)} \frac{d(b)}{|E|} > \ell\delta$, then $\Pr_{v_{\mathsf{B}} \leftarrow \mathcal{D}_{\mathsf{B}}}\left[\text{Intersect}_{\mathcal{I}}(\mathsf{view}_{\mathsf{A}}(a), v_{\mathsf{B}})\right] > \ell\delta$, contradicting equation (44). An analogous argument shows that $\sum_{a \in \tilde{E}(b)} d(a) \leq \ell\delta \cdot |E|$ for every $b \in V_{\mathcal{B}}$. Clearly, the degree of each vertex is at least 1 and $\ell\delta \leq 1/4$, and hence, the following fact concludes the proof:

Fact 5.16. *[1, claim4.6] Let $G = (V_{\mathcal{A}}, V_{\mathcal{B}}; E)$ be a nonempty bipartite graph. Assume there exists $\gamma \leq 1/2$ such that $|\tilde{E}(v)| \leq \gamma|E|$ 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 5.6.

Proof of Lemma 5.6. The first part of the lemma immediately follows from Claim 5.15. We prove the second part of the lemma by showing that

$$\Pr_{v \leftarrow \mathcal{D}^+ | v_{\mathsf{B}} = v_{\mathsf{B}}^+}\left[\text{Intersect}_{\mathcal{I}}(v_{\mathsf{A}}, v_{\mathsf{B}}^+)\right] \leq 4\delta \cdot \ell_{|\bar{t}|+1}(v_{\mathsf{B}}^+) \tag{45}$$

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 Eq. (45) 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 $\mathsf{view}_{\mathsf{B}}(b) = v_{\mathsf{B}'}$, let $N(b)$ be b's neighbours in G and let $\mathcal{S} = \left\{a \in N(b) \colon q' \in \mathsf{view}_{\mathsf{A}}(a)\right\}$. Since $\Pr_{v \leftarrow \mathcal{D}}[q \in v \wedge (q, \cdot) \notin \mathcal{I}] \leq \delta$, we have that

$$\frac{\sum_{a \in \mathcal{S}} d(a)}{|E|} \leq \delta \tag{46}$$

and therefore

$$\begin{aligned}\Pr_{a \leftarrow N(b)}[a \in \mathcal{S}] = \frac{|\mathcal{S}|}{d(b)} &\leq \frac{|\mathcal{S}|}{(1 - 2\ell\delta)|V_{\mathcal{A}}|} \leq \frac{|\mathcal{S}||V_{\mathcal{B}}|}{(1 - 2\ell\delta)|E|} \\ &\leq \frac{\sum_{a \in \mathcal{S}} d(a)}{(1 - 2\ell\delta)^2|E|} \leq \frac{\delta}{(1 - 2\ell\delta)^2} \leq 4\delta\end{aligned} \tag{47}$$

The first and third inequalities hold by Claim 5.15, the second since $|E| \leq |V_{\mathcal{A}}||V_{\mathcal{B}}|$, and the fourth by Eq. (46). Since Eq. (47) holds for every $b \in V_{\mathsf{B}}$ with $\mathsf{view}_{\mathsf{B}}(b) = v_{\mathsf{B}'}$, it follows that

$$\Pr_{v \leftarrow \mathcal{D}^{+} | v_{\mathsf{B}} = v_{\mathsf{B}}^{+}} \left[q' \in v_{\mathsf{A}} \right] \leq 4\delta. \tag{48}$$

It follows that $\Pr_{v \leftarrow \mathcal{D}^{+} | v_{\mathsf{B}} = v_{\mathsf{B}}^{+}} \left[\mathrm{Intersect}_{\mathcal{I}}(v_{\mathsf{A}}, v_{\mathsf{B}}^{+}) \right] \leq 4\delta$, concluding the proof of Eq. (45), and thus the proof of the lemma. □

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] B. Barak and M. Mahmoody-Ghidary. Lower bounds on signatures from symmetric primitives. In *Proceedings of the 48th Annual Symposium on Foundations of Computer Science (FOCS)*, 2007.

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

[4] 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.

[5] 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.

[6] B. Chor and E. Kushilevitz. A zero-one law for boolean privacy. *SIAM J. Discrete Math.,* 4(1):36–47, 1991.

[7] 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.

[8] D. Dachman-Soled, M. Mahmoody, and T. Malkin. Can optimally-fair coin tossing be based on one-way functions? In *TCC*, pages 217–239, 2014.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] A. Groce, J. Katz, and A. Yerukhimovich. Limits of computational differential privacy in the client/server setting. In *Theory of Cryptography, Eighth Theory of Cryptography Conference, TCC 2011*, pages 417–431, 2011.

[16] 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.

[17] I. Haitner, E. Omri, and H. Zarosim. Limits on the usefulness of random oracles. In *Theory of Cryptography, Tenth Theory of Cryptography Conference, TCC 2013*, pages 437–456, 2013.

[18] 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.

[19] 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.

[20] 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.

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

[22] 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.

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

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

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

[26] 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.

[27] 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.

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

[29] 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.

[30] 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.