On the Round Complexity of OT Extension

Sanjam Garg[1(✉)], Mohammad Mahmoody[2], Daniel Masny[1], and Izaak Meckler[1]

[1] University of California, Berkeley, Berkeley, USA
sanjamg@berkeley.edu
[2] University of Virginia, Charlottesville, USA

Abstract. We show that any OT extension protocol based on one-way functions (or more generally any symmetric-key primitive) either requires an additional round compared to the base OTs or must make a non-black-box use of one-way functions. This result also holds in the semi-honest setting or in the case of certain setup models such as the common random string model. This implies that OT extension in any secure computation protocol must come at the price of an additional round of communication or the non-black-box use of symmetric key primitives. Moreover, we observe that our result is tight in the sense that positive results can indeed be obtained using non-black-box techniques or at the cost of one additional round of communication.

1 Introduction

Multiparty secure computation (MPC) [Yao82,GMW87] allows mutually distrustful parties to compute a joint function on their inputs, from which the parties learn their corresponding outputs but nothing more. Oblivious transfer (OT) [Rab81,EGL85,BCR87,Kil88,IPS08] is the fundamental building block for two and multiparty secure computation.

An OT protocol is a two-party protocol between a sender with inputs x_0, x_1 and a receiver with input bit b. An OT protocol allows the receiver to only learn x_b while b remains hidden from the sender. OT is a very powerful tool and is

S. Garg—Research supported in part from DARPA/ARL SAFEWARE Award W911NF15C0210, AFOSR Award FA9550-15-1-0274, AFOSR YIP Award, DARPA and SPAWAR under contract N66001-15-C-4065, a Hellman Award and research grants by the Okawa Foundation, Visa Inc., and Center for Long-Term Cybersecurity (CLTC, UC Berkeley). The views expressed are those of the author and do not reflect the official policy or position of the funding agencies.

M. Mahmoody—Supported by NSF CAREER award CCF-1350939, a subcontract on AFOSR Award FA9550-15-1-0274, and University of Virginia's SEAS Research Innovation Award.

D. Masny—Supported by the Center for Long-Term Cybersecurity (CLTC, UC Berkeley).

H. Shacham and A. Boldyreva (Eds.): CRYPTO 2018, LNCS 10993, pp. 545–574, 2018.
https://doi.org/10.1007/978-3-319-96878-0_19

sufficient to realize any secure computation functionality [Kil88, IPS08]. Nevertheless, all known constructions of OT have the drawback of being significantly less efficient than "symmetric-key primitives" like block ciphers or hash functions. This comparatively low efficiency seems to be unavoidable as black-box constructions of OT from one-way functions are known to be impossible [IR89]. Overcoming this difficulty, one promising approach is to use OT *extension*. OT extension allows a sender and a receiver to extend a relatively small number of base OTs to a much larger number of OTs using only symmetric-key primitives (e.g., one-way functions, pseudorandom generators, collision-resistant hash functions, etc.), which are indeed much cheaper.

Beaver first proposed the idea of such an OT extension protocol [Bea96]. Beaver's protocol solely relied on a security parameter number of base OTs and, perhaps surprisingly, only on a pseudorandom generator (PRG). This insight – that a small number of inefficient base OTs could be efficiently extended to a large number of OTs – has been a crucial step in overcoming the efficiency limitation of OT in particular and multiparty computation in general. Beaver's construction, however, made an expensive *non-black-box* use of the underlying PRG leading to inefficient protocols.

In an influential work, Ishai, Kilian, Nissim and Pentrank [IKNP03] obtained an OT extension (referred to as IKNP) which made only *black-box* use of the underlying cryptographic primitive, which could be realized using a random oracle. This yielded a significantly more efficient protocol in comparison to Beaver's protocol. They also observed that the random oracle in their construction can be relaxed to the notion of a correlation robust hash function. Follow up works on OT extension achieve security against stronger adversaries [NNOB12, ALSZ15] or reduce communication and computation costs [KK13].

The practical impact of the OT extension protocols has been enormous. OT extension can be used to improve the computational efficiency of virtually any implementation of secure MPC. In particular, the standard recipe for realizing efficient secure computation protocols is as follows. We start with the OT-hybrid model where everyone has access to an ideal OT functionality called OT-hybrid. Then instantiate an OT extension using the OT-hybrid, which implies that only black-box access to the OTs is used. An efficient secure computation protocol is then realized using OT extension to minimize the number of public-key operations. Use of OT extension yields remarkable efficiency gains for many implemented protocols (e.g. see [ALSZ13]).

In addition to the computational efficiency, round complexity is another parameter of concern in the construction of efficient secure computation protocols. Significant research effort has been made toward realizing round efficient OT [NP01, AIR01, HK12, PVW08] and round efficient two-party [KO04, ORS15] and multiparty [BMR90, AJL+12, GGHR14, MW16, GMPP16, GS17a, BL17, GS17b] secure computation protocols. Several of these protocols are also black-box in the use of the underlying cryptographic primitives. However, all these works only yield protocols (with a given round complexity) using a large number public-key operations. Ideally, we would like to construct OT extension protocols that can be used to reduce the number of public-key operations

needed in these protocols while preserving the round complexity and the black-box nature of the underlying protocol. This brings us to our following main question:

Can we realize a round-preserving OT extension protocol which makes only black-box use of "symmetric-key" cryptographic primitives?

The random oracle model (ROM) accurately captures the black-box use of such symmetric-key primitives, as it directly provides us with ideally strong hash functions as well as block-ciphers or even ideal ciphers [CPS08,HKT11]. Therefore, in order to answer the above question, we study the possibility of OT extension protocols in the ROM that preserve the round complexity.[1]

1.1 Our Results

We provide a negative answer to the above main question. In other words, we show that any OT extension protocol based on so called symmetric-key primitives, requires either an additional round compared to the base OTs or must make a non-black-box use of symmetric-key primitives. We capture black-box use of one-way functions, or even correlation-robust hash functions, as well as common random string setup[2] by proving our impossibility result under the idealized notion of these primitives which is provided by a random oracle. In particular, we prove the following theorem.

Theorem 1 (Impossibility of round-preserving OT extension in ROM–*Informally Stated*). *Suppose a sender $\mathcal{S}$ and a receiver $\mathcal{R}$ want to perform m OTs in r rounds using a random oracle, and they both have access to n, r-round OTs (i.e. the receiver obtains its outputs at the end of round r) where $n < m$. Then, if $\mathcal{S}$ and $\mathcal{R}$ can ask polynomially many more queries to the random oracle, one of them could always break security of the m OTs.*

Theorem 1 holds even for an extension from n *string* OTs to $m = n + 1$ *bit* OTs, and even for the setting of semi-honest security. It also gives an alternative, and arguably simpler, proof to Beaver's impossibility result that information-theoretically secure OT extension does not exist in the plain model [Bea96]. We sketch the main ideas in Sect. 1.2 and provide the details in Sect. 3.

Additionally, we observe that our results are tight in two different ways. First, the IKNP protocol [IKNP03] realizes black-box OT extension using one additional round. Second, our result is also tight with respect to the black-box use of the symmetric-key primitives captured by random oracles. Beaver's original protocol provided OT extension in which the receiver has no control over which input he receives (it will be chosen at random). This notion of OT is often referred to as "random" OT. The known generic way of going from "random"

[1] The only symmetric-key primitive not directly implied by a random oracle is one-way permutations. However, most negative results in the random oracle model, including our work, extend to one-way permutations using standard techniques [IR89].

[2] Note that a random oracle also provides a common random string for free.

OTs to "chosen" OTs will add another round [EGL85]. We observe (see the full version [GMMM17]) that Beaver's original non-black-box OT extension protocol [Bea96], which only relies on a PRG, can be modified to provide round-preserving "chosen-input" OTs, but this result will require *non-black-box* use of the PRG.

We remark that our results have implications in several other settings, for example, in the plain model under malicious security. In this setting, an OT protocol takes at least 4 rounds [KO04,ORS15]. Therefore, our results imply that in this setting, black-box OT extension protocols must be at least five rounds while a non-black-box construction with four rounds can be realized. Another example is the correlated setup model [FKN94,IKM+13,BGI+14] where our results imply that there is no non-interactive OT extension even in the presence of a random oracle. Interestingly, this setting behaves very differently from a setting of shared randomness, where the amount of shared randomness can be easily increased by using the random oracle as a PRG. On the contrary, in case of a single communication round, the IKNP protocol [IKNP03] can be used to increase the amount of correlated randomness in this setting.

Finally, we note that our impossibility result of Theorem 1 also holds for the case of random *permutation* oracle model. The proof of Theorem 1 directly extends to this setting using the standard trick introduced in [IR89]. Namely, the attacker can always ask all the oracle queries of input lengths at most $c \cdot \log \kappa$ for sufficiently large constant c, in which case, the probability of the honest parties, the simulator, or the attacker (of the random oracle model) itself getting a collision while accessing the random oracle on input of length $> c \log \kappa$ is sufficiently small. However, without collisions, (length preserving) random oracles and random permutation oracles are the same.

1.2 Technical Overview

In this section, we explain the key ideas behind the proof of our main impossibility result of Theorem 1. For a formal treatment, see Sect. 3. In a nutshell, we first present an entropy-based information-theoretic argument for the plain model, where there are no oracles involved beyond the hybrid OTs. We then extend our attack to the random oracle model, by making use of the 'dependency learner' of [IR89,BMG09,HOZ16,BM17], which is a algorithms that allows us to 'approximate' certain plain-model arguments also in the random oracle model. As we will see, the combination of these two steps will make crucial use of the *round-preserving* property of the (presumed) OT-extension construction.

To explain the core ideas behind our proofs, it is instructive to even define a 2-round-preserving OT extension protocol, to see how the definition accurately models the concept of round-preserving OT extension, and because we are particularly interested in ruling out black-box 2-round OT extension protocols. Below, we first describe the notation and the simplifying assumptions for this special case (of 2-round extensions), before going over the ideas behind the proof.

Notation and the simplified setting. Here we define some basic notations and also state some simplifying assumptions, some of which are without loss of

generality when we focus on 2-round-preserving OT extensions, and the rest are relaxed when proving the formal attack in Sect. 3. Here we focus on the case of extending n instances of OT, into m instances for some $m \gg n$. (This is without loss of generality as even "one-more" OT extension, i.e., $m = n+1$, can be used to get polynomially many more OTs – e.g., see [LZ13].)

Inputs and outputs: Let $[m] := \{1, \ldots, m\}$. Suppose $\vec{b} = (b_1, \ldots, b_m) \in \{0,1\}^m$ are the choice bits of the receiver $\mathcal{R}$ and $x = (x_i^0, x_i^1)_{i\in[m]} \in \{0,1\}^{2m}$ are the pairs of bits that the sender holds as its input. (Our main negative result holds even if the hybrid $\mathbb{OT}_n$ provides string OTs, but in this simplified exposition, we work with bit OTs.) The receiver $\mathcal{R}$ wishes to get output $(x_i^{b_i})_{i\in[m]}$.

The oracle and OT hybrid: The two parties have access to a random oracle $\mathbf{H}$ as well as n instances of a OT-hybrid functionality for *bit* inputs which we denote with $\mathbb{OT}_n$. When using $\mathbb{OT}_n$, $\mathcal{R}$ and $\mathcal{S}$ will *not* reverse their roles such that $\mathcal{R}$ always receives the output from $\mathbb{OT}_n$. This is without loss of generality for a round preserving OT extension for the following reason. First note that the last message of the constructed OT should be from the sender to the receiver, as otherwise it could be dropped. Moreover, we use a hybrid $\mathbb{OT}_n$ that *requires* (here) two rounds to deliver its output. Therefore, if both the used hybrid OTs and constructed OTs have the same (here two) rounds, the last messages of the hybrid and the constructed OTs should both go from the sender to the receiver. Thus, we model the 2-round-preserving OT extension in the ROM as follows.

1. $\mathcal{R}$ sends a single message t_1 to $\mathcal{S}$, and it also chooses and submits the input $\vec{c} = (c_1, \ldots, c_n)$ to the hybrid $\mathbb{OT}_n$.
2. $\mathcal{S}$ sends a single message t_2 to $\mathcal{R}$, and it also chooses and submits inputs $(y_i^0, y_i^1)_{i\in[n]}$ to the hybrid $\mathbb{OT}_n$.
3. $\mathcal{R}$ also receives $\gamma = (y_i^{c_i})_{i\in[n]}$ from $\mathbb{OT}_n$.
4. $\mathcal{R}$ outputs what is supposed to be $(x_i^{b_i})_{i\in[m]}$.

We assume in this simplified exhibition that the protocol has *perfect* completeness, namely the receiver obtains the correct answer with probability one.

An information theoretic attack for the no-oracle setting. Our starting point is an inefficient (information theoretic) attack on OT extension when there are no oracles involved. The fact that OT extension protocols, regardless of their round complexity, can *not* be information theoretically secure was already shown by Beaver [Bea96], and the work of Lindell and Zarosim [LZ13] improved that result to derive one-way functions from OT extensions. As we will see, our information theoretic attack has the main feature that in the *round-preserving* OT extension setting, it can be adapted to the random oracle model by also using tools and ideas from [IR89, BMG09, HOZ16] where new challenges arise.

Now we describe an attack for the sender and an attack for the receiver in the case that they pick their inputs $\vec{b}, x$ uniformly at random, and will show that at least one of these attacks will succeed with non-negligible probability. Ruling out the possibility of secure OT for the random-inputs case is stronger and it rules out the general (selected-input) case as well. Also note that when we refer

to *attacking* parties here, what we formally mean, is a semi-honest execution of the protocol, followed by a distinguisher (as part of the attack) who is able to use the view of the honest execution to make distinguishing predictions that shall not be possible in case of semi-honest security. For simplicity, we combine these two steps and simply refer to them as the attacker.

- **Attacking sender $\widehat{\mathcal{S}}$.** Since $\widehat{\mathcal{S}}$ gets no output, in a secure protocol the random input $\vec{b} \in \{0,1\}^m$ of the receiver shall remain indistinguishable from a uniform $\mathbf{U}_m$ in eyes of the receiver who knows the transcript $T = (t_1, t_2)$. (See Lemma 19 for a formalization.) Therefore, a natural attacking strategy for the sender $\widehat{\mathcal{S}}$ is to look at the transcript T at the end, and based on that information, try to distinguish the true $\vec{b}$ (in case it is revealed to him) from a random uniform string $\mathbf{U}_m$ of length m.[3] Thus, if the distribution of $(\vec{b}, T)$, is ε-far from $(\mathbf{U}_m, T)$ for *non-negligible* ε, the protocol is not secure, because given the transcript T an efficient sender can distinguish $\vec{b}$ from $\mathbf{U}_m$.
- **Attacking receiver $\widehat{\mathcal{R}}$.** After running the protocol honestly to get the actual output for the honestly chosen input $\vec{b}$, the cheating receiver $\widehat{\mathcal{R}}$ tries to also find *another* input $\vec{b}' \neq \vec{b}$ together with its corresponding correct output $\{x_i^{b'_i}\}_{i\in[m]}$. If $\widehat{\mathcal{R}}$ could indeed do so, it would be a successful attack since in at least one of the locations $i \in [m]$, the receiver will read both of (x_i^0, x_i^1), though that shall not be possible for semi-honest secure protocols. (See Lemma 20 for a formalization.) By the perfect completeness of the protocol,[4] all $\widehat{\mathcal{R}}$ needs to do is to find another fake view $V'_{\mathcal{R}}$ for the receiver such that: (1) $V'_{\mathcal{R}}$ contains $\vec{b}' \neq \vec{b}$ as its input, and that (2) $V'_{\mathcal{R}}$ is consistent with the transcript T, the input c given to $\mathbb{OT}_n$, as well as the output γ obtained from it. Finding such $V'_{\mathcal{R}}$ efficiently, violates sender's security.

One of $\widehat{\mathcal{S}}, \widehat{\mathcal{R}}$ succeeds: an entropy-based argument. If the attacking sender $\widehat{\mathcal{S}}$ described above does not succeed with a non-negligible advantage, it means that $(\vec{b}, T)$, as a random variable, is statistically close to $(\mathbf{U}_m, T)$, which in turn implies that (with high probability over T) conditioned on the transcript T, the receiver's input $\vec{b}$ has close to (full) m bits of entropy.[5] (See Lemma 14 for a formalization of this argument.) Therefore, if the malicious receiver $\widehat{\mathcal{R}}$, after finishing the honest execution encoded in the view $\mathbf{V}_{\mathcal{R}}$, "re-samples" a fake view $V'_{\mathcal{R}}$ from the distribution of its view conditioned on T, denoted $(\mathbf{V}_{\mathcal{R}} \mid T)$, then it will get a different $\vec{b}' \neq \vec{b}$, as part of $V'_{\mathcal{R}}$ with some noticeable probability. (See Lemma 15 for a formalization of this argument.) However, as described above, the attacking receiver $\widehat{\mathcal{R}}$ also needs to condition its sampled view

[3] Technically, the true input $\vec{b}$ or independent random input $\mathbf{U}_n$ are not given to the sender in an actual execution of the protocol, but for a secure protocol, these two shall remain indistinguishable even if revealed (see Lemma 19).

[4] Our formal proof of Sect. 3 does not assume perfect completeness.

[5] This is why we choose to work with Shannon entropy, as we want distributions close to $\mathbf{U}_m$ to have almost full entropy; this does not hold e.g., for min-entropy.

$V'_{\mathcal{R}} \leftarrow (\mathbf{V}'_{\mathcal{R}} \mid T, \vec{c}, \gamma)$ on its input c given to $\mathbb{OT}_n$ and the output γ obtained from it to get a *correct* output $\{x_i^{b'_i}\}_{i\in[m]}$ for the new fake input $\vec{b}' \neq \vec{b}$. It can be shown that if $m > |\vec{c}| + |\gamma| = 2n$, then there is still enough entropy left in the sampled $\vec{b}'$, even after further conditioning on $\vec{c}, \gamma$ (and transcript T). Therefore, if $m \gg 2n$, then at least one of the attacks succeeds with non-negligible probability.

Polynomial-query attacks in the random oracle model. The above information theoretic argument for the no-oracle case no longer works when we move to the ROM for the following simple reason. A fresh fake sample $V'_{\mathcal{R}}$ for the receiver's view that is consistent with the transcript T and OT-hybrid inputs c and output γ might be *inconsistent* with oracle query-answer pairs that already exist in sender's view, because the fake view $V'_{\mathcal{R}}$ might make up some answers to some oracle queries that are also asked by the sender but received a *different* answer from the actual oracle. Therefore, we will not have any guarantee that the faked sampled view of the sender leads to *correct* outputs for the new fake input $\vec{b}'$. In fact, because we already know that OT extension in the random oracle model *is* possible [IKNP03], the above issue is inherent when we try to extend our attack to the ROM. However, we have not yet used the fact that we are aiming at attacks that succeed for *round-preserving* OT extensions. Below, we first describe a natural (yet insufficient) idea for extending our information-theoretic attack to the ROM, and then will extend this idea further by also relying on the round-preserving aspect of the construction.

1st try: using "dependency learner" of [IR89, BMG09, HOZ16, BM17]**.** As described above, when we move to the oracle setting, the random oracle $\mathbf{H}$ creates further correlation between the views of $\mathcal{S}$ and $\mathcal{R}$ beyond what the transcript (or $\mathbb{OT}_n$) does. One natural idea for removing the correlation made by a random oracle between the views of two parties is to use the so-called 'dependency learner' Eve algorithm of [BMG09, BMG09, HOZ16, BM17] (see Theorem 17). The Eve algorithm is a deterministic algorithm such that for any inputless, two-party, protocol $\mathcal{A}, \mathcal{B}$ in the ROM, given the public transcript T of the interaction between $\mathcal{A}, \mathcal{B}$, Eve asks polynomially-many oracle queries from the random oracle $\mathbf{H}$ in a way that conditioned on the view of Eve (that includes T and its oracle query-answer pairs $P_{\mathcal{E}}$) the views of $\mathcal{A}, \mathcal{B}$ become close to *independent* random variables.[6] The magic of the algorithm Eve is that, because both parties can run it at the end, the parties can pretend that $P_{\mathcal{E}}$ is also part of the transcript, and thus we get an augmented transcript $V_{\mathcal{E}} = (T, P_{\mathcal{E}})$ that includes (almost) all of the correlation between the views of the two parties.

The above simple idea fails, however, because of the additional involvement of $\mathbb{OT}_n$ in the protocol, which creates further correlation between the views of the parties. Consequently, this seemingly simple issue prevents us from being able to run the Eve algorithm to (almost) eliminate the correlation between $\mathcal{S}, \mathcal{R}$ views, as the Eve algorithm only applies to inputless protocols in the ROM that have *no other* source of communication other than the transcript.

[6] In the plain model, the views of two interacting parties *are* independent given the transcript, and this enables the information theoretic attack against OT extension.

2nd try: using the dependency learner over a *shortened* protocol. Recall that we are dealing with round-preserving OT extensions, and have not used this property yet! One consequence of this assumption is that we can now assume that the OT-hybrid output γ is sent to the receiver *after* the last message t_2 is sent. Now, if we *stop* the execution of $\mathcal{R}$ right after t_2 is sent and call this modified variant of $\mathcal{R}$ the algorithm $\mathcal{R}_1$, even though the input $\vec{c}$ is submitted to $\mathbb{OT}_n$ by $\mathcal{R}_1$, no output is received by $\mathcal{R}_1$ from $\mathbb{OT}_n$ yet, therefore we would not have any correlated randomness distributed between the parties through $\mathbb{OT}_n$ hybrid. Therefore, our new modified two party protocol $\mathcal{S}, \mathcal{R}_1$ would be a simple inputless protocol in the ROM over which we can execute the dependency learner Eve over its transcript $T = (t_1, t_2)$. Indeed, if we run Eve with respect to $\mathcal{S}, \mathcal{R}_1$, Eve will gather enough information about the oracle encoded in its oracle query-answer set $(P_{\mathcal{E}})$ so that the views of $\mathcal{S}$ and $\mathcal{R}_1$, conditioned on Eve's view $(T, P_{\mathcal{E}})$, would be *close* to a product distribution. Therefore, we can hope to again use an approximate version of our information theoretic argument in the no-oracle setting by interpreting $T' = (T, P_{\mathcal{E}})$ as the new 'transcript'.

Finishing receiver's execution. The above argument (of applying the dependency learner Eve over a shortened variant $\mathcal{R}_1$ of $\mathcal{R}$) still does not lead to an actual attack, because we need to *finish* the execution of $\mathcal{R}_1$, which is only a *partial* execution of the receiver, to obtain the actual output corresponding to the fake input $\vec{b}'$. Only then, we can call $\widehat{\mathcal{R}}$ a successful attack. With this goal in mind, let us call $\mathcal{R}_2$ to be the rest of the execution of the cheating receiver which starts right after finishing the first part $\mathcal{R}_1$. Namely, $\mathcal{R}_2$ takes over the computation of $\mathcal{R}_1$ to finish it, and the first thing it receives is the output γ of $\mathbb{OT}_n$. However, to obtain the actual output, there might be further necessary oracle calls to the random oracle $\mathbf{H}$. Since $\widehat{\mathcal{R}}$ is interested to know the output $\vec{b}'$ planted in the fake view $V'_{\mathcal{R}}$, the execution of $\mathcal{R}_2$ using $V'_{\mathcal{R}}$ needs to pretend that $V'_{\mathcal{R}}$ is the *actual view* of the receiver, which in turn implies pretending that the original honest view $V_{\mathcal{R}}$ does not exist.

Leveraging on the lack of intersection queries. Interestingly, it turns out that another crucial property of the dependency learner algorithm Eve (i.e. Part 2 of Theorem 17) allows us to get a consistent execution of $\mathcal{R}_2$ using the fake view $V'_{\mathcal{R}}$ while pretending that the original honest (non-fake) execution of the receiver (encoded in view $V_{\mathcal{R}}$) does not exist. Namely, Eve's algorithm guarantees that, with high probability over $T' = (T, P_{\mathcal{E}})$, there will be no 'intersection queries' between the set of queries asked by the honest sender and the original (i.e., honest) partial execution of the receiver that obtains the first output (of input $\vec{b}$) for the attacker $\widehat{\mathcal{R}}$. In a nutshell, what we do to finish the execution of $\mathcal{R}_2$ is to answer with fresh random strings, any query q that is *not* learned by Eve but *is* in the view of the original honest receiver's execution. In Sect. 3 we show that a careful case analysis proves this strategy to lead to a correct continuation of the fake view $V'_{\mathcal{R}}$ obtaining the right output for the fake input $\vec{b}'$.

Organization. In Sect. 2 we describe the basic notation, main definitions, and some useful lemmas. In Sect. 3 we formalize and prove our main impossibility

result of Theorem 1. In the full version [GMMM17] we observe that Beaver's non-black-box round-preserving OT extension [Bea96] could be "chosen input".

2 Preliminaries

Logarithms in this work are taken base 2. For a bit b, we denote bit $1-b$ by $\bar{b}$. We use PPT to denote a probabilistic, polynomial-time Turing machine.

Notation on random variables. All the distributions and random variables in this paper are *finite*. We use **bold font** to denote random variables. We usually use the same non-bold letter for samples form the random variables, so by $Q \leftarrow \mathbf{Q}$ we indicate that Q is sampled from the distribution of the random variable $\mathbf{Q}$. By $(\mathbf{X}, \mathbf{Y})$ we denote a *joint* distribution over random variables $\mathbf{X}$ and $\mathbf{Y}$. By $\mathbf{X} \equiv \mathbf{Y}$ we denote that $\mathbf{X}$ and $\mathbf{Y}$ are identically distributed. For jointly distributed $(\mathbf{X}, \mathbf{Y}, \mathbf{Z})$, when random variable $\mathbf{Z}$ is clear from the context, by $((\mathbf{X}, \mathbf{Y}) \mid Z)$ we denote the distribution of $(\mathbf{X}, \mathbf{Y})$ conditioned on $\mathbf{Z} = Z$. By $(\mathbf{X} \times \mathbf{Y})$ we denote a product distribution in which $\mathbf{X}$ and $\mathbf{Y}$ are sampled *independently* from their marginal distributions. For jointly distributed $(\mathbf{X}, \mathbf{Y}, \mathbf{Z})$ and any $Z \leftarrow \mathbf{Z}$, we denote $((\mathbf{X}|Z) \times (\mathbf{Y}|Z))$ by $(\mathbf{X} \times \mathbf{Y})|Z$. For a finite set S, by $x \leftarrow \mathsf{S}$ we denote that x is sampled from S uniformly at random. By $\mathrm{Supp}(\mathbf{X})$ we denote the *support set* of the random variable $\mathbf{X}$, defined as $\mathrm{Supp}(\mathbf{X}) := \{x \mid \Pr[\mathbf{X} = x] > 0\}$. $\mathbf{U}_n$ is the uniform distribution over $\{0,1\}^n$.

Notation on events. An event B is simply a set, so for any random variable $\mathbf{X}$, the probability $\Pr[\mathbf{X} \in \mathsf{B}] := \Pr[\mathbf{X} \in \mathsf{B} \cap \mathrm{Supp}(\mathbf{X})]$ is well defined. More formally, we assume $\mathsf{B} \subseteq \mathsf{U}$ is a subset of the 'universe' set U where $\mathrm{Supp}(\mathbf{X}) \subseteq \mathsf{U}$ for any 'relevant' random variable $\mathbf{X}$ (in our analyses). In particular, we could refer to the same event B across different random variables. For any particular sample $X \leftarrow \mathbf{X}$, we say that the event B *holds over* X iff $X \in \mathsf{B}$.[7] For an event B by $\overline{\mathsf{B}}$ we denote to the complement (with respect to the underlying universe U). Therefore, $\Pr[\mathbf{X} \in \overline{\mathsf{B}}] = 1 - \Pr[\mathbf{X} \in \mathsf{B}]$ is always well defined. By $\Pr_{\mathcal{D}}[\mathsf{B}]$ or $\Pr[\mathsf{B}; \mathcal{D}]$ we mean the probability of B for sampling process described by $\mathcal{D}$.

2.1 Lemmas About Statistical Distance and Mutual Dependency

Definition 2 ((Conditional) statistical distance). *By* $\mathsf{SD}(\mathbf{X}, \mathbf{Y})$ *we denote the* statistical distance *between random variables* $\mathbf{X}, \mathbf{Y}$ *defined as*

$$\mathsf{SD}(\mathbf{X}, \mathbf{Y}) = \max_{\mathsf{B}} \Pr[\mathbf{X} \in \mathsf{B}] - \Pr[\mathbf{Y} \in \mathsf{B}] = \frac{1}{2} \cdot \sum_{Z} |\Pr[\mathbf{X} = Z] - \Pr[\mathbf{Y} = Z]|.$$

We call $\mathbf{X}$ *and* $\mathbf{Y}$ ε-close*, denoted by* $\mathbf{X} \approx_{\varepsilon} \mathbf{Y}$*, if* $\mathsf{SD}(\mathbf{X}, \mathbf{Y}) \leq \varepsilon$.

[7] In this terminology, B is seen as a *property* that holds for all $X \in \mathsf{B}$, but not for the rest. In fact, we define our events B using properties over objects in the universe.

For an event A*, we let* $\mathsf{SD}_{\mathsf{A}}(\mathbf{X}, \mathbf{Y}) = \mathsf{SD}((\mathbf{X} \mid \mathsf{A}), (\mathbf{Y} \mid \mathsf{A}))$*, denote the* conditional *statistical distance of* $\mathbf{X}, \mathbf{Y}$*, and for correlated random variable* $\mathbf{Z}$*, by* $\mathsf{SD}_Z(\mathbf{X}, \mathbf{Y})$ *we denote* $\mathsf{SD}((\mathbf{X} \mid \mathbf{Z} = Z), (\mathbf{Y} \mid \mathbf{Z} = Z))$*, and we also let*

$$\mathsf{SD}_{\mathbf{Z}}(\mathbf{X}, \mathbf{Y}) = \mathop{\mathbb{E}}_{Z \leftarrow \mathbf{Z}} \mathsf{SD}_Z(\mathbf{X}, \mathbf{Y}).$$

In the following lemma, is a well-known[8] fact stating that statistical distance is the maximum advantage of distinguishing two distributions.

Lemma 3. *Let* D *be any potentially randomized (distinguishing) algorithm. Then:* $\Pr[D(\mathbf{X}) = 1] - \Pr[D(\mathbf{Y}) = 1] \leq \mathsf{SD}(\mathbf{X}, \mathbf{Y})$ *and the equality can be achieved by any 'canonical' distinguisher such that:* $C(Z) = 1$ *if* $\Pr[\mathbf{X} = Z] > \Pr[\mathbf{Y} = Z]$*, and* $C(Z) = 0$ *if* $\Pr[\mathbf{X} = Z] < \Pr[\mathbf{Y} = Z]$.

The following well-known lemma[9] states that statistically close distributions could be sampled jointly while they are equal with high probability.

Lemma 4 (Coupling vs. statistical distance). $\mathsf{SD}(\mathbf{X}, \mathbf{Y}) \leq \varepsilon$ *iff there is a way to* jointly *sample* $(\mathbf{X}, \mathbf{Y})$ *such that* $\Pr[\mathbf{X} = \mathbf{Y}] \geq 1 - \varepsilon$.

The following lemma says that if $\mathbf{X} \equiv \mathbf{X}'$ in two pairs of jointly distributed random variables $(\mathbf{X}, \mathbf{Y}), (\mathbf{X}', \mathbf{Y}')$, then the statistical distance of the two pairs could be written as a linear combination of conditional probabilities.

Proposition 5. $\mathsf{SD}((\mathbf{X}, \mathbf{Y}), (\mathbf{X}, \mathbf{Y}')) = \mathbb{E}_{X \leftarrow \mathbf{X}} \mathsf{SD}((\mathbf{Y} \mid X), (\mathbf{Y}' \mid X))$*. Moreover, if* $\mathsf{SD}((\mathbf{X}, \mathbf{Y}), (\mathbf{X}, \mathbf{Y}')) = \varepsilon$*, any canonical distinguisher* D *of the following form* ε*-distinguishes* $(\mathbf{X}, \mathbf{Y})$ *from* $(\mathbf{X}, \mathbf{Y}')$*:*

- *If* $\Pr[\mathbf{Y} = Y \mid X] > \Pr[\mathbf{Y}' = Y \mid X]$*, then* $D(X, Y) = 1$.
- *If* $\Pr[\mathbf{Y} = Y \mid X] < \Pr[\mathbf{Y}' = Y \mid X]$*, then* $D(X, Y) = 0$.
- *If* $\Pr[\mathbf{Y} = Y \mid X] = \Pr[\mathbf{Y}' = Y \mid X]$*, then* $D(X, Y) \in \{0, 1\}$ *arbitrarily.*

Proof. We prove both parts using Lemma 3. By Lemma 3, $\mathsf{SD}((\mathbf{X}, \mathbf{Y}), (\mathbf{X}, \mathbf{Y}'))$ equals the maximum advantage by which a distinguisher D can distinguish the two distributions $(\mathbf{X}, \mathbf{Y}), (\mathbf{X}, \mathbf{Y}')$. Now, such D is always given a sample $X \leftarrow \mathbf{X}$ from $\mathbf{X} \equiv \mathbf{X}'$ first, conditioned on which it has to maximize $\Pr[D(\mathbf{Y} \mid X) = 1] - \Pr[D(\mathbf{Y}' \mid X) = 1]$. However, for each X, the maximum of $\Pr[D(\mathbf{Y} \mid X) = 1] - \Pr[D(\mathbf{Y}' \mid X) = 1]$ is again described by Lemma 3 to be equal to $\mathsf{SD}((\mathbf{Y} \mid X), (\mathbf{Y}' \mid X))$. Furthermore, the canonical distinguisher described above works due to the canonical distinguisher of Lemma 3 □

The following definition from [BM17] is a measure of correlation between jointly distributed pairs of random variables.

[8] For example, see Exercise 8.61 from [Sho09].
[9] For example, see Lemma 3.6 of [Ald83] for a proof.

Definition 6 ((Conditional) mutual dependency [BM17]). *For a joint distribution* $(\mathbf{X}, \mathbf{Y})$*, we define their* mutual-dependency *as* $\mathsf{MutDep}(\mathbf{X}, \mathbf{Y}) = \mathsf{SD}((\mathbf{X}, \mathbf{Y}), (\mathbf{X} \times \mathbf{Y}))$*. For correlated* $(\mathbf{X}, \mathbf{Y}, \mathbf{Z})$*, and for* $Z \leftarrow \mathbf{Z}$*, we define*

$$\mathsf{MutDep}_Z(\mathbf{X}, \mathbf{Y}) = \mathsf{SD}_Z((\mathbf{X}, \mathbf{Y}), (\mathbf{X} \times \mathbf{Y})) = \mathsf{SD}(((\mathbf{X}, \mathbf{Y})|Z), (\mathbf{X}|Z \times \mathbf{Y}|Z))$$

to be the mutual dependency of $\mathbf{X}, \mathbf{Y}$ *conditioned on the given* Z*, and we let*

$$\mathsf{MutDep}_{\mathbf{Z}}(\mathbf{X}, \mathbf{Y}) = \mathop{\mathbb{E}}_{Z \leftarrow \mathbf{Z}} \mathsf{MutDep}_Z(\mathbf{X}, \mathbf{Y}).$$

The following proposition follows from Proposition 5 and Definition 6.

Proposition 7. *It holds that* $\mathsf{MutDep}(\mathbf{X}, \mathbf{Y}) = \mathbb{E}_{X \leftarrow \mathbf{X}} \mathsf{SD}((\mathbf{Y} \mid X), \mathbf{Y})$.

Lemma 8. *For a joint distribution* $(\mathbf{X}, \mathbf{Y})$*, the statistical distance between the following distributions is at most* $2 \cdot \mathsf{MutDep}(\mathbf{X}, \mathbf{Y})$*. (Note how* Y, Y' *are flipped.)*

1. *Sample* $(X, Y) \leftarrow (\mathbf{X}, \mathbf{Y})$*, independently sample* $Y' \leftarrow \mathbf{Y}$*, output* (X, Y, Y')*.*
2. *Sample* $(X, Y) \leftarrow (\mathbf{X}, \mathbf{Y})$*, independently sample* $Y' \leftarrow \mathbf{Y}$*, output* (X, Y', Y)*.*

Proof. The following hybrid distribution is $\mathsf{MutDep}(\mathbf{X}, \mathbf{Y})$-far from either of the distributions in Lemma 8. Sample $X \leftarrow \mathbf{X}, Y_1, Y_2 \leftarrow \mathbf{Y}$ all independently and output (X, Y_1, Y_2). Therefore, the claim follows from the triangle inequality. □

Lemma 9. *Let* $\mathbf{X} = (\mathbf{A}, \mathbf{B}, \mathbf{C})$ *be correlated random variables. Let another joint distribution* $\mathbf{X}' = (\mathbf{A}', \mathbf{B}', \mathbf{C}')$ *be defined as follows.*

- *Sample* $A' \leftarrow \mathbf{A}$*, then* $C' \leftarrow (\mathbf{C} \mid \mathbf{A} = A')$*, then* $B' \leftarrow (\mathbf{B} \mid \mathbf{C} = C')$*, and output the sample* $X' = (A', B', C')$*.*

Then $\mathsf{SD}(\mathbf{X}, \mathbf{X}') = \mathsf{MutDep}_{\mathbf{C}}(\mathbf{A}, \mathbf{B})$*. Furthermore, if* $\mathbf{C} = f(\mathbf{B})$ *is a function of only* $\mathbf{B}$ *(in the joint distribution* $\mathbf{X}$*) then* $\mathsf{SD}(\mathbf{X}, \mathbf{X}') \leq 2 \cdot \mathsf{MutDep}(\mathbf{A}, \mathbf{B})$*.*

Remark 10. Before proving Lemma 9, note that the second conclusion would be false if $\mathbf{C}$ could also depend on $\mathbf{A}$. For example, consider the case where $\mathbf{A}, \mathbf{B}, \mathbf{C}$ are all random bits conditioned on $\mathbf{A} \oplus \mathbf{B} \oplus \mathbf{C} = 0$. In that case, without conditioning on $\mathbf{C}$, $\mathsf{MutDep}(\mathbf{A}, \mathbf{B}) = 0$ as $\mathbf{A}, \mathbf{B}$ are independent. However, given any specific bit $C \leftarrow \mathbf{C}$, the distributions of $\mathbf{A}, \mathbf{B}$ would be correlated, and their conditional mutual-dependency would be $1/2$, so $\mathsf{MutDep}_{\mathbf{C}}(\mathbf{A}, \mathbf{B}) = 1/2$.

Proof (of Lemma 9). First, we show $\mathsf{SD}(\mathbf{X}, \mathbf{X}') = \mathsf{MutDep}_{\mathbf{C}}(\mathbf{A}, \mathbf{B})$. Note that $\mathbf{C} \equiv \mathbf{C}'$, so we can apply Proposition 7. For a given $C \leftarrow \mathbf{C} \equiv \mathbf{C}'$, for $(\mathbf{X} \mid C)$ we will sample $(\mathbf{A}, \mathbf{B})$ jointly, while in $(\mathbf{X}' \mid \mathbf{C}' = C)$ we will sample from $(\mathbf{A} \mid C) \equiv (\mathbf{A}' \mid \mathbf{C}' = C)$ and $(\mathbf{B} \mid C) = (\mathbf{B}' \mid \mathbf{C}' = C)$ independently from their marginal distributions. Now, we show that $\mathsf{SD}(\mathbf{X}, \mathbf{X}') \leq 2 \cdot \mathsf{MutDep}(\mathbf{X}, \mathbf{Y})$, if we further know that $\mathbf{C}$ is only a function of $\mathbf{B}$. Consider a third joint distribution $\mathbf{X}'' = (\mathbf{A}'', \mathbf{B}'', \mathbf{C}'') \equiv (\mathbf{A} \times (\mathbf{B}, \mathbf{C}))$; namely, $(\mathbf{B}'', \mathbf{C}'') \equiv (\mathbf{B}, \mathbf{C})$, and $\mathbf{A}''$ is sampled from the marginal distribution of $\mathbf{A}$. Firstly, note that for every $A \leftarrow \mathbf{A}, B \leftarrow \mathbf{B}$, it holds that $(\mathbf{C}'' \mid \mathbf{A}'' = A, \mathbf{B}'' = B) \equiv (\mathbf{C} \mid \mathbf{B} = B) \equiv (\mathbf{C} \mid \mathbf{A} = A, \mathbf{B} = B)$,

because $\mathbf{A}''$ is independently sampled from $(\mathbf{B}'', \mathbf{C}'')$, and that $\mathbf{C} = f(\mathbf{B})$ is only a function of $\mathbf{B}$. Therefore, because the conditional distribution of $\mathbf{C} \equiv \mathbf{C}''$ is the same given $(\mathbf{A}'' = A, \mathbf{B}'' = B)$ or $(\mathbf{A} = A, \mathbf{B} = B)$, by Lemma 12,

$$\mathsf{SD}(\mathbf{X}, \mathbf{X}'') = \mathsf{SD}((\mathbf{A}, \mathbf{B}), (\mathbf{A}'', \mathbf{B}'')) = \mathsf{MutDep}(\mathbf{A}, \mathbf{B}). \tag{1}$$

Secondly, for all $A \leftarrow \mathbf{A}, C \leftarrow \mathbf{C}$, it holds that $(\mathbf{B}'' \mid \mathbf{A}'' = A, \mathbf{C}'' = C) \equiv (\mathbf{B} \mid \mathbf{C} = C) \equiv (\mathbf{B}' \mid \mathbf{A}' = A, \mathbf{C}' = C)$, so by Lemma 12, it holds that

$$\mathsf{SD}(\mathbf{X}', \mathbf{X}'') = \mathsf{MutDep}(\mathbf{A}, \mathbf{C}) \leq \mathsf{SD}(\mathbf{X}, \mathbf{X}'') = \mathsf{MutDep}(\mathbf{A}, \mathbf{B}). \tag{2}$$

Therefore, by the triangle inequality and Eqs. (1) and (2), it holds that $\mathsf{SD}(\mathbf{X}, \mathbf{X}') \leq \mathsf{SD}(\mathbf{X}, \mathbf{X}'') + \mathsf{SD}(\mathbf{X}', \mathbf{X}'') \leq 2 \cdot \mathsf{MutDep}(\mathbf{A}, \mathbf{B})$. □

Variations of the following lemma are used in previous works.[10] It states an intuitive way to bound the statistical distance of sequences of random variables in systems where there exist some low-probability 'bad' events, and conditioned on those bad events not happening the two systems proceed statistically closely. Here we only need this specific variant for random systems with two blocks.

Lemma 11 (Bounding statistical distance of pairs). *Let* $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2)$ *and* $\mathbf{X}' = (\mathbf{X}'_1, \mathbf{X}'_2)$ *be two jointly distributed pairs of random variables where* $\mathsf{SD}(\mathbf{X}_1, \mathbf{X}'_1) \leq \alpha$. *Let* B *be an event (i.e. an arbitrary set) such that for every* $X_1 \in \mathrm{Supp}(\mathbf{X}_1) \cap \mathrm{Supp}(\mathbf{X}'_1) \setminus \mathsf{B}$ *it holds that* $\mathsf{SD}((\mathbf{X}_2 \mid \mathbf{X}_1 = X_1), (\mathbf{X}'_2 \mid \mathbf{X}'_1 = X_1)) \leq \beta$. *Then, it holds that*

$$\mathsf{SD}(\mathbf{X}, \mathbf{X}') \leq \alpha + \beta + \Pr[\mathbf{X}_1 \in \mathsf{B}].$$

Proof. Using two direct applications of Lemma 4, we show how to sample $(\mathbf{X}, \mathbf{X}')$ *jointly* in a way that $\Pr[\mathbf{X} = \mathbf{X}'] \geq 1 - (\alpha + \beta + \rho)$ where $\Pr[\mathbf{X}_1 \in \mathsf{B}] = \rho$. Then Lemma 11 follows (again by an application of Lemma 4).

Firstly, by Lemma 4 we can sample $(\mathbf{X}_1, \mathbf{X}'_1)$ jointly, while $\Pr[\mathbf{X}_1 = \mathbf{X}'_1] \geq 1 - \alpha$. Now, we *expand* the joint sampling of $(\mathbf{X}_1, \mathbf{X}'_1)$ to a full joint sampling of $(\mathbf{X}, \mathbf{X}') \equiv (\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}'_1, \mathbf{X}'_2)$ as follows. We first sample $(X_1, X'_1) \leftarrow (\mathbf{X}_1, \mathbf{X}'_1)$ from their joint distribution. Then, for each sampled (X_1, X'_1), we sample the distributions $(\mathbf{X}_2, \mathbf{X}'_2 \mid X_1, X'_1)$ also *jointly* such that $\Pr[\mathbf{X}_2 = \mathbf{X}'_2 \mid X_1, X'_1] = 1 - \mathsf{SD}((\mathbf{X}_2 \mid X_1), (\mathbf{X}'_2 \mid X'_1))$. We can indeed do such joint sampling, again by applying Lemma 4, but this time we apply that lemma to the conditional distributions $(\mathbf{X}_2 \mid X_1, X'_1) \equiv (\mathbf{X}_2 \mid X_1)$ and $(\mathbf{X}'_2 \mid X_1, X'_1) \equiv (\mathbf{X}'_2 \mid X'_1)$.

Now, we lower bound $\Pr[\mathbf{X}_1 = \mathbf{X}'_1 \wedge \mathbf{X}_2 = \mathbf{X}'_2]$ when we sample all the blocks through the joint distribution $(\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}'_1, \mathbf{X}'_2)$ defined above. First, we know that $\Pr[\mathbf{X}_1 = \mathbf{X}'_1] \geq 1 - \alpha$ and $\Pr[\mathbf{X}_1 \notin \mathsf{B}] \geq 1 - \rho$, therefore $\Pr[\mathbf{X}_1 = \mathbf{X}'_1 \notin \mathsf{B}] \geq 1 - \alpha - \rho$. Moreover, for any such $X_1 \in \mathrm{Supp}(\mathbf{X}_1) \cap \mathrm{Supp}(\mathbf{X}'_1) \setminus \mathsf{B}$, we have

$$\Pr[\mathbf{X}_2 = \mathbf{X}'_2 \mid \mathbf{X}_1 = \mathbf{X}'_1 = X_1] \geq 1 - \mathsf{SD}((\mathbf{X}_2 \mid X_1), (\mathbf{X}'_2 \mid \mathbf{X}'_1 = X_1)) \geq 1 - \beta.$$

Therefore, the lemma follows by a union bound. □

[10] For example see Lemma 2.2 of [GKLM12].

The following useful lemma could be derived as a special case of Lemma 11 above by letting $\mathsf{B} = \mathrm{Supp}(\mathbf{X}_1) \cup \mathrm{Supp}(\mathbf{X}'_1)$ and $\beta = 0$.

Lemma 12. *If* $(\mathbf{X}, \mathbf{Y}), (\mathbf{X}', \mathbf{Y}')$ *are joint distributions and* $(\mathbf{Y} \mid X) \equiv (\mathbf{Y}' \mid \mathbf{X}' = X)$ *for all* $X \in \mathrm{Supp}(\mathbf{X}) \cap \mathrm{Supp}(\mathbf{X}')$*, then* $\mathsf{SD}((\mathbf{X}, \mathbf{Y}), (\mathbf{X}', \mathbf{Y}'')) = \mathsf{SD}(\mathbf{X}, \mathbf{X}')$.

2.2 Lemmas About Shannon Entropy

Definition 13 ((Conditional) Shannon entropy). *For a random variable* $\mathbf{X}$*, its Shannon entropy is defined as* $\mathrm{H}(\mathbf{X}) = \mathbb{E}_{X \leftarrow \mathbf{X}} \log(1/\Pr[\mathbf{X} = X])$*. The conditional (Shannon) entropy is defined as* $\mathrm{H}(\mathbf{X} \mid \mathbf{Y}) = \mathbb{E}_{Y \leftarrow \mathbf{Y}} \mathrm{H}(\mathbf{X} \mid Y)$*. The binary (Shannon) entropy function* $\mathrm{H}(\varepsilon) = -p \log p - (1-p)\log(1-p)$ *is equal to the entropy of a Bernoulli process with probability* ε*.*[11]

Jensen's inequality implies that we always have $\mathrm{H}(\mathbf{X}) \geq \mathrm{H}(\mathbf{X} \mid \mathbf{Y}) \geq \mathbf{0}$.

Lemma 14 (Lower bounding entropy using statistical distance). *Suppose* $\mathsf{SD}(\mathbf{X}, \mathbf{U}_n) \leq \varepsilon$*. Then* $\mathrm{H}(\mathbf{X}) \geq (1-\varepsilon) \cdot n - \mathrm{H}(\varepsilon)$.

Proof. Since $\mathsf{SD}(\mathbf{X}, \mathbf{U}_n) \leq \varepsilon$, using Lemma 4 we can sample $(\mathbf{X}, \mathbf{U}_n)$ jointly such that $\Pr[\mathbf{X} \neq \mathbf{U}_n] \leq \varepsilon$. In this case, we have

$$n = \mathrm{H}(\mathbf{U}_n) \leq \mathrm{H}(\mathbf{X}_n, \mathbf{U}_n) = \mathrm{H}(\mathbf{X}) + \mathrm{H}(\mathbf{U}_n \mid \mathbf{X}) \leq \mathrm{H}(\mathbf{X}) + \mathrm{H}(\varepsilon) + \varepsilon \cdot \log(2^n - 1)$$

where the last inequality follows from Fano's lemma [Fan68]. Therefore, we get $\mathrm{H}(\mathbf{X}) \geq (1-\varepsilon) \cdot n - \mathrm{H}(\varepsilon)$. □

Lemma 15 (Upper-bounding collision probability using (conditional) Shannon entropy). *Suppose* $\mathrm{Supp}(\mathbf{X}) \subseteq \{0,1\}^n$.

1. *If* $\mathrm{H}(\mathbf{X}) \geq 2/3$*, then it holds that*

$$\Pr_{X_1, X_2 \leftarrow \mathbf{X}}[X_1 \neq X_2] \geq \frac{1}{10n}.$$

2. *If* $\mathrm{H}(\mathbf{X} \mid \mathbf{Y}) \geq 5/6$ *for a jointly distributed* $(\mathbf{X}, \mathbf{Y})$*, then it holds that*

$$\Pr_{Y \leftarrow \mathbf{Y}, X_1, X_2 \leftarrow (\mathbf{X} \mid Y)}[X_1 \neq X_2] \geq \frac{1}{60 \cdot n^2}.$$

Proof. First, we prove Part 1. In the following let $\varepsilon = 1/(10n) \leq 1/10$. Our first goal is to show that $\Pr_{X_1, X_2 \leftarrow \mathbf{X}}[X_1 \neq X_2] \geq \varepsilon$. There are two cases to consider:

[11] The notation is well defined: If the input ε is a real number, by $\mathrm{H}(\varepsilon)$ we mean the binary entropy, and otherwise we mean the entropy of a random variable.

1. Case (1): Suppose first that there is some $\mathsf{A} \subseteq \mathrm{Supp}(\mathbf{X})$ with $\varepsilon \leq p_{\mathsf{A}} = \Pr_{X \leftarrow \mathbf{X}}[X \in \mathsf{A}] \leq 1 - \varepsilon$. Then, letting $\mathsf{B} = \mathrm{Supp}(\mathbf{X}) \setminus \mathsf{A}$, we also have $\varepsilon \leq \Pr_{X \leftarrow \mathbf{X}}[X \in \mathsf{B}] \leq 1 - \varepsilon$. Since A and B are disjoint, we have

$$\begin{aligned}\Pr_{X_1, X_2 \leftarrow \mathbf{X}}[X_1 \neq X_2] &\geq \Pr_{X_1, X_2 \leftarrow \mathbf{X}}[X_1 \in \mathsf{A}, X_2 \in \mathsf{B} \text{ or } X_1 \in \mathsf{B}, X_2 \in \mathsf{A}] \\ &= 2 \cdot p_{\mathsf{A}} \cdot (1 - p_{\mathsf{A}}) \geq 2 \cdot \varepsilon \cdot (1 - \varepsilon) = 2 \cdot \varepsilon - 2 \cdot \varepsilon^2 \geq \varepsilon.\end{aligned}$$

   The last inequality follows from $\varepsilon \leq 1/10$, which implies $\varepsilon \geq 2\varepsilon^2$.
2. If we are not in Case (1) above, then for every $\mathsf{A} \subseteq \mathrm{Supp}(\mathbf{X})$, $\Pr_{X \leftarrow \mathbf{X}}[X \in \mathsf{A}] < \varepsilon$ or $\Pr_{X \leftarrow \mathbf{X}}[X \in \mathsf{A}] > 1 - \varepsilon$. In particular, for every $X \in \mathrm{Supp}(\mathbf{X})$, we have $\Pr[\mathbf{X} = X] < \varepsilon$ or $\Pr[\mathbf{X} = X] > 1 - \varepsilon$. Now there are two cases:
   (a) For all $X \in \mathrm{Supp}(\mathbf{X})$, $\Pr[\mathbf{X} = X] < \varepsilon$. In this case, because $\varepsilon < 1/10$, we can build a set $\mathsf{A} \subseteq \mathrm{Supp}(\mathbf{X})$ that implies being in Case (1). Namely, let $\mathsf{A}_0, \mathsf{A}_1, \ldots, \mathsf{A}_m$ be a sequence of sets where $\mathsf{A}_i = \{1, \ldots, i\} \subseteq [m] = \mathrm{Supp}(\mathbf{X})$. Suppose i is the smallest number for which $\Pr[\mathbf{X} \in \mathsf{A}_i] \geq \varepsilon$, which means $\Pr[\mathbf{X} \in \mathsf{A}_{i-1}] < \varepsilon$. In this case we have:

$$\Pr[\mathbf{X} \in \mathsf{A}_i] \leq \Pr[\mathbf{X} \in \mathsf{A}_{i-1}] + \Pr[\mathbf{X} = i] < 2\varepsilon < 1 - \varepsilon$$

   where the last inequality follows from $\varepsilon < 1/10$.
   (b) There is some $X \in \mathrm{Supp}(\mathbf{X})$ where $\Pr[\mathbf{X} = x] > 1 - \varepsilon$. Now suppose we sample $\mathbf{X}$ jointly with a Boolean $\mathbf{B}$ where $\mathbf{B} = 0$ iff $\mathbf{X} = X$. So, we get:

$$\begin{aligned}2/3 \leq \mathrm{H}(\mathbf{X}) &\leq \mathrm{H}(\mathbf{B}) + \mathrm{H}(\mathbf{X} \mid \mathbf{B}) \\ &= \mathrm{H}(\mathbf{B}) + \Pr[\mathbf{B} = 0] \cdot \mathrm{H}(\mathbf{X} \mid \mathbf{B} = 0) + \Pr[\mathbf{B} = 1] \cdot \mathrm{H}(\mathbf{X} \mid \mathbf{B} = 1) \\ &< \mathrm{H}(\varepsilon) + \Pr[\mathbf{B} = 0] \cdot 0 + \varepsilon \cdot n \\ &\leq \mathrm{H}(1/10) + 1/10 \\ &< 1/2 + 1/10 \text{ (because } H(1/10) < 1/2)\end{aligned}$$

   which is a contradiction.

Now we prove Part 2. Because we have $\mathrm{H}(\mathbf{X} \mid \mathbf{Y}) \geq 5/6$, and because $\mathrm{H}(\mathbf{X} \mid Y) \leq n$ for any $Y \leftarrow \mathbf{Y}$, by an averaging argument it holds that $\Pr_{Y \leftarrow \mathbf{Y}}[\mathrm{H}(\mathbf{X} \mid Y) > 2/3] \geq 1/(6n)$. That is because otherwise, $\mathrm{H}(\mathbf{X} \mid \mathbf{Y})$ would be at most $(2/3) \cdot (1 - 1/(6n)) + n \cdot (1/(6n)) < 5/6$. Therefore, with probability at least $1/(6n)$ we get $Y \leftarrow \mathbf{Y}$ for which we have

$$\Pr[X_1 \neq X_2; Y \leftarrow \mathbf{Y}, X_1, X_2 \leftarrow (\mathbf{X} \mid Y)] \geq 1/(10n).$$

The claim then follows by using the chain rule. □

2.3 Lemmas About the Random Oracle Model

Definition 16 (Random Oracles). *A* random oracle $\mathbf{H}(\cdot)$ *is a randomized function such that for all* $x \in \{0,1\}^*$, $\mathbf{H}(x)$ *is independently mapped to a random string of the same length* $|x|$.

Even though the above definition is for infinite random oracles, in this work we are only interested and only use *finite* random oracles, as there is always an upper bound (based on the security parameter) on the maximum length of the queries asked by a polynomial time algorithm.

Notation on oracle-aided algorithms. For any view $V_{\mathcal{A}}$ of a party $\mathcal{A}$ with access to some oracle O, by $\mathcal{Q}(V_{\mathcal{A}})$ we refer to the set of queries to O in the view $V_{\mathcal{A}}$, and by $\mathcal{P}(V_{\mathcal{A}})$ we denote the set of oracle query-answer pairs in $V_{\mathcal{A}}$. So, $\mathcal{Q}(\cdot), \mathcal{P}(\cdot)$ are operators that extract the queries or query-answer pairs. When it is clear from the context, we might simply use $Q_{\mathcal{A}} = \mathcal{Q}(V_{\mathcal{A}})$ and by $P_{\mathcal{A}} = \mathcal{P}(V_{\mathcal{A}})$. When $\mathcal{A}$ is an interactive algorithm, if $\mathcal{A}$ has no inputs and uses randomness $r_{\mathcal{A}}$, and if T is the transcript of the interaction, then $V_{\mathcal{A}} = (r_{\mathcal{A}}, T, P_{\mathcal{A}})$.

Variants of the following lemma were implicit in [IR89, BMG09] and stated in [DLMM11]. See the works of [HOZ16, BM17] for formal proofs.

Theorem 17 (Dependency learner [IR89, BMG09, HOZ16, BM17]). *Let $(\mathcal{A}, \mathcal{B})$ be an interactive protocol between Alice and Bob in which they might use private randomness (but no inputs otherwise) and they each ask at most m queries to a random oracle $\mathbf{H}$. Then, there is a deterministic eavesdropping algorithm Eve (whose algorithm might depend on Alice and Bob and) who gets as input $\delta \in [0,1]$ and the transcript T of the messages exchanged between Alice and Bob, asks at most $\mathrm{poly}(m/\delta)$ queries to the random oracle $\mathbf{H}$, and we have:*

1. *The average of the statistical distance between $(\mathbf{V}_{\mathcal{A}}, \mathbf{V}_{\mathcal{B}})$ and $(\mathbf{V}_{\mathcal{A}} \times \mathbf{V}_{\mathcal{B}})$ conditioned on $\mathbf{V}_{\mathcal{E}}$ is at most δ. Namely,*

$$\mathsf{MutDep}_{\mathbf{V}_{\mathcal{E}}}(\mathbf{V}_{\mathcal{A}}, \mathbf{V}_{\mathcal{B}}) = \mathop{\mathbb{E}}_{V_{\mathcal{E}} \leftarrow \mathbf{V}_{\mathcal{E}}} \mathsf{MutDep}((\mathbf{V}_{\mathcal{A}} \mid V_{\mathcal{E}}), (\mathbf{V}_{\mathcal{B}} \mid V_{\mathcal{E}})) \leq \delta.$$

2. *The probability that Alice and Bob have an 'intersection query' outside of the queries asked by Eve to the random oracle is bounded as follows:*

$$\Pr[\mathcal{Q}(\mathbf{V}_{\mathcal{A}}) \cap \mathcal{Q}(\mathbf{V}_{\mathcal{B}}) \not\subseteq \mathcal{Q}(\mathbf{V}_{\mathcal{E}})] \leq \delta.$$

The two parts of Theorem 17 could be derived from each other, but doing that is not trivial and involves asking *more* oracle queries from the oracle. We will use both of the properties in our formal proof of our main result in Sect. 3.

Notation for indistinguishability in the ROM. For families of random variables $\{\mathbf{X}_\kappa\}, \{\mathbf{Y}_\kappa\}$ by $\mathbf{X}_\kappa \equiv_c \mathbf{Y}_\kappa$ we mean that $\{\mathbf{X}_\kappa\}, \{\mathbf{Y}_\kappa\}$ are indistinguishable against nonuniform PPT algorithms. When we are in the random oracle model, we use the same notation $\mathbf{X}_\kappa \equiv_c \mathbf{Y}_\kappa$ when the distinguishers are $\mathrm{poly}(\kappa)$-query algorithms. Namely, for any $\mathrm{poly}(\kappa)$-query oracle-aided algorithm D there is a negligible function ε, such that $\Pr[D(\mathbf{X}_\kappa) = 1] - \Pr[D(\mathbf{Y}_\kappa) = 1] \leq \varepsilon(\kappa)$, where the probabilities are over the inputs $\mathbf{X}_\kappa, \mathbf{Y}_\kappa$ and the randomness of D and the oracle $\mathbf{H}$. When κ is clear from the context, we write $\mathbf{X} \equiv_c \mathbf{Y}$ for simplicity.

2.4 OT and its Multi-Input Variant k-OT in the ROM

In this subsection, we recall the notions of OT and its multi-input version on k inputs, denoted k-OT. We will also prove basic lemmas that allows us to prove the existence of *attacks* against semi-honest security of k-OT.

We start by defining (multi-) oblivious transfer (OT) formally.

Definition 18 (k-OT). *A k-parallel 1-out-of-2 oblivious transfer (OT). functionality (k-OT) is a two-party functionality between a sender $\mathcal{S}$ and a receiver $\mathcal{R}$ as follows. The sender has input $\{x_i^0, x_i^1\}_{i\in[k]}$ which are arbitrary strings, and the receiver has the input $\vec{b} \in \{0,1\}^k$. The sender receives no output at the end, while the receiver receives $\{x_i^{b_i}\}_{i\in[k]}$.*

Semi-honest security of k-OT. We use standard definition of simulation-based security, see e.g. [Lin16]. In particular, for any semi-honest secure OT protocol between $\mathcal{S}$ and $\mathcal{R}$, there are two PPT simulator $\mathsf{Sim}_{\mathcal{S}}, \mathsf{Sim}_{\mathcal{R}}$ such that for any input $\vec{b}$ of $\mathcal{R}$ and any input $x = \{x_i^0, x_i^1\}_{i\in[k]}$ for $\mathcal{S}$, it holds that:

$$\mathsf{Sim}_{\mathcal{S}}(x) \equiv_c \mathbf{V}_{\mathcal{S}}(x, \vec{b}) \quad \text{and} \quad \mathsf{Sim}_{\mathcal{R}}(\vec{b}, \{x_i^{b_i}\}_{i\in[k]}) \equiv_c \mathbf{V}_{\mathcal{R}}(x, \vec{b}).$$

Plain model security vs. the ROM security. In the plain model all the parties (including the simulator and the adversary and the distinguishers) are PPT algorithms. In the random random oracle model we, the honest parties and the simulators are oracle-aided PPTs, while the *distinguishers* are poly(κ)-query (computationally unbounded) algorithms accessing the random oracle $\mathbf{H}$.[12] Recall that by the notation defined at the end of Sect. 2 we can use the same notation $\equiv_c$ for indistinguishably against poly(κ) distinguishers in the ROM.

Sufficient conditions for breaking the semi-honest security of k-OT. We now state and prove two simple lemmas showing that the attacks that we construct in Sect. 3 are indeed attacks according to the standard definition of simulation-based security, see e.g. [Lin16]. The following lemma, states the intuitive fact that in any OT protocol, the input of the sender should remain indistinguishable from a random string, if the receiver chooses its input randomly.

Lemma 19. *Let $(\mathcal{S}, \mathcal{R})$ be a semi-honest secure m-OT protocol in the plain model (resp., in the ROM) in which the receiver's inputs are chosen uniformly at random from $\{0,1\}^m$, and in which $\mathcal{S}, \mathcal{R}$ are PPTs (resp., oracle-aided PPTs). Fix any input x for the sender. Let $\vec{\mathbf{b}} \equiv \mathbf{U}_m$ be the uniformly random inputs of the receiver and $\mathbf{V}_{\mathcal{S}}(x, \vec{\mathbf{b}})$ the random variable denoting the view of the sender (for inputs $x, \vec{\mathbf{b}}$ being used by the sender and the receiver). Then we have*

$$(\mathbf{V}_{\mathcal{S}}(x, \vec{\mathbf{b}}), \vec{\mathbf{b}}) \equiv_c (\mathbf{V}_{\mathcal{S}}(x, \vec{\mathbf{b}}) \times \mathbf{U}_m).$$

(Recall that in the ROM, the distinguisher is an poly(κ)-query, computationally unbounded, algorithm for security parameter κ.)

[12] Note that this definition is for a setting where the random oracle is the sole source of hardness. E.g., this is how the security of the protocol in [IKNP03] could be proved.

Proof. We prove the lemma for the computational setting in the plain model where the distinguishers are PPT algorithms. The same proof holds for the random oracle model in which the distinguishers are poly(κ)-query oracle-aided algorithms accessing a random oracle $\mathbf{H}$, where κ is the security parameter.

By the security definition of OT, there is a PPT simulator $\mathsf{Sim}_{\mathcal{S}}$ such that for any input $\vec{b}$ of $\mathcal{R}$ it simulates the view of $\mathcal{S}$:

$$\mathsf{Sim}_{\mathcal{S}}(x) \equiv_c \mathbf{V}_{\mathcal{S}}(x, \vec{b}).$$

Hence, by averaging over $\vec{b} \leftarrow \vec{\mathbf{b}}$, we have $(\mathbf{V}_{\mathcal{S}}(x, \vec{\mathbf{b}}), \vec{\mathbf{b}}) \equiv_c (\mathsf{Sim}_{\mathcal{S}}(x), \vec{\mathbf{b}})$ for uniform $\vec{\mathbf{b}}$. (In other words, if the latter two were distinguishable, one could distinguish $\mathsf{Sim}_{\mathcal{S}}(x)$ from $\mathbf{V}_{\mathcal{S}}(x, \vec{b})$ by the same advantage for some $\vec{b}$.) Since $\mathsf{Sim}_{\mathcal{S}}(x)$ is independent of the receiver's input $\vec{\mathbf{b}}$, we conclude

$$(\mathbf{V}_{\mathcal{S}}(x, \vec{\mathbf{b}}), \vec{\mathbf{b}}) \equiv_c (\mathsf{Sim}_{\mathcal{S}}(x), \vec{\mathbf{b}}) \equiv \mathsf{Sim}_{\mathcal{S}}(x) \times \mathbf{U}_m \equiv_c \mathbf{V}_{\mathcal{S}}(x, \vec{\mathbf{b}}) \times \mathbf{U}_m. \qquad \square$$

Lemma 20. *Let $(\mathcal{S}, \mathcal{R})$ be a semi-honest secure m-OT protocol in which the sender's inputs are chosen uniformly at random and in which $\mathcal{S}, \mathcal{R}$ are PPTs (resp., oracle-aided PPTs). Fix any input vector $\vec{b}$ for the receiver. Let $\mathbf{x} = \{\mathbf{x}_i^0, \mathbf{x}_i^1\}_{i\in[m]}$ be uniformly random inputs for the sender, and let $\mathbf{V}_{\mathcal{R}}(\mathbf{x}, \vec{b})$ be the random variable denoting the view of the receiver (when the inputs $\mathbf{x}, \vec{b}$ are used by the two parties). Then, it holds that*

$$(\mathbf{V}_{\mathcal{R}}(\mathbf{x}, \vec{b}), \{\mathbf{x}_i^{\bar{b}_i}\}_{i\in[m]}) \equiv_c (\mathbf{V}_{\mathcal{R}}(\mathbf{x}, \vec{b}) \times \{\mathbf{x}'_i\}_{i\in[m]})$$

where $\mathbf{x}'_i$'s are independent uniformly random strings of the appropriate length.

Proof. As in the proof of Lemma 19, we only prove the lemma for the computational setting in the plain model where the distinguishers are PPT algorithms. The same proof holds for random oracle model in which the distinguishers are poly(κ)-query algorithms in the random oracle model, for security parameter κ.

By the security definition of m-OT, there is a PPT simulator $\mathsf{Sim}_{\mathcal{R}}$ such that for any input $\{x_i^0, x_i^1\}_{i\in[m]}$ of $\mathcal{S}$ it simulates the view of $\mathcal{R}$:

$$\mathsf{Sim}_{\mathcal{R}}(\vec{b}, \{x_i^{b_i}\}_{i\in[m]}) \equiv_c \mathbf{V}_{\mathcal{R}}(x, \vec{b}).$$

Hence $(\mathbf{V}_{\mathcal{R}}(\mathbf{x}, \vec{b}), \{\mathbf{x}_i^{\bar{b}_i}\}_{i\in[m]}) \equiv_c (\mathsf{Sim}_{\mathcal{R}}(\vec{b}, \{\mathbf{x}_i^{b_i}\}_{i\in[m]}), \{\mathbf{x}_i^{\bar{b}_i}\}_{i\in[m]})$ holds for uniform $\mathbf{x}$. Since $\mathsf{Sim}_{\mathcal{R}}(\vec{b}, \{\mathbf{x}_i^{b_i}\}_{i\in[m]})$ is independent of $\{\mathbf{x}_i^{\bar{b}_i}\}_{i\in[m]}$ (i.e., the sender's input that is not learned by receiver), similarly to Lemma 19, we conclude that

$$(\mathbf{V}_{\mathcal{R}}(\mathbf{x}, \vec{b}), \{\mathbf{x}_i^{\bar{b}_i}\}_{i\in[m]}) \equiv_c \mathbf{V}_{\mathcal{R}}(\mathbf{x}, \vec{b}) \times \{\mathbf{x}'_i\}_{i\in[m]}. \qquad \square$$

Remark 21. In Sect. 3, we will use Lemma 19 for getting a (computationally unbounded) poly(κ)-query attacking sender. Namely, instead of directly breaking the semi-honest security definition of k-OT, our attacking semi-honest sender (or more accurately, the distinguisher), will pursue a different goal based on

Lemma 19. Namely, based on his own view, the attacking sender will try to distinguish the receiver's actual input $\vec{b}$ from an independently uniform string.

Similarly, we will use Lemma 20 to get a (computationally unbounded) poly(κ)-query attacking receiver. Namely, our attacking semi-honest receiver (i.e., the distinguisher) will find another input $\vec{b}' \neq \vec{b}$ and read sender's inputs according to $\vec{b}'$. Doing so would be a successful attack by Lemma 20.

3 Impossibility of Round-preserving OT Extension in the Random Oracle Model

In this section we formally state and prove our main impossibility result, Theorem 1. We start by formalizing the model for round-preserving OT extension.

OT extension is the task of using a limited number of "base OTs" to generate an increased number of OTs. The weakest possible form of OT extension is using n base OTs to construct $n+1$ OTs, but doing so is also sufficient as this can be repeated to get further extension (e.g., see [LZ13]). In our definition of OT extension, we model base OTs with an OT-hybrid functionality. This functionality can be seen as a trusted third party that receives the inputs of sender and receiver over a perfectly secure channel and sends to the receiver the output of the base OTs. The presence of an OT-hybrid functionality is often referred to as the OT-hybrid model [IKNP03].

Here, we are particularly interested in the notion of a *round-preserving* OT extension protocol. Intuitively, this is an OT extension which uses the same number of rounds as the base OTs that implement the OT-hybrid functionality. Given an r-round-preserving OT extension protocol E from n-OTs to $(n+1)$-OTs, one may then instantiate $\mathbb{OT}_n$ with a concrete r-round OT to obtain $(n+1)$-OT that also works in r rounds.

The following definition formalizes the hybrid model using which we model OT extension protocols that preserve the round complexity of the base OTs. We first describe the model, and then we will discuss some subtle aspects of it.

Definition 22 (Round-preserving OT extension). *A round-preserving OT extension protocol is a 2-party protocol with the following form.*

1. *$\mathcal{S}$ has input $\{x_i^0, x_i^1\}_{i\in[n+1]}$ and $\mathcal{R}$ has input $\vec{b} = (b_1, \ldots, b_{n+1})$.*
2. *Both of $\mathcal{S}, \mathcal{R}$ can query the random oracle $\mathbf{H}$ at any time.*
3. *$\mathcal{R}$ and $\mathcal{S}$ exchange $r = \text{poly}(\kappa)$ number of messages $t_1, \ldots, t_r$.*
4. *By the time $\mathcal{S}$ sends the final message t_r to $\mathcal{R}$, $\mathcal{S}$ has submitted its inputs $\{y_i^0, y_i^1\}_{i\in[n]}$ and $\mathcal{R}$ has submitted its input $\vec{c} = (c_1, \ldots, c_n)$ to $\mathbb{OT}_n$.*
5. *Right after $\mathcal{S}$ sends the final message to $\mathcal{R}$, $\mathcal{R}$ receives $\{y_i^{c_i}\}_{i\in[n]}$ from $\mathbb{OT}_n$.*
6. *$\mathcal{R}$ outputs, perhaps after more queries to $\mathbf{H}$, what is supposed to be $\{x_i^{b_i}\}_{i\in[n+1]}$.*

The completeness and semi-honest security of OT extension is defined based on the semi-honest security of k-OT (Definition 18) for $k = n + 1$.

When to submit inputs to hybrid $\mathbb{OT}_n$. We emphasize that the output from the OT-hybrid functionality is received only after the final message has been sent. This is the case because the OT-hybrid functionality in an r-round OT extension protocol is implemented using an r-round base OT protocol, which produces its output after receiving the final message. In this definition, the parties choose their inputs for $\mathbb{OT}_n$ at some points before the last message. Note that, "naturally" the inputs to a r-round OT functionality should be submitted at the beginning, but allowing the parties to choose their inputs to $\mathbb{OT}_n$ more flexibly only makes our impossibility result stronger.

In Definition 22, messages exchanged in an extension protocol are not allowed to depend on the intermediate messages of the base OT protocol. This is justified since these messages are simulatable. Moreover, without loss of generality, we assume that $\mathbb{OT}_n$ is never used in the "opposite" direction (with the sender acting as the receiver and the receiver as the sender), because then there would be not enough rounds for the output of $\mathbb{OT}_n$ affecting any message sent to the receiver, who is the only party with an output. Indeed, not surprisingly, the known protocols [WW06] for switching the sender/receiver roles of the OT require additional rounds. This role-switching is used in the OT extension of the IKNP protocol [IKNP03], which also requires one more round. In fact, our impossibility result shows that the result of [IKNP03] is round-optimal (though it is not round-preserving) among all black-box protocols for OT extension using symmetric-key primitives.

Based on Definition 22 above, we can now state Theorem 1 formally.

Theorem 23. *Let $(\mathcal{S}, \mathcal{R})$ be a* round-preserving *OT extension protocol (according to Definition 22) with security parameter κ using random oracle* $\mathbf{H}$ *as follows.*

1. *The $n \leq \text{poly}(\kappa)$ OTs modeled by $\mathbb{OT}_n$ are allowed to be* string *OTs.*
2. *$(\mathcal{S}, \mathcal{R})$ implement* bit *$(n+1)$-OT with $\lambda = \text{negl}(\kappa)$ completeness error.*
3. *Either of $(\mathcal{S}, \mathcal{R})$ ask at most $m = \text{poly}(\kappa)$ queries to the random oracle* $\mathbf{H}$.

Then the constructed $(n+1)$-OT cannot be (even semi-honest) secure for both of $\mathcal{S}$ or $\mathcal{R}$ against adversaries who can ask $\text{poly}(m \cdot n) \leq \text{poly}(\kappa)$ queries to $\mathbf{H}$.

In particular, either of $\mathcal{S}$ or $\mathcal{R}$ can execute the protocol honestly, then ask $\text{poly}(\kappa)$ more queries, and then break the (semi-honest) security of the constructed bit $(n+1)$-OT by advantage $\frac{1}{\text{poly}(n)} \geq \frac{1}{\text{poly}(\kappa)}$ according to either of the attacks described in Lemma 19 or Lemma 20.

The above theorem proves that for any round-preserving OT extension protocol, there is always a $\text{poly}(\kappa)$-query attack by one of the parties that succeeds in breaking the semi-honest security of the protocol with non-negligible advantage $1/\text{poly}(\kappa)$. In fact, we show how to break the security of such protocols even when the main inputs (but not those of the hybrid $\mathbb{OT}_n$) are chosen at random.

3.1 Proving Theorem 23

In the rest of this section, we prove Theorem 23 above.

Notation. First we clarify our notation used.

- $\vec{b} = (b_1, \ldots, b_{n+1}) \in \{0,1\}^{n+1}$ is $\mathcal{R}$'s own input, and it submits $\vec{c} = (c_1, \ldots, c_n) \in \{0,1\}^n$ as its input to $\mathbb{OT}_n$ during the execution of the protocol.
- $(x_i^0, x_i^1)_{i \in [n+1]}$ is $\mathcal{S}$'s input, and it submits $\{y_i^0, y_i^1\}_{i \in [n]}$ as its input to $\mathbb{OT}_n$.
- For $r \in \mathbb{N}$, $T = (t_1, \ldots, t_r)$ is the transcript of the protocol.
- γ is the output of $\mathbb{OT}_n$ that $\mathcal{R}$ receives after t_r is sent to $\mathcal{R}$.
- $V_{\mathcal{S}}$ and $V_{\mathcal{R}}$ denote, in order, the views of $\mathcal{S}$ and $\mathcal{R}$, where $V_{\mathcal{R}}$ only includes the receiver's view *before receiving* γ from $\mathbb{OT}_n$.

We will show that by asking poly(κ) queries after executing the protocol honestly: either the sender can distinguish the receiver's *uniformly random* input from an actual independent random string, which is an attack by Lemma 19, or the receiver can read both of sender's inputs for an index i with non-negligible probability[13]), which is an attack by Lemma 20.

We first define each party's attack and then will prove that one of them will succeed with non-negligible probability. Both attacks will make heavy use of the 'dependency learning' attack of Theorem 17. We will use that lemma for some sufficiently small parameter δ that will be chosen when we analyze the attacks.

Construction 24 (Sender's attack $\widehat{\mathcal{S}}$). *Here $\widehat{\mathcal{S}}$ tries to distinguish between an independently sampled random string from $\{0,1\}^{n+1}$ and the actual input $\vec{b}$ (chosen at random and then) used by the receiver, based on the transcript T of the (honestly executed protocol) and its knowledge about the random oracle* $\mathbf{H}$.

1. *$\widehat{\mathcal{S}}$ chooses its own input $x = (x_i^0, x_i^1)_{i \in [n+1]}$ uniformly at random.*
2. *After the last message t_r is sent, $\widehat{\mathcal{S}}$ runs the Eve algorithm of Theorem 17 over the full transcript $T = (t_1, \ldots, t_r)$ for sufficiently small δ (to be chosen later) over the following modified version $(\mathcal{S}, \mathcal{R}_1)$ of the original protocol, to learn a set of oracle query-answer pairs $P_{\mathcal{E}}$.*
   - *$\mathcal{S}$ and $\mathcal{R}$ choose their inputs uniformly at random.*
   - *$\mathcal{R}_1$ stops right after the last message is sent (right before γ is delivered).*

   Note that even though $\mathcal{S}, \mathcal{R}_1$ submit some inputs to $\mathbb{OT}_n$, because no outputs are received by $\mathcal{R}_1$ and because all inputs are chosen at random, this is a randomized "inputless" protocol between $\mathcal{S}, \mathcal{R}_1$ for which we can indeed run the attacker Eve of Theorem 17.
3. *$\widehat{\mathcal{S}}$ then considers the distribution $(\mathbf{V}_{\mathcal{R}} \mid \mathbf{V}_{\mathcal{E}} = V_{\mathcal{E}})$ conditioned on the obtained Eve view $V_{\mathcal{E}} = (T, P_{\mathcal{E}})$, where T is the transcript and $P_{\mathcal{E}}$ are the oracle query-answer pairs learned by Eve.[14] Then, given an input from $\{0,1\}^{n+1}$, $\widehat{\mathcal{S}}$ tries*

[13] One can always guess a bit with probability 1/2, however, if the receiver specifies explicitly that she has found both inputs of the sender correctly with non-negligible probability, this is a violation of security and cannot be simulated efficiently in the ideal world. Our attacking receiver will indeed specify when she succeeds.

[14] More formally, the distinguishing task is done by the distinguisher, and thus $\widehat{\mathcal{S}}$ tries to obtain a view that is not simulatable. However, for simplicity of the exposition, we combine the semi-honest attacker and the distinguisher.

to use the maximum-likelihood method to distinguish receiver's input $\vec{b}$ *from a random string. Namely, given a string* β, $\widehat{\mathcal{S}}$ *outputs* 1 *if* $\Pr[\vec{\mathbf{b}} = \beta \mid V_{\mathcal{E}}] > 2^{-(n+1)}$, *where* $\vec{\mathbf{b}}$ *is the random variable denoting the receiver's input* $\vec{b}$, *and it outputs* 0 *otherwise. In other words,* $\widehat{\mathcal{S}}$, *outputs* 1 *if the given* β, *from the eyes of Eve, is* more likely *to be the actual receiver's input* $\vec{b}$ *than being sampled from* $\mathbf{U}_{n+1}$ *independently.*

An interesting thing about the above attack is that here the sender somehow chooses to 'forget' about its own view and only considers Eve's view (which still includes the transcript), but doing this is always possible since Eve's view is part of the attacking sender's view.

Construction 25 (Receiver's attack $\widehat{\mathcal{R}}$). $\widehat{\mathcal{R}}$ *follows the protocol honestly, denoted by the honest execution* $\mathcal{R}$, *but its goal is to obtain also* another *output* not *corresponding to its original input* $\vec{b}$. *(Doing this would establish an attack by Lemma 20.) In order to get to this goal, in addition to executing* $\mathcal{R}$ *honestly to obtain the 'default' output* $(x_i^{b_i})_{i\in[n+1]}$ *with respect to* $\vec{b}$, *the cheating receiver* $\widehat{\mathcal{R}}$ *also runs the following algorithm, denoted by* $\mathcal{R}'$, *that tries to find the output with respect to some other input* $\vec{b}' \neq \vec{b}$. $\mathcal{R}'$ *will try to pick* $\vec{b}' \neq \vec{b}$ *in a way that it remains consistent with the transcript* T *as well as the received OT-hybrid output* γ *(by enforcing the consistency with the OT-hybrid input* $\vec{c}$*), so that the obtained output is correct with respect to* $\vec{b}'$. *Formally, the algorithm* $\mathcal{R}'$ *is equal to* $\mathcal{R}$ *until the last message* t_r *is sent from* $\mathcal{S}$ *(i.e., we refer to this partial execution as* $\mathcal{R}_1$*), but then* $\mathcal{R}'$ *(as part of the attack* $\widehat{\mathcal{R}}$*) diverges from* $\mathcal{R}$*'s execution as follows.*

1. *After the last message* t_r *is sent by the sender* $\mathcal{S}$, *the cheating receiver* $\widehat{\mathcal{R}}$ *runs the Eve algorithm of Theorem 17 over the* same *input-less protocol* $(\mathcal{S}, \mathcal{R}_1)$ *used by* $\widehat{\mathcal{S}}$ *in Construction 24 (where inputs are chosen at random and the protocol ends when* t_r *is sent) to obtain Eve's view* $V_{\mathcal{E}} = (T, P_{\mathcal{E}})$ *for the same* δ *used by* $\widehat{\mathcal{S}}$ *in Construction 24.*
2. $\widehat{\mathcal{R}}$ *then samples from the distribution* $V'_{\mathcal{R}} \leftarrow (\mathbf{V}_{\mathcal{R}} \mid \mathbf{V}_{\mathcal{E}} = V_{\mathcal{E}}, \vec{\mathbf{c}} = \vec{c})$ *where* $\mathbf{V}_{\mathcal{R}}$ *denotes the random variable encoding the view of the inputless protocol* $(\mathcal{S}, \mathcal{R}_1)$ *over which the Eve algorithm is executed. Now,* $\widehat{\mathcal{R}}$ *interprets* $V'_{\mathcal{R}}$ *as the (partial) execution of* $\mathcal{R}'$ *till* t_r *is sent (i.e., only reflecting the* $\mathcal{R}_1$ *part), and it continues executing* $\mathcal{R}'$ *to a full execution of the receiver as follows.*
3. *Upon receiving* γ *from* $\mathbb{OT}_n$, $\mathcal{R}'$ *continues the protocol (as the receiver) using the partial view* $V'_{\mathcal{R}}$ *and* γ *as follows. Note that in order to finish the execution, all we have to do is to describe how each oracle query* q *made by the (remaining execution of)* $\mathcal{R}'$ *is answered. Let* $\mathcal{L}$ *be an empty set and then update it inductively, whenever a new query* q *is asked by* $\mathcal{R}'$, *as follows.*
   (a) *If* $q \in \mathcal{Q}(V'_{\mathcal{R}})$, *then use the corresponding answer specified in* $V'_{\mathcal{R}}$.
   (b) *Otherwise, if* $(q, a) \in \mathcal{L}$ *for some* a, *use* a *as answer to* q.
   (c) *Otherwise, if* $q \in \mathcal{Q}(V_{\mathcal{R}}) \setminus (\mathcal{Q}(V_{\mathcal{E}}) \cup \mathcal{Q}(V'_{\mathcal{R}}))$,[15] *pick a random answer* a *for query* q, *and also add* (q, a) *to* $\mathcal{L}$ *for the future.*

[15] To have $q \in \mathcal{Q}(V_{\mathcal{R}}) \setminus (\mathcal{Q}(V_{\mathcal{E}}) \cup \mathcal{Q}(V'_{\mathcal{R}}))$ means that q is not asked by Eve and it is not in the fake receiver's view $V'_{\mathcal{R}}$ (for partial execution $\mathcal{R}_1$), but q is in the *honest* original execution of $\mathcal{R}_1$.

(d) Otherwise, ask q from the real random oracle $\mathbf{H}$.
When the emulation of $\mathcal{R}'$ is completed, output whatever is obtained as the output of $\mathcal{R}'$ corresponding to the input $\vec{b}'$ described in $V'_{\mathcal{R}}$.

Now we show that at least one of the attacks $\widehat{\mathcal{S}}, \widehat{\mathcal{R}}$ above succeeds.

Claim 1. *Either the attacking sender $\widehat{\mathcal{S}}$ of Construction 24 will distinguish $\vec{\mathbf{b}}$ from $\mathbf{U}_{n+1}$ with advantage at least $\Omega(1/n)$, or the attacking receiver $\widehat{\mathcal{R}}$ of Construction 25 can obtain correct outputs corresponding to its random $\vec{b}$ as well as some $\vec{b}' \neq \vec{b}$ with probability at least $\Omega(1/n^2) - O(\lambda + \delta)$, where λ is the completeness error of the protocol and δ is the selected Eve parameter.*

Proving Theorem 23 using Claim 1. Because $\lambda = \mathrm{negl}(\kappa) < o(1/n^2)$, by choosing $\delta = o(1/n^2)$ in Claim 1, either the attacking sender of Construction 24 will break the security by Lemma 19, or the attacking receiver of Construction 25 succeeds in breaking the security with advantage $\Omega(1/n^2)$ (by asking $\mathrm{poly}(\kappa)$ oracle queries) by Lemma 20. In the following, we will prove Claim 1.

3.2 Proof of Claim 1

In this subsection, we will prove Claim 1. Let $\varepsilon = 1/(1000n + 1000)$.

When $\widehat{\mathcal{S}}$ succeeds. If it holds that $\mathsf{SD}_{\mathbf{V}_\mathcal{E}}(\vec{\mathbf{b}}, \mathbf{U}_{n+1}) \geq \varepsilon$, then because the attacking $\widehat{\mathcal{S}}$ of Construction 24 is indeed using the canonical distinguisher of Proposition 5 (i.e., the maximum likelihood predicate), by Lemma 3 and Proposition 5, $\widehat{\mathcal{S}}$ will be able to ε-distinguish the true randomly chosen input $\vec{\mathbf{b}}$ of the receiver $\mathcal{R}$ from a uniform string $\mathbf{U}_{n+1}$ by advantage at least ε. Therefore, by Lemma 19, $\widehat{\mathcal{R}}$ succeeds in breaking the security with non-negligible advantage ε.

So, in what follows we assume that $\widehat{\mathcal{S}}$ does not succeed, and based on this we show that $\widehat{\mathcal{R}}$ does indeed succeed in its attack.

When $\widehat{\mathcal{R}}$ succeeds. In what follows we always assume

$$\mathop{\mathbb{E}}_{V_\mathcal{E} \leftarrow \mathbf{V}_\mathcal{E}} \mathsf{SD}((\vec{\mathbf{b}} \mid V_\mathcal{E}), \mathbf{U}_{n+1}) = \mathsf{SD}_{\mathbf{V}_\mathcal{E}}(\vec{\mathbf{b}}, \mathbf{U}_{n+1}) < \varepsilon = \frac{1}{1000n + 1000} \tag{3}$$

and we will show, using Inequality (3) and Lemma 20, that the receiver's attacker $\widehat{\mathcal{R}}$ will succeed with the non-negligible probability. First note that by just continuing the protocol honestly, the receiver will indeed find the right output for its sampled $\vec{b}$ with probability at least $1 - \lambda$ where λ is the completeness error. So all we have to prove is that with probability $\Omega(1/n^2) - O(\delta) - \lambda$, it will simultaneously hold that (1) $\vec{b}' \neq \vec{b}$ and (2) the receiver $\mathcal{R}'$ gets the output corresponding to $\vec{b}'$ (and sender's actual input x). To prove this, it will suffice to prove the following two statements:

- $\Pr[\vec{\mathbf{b}}' \neq \vec{\mathbf{b}}] \geq \Omega(1/n^2)$ where $\vec{\mathbf{b}}$ and $\vec{\mathbf{b}}'$ are the random variables denoting the original and the fake inputs of $\widehat{\mathcal{R}}$.
- The receiver will get the right answer for $\vec{b}'$ with probability $1 - O(\delta) - \lambda$.

Then, by a union bound, we can conclude that the $\widehat{\mathcal{R}}$ will indeed manage to launch a successful attack with probability $\Omega(1/n^2) - O(\delta + \lambda)$. In the following we will formalize and prove the above two claims in forms of Claims 2 and 3.

Claim 2. *If Inequality (3) holds, then* $\Pr[\vec{\mathbf{b}}' \neq \vec{\mathbf{b}}] \geq \Omega(1/n^2)$ *where the probability is over the randomness of the sender* $\mathcal{S}$*, cheating receiver* $\widehat{\mathcal{R}}$*, and* $\mathbf{H}$*.*

Proof. By sampling the components of the system 'in reverse', we can imagine that first $(T, P_{\mathcal{E}}) = V_{\mathcal{E}} \leftarrow \mathbf{V}_{\mathcal{E}}$ is sampled from its corresponding marginal distribution, then $\vec{c} \leftarrow (\vec{\mathbf{c}} \mid V_{\mathcal{E}})$ is sampled, then $(V_{\mathcal{S}}, V_{\mathcal{R}}) \leftarrow ((\mathbf{V}_{\mathcal{S}}, \mathbf{V}_{\mathcal{R}}) \mid V_{\mathcal{E}}, \vec{c})$, and finally $V'_{\mathcal{R}} \leftarrow (\mathbf{V}_{\mathcal{R}} \mid V_{\mathcal{E}}, \vec{c})$ are sampled, each conditioned on previously sampled components of the system. We will rely on this order of sampling in our arguments below. However, we can ignore the sampling of $V_{\mathcal{S}}$, when we want to compare the components $V_{\mathcal{R}}, V'_{\mathcal{R}}$ and the relation between $\vec{b}, \vec{b}'$. Thus, we can think of $V_{\mathcal{R}}, V'_{\mathcal{R}}$ as two independent samples from the same distribution $(\mathbf{V}_{\mathcal{R}} \mid V_{\mathcal{E}}, \vec{c})$. Consequently, $\vec{b}, \vec{b}'$ are also two independent samples from $(\vec{\mathbf{b}} \mid V_{\mathcal{E}}, \vec{c})$.

By Inequality (3) and an averaging argument over the sampled $V_{\mathcal{E}} \leftarrow \mathbf{V}_{\mathcal{E}}$, with probability at least $1 - 1/10$ over the choice of $V_{\mathcal{E}} \leftarrow \mathbf{V}_{\mathcal{E}}$, it holds that $\mathsf{SD}_{V_{\mathcal{E}}}(\vec{\mathbf{b}}, \mathbf{U}_{n+1}) < \varepsilon' = \frac{1}{100n+100}$. We call such $V_{\mathcal{E}}$ a 'good' sample. For any good $V_{\mathcal{E}}$, using Lemma 14 it holds that $\mathrm{H}(\vec{\mathbf{b}} \mid V_{\mathcal{E}}) \geq (1-\varepsilon') \cdot (n+1) - \mathrm{H}(\varepsilon')$, and since the length of $\vec{c}$ is n, by further conditioning on random variable $\vec{\mathbf{c}}$ we have:

$$\mathrm{H}(\vec{\mathbf{b}} \mid V_{\mathcal{E}}, \vec{\mathbf{c}}) \geq (1-\varepsilon') \cdot (n+1) - n - \mathrm{H}(\varepsilon') = 1 - \varepsilon' \cdot (n+1) - \mathrm{H}(\varepsilon') \geq 9/10$$

where the last inequality follows from $\varepsilon' \leq 1/200$, and $\mathrm{H}(1/200) < 1/20$. Therefore, by Lemma 15 (using $\mathbf{X} = \vec{\mathbf{b}}, \mathbf{Y} = (V_{\mathcal{E}}, \vec{\mathbf{c}})$) we conclude that the event $\vec{b} \neq \vec{b}'$ happens with probability at least $\Omega(1/n^2)$. Finally, since $V_{\mathcal{E}}$ is a good sample with probability $\Omega(1)$, we can still conclude that $\vec{b} \neq \vec{b}'$ happens with probability at least $\Omega(1/n^2)$, finishing the proof of Claim 2. □

Claim 3. *If Inequality (3) holds, then with probability* $1 - \lambda - O(\delta)$ *(over the randomness of the honest sender* $\mathcal{S}$*, the cheating receiver* $\widehat{\mathcal{R}}$*, and the oracle* $\mathbf{H}$*) the cheating receiver* $\mathcal{R}'$ *obtains the correct answer for* $\vec{b}'$ *(i.e.,* $x_1^{b'_1}, \ldots, x_{n+1}^{b'_{n+1}}$*).*

Proof. We want to argue that the full sampled view of the fake receiver $\mathcal{R}'$ (including $V'_{\mathcal{E}}$ followed by the computation as described in the fake execution $\mathcal{R}'$ as part of $\widehat{\mathcal{R}}$) will be statistically close to an actual honest execution of the protocol (i.e., a full execution of $\mathcal{R}$ over random input). For this goal, we define and compare the outcomes of the following experiments. For clarity, and because we use the same names for random variables in different experiments, we might use $\langle \mathbf{X} \rangle_{\mathsf{Z}}$ to emphasize that we are referring to $\mathbf{X}$ in the experiment Z.

Outputs of experiments. The output of the experiments below are vectors with six components. Therefore, the order of the elements in these vectors is very important, and e.g., if we change their order, that changes the actual output.

– Real **experiment.** This experiment outputs $\langle V_{\mathcal{E}}, \vec{c}, V_{\mathcal{S}}, V_{\mathcal{R}}, V'_{\mathcal{R}}, P' \rangle_{\mathsf{Real}}$ where $V_{\mathcal{E}}$ is Eve's view, $V_{\mathcal{S}}$ is sender's view, $V_{\mathcal{R}}$ is receiver's honestly generated view

(till last message is sent), $V'_{\mathcal{R}}$ is the sampled fake view of $\mathcal{R}'$ only till last message is sent ($V_{\mathcal{R}}, V'_{\mathcal{R}}$ are both part of the view of $\widehat{\mathcal{R}}$), and P' is the set of query-answer pairs that $\mathcal{R}'$ generates after γ (i.e., the message coming from $\mathbb{OT}_n$ after the last message is sent) is sent (some of which are answered using real oracle $\mathbf{H}$ and the rest are emulated using random coin tosses).

– Ideal **experiment.** In this experiment, we also sample a fake receiver's view $V'_{\mathcal{R}}$ the same as in the Real experiment, but then there is no real attack happening and we use the real oracle $\mathbf{H}$ to obtain the query-answer pairs P to finish the computation of $\mathcal{R}$ (which is the original honest execution) using the honest partial view $V_{\mathcal{R}}$. At the end we output $\langle V_{\mathcal{E}}, \vec{c}, V_{\mathcal{S}}, V'_{\mathcal{R}}, V_{\mathcal{R}}, P\rangle_{\mathsf{Ideal}}$. Other the change from P' to P, note the crucial that we are switching the locations of the real and fake receiver views $V_{\mathcal{R}}, V'_{\mathcal{R}}$ in the output vector.

Remark 26 (Why not containing γ explicitly in outputs of experiments?). Note that even though γ is not included explicitly in the output of the experiment, it is implicitly there, because γ is a deterministic function of $V_{\mathcal{S}}$ and $\vec{c}$. In particular, because both $V_{\mathcal{R}}, V'_{\mathcal{R}}$ are consistent with $\vec{c}$, they can both lead to correct answers for sender inputs $\vec{b}, \vec{b}'$. In addition, if we *did* include γ in the outputs of the experiments, it would *not* change their statistical distance.

Remark 27 (Why outputting $V_{\mathcal{R}}, V'_{\mathcal{R}}$ both?). Note that our final goal is to show that the fake view $V'_{\mathcal{R}}$ in the Real experiment 'behaves closely' to the actual honest view $V_{\mathcal{R}}$ in the Ideal experiment. So, one might wonder why we include both in the analysis of the experiments. The reason is that the honest and fake views $V_{\mathcal{R}}, V'_{\mathcal{R}}$ in the Real experiment are *not* independent of each other, so if we want to continue the execution of $V'_{\mathcal{R}}$ in the Real experiment to finish the view of $\mathcal{R}'$ (to get the output corresponding to the fake input $\vec{b}'$) we need to be aware of the oracle queries whose answers are already fixed as part of the view of $V_{\mathcal{R}}$. The reason is that we have to answer (some of them) intentionally at random, because corresponding queries in the Ideal experiment are being asked *for the first time.* In order to answer such queries the same way that they are answered in the Ideal experiment, we need to keep track of them in both experiments and avoid some 'bad' events that prevent us from answering from the right distribution.

To prove Claim 3, it is enough to prove $O(\delta)$-closeness of experiments. If we show that the outputs of the two experiments are $O(\delta)$ (statistically) close, then by the completeness error in the ideal word, which is at most λ, we could conclude that the completeness error in the real world over the randomness of $\langle \mathbf{V}_{\mathcal{E}}, \mathbf{V}_{\mathcal{S}}, \mathbf{V}_{\mathcal{R}}, \mathbf{P}\rangle_{\mathsf{Ideal}}$ is at most $\lambda + O(\delta)$, where the completeness now means that the fake view of the attacking receiver is obtaining the right answer!

To prove that the two experiments' outputs are $O(\delta)$ close, we do the following:

1. We first prove that $\langle \mathbf{V}_{\mathcal{E}}, \vec{\mathbf{c}}, \mathbf{V}_{\mathcal{S}}, \mathbf{V}_{\mathcal{R}}, \mathbf{V}'_{\mathcal{R}}\rangle_{\mathsf{Real}} \approx_{O(\delta)} \langle \mathbf{V}_{\mathcal{E}}, \vec{\mathbf{c}}, \mathbf{V}_{\mathcal{S}}, \mathbf{V}'_{\mathcal{R}}, \mathbf{V}_{\mathcal{R}}\rangle_{\mathsf{Ideal}}$.
2. Then we show that $\Pr[\langle \mathbf{V}_{\mathcal{E}}, \vec{\mathbf{c}}, \mathbf{V}_{\mathcal{S}}, \mathbf{V}_{\mathcal{R}}, \mathbf{V}'_{\mathcal{R}}\rangle_{\mathsf{Real}} \in \mathsf{B}] \leq \delta$ for some 'bad' event B. (Recall that an event in this work is simply a set, and the same set can be used as an event for different random variables, as long as their samples are

inside a universe where B is also defined.) Intuitively, the bad event captures the event fact that an 'intersection' query exists between the views of the sender and the receiver that is missed by Eve. Indeed, we could also bound the probability of the same event B in the Ideal experiment, however we simply bound it in Real and that turns out to be enough.

3. Finally, we show that as long as the event B does not happen over the sampled $\alpha = \langle V_{\mathcal{E}}, \vec{c}, V_{\mathcal{S}}, V_{\mathcal{R}}, V'_{\mathcal{R}} \rangle_{\mathsf{Real}} \leftarrow \langle \mathbf{V}_{\mathcal{E}}, \vec{\mathbf{c}}, \mathbf{V}_{\mathcal{S}}, \mathbf{V}_{\mathcal{R}}, \mathbf{V}'_{\mathcal{R}} \rangle_{\mathsf{Real}}$ (i.e., $\alpha \notin \mathsf{B}$) and if the sampled prefixes of the outputs are equal $\alpha = \langle V_{\mathcal{E}}, \vec{c}, V_{\mathcal{S}}, V'_{\mathcal{R}}, V_{\mathcal{R}} \rangle_{\mathsf{Ideal}} = \langle V_{\mathcal{E}}, \vec{c}, V_{\mathcal{S}}, V_{\mathcal{R}}, V'_{\mathcal{R}} \rangle_{\mathsf{Real}}$, then the corresponding distributions

$$(\langle \mathbf{P} \rangle_{\mathsf{Ideal}} \mid \langle V_{\mathcal{E}}, \vec{c}, V_{\mathcal{S}}, V'_{\mathcal{R}}, V_{\mathcal{R}} \rangle_{\mathsf{Ideal}}) \equiv (\langle \mathbf{P}' \rangle_{\mathsf{Real}} \mid \langle V_{\mathcal{E}}, \vec{c}, V_{\mathcal{S}}, V_{\mathcal{R}}, V'_{\mathcal{R}} \rangle_{\mathsf{Real}})$$

will be identically distributed.

If we prove the above 3 claims, the $O(\delta)$ closeness of the experiments' outputs will follow from Lemma 11, which will finish the proof of Claim 3. To apply Lemma 11, we let $\mathbf{X}_1 = \langle \mathbf{V}_{\mathcal{E}}, \vec{\mathbf{c}}, \mathbf{V}_{\mathcal{S}}, \mathbf{V}_{\mathcal{R}}, \mathbf{V}'_{\mathcal{R}} \rangle_{\mathsf{Real}}, \mathbf{X}_2 = \langle \mathbf{P}' \rangle_{\mathsf{Real}}, \mathbf{X}'_1 = \langle \mathbf{V}_{\mathcal{E}}, \vec{\mathbf{c}}, \mathbf{V}_{\mathcal{S}}, \mathbf{V}'_{\mathcal{R}}, \mathbf{V}_{\mathcal{R}} \rangle_{\mathsf{Ideal}}, \mathbf{X}'_2 = \langle \mathbf{P} \rangle_{\mathsf{Ideal}}$. We will prove the above 3 items through Claims 4, 5 and 6 below.

Claim 4. $\langle \mathbf{V}_{\mathcal{E}}, \vec{\mathbf{c}}, \mathbf{V}_{\mathcal{S}}, \mathbf{V}_{\mathcal{R}}, \mathbf{V}'_{\mathcal{R}} \rangle_{\mathsf{Real}} \approx_{O(\delta)} \langle \mathbf{V}_{\mathcal{E}}, \vec{\mathbf{c}}, \mathbf{V}_{\mathcal{S}}, \mathbf{V}'_{\mathcal{R}}, \mathbf{V}_{\mathcal{R}} \rangle_{\mathsf{Ideal}}$.

Proof. By Part 1 of Theorem 17 it holds that in the real world:

$$\mathop{\mathbb{E}}_{(V_{\mathcal{E}}) \leftarrow (\mathbf{V}_{\mathcal{E}}, \vec{\mathbf{c}})} \mathsf{MutDep}((\mathbf{V}_{\mathcal{S}}, \mathbf{V}_{\mathcal{R}})_{\mathsf{Real}} \mid V_{\mathcal{E}}) \leq \delta.$$

By averaging over $V_{\mathcal{E}} \leftarrow \mathbf{V}_{\mathcal{E}}$ and then using Lemma 9 (and letting $\mathbf{C} := \vec{\mathbf{c}}, \mathbf{B} := \mathbf{V}_{\mathcal{R}}, \mathbf{A} := \mathbf{V}_{\mathcal{S}}$) and noting that $\vec{c}$ is only a function of $V_{\mathcal{R}}$, it holds that

$$\mathop{\mathbb{E}}_{(V_{\mathcal{E}}, \vec{c}) \leftarrow (\mathbf{V}_{\mathcal{E}}, \vec{\mathbf{c}})} \mathsf{MutDep}((\mathbf{V}_{\mathcal{S}}, \mathbf{V}_{\mathcal{R}})_{\mathsf{Real}} \mid V_{\mathcal{E}}, \vec{c}) \leq 2\delta.$$

For a fixed $(V_{\mathcal{E}}, \vec{c}) \leftarrow (\mathbf{V}_{\mathcal{E}}, \vec{\mathbf{c}})$, we can use Lemma 8 (by letting $\mathbf{X} \equiv (\mathbf{V}_{\mathcal{S}} \mid V_{\mathcal{E}}, \vec{c})$ and $\mathbf{Y} \equiv (\mathbf{V}_{\mathcal{R}} \mid V_{\mathcal{E}}, \vec{c})$) and then average over $(V_{\mathcal{E}}, \vec{c}) \leftarrow (\mathbf{V}_{\mathcal{E}}, \vec{\mathbf{c}})$ to conclude

$$\mathop{\mathbb{E}}_{(V_{\mathcal{E}}, \vec{c}) \leftarrow (\mathbf{V}_{\mathcal{E}}, \vec{\mathbf{c}})} \mathsf{SD}((\langle \mathbf{V}_{\mathcal{S}}, \mathbf{V}_{\mathcal{R}}, \mathbf{V}'_{\mathcal{R}} \rangle_{\mathsf{Real}} \mid V_{\mathcal{E}}, \vec{c}), (\langle \mathbf{V}_{\mathcal{S}}, \mathbf{V}'_{\mathcal{R}}, \mathbf{V}_{\mathcal{R}} \rangle_{\mathsf{Ideal}} \mid V_{\mathcal{E}}, \vec{c})) \leq 4\delta.$$

Finally, by Proposition 5, the left side of the above inequality is the same as $\mathsf{SD}(\langle \mathbf{V}_{\mathcal{E}}, \vec{\mathbf{c}}, \mathbf{V}_{\mathcal{S}}, \mathbf{V}_{\mathcal{R}}, \mathbf{V}'_{\mathcal{R}} \rangle_{\mathsf{Real}}, \langle \mathbf{V}_{\mathcal{E}}, \vec{\mathbf{c}}, \mathbf{V}_{\mathcal{S}}, \mathbf{V}'_{\mathcal{R}}, \mathbf{V}_{\mathcal{R}} \rangle_{\mathsf{Ideal}})$, finishing the proof. □

In the definition below, roughly speaking, the 'bad' event B contains possible outputs of the experiments for which some intersection queries exist between the views of the sender $\mathcal{S}$ and the receiver $\mathcal{R}$ that are missed by the Eve algorithm.

Definition 28 (The bad event B). *Let U be a 'universe' containing all possible outputs of the two experiments (and maybe more elements) defined as follows:*

$$\{\langle z_1, \ldots z_5 \rangle \mid z_1 \in \mathrm{Supp}(\mathbf{V}_{\mathcal{E}}), z_2 \in \mathrm{Supp}(\vec{\mathbf{c}}), z_3 \in \mathrm{Supp}(\mathbf{V}_{\mathcal{S}}), z_4, z_5 \in \mathrm{Supp}(\mathbf{V}_{\mathcal{R}})\}.$$

Let the 'bad' event $\mathsf{B} \subseteq \mathsf{U}$ *be the set that:*

$$\mathsf{B} = \{\alpha = \langle z_1, z_2, z_3, z_4, z_5\rangle \mid \alpha \in \mathsf{U}, \mathcal{Q}(z_4) \cap \mathcal{Q}(z_3) \not\subseteq \mathcal{Q}(z_1)\}$$

Namely, if we interpret z_1, z_3, z_4 *as views of oracle-aided algorithms and extract their queries, it holds that* $\mathcal{Q}(z_4) \cap \mathcal{Q}(z_3) \not\subseteq \mathcal{Q}(z_1)$.

The following claim implies that with high probability, a sample from the output of the Real experiment does not fall into B. (In other words, the property by which B is defined, does not hold over the sampled output).

Claim 5. $\Pr[\langle \mathbf{V}_{\mathcal{E}}, \vec{\mathbf{c}}, \mathbf{V}_{\mathcal{S}}, \mathbf{V}_{\mathcal{R}}, \mathbf{V}'_{\mathcal{R}}\rangle_{\mathsf{Real}} \in \mathsf{B}] \leq \delta$.

Proof. The claim directly follows from the second property of Eve's algorithm (i.e., Part 2 in Theorem 17). Namely, a sample

$$\alpha = \langle z_1, z_2, z_3, z_4, z_5\rangle \leftarrow \langle \mathbf{V}_{\mathcal{E}}, \vec{\mathbf{c}}, \mathbf{V}_{\mathcal{S}}, \mathbf{V}_{\mathcal{R}}, \mathbf{V}'_{\mathcal{R}}\rangle_{\mathsf{Real}}$$

will have components corresponding to $z_1 = V_{\mathcal{E}}, z_3 = V_{\mathcal{S}}, z_4 = V_{\mathcal{R}}$, and so by Part 2 of Theorem 17 we know that with probability at least $1 - \delta$ it holds that $\mathcal{Q}(V_{\mathcal{S}}) \cap \mathcal{Q}(V_{\mathcal{R}}) \subseteq \mathcal{Q}(V_{\mathcal{E}})$. Therefore, $\alpha \in \mathsf{B}$ would happen in Real experiment with probability at most δ. □

Remark 29 (Other possible choices for defining bad event B *and stating Claim 5).* One can also define an alternative version B' of the bad event B based on the modified condition $\mathcal{Q}(z_5) \cap \mathcal{Q}(z_3) \not\subseteq \mathcal{Q}(z_1)$ (i.e., using z_5 instead of z_4), and one can also choose either of Real or Ideal experiments for bounding the probability of the bad event (B or B') by $O(\delta)$. This gives rise to four possible ways of defining the bad event and bounding it in an experiment. We note that all four cases above (i.e., both variations of the bad event B or B' in both of the Real and the Ideal) experiments can be proved to happen with probability at most $O(\delta)$. Furthermore, all of these four possible choices could be used (together with Lemma 11) for bounding the statistical distance of the output of experiments Real and Ideal by $O(\delta)$. In fact, once we show that statistical distance of the output of experiments Real and Ideal is $O(\delta)$, we can go back and derive all four combinations (of choosing the bad event from B or B' and stating Claim 5 in either of Real or Ideal experiments) to be true. Thus, basically all of these four choices are "equivalent" up to constant factors in the bound we get in Claim 5. Nonetheless, among these four choices, we found the choice of the bad event B according to Definition 28 and stating Claim 5 in the Real experiment to be the simplest choice to prove (using Theorem 17) and use for proving Claim 3 (by bounding the statistical distance of the outputs of experiments using Lemma 11).

Claim 6. *If samples* $\alpha = \langle V_{\mathcal{E}}, \vec{c}, V_{\mathcal{S}}, V'_{\mathcal{R}}, V_{\mathcal{R}}\rangle_{\mathsf{Ideal}} = \langle V_{\mathcal{E}}, \vec{c}, V_{\mathcal{S}}, V_{\mathcal{R}}, V'_{\mathcal{R}}\rangle_{\mathsf{Real}}$ *are equal, and if event* B *does not happen over the sample* α *(i.e.,* $\alpha \notin \mathsf{B}$*), then*

$$(\langle \mathbf{P}\rangle_{\mathsf{Ideal}} \mid \langle V_{\mathcal{E}}, \vec{c}, V_{\mathcal{S}}, V'_{\mathcal{R}}, V_{\mathcal{R}}\rangle_{\mathsf{Ideal}}) \equiv (\langle \mathbf{P}'\rangle_{\mathsf{Real}} \mid \langle V_{\mathcal{E}}, \vec{c}, V_{\mathcal{S}}, V_{\mathcal{R}}, V'_{\mathcal{R}}\rangle_{\mathsf{Real}}).$$

Proof. We show that conditioned on the same sample α being the prefix of the outputs of the two experiments, the random process that generates the last components $\langle \mathbf{P}' \rangle_{\mathsf{Real}}$ and $\langle \mathbf{P} \rangle_{\mathsf{Ideal}}$ are identically distributed in the two experiments.

After sampling $\alpha = \langle V_{\mathcal{E}}, \vec{c}, V_{\mathcal{S}}, V'_{\mathcal{R}}, V_{\mathcal{R}} \rangle_{\mathsf{Ideal}}$, every new query q will be answered as follows in Ideal: If q is already in $\mathcal{Q}(V_{\mathcal{E}}) \cup \mathcal{Q}(V_{\mathcal{S}}) \cup \mathcal{Q}(V_{\mathcal{R}})$ then the answer is already fixed and that answer will be used, otherwise q will be answered at random (by the random oracle $\mathbf{H}$). Since we are assuming $\langle V_{\mathcal{E}}, \vec{c}, V_{\mathcal{S}}, V'_{\mathcal{R}}, V_{\mathcal{R}} \rangle_{\mathsf{Ideal}} = \langle V_{\mathcal{E}}, \vec{c}, V_{\mathcal{S}}, V_{\mathcal{R}}, V'_{\mathcal{R}} \rangle_{\mathsf{Real}}$, we would like to prove that in the Real experiment, q is answered similarly. Indeed, we will prove that in the Real experiment, if q is already in $\langle \mathcal{Q}(V_{\mathcal{E}}) \cup \mathcal{Q}(V_{\mathcal{S}}) \cup \mathcal{Q}(V'_{\mathcal{R}}) \rangle_{\mathsf{Real}}$ then the fixed answer will be used, and otherwise q will be answered at random. We make the following case study in the Real experiment based on the algorithm of $\widehat{\mathcal{R}}$ from Construction 25. (In the second case below we make a crucial use of the fact that the event B has not happened over the current sample $\langle V_{\mathcal{E}}, \vec{c}, V_{\mathcal{S}}, V'_{\mathcal{R}}, V_{\mathcal{R}} \rangle_{\mathsf{Ideal}} = \langle V_{\mathcal{E}}, \vec{c}, V_{\mathcal{S}}, V_{\mathcal{R}}, V'_{\mathcal{R}} \rangle_{\mathsf{Real}}$.)

1. If $q \in \langle \mathcal{Q}(V'_{\mathcal{R}}) \rangle_{\mathsf{Real}}$, then $\widehat{\mathcal{R}}$ uses the answer stated in $V'_{\mathcal{R}}$. Otherwise:
2. if $q \in \langle \mathcal{Q}(V_{\mathcal{R}}) \setminus (\mathcal{Q}(V_{\mathcal{E}}) \cup \mathcal{Q}(V'_{\mathcal{R}})) \rangle_{\mathsf{Real}}$, $\widehat{\mathcal{R}}$ answers q at random (and keeps its answer in a list $\mathcal{L}$ to reuse in case of being asked again). In the ideal world, this query q would be part of the *fake* view $\langle V'_{\mathcal{R}} \rangle_{\mathsf{Ideal}}$ (recall the fake and real views are switched across the Real vs. Ideal experiments) which is *ignored* in the Ideal world when we generate $\langle P \rangle_{\mathsf{Ideal}}$, and so we have two cases:
   (a) If q is already in $\langle \mathcal{Q}(V_{\mathcal{S}}) \rangle_{\mathsf{Real}}$, it means that $\alpha \in \mathsf{B}$ for $\alpha = \langle V_{\mathcal{E}}, \vec{c}, V_{\mathcal{S}}, V'_{\mathcal{R}}, V_{\mathcal{R}} \rangle_{\mathsf{Ideal}} = \langle V_{\mathcal{E}}, \vec{c}, V_{\mathcal{S}}, V_{\mathcal{R}}, V'_{\mathcal{R}} \rangle_{\mathsf{Real}}$ which is not true.
   (b) Otherwise, $q \notin \langle \mathcal{Q}(V_{\mathcal{S}}) \rangle_{\mathsf{Real}} = \langle \mathcal{Q}(V'_{\mathcal{S}}) \rangle_{\mathsf{Ideal}}$, which means that q is a new query in the ideal world, and so it is answered at random, just like how it is answered in the real world by the attacker $\widehat{\mathcal{R}}$.
3. If above cases do not happen, but q is still part of $\langle \mathcal{Q}(V_{\mathcal{E}}) \cup \mathcal{Q}(V_{\mathcal{S}}) \rangle_{\mathsf{Real}}$, $\widehat{\mathcal{R}}$ would forward this query to be asked from the actual random oracle $\mathbf{H}$ which would also get the correct answer (i.e., the same answer stated in $V_{\mathcal{E}}$ or $V_{\mathcal{S}}$).

Therefore, in all cases q will be answered from the same distribution across the Real and Ideal experiments. This shows that the process of generating the last component of the output of these experiments is identically distributed. □

This finishes the proof of Claim 3. □

References

[AIR01] Aiello, W., Ishai, Y., Reingold, O.: Priced oblivious transfer: how to sell digital goods. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 119–135. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44987-6_8

[AJL+12] Asharov, G., Jain, A., López-Alt, A., Tromer, E., Vaikuntanathan, V., Wichs, D.: Multiparty computation with low communication, computation and interaction via threshold FHE. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 483–501. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29011-4_29

[Ald83] Aldous, D.: Random walks on finite groups and rapidly mixing Markov chains. In: Azéma, J., Yor, M. (eds.) Séminaire de Probabilités XVII 1981/82. LNM, vol. 986, pp. 243–297. Springer, Heidelberg (1983). https://doi.org/10.1007/BFb0068322

[ALSZ13] Asharov, G., Lindell, Y., Schneider, T., Zohner, M.: More efficient oblivious transfer and extensions for faster secure computation. In: Sadeghi, A.-R., Gligor, V.D., Yung, M. (eds.) ACM CCS 2013, Berlin, Germany, 4–8 November 2013, pp. 535–548. ACM Press (2013)

[ALSZ15] Asharov, G., Lindell, Y., Schneider, T., Zohner, M.: More efficient oblivious transfer extensions with security for malicious adversaries. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015, Part I. LNCS, vol. 9056, pp. 673–701. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46800-5_26

[BCR87] Brassard, G., Crepeau, C., Robert, J.-M.: All-or-nothing disclosure of secrets. In: Odlyzko, A.M. (ed.) CRYPTO 1986. LNCS, vol. 263, pp. 234–238. Springer, Heidelberg (1987). https://doi.org/10.1007/3-540-47721-7_17

[Bea96] Beaver, D.: Correlated pseudorandomness and the complexity of private computations. In: 28th ACM STOC, Philadephia, PA, USA, 22–24 May 1996, pp. 479–488. ACM Press (1996)

[BGI+14] Beimel, A., Gabizon, A., Ishai, Y., Kushilevitz, E., Meldgaard, S., Paskin-Cherniavsky, A.: Non-interactive secure multiparty computation. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014, Part II. LNCS, vol. 8617, pp. 387–404. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-44381-1_22

[BL17] Benhamouda, F., Lin, H.: k-round multiparty computation from k-round oblivious transfer via garbled interactive circuits. Cryptology ePrint Archive, Report 2017/1125 (2017). EUROCRYPT 2018

[BM17] Barak, B., Mahmoody, M.: Merkle's key agreement protocol is optimal: an $O(n^2)$ attack on any key agreement from random oracles. J. Cryptol. **30**(3), 699–734 (2017)

[BMG09] Barak, B., Mahmoody-Ghidary, M.: Merkle puzzles are optimal — an $O(n^2)$-query attack on any key exchange from a random oracle. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 374–390. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-03356-8_22

[BMR90] Beaver, D., Micali, S., Rogaway, P.: The round complexity of secure protocols (extended abstract). In: 22nd ACM STOC, Baltimore, MD, USA, 14–16 May 1990, pp. 503–513. ACM Press (1990)

[CPS08] Coron, J.-S., Patarin, J., Seurin, Y.: The random oracle model and the ideal cipher model are equivalent. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 1–20. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-85174-5_1

[DLMM11] Dachman-Soled, D., Lindell, Y., Mahmoody, M., Malkin, T.: On the black-box complexity of optimally-fair coin tossing. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 450–467. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-19571-6_27

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

[Fan68] Fano, R.M.: Transmission of Information. A Ststistical Theory of Communications. MIT Press, Cambridge (1968)

[FKN94] Feige, U., Kilian, J., Naor, M.: A minimal model for secure computation (extended abstract). In: 26th ACM STOC, Montréal, Québec, Canada, 23–25 May 1994, pp. 554–563. ACM Press (1994)

[GGHR14] Garg, S., Gentry, C., Halevi, S., Raykova, M.: Two-round secure MPC from indistinguishability obfuscation. In: Lindell, Y. (ed.) TCC 2014. LNCS, vol. 8349, pp. 74–94. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-54242-8_4

[GKLM12] Goyal, V., Kumar, V., Lokam, S., Mahmoody, M.: On black-box reductions between predicate encryption schemes. In: Cramer, R. (ed.) TCC 2012. LNCS, vol. 7194, pp. 440–457. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-28914-9_25

[GMMM17] Garg, S., Mahmoody, M., Masny, D., Meckler, I.: On the round complexity of OT extension. Cryptology ePrint Archive, Report 2017/1187 (2017). https://eprint.iacr.org/2017/1187

[GMPP16] Garg, S., Mukherjee, P., Pandey, O., Polychroniadou, A.: The exact round complexity of secure computation. In: Fischlin, M., Coron, J.-S. (eds.) EUROCRYPT 2016, Part II. LNCS, vol. 9666, pp. 448–476. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49896-5_16

[GMW87] Goldreich, O., Micali, S., Wigderson, A.: How to play any mental game or a completeness theorem for protocols with honest majority. In: Aho, A. (ed.) 19th ACM STOC, New York City, NY, USA, 25–27 May 1987, pp. 218–229. ACM Press (1987)

[GS17a] Garg, S., Srinivasan, A.: Garbled protocols and two-round MPC from bilinear maps. In: 58th FOCS, pp. 588–599. IEEE Computer Society Press (2017)

[GS17b] Garg, S., Srinivasan, A.: Two-round multiparty secure computation from minimal assumptions. Cryptology ePrint Archive, Report 2017/1156 (2017). EUROCRYPT 2018

[HK12] Halevi, S., Kalai, Y.T.: Smooth projective hashing and two-message oblivious transfer. J. Cryptol. **25**(1), 158–193 (2012)

[HKT11] Holenstein, T., Künzler, R., Tessaro, S.: The equivalence of the random oracle model and the ideal cipher model, revisited. In: Fortnow, L., Vadhan, S.P. (eds.) 43rd ACM STOC, San Jose, CA, USA, 6–8 June 2011, pp. 89–98. ACM Press (2011)

[HOZ16] Haitner, I., Omri, E., Zarosim, H.: Limits on the usefulness of random oracles. J. Cryptol. **29**(2), 283–335 (2016)

[IKM+13] Ishai, Y., Kushilevitz, E., Meldgaard, S., Orlandi, C., Paskin-Cherniavsky, A.: On the power of correlated randomness in secure computation. In: Sahai, A. (ed.) TCC 2013. LNCS, vol. 7785, pp. 600–620. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36594-2_34

[IKNP03] Ishai, Y., Kilian, J., Nissim, K., Petrank, E.: Extending oblivious transfers efficiently. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 145–161. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-45146-4_9

[IPS08] Ishai, Y., Prabhakaran, M., Sahai, A.: Founding cryptography on oblivious transfer – efficiently. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 572–591. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-85174-5_32

[IR89] Impagliazzo, R., Rudich, S.: Limits on the provable consequences of one-way permutations. In: Proceedings of the 21st Annual ACM Symposium on Theory of Computing (STOC), pp. 44–61. ACM Press (1989)

[Kil88] Kilian, J.: Founding cryptography on oblivious transfer. In: Proceedings of the 20th Annual ACM Symposium on Theory of Computing (STOC), pp. 20–31 (1988)

[KK13] Kolesnikov, V., Kumaresan, R.: Improved OT extension for transferring short secrets. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part II. LNCS, vol. 8043, pp. 54–70. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40084-1_4

[KO04] Katz, J., Ostrovsky, R.: Round-optimal secure two-party computation. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 335–354. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-28628-8_21

[Lin16] Lindell, Y.: How to simulate it - a tutorial on the simulation proof technique. Cryptology ePrint Archive, Report 2016/046 (2016)

[LZ13] Lindell, Y., Zarosim, H.: On the feasibility of extending oblivious transfer. In: Sahai, A. (ed.) TCC 2013. LNCS, vol. 7785, pp. 519–538. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36594-2_29

[MW16] Mukherjee, P., Wichs, D.: Two round multiparty computation via multi-key FHE. In: Fischlin, M., Coron, J.-S. (eds.) EUROCRYPT 2016, Part II. LNCS, vol. 9666, pp. 735–763. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49896-5_26

[NNOB12] Nielsen, J.B., Nordholt, P.S., Orlandi, C., Burra, S.S.: A new approach to practical active-secure two-party computation. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 681–700. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32009-5_40

[NP01] Naor, M., Pinkas, B.: Efficient oblivious transfer protocols. In: Kosaraju, S.R. (ed.) 12th SODA, Washington, DC, USA, 7–9 January 2001, pp. 448–457. ACM-SIAM (2001)

[ORS15] Ostrovsky, R., Richelson, S., Scafuro, A.: Round-optimal black-box two-party computation. In: Gennaro, R., Robshaw, M. (eds.) CRYPTO 2015, Part II. LNCS, vol. 9216, pp. 339–358. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48000-7_17

[PVW08] Peikert, C., Vaikuntanathan, V., Waters, B.: A framework for efficient and composable oblivious transfer. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 554–571. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-85174-5_31

[Rab81] Rabin, M.: How to exchange secrets by oblivious transfer. Technical report TR-81, Harvard Aiken Computation Laboratory (1981)

[Sho09] Shoup, V.: A Computational Introduction to Number Theory and Algebra. Cambridge University Press, Cambridge (2009)

[WW06] Wolf, S., Wullschleger, J.: Oblivious transfer is symmetric. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 222–232. Springer, Heidelberg (2006). https://doi.org/10.1007/11761679_14

[Yao82] Yao, A.C.-C.: Protocols for secure computations (extended abstract). In: 23rd FOCS, Chicago, Illinois, 3–5 November 1982, pp. 160–164. IEEE Computer Society Press (1982)