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
[2] University of Virginia

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 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 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].

⋆ University of California, Berkeley. Research supported in part from AFOSR YIP Award, DARPA/ARL SAFEWARE Award W911NF15C0210, AFOSR Award FA9550-15-1-0274, 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.

⋆⋆ University of Virginia, mohammad@virginia.edu. Supported by NSF CAREER award CCF-1350939 and University of Virginia's SEAS Research Innovation Award.

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

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] secure computation protocols. Ideally, we would like to construct OT extension protocols that can be used to reduce the number of public-key operations needed 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" cryptographic primitives?

Random oracle model (ROM) accurately captures the black-box use of such symmetric 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.[3]

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 "symmetric key" primitives, i.e. primitives which are implied by random oracle model, requires either an additional round compared to the base OTs or must make a non-black-box use of "symmetric key" primitives. This result also holds in the semi-honest setting or in the case of a setup model such as the common random string.[4] 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, as we observe in Section 5, a variant of Beaver's original protocol [Bea96] can indeed achieve OT extension in two rounds through a non-black-box use of the code of one-way functions.[5]

In our main impossibility result, we capture black-box use of one-way functions, or correlation-robust hash functions 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 it 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 the joint security of the m OTs.*

Theorem 1 holds even for an extension from n *string* OTs to $m = n + 1$ bit OTs. 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 Section 1.2 and provide the details in Section 4.

Our result is also tight with respect to the black-box use of the symmetric primitives captured by random oracles. Namely, we observe 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" OT, but this result will also require *non-black-box* use of the PRG. Beaver's original protocol only provided OT extension in which the receiver has no control over which input he receives.

[3] 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].

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

[5] Beaver's OT extension protocol [Bea96] obtains new extended OTs for *randomly selected* receiver inputs.

This notion of OT is often referred to as "random" OT. The known generic way of going from "random" OTs to "chosen" OTs will add another round [EGL85]. However, in Section 5 we demonstrate that a modification to Beaver's protocol leads to a direct way of getting "chosen" OTs in the same number of rounds.

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. Therefore our results imply that 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 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. (Note that there are only $\mathrm{poly}(\kappa)$ such queries.) In that 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. Finally, 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 our main impossibility result of Theorem 1. For a formal treatment see Section 4. In a nutshell, we first present an entropy-based, information theoretic attack for the plain model, where there is no oracle involved. We then extend our attack to the random oracle model, by making use of the 'dependency learner' of [IR89, BMG09, HOZ16, BM17] that allows us to 'approximate' arguments that apply in the plain model 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 construction.

Notation and simplifying assumptions. Here we define some basic notations and also state some simplifying assumptions, some of which are without loss of generality when we focus on round-preserving OT extensions and the rest are relaxed when proving the formal attack in Section 4. Here we focus on the case of extending n instances of OT, into m instances for some $m \gg n$.[6] Suppose $b = (b_1, \ldots, b_m) \in \{0,1\}^m$ are the choice bits of the receiver $\mathcal{R}$ and

[6] 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].

$x = \{x_i^0, x_i^1\}_{i\in[m]} \in \{0,1\}^{2m}$ are the pairs of bits[7] that the sender holds as its input. The receiver $\mathcal{R}$ wishes to obtain $\{x_i^{b_i}\}_{i\in[m]}$ as its output. 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 $\mathbb{OT}_n$. For simplicity, suppose $\mathcal{S}$ and $\mathcal{R}$ do not reverse roles when using $\mathbb{OT}_n$.[8] Let $c \in \{0,1\}^n$ be the bits that $\mathcal{R}$ submits to $\mathbb{OT}_n$ and $\{y_i^0, y_i^1\}_{i\in[n]}$ the input that $\mathcal{S}$ submits to $\mathbb{OT}_n$. Suppose for now that they submit their input before exchanging messages.[9] Suppose $\gamma = \{y_i^{c_i}\}_{i\in[n]}$ is vector of bits that $\mathcal{R}$ receives from $\mathbb{OT}_n$, and suppose that the hybrid $\mathbb{OT}_n$ delivers the output to $\mathcal{R}$ at some well-defined point during the protocol[10]. Suppose $T = (t_1, t_2, \dots)$ is the transcript of the protocol. We also assume for simplicity that the protocol has *perfect* completeness. For more standard notation, the reader might find the definitions at the beginning of Section 2 useful.

An information theoretic attack for no-oracle setting. Our starting point is an inefficient (information theoretic) attack on OT extension when there is no oracle involved. The fact that OT extension protocols (regardless of their round complexity) can *not* be information theoretically secure was already shown in the work of 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 b, x uniformly at random. 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.

– **Attacking sender $\widehat{\mathcal{S}}$.** Since $\widehat{\mathcal{S}}$ gets no output, in a secure protocol the random input $b \in \{0,1\}^m$ of the receiver shall remain indistinguishable from a uniform $\mathbf{U}_m$ in eyes of the receiver who knows the transcript T. (See Lemma 8 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 b (in case it is revealed to him) from a random uniform string $\mathbf{U}_m$ of length m.[11] Thus, if the distribution of (b, T) is ε-far from $(\mathbf{U}_m, T)$ for *non-negligible* ε, the protocol is not secure, because

[7] The general negative result holds even if the hybrid $\mathbb{OT}_n$ provides string OTs, but in this simplified exposition, we work with bit OTs.

[8] Indeed, when we focus on the round-preserving case, this assumption will become without loss of generality.

[9] In our final result, we allow them to submit their inputs at an arbitrary point during the interaction of messages.

[10] When we focus on round-preserving case, this output γ would be sent after the last message is sent from the sender to the receiver.

[11] Technically, this distinguishing task is left for the distinguisher of the simulator of the OT protocol, but here we simply see the attacker $\widehat{\mathcal{S}}$ as a combination of the attacking semi-honest sender and the distinguisher. See Lemma 8.

(on average over the value of the transcript T) the sender can distinguish b from $\mathbf{U}_m$ by advantage $\geq \varepsilon$.

– **Attacking receiver $\widehat{\mathcal{R}}$.** A natural attacking strategy $\widehat{\mathcal{R}}$ for the receiver is the following. After running the protocol honestly to get the output for the honestly chosen input b, the receiver tries to find also another input $b' \neq b$ together with its corresponding output $\{x_i^{b'_i}\}_{i\in[m]}$. If $\widehat{\mathcal{R}}$ could indeed do so, it would be an acceptable attack since in at least one of the locations $i \in [m]$ the receiver will read both of (x_i^0, x_i^1). (See Lemma 9 for a formalization.) By relying on the perfect completeness of the protocol,[12] all $\widehat{\mathcal{R}}$ needs to do is to find another fake view $V'_{\mathcal{R}}$ for himself such that: (1) $V'_{\mathcal{R}}$ contains $b' \neq b$ (2) $V'_{\mathcal{R}}$ is consistent with the transcript T, the input c given to $\mathbb{OT}_n$, and the output γ obtained from it.[13] If such $V'_{\mathcal{R}}$ could be sampled, it would violate the security of the sender.

One of $\widehat{\mathcal{S}}, \widehat{\mathcal{R}}$ succeeds: an entropy-based argument. If the described sender attacker $\widehat{\mathcal{S}}$ does not succeed with non-negligible advantage, it means that (b, T) is statistically close to $(\mathbf{U}_m, T)$, which in turn implies that (on average over T) conditioned on the transcript T, the receiver's input b has close to the full m bits of entropy.[14] (See Lemma 6 for formalization.) Therefore, if the malicious receiver $\widehat{\mathcal{R}}$ (*re-samples*) a fake view $V'_{\mathcal{R}}$ from the distribution $(\mathbf{V}_{\mathcal{R}} \mid T)$, after finishing the honest executing encoded in the view $\mathbf{V}_{\mathcal{R}}$, it will get a different $b' \neq b$ with some noticeable probability. (See Lemma 7 for a formalization.) However, as described above, the receiver attacker $\widehat{\mathcal{R}}$ also needs to condition its sampled view $V'_{\mathcal{R}} \leftarrow (\mathbf{V}'_{\mathcal{R}} \mid T, c, \gamma)$ 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 $b' \neq b$. It can be shown that if $m > |c| + |\gamma| = 2n$, then there is still enough entropy left in the sampled b' even after further conditioning on c, γ (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 information theoretic argument above for the no-oracle case breaks down completely when we move to the random-oracle-aided model of computation 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

[12] Our formal proof of Section 4 does not assume perfect completeness.

[13] If we *only* sample receiver's view till receiving γ, we do *not* have to condition on γ, as it is only a function of the sender's view and receiver's input to $\mathbb{OT}_n$, which is part of her view till $\mathbb{OT}_n$ output is delivered. We will rely on this point in the description and analysis of our final attack for the case of round-preserving OT extensions.

[14] In fact, this step of the proof is the reason behind our choice of working with Shannon entropy rather than other notions of entropy such as min-entropy, as we want distributions close to $\mathbf{U}_m$ to have almost full entropy – the latter property does not hold e.g., for min-entropy.

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 the sender leads to correct outputs for the new fake input b'. In fact, because we already know that OT extension in the random oracle model *is* possible [IKNP03], we know that the above issue is an inherent obstacle when we try to extend our attack to the ROM naively. However, we have not yet used the fact that we are aiming at attacks that succeed for *round-preserving* OT extensions. In order to see how the round-preserving aspect comes to help, we first apply a natural (yet insufficient) idea for extending our information-theoretic attack to the ROM. We will then rely on the round-preserving feature to resolve the remaining issues.

1st try: using the *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 getting rid of 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, HOZ16, BM17]. (See Part 1 of Theorem 2.) 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.[15] The magic of the algorithm Eve is that, because both parties can run it at the end, they can interpret $P_{\mathcal{E}}$ to be also part of the transcript, and so we get an augmented transcript $V_{\mathcal{E}} = (T, P_{\mathcal{E}})$ that includes (almost) all of the correlation between the parties' views.

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

2nd try: using the dependency learner over a *shortened* protocol. Recall that we are dealing with round-preserving OT extensions. 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_r is sent in round r. Now, if we *stop* the execution of $\mathcal{R}$ right after t_r is sent and call this modified variant of $\mathcal{R}$ the algorithm $\mathcal{R}_1$, even though the input c is submitted to $\mathbb{OT}_n$ by $\mathcal{R}_1$, no output is yet received by $\mathcal{R}_1$ from $\mathbb{OT}_n$, therefore we would not get any correlated randomness distributed between the parties through $\mathbb{OT}_n$. Therefore,

[15] 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.

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 $T = (t_1, \ldots, t_r)$. 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 being 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'.

The above argument (of applying the independence learning over a shortened variant of $\mathcal{S}$) still does not lead to an actual attack, because we need to *finish* the execution of $\mathcal{R}_1$ (as a partial $\mathcal{R}$ execution) to obtain the actual output corresponding to the fake input b' to be able to call it an attack done by the receiver. For that purpose, let us call $\mathcal{R}_2$ the rest of the execution of $\mathcal{R}$ after finishing the first part $\mathcal{R}_1$ till receiving γ from $\mathbb{OT}_n$. $\mathcal{R}_2$ takes over the computation of $\mathcal{R}_1$ 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 $\mathbf{H}$. In order to finish the execution of $\mathcal{R}_2$ by continuing the fake sampled (partial) view $V'_{\mathcal{R}}$ for (partial receiver algorithm) $\mathcal{R}_1$, we need to pretend that $V'_{\mathcal{R}}$ is the actual view of the receiver by pretending that the original honest view $V_{\mathcal{R}}$ (containing the original random input b and receiver's side till receiving γ) does not exist at all.

Relying on lack of intersection queries. Interestingly, it turns out that another crucial property of the Eve algorithm (i.e. Part 2 of Theorem 2) would allow us to get a consistent execution of $\mathcal{R}_2$ while pretending that the original honest (non-fake) execution of the receiver (encoded in view $V_{\mathcal{R}}$) does not exist. Namely, Eve guarantees that with high probability, there will be no 'intersection query' between the set of queries asked by the actual sender and the original (i.e., honest) partial execution of $\mathcal{R}$ (that obtains the fist output for the attacker $\widehat{\mathcal{R}}$). In a nutshell, what we do to finish the execution of $\mathcal{R}_2$ is to answer randomly any query q that is *not* learned by Eve, but *is* in the view of the original honest receiver's execution. See Section 4 for details and formal proofs.

Remark 1 (No need for conditioning on γ anymore). When we moved to the *round-preserving* case, the cheating receiver can sample its fake view $V'_{\mathcal{R}}$ for the partial execution $\mathcal{R}_1$ slightly differently. Namely, in this case, we do *not* have to further condition on γ (i.e., only need to condition on the input c for $\mathbb{OT}_n$) when we sample the fake view $V'_{\mathcal{R}}$. The reason is that γ is a function of the sender's view and c (which is part of receiver's partial view for partial execution $\mathcal{R}_1$). This observation also allows us to handle a *string* based OT-hybrid functionality, as we do not loose a lot of entropy by conditioning on the $\mathbb{OT}_n$ outputs.

Organization. In Section 2 we define the basic notation, definitions, and tools used throughout this paper. In particular, that section includes definitions and lemmas on statistical distance (Subsection 2.1), Shannon entropy (Subsection 2.2), random oracle model (Subsection 2.3). In Section 3 we formalize the notion of

round-preserving OT extension. In Section 4 we prove our main impossibility result of Theorem 1. In Section 5 we observe that Beaver's non-black-box round-preserving OT extension protocol [Bea96] can be adapted to *chosen* inputs.

2 Preliminaries

Unless stated otherwise, logarithms in this work are taken base 2 and are denoted by $\lg(\cdot)$. 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 $\mathbf{X}, \mathbf{Y}$. 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 $Z \leftarrow \mathbf{Z}$, by $(\mathbf{X} \times \mathbf{Y})|Z$ we denote $((\mathbf{X}|Z) \times (\mathbf{Y}|Z))$. 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 $\mathbf{X}$ over which we are interested in the probability of B. 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 $X \in \mathsf{B}$.[16] For an event B by $\overline{\mathsf{B}}$ we denote to the complement event of B with respect to the underlying universe U. Therefore, $\Pr[\mathbf{X} \in \overline{\mathsf{B}}]$ is always well defined and is equal to $1 - \Pr[\mathbf{X} \in \mathsf{B}]$. By $\Pr_{\mathcal{D}}[\mathsf{B}]$ or $\Pr[\mathsf{B}; \mathcal{D}]$ we mean the probability of B happening where the sampling process is described in $\mathcal{D}$.

2.1 Lemmas about Statistical Distance

Definition 1 (Statistical distance). *By* $\mathsf{SD}(\mathbf{X}, \mathbf{Y})$ *we denote the* statistical distance *between random variables* $\mathbf{X}, \mathbf{Y}$ *defined as*

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

[16] 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 as properties over objects sampled from the support set of the relevant random variables.

We call $\mathbf{X}$ *and* $\mathbf{Y}$ ε-close*, denoted by* $\mathbf{X} \approx_\varepsilon \mathbf{Y}$*, if* $\mathsf{SD}(\mathbf{X}, \mathbf{Y}) \leq \varepsilon$*. For an event* A*, we let* $\mathsf{SD}_{\mathsf{A}}(\mathbf{X}, \mathbf{Y}) = \mathsf{SD}((\mathbf{X} \mid \mathsf{A}), (\mathbf{Y} \mid \mathsf{A}))$*, 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, the first part is a well-known[17] fact stating that statistical distance is the maximum advantage of distinguishing two distributions. The second states that by adding information from the same marginal distributions, the statistical distance does not change.

Lemma 1. *Let* D *be any potentially randomized (distinguishing) algorithm. Then:*

1. $\Pr[D(\mathbf{X}) = 1] - \Pr[D(\mathbf{Y}) = 1] \leq \mathsf{SD}(\mathbf{X}, \mathbf{Y})$ *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* $\Pr[\mathbf{X} = Z] < \Pr[\mathbf{Y} = Z]$*.*
2. *If* $(\mathbf{X}, \mathbf{Y}), (\mathbf{X}', \mathbf{Y}')$ *are joint distributions and* $(\mathbf{Y} \mid \mathbf{X} = X) \equiv (\mathbf{Y}' \mid \mathbf{X}' = X)$ *for all* $X \in \mathrm{Supp}(\mathbf{X}) \cap \mathrm{Supp}(\mathbf{X}')$*, then* $\mathsf{SD}((\mathbf{X}, \mathbf{Y}), (\mathbf{X}', \mathbf{Y}'')) = \mathsf{SD}(\mathbf{X}, \mathbf{X}')$*.*

Proof. We prove the second part using the first part. $\mathsf{SD}((\mathbf{X}, \mathbf{Y}), (\mathbf{X}', \mathbf{Y}'')) \geq \mathsf{SD}(\mathbf{X}, \mathbf{X}')$ because a distinguisher of $(\mathbf{X}, \mathbf{Y})$ from $(\mathbf{X}', \mathbf{Y}'')$ can always ignore the second component. On the other hand, if $(\mathbf{Y} \mid X) \equiv (\mathbf{Y}' \mid \mathbf{X}' = X)$, then $\mathsf{SD}((\mathbf{X}, \mathbf{Y}), (\mathbf{X}', \mathbf{Y}'')) \leq \mathsf{SD}(\mathbf{X}, \mathbf{X}')$, because a distinguisher for $\mathbf{X}$ from $\mathbf{X}'$ can always *add* the second component from its marginal distribution and use the distinguisher of $(\mathbf{X}, \mathbf{Y})$ from $(\mathbf{X}', \mathbf{Y}'')$. □

The following well-known lemma[18] states that statistically close distributions could be sampled jointly while they are equal with high probability.

Lemma 2 (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 1. $\mathsf{SD}((\mathbf{X}, \mathbf{Y}), (\mathbf{X}, \mathbf{Y}')) = \mathbb{E}_{X \leftarrow \mathbf{X}} \mathsf{SD}((\mathbf{Y} \mid X), (\mathbf{Y}' \mid X))$.

Proof. By Lemma 1, $\mathsf{SD}((\mathbf{X}, \mathbf{Y}), (\mathbf{X}, \mathbf{Y}'))$ 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, and when the second sample Y arrives, it has to judge whether it is sampled from $(\mathbf{Y} \mid X)$ or $(\mathbf{Y}' \mid X)$. But, for each X, the maximum probability of outputting correctly is again described by Lemma 1 to be equal to $\mathsf{SD}((\mathbf{Y} \mid X), (\mathbf{Y}' \mid X))$. □

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

[17] For example, see Exercise 8.61 from [Sho09] for a proof.
[18] For example, see lemma 3.6 of [Ald83] for a proof.

Definition 2 (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 form Proposition 1 and Definition 2.

Proposition 2. *It holds that* $\mathsf{MutDep}(\mathbf{X}, \mathbf{Y}) = \mathbb{E}_{X \leftarrow \mathbf{X}} \mathsf{SD}((\mathbf{Y} \mid X), \mathbf{Y})$.

Lemma 3. *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 3. 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 4. *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 2. Before proving Lemma 4, 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 4). 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 2. 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 Part 2 of Lemma 1, it holds that

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

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 Part 2 of Lemma 1, 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 Equations (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[19] of the following lemma are used in previous works. 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 5 (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 2, 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 5 follows (again by an application of Lemma 2).

Firstly, by Lemma 2 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 2, 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. □

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

2.2 Lemmas about Shannon Entropy

Definition 3 (Shannon entropy). *For a random variable* $\mathbf{X}$*, its Shannon entropy is defined as* $\mathrm{H}(\mathbf{X}) = \mathbb{E}_{X \leftarrow \mathbf{X}} \lg(1/\Pr[\mathbf{X} = X])$*. 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*.*[20]

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

Lemma 6 (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 2 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

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

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 7 (Upper-bounding collision probability using 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*

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

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:

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 $\varepsilon \geq 2\varepsilon^2$.

[20] 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.

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 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

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 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 $V_{\mathcal{A}} = (r_{\mathcal{A}}, T, P_{\mathcal{A}})$.

Definition 4 (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.

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

Theorem 2 (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 2 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 Section 4.

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.

3 Round-preserving OT Extension

In this section, we formalize the notions of OT, and round preserving OT extension, and we will also prove basic lemmas that allows us to prove the existence of *attacks* against OT extensions.

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

Definition 5 (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 $\{y_i^0, y_i^1\}_{i\in[k]}$ where y_i^b are arbitrary strings, and the receiver has the input $(c_1, \ldots, c_k) \in \{0,1\}^k$. The sender receives no output at the end, while the receiver receives $\{y_i^{c_i}\}_{i\in[k]}$.

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 b of $\mathcal{R}$ and any input $x = \{x_i^0, x_i^1\}_{i\in[n+1]}$ for $\mathcal{S}$, it holds that:

$$\mathsf{Sim}_{\mathcal{S}}(x) \equiv_c \mathbf{V}_{\mathcal{S}}(x, b) \quad \text{and} \quad \mathsf{Sim}_{\mathcal{R}}(b, \{x_i^{b_i}\}_{i\in[n+1]}) \equiv_c \mathbf{V}_{\mathcal{R}}(x, b).$$

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 all the parties are poly(κ)-query algorithms accessing a random oracle $\mathbf{H}$.[21] Recall that by the notation defined at the end of Section 2 we can use the same notation $\equiv_c$ for indistinguishably against poly(κ) distinguishers in the ROM.

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 this can be composed to get after one extension is possible (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].

In this work, we are 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. It is instructive to first define a 2-round-preserving OT extension protocol, both 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.

A 2-round-preserving OT extension protocol has the following form.

1. $\mathcal{S}$ has input $\{x_i^0, x_i^1\}_{i\in[n+1]}$ and $\mathcal{R}$ has input $b = (b_1, \ldots, b_{n+1})$.
2. $\mathcal{R}$ sends a single message to $\mathcal{S}$.
3. $\mathcal{R}$ chooses input $c = (c_1, \ldots, c_n)$ and $\mathcal{S}$ chooses inputs $\{y_i^0, y_i^1\}_{i\in[n]}$ for a hybrid functionality $\mathbb{OT}_n$.
4. $\mathcal{S}$ sends a single message to $\mathcal{R}$ and $\mathcal{R}$ also receives $\gamma = \{y_i^{c_i}\}_{i\in[n]}$from $\mathbb{OT}_n$.
5. $\mathcal{R}$ outputs what is supposed to be $\{x_i^{b_i}\}_{i\in[n+1]}$.

Given a 2-round-preserving OT extension protocol E from n to $n+1$, we may instantiate $\mathbb{OT}_n$ with a concrete 2-round OT to obtain $(n+1)$-OT. This can be done by running the base protocol in parallel with the extension protocol.

[21] In fact, we still want to have honest parties and the simulator to be **PPT** *oracle-aided* algorithms, but our impossibility results apply even if they are not.

Our impossibility result is not restricted to the case of 2-round protocols, and so we can generalize the above definition to the following model.

Definition 6 (Round-preserving OT Extension). *A round-preserving OT extension protocol is a 2-party protocol with the following form. In the random oracle model, both parties can query the random oracle* $\mathbf{H}$ *all along the protocol.*

1. $\mathcal{S}$ *has input* $\{x_i^0, x_i^1\}_{i\in[n+1]}$ *and* $\mathcal{R}$ *has input* $b = (b_1, \ldots, b_{n+1})$.
2. $\mathcal{R}$ *and* $\mathcal{S}$ *exchange an arbitrary number of messages.*
3. *At some point before* $\mathcal{S}$ *sends the final message to* $\mathcal{R}$, $\mathcal{S}$ *submits its inputs* $\{y_i^0, y_i^1\}_{i\in[n]}$ *and* $\mathcal{R}$ *submits its input* $c = (c_1, \ldots, c_n)$ *to* $\mathbb{OT}_n$.
4. $\mathcal{S}$ *sends a final message to* $\mathcal{R}$ *and* $\mathcal{R}$ *also receives* $\{y_i^{c_i}\}_{i\in[n]}$ *from* $\mathbb{OT}_n$.
5. $\mathcal{R}$ *outputs what is supposed to be* $\{x_i^{b_i}\}_{i\in[n+1]}$.

The semi-honest security of OT extension is defined based on the semi-honest security of the constructed primitive, namely k-OT (defined above) 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 because if the OT-hybrid functionality in an r-round OT extension protocol were implemented using an r-round base OT protocol, the output would only be available after the final message had been sent. In this definition, the parties choose their inputs for $\mathbb{OT}_n$ at some point 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 $\mathbb{OT}_n$ in the end only makes our impossibility result stronger.

In our definition, 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 are not enough rounds such that the output of $\mathbb{OT}_n$ affects any message sent to the receiver. 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.

Lemmas for breaking the semi-honest security of OT extension. We now state and prove two simple lemmas showing that the attacks that we construct in Section 4 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 8. *Let $(\mathcal{S},\mathcal{R})$ be a semi-honest secure, round-preserving OT extension protocol in which the receiver's inputs are chosen uniformly at random and in which $\mathcal{S},\mathcal{R}$ are* **PPT***s (resp., oracle-aided* $\mathrm{poly}(\kappa)$*-query machines). Fix any input x for the sender. Let $\mathbf{b} \equiv \mathbf{U}_{n+1}$ be the uniformly random inputs of the receiver and $\mathbf{V}_{\mathcal{S}}(x,\mathbf{b})$ the random variable denoting the view of the sender (for inputs $x,\mathbf{b}$ being used by the sender and the receiver). Then we have*

$$(\mathbf{V}_{\mathcal{S}}(x,\mathbf{b}),\mathbf{b}) \equiv_c (\mathbf{V}_{\mathcal{S}}(x,\mathbf{b}) \times \mathbf{U}_{n+1})$$

where $\mathbf{U}_{n+1}$ is independent sampled.

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 all the involved algorithms are $\mathrm{poly}(\kappa)$-query oracle-aided algorithms accessing a random oracle $\mathbf{H}$.

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

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

Hence, by averaging over $b \leftarrow \mathbf{b}$, we have $(\mathbf{V}_{\mathcal{S}}(x,\mathbf{b}),\mathbf{b}) \equiv_c (\mathsf{Sim}_{\mathcal{S}}(x),\mathbf{b})$ for uniform $\mathbf{b}$. Since $\mathsf{Sim}_{\mathcal{S}}(x)$ is independent of the receiver's input $\mathbf{b}$, we conclude

$$(\mathbf{V}_{\mathcal{S}}(x,\mathbf{b}),\mathbf{b}) \equiv_c (\mathsf{Sim}_{\mathcal{S}}(x),\mathbf{b}) \equiv \mathsf{Sim}_{\mathcal{S}}(x) \times \mathbf{U}_{n+1} \equiv_c \mathbf{V}_{\mathcal{S}}(x,\mathbf{b}) \times \mathbf{U}_{n+1}. \quad \square$$

Remark 3. In Section 4, we will consider the following attack by a sender. The sender will solely based on his own view try to distinguish the receivers input b from an independently uniform string. We use Lemma 8, to argue that a non-negligible distinguishing probability of the sender will break the simulation based security of OT. Notice that Lemma 8 requires the indistinguishability of $(\mathbf{V}_{\mathcal{S}}(x,\mathbf{b}),\mathbf{b}) \equiv_c \mathbf{V}_{\mathcal{S}}(x,\mathbf{b}) \times \mathbf{U}_{n+1}$. Since $\mathbf{V}_{\mathcal{S}}(x,\mathbf{b})$ is identically distributed in both distributions, in order to show an ε distinguisher between $(\mathbf{V}_{\mathcal{S}}(x,\mathbf{b}),\mathbf{b})$ and $\mathbf{V}_{\mathcal{S}}(x,\mathbf{b}) \times \mathbf{U}_{n+1}$ it is equivalent to show that the average advantage of $D(\mathbf{V}_{\mathcal{S}}(x,\mathbf{b}),\cdot)$ (over the randomness of $\mathbf{V}_{\mathcal{S}}(x,\mathbf{b})$) to distinguishing $\mathbf{b}$ from U_{n+1} is at least ε. More generally, for distinguishing any pairs $(\mathbf{X},\mathbf{Y}),(\mathbf{X},\mathbf{Y}')$ it holds

$$\begin{aligned}&\Pr[D(\mathbf{X},\mathbf{Y})=1] - \Pr[D(\mathbf{X},\mathbf{Y}')=1]\\ &= \mathop{\mathbb{E}}_{X\leftarrow\mathbf{X}}\big[\Pr[D(X,\mathbf{Y}\mid X)=1] - \Pr[D(X,\mathbf{Y}'\mid X)]=1\big].\end{aligned}$$

Hence, an ε distinguisher achieves an average of ε distinguishing of the second component (over the randomness of the first component). Therefore, the outlined attack by a sender will imply a distinguisher for Lemma 8.

Lemma 9. *Let $(\mathcal{S},\mathcal{R})$ be a semi-honest secure, round-preserving OT protocol in which the sender's inputs are chosen uniformly at random and in which $\mathcal{S},\mathcal{R}$ are* **PPT***s (resp., oracle-aided* $\mathrm{poly}(\kappa)$*-query machines). Fix any input vector b for the receiver. Let $\mathbf{x} = \{\mathbf{x}_i^0,\mathbf{x}_i^1\}_{i\in[n+1]}$ be the uniformly random inputs of the*

sender and let $\mathbf{V}_{\mathcal{R}}(\mathbf{x}, b)$ *be the random variable denoting the view of the receiver (when the inputs* $\mathbf{x}, b$ *are used by the parties). Then, it holds that*

$$(\mathbf{V}_{\mathcal{R}}(\mathbf{x}, b), \{\mathbf{x}_i^{\overline{b_i}}\}_{i\in[n+1]}) \equiv_c (\mathbf{V}_{\mathcal{R}}(\mathbf{x}, b) \times \{\mathbf{x}'_i\}_{i\in[n+1]})$$

where the $\mathbf{x}'_i$ *are independent, uniform random strings of the appropriate length.*

Proof. As in the proof of Lemma 8, w 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 algorithms are $\mathrm{poly}(\kappa)$-query algorithms in the random oracle model.

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

$$\mathsf{Sim}_{\mathcal{R}}(b, \{x_i^{b_i}\}_{i\in[n+1]}) \equiv_c \mathbf{V}_{\mathcal{R}}(x, b).$$

Hence $(\mathbf{V}_{\mathcal{R}}(\mathbf{x}, b), \{\mathbf{x}_i^{\overline{b_i}}\}_{i\in[n+1]}) \equiv_c (\mathsf{Sim}_{\mathcal{R}}(b, \{\mathbf{x}_i^{b_i}\}_{i\in[n+1]}), \{\mathbf{x}_i^{\overline{b_i}}\}_{i\in[n+1]})$ holds for uniform $\mathbf{x}$. Since $\mathsf{Sim}_{\mathcal{R}}(b, \{\mathbf{x}_i^{b_i}\}_{i\in[n+1]})$ is independent of $\{\mathbf{x}_i^{\overline{b_i}}\}_{i\in[n+1]}$ (i.e., the sender's input that is not learned by receiver), we conclude that

$$(\mathbf{V}_{\mathcal{R}}(\mathbf{x}, b), \{\mathbf{x}_i^{\overline{b_i}}\}_{i\in[n+1]}) \equiv_c \mathbf{V}_{\mathcal{R}}(\mathbf{x}, b) \times \{\mathbf{x}'_i\}_{i\in[n+1]}.$$

□

Remark 4. Similar to the attack by a sender, we will consider an attack by the receiver that distinguishes the senders input $\{x_i^{\overline{b_i}}\}_{i\in[n+1]}$ from uniform. Even more, the receiver will learn one of these input strings with non-negligible probability. By Lemma 9 and the same reasoning as in case of Lemma 8 (see also Remark 3) this will break the simulation based security of OT.

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

In this section we formally prove our main negative result by showing that there is no round-preserving OT extension (as in Definition 6) in the random oracle model. Namely, we show that for any such protocol, there is always a $\mathrm{poly}(\kappa)$-query attack by one of the parties succeeding (in breaking the security) with advantage $1/\mathrm{poly}(\kappa)$. In fact, we show how to break the security of such protocols even when the inputs are chosen at random. Namely, even *random-input* OT extension protocols cannot be secure in the random oracle model.

Theorem 3. *Let* $(\mathcal{S}, \mathcal{R})$ *be a* round-preserving *OT extension protocol (according to Definition 6) with security parameter* κ *using random oracle* $\mathbf{H}$ *as follows.*

1. *The* $n \leq \mathrm{poly}(\kappa)$ *OTs modeled by* $\mathbb{OT}_n$ *are allowed to be* string *OTs.*
2. $(\mathcal{S}, \mathcal{R})$ *implement* bit $(n+1)$*-OT with* $\lambda = \mathrm{negl}(\kappa)$ *completeness error.*

3. *Either of* $(\mathcal{S}, \mathcal{R})$ *ask at most* $m = \text{poly}(\kappa)$ *queries to the random oracle* $\mathbf{H}$.

Then either $\mathcal{S}$ *or* $\mathcal{R}$ *can ask* $\text{poly}(m \cdot n) \leq \text{poly}(\kappa)$ *queries to* $\mathbf{H}$*, after they execute the protocol in a semi-honest way, and at the end break the security of their bit* $(n+1)$*-OT by advantage* $\frac{1}{\text{poly}(n)} \geq \frac{1}{\text{poly}(\kappa)}$.

In the rest of this section, we prove Theorem 3 above.

Notation. First we clarify our notation again.

- $b = (b_1, \ldots, b_{n+1}) \in \{0,1\}^{n+1}$ is $\mathcal{R}$'s input, and it submits $c = (c_1, \ldots, c_n) \in \{0,1\}^n$ as its input to $\mathbb{OT}_n$.
- $\{x_i^0, x_i^1\}_{i \in [n+1]}$ 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 $\mathbb{OT}_n$.

Informally speaking, we will show that by asking $\text{poly}(\kappa)$ 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 8, or the receiver can read both of sender's inputs for an index i with non-negligible probability[22]), which is an attack by Lemma 9.

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 2. We will use that lemma for some sufficiently small parameter δ that will be chosen when we analyze the attacks.

Construction 4 (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* b *(chosen at random and then) used by the receiver, given the transcript of the protocol* T *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 2 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 2.*

[22] 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.

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.*[23] *Then, given an input from $\{0,1\}^{n+1}$, $\widehat{\mathcal{S}}$ tries to use the maximum-likelihood method to distinguish receiver's input b from a random string. Namely, given a string β, $\widehat{\mathcal{S}}$ outputs 1 if $\Pr[\mathbf{b} = \beta \mid V_{\mathcal{E}}] > 2^{-(n+1)}$, where $\mathbf{b}$ is the random variable denoting the receiver's input 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 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 5 (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 b. Doing this would establish an attack by Lemma 9. 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 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 $b' \neq b$. $\mathcal{R}'$ will try to pick $b' \neq b$ 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 c), so that the obtained output is correct with respect to 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 $\mathcal{R}'$ 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 2 over the* same *input-less protocol $(\mathcal{S}, \mathcal{R}_1)$ used by $\widehat{\mathcal{S}}$ in Construction 4 (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 4.*
2. *$\widehat{\mathcal{R}}$ then samples from the distribution $V'_{\mathcal{R}} \leftarrow (\mathbf{V}_{\mathcal{R}} \mid \mathbf{V}_{\mathcal{E}} = V_{\mathcal{E}}, \mathbf{c} = c)$ 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 $V'_{\mathcal{R}}, \gamma$ 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 (continued execution of) $\mathcal{R}'$ is answered. First let $\mathcal{L}$, to be an empty list (of query-answer pairs), and update it inductively, whenever a new query q is asked by $\mathcal{R}'$, as follows.*

[23] 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.

(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 $q \in \mathcal{Q}(V_{\mathcal{R}}) \setminus (\mathcal{Q}(V_{\mathcal{E}}) \cup \mathcal{Q}(V'_{\mathcal{R}}))$,[24] pick a random answer a for query q, and also add (q, a) to $\mathcal{L}$ for the future.
(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 corresponding to the input 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 4 will distinguish $\mathbf{b}$ from $\mathbf{U}_{n+1}$ with advantage at least $\Omega(1/n)$, or the attacking receiver $\widehat{\mathcal{R}}$ of Construction 5 can obtain correct outputs corresponding to its random b as well as some $b' \neq b$ 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 3 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 or the attacking receiver succeeds with advantage $\Omega(1/n^2)$ by asking $\mathrm{poly}(\kappa)$ oracle queries.

4.1 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 $\mathsf{SD}_{\mathbf{V}_{\mathcal{E}}}(\mathbf{b}, \mathbf{U}_{n+1}) \geq \varepsilon$, then by Lemma 1, it holds that the sender's attacking strategy $\widehat{\mathcal{S}}$ of Construction 4 will be able to ε-distinguish the true randomly chosen input $\mathbf{b}$ of the receiver $\mathcal{R}$ from a uniform string $\mathbf{U}_{n+1}$ by advantage at least ε. Therefore, by Lemma 8, $\widehat{\mathcal{R}}$ succeeds in breaking the security with non-negligible advantage ε. (See Remark 3 on how a distinguisher between $\mathbf{b}$ and $\mathbf{U}_{n+1}$ enables the attacker of Lemma 8.)

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

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

and we will show, using Inequality (3) and Lemma 9, 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 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) $b' \neq b$ and (2) the receiver $\mathcal{R}'$ gets the output corresponding to b' (and sender's actual input x). It will suffice to prove the following two:

[24] 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 $\mathcal{R}_1$.

- $\Pr[\mathbf{b}' \neq \mathbf{b}] \geq \Omega(1/n^2)$ where $\mathbf{b}$ and $\mathbf{b}'$ are the random variables denoting the original and the fake inputs of $\widehat{\mathcal{R}}$.
- The receiver will get the right answer for 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* $\Pr[\mathbf{b}' \neq \mathbf{b}] \geq \Omega(1/n^2)$ *where the probability is over the randomness of the honest sender* $\mathcal{S}$ *and the cheating receiver* $\widehat{\mathcal{R}}$*, and the oracle* $\mathbf{H}$*.*

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 $c \leftarrow (\mathbf{c} \mid V_{\mathcal{E}})$ is sampled, then $(V_{\mathcal{S}}, V_{\mathcal{R}}) \leftarrow ((\mathbf{V}_{\mathcal{S}}, \mathbf{V}_{\mathcal{R}}) \mid V_{\mathcal{E}}, c)$, and finally $V'_{\mathcal{R}} \leftarrow (\mathbf{V}_{\mathcal{R}} \mid V_{\mathcal{E}}, c)$ 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 b, b'. Thus, we can think of $V_{\mathcal{R}}, V'_{\mathcal{R}}$ as two independent samples from the same distribution $(\mathbf{V}_{\mathcal{R}} \mid V_{\mathcal{E}}, c)$. Consequently, b, b' are also two independent samples from $(\mathbf{b} \mid V_{\mathcal{E}}, c)$.

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 $\mathsf{SD}_{V_{\mathcal{E}}}(\mathbf{b}, \mathbf{U}_{n+1}) < \varepsilon' = \frac{1}{100n+100}$. We call such $V_{\mathcal{E}}$ a 'good' sample. For any good $V_{\mathcal{E}}$, using Lemma 6 it holds that $\mathrm{H}(\mathbf{b} \mid V_{\mathcal{E}}) \geq (1 - \varepsilon') \cdot (n + 1) - \mathrm{H}(\varepsilon')$, and since the length of c is n, by further conditioning on random variable $\mathbf{c}$ we have:

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

where the last inequality follows from $\varepsilon' \leq 1/200$, and $\mathrm{H}(1/200) < 1/20$. Therefore, by Lemma 7 (using $\mathbf{X} = \mathbf{b}, \mathbf{Y} = (V_{\mathcal{E}}, \mathbf{c})$) we conclude that the event $b \neq b'$ 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 $b \neq b'$ 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* 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 the different experiments, we might use $\langle \mathbf{X} \rangle_{\mathsf{Z}}$ to indicate a random variable $\mathbf{X}$ 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 $\langle V_{\mathcal{E}}, c, V_{\mathcal{S}}, V_{\mathcal{R}}, V'_{\mathcal{R}}, P' \rangle_{\mathsf{Real}}$ 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 $\langle V_{\mathcal{E}}, c, V_{\mathcal{S}}, V'_{\mathcal{R}}, V_{\mathcal{R}}, P \rangle_{\mathsf{Ideal}}$. 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 5 (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 $V_{\mathcal{S}}, c$. In particular, because both $V_{\mathcal{R}}, V'_{\mathcal{R}}$ are consistent with c, they can both lead to correct answers for b, b' which are their inputs. Moreover, if we *did* include γ in the outputs of the experiments, it would *not* change their statistical distance.

Remark 6 (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 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.

It is enough to prove $O(\delta)$-closeness. In the following we will show that the outputs of the two experiments are $O(\delta)$-close. After that, because the completeness error in the ideal word is at most λ, and this is a probability over the random output $\langle \mathbf{V}_{\mathcal{E}}, \mathbf{V}_{\mathcal{S}}, \mathbf{V}_{\mathcal{R}}, \mathbf{P} \rangle_{\mathsf{Ideal}}$, we 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 $\lambda + O(\delta)$, where the completeness now means that the fake view is getting the right answer!

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

1. We first prove that $\langle \mathbf{V}_\mathcal{E}, \mathbf{c}, \mathbf{V}_\mathcal{S}, \mathbf{V}_\mathcal{R}, \mathbf{V}'_\mathcal{R} \rangle_{\mathsf{Real}} \approx_{O(\delta)} \langle \mathbf{V}_\mathcal{E}, \mathbf{c}, \mathbf{V}_\mathcal{S}, \mathbf{V}'_\mathcal{R}, \mathbf{V}_\mathcal{R} \rangle_{\mathsf{Ideal}}$.
2. Then we show that $\Pr[\langle \mathbf{V}_\mathcal{E}, \mathbf{c}, \mathbf{V}_\mathcal{S}, \mathbf{V}_\mathcal{R}, \mathbf{V}'_\mathcal{R} \rangle_{\mathsf{Real}} \in \mathsf{B}] \leq \delta$ 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 $\alpha = \langle V_\mathcal{E}, c, V_\mathcal{S}, V_\mathcal{R}, V'_\mathcal{R} \rangle_{\mathsf{Real}} \leftarrow \langle \mathbf{V}_\mathcal{E}, \mathbf{c}, \mathbf{V}_\mathcal{S}, \mathbf{V}_\mathcal{R}, \mathbf{V}'_\mathcal{R} \rangle_{\mathsf{Real}}$ (i.e., $\alpha \notin \mathsf{B}$) and if the sampled prefixes of the outputs are equal $\langle V_\mathcal{E}, c, V_\mathcal{S}, V'_\mathcal{R}, V_\mathcal{R} \rangle_{\mathsf{Ideal}} = \langle V_\mathcal{E}, c, V_\mathcal{S}, V_\mathcal{R}, V'_\mathcal{R} \rangle_{\mathsf{Real}}$, then the corresponding distributions

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

will be identically distributed.

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

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

Proof. By Part 1 of Theorem 2 it holds that in the real world:

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

By averaging over $V_\mathcal{E} \leftarrow \mathbf{V}_\mathcal{E}$ and then using Lemma 4 (and letting $\mathbf{C} := \mathbf{c}, \mathbf{B} := \mathbf{V}_\mathcal{R}, \mathbf{A} := \mathbf{V}_\mathcal{S}$) and noting that c is only a function of $V_\mathcal{R}$, it holds that

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

For a fixed $(V_\mathcal{E}, c) \leftarrow (\mathbf{V}_\mathcal{E}, \mathbf{c})$, we can use Lemma 3 (by letting $\mathbf{X} \equiv (\mathbf{V}_\mathcal{S} \mid V_\mathcal{E}, c)$ and $\mathbf{Y} \equiv (\mathbf{V}_\mathcal{R} \mid V_\mathcal{E}, c)$) and then average over $(V_\mathcal{E}, c) \leftarrow (\mathbf{V}_\mathcal{E}, \mathbf{c})$ to conclude

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

Finally, by Proposition 1, the left side of the above inequality is the same as $\mathsf{SD}(\langle \mathbf{V}_\mathcal{E}, \mathbf{c}, \mathbf{V}_\mathcal{S}, \mathbf{V}_\mathcal{R}, \mathbf{V}'_\mathcal{R} \rangle_{\mathsf{Real}}, \langle \mathbf{V}_\mathcal{E}, \mathbf{c}, \mathbf{V}_\mathcal{S}, \mathbf{V}'_\mathcal{R}, \mathbf{V}_\mathcal{R} \rangle_{\mathsf{Ideal}})$, 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 7 (The bad event B). *Let U be a 'universe' containing all possible outputs of the two experiments (and maybe more elements) defined as follows:*

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

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

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

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

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 2 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 7 (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 5) 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 7 and stating Claim 5 in the Real experiment to be the simplest choice to prove (using Theorem 2) and use for proving Claim 3 (by bounding the statistical distance of the outputs of experiments using Lemma 5).

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

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

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 $\langle V_{\mathcal{E}}, c, V_{\mathcal{S}}, V'_{\mathcal{R}}, V_{\mathcal{R}}\rangle_{\mathsf{Ideal}}$, 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. Since we are assuming $\langle V_{\mathcal{E}}, c, V_{\mathcal{S}}, V'_{\mathcal{R}}, V_{\mathcal{R}}\rangle_{\mathsf{Ideal}} = \langle V_{\mathcal{E}}, c, V_{\mathcal{S}}, V_{\mathcal{R}}, V'_{\mathcal{R}}\rangle_{\mathsf{Real}}$, 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 $\mathcal{Q}(V_{\mathcal{E}}) \cup \mathcal{Q}(V_{\mathcal{S}}) \cup \mathcal{Q}(V'_{\mathcal{R}})$ 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 5. (In the second case below we make a crucial use of the fact that the event B has not happened over the current sample $\langle V_{\mathcal{E}}, c, V_{\mathcal{S}}, V'_{\mathcal{R}}, V_{\mathcal{R}}\rangle_{\mathsf{Ideal}} = \langle V_{\mathcal{E}}, c, V_{\mathcal{S}}, V_{\mathcal{R}}, V'_{\mathcal{R}}\rangle_{\mathsf{Real}}$.)

- If $q \in \mathcal{Q}(V'_{\mathcal{R}})$, then $\widehat{\mathcal{R}}$ uses the answer stated in $V'_{\mathcal{R}}$.
- if $q \in \mathcal{Q}(V_{\mathcal{R}}) \setminus (\mathcal{Q}(V_{\mathcal{E}}) \cup \mathcal{Q}(V'_{\mathcal{R}}))$, $\widehat{\mathcal{R}}$ 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 (recall the fake and real views are switched across the Real vs. Ideal experiments) and so it would also be answered at random except if q is already in $\mathcal{Q}(V_{\mathcal{S}})$. However, if this happens, it means that event B holds over the sample $\alpha = \langle V_{\mathcal{E}}, c, V_{\mathcal{S}}, V'_{\mathcal{R}}, V_{\mathcal{R}}\rangle_{\mathsf{Ideal}} = \langle V_{\mathcal{E}}, c, V_{\mathcal{S}}, V_{\mathcal{R}}, V'_{\mathcal{R}}\rangle_{\mathsf{Real}}$ (i.e., $\alpha \in \mathsf{B}$) which we assumed is not the case.
- If above cases do not happen, but q is still part of $\mathcal{Q}(V_{\mathcal{E}})\cup\mathcal{Q}(V_{\mathcal{S}})$ in the Real experiment, $\widehat{\mathcal{R}}$ would ask it 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. □

Finishing the proofs of Claims 2 and 3 also finishes the proof of Claim 1.

5 Non-Black Box, Round-Preserving OT Extension

5.1 Garbled Circuits

Circuit garbling is a primitive that was introduced and constructed by Yao [Yao86]. Intuitively, it transforms a circuit and its inputs in a way such that the circuit can only be evaluated for one specific input, but for any other input the output remains hidden. More formally, a circuit garbling consists of two **PPT** algorithms Garble, Eval as follows.

Garble $(1^\kappa, C)$**:** Garble takes as input a circuit C and outputs a garbled Circuit $\hat{C}$ together with input labels $\{L_i^0, L_i^1\}_{i\in[n]}$, where n is the input length of C.

Eval $(\hat{C}, \{L_i^{x_i}\}_{i\in[n]})$**:** Eval takes as input a garbled circuit $\hat{C}$ and a set of input labels $\{L_i^{x_i}\}_{i\in[n]}$ that correspond to an input $x \in \{0,1\}^n$. It evaluates circuit $\hat{C}$ and outputs its output.

A circuit garbling is correct if for any circuit C and any input $x \in \{0,1\}^n$

$$\Pr[\mathsf{Eval}(\hat{C}, \{L_i^{x_i}\}_{i\in[n]}) = C(x); (\hat{C}, \{L_i^0, L_i^1\}_{i\in[n]}) \leftarrow \mathsf{Garble}(1^\kappa, C)] \geq 1 - \mathrm{negl}(\kappa),$$

where the probability is taken over the random coins of Eval and Garble.

A simulator Sim for a circuit garbling takes as input an output $C(x)$ for a circuit C and input x and outputs a garbled circuit $\hat{C}$ and a set of input labels $\{L_i^{x_i}\}_{i\in[n]}$. A circuit garbling is secure if for any **PPT** algorithm A, any circuit C and any input x

$$|\Pr[\mathsf{A}(\hat{C}, \{L_i^{x_i}\}_{i\in[n]}) = 1] - |\Pr[\mathsf{A}(\hat{C}', \{L'^{x_i}_i\}_{i\in[n]}) = 1]| \leq \mathrm{negl}(\kappa),$$

where the probabilities are taken over the random coins of A, Garble, and Sim and $(\hat{C}, \{L_i^0, L_i^1\}_{i\in[n]}) \leftarrow \mathsf{Garble}(1^\kappa, C)$, $(\hat{C}', \{L'^{x_i}_i\}_{i\in[n]}) \leftarrow \mathsf{Sim}(1^\kappa, C(x), \mathsf{top}(C))$. Here, $\mathsf{top}(C)$ denotes the topology of C.

5.2 Non-Black Box, Round-Preserving OT Extension

We observe that Beaver's OT extension protocol [Bea96] can be adapted to yield 1-out-of-2 "chosen" OT rather than "random" OT. In the following we describe this adaptation. This is a simple extension of Beaver's protocol that to the best of our knowledge has not appeared in the literature before. It is a round-preserving OT extension protocol, secure against honest but curious adversaries. It is unaffected by our impossibility result as it makes non-black-box use of a cryptographic functionality – namely, a pseudo-random generator.

Let $\mathsf{PRG} : \{0,1\}^\kappa \to \{0,1\}^{\kappa+\ell}$ be a pseudo-random generator and Garble a circuit garbling. $\mathcal{S}$ and $\mathcal{R}$ interact in the following way, where $b = (b_1, \ldots, b_{\kappa+\ell})$ are the receiver's choice bits and $\{x_i^0, x_i^1\}_{i\in[\kappa+\ell]}$ is the sender's input.

1. In the first round, $\mathcal{R}$ chooses a random seed $s \leftarrow \{0,1\}^\kappa$. He sends $a = \mathsf{PRG}(s) \oplus b$ to the sender and s to the OT-hybrid functionality $\mathbb{OT}_\kappa$.
2. $\mathcal{S}$ computes the circuit $C_{[a,\{x_i^0,x_i^1\}_{i\in[\kappa+\ell]}]}(s')$: output $\{x_i^{b'_i}\}_{i\in[\kappa+\ell]}$ for $b' = a \oplus \mathsf{PRG}(s')$, where s' is its input and $a, \{x_i^0, x_i^1\}_{i\in[\kappa+\ell]}$ are hardwired within the circuit. She then computes $(\hat{C}, \{L_i^0, L_i^1\}_{i\in[\kappa]}) \leftarrow \mathsf{Garble}(C)$ where $\{L_i^0, L_i^1\}_{i\in[\kappa]}$ are the input labels.
3. $\mathcal{S}$ sends $\{L_i^0, L_i^1\}_{i\in[\kappa]}$ to $\mathbb{OT}_\kappa$ and sends $\hat{C}$ to $\mathcal{R}$.
4. $\mathcal{R}$ receives $\{L_i^{s_i}\}_{i\in[\kappa]}$ from $\mathbb{OT}_\kappa$ and $\hat{C}$ from $\mathcal{S}$. He uses the labels to evaluate $\hat{C}$ yielding $C_{[a,\{x_i^0,x_i^1\}_{i\in[\kappa+\ell]}]}(s) = \{x_i^{b_i}\}_{i\in[\kappa+\ell]}$ as his output.

It is straightforward to see that this protocol is perfectly correct. I.e., the receiver always produces the correct outputs.

Theorem 6. *Let* PRG *be* $\varepsilon_{\mathsf{PRG}}$ *indistinguishable from uniform and* Garble *have* $\varepsilon_{\mathsf{Garble}}$ *simulation security. Then, the proposed OT extension is* $\varepsilon_{\mathsf{PRG}}$ *simulatable for an honest but curious* **PPT** *sender and* $\varepsilon_{\mathsf{Garble}}$ *simulatable for an honest but curious* **PPT** *receiver.*

Proof. Let there be a honest but curious **PPT** receiver $\mathcal{R}$. Then, we construct a simulator $\mathsf{Sim}_{\mathcal{R}}$ that simulates up to $\varepsilon_{\mathsf{Garble}}$ adversary $\mathcal{R}$ in the ideal world. $\mathsf{Sim}_{\mathcal{R}}$ has access to an ideal functionality which he can query a single time on b to receive $\{x_i^{b'_i}\}_{i\in[\kappa+\ell]}$.

Notice that an honest but curious receiver will send $a = \mathsf{PRG}(s) \oplus b$ to the sender and s to the OT-hybrid functionality. In the end, $\mathcal{R}$ expects the output of the OT-hybrid functionality $\{L_i^{s_i}\}_{i\in[\kappa]}$ and a message from the sender, which is the garbled circuit $\hat{C}$. We construct a simulator $\mathsf{Sim}_{\mathcal{R}}$ as follows. $\mathsf{Sim}_{\mathcal{R}}$ receives a, s from A and computes $b' := a \oplus \mathsf{PRG}(s)$. Then, $\mathsf{Sim}_{\mathcal{R}}$ sends b' to the ideal functionality to receive $\{x_i^{b'_i}\}_{i\in[\kappa+\ell]}$. This will serve as input to the simulator of the garbling procedure Garble which outputs $\hat{C}$ together with the labels $\{L_i^{s_i}\}_{i\in[\kappa]}$ for input s. Note that the topology of C is public. Now $\mathsf{Sim}_{\mathcal{R}}$ computes its output, i.e. the view of $\mathcal{R}$ for input b', messages s, $\{L_i^{s_i}\}_{i\in[\kappa]}$ and $\hat{C}$.

Notice that both views are identical except to the distribution of $\{L_i^{s_i}\}_{i\in[\kappa]}$ and $\hat{C}$. To argue that the view of $\mathcal{R}$ and the output of $\mathsf{Sim}_{\mathcal{R}}$ are close, we use a distinguisher D against the circuit garbling. Let D' be a distinguisher that distinguishes $\mathbf{V}_{\mathcal{R}}$ and the output of $\mathsf{Sim}_{\mathcal{R}}$. D receives challenge $\hat{C}$ and $\{L_i^{s_i}\}_{i\in[\kappa]}$, which is either a partial output of $\mathsf{Garble}(C)$ or the output of $\mathsf{Sim}_{\mathsf{Garble}}(C(s))$. Simultaneously, D generates a view of $\mathcal{R}$ for $\hat{C}$, $\{L_i^{s_i}\}_{i\in[\kappa]}$ and random remaining inputs. D invokes D' on this view and outputs D''s output. Therefore the probability that D breaks the garbling is lower bounded by the probability that D' distinguishes the views. More formally,

$$\begin{aligned}\varepsilon_{\mathsf{Garble}} &= |\Pr[\mathbf{D}^{\mathsf{D}'}(\hat{C}, \{L_i^{s_i}\}_{i\in[\kappa]}) = 1] - \Pr[\mathbf{D}^{\mathsf{D}'}(\hat{C}', \{L'^{i}_{s_i}\}_{i\in[\kappa]}) = 1]| \\ &\geq |\Pr[\mathsf{D}'(\mathbf{V}_{\mathcal{R}}) = 1] - \Pr[\mathsf{D}'(\mathsf{Sim}_{\mathcal{R}}) = 1]|,\end{aligned}$$

where

$$(\hat{C}, \{L_i^0, L_i^1\}_{i\in[\kappa]}) \leftarrow \mathsf{Garble}(1^\kappa, C)$$

and $(\hat{C}', \{L'^{x_i}_i\}_{i\in[\kappa]}) \leftarrow \mathsf{Sim}(1^\kappa, C(x), \mathsf{top}(C))$.

For the security against a honest but curious sender $\mathcal{S}$, we construct a simulator $\mathsf{Sim}_{\mathcal{S}}$. Notice that $\mathcal{S}$ receives a message $a = \mathsf{PRG}(s) \oplus b$ and sends a garbled circuit $\hat{C}$ to $\mathcal{R}$ and its labels $\{L_i^0, L_i^1\}_{i\in[\kappa]}$. We construct a simulator $\mathsf{Sim}_{\mathcal{S}}$ as follows. $\mathsf{Sim}_{\mathcal{S}}$ chooses a uniformly at random. Since $\mathcal{S}$ follows the protocol, $\mathsf{Sim}_{\mathcal{S}}$ can evaluate $\hat{C}$ for $\{L_i^0\}_{i\in[\kappa]}$ and for $\{L_i^1\}_{i\in[\kappa]}$. Therefore $\mathsf{Sim}_{\mathcal{S}}$ will learn from these evaluations all input strings $\{y_i^0, y_i^1\}_{i\in[\kappa+\ell]}$, which $\mathsf{Sim}_{\mathcal{S}}$ will send to the ideal functionality $\mathcal{F}$. Similarly, we will show that the view of $\mathcal{S}$ and the output of $\mathsf{Sim}_{\mathcal{S}}$ are $\varepsilon_{\mathsf{PRG}}$ close. Notice that both views are identical except message a.

We construct a distinguisher D that will distinguish $\mathsf{PRG}(s)$ from uniform given a distinguisher D' for the view of $\mathcal{S}$ and the output of $\mathsf{Sim}_{\mathcal{S}}$. D receives

a challenge u which is either uniform or $\mathsf{PRG}(s)$ and an generates the view of $\mathcal{S}$ for a and random remaining inputs. D invokes D' on the generated view and outputs the output of D'. Hence, D breaks the security of PRG with probability

$$\begin{aligned}\varepsilon_{\mathsf{PRG}} &= |\Pr[\mathsf{D}^{\mathsf{D}'}(u) = 1; u = \mathsf{PRG}(s)] - \Pr[\mathsf{D}^{\mathsf{D}'}(u) = 1; u \text{ uniform}]| \\ &\geq |\Pr[\mathsf{D}'(\mathbf{V}_{\mathcal{R}}) = 1] - \Pr[\mathsf{D}'(\mathsf{Sim}_{\mathcal{R}}) = 1]|.\end{aligned}$$

□

References

AIR01. William Aiello, Yuval Ishai, and Omer Reingold. Priced oblivious transfer: How to sell digital goods. In Birgit Pfitzmann, editor, *EUROCRYPT 2001*, volume 2045 of *LNCS*, pages 119–135, Innsbruck, Austria, May 6–10, 2001. Springer, Heidelberg, Germany. 2

AJL+12. Gilad Asharov, Abhishek Jain, Adriana López-Alt, Eran Tromer, Vinod Vaikuntanathan, and Daniel Wichs. Multiparty computation with low communication, computation and interaction via threshold FHE. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 483–501, Cambridge, UK, April 15–19, 2012. Springer, Heidelberg, Germany. 2

Ald83. David Aldous. Random walks on finite groups and rapidly mixing markov chains. In *Séminaire de Probabilités XVII 1981/82*, pages 243–297. Springer, 1983. 10

ALSZ13. Gilad Asharov, Yehuda Lindell, Thomas Schneider, and Michael Zohner. More efficient oblivious transfer and extensions for faster secure computation. In Ahmad-Reza Sadeghi, Virgil D. Gligor, and Moti Yung, editors, *ACM CCS 13*, pages 535–548, Berlin, Germany, November 4–8, 2013. ACM Press. 2

ALSZ15. Gilad Asharov, Yehuda Lindell, Thomas Schneider, and Michael Zohner. More efficient oblivious transfer extensions with security for malicious adversaries. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part I*, volume 9056 of *LNCS*, pages 673–701, Sofia, Bulgaria, April 26–30, 2015. Springer, Heidelberg, Germany. 2

BCR87. Gilles Brassard, Claude Crépeau, and Jean-Marc Robert. All-or-nothing disclosure of secrets. In Andrew M. Odlyzko, editor, *CRYPTO'86*, volume 263 of *LNCS*, pages 234–238, Santa Barbara, CA, USA, August 1987. Springer, Heidelberg, Germany. 1

Bea96. Donald Beaver. Correlated pseudorandomness and the complexity of private computations. In *28th ACM STOC*, pages 479–488, Philadephia, PA, USA, May 22–24, 1996. ACM Press. 2, 3, 5, 9, 28

BGI+14. Amos Beimel, Ariel Gabizon, Yuval Ishai, Eyal Kushilevitz, Sigurd Meldgaard, and Anat Paskin-Cherniavsky. Non-interactive secure multiparty computation. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part II*, volume 8617 of *LNCS*, pages 387–404, Santa Barbara, CA, USA, August 17–21, 2014. Springer, Heidelberg, Germany. 4

BM17. Boaz Barak and Mohammad Mahmoody. Merkle's key agreement protocol is optimal: An $O(n^2)$ attack on any key agreement from random oracles. *Journal of Cryptology*, 30(3):699–734, Jul 2017. 4, 7, 10, 11, 15

BMG09. Boaz Barak and Mohammad Mahmoody-Ghidary. Merkle puzzles are optimal - an $O(n^2)$-query attack on any key exchange from a random oracle. In Shai Halevi, editor, *Advances in Cryptology - CRYPTO 2009, 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2009. Proceedings*, volume 5677 of *Lecture Notes in Computer Science*, pages 374–390. Springer, 2009. 4, 5, 7, 15

BMR90. Donald Beaver, Silvio Micali, and Phillip Rogaway. The round complexity of secure protocols (extended abstract). In *22nd ACM STOC*, pages 503–513, Baltimore, MD, USA, May 14–16, 1990. ACM Press. 2

CPS08. Jean-Sébastien Coron, Jacques Patarin, and Yannick Seurin. The random oracle model and the ideal cipher model are equivalent. In David Wagner, editor, *CRYPTO 2008*, volume 5157 of *LNCS*, pages 1–20, Santa Barbara, CA, USA, August 17–21, 2008. Springer, Heidelberg, Germany. 3

DLMM11. Dana Dachman-Soled, Yehuda Lindell, Mohammad Mahmoody, and Tal Malkin. On black-box complexity of optimally-fair coin-tossing. In *Theory of Cryptography Conference - TCC 2011*, 2011. 15

EGL85. Shimon Even, Oded Goldreich, and Abraham Lempel. A randomized protocol for signing contracts. *Communications of the ACM*, 28(6):637–647, 1985. 1, 4

Fan68. Robert M Fano. *Transmission of Information. A Ststistical Theory of Communications*. Mit Press, 1968. 13

FKN94. Uriel Feige, Joe Kilian, and Moni Naor. A minimal model for secure computation (extended abstract). In *26th ACM STOC*, pages 554–563, Montréal, Québec, Canada, May 23–25, 1994. ACM Press. 4

GGHR14. Sanjam Garg, Craig Gentry, Shai Halevi, and Mariana Raykova. Two-round secure MPC from indistinguishability obfuscation. In Yehuda Lindell, editor, *TCC 2014*, volume 8349 of *LNCS*, pages 74–94, San Diego, CA, USA, February 24–26, 2014. Springer, Heidelberg, Germany. 2

GKLM12. Vipul Goyal, Virendra Kumar, Satya Lokam, and Mohammad Mahmoody. On black-box reductions between predicate encryption schemes. *Theory of Cryptography*, pages 440–457, 2012. 12

GMPP16. Sanjam Garg, Pratyay Mukherjee, Omkant Pandey, and Antigoni Polychroniadou. The exact round complexity of secure computation. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 448–476, Vienna, Austria, May 8–12, 2016. Springer, Heidelberg, Germany. 2

GMW87. Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game or A completeness theorem for protocols with honest majority. In Alfred Aho, editor, *19th ACM STOC*, pages 218–229, New York City, NY, USA, May 25–27, 1987. ACM Press. 1

HK12. Shai Halevi and Yael Tauman Kalai. Smooth projective hashing and two-message oblivious transfer. *Journal of Cryptology*, 25(1):158–193, January 2012. 2

HKT11. Thomas Holenstein, Robin Künzler, and Stefano Tessaro. The equivalence of the random oracle model and the ideal cipher model, revisited. In Lance Fortnow and Salil P. Vadhan, editors, *43rd ACM STOC*, pages 89–98, San Jose, CA, USA, June 6–8, 2011. ACM Press. 3

HOZ16. Iftach Haitner, Eran Omri, and Hila Zarosim. Limits on the usefulness of random oracles. *Journal of Cryptology*, 29(2):283–335, 2016. 4, 5, 7, 15

IKM+13. Yuval Ishai, Eyal Kushilevitz, Sigurd Meldgaard, Claudio Orlandi, and Anat Paskin-Cherniavsky. On the power of correlated randomness in secure computation. In Amit Sahai, editor, *TCC 2013*, volume 7785 of *LNCS*, pages 600–620, Tokyo, Japan, March 3–6, 2013. Springer, Heidelberg, Germany. 4

IKNP03. Yuval Ishai, Joe Kilian, Kobbi Nissim, and Erez Petrank. Extending oblivious transfers efficiently. In Dan Boneh, editor, *CRYPTO 2003*, volume 2729 of *LNCS*, pages 145–161, Santa Barbara, CA, USA, August 17–21, 2003. Springer, Heidelberg, Germany. 2, 3, 4, 7, 16, 17

IPS08. Yuval Ishai, Manoj Prabhakaran, and Amit Sahai. Founding cryptography on oblivious transfer - efficiently. In David Wagner, editor, *CRYPTO 2008*, volume 5157 of *LNCS*, pages 572–591, Santa Barbara, CA, USA, August 17–21, 2008. Springer, Heidelberg, Germany. 1

IR89. Russell Impagliazzo and Steven Rudich. Limits on the provable consequences of one-way permutations. In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing (STOC)*, pages 44–61. ACM Press, 1989. 1, 3, 4, 5, 7, 15

Kil88. Joe Kilian. Founding cryptography on oblivious transfer. In *Proceedings of the 20th Annual ACM Symposium on Theory of Computing (STOC)*, pages 20–31, 1988. 1

KK13. Vladimir Kolesnikov and Ranjit Kumaresan. Improved OT extension for transferring short secrets. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 54–70, Santa Barbara, CA, USA, August 18–22, 2013. Springer, Heidelberg, Germany. 2

KO04. Jonathan Katz and Rafail Ostrovsky. Round-optimal secure two-party computation. In Matthew Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 335–354, Santa Barbara, CA, USA, August 15–19, 2004. Springer, Heidelberg, Germany. 2

Lin16. Yehuda Lindell. How to simulate it - A tutorial on the simulation proof technique. Cryptology ePrint Archive, Report 2016/046, 2016. http://eprint.iacr.org/2016/046. 16, 17

LZ13. Yehuda Lindell and Hila Zarosim. On the feasibility of extending oblivious transfer. In Amit Sahai, editor, *TCC 2013*, volume 7785 of *LNCS*, pages 519–538, Tokyo, Japan, March 3–6, 2013. Springer, Heidelberg, Germany. 4, 5, 16

MW16. Pratyay Mukherjee and Daniel Wichs. Two round multiparty computation via multi-key FHE. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 735–763, Vienna, Austria, May 8–12, 2016. Springer, Heidelberg, Germany. 2

NNOB12. Jesper Buus Nielsen, Peter Sebastian Nordholt, Claudio Orlandi, and Sai Sheshank Burra. A new approach to practical active-secure two-party computation. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 681–700, Santa Barbara, CA, USA, August 19–23, 2012. Springer, Heidelberg, Germany. 2

NP01. Moni Naor and Benny Pinkas. Efficient oblivious transfer protocols. In S. Rao Kosaraju, editor, *12th SODA*, pages 448–457, Washington, DC, USA, January 7–9, 2001. ACM-SIAM. 2

ORS15. Rafail Ostrovsky, Silas Richelson, and Alessandra Scafuro. Round-optimal black-box two-party computation. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part II*, volume 9216 of *LNCS*, pages

339–358, Santa Barbara, CA, USA, August 16–20, 2015. Springer, Heidelberg, Germany. 2

PVW08. Chris Peikert, Vinod Vaikuntanathan, and Brent Waters. A framework for efficient and composable oblivious transfer. In David Wagner, editor, *CRYPTO 2008*, volume 5157 of *LNCS*, pages 554–571, Santa Barbara, CA, USA, August 17–21, 2008. Springer, Heidelberg, Germany. 2

Rab81. M. Rabin. How to exchange secrets by oblivious transfer. Technical Report TR-81, Harvard Aiken Computation Laboratory, 1981. 1

Sho09. Victor Shoup. *A computational introduction to number theory and algebra*. Cambridge university press, 2009. 10

WW06. Stefan Wolf and Jürg Wullschleger. Oblivious transfer is symmetric. In Serge Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 222–232, St. Petersburg, Russia, May 28 – June 1, 2006. Springer, Heidelberg, Germany. 17

Yao82. Andrew Chi-Chih Yao. Protocols for secure computations (extended abstract). In *23rd FOCS*, pages 160–164, Chicago, Illinois, November 3–5, 1982. IEEE Computer Society Press. 1

Yao86. Andrew Chi-Chih Yao. How to generate and exchange secrets (extended abstract). In *27th FOCS*, pages 162–167, Toronto, Ontario, Canada, October 27–29, 1986. IEEE Computer Society Press. 27