On the Feasibility of Extending Oblivious Transfer*

Yehuda Lindell and Hila Zarosim

Dept. of Computer Science
Bar-Ilan University, Israel
lindell@biu.ac.il,zarosih@cs.biu.ac.il

December 24, 2012

Abstract. Oblivious transfer is one of the most basic and important building blocks in cryptography. As such, understanding its cost is of prime importance. Beaver (STOC 1996) showed that it is possible to obtain $\mathsf{poly}(n)$ oblivious transfers given only n actual oblivious transfer calls and using one-way functions, where n is the security parameter. In addition, he showed that it is impossible to extend oblivious transfer information theoretically. The notion of extending oblivious transfer is important theoretically (to understand the complexity of computing this primitive) and practically (since oblivious transfers can be expensive and thus extending them using only one-way functions is very attractive).
Despite its importance, very little is known about the feasibility of extending oblivious transfer, beyond the fact that it is impossible information theoretically. Specifically, it is not known whether or not one-way functions are actually necessary for extending oblivious transfer, whether or not it is possible to extend oblivious transfers with adaptive security, and whether or not it is possible to extend oblivious transfers when starting with $O(\log n)$ oblivious transfers. In this paper, we address these questions and provide almost complete answers to all of them. We show that the existence of any oblivious transfer extension protocol with security for static semi-honest adversaries implies one-way functions, that an oblivious transfer extension protocol with adaptive security implies oblivious transfer with static security, and that the existence of an oblivious transfer extension protocol from only $O(\log n)$ oblivious transfers implies oblivious transfer itself.

1 Introduction

Background – extending oblivious transfer. In the oblivious transfer problem [17, 5], a sender holds a pair of input bits (b_0, b_1) and enables a receiver to obtain one of them at its choice. The security requirements are that the sender

* This research was supported by the Israel Science Foundation (grant No. 189/11). Hila Zarosim is grateful to the Azrieli Foundation for the award of an Azrieli Fellowship.

learns nothing about which input is obtained by the receiver, while the receiver learns only one bit.

Oblivious transfer is one of the most basic and important primitives in cryptography in general, and in secure computation in particular. Oblivious transfer is used in almost all general protocols for secure computation with no honest majority (e.g., see [20, 7]), and has been shown to imply essentially all basic cryptographic tasks [13]. Due to its importance, the complexity of computing oblivious transfer is of great importance. Oblivious transfer can be constructed from enhanced trapdoor permutations [5, 9] and from homomorphic encryption [1]. In addition, it is known that it is not possible to construct oblivious transfer from public-key encryption (or one-way functions and permutations) in a black-box manner [6]. Thus, oblivious transfer requires quite strong hardness assumptions (at least when considering black-box constructions, and no nonblack-box constructions from weaker assumptions are known).

Due to the importance of oblivious transfer and its cost, Beaver asked whether or not it is possible to use a small number of oblivious transfers and a weaker assumption like one-way functions in order to obtain many oblivious transfers [3]; such a construction is called an $\mathsf{OT\ extension}$. Beaver answered this question in the affirmative and in a beautiful construction showed how to obtain $\mathsf{poly}(n)$ oblivious transfers given ideal calls to $O(n)$ oblivious transfers and using a pseudorandom generator and symmetric encryption, which can both be constructed from any one-way function. In addition, he showed that OT extensions cannot be achieved information theoretically. These results of [3] are of great importance theoretically since they deepen our understanding of the complexity of oblivious transfer. In addition, OT extensions are of interest practically, since oblivious transfer is much more expensive than symmetric primitives. Thus, OT extensions can potentially be used to speed up protocols that rely on many oblivious transfers. In this direction, efficient OT extensions (based on a stronger assumption than one-way functions) were presented in [11].

This paper – a feasibility study of OT extensions. In this paper, we ask the following questions:

1. *What is the minimal assumption required for constructing OT extensions?* It has been shown that one-way functions suffice, and that OT extensions cannot be carried out information theoretically [3]. However, it is theoretically possible that OT extensions can be achieved under a weaker assumption than that of the existence of one-way functions. Admittedly, it is hard to conceive of a cryptographic construction that is not information theoretic and does not require one-way functions. However, a proof that one-way functions really are necessary is highly desired.
2. *Can oblivious transfer be extended with adaptive security?* The known constructions of OT extensions maintain security only in the presence of static corruptions, where the set of corrupted parties is fixed before the protocol begins. This is because the messages sent by the sender in the constructions of [3, 11] are binding with respect to the sender's input strings, and so an adaptive simulator cannot explain a transcript in multiple ways. Nothing is

known about whether or not adaptively secure OT extensions exist without assuming erasures[1].

3. *How many oblivious transfers are needed for extensions?* In the constructions of [3, 11], one must start with $O(n)$ oblivious transfers where n is the security parameter. These constructions can also be made to work when a superlogarithmic number $\omega(\log n)$ of oblivious transfers are given. However, they completely break down if $O(\log n)$ oblivious transfers only are available. We ask whether or not it is possible to extend a logarithmic number of oblivious transfers.

We prove the following theorems:

Theorem 1.1 *If there exists an OT extension protocol from n to $n+1$ (with security in the presence of static semi-honest adversaries), then there exist one-way functions.*

Thus, one-way functions are *necessary and sufficient* for OT extensions.

Theorem 1.2 *If there exists an OT extension protocol from n to $n+1$ that is secure in the presence of* adaptive *semi-honest adversaries, then there exists an oblivious transfer protocol that is secure in the presence of static semi-honest adversaries.*

This means that the construction of an adaptive OT extension protocol involves constructing statically secure oblivious transfer from scratch. This can still be meaningful, since adaptive oblivious transfer cannot be constructed from static oblivious transfer in a black-box manner [15]. However, it does demonstrate that adaptive OT extensions based on weaker assumptions than those necessary for static oblivious transfer do not exist.

Theorem 1.3 *If there exists an OT extension protocol from $f(n) = \mathcal{O}(\log n)$ to $f(n)+1$ that is secure in the presence of static malicious adversaries, then there exists an oblivious transfer protocol that is secure in the presence of static malicious adversaries.*

This demonstrates that in order to extend only a logarithmic number of oblivious transfers (with security for *malicious* adversaries), one has to construct an oblivious transfer protocol from scratch. Thus, meaningful OT extensions exist only if one starts with a superlogarithmic number of oblivious transfers.

We stress that all of our results are unconditional, and are not black-box separations. Rather, we construct concrete one-way functions and OT protocols in order to prove our results.

Our results provide quite a complete picture regarding the feasibility of constructing OT extensions. The construction of [3] is optimal in terms of the computational assumption, and the constructions of [3, 11] are optimal in terms of

[1] Note that in the erasures model, an OT extension can be constructed from one-way functions using the original construction of Beaver and the two-party computation protocol of [14] that is adaptively secure with erasures and is based on Yao's protocol.

the number of oblivious transfers one starts with. Finally, the fact that no OT extensions are known for the setting of adaptive corruptions is somewhat explained by Theorem 2.

Open questions. Theorem 2 shows that there do not exist adaptively secure OT extensions based on weaker assumptions than what is needed for *statically secure* OT. However, we do not know how to construct an adaptively secure OT extension even from statically secure OT. Thus, the question of whether or not it is possible to construct an adaptively secure OT extension from an assumption weaker than adaptive OT is still open.

Theorem 3 holds only with respect to OT-extensions that are secure against *malicious* adversaries. For the case of semi-honest adversaries, the question of whether one can construct an an OT-extension from $f(n) = \mathcal{O}(\log n)$ to $f(n)+1$ from an assumption weaker than statically secure OT protocol is open.

In this paper, we have investigated OT extensions. However, the basic question of extending a cryptographic primitive using a weaker assumption than that needed for obtaining the primitive from scratch is of interest in other contexts as well. For example, hybrid encryption (where one encrypts a symmetric key using an asymmetric scheme, and then encrypts the message using a symmetric scheme) is actually an extension of public-key encryption that requires one-way functions only.

A primitive that could certainly benefit from a study such as this one is *key agreement*. In this context, the question is whether it is possible for two parties to agree on an $m + 1$-bit long key, given an m-bit key, under assumptions that are weaker than those required for constructing a secure key-agreement from scratch. In the basic case, it is clear that OWFs are necessary and sufficient for any nontrivial KA extension that starts with n bits (where n is the security parameter). A more interesting question regarding this problem relates to the adaptive setting. Specifically, since adaptive key agreement is very expensive, it would be very beneficial if one could extend this primitive more efficiently and/or under weaker assumptions.

2 Definitions and Notations

We denote the security parameter by n, and we denote by U_n a random variable uniformly distributed over $\{0,1\}^n$. We say that a function $\mu : \mathbb{N} \to \mathbb{N}$ is negligible if for every positive polynomial $p(\cdot)$ and all sufficiently large n it holds that $\mu(n) < \frac{1}{p(n)}$. We use the abbreviation PPT to denote probabilistic polynomial-time. We denote the bits of a string $x \in \{0,1\}^n$ by $x_1, \ldots, x_n$; for a subscripted string x_b, we denote the bits by $x_b^1, \ldots, x_b^n$. In addition, for strings $x_0, x_1, \sigma \in \{0,1\}^n$ we denote by x_σ the string $x_{\sigma_1}^1, \ldots, x_{\sigma_n}^n$.

For two distribution ensembles $X = \{X(a,n)\}$ and $Y = \{Y(a,n)\}$ with $a \in \{0,1\}^*$ and $n \in \mathbb{N}$, we write $X \stackrel{\text{c}}{\equiv} Y$ if they are computationally indistinguishable, and we write $X \stackrel{\text{s}}{\equiv} Y$ if they are statistically close. We also denote by $SD(X,Y)$ the statistical distance between X and Y.

Interactive Protocols. Let $\pi = \langle A, B \rangle$ be an interactive protocol for computing a functionality f. We denote $f = (f_A, f_B)$, where f_A is the first output of f (for party A) and f_B is the second output of f (for party B).

The random variable $\text{VIEW}_A^\pi(x_A, x_B)$ denotes the view of the party A in an execution of π with inputs x_A for A and x_B for B, where the random tapes of the parties are uniformly chosen. Note that a view of a party contains its input, randomness and the messages it has received during the execution.

The random variable $\text{OUTPUT}_A^\pi(x_A, x_B)$ denotes the output of the party A in an execution of π with inputs x_A for A and x_B for B, where the random tapes of the parties are uniformly chosen.

Definition 2.1 *Let $f(\cdot, \cdot)$ be a deterministic binary functionality, let $\pi = \langle A, B \rangle$ be an interactive protocol and let n be the security parameter. We say that π* computes the functionality f *if there exists a negligible function* $\mathsf{negl}(\cdot)$ *such that for all n, x_A and x_B:*

$$\Pr\left[\langle A(1^n, x_A), B(1^n, x_B)\rangle = (f_A(x_A, x_B), f_B(x_A, x_B))\right] \geq 1 - \mathsf{negl}(n).$$

Definition 2.2 *Let $\pi = \langle A, b \rangle$ be a protocol that computes a deterministic functionality $f = (f_A, f_B)$. Protocol π* securely computes f in the presence of static semi-honest adversaries *if there exist two PPT algorithms $\mathcal{S}_A$ and $\mathcal{S}_B$ such that: $\{\mathcal{S}_A(1^n, x_A, f_A(x_A, x_B))\} \stackrel{c}{\equiv} \{\text{VIEW}_A^\pi(1^n, x_A, x_B)\}$ and $\{\mathcal{S}_B(1^n, x_B, f_B(x_A, x_B))\} \stackrel{c}{\equiv} \{\text{VIEW}_B^\pi(1^n, x_A, x_B)\}$ where $x_A, x_B \in \{0,1\}^*$ and $n \in \mathbb{N}$.*

Security in the presence of malicious adversaries. To define security in the presence of malicious adversaries, we use the ideal/real framework as defined by Canetti in [4]. Loosely speaking, in this approach we formalize the real-life computation as a setting where the parties, given their private inputs, interact according to the protocol in the presence of a real-life adversary that controls a set of corrupted parties. The real-life adversary can be either static (where the set of corrupted parties is fixed before the protocol begins) or adaptive (where the adversary can choose to corrupt parties during the protocol execution based on what it sees). At the end of the computation, the honest parties output what is specified by the protocol and the adversary outputs some arbitrary function of its view. If the adversary is adaptive, there is an additional entity $\mathcal{Z}$, called the environment, who sees the output of all of the parties. In addition, there is a "postexecution phase", where $\mathcal{Z}$ can instruct the adversary to also corrupt parties after the execution of the protocol ends (and the transcript is fixed, implying that "rewinding" is no longer allowed). At the end of the postexecution phase, $\mathcal{Z}$ outputs some function of its view.

Next we consider an ideal process, where an ideal-world adversary controls a set of corrupted parties. Then, in the computation phase, all parties send their inputs to some incorruptible trusted party. The ideal-world adversary sends inputs on behalf of the corrupted parties. The trusted party evaluates the function and hands each party its output. The honest parties then output whatever they

received from the trusted party and the ideal-world adversary outputs some arbitrary value. Similarly to the real-life setting, in the case of adaptive security, there is an environment $\mathcal{Z}$ who sees all outputs and can instruct the adversary to also corrupt parties in the postexecution phase. At the end of the postexecution phase, $\mathcal{Z}$ outputs some function of its view.

Loosely speaking, a protocol π is secure in the presence of static malicious adversaries, if for every static malicious real-life adversary $\mathcal{A}$, there exists a static malicious ideal-world adversary $\mathcal{SIM}$ such that the distribution obtained in a real-life execution of π with adversary $\mathcal{A}$ is indistinguishable from the distribution obtained in a ideal-world with adversary $\mathcal{SIM}$. Likewise, a protocol π is secure in the presence of adaptive malicious adversaries, if for every adaptive malicious real-life adversary $\mathcal{A}$ and environment $\mathcal{Z}$, there exists an adaptive malicious ideal-world adversary $\mathcal{SIM}$ such that the output of $\mathcal{Z}$ in a real-life execution of π with adversary $\mathcal{A}$ is indistinguishable from its output in a ideal-world with adversary $\mathcal{SIM}$.

Security in the presence of adaptive semi-honest adversaries is defined in the same way as adaptive malicious adversaries, except that the adversary only sees the internal state of a corrupted party but cannot instruct it to deviate from the protocol specification. For full definitions see [4].

The hybrid model. Let ϕ be a functionality. The ϕ-hybrid model is defined as follows. The real-life model for protocol π is augmented with an incorruptible trusted party T for evaluating the functionality ϕ, and the parties are allowed to make calls to the ideal functionality ϕ by sending their ϕ-inputs to T. If we consider malicious adversaries, the adversary specifies the inputs of all parties under its control. If the adversary is semi-honest, then even the corrupted parties hand T inputs as specified by the protocol π. At each invocation of ϕ, the trusted party T sends the parties their respective outputs.

We stress that if π is in the ϕ-hybrid model, then a view of a party A contains also the inputs sent by A to the functionality ϕ and the outputs sent to A by T computing ϕ.

Oblivious transfer and extensions. We are now ready to define oblivious transfer and OT extensions.

Definition 2.3 *The* bit oblivious transfer functionality OT *is defined by* $OT((b_0, b_1), \sigma) = (\lambda, b_\sigma)$. *The* parallel oblivious transfer functionality $m \times OT$ *is defined for strings* $x_0, x_1, \sigma \in \{0,1\}^m$ *as follows:* $m \times OT((x_0, x_1), \sigma) = (\lambda, (x^1_{\sigma_1}, \ldots, x^m_{\sigma_m})) = (\lambda, x_\sigma)$ *(recall that* x_σ *denotes the string* $x^1_{\sigma_1}, \ldots, x^n_{\sigma_n}$*).*

We denote by OT^k the ideal functionality of k independent OT computations. We stress that OT^k is not the same as $k \times OT$, since in the latter all of the inputs are given at once whereas in OT^k the inputs can be chosen over time (in particular, the receiver can choose its inputs as a function of the previous outputs it received). Using this notation, we have that an OT extension protocol is a protocol that securely computes $m \times OT$ given access to OT^k, where $k < m$. Formally:

Definition 2.4 (OT-extension) *Let π be a protocol and let $k, m : \mathbb{N} \to \mathbb{N}$ be two functions where $k(n) < m(n)$ for all n. We say that π is an* OT-extension from $k = k(n)$ to $m = m(n)$ *if π securely computes the $m \times OT$ functionality in the OT^k-hybrid model.*

OT extensions – two technical propositions. We present two propositions that we use throughout the paper. Beaver showed that OT can be precomputed [2]. That is, it is possible to first compute OT on random inputs and then use the result to later compute an OT on any input. Stated formally:

Proposition 2.5 (Beaver [2]) *Let $m = m(n)$ be a polynomial. If there exists a protocol that securely computes the $m \times OT$ functionality, then there exists a protocol that securely computes the OT^m ideal functionality.*

Proposition 2.5 shows that Definition 2.4 could have been stated as a protocol that securely computes OT^m in the OT^k (or even the $k \times OT$) hybrid model.

The fact that a single extension implies many has been stated many times in the literature (e.g., [3]) and is well accepted folklore, but has not been formally proved. In the full version of this paper [16], we sketch a proof of this. We stress that this holds irrespectively of how many oblivious transfers you start with (even if only a *constant number*), as long as only a polynomial number of transfers are derived. We state the proposition for adaptive malicious adversaries and observe that it holds for all four combinations of static/adaptive and semi-honest/malicious adversaries.

Proposition 2.6 *Let $f : \mathbb{N} \to \mathbb{N}$ be* any *polynomially-bounded function, and let n be the security parameter. If there exists a protocol π that is an OT-extension from $f(n)$ to $f(n)+1$ that is secure in the presence of adaptive malicious adversaries, then for every polynomial $p(\cdot)$ there exists an OT-extension protocol from $f(n)$ to $p(n)$ that is secure in the presence of adaptive malicious adversaries.*

3 OT Extensions Imply One-Way Functions

In this section we show that the existence of an OT extension protocol implies the existence of one-way functions. We prove the theorem for any OT extension that is secure in the presence of static semi-honest adversaries (thus the theorem also holds when the OT extension is secure in the presence of adaptive and/or malicious adversaries, since these variants all imply security for static semi-honest adversaries).

Theorem 3.1 *Let n be the security parameter. If there exists a protocol that is an OT-extension from n to $n+1$ that is secure in the presence of static semi-honest adversaries, then there exist one-way functions.*

Proof Sketch: To prove this, we use an information-theoretic lower bound given in [18] to show that the existence of a protocol π that is an OT-extension

from n to $n+1$ implies the existence of two polynomial-time constructible probability ensembles that are computationally indistinguishable and yet their statistical distance is noticeable. The fact that this implies one-way functions was shown in [8].

We define two polynomial-time constructible probability ensembles $\mathsf{RL} = \{\mathsf{RL}_n\}_{n\in\mathbb{N}}$ and $\mathsf{SM} = \{\mathsf{SM}_n\}_{n\in\mathbb{N}}$ that are computationally indistinguishable, but have noticeable statistical distance. Let $\mathcal{S}_S$ and $\mathcal{S}_R$ be the two simulators that are guaranteed to exist for π by its semi-honest security. We begin by defining the probability ensembles RL and SM, that represent the real and the simulated transcripts, respectively.

RL_n: First, a party $P \in \{S, R\}$ is chosen at random. Then, inputs for both parties $x_0, x_1, \sigma \in \{0,1\}^n$ are chosen uniformly at random and the real protocol π is executed on inputs (x_0, x_1) for the sender and σ for the receiver. The output of RL_n is a pair (v, ω) where v is the view of party P in the execution described above and ω is the output of the other party.

SM_n: Similarly to the above, a party $P \in \{S, R\}$ and inputs $x_0, x_1, \sigma \in \{0,1\}^n$ are chosen uniformly at random. Then, the simulator $\mathcal{S}_P$ (that is, $\mathcal{S}_R$ is $P = R$ and $\mathcal{S}_S$ if $P = S$) is executed on the corresponding input and output of party P. The output of SM_n is a pair (v, ω) where v is the view generated by the simulator and ω is the output of the other party as defined by the functionality.

We now prove that the ensembles RL and SM are computationally indistinguishable but statistically far. The fact that they are computationally indistinguishable can be derived from the (computational) security of π. Specifically, for every $P \in \{S, R\}$, it holds that the view generated by the simulator $\mathcal{S}_P$ is computationally indistinguishable from a real view of P in an execution of π, and hence it can be easily shown that RL and SM are computationally indistinguishable. Intuitively, the fact that the two ensembles are statistically far apart follows from the fact that OT cannot be extended with statistical security [3] and so the ensembles cannot be statistically close. However, this argument is not sufficient, because it only implies that RL and SM are *not statistically close*, whereas what we need to show is that the two ensembles are *statistically far apart*. Specifically, the impossibility result of [3] only shows that there exists a polynomial $p(\cdot)$ such that *for infinitely many n's*, the statistical distance between RL_n and SM_n is $\frac{1}{p(n)}$, while the existence of one-way functions as proven in [8] only follows if there exists a polynomial $p(\cdot)$ such that *for all sufficiently large n's*, the statistical distance between RL_n and SM_n is $\frac{1}{p(n)}$. We therefore use the recent *non-asymptotic bound* on the statistical distance shown by [18], and use it to derive the following:

Claim 3.2 *There exists a polynomial $p(\cdot)$ such that for all sufficiently large n's the statistical distance between RL_n and SM_n is at least $1/p(n)$. Stated differently, the ensembles RL and SM have noticeable statistical distance.*

The proof of Claim 3.2 appears in [16]. Applying [8], as mentioned above, we conclude that one-way functions exist, and this concludes the proof sketch. ∎

4 Adaptive Security

In this section we consider the feasibility of constructing OT-extension protocols that are secure in the presence of adaptive adversaries. It is easy to see that the OT-extension protocols of Beaver [3] and Ishai et al. [11] are not secure when considering adaptive security. This is because the receiver's view is essentially a binding commitment to all of the sender's inputs.[2] This raises the question as to whether there exists an OT extension protocol at all in the presence of adaptive adversaries. Of course, if the existence of an OT extension protocol (that is secure for adaptive adversaries) implies OT that is secure for adaptive adversaries, then this means that only a trivial OT extension that constructs OT from scratch exists. We provide a partial answer to this question and show that a protocol for OT-extension that is secure in the presence of adaptive adversaries implies the existence of an OT protocol that is secure in the presence of *static* adversaries. Thus, any protocol for extending OT that maintains adaptive security needs to assume, at the very least, the existence of a statically secure protocol for OT. We state and prove this for semi-honest adversaries; an analogous theorem for malicious adversaries can be obtained by applying a GMW-type compiler. Formally, we prove the following theorem (the intuition appears immediately after Protocol 4.2 below):

Theorem 4.1 *Let n be the security parameter. If there exists an OT-extension protocol from n to $n+1$ that is secure in the presence of adaptive semi-honest adversaries, then there exists an OT protocol that is secure in the presence of static semi-honest adversaries.*

Proof. We prove the theorem by building an OT protocol that is secure in the presence of static adversaries from any OT extension from n to $4n$ that is secure in the presence of adaptive adversaries. (Note that by Proposition 2.6, an OT extension from n to $4n$ exists if there exists an extension from n to $n+1$.) We first present the construction of the OT protocol for static adversaries and then provide intuition as to why it is secure.

Let $\pi = \langle S, R \rangle$ be a protocol that securely computes the $4n \times OT$ functionality in the OT^n-hybrid model in the presence of adaptive semi-honest adversaries. We assume that all of the ideal calls to OT in π are such that S plays the sender and R plays the receiver. This is without loss of generality since the roles in OT can always be reversed [19]. We construct an OT protocol $\hat{\pi}$ in the plain model (i.e., with no calls to an ideal OT functionality), as follows:

Protocol 4.2 (OT protocol $\hat{\pi} = \langle \hat{S}, \hat{R} \rangle$ for Static Adversaries)

– **Inputs:** *Sender $\hat{S}$ has $b_0, b_1 \in \{0,1\}$ and receiver $\hat{R}$ has $\sigma \in \{0,1\}$.*

[2] In [3] a Yao garbled circuit is used which is binding when instantiated with known encryption methods. Likewise, [11] uses correlation-robust hash functions for which it is hard to find collisions, which is exactly what is needed in order to "explain the transcript" in different ways as is needed for proving adaptive security.

– **The protocol:**

1. *$\hat{S}$ chooses two random strings $\alpha_0, \alpha_1 \in \{0,1\}^{4n}$.*
2. *$\hat{S}$ and $\hat{R}$ run the extension protocol π as follows:*
   (a) *$\hat{S}$ plays the sender S in π with inputs (α_0, α_1).*
   (b) *$\hat{R}$ plays R in π with input σ^{4n} (i.e., the string of length $4n$ with all bits set to σ).*
   (c) *The parties follow the instructions of π exactly except that whenever π instructs them to make an ideal call to the OT functionality with input (β_0, β_1) for S and input τ for R, the sender $\hat{S}$ sends the pair (β_0, β_1) to $\hat{R}$, and $\hat{R}$ proceeds to run R with output β_τ from the simulated ideal call.*
   (d) *Let $\gamma \in \{0,1\}^{4n}$ denote the output of R in the execution of π.*
3. *$\hat{S}$ chooses two random strings $r_0, r_1 \in_R \{0,1\}^{4n}$ and sets:*

$$z_0 = \langle \alpha_0, r_0 \rangle \oplus b_0 \quad \text{and} \quad z_1 = \langle \alpha_1, r_1 \rangle \oplus b_1.$$

   $\hat{S}$ sends (r_0, z_0) and (r_1, z_1) to $\hat{R}$.

– **Output:** *$\hat{R}$ outputs $z_\sigma \oplus \langle \gamma, r_\sigma \rangle$.*

It is clear that $\hat{\pi}$ correctly computes the *OT* functionality. This is because by the correctness of the *OT* extension protocol, R will output $\gamma = \alpha_\sigma$ in Step 2d, except with negligible probability. Thus, $z_\sigma \oplus \langle \gamma, r_\sigma \rangle = z_\sigma \oplus \langle \alpha_\sigma, r_\sigma \rangle = b_\sigma$, as required.

We proceed to prove that π securely computes the *OT* functionality in the presence of semi-honest adversaries. We begin with the intuition. If $\hat{S}$ and $\hat{R}$ were to run the original extension protocol π with the ideal calls, then it is clear that $\hat{\pi}$ is a secure *OT* protocol. This is because $\hat{S}$ learns nothing about σ, and $\hat{R}$ learns α_σ but nothing about $\alpha_{1-\sigma}$. Thus, $\hat{R}$ learns b_σ but nothing about $b_{1-\sigma}$ (observe that $\langle \alpha_{1-\sigma}, r_{1-\sigma} \rangle$ hides $b_{1-\sigma}$ by the fact that $\alpha_{1-\sigma}$ is random). Now, in $\hat{\pi}$ the difference is that $\hat{S}$ sends both inputs to $\hat{R}$ in every ideal *OT* call within the execution of π. Clearly, $\hat{S}$'s view can be simulated since its view is identical to the case that π with the ideal *OT* calls is used. In contrast, $\hat{R}$ learns more information since it obtains both sender inputs in all ideal *OT* calls. Since the inputs to each ideal call are a single bit, we have that $\hat{R}$ obtains n more bits of information than in the original extension protocol using ideal *OT* calls. However, $\alpha_{1-\sigma}$ is $4n$ bits long and so still must have high entropy even given the n additional bits of information learned. This entropy is enough to hide $b_{1-\sigma}$ since $\langle \alpha_{1-\sigma}, r_{1-\sigma} \rangle$ is a perfect universal hash function, and so a good randomness extractor.

The above seems to have nothing to do with the fact that the extension protocol π is secure in the presence of *adaptive adversaries*. However, the argument that just n more bits of information are obtained is valid only in this case. Specifically, by the definition of security in the presence of adaptive adversaries, the simulator must be able to simulate in the case that the receiver is corrupted at the onset, and the sender is corrupted at the end after the protocol concludes

(formally, in the "post-execution corruption phase"). This means that the simulator must first generate a receiver-view (given the receiver's input and output), and must then later generate a sender-view (given the sender's input) that is consistent with the *already fixed* receiver-view that it previously generated. This sender-view contains, amongst other things, the inputs that the sender uses in all of the n ideal calls to the OT functionality within the extension protocol π. Thus, it is possible to add these inputs of the sender to the previously generated receiver-view (we call this the extended receiver view) and the result is the receiver-view in the modified extension protocol used in Step 2 of $\hat{\pi}$; in particular, both sender's inputs to all ideal OT calls appear. Observe that only n bits of additional information are added to the receiver view in order to obtain the extended view, and so there are at most 2^n extended views for any given receiver view. However, there are 2^{4n} different possible strings $\alpha_{1-\sigma}$. The crucial point here is that the above implies that many different possible strings $\alpha_{1-\sigma}$ must be consistent with any given extended view (except with negligible probability). This relies critically on the fact that the receiver-view is fixed before the sender corruption and so the same extended receiver-view must be consistent with many different sender inputs to the ideal OT calls. Now, once we have that many different possible $\alpha_{1-\sigma}$ strings are consistent, we can use the fact that $\alpha_{1-\sigma}$ is randomly chosen to apply the leftover hash lemma and conclude that $\langle \alpha_{1-\sigma}, r_{1-\sigma} \rangle$ is a bit that is statistically close to uniform. We now proceed to the formal proof.

Corrupted sender: The case of a corrupted sender is straightforward since the sender $\hat{S}$ receives no information in Step 2 of $\hat{\pi}$ beyond what it receives in a real execution of π with ideal OT calls. Thus the simulator that is assumed to exist for the sender S in π can be used to generate the exact view of $\hat{S}$ in Step 2 of $\hat{\pi}$. Since $\hat{S}$ receives no messages beyond in Step 2, there is nothing more to be added to the view of $\hat{S}$.

Corrupted receiver: In order to construct our simulator $\mathcal{S}_{\hat{R}}$ for the corrupted receiver $\hat{R}$ in $\hat{\pi}$, we first define a specific simulator $\mathcal{SIM}$ for the extension protocol π for the adaptive setting. Let $\mathcal{A}$ and $\mathcal{Z}$ be the following real-life semi-honest adversary and environment for π; see Section 2 for a brief overview of the definition of adaptive security, and [4] for full definitions. At the beginning of the execution of π, the adversary $\mathcal{A}$ corrupts the receiver and learns its input $\sigma \in \{0,1\}^{4n}$. It then follows the honest strategy for R and at the end of the execution, outputs its entire view. In the post-execution phase, $\mathcal{Z}$ generates a "corrupt S" message, sends it to $\mathcal{A}$ who corrupts S and hands $\mathcal{Z}$ the internal view of S. $\mathcal{Z}$ then outputs its internal view (note that it contains views of both R and S). Let $\mathcal{SIM}$ be the ideal-process adversary that is guaranteed to exist for this $\mathcal{A}$ and $\mathcal{Z}$ by the security of π. We remark that $\mathcal{SIM}$ generates a view of an execution of π in the OT-hybrid model, where ideal calls are used for the n invocations of OT. We use $\mathcal{SIM}$ to construct the simulator $\mathcal{S}_{\hat{R}}$ for the case of a corrupted receiver in $\hat{\pi}$.

Construction 4.3 ($\mathcal{S}_{\hat{R}}$) *$\mathcal{S}_{\hat{R}}$ receives σ and b_σ as input and works in three stages as follows:*

1. Stage 1 – obtain simulated receiver-view in π:
   (a) *Choose a random string $\alpha_\sigma \in_R \{0,1\}^{4n}$ as the "output of π" and a random tape $r_{\mathcal{SIM}}$ for $\mathcal{SIM}$ of the appropriate length.*
   (b) *Start an execution of $\mathcal{SIM}$ with random-tape $r_{\mathcal{SIM}}$. When $\mathcal{SIM}$ corrupts the receiver, hand σ^{4n} to $\mathcal{SIM}$ as the input of R.*
   (c) *In the computation stage, play the role of the trusted party and send α_σ to $\mathcal{SIM}$ as the output of R from $4n \times OT$. (Since we are in the semi-honest setting, R always sends its specified input σ^{4n} and so the output that it would receive is always α_σ.)*
   (d) *Let v_R be the output of $\mathcal{SIM}$ at the end of the execution phase (this consists of a view for the receiver). If v_R is not consistent with σ^{4n} and α_σ,[3] return $\perp$ and abort. Otherwise, proceed to the next stage.*
2. Stage 2 – obtain extended receiver-view:
   (a) *Choose a random string $\alpha_{1-\sigma} \in \{0,1\}^{4n}$.*
   (b) *Send a "corrupt S" message to $\mathcal{SIM}$ on behalf of $\mathcal{Z}$. When $\mathcal{SIM}$ corrupts the sender, hand (α_0, α_1) to $\mathcal{SIM}$ as the input of S.*
   (c) *Let v_S be the view of the sender sent by $\mathcal{SIM}$ to $\mathcal{Z}$. If v_S is not consistent with v_R and the inputs, output $\perp$ and abort. If v_S is consistent with v_R and the inputs, then for each of the n calls for the ideal OT functionality, extend v_R by appending the other input used by the sender (as appear in v_S) into the view v_R (note that v_R already contains one of the inputs used by the sender in each call since the receiver receives one output in each ideal call). Let v'_R be the extended view.*
3. Stage 3 – complete simulation:
   (a) *Choose two random strings $r_0, r_1 \in \{0,1\}^{4n}$; let $z_\sigma = \langle \alpha_\sigma, r_\sigma \rangle \oplus b_\sigma$ (where b_σ is from the input of $\mathcal{S}_{\hat{R}}$) and let $z_{1-\sigma}$ be a random bit.*
   (b) *Output v'_R, r_0, r_1, z_0, z_1.*

We prove that:

$$\left\{\mathcal{S}_{\hat{R}}(1^n, \sigma, b_\sigma)\right\}_{b_0,b_1,\sigma\in\{0,1\},n\in\mathbb{N}} \stackrel{c}{\equiv} \left\{\text{VIEW}^{\hat{\pi}}_{\hat{R}}(1^n, b_0, b_1, \sigma)\right\}_{b_0,b_1,\sigma\in\{0,1\},n\in\mathbb{N}} \quad (1)$$

To prove Eq. (1), we consider a hybrid simulator $\mathcal{S}^h$ that receives as input $b_{1-\sigma}$ in addition to the input (σ, b_σ) of $\mathcal{S}_{\hat{R}}$. It then works exactly as $\mathcal{S}_{\hat{R}}$ except that in Stage 3 of the simulation it sets $z_{1-\sigma} = \langle \alpha_{1-\sigma}, r_{1-\sigma} \rangle \oplus b_{1-\sigma}$ (instead of setting $z_{1-\sigma}$ to a random bit as $\mathcal{S}_{\hat{R}}$ does).

We first prove that the output of the hybrid simulator is indistinguishable from the receiver view in a real execution. That is, we prove that:

$$\left\{\mathcal{S}^h(1^n, \sigma, b_0, b_1)\right\} \stackrel{c}{\equiv} \left\{\text{VIEW}^{\hat{\pi}}_{\hat{R}}(1^n, b_0, b_1, \sigma)\right\} \quad (2)$$

[3] We say that a view is consistent with inputs and outputs if when running the party on the given view and input, it outputs the correct output.

The only difference between the two distributions is that in $\text{VIEW}^{\hat{\pi}}_{\hat{R}}(1^n, b_0, b_1, \sigma)$, the "extended view of R" (including both inputs used by the sender in each ideal OT call) is generated in a real execution of π, whereas in $\mathcal{S}^h(1^n, \sigma, b_0, b_1)$ the extended view is generated by $\mathcal{SIM}$ after the corruption at the end. So intuitively the guarantee that $\mathcal{SIM}$ is a good simulator implies that the two ensembles are computationally indistinguishable. Formally, we define a machine $\mathcal{D}$ that receives the output of $\mathcal{Z}$ after an execution of π in the adaptive setting, and attempts to determine whether it obtained a pair of receiver/sender views from a real or ideal execution. $\mathcal{D}$ generates an extended receiver-view from the pair of receiver/sender views that it received, and in addition computes the messages $(r_0, z_0), (r_1, z_1)$ using the correct sender inputs b_0, b_1 (that it's given as auxiliary input) and using the strings α_0, α_1 that appear in $\mathcal{Z}$'s output. Finally, $\mathcal{D}$ outputs the extended receiver-view together with the last message; this constitutes a view of the receiver $\hat{R}$ in $\hat{\pi}$. It is immediate that if $\mathcal{D}$ received a pair of views from a real execution of π then it outputs a view which is *identical* to $\text{VIEW}^{\hat{\pi}}_{\hat{R}}(1^n, b_0, b_1, \sigma)$. In contrast, if $\mathcal{D}$ received a pair of views generated by $\mathcal{SIM}$ in an ideal execution, then it outputs a view which is *identical* to $\mathcal{S}^h(1^n, \sigma, b_0, b_1)$. Thus, Eq. (2) follows from the security of π with simulator $\mathcal{SIM}$.

We now proceed to prove that the output of $\mathcal{S}_{\hat{R}}$ is statistically close to the output of the hybrid simulator S^h. That is:

$$\left\{\mathcal{S}_{\hat{R}}(1^n, \sigma, b_\sigma)\right\}_{b_0,b_1,\sigma\in\{0,1\},n\in\mathbb{N}} \stackrel{\text{s}}{\equiv} \left\{\mathcal{S}^h(1^n, \sigma, b_0, b_1)\right\}_{b_0,b_1,\sigma\in\{0,1\},n\in\mathbb{N}} \quad (3)$$

First note that $\mathcal{S}_{\hat{R}}$ and $\mathcal{S}^h$ work identically in the first two stages of the simulation and differ only in how $z_{1-\sigma}$ is computed. In particular, the distributions over the extended views generated by $\mathcal{S}_{\hat{R}}$ and by $\mathcal{S}^h$ are identical; let $V'_{\hat{R}}(1^n, \sigma)$ denote this distribution.

The first step is to show that with probability negligibly close to 1, there are exponentially many strings $\alpha_{1-\sigma}$ that are consistent with an extended view generated by $\mathcal{SIM}$ (as run by $\mathcal{S}^h$ or equivalently $\mathcal{S}_{\hat{R}}$). Fix $\sigma \in \{0,1\}$ and b_σ (the following holds for all σ, b_σ and we fix them here for clarity). For a given random tape $r_{\mathcal{SIM}}$ of $\mathcal{SIM}$ and a given α_σ, let v_R be the (regular, non-extended) view generated by $\mathcal{SIM}$ with random tape $r_{\mathcal{SIM}}$ and α_σ in the execution phase. Let $\Delta(r_{\mathcal{SIM}}, \alpha_\sigma)$ be the set of all strings $\alpha_{1-\sigma}$ of size $4n$ for which the views v_R, v_S generated by $\mathcal{SIM}$ with random tape $r_{\mathcal{SIM}}$ and inputs α_σ and $\alpha_{1-\sigma}$ in the computation and post-execution phases, respectively, are all *consistent* (we have already fixed σ and b_σ so consistency is also with respect to these values; see Footnote 3). Note that if $\mathcal{S}^h$ or $\mathcal{S}_{\hat{R}}$ would output $\perp$ in the first stage (i.e., if v_R is not consistent with the input and output) when choosing $r_{\mathcal{SIM}}, \alpha_\sigma$ then $\Delta(r_{\mathcal{SIM}}, \alpha_\sigma)$ is *empty*.

We now prove that for every $\sigma, b_\sigma \in \{0,1\}$, there exists a negligible function μ such that

$$\Pr_{r_{\mathcal{SIM}},\alpha_\sigma}\left[\,|\Delta(r_{\mathcal{SIM}}, \alpha_\sigma)| \geq 2^{3n}\right] \geq 1 - \mu(n).$$

Intuitively, this holds because if $\Delta(r_{\mathcal{SIM}}, \alpha_\sigma)$ is "small", then $\mathcal{SIM}$ would fail with high probability. Formally, assume that $\Pr_{r_{\mathcal{SIM}},\alpha_\sigma}[|\Delta(r_{\mathcal{SIM}}, \alpha_\sigma)| \geq 2^{3n}]$ is non-negligibly smaller than 1. We consider two cases:

1. With non-negligible probability, the view v_R generated by $\mathcal{SIM}$ with random tape $r_{\mathcal{SIM}}$ and α_σ cause $\mathcal{S}^h$ and $\mathcal{S}_{\hat{R}}$ to output $\perp$ (i.e., it is not consistent with the inputs/outputs): In this case, a distinguisher $\mathcal{Z}$ easily distinguishes the output of $\mathcal{SIM}$ from the views of v_R, v_S in a real execution of π since in a real execution the views are consistent except with negligible probability.
2. With non-negligible probability, v_R *is* consistent but $|\Delta(r_{\mathcal{SIM}}, \alpha_\sigma)| < 2^{3n}$: In this case, it is possible to distinguish a real execution of π from an ideal execution with $\mathcal{SIM}$ because the probability that a random $\alpha_{1-\sigma}$ is in $\Delta(r_{\mathcal{SIM}}, \alpha_\sigma)$ is less than $\frac{2^{3n}}{2^{4n}} = 2^{-n}$. Thus, the environment $\mathcal{Z}$ can just supply a random $\alpha_{1-\sigma}$ and see if in the post-execution corruption it receives a consistent view. In the real execution it will always receive a consistent view. However, in the ideal (simulated) execution, it will receive a consistent view with probability less than 2^{-n}. This is due to the fact that when $\alpha_{1-\sigma} \notin \Delta(r_{\mathcal{SIM}}, \alpha_\sigma)$ the view is *not* consistent. Thus, $\mathcal{Z}$ distinguishes with probability $(1 - 2^{-n})$ times the probability that this case occurs, which is non-negligible.

We stress that the calculation in the second case holds since the view of the receiver v_R is fixed before the post-execution phase and thus is fixed before $\alpha_{1-\sigma}$ is essentially chosen.

We now fix $r^*_{\mathcal{SIM}}$ and α^*_σ for which $|\Delta(r^*_{\mathcal{SIM}}, \alpha^*_\sigma)| \geq 2^{3n}$ and prove that the outputs of $\mathcal{S}^h$ and $\mathcal{S}_{\hat{R}}$ are statistically close for such $r^*_{\mathcal{SIM}}$ and α^*_σ. First, recall that an extended view v'_R is obtained by concatenating the other (previously not received) input of the sender in the n calls to the ideal OT to the view v_R. Since there are 2^n possible "other sender inputs" in the n ideal OT calls, it follows that for any given receiver-view v_R (which is fully determined by $r^*_{\mathcal{SIM}}$ and α^*_σ; recall that σ, b_σ are already fixed) there are at most 2^n possible associated extended views. (Again, this relies on the fact that the receiver-view is fixed before the post-execution corruption phase.)

Now, since there are 2^n possible extended views, we can partition the at least 2^{3n} consistent strings $\alpha_{1-\sigma} \in \Delta(r^*_{\mathcal{SIM}}, \alpha^*_\sigma)$ so that each partition contains the set of strings $\alpha_{1-\sigma}$ that yield the extended view v'_R. Equivalently, we associate $\alpha_{1-\sigma}$ with v'_R if $\mathcal{SIM}$ with $r^*_{\mathcal{SIM}}$ and α^*_σ outputs the extended view v'_R when given $\alpha_{1-\sigma}$ in the post-execution corruption phase. We denote by $\Gamma(v'_R, r^*_{\mathcal{SIM}}, \alpha^*_\sigma)$ the set of all strings $\alpha_{1-\sigma} \in \Delta(r^*_{\mathcal{SIM}}, \alpha^*_\sigma)$ which are associated with v'_R, as described above.

We argue that the probability of obtaining an extended view v'_R for which $|\Gamma(v'_R, r^*_{\mathcal{SIM}}, \alpha^*_\sigma)| < 2^n$ is at most 2^{-n} (i.e., an extended view for which the set of associated strings $\alpha_{1-\sigma}$ is small is obtained with probability at most 2^{-n}). We stress that the probability is over the choice of $\alpha_{1-\sigma}$ (all other randomness is fixed).

In order to see this, observe that the fact that $|\Delta(r^*_{\mathcal{SIM}}, \alpha^*_\sigma)| \geq 2^{3n}$ implies that there are at least 2^{3n} strings $\alpha_{1-\sigma}$ that are associated with *some* extended view v'_R. Now, for every v'_R for which $|\Gamma(v'_R, r^*_{\mathcal{SIM}}, \alpha^*_\sigma)| < 2^n$, we have that v'_R is generated by less than 2^n of those 2^{3n} strings. Thus, such a v'_R is obtained with probability less than $2^n/2^{3n} = 2^{-2n}$. By union bound over the 2^n possible

extended views v'_R (which also bounds the number of extended views for which $|\Gamma(v'_R, r^*_{\mathcal{SIM}}, \alpha^*_\sigma)| < 2^n$) we conclude that

$$\Pr\left[\,|\Gamma(v'_R, r^*_{\mathcal{SIM}}, \alpha^*_\sigma)| < 2^n\right] < 2^n \cdot \frac{1}{2^{2n}} = \frac{1}{2^n} \tag{4}$$

where the probability is over the choice of $\alpha_{1-\sigma}$.

From Eq. (4), we know that when $\alpha_{1-\sigma}$ is random, the probability that we will obtain an extended view v'_R such that $\Gamma(v'_R, r^*_{\mathcal{SIM}}, \alpha^*_\sigma)$ is small (with less than 2^n strings $\alpha_{1-\sigma}$ associated with it) is less than 2^{-n}. We therefore proceed by conditioning further over views v'_R for which $|\Gamma(v'_R, r^*_{\mathcal{SIM}}, \alpha^*_\sigma)| \geq 2^n$. Specifically, we argue that the distributions generated by $\mathcal{S}_{\hat{R}}$ and $\mathcal{S}^h$ are statistically close, conditioned on $r^*_{\mathcal{SIM}}, \alpha^*_\sigma$ such that $|\Delta(r^*_{\mathcal{SIM}}, \alpha^*_\sigma)| \geq 2^{3n}$ and conditioned on the extended view being a specific v'^*_R for which $\left|\Gamma(v'^*_R, r^*_{\mathcal{SIM}}, \alpha^*_\sigma)\right| \geq 2^n$.

First, observe that since $\alpha_{1-\sigma}$ is chosen uniformly and independently of $r^*_{\mathcal{SIM}}, \alpha_\sigma$, it is uniformly distributed in $\Gamma(v'^*_R, r^*_{\mathcal{SIM}}, \alpha^*_\sigma)$, when conditioning on all of the above. (The conditioning over v'^*_R is equivalent to saying that $\alpha_{1-\sigma}$ is uniform in $\Gamma(v'^*_R, r^*_{\mathcal{SIM}}, \alpha^*_\sigma)$ instead of being uniform in $\{0,1\}^{4n}$.) Second, recall that $\Gamma(v'^*_R, r^*_{\mathcal{SIM}}, \alpha^*_\sigma)$ is a set of size at least 2^n. Third, note that $H_{r_{1-\sigma}}(x) = \langle r_{1-\sigma}, x\rangle)$ is a universal hash function from $\{0,1\}^{4n}$ to $\{0,1\}$. Thus, by the Leftover Hash Lemma (the version given in [12]), it holds that:

$$SD\left((r_{1-\sigma}, \langle r_{1-\sigma}, \alpha_{1-\sigma}\rangle), (r_{1-\sigma}, U_1)\right) \leq \frac{1}{2^{(n-1)/2}}$$

where SD denotes statistical distance and U_1 denotes the uniform distribution over $\{0,1\}$ (as above, this statistical distance is computed when conditioned over $v'^*_R, r^*_{\mathcal{SIM}}, \alpha^*_\sigma$). Thus, these random variables are statistically close, conditioned on $v'^*_R, r^*_{\mathcal{SIM}}, \alpha^*_\sigma$ as above. Noting that in the output of $\mathcal{S}_{\hat{R}}$ we have $(r_{1-\sigma}, z_{1-\sigma}) = (r_{1-\sigma}, U_1)$, and in the output of $\mathcal{S}^h$ we have that $(r_{1-\sigma}, z_{1-\sigma}) = (r_{1-\sigma}, \langle r_{1-\sigma}, \alpha_{1-\sigma}\rangle)$, we conclude that

$$\left\{\mathcal{S}_{\hat{R}}(1^n, \sigma, b_\sigma) \mid v'^*_R, r^*_{\mathcal{SIM}}, \alpha^*_\sigma\right\} \stackrel{\text{s}}{\equiv} \left\{\mathcal{S}^h(1^n, \sigma, b_0, b_1) \mid v'^*_R, r^*_{\mathcal{SIM}}, \alpha^*_\sigma\right\}$$

where the conditioning is as described above. We reiterate that this holds since the extended views and the pair (r_σ, z_σ) are generated in an identical way by $\mathcal{S}_{\hat{R}}$ and $\mathcal{S}^h$, and the only difference is with respect to $(r_{1-\sigma}, z_{1-\sigma})$. Eq. (3) follows from the fact that we condition here on events that occur with all but negligible probability (and the events have identical probability with $\mathcal{S}_{\hat{R}}$ and $\mathcal{S}^h$). Combining Eq. (2) with Eq. (3), we derive Eq. (1), thereby completing the proof of Theorem 4.1.

Corollary – lengthening string OT. Observe that in our proof above the receiver always uses σ^{4n} for input. Thus, it follows that the theorem holds even if the receiver is interested in only obtaining the string of all of the "0 inputs" or the string of all of the "1 inputs". Stated differently, our proof holds also for the problem of lengthening string OT; i.e., for the problem of obtaining a *single string OT* for strings of length $n+1$ or more, given a *single string OT* for strings of length n.

5 OT Extensions Require Super-Logarithmic Calls

Theorem 5.1 *Let $f : \mathbb{N} \to \mathbb{N}$ be a function such that $f(n) \in \mathcal{O}(\log n)$, and let n be the security parameter. Then, if there exists a protocol π that is an OT-extension from $f(n)$ to $f(n)+1$ that is secure in the presence of malicious adversaries, then there exists a protocol for the OT functionality that is secure in the presence of malicious adversaries.*

Proof. Intuitively, in an OT extension protocol using only $\mathcal{O}(\log n)$ ideal OT calls, it is possible for the receiver to guess the bits that it would receive as output from these calls instead of actually running them. Since there are only $\mathcal{O}(\log n)$ calls, the probability that the receiver guesses correctly is $2^{-\mathcal{O}(\log n)} = 1/\mathsf{poly}(n)$. This idea can be used to construct an OT protocol that is weak in the sense that full privacy is maintained, but correctness only holds with probability $1/2 + 1/\mathsf{poly}(n)$. We stress that a naive attempt to implement the above idea will not work since it is necessary to ensure that if the receiver's guesses are incorrect then it still outputs the correct output of the protocol with probability almost 1/2. Otherwise, the "advantage" in obtaining the correct output when the receiver guesses correctly can be canceled out by the "disadvantage" when the receiver guesses incorrectly. We therefore use a similar technique as in the proof regarding adaptive adversaries above. Specifically, we use the fact that an extension from $f(n)$ to $f(n)+1$ implies an extension from $f(n)$ to n, and then use this to obliviously transfer n random bits. The actual oblivious transfer is carried out by applying a universal hash function to the random strings and using the result to mask the actual bits being transferred. This ensures that we obtain correctness that is noticeable greater than 1/2 and so can be amplified. However, in addition, we also have to claim that privacy is maintained. This is not immediate since the receiver does not follow the specified protocol (rather, it chooses the outputs from the ideal OT calls at random, and this may effect the other messages that it sends). By requiring that the extension protocol be secure for malicious adversaries, this ensures that the receiver cannot learn more by behaving in this way. In addition, we show that a malicious sender can also achieve the same affect by inputting a random bit (for both sender inputs) in each ideal OT call. This implies that a malicious sender can also not learn anything by the receiver behaving in this way. We now proceed to the formal proof.

Throughout the proof, we will construct protocols that are secure for *semi-honest adversaries* only. This suffices since semi-honest OT implies malicious OT [7, 10]. Let $f : \mathbb{N} \to \mathbb{N}$ be a function such that $f(n) \in \mathcal{O}(\log n)$ and let $\pi = \langle S, R \rangle$ be a protocol such that on security parameter n and inputs $x_0, x_1 \in \{0,1\}^{f(n)+1}$ and $\sigma \in \{0,1\}^{f(n)+1}$ securely computes the $(f(n)+1) \times OT$ functionality in the $OT^{f(n)}$-hybrid model (that is, making at most $f(n)$ calls to an ideal OT). We assume that π is secure in the presence of malicious adversaries. We assume that in all of these calls, R is the one to receive output (this is without loss of generality since oblivious transfer is symmetric [19] and so the roles can be reversed by adding additional messages in π). We show how to construct a protocol for computing the OT functionality without any further

assumptions other than the existence of an extension protocol π with the parameters in the theorem statement. This is achieved in two steps. First, we use the OT-extension from $f(n) = \mathcal{O}(\log n)$ to n to construct a protocol $\tilde{\pi}$ which is simulatable and therefore fully secure, but whose error might be large. Then we amplify the correctness of the protocol using multiple executions. As we show, this can be done once the basic protocol is fully secure.

Step 1 – constructing a weak-OT. We begin by formally defining weak-OT, which is an oblivious transfer for semi-honest adversaries that has weak correctness but full simulation security.[4] We then show how to construct a weak-OT protocol $\tilde{\pi} = \langle \tilde{S}, \tilde{R} \rangle$ from an OT-extension from $f(n)$ to n. Note that by Proposition 2.6, if there exists an extension protocol from $f(n)$ to $f(n)+1$, then there exists an extension protocol from $f(n)$ to n.

Definition 5.2 (Weak-OT) *A two-party protocol $\pi = \langle S, R \rangle$ is a* weak-OT *if the following hold:*

- **Weak-correctness:** *There exists a polynomial $p(\cdot)$ such that for all $b_0, b_1, \sigma \in \{0,1\}$ and all large enough n's,* $\Pr[\text{OUTPUT}_R^{\pi}(1^n, b_0, b_1, \sigma) = b_\sigma] \geq \frac{1}{2} + \frac{1}{p(n)}$.
- **Privacy:** *There exists PPT machines $\mathcal{S}_R$ and $\mathcal{S}_S$ such that*

$$\{\mathcal{S}_R(1^n, \sigma, b_\sigma)\}_{b_0,b_1,\sigma\in\{0,1\},n\in\mathbb{N}} \stackrel{c}{\equiv} \{\text{VIEW}_R^{\pi}(1^n, b_0, b_1, \sigma)\}_{b_0,b_1,\sigma\in\{0,1\},n\in\mathbb{N}}$$

$$\{\mathcal{S}_S(1^n, b_0, b_1)\}_{b_0,b_1,\sigma\in\{0,1\},n\in\mathbb{N}} \stackrel{c}{\equiv} \{\text{VIEW}_S^{\pi}(1^n, b_0, b_1, \sigma)\}_{b_0,b_1,\sigma\in\{0,1\},n\in\mathbb{N}}$$

Let $\alpha_0, \alpha_1, c \in \{0,1\}^n$ be n-bit strings. Let $\alpha_0 = \alpha_0^1, \ldots, \alpha_0^n$, $\alpha_1 = \alpha_1^1, \ldots, \alpha_1^n$, and $c = c_1, \ldots, c_n$. Recall that $\alpha_c = \alpha_{c_1}^1, \alpha_{c_2}^2, \ldots, \alpha_{c_n}^n$; that is, the ith bit of α_c is either α_0^i or α_1^i, depending on the value of c_i.

Let $\pi = \langle S, R \rangle$ be an OT-extension protocol from $f(n) = \mathcal{O}(\log n)$ to n. We construct a weak OT protocol $\tilde{\pi} = \langle \tilde{S}, \tilde{R} \rangle$ as follows:

Protocol 5.3 (A weak-OT with no ideal OT calls)

- **Inputs:** *Sender $\tilde{S}$ has two bits $b_0, b_1 \in \{0,1\}$, and receiver $\tilde{R}$ has $\sigma \in \{0,1\}$.*
- **The protocol:**
  1. *$\tilde{S}$ chooses two random strings $\alpha_0, \alpha_1 \in_R \{0,1\}^n$.*
  2. *$\tilde{R}$ chooses a random string $c \in_R \{0,1\}^n$.*
  3. *$\tilde{S}$ and $\tilde{R}$ simulate an execution of the extension protocol π, as follows:*
     (a) *$\tilde{S}$ plays the role of the sender S with input $\alpha_0, \alpha_1 \in \{0,1\}^n$ and $\tilde{R}$ plays the role of the receiver R with input $c \in \{0,1\}^n$.*
     (b) *Whenever π instructs the parties to make an OT call, the parties make no call and $\tilde{R}$ chooses a random bit as its output from the call. We denote by $\beta_1, \ldots, \beta_{f(n)}$ the random bits chosen by $\tilde{R}$ as the OT outputs.*
     (c) *Let $\gamma \in \{0,1\}^n$ denote the receiver-output of the simulation of π received by $\tilde{R}$.*

[4] Note that we cannot cast this as a special case of Definition 2.2 since full correctness is required there by stating that π computes f.

4. *$\tilde{R}$ chooses a random $c' \in_R \{0,1\}^n$ and sends (c_0, c_1) to $\tilde{S}$, where $c_\sigma = c$ and $c_{1-\sigma} = c'$.*
5. *$\tilde{S}$ chooses two random strings $r_0, r_1 \in_R \{0,1\}^n$, computes $z_0 = \langle r_0, \alpha_{c_0}\rangle \oplus b_0$ and $z_1 = \langle r_1, \alpha_{c_1}\rangle \oplus b_1$, and sends $(r_0, z_0), (r_1, z_1)$ to $\tilde{R}$.*

– **Output:** *$\tilde{S}$ outputs nothing and $\tilde{R}$ outputs* $\mathsf{out} = z_\sigma \oplus \langle r_\sigma, \gamma\rangle$.

We now prove that Protocol 5.3, also denoted $\tilde{\pi}$, is a weak-OT protocol. Intuitively, weak correctness holds because $\tilde{R}$ correctly guesses the outputs of the OT calls with probability $1/2^{f(n)}$ in which case $\gamma = \alpha_c$ by the correctness of π (except with negligible probability), and thus $\langle r_\sigma, \gamma\rangle = \langle r_\sigma, \alpha_c\rangle$ and $\mathsf{out} = b_\sigma$. In addition, when the guesses made by $\tilde{R}$ are not correct, it still outputs b_σ with probability $1/2$. This holds because when r is random, the function $\langle r, \cdot\rangle$ is a universal hash function, and so $\langle r_\sigma, \gamma\rangle$ is uniformly distributed and equals $\langle r_\sigma, \alpha_c\rangle$ with probability $1/2$. See [16] for the full proof.

We proceed to prove *privacy*, by constructing $\mathcal{S}_{\tilde{S}}$ and $\mathcal{S}_{\tilde{R}}$ as required. We start by constructing the simulator $\mathcal{S}_{\tilde{S}}$ for the case that the sender is corrupted. To prove this we use the fact that the original protocol π is secure in the presence of malicious adversaries. Consider a malicious adversary $\mathcal{A}$ for π that controls the sender and learns its input $\alpha_0, \alpha_1 \in \{0,1\}^n$. $\mathcal{A}$ follows the honest strategy for S except that it chooses random bits $\beta_1, \ldots, \beta_n$ and then in the jth call to the ideal OT functionality, it uses β_j as both sender inputs to the OT call (ensuring that R receives β_j). We stress that in the rest of the execution, it behaves as if it has used the correct inputs that were supposed to be sent to the OT calls. Observe that the view of $\mathcal{A}$ in an execution of π is *identically distributed* to the view of $\tilde{S}$ in the simulation of π run in Step 3 of Protocol 5.3. Let $\mathcal{SIM}$ be the simulator that is guaranteed to exist for $\mathcal{A}$ by the security of π. We construct the simulator $\mathcal{S}_{\tilde{S}}$ using $\mathcal{SIM}$:

Construction 5.4 ($\mathcal{S}_{\tilde{S}}$) *: Upon input $b_0, b_1 \in \{0,1\}$, $\mathcal{S}_{\tilde{S}}$ works as follows:*

1. *$\mathcal{S}_{\tilde{S}}$ chooses two random strings $\alpha_0, \alpha_1 \in_R \{0,1\}^n$ and runs $\mathcal{SIM}$ with sender-inputs α_0, α_1. Let v_S be the sender-view output by $\mathcal{SIM}$ at the end of its execution ($\mathcal{SIM}$ also sends input to the trusted party, but this is ignored by $\mathcal{S}_{\tilde{S}}$).*
2. *$\mathcal{S}_{\tilde{S}}$ chooses two random strings $c_0, c_1 \in_R \{0,1\}^n$ as the message received from $\tilde{R}$ in Step 4 of Protocol 5.3, and outputs $v_{\tilde{S}} = (v_S, c_0, c_1)$.*

The fact that $\mathcal{S}_{\tilde{S}}$ is a good simulator follows immediately from the fact that $\mathcal{SIM}$ generates a sender-view that is indistinguishable from what $\mathcal{A}$ would see in a real execution of π. Since we have already observed that the view of $\tilde{S}$ in Step 3 of Protocol 5.3 is identical to the view of $\mathcal{A}$ above in π, it follows that v_S is indistinguishable from $\tilde{S}$'s view in Step 3 of Protocol 5.3. Next observe that a distinguisher $\mathcal{D}$ for $\mathcal{SIM}$ and π obtains the input/output used (α_0, α_1, c) and thus can extend the view of the sender to include c_0, c_1 where $c_\sigma = c$, and c is the input of R into the execution of π with $\mathcal{A}$ (we can assume that D knows σ as auxiliary input). Thus, the view of $\tilde{S}$ in Protocol 5.3 (resp., as generated by simulator $\mathcal{S}_{\tilde{S}}$) can be perfectly constructed by $\mathcal{D}$ from the real view v_S of S in π

(resp., from a simulated view v_S of S as generated by $\mathcal{SIM}$). This implies that if the output of $\mathcal{S}_{\tilde{S}}$ can be distinguished from the view of $\tilde{S}$ in a real execution of Protocol 5.3, then the output of $\mathcal{SIM}$ can be distinguished from the view of $\mathcal{A}$ in a real execution of π, in contradiction to the security of π with simulator $\mathcal{SIM}$. The formal reduction is straightforward.

We now proceed to construct a simulator $\mathcal{S}_{\tilde{R}}$ for the case that the receiver is corrupted. As above, we consider a malicious adversary $\mathcal{A}$ for π as follows. $\mathcal{A}$ receives the receiver's input $c \in \{0,1\}^n$ and follows the honest receiver strategy except that in each of the calls to the ideal OT functionality, it chooses a random bit β_j and proceeds with β_j as the output of the ideal OT. Let $\mathcal{SIM}$ be the simulator that is guaranteed to exist for $\mathcal{A}$ by the security of π. We use it construct the simulator $\mathcal{S}_{\tilde{R}}$ (recall that $\mathcal{SIM}$ works in the setting for malicious adversaries and thus interacts with a trusted party and sends a receiver-input which is not necessarily the prescribed receiver-input):

Construction 5.5 ($\mathcal{S}_{\tilde{R}}$) *: Upon input $\sigma, b_\sigma \in \{0,1\}$, $\mathcal{S}_{\tilde{R}}$ works as follows:*

1. *$\mathcal{S}_{\tilde{R}}$ chooses three random strings $\alpha_0, \alpha_1, c \in_R \{0,1\}^n$.*
2. *$\mathcal{S}_{\tilde{R}}$ runs $\mathcal{SIM}$ with receiver input c.*
3. *When $\mathcal{SIM}$ sends some $c^* \in \{0,1\}^n$ to the trusted party, $\mathcal{S}_{\tilde{R}}$ hands α_{c^*} as the receiver-output to $\mathcal{SIM}$ from the trusted party. Let v_R be the output of $\mathcal{SIM}$.*
4. *$\mathcal{S}_{\tilde{R}}$ chooses random strings $c', r_0, r_1 \in_R \{0,1\}^n$, and sets $c_\sigma = c$ and $c_{1-\sigma} = c'$. Then, $\mathcal{S}_{\tilde{R}}$ computes $z_\sigma = \langle r_\sigma, \alpha_{c_\sigma} \rangle \oplus b_\sigma$ and sets $z_{1-\sigma} \in_R \{0,1\}$ to be a random bit.*
5. *$\mathcal{S}_{\tilde{R}}$ outputs a receiver view $(c_0, c_1, v_R, r_0, z_0, r_1, z_1)$. (Note that c_0, c_1 are actually part of $\tilde{R}$'s random tape, since they are chosen by $\tilde{R}$.)*

Intuitively, the two differences between the simulated and real executions are (a) the execution of π is simulated using $\mathcal{SIM}$ (which is indistinguishable by assumption), and (b) $z_{1-\sigma}$ is generated randomly instead of being computed as $z_{1-\sigma} = \langle r_{1-\sigma}, \alpha_{c_{1-\sigma}} \rangle \oplus b_{1-\sigma}$. However, since $c_{1-\sigma} = c'$ is chosen at random independently of the execution, and since $\mathcal{SIM}$ learns only the bits in the sender's input that correspond to c^*, with high probability there is enough uncertainty about $\langle \alpha_{c_{1-\sigma}}, r_{1-\sigma} \rangle$ and thus $z_{1-\sigma}$ is statistically close to a random bit. This is formally proven in [16]. We conclude that Protocol 5.3 is a weak-OT protocol.

Step 2 – full-OT from weak-OT. The last step to transform weak OT to full OT simply works by running multiple executions and taking the majority result. Since the weak OT is fully secure, and it is only the correctness that is weak, this preserves security and so achieves what is needed. This concludes the proof.

Acknowledgements. We thank Yuval Ishai for helpful discussions.

References

1. W. Aiello, Y. Ishai and O. Reingold. Priced Oblivious Transfer: How to Sell Digital Goods. In *EUROCRYPT 2001*, Springer-Verlag (LNCS 2045), pages 110–135, 2001.
2. D. Beaver. Precomputing Oblivious Transfer. In *CRYPTO'95*, Springer-Verlag (LNCS 963), pages 97–109, 1995.
3. D. Beaver. Correlated Pseudorandomness and the Complexity of Private Computations. In the 28*th STOC*, pages 479–488, 1996.
4. R. Canetti. Security and Composition of Multiparty Cryptographic Protocols. *Journal of Cryptology*, 13(1):143–202, 2000.
5. S. Even, O. Goldreich and A. Lempel. A Randomized Protocol for Signing Contracts. In *Communications of the ACM,* 28(6):637–647, 1985.
6. Y. Gertner, S. Kannan, T. Malkin, O. Reingold, and M. Viswanathan. The Relationship Between Public Key Encryption and Oblivious Transfer. In the 41*st FOCS*, page 325–335, 2000.
7. O. Goldreich, S. Micali and A. Wigderson. How to Play any Mental Game – A Completeness Theorem for Protocols with Honest Majority. In 19*th STOC,* pages 218–229, 1987. For details see [9].
8. O. Goldreich. A Note on Computational Indistinguishability. *Information Processing Letters*, 34(6):277–281, 1990.
9. O. Goldreich. *Foundations of Cryptography: Volume 2 – Basic Applications*. Cambridge University Press, 2004.
10. I. Haitner, Y. Ishai, E. Kushilevitz, Y. Lindell and E. Petrank. Black-Box Constructions of Protocols for Secure Computation. *SIAM Journal on Computing*, 40(2):225–266, 2011.
11. Y. Ishai, J. Kilian, K. Nissim and E. Petrank. Extending Oblivious Transfer Efficiently. In *CRYPTO 2003*, Springer (LNCS 2729), pages 145–161, 2003.
12. R. Impagliazzo and D. Zuckerman. How to Recycle Random Bits. In the 30*th FOCS*, 248–253, 1989.
13. J. Kilian. Founding Cryptography on Oblivious Transfer. In the 20*th STOC*, pages 20–31, 1988.
14. Y Lindell. Adaptively Secure Two-Party Computation with Erasures. In *CT-RSA 2009)*, Springer (LNCS 5473), pages 117–132, 2009.
15. Y. Lindell and H. Zarosim. Adaptive Zero-Knowledge Proofs and Adaptively Secure Oblivious Transfer. In the *Journal of Cryptology,* 24(4):761-799, 2011.
16. Y. Lindell and H. Zarosim. On the Feasibility of Extending Oblivious Transfer. *Cryptology ePrint Archive: Report 2012/333*, 2012.
17. M. Rabin. How to Exchange Secrets by Oblivious Transfer. *Tech. Memo TR-81,* Aiken Computation Laboratory, Harvard University, 1981.
18. S. Winkler and J. Wullschleger. On The Efficiency Of Classical And Quantum Oblivious Transfer Reductions. In *CRYPTO 2010*, Springer (LNCS 6223), pages 707–723, 2010.
19. S. Wolf and J. Wullschleger. Oblivious Transfer is Symmetric. In *EUROCRYPT 2006*, Springer (LNCS 4004), pages 222–232, 2006.
20. A. Yao. How to Generate and Exchange Secrets. In the 27*th FOCS*, pages 162–167, 1986.