Two-Round Oblivious Transfer from CDH or LPN

Nico Döttling Sanjam Garg* Mohammad Hajiabadi Daniel Masny† Daniel Wichs‡

April 22, 2019

Abstract

We show a new general approach for constructing maliciously secure two-round oblivious transfer (OT). Specifically, we provide a generic sequence of transformations to upgrade a very basic notion of two-round OT, which we call *elementary OT*, to UC-secure OT. We then give simple constructions of elementary OT under the Computational Diffie-Hellman (CDH) assumption or the Learning Parity with Noise (LPN) assumption, yielding the first constructions of malicious (UC-secure) two-round OT under these assumptions. Since two-round OT is complete for two-round 2-party and multi-party computation in the malicious setting, we also achieve the first constructions of the latter under these assumptions.

1 Introduction

Oblivious transfer (OT) [Rab05, EGL85], is a fundamental primitive in cryptography. An OT protocol consists of two parties: a *sender* and a *receiver*. The sender's input is composed of two strings (m_0, m_1) and the receiver's input is a bit c. At the end of the execution of the OT protocol, the receiver should only learn the value m_c, but should not learn anything about the other value m_{1-c}. The sender should gain no information about the choice bit c. This very simple primitive is often used as the foundational building block for realizing secure computation protocols [Yao82, GMW87]. Thus, the efficiency characteristics of the OT protocol directly affect the efficiency of the resulting secure computation protocol. As such, several notions of OT, achieving varying security and efficiency properties, have been devised (see e.g., [Lin16]). Ideally, we want to achieve a *simulation-based* definition of OT, where we require that malicious behavior in the real world can be simulated in an ideal world with an ideal OT functionality, and even more desirably, we want to do so in the *universal composability* (UC) framework [Can01].

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

†Research supported by the Center for Long-Term Cybersecurity (CLTC, UC Berkeley).

‡Research supported by NSF grants CNS-1314722, CNS-1413964, CNS-1750795 and the Alfred P. Sloan Research Fellowship.

OT in Two-Rounds. As the name suggests, a two-round OT protocols allows the OT functionality to be implemented in just the minimal two-rounds of communication. Namely, the receiver sends the first-round message based on her input bit c. Next, using his input (m_0, m_1) and the first message of the protocol, the sender generates and sends the second-round message of the protocol. Finally, the receiver uses the second-round protocol message to recover m_c.

OT protocols that require *only* two rounds of communication are often desirable. Most importantly, two-round OT protocols are complete (necessary and sufficient) for general two-round (i.e., round optima) two-party [Yao82] and multi-party secure computation (2PC,MPC) [GS18, BL18] in both the semi-honest and malicious settings. Unfortunately, constructing two-round OT is typically much harder than constructing OT protocols with a larger round complexity. In particular, by relying on ZK proofs, we can construct constant-round malicious OT assuming only constant-round semi-honest OT and the latter follows from essentially all known assumptions that imply public-cryptography. On the other hand, no such equivalence is known for 2-round protocols since zero-knowledge proofs add more round. Furthermore, we know that two-round simulation-secure malicious OT is impossible in the plain model, and therefore we consider security in the common reference string (CRS) model.

Assumptions. Over the years, tremendous progress has been made in constructing both *semi-honest* and *maliciously* secure two-round OT protocols [CCM98, NP01, AIR01, DHRS04, PVW08, HK12, BD18] from a wide variety of assumptions. However, there are still gaps in our understanding — namely, constructing two-round OT typically requires stronger assumptions than what known to be sufficient for just OT. This is especially true for the case of maliciously secure OT. In this work, we attempt to bridge this gap. More specifically, we ask:

Can maliciously secure two-round OT and be based on the Computational Diffie-Hellman (CDH) assumption or the Learning Parity with Noise (LPN) assumption?

Since two-round malicious (UC) OT is complete for two-round malicious (UC) 2PC and MPC, the above is equivalent to asking whether the latter can be instantiated under the CDH and LPN assumptions. While constructions of UC-secure two-round OT under the Decisional Diffie-Hellman (DDH) assumption and the Learning with Errors (LWE) assumption are known [PVW08], the question of constructing the same under CDH and LPN has so far remained open. Moreover, we do not even have two-round constructions under CDH or LPN that satisfy any alternate weaker notions of malicious OT security that have been previously proposed in the literature.

1.1 Why is Two-Round Maliciously Secure OT Difficult?

One reason that (two-round) OT is difficult to construct is that this notion is even difficult to define. Simulation-based definitions of security are complex and impose requirements that often seem stronger than necessary and hard to achieve. Unlike (say) public-key encryption, where we have simple game-based definitions that imply simulation-based (semantic) security, we do not have any simpler definitions of malicious OT security that suffice for simulation. All prior attempts from the literature to weaken the definition of OT security are still complex and require some form of extraction/simulation. In particular, to meaningfully define that the malicious receiver only learns one of the two sender values m_0, m_1, all known definitions require that we can somehow *extract* the receiver's choice bit c from the first OT message and then argue that the second message hides the value m_{1-c}.

To meet any such extraction-based definition, we need to start with an OT where the receiver's choice bit is statistically committed in the first OT message. This seems like a significant restriction. For example there is a natural construction of OT from CDH due to Bellare and Micali [BM90], which achieves semi-honest security in the standard model or a weak form of malicious security in the random-oracle model. However, in this construction, the first message only commits the receiver computationally to the choice bit and hence there is no hope of extracting it. Therefore, it appears difficult to prove any meaningful notion of malicious security without resorting to the random oracle model.

Overall, we are aware of only two approaches towards achieving maliciously-secure OT. The first starts with semi-honest OT and then compiles it to malicious OT using zero-knowledge proofs. Unfortunately, if we want two-round OT we would need to use non-interactive zero-knowledge (NIZK) proofs and we do not have instantiations of such NIZKs under many natural assumptions such as CDH or LPN (or LWE). The other approach, used by Peikert, Vaikuntanathan and Waters [PVW08] (and to some extent also e.g., [NP01, AIR01, BD18]) takes advantage of a statistically "lossy" mode of DDH/LWE based encryption. Unfortunately, we do not have any such analogous "lossy" mode for CDH/LPN based encryption and therefore this approach too appears to be fundamentally stuck.

1.2 Our Results

In this work, we give a new general approach for constructing UC-secure two-round OT.[1] Specifically, we introduce an extremely weak and simple notion of two-round OT, which we call *elementary* OT. This notion is defined via a game-based definition and, in contrast to all prior notions of OT, does not rely on an extractor. We then provide a series of generic transformations that upgrade the security of elementary OT, eventually culminating in a UC-secure two-round OT. These transformations are the main technically challenging contributions of the paper. Lastly, we show simple constructions of two-round *elementary* OT under the Computational Diffie-Hellman (CDH) assumption or the Learning Parity with Noise (LPN) assumption, yielding the first constructions of UC-secure two-round OT under these assumptions. We rely on a variant of LPN with noise-rate $1/n^{\varepsilon}$ for some arbitrary constant $\varepsilon > \frac{1}{2}$.[2]

Applications to Two-round MPC. As mentioned earlier, two-round OT is known to be complete for constructing two-round MPC [GS18, BL18]. Thus, our results also yield the first constructions of two-round malicious (UC-secure) MPC under the Computational Diffie-Hellman (CDH) assumption or the Learning Parity with Noise (LPN) assumption.

Open problems. Interestingly, our generic transformations use garbled circuits that make a non-black-box use of the underlying cryptographic primitives. We leave it as an open problem to obtain a black-box construction or show the impossibility thereof.

[1]Although we achieve UC security, it does not appear that achieving stand-alone security would make the problem our our solutions significantly simpler.

[2]This is marginally stronger than the variant used in constructing public-key encryption due to Alekhnovich [Ale03], which relies on a noise-rate $1/\Theta(n^{1/2})$.

2 Technical Overview

Our results are obtained via a sequence of transformations between various notions of OT. We give an overview of this sequence in Figure 1 and explain each of the steps below. All of the notions of OT that we consider are two-round and can rely on a *common reference string* (CRS), which is generated by a trusted third party and given to both the sender and the receiver. For simplicity, we often ignore the CRS in the discussion below.

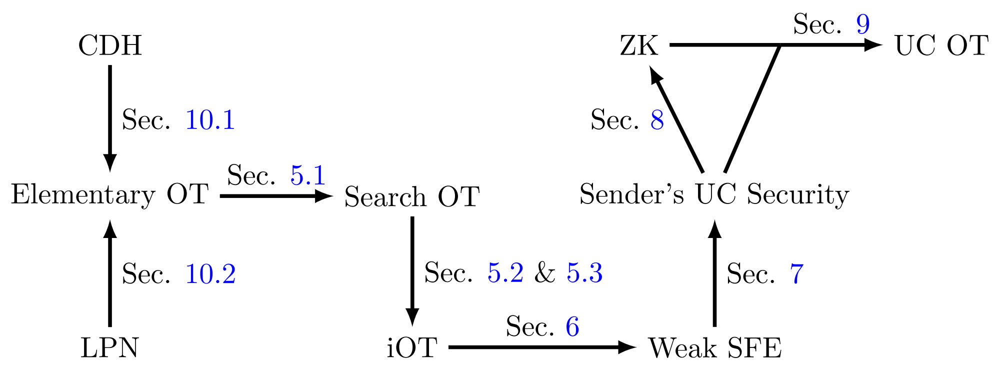

Figure 1: Sequence of transformations leading to our results.

Elementary OT. We begin by defining an extremely weak and simple notion of OT, called elementary OT. The receiver uses her choice bit c to generate a first round message otr. The sender then uses otr to generate a second-round message ots together with two values y_0, y_1. The receiver gets ots and uses it to recover the value y_c. Note that, unlike in standard OT, the sender does not choose the two values y_0, y_1 himself, but instead generates them together with ots. (One may think of this as analogous to the distinction between key-encapsulation and encryption.) The security of elementary OT is defined via the following two game-based requirements:

1. Receiver Security: The receiver's choice bit c is computationally hidden by the first-round OT message otr.

2. Sender Security: A malicious receiver who creates the first-round message otr maliciously and is then given an honestly generated second-round message ots cannot simultaneously output both of the values y_0, y_1 except with negligible probability.

Note that elementary OT provides a very weak notion of sender security. Firstly, it only provides unpredictability, rather than indistinguishability, based security – the malicious receiver cannot output both values y_0, y_1, but may learn some partial information about each of the two values. Second of all, it does not require that the there is a consistent bit w such that the value y_w is hidden from the malicious receiver – it may be that, even after the receiver maliciously chooses otr, for some choices of ots she learns y_0 and for other choices she learns y_1. We fix the second issue first.

From Elementary OT to Search OT. We define a strengthening of elementary OT, which we call *search OT*. The syntax and the receiver security remain the same. For sender security, we still keep an unpredictability (search) based security definition. But now we want to ensure that, for any choice of the malicious receiver's message otr, there is a consistent bit w such that y_w is hidden. We want to capture this property without requiring the existence of an (even inefficient) extractor that can find such w. We do so as follows. For any choice of the malicious receiver's first message otr (along with all her random coins and the CRS), we define two probabilities $\varepsilon_0, \varepsilon_1$ which denote the probability of the receiver outputting y_0 and y_1 respectively, taken only over the choice of ots. We require that for any polynomial p, with overwhelming probability over the receiver's choices, at least one of ε_0 or ε_1 is smaller than $1/p$. In particular, this means that with overwhelming probability over the malicious receiver's choice of otr, there is a fixed and consistent bit w such that the receiver will be unable to recover y_w from the sender's message ots. Note that the value w may not be extractable (even inefficiently) from otr alone since the way that w is defined is "adversary-dependent".

To go from elementary OT to search OT, we rely on techniques from "hardness amplification". The difficulty of using a search-OT adversary to break elementary-OT security is that a search-OT adversary can, for example, have $\varepsilon_0 = \varepsilon_1 = \frac{1}{2}$, but for half the value of ots it outputs the correct y_0 and for half it outputs the correct y_1, yet it never output both correct values simultaneously. However, if we could ensure that $\varepsilon_0, \varepsilon_1$ are both much larger than $\frac{1}{2}$, then this could not happen. We use hardness amplification to achieve this. In particular, we construct search OT scheme from elementary OT by having the sender generate λ (security parameter) different second-round messages of the elementary OT and set the search OT values to be the concatenations $\mathsf{OTS} = (\mathsf{ots}^1, \ldots, \mathsf{ots}^\lambda)$ and $Y_0 = (y_0^1, \ldots, y_0^\lambda), Y_1 = (y_1^1, \ldots, y_1^\lambda)$. By hardness amplification, if for some choice of otr the malicious receiver can separately predict each of Y_0, Y_1 with probability better than some inverse polynomial $1/p$, then that means it can separately predict each of the components y_0, y_1 with extremely high probability $> \frac{3}{4}$, and by the union bound, can therefore predict both components y_0, y_1 simultaneously with probability $> \frac{1}{4}$.

From Search OT to Indistinguishability OT. Next, we define a notion that we call *indistinguishability OT*. Here, just like in standard OT, the sender gets to choose his two values m_0, m_1 himself, rather than having the scheme generate values y_0, y_1 for him, as was the case in elementary and search OT. The receiver security remains the same as in elementary and search OT: the receiver's choice bit c is hidden by her first-round message otr. The sender security is defined in a similar manner to search OT, except that we now require indistinguishability rather than unpredictability. In particular, the malicious receiver chooses two values m_0, m_1 and a maliciously generated otr. For any such choice, we define two probabilities $\varepsilon_0, \varepsilon_1$, where ε_b denotes the receiver's advantage, calculated only over the random coins of the sender, in distinguishing between ots generated with the messages (m_0, m_1) versus (m'_0, m'_1) where m'_b is uniformly random and $m'_{1-b} = m_{1-b}$. We require that for any polynomial p, with overwhelming probability over the receiver's choices, at least one of ε_0 or ε_1 is smaller than $1/p$. In particular, this means that, with overwhelming probability, the malicious receiver's choice of otr fixes a consistent bit w such that the receiver does not learn anything about m_w.

To go from search OT to indistinguishability OT with 1-bit values m_0, m_1, we rely on the Goldreich-Levin hardcore bit [GL89]. In particular, we use search OT to generate ots along with values y_0, y_1 and then use the Goldreich-Levin hardcore bits of y_0, y_1 to mask m_0, m_1 respectively. To then allow for multi-bit values m_0, m_1, we simply have the sender send each bit separately, by reusing the same receiver message otr for all bits.

From Indistinguishability OT to Weak SFE. Next, we generalize from OT and define a weak form of (two-round) *secure function evaluation (weak-SFE)*. Here, there is a receiver with an input x and a sender with a circuit f. The receiver learns the output $f(x)$ in the second round. We define a very simple (but weak) game-based notion of malicious security, without relying on a simulator or extractor:

- Receiver Security: The receiver's first-round message hides the input x from the sender.
- Sender Security: A malicious receiver cannot distinguish between any two functionally equivalent circuits f_0, f_1 used by the sender.

We show how to compile indistinguishability OT to weak SFE. Indeed, the construction is the same as the standard construction of (standard) SFE from (standard) OT: the receiver sends first-round OT messages corresponding to the bits of the input x and the sender creates a garbled circuit for f and uses the two input labels as the values for the second-round OT messages.

The proof of sender security, however, is very different than that for the standard construction of SFE from OT, which relies on extracting the receiver's OT choice bits. Instead, we rely on technical ideas that are similar to and inspired by those recently used in the context of *distinguisher-dependent simulation* [JKKR17] and have a sequence of hybrids that depends on the adversary. More concretely, indistinguishability OT guarantees that for each input wire, there is some bit w such that the adversary cannot tell if we replace the label for w by uniform. However, this bit w is defined in an adversary-dependent manner. This effectively allows us to extract the adversary's OT choice bits. Therefore, we have a sequence of adversary-dependent hybrids where we switch the OT values used by the sender and replace the labels for the bits w by random values. We then rely on garbled circuit security to argue that garblings of f_0 and f_1 are indistinguishable, and conclude that the adversary's advantage is negligible.

Formalizing the above high-level approach is the most technically involved component of the paper.

From Weak SFE to OT with UC Sender Security. We show how to go from weak SFE to an OT scheme that has UC-security for the sender. In particular, this means we can extract the choice bit c from the receiver's first-round message otr and simulate the sender's second-round message ots given only m_c, without knowing the "other" value m_{1-c}. For the receiver's security, we maintain the same indistinguishability-based requirement as in elementary/search/indistinguishability OT, which guarantees that the choice bit c is hidden by the first-round OT message otr. We refer to this as a "half-UC OT" for short. This is the first step where we introduce a simulation/extraction based notion of security.

Our compiler places a public-key pk of a public-key encryption (PKE) scheme to the CRS. The receiver encrypts her choice bit c under pk using randomness r and sends the resulting ciphertext $\mathsf{ct} = \mathsf{E}_{\mathsf{pk}}(c; r)$ as part of her first-round OT message. At the same time, the receiver and sender run an instance of weak SFE, where the receiver's input is $x = (c, r)$ and the sender's circuit is $f_{\mathsf{pk},\mathsf{ct},m_0,m_1}(c, r)$, which output m_c if $\mathsf{ct} = \mathsf{E}_{\mathsf{pk}}(c; r)$ and $\perp$ otherwise. The indistinguishability-based security of the receiver directly follows from that of the SFE and the PKE, which together guarantees that c is hidden by the first-round message. To argue UC security of the sender, we now extract the receiver's bit c by decrypting the ciphertext ct. If ct is an encryption of c then $f_{\mathsf{pk},\mathsf{ct},m_0,m_1}$ is functionally equivalent to $f_{\mathsf{pk},\mathsf{ct},m'_0,m'_1}$ where $m'_c = m_c$ and m'_{1-c} is replaced by an arbitrary value, say all 0s. Therefore, we can simulate the sender's second-round OT message by using the circuit $f_{\mathsf{pk},\mathsf{ct},m'_0,m'_1}$, which only relies on knowledge of m_c without knowing m_{1-c}, and weak SFE security guarantees that this is indistinguishable from the real world.

From UC Sender Security to Full UC OT. Finally, we show how to use an OT scheme with UC-security of the sender and indistinguishability-based security for the receiver ("half-UC OT") to get a full UC-secure OT. In particular, this means that we need to simulate the receiver's first-round message without knowing c and extract two values m_0, m_1 from a malicious sender such that, if the receiver's bit was c, he would get m_c.

Before we give our actual construction, it is useful to examine a naive proposal and why it fails. In the naive proposal, the sender commits to both values m_0, m_1 using an extractable commitment (e.g., PKE where the public key is in the CRS); the parties use a half-UC OT where the sender puts the two decommitments as his OT values and also sends the commitments as part of the second-round OT message. We can extract two values m_0, m_1 from the commitment and are guaranteed that the receiver either outputs the value m_c or $\perp$ (if the decommitment he receives via the underlying OT is incorrect). But we are unable to say which of the two cases will occur. This is insufficient for full security.

We solve the above problem via two steps:

- We first give a solution using a two-round zero-knowledge (ZK) argument and an extractable commitment (both in the CRS model). The sender and receiver run the half-UC OT protocol where the receiver uses her choice bit c and the sender uses his two values m_0, m_1. In the first round, the receiver also sends the first-round verifier message of the ZK argument. In the second round, the sender also commits to his two messages m_0, m_1 using an extractable commitment and uses the ZK argument system to prove that he computed the second-round OT message correctly using the same values m_0, m_1 as in the commitment. This provides UC security for the receiver since, if the ZK argument verifies, we can extract the values m_0, m_1 from the commitment and know that the receiver would recover the correct value m_c. The transformation also preserves UC security for the sender since the ZK argument can be simulated.

- We then show how to construct a two-round ZK argument using half-UC OT. We rely on a Σ-protocol for NP where the prover sends a value a, receives a 1-bit challenge $b \in \{0, 1\}$, and sends a response z; the verifier checks that the transcript (a, b, z) is valid for the statement being proved and accepts or rejects accordingly. We can compile a Σ-protocol to a two-round ZK argument using OT. The verifier sends a first-round OT message for a random bit b. The prover chooses a and computes both responses z_0, z_1 corresponding to both possible values of the challenge b; he then sends a and uses z_0, z_1 as the values for the second-round OT message. The verifier recovers z_b from the OT and checks that (a, b, z_b) is a valid transcript of the Σ-protocol. We repeat this in parallel λ (security parameter) times to get negligible soundness error. It turns out that we can prove ZK security by relying on the UC-security for the sender; we can extract the OT choice bits b in each execution and then simulate the Σ-protocol transcript after knowing the challenge bit b. It would also be easy to prove soundness using UC-security for the receiver, but we want to only rely on a "half-UC" OT where we only have indistinguishability security of the receiver. To solve this, we rely on a special type of "extractable" Σ-protocol [HL18] in the CRS model, where, for every choice of a there is a unique "bad challenge" b such that, if the statement is false, there exists a valid response z

that results in a valid transcript (a, b, z). Furthermore, this unique bad challenge b should be efficiently extractable from a using a trapdoor to the CRS. Such "extractable" Σ-protocols can be constructed from only public-key encryption. If the Σ-protocol is extractable and the OT scheme has indistinguishability-based receiver security then the resulting two-round ZK is computationally sound. This is because, the only way that the prover can succeed is if in each of the λ invocations he chooses a first message a such that the receiver's OT choice bit b is the unique bad challenge for a, but this means that the prover can predict the receiver's OT choice bits (the reduction uses the trapdoor for the Σ-protocol to extract the unique bad challenge from a).

Combined together, the above two steps give a general compiler from half-UC OT to fully secure UC OT.

Instantiation from CDH. We now give our simple instantiation of elementary OT under the CDH assumption. The construction is based on a scheme of Bellare and Micali [BM90], which achieves a weak form of malicious security in the random-oracle model. Our protocol is somewhat simplified and does not require a random oracle. Recall that the CDH assumption states that, given a generator g of some cyclic group $\mathbb{G}$ of order p, along with values g^a, g^b for random $a, b \in \mathbb{Z}_p$, it is hard to compute g^{ab}.

The CRS of the OT scheme consists of $A = g^a$ for random $a \in \mathbb{Z}_p$. The receiver with a choice bit c computes two value $h_c = g^r$ and $h_{1-c} = A/h_c$ for a random $r \in \mathbb{Z}_p$ and sends $\mathsf{otr} := h_0$ as the first-round OT message. The sender computes $h_1 = A/h_0$. It chooses a random $b \in \mathbb{Z}_p$, sets $\mathsf{ots} := B = g^b$ as the second-round message, and generates the two values $y_0 = h_0^b, y_1 = h_1^b$. The receiver outputs $\hat{y}_c = B^r$.

This ensures correctness since $\hat{y}_c = B^r = g^{br} = h_c^b = y_c$. Also, h_0 is uniformly random over $\mathbb{G}$ no matter what the receiver bit c is, and therefore this provides (statistic) indistinguishability-based receiver security. Lastly, we argue that we get elementary OT security for the sender, meaning that a malicious receiver cannot simultaneously compute both y_0, y_1. Note that the only values seen by the malicious receiver during the game are $A = g^a, B = g^b$. If the receiver outputs $y_0 = h_0^b, y_1 = h_1^b = (A/h_0)^b$ then we can use these values to compute $y_0 \cdot y_1 = A^b = g^{ab}$, which breaks CDH.

Instantiation from LPN. We also give a simple instantiation of elementary OT under the LPN assumption. This construction closely mirrors the CDH one. We use a variant of the LPN problem with noise-rate $1/n^{\varepsilon}$ for an arbitrary constant $\varepsilon > \frac{1}{2}$. We also rely on a variant of the LPN problem where the secret is chosen from the error distribution, which is known to be equivalent to standard LPN where the secret is uniformly random [ACPS09]. In particular this variant of the LPN problem states that, for a Bernoulli distribution $\mathcal{B}_\rho$ which outputs 1 with probability $\rho = 1/n^{\varepsilon}$, and for $A \leftarrow \mathbb{Z}_2^{n \times n}$, $s, e \leftarrow \mathcal{B}_\rho^n$, the values $(A, sA + e)$ are indistinguishable from uniformly random values.

The CRS of the OT scheme consists of a tuple (A, v) where $A \leftarrow \mathbb{Z}_2^{n \times n}$ and $v \leftarrow \mathbb{Z}_2^n$. The receiver chooses $x, e \leftarrow \mathcal{B}_\rho^n$ and sets $h_c = Ax + e$ and $h_{1-c} = v - h_c$ and sends $\mathsf{otr} = h_0$ as the first-round OT message. The sender computes $h_1 = h_0 + v$, chooses $S, E \leftarrow \mathcal{B}_\rho^{\lambda \times n}$ where λ is the security parameter and sends $\mathsf{ots} := B = SA + E$ as the second-round OT message. The sender computes the values $y_0 = Sh_0, y_1 = Sh_1$. The receiver outputs $\hat{y}_c = Bx$.

This ensures correctness with a small inverse-polynomial error probability. In particular, $y_c = Sh_c = S(Ax+e) = Bx + Se - Ex = \hat{y}_c + (Se - Ex)$ where $Ex + Se = 0$ except with a small error probability, which we can make an arbitrarily small inverse polynomial in λ by setting n to be a sufficiently large polynomial in λ. The receiver's (computational) indistinguishability-based security holds under LPN since h_0 is indistinguishable from uniform no matter what c is. We also get elementary OT security for the sender under the LPN assumption. A malicious receiver only sees the values A, v and $B = SA + E$ during the game. If the receiver outputs $y_0 = Sh_0, y_1 = Sh_1$, then we can use it to compute $y_0 + y_1 = S(h_0 + h_1) = Sv$. But, since S is hard to compute given A, B, we can argue that Sv is indistinguishable form uniform under the LPN assumption, by thinking of the i'th of Sv as a Goldreich-Levin hardcore bit for the i'th row of S. Therefore, is should be hard to output Sv except with negligible probability.

The fact that we get a small (inverse polynomial) error probability does not affect the security of the generic transformations going from elementary OT to indistinguishability OT for 1-bit messages. Then, when we go from 1-bit messages to multi-bit messages we can also use an error-correcting code to amplify correctness and get a negligible correctness error.

3 Preliminaries

Notation. We use λ for the security parameter. We use $\stackrel{c}{\equiv}$ to denote computational indistinguishability between two distributions and use $\equiv$ to denote two distributions are identical. For a distribution D we use $x \stackrel{\$}{\leftarrow} D$ to mean x is sampled according to D and use $y \in D$ to mean y is in the support of D. For a set S we overload the notation to use $x \stackrel{\$}{\leftarrow} \mathsf{S}$ to indicate that x is chosen uniformly at random from S.

3.1 Basic Inequalities

Lemma 3.1 (Markov Inequality for Advantages)**.** *Let $A(Z)$ and $B(Z)$ be two random variables depending on a random variable Z and potentially additional random choices. Assume that $|\Pr_Z[A(Z) = 1] - \Pr_Z[B(Z) = 1]| \geq \epsilon \geq 0$. Then*

$$\Pr_Z[|\Pr[A(Z) = 1] - \Pr[B(Z) = 1]| \geq \epsilon/2] \geq \epsilon/2.$$

Proof. Let $a := \Pr_Z[|\Pr[A(Z) = 1] - \Pr[B(Z) = 1]| \geq \epsilon/2]$. We have $\epsilon \leq a \times 1 + (1-a) \times \epsilon/2$. Since $0 \leq 1 - a \leq 1$, we obtain $\epsilon \leq a + \epsilon/2$. The inequality now follows. □

Theorem 3.2 (Hoeffding Inequality)**.** *Let $X_1, \ldots, X_N \in [0,1]$ be i.i.d. random variables with expectation $\mathsf{E}[X_1]$. Then it holds that*

$$\Pr\left[\left|\frac{1}{N}\sum_i X_i - \mathsf{E}[X_1]\right| > \delta\right] \leq 2e^{-2N\delta^2}.$$

3.2 Standard Primitives

Definition 3.3 (PKE)**.** *The notion of CPA security for a PKE scheme $\mathsf{PKE} = (\mathsf{KeyGen}, \mathsf{E}, \mathsf{Dec})$ is standard. We say that PKE is perfectly correct if $\Pr[\exists(\mathsf{m}, \mathsf{r})$ s.t. $\mathsf{Dec}(\mathsf{sk}, \mathsf{E}(\mathsf{pk}, \mathsf{m}; \mathsf{r})) \neq \mathsf{m}] = \mathsf{negl}(\lambda)$, where $(\mathsf{pk}, \mathsf{sk}) \stackrel{\$}{\leftarrow} \mathsf{KeyGen}(1^\lambda)$.*

Definition 3.4 (Garbled Circuits)**.** *A garbling scheme for a class of circuits $\mathcal{C}$ with n-bit inputs consists of* $(\mathsf{Garble}, \mathsf{Eval}, \mathsf{Sim})$ *with the following correctness and security properties.*

- *Correctness: for all* $\mathsf{C} \in \mathcal{C}$ *and all* $\mathsf{x} \in \{0,1\}^n$*, we have* $\Pr[\mathsf{Eval}(\widehat{\mathsf{C}}, \mathsf{GarbleInput}(\vec{\mathsf{lb}}^0, \vec{\mathsf{lb}}^1, \mathsf{x})) = \mathsf{C}(\mathsf{x})] = 1$*, where* $(\widehat{\mathsf{C}}, \vec{\mathsf{lb}}^0, \vec{\mathsf{lb}}^1) \xleftarrow{\$} \mathsf{Garble}(1^\lambda, \mathsf{C})$*,* $\vec{\mathsf{lb}}^0 := (\mathsf{lb}_1^0, \ldots, \mathsf{lb}_n^0)$*,* $\vec{\mathsf{lb}}^1 := (\mathsf{lb}_1^1, \ldots, \mathsf{lb}_n^1)$ *and we define* $\mathsf{GarbleInput}(\vec{\mathsf{lb}}^0, \vec{\mathsf{lb}}^1, \mathsf{x}) := (\mathsf{lb}_1^{\mathsf{x}_1}, \ldots, \mathsf{lb}_n^{\mathsf{x}_n})$.

- *Security: For any* $\mathsf{C} \in \mathcal{C}$ *and* $\mathsf{x} \in \{0,1\}^n$*:* $(\widehat{\mathsf{C}}, \mathsf{GarbleInput}(\vec{\mathsf{lb}}^0, \vec{\mathsf{lb}}^1, \mathsf{x})) \stackrel{c}{\equiv} \mathsf{Sim}(1^\lambda, \mathsf{C}(\mathsf{x}))$*, where* $(\widehat{\mathsf{C}}, \vec{\mathsf{lb}}^0, \vec{\mathsf{lb}}^1) \xleftarrow{\$} \mathsf{Garble}(1^\lambda, \mathsf{C})$.

4 Definitions of Two-Round Oblivious Transfer

A two-round oblivious transfer (OT) protocol (we use the definition from [BGI+17]) is given by algorithms $(\mathsf{Setup}, \mathsf{OT}_1, \mathsf{OT}_2, \mathsf{OT}_3)$, where the setup algorithm Setup generates a CRS value $\mathsf{crs} \xleftarrow{\$} \mathsf{Setup}(1^\lambda)$.[3] The receiver runs the algorithm OT_1 which takes crs and a choice bit $c \in \{0,1\}$ as input and outputs $(\mathsf{otr}, \mathsf{st})$. The receiver then sends otr to the sender, who obtains ots by evaluating $\mathsf{OT}_2(1^\lambda, \mathsf{otr}, \mathsf{m}_0, \mathsf{m}_1)$, where m_0 and m_1 (such that $\mathsf{m}_0, \mathsf{m}_1 \in \{0,1\}\lambda$) are its inputs. The sender then sends ots to the receiver who obtains m_c by evaluating $\mathsf{OT}_3(1^\lambda, \mathsf{st}, \mathsf{ots})$.

4.1 Correctness

We say that a two-round OT scheme is *perfectly correct*, if with probability $1 - \mathsf{negl}(\lambda)$ over the choice of $\mathsf{crs} \xleftarrow{\$} \mathsf{Setup}(1^\lambda)$ the following holds: for every choice bit $c \in \{0,1\}$ of the receiver and input messages m_0 and m_1 of the sender, and for any $(\mathsf{otr}, \mathsf{st}) \in \mathsf{OT}_1(\mathsf{crs}, c)$ and $\mathsf{ots} \in \mathsf{OT}_2(\mathsf{crs}, \mathsf{otr}, \mathsf{m}_0, \mathsf{m}_1)$, we have $\mathsf{OT}_3(\mathsf{st}, \mathsf{ots}) = \mathsf{m}_c$. (Recall that $x \in \mathcal{D}$ for a distributions $\mathcal{D}$ means that x is in the support of $\mathcal{D}$.)

4.2 Receiver's Security Notions

We consider two notions of receiver's security — namely, notions that require security against a malicious sender. We describe them next.

Receiver's indistinguishability security. For every non-uniform polynomial-time adversary $\mathcal{A}$: $|\Pr[\mathcal{A}(\mathsf{crs}, \mathsf{OT}_1(\mathsf{crs}, 0)) = 1] - \Pr[\mathcal{A}(\mathsf{crs}, \mathsf{OT}_1(\mathsf{crs}, 1)) = 1]| = \mathsf{negl}(\lambda)$, where $\mathsf{crs} \xleftarrow{\$} \mathsf{Setup}(1^\lambda)$.

Receiver's UC-security. We work in Canetti's UC framework with static corruptions [Can01]. We assume familiarity with this model. We use $\mathcal{Z}$ for denoting the underlying environment. For a real protocol Π and an adversary $\mathcal{A}$, we use $\mathsf{EXEC}_{\Pi,\mathcal{A},\mathcal{Z}}$ to denote the real-world ensemble. Also, for an ideal functionality $\mathcal{F}$ and an adversary $\mathcal{S}$ we denote $\mathsf{IDEAL}_{\mathcal{F},\mathcal{S},\mathcal{Z}}$ to denote the ideal-world ensemble.

We say that an OT protocol OT is receiver-UC secure if for any adversary $\mathcal{A}$ corrupting the sender, there exists a simulator $\mathcal{S}$ such that for all environments $\mathcal{Z}$:

[3] Some variants of two-round OT do not need a CRS. In this case, we will assume Setup as the identity function.

$$\mathsf{IDEAL}_{\mathcal{F}_{\mathrm{OT}},\mathcal{S},\mathcal{Z}} \overset{c}{\equiv} \mathsf{EXEC}_{\mathsf{OT},\mathcal{A},\mathcal{Z}},$$

where the ideal functionality $\mathcal{F}_{\mathrm{OT}}$ is defined in Figure 2. (We will follow the same style as in [CLOS02, PVW08].)

$\mathcal{F}_{\mathrm{OT}}$ interacts with an ideal sender $\mathbf{S}$ and an ideal receiver $\mathbf{R}$.

1. On input $(\mathsf{sid}, \text{sender}, \mathsf{m}_0, \mathsf{m}_1)$ from the sender, store $(\mathsf{m}_0, \mathsf{m}_1)$.
2. On input $(\mathsf{sid}, \text{receiver}, b)$, check if a pair of inputs $(\mathsf{m}_0, \mathsf{m}_1)$ has been already recorded for session sid; if so, send m_b to $\mathbf{R}$ and send sid to the adversary and halt; else, send nothing.

Figure 2: Ideal Functionality $\mathcal{F}_{\mathrm{OT}}$

Since our OT protocols are in the CRS model, we also give the $\mathcal{F}_{\mathrm{CRS}}$ idea functionality below.

$\mathcal{F}^{\mathcal{D}}_{\mathrm{CRS}}$: parameterized over a distribution $\mathcal{D}$, run by parties $P_1, \ldots, P_n$, and an adversary $\mathcal{S}$:

- Whenever receiving message a message (sid, P_i, P_j) from party P_i, sample $\mathsf{crs} \overset{\$}{\leftarrow} \mathcal{D}$ and send $(\mathsf{sid}, \mathsf{crs})$ to P_i and send $(\mathsf{sid}, \mathsf{crs}, P_i, P_j)$ to $\mathcal{S}$. Whenever receiving the message (sid, P_i, P_j) from P_j, send $(\mathsf{sid}, \mathsf{crs})$ to P_j and $\mathcal{S}$.

Figure 3: Ideal Functionality $\mathcal{F}^{\mathcal{D}}_{\mathrm{CRS}}$ [CR03]

4.3 Sender's Security Notions

We consider several different notions of sender's security that we define below. In the first two notions of security, namely elementary and search notions, we change the syntax of OT_2 a bit. More specifically, instead of taking m_0 and m_1 as input, OT_2 outputs two masks y_0 and y_1 where the receiver only gets y_c, where c is the receiver's choice bit.

Sender's Elementary Security. The elementary sender security corresponds to the weakest security notion against a malicious receiver that is considered in this work. This notion requires that the receiver actually compute both the strings y_0 and y_1 used by the sender. Let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be an adversary. Consider the following experiment $\mathsf{Exp}^{\lambda}_{\mathsf{eOT}}(\mathcal{A})$:

1. Run $\mathsf{crs} \overset{\$}{\leftarrow} \mathsf{Setup}(1^\lambda)$.
2. Run $(\mathsf{otr}, \mathsf{st}) \overset{\$}{\leftarrow} \mathcal{A}_1(1^\lambda, \mathsf{crs})$
3. Compute $(\mathsf{ots}, \mathsf{y}_0, \mathsf{y}_1) \overset{\$}{\leftarrow} \mathsf{OT}_2(\mathsf{crs}, \mathsf{otr})$
4. Compute $(\mathsf{y}^*_0, \mathsf{y}^*_1) \overset{\$}{\leftarrow} \mathcal{A}_2(\mathsf{st}, \mathsf{ots})$ and output 1 iff $(\mathsf{y}^*_0, \mathsf{y}^*_1) = (\mathsf{y}_0, \mathsf{y}_1)$

We say that a scheme satisfies eOT security if $\Pr[\mathsf{Exp}^{\lambda}_{\mathsf{eOT}}(\mathcal{A}) = 1] = \mathsf{negl}(\lambda)$.

Sender's Search Security. Next, we consider the search security notion. In this stronger security notion, the adversary is expected to still compute both y_0 and y_1 but perhaps not necessarily at the same time. More formally, let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be an adversary where $\mathcal{A}_2$ outputs a message y^*. Consider the following experiment $\mathsf{Exp}_{\mathsf{sOT}}^{\mathsf{crs},\mathsf{r},w}(\mathcal{A})$, indexed by a crs, random coins $\mathsf{r} \in \{0,1\}^\lambda$ and a bit $w \in \{0,1\}$.

1. Run $(\mathsf{otr}, \mathsf{st}) \xleftarrow{\$} \mathcal{A}_1(1^\lambda, \mathsf{crs}; \mathsf{r})$
2. Compute $(\mathsf{ots}, \mathsf{y}_0, \mathsf{y}_1) \xleftarrow{\$} \mathsf{OT}_2(\mathsf{crs}, \mathsf{otr})$
3. Compute $\mathsf{y}^* \xleftarrow{\$} \mathcal{A}_2(\mathsf{st}, \mathsf{ots}, w)$ and output 1 iff $\mathsf{y}^* = \mathsf{y}_w$

We say a PPT adversary $\mathcal{A}$ breaks the sender search privacy if there exist a non-negligible function ϵ such that

$$\Pr_{\mathsf{crs},\mathsf{r}}[\Pr[\mathsf{Exp}_{\mathsf{sOT}}^{\mathsf{crs},\mathsf{r},0}(\mathcal{A}) = 1] > \epsilon \text{ and } \Pr[\mathsf{Exp}_{\mathsf{sOT}}^{\mathsf{crs},\mathsf{r},1}(\mathcal{A}) = 1] > \epsilon] > \epsilon,$$

where $\mathsf{crs} \xleftarrow{\$} \mathsf{Setup}(1^\lambda)$ and $\mathsf{r} \xleftarrow{\$} \{0,1\}^\lambda$.

Sender's Indistinguishability Security (iOT). Moving on, we consider the sender's indistinguishability security notion (or the iOT notion for short). In this notion, we require that the receiver does not learn any information about either m_0 or m_1. More formally, let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be an adversary where $\mathcal{A}_2$ outputs a bit s. Consider the following experiment $\mathsf{Exp}_{\mathsf{iOT}}^{\mathsf{crs},\mathsf{r},w,b}(\mathcal{A})$, indexed by a crs, random coins $\mathsf{r} \in \{0,1\}^\lambda$, a bit $w \in \{0,1\}$ and a bit $b \in \{0,1\}$.

1. Run $(\mathsf{m}_0, \mathsf{m}_1, \mathsf{otr}, \mathsf{st}) \xleftarrow{\$} \mathcal{A}_1(1^\lambda, \mathsf{crs}; \mathsf{r})$
2. If $b = 0$ compute $\mathsf{ots} \xleftarrow{\$} \mathsf{OT}_2(\mathsf{crs}, \mathsf{otr}, \mathsf{m}_0, \mathsf{m}_1)$
3. Otherwise, if $b = 1$ compute $\mathsf{ots} \xleftarrow{\$} \mathsf{OT}_2(\mathsf{crs}, \mathsf{otr}, \mathsf{m}'_0, \mathsf{m}'_1)$ where $\mathsf{m}'_w \xleftarrow{\$} \{0,1\}^n$ and $\mathsf{m}'_{1-w} = \mathsf{m}_{1-w}$.
4. Compute and output $s \xleftarrow{\$} \mathcal{A}_2(\mathsf{st}, \mathsf{ots})$

Define the advantage of $\mathcal{A}$ as $\mathsf{Adv}_{\mathsf{iOT}}^{\mathsf{crs},\mathsf{r},w}(\mathcal{A}) = |\Pr[\mathsf{Exp}_{\mathsf{iOT}}^{\mathsf{crs},\mathsf{r},w,0}(\mathcal{A}) = 1] - \Pr[\mathsf{Exp}_{\mathsf{iOT}}^{\mathsf{crs},\mathsf{r},w,1}(\mathcal{A}) = 1]|$. We say a PPT adversary $\mathcal{A}$ breaks the sender's indistinguishability security if there exist a non-negligible function ϵ such that

$$\Pr_{\mathsf{crs},\mathsf{r}}[\mathsf{Adv}_{\mathsf{iOT}}^{\mathsf{crs},\mathsf{r},0}(\mathcal{A}) > \epsilon \text{ and } \mathsf{Adv}_{\mathsf{iOT}}^{\mathsf{crs},\mathsf{r},1}(\mathcal{A}) > \epsilon] > \epsilon,$$

where $\mathsf{crs} \xleftarrow{\$} \mathsf{Setup}(1^\lambda)$ and $\mathsf{r} \xleftarrow{\$} \{0,1\}^\lambda$.

In the experiment above, if the two messages m_0 and m_1 are single-bits, then call the notion bit iOT. Otherwise, we call the notion string iOT.

Sender's UC-security. We say that an OT protocol OT is sender-UC secure if for any adversary $\mathcal{A}$ corrupting the receiver, there exists a simulator $\mathcal{S}$ such that for all environments $\mathcal{Z}$:

$$\mathsf{IDEAL}_{\mathcal{F}_{\mathrm{OT}},\mathcal{S},\mathcal{Z}} \stackrel{c}{\equiv} \mathsf{EXEC}_{\mathsf{OT},\mathcal{A},\mathcal{Z}},$$

where the ideal functionality $\mathcal{F}_{\mathrm{OT}}$ is defined in Figure 2.

Definition 4.1. *For $\mathcal{X} \in \{$elementary, search, indistinguishability$\}$, we call a two-round OT scheme $\mathcal{X}$-secure if it has sender's $\mathcal{X}$ security and receiver's indistinguishability security. Moreover, we call a two-round OT scheme UC-secure if it has sender's UC-security and receiver's UC-security.*

5 Transformations for Achieving Sender's Indistinguishability

In this section, we give a sequence of transformations which leads us to sender's indistinguishability security, starting with sender's elementary security.

5.1 From Elementary OT to Search OT

We rely on a result of [CHS05] on hardness amplification of weakly verifiable puzzles. In such puzzles, a puzzle generator can efficiently verify solutions but others need not be able to; we rely on a restricted case where the solution is unique and the puzzle generator generates the puzzle with the solution. The result essentially says that solving many puzzles is much harder than solving a single puzzle. For simplicity, we state a simplified version of their result (restatement of Lemma 1 in [CHS05]) with a restricted range of parameters. It shows that, if there is a "weak solver" that has some inverse polynomial advantage in solving λ puzzles simultaneously, then there is an "amplified solver" that has extremely high advantage (arbitrarily close to 1) in solving an individual puzzle.

Lemma 5.1 (Hardness Amplification [CHS05])**.** *For every polynomial p and every constant $\delta > 0$ there exists a PPT algorithm Amp such that the following holds for all sufficiently large $\lambda \in \mathbb{N}$. Let G be some distribution over pairs $(\mathsf{puzzle}, \mathsf{solution}) \leftarrow G$. Let WS be a "weak solver" such that*

$$\Pr[\mathsf{WS}(\mathsf{puzzle}_1, \ldots, \mathsf{puzzle}_\lambda) = (\mathsf{solution}_1, \ldots, \mathsf{solution}_\lambda)] \geq 1/p(\lambda)$$

where $(\mathsf{puzzle}_i, \mathsf{solution}_i) \stackrel{\$}{\leftarrow} G$ for $i \in \{1, \ldots, \lambda\}$. Then

$$\Pr[\mathsf{Amp}^{\mathsf{WS},G}(1^\lambda, \mathsf{puzzle}^*) = \mathsf{solution}^*] \geq \delta$$

where $(\mathsf{puzzle}^*, \mathsf{solution}^*) \stackrel{\$}{\leftarrow} G$.

Construction of Search OT. Let $\Pi = (\mathsf{Setup}, \mathsf{OT}_1, \mathsf{OT}_2, \mathsf{OT}_3)$ be an elementary OT. We construct a search OT scheme $\Pi' = (\mathsf{Setup}, \mathsf{OT}_1, \mathsf{OT}'_2, \mathsf{OT}'_3)$ as follows:

- $(\mathsf{ots}', Y_0, Y_1) \stackrel{\$}{\leftarrow} \mathsf{OT}'_2(\mathsf{otr}')$: Sample $(\mathsf{ots}^i, y_0^i, y_1^i) \stackrel{\$}{\leftarrow} \mathsf{OT}_2(\mathsf{crs}, \mathsf{otr})$ for $i = 1, \ldots, \lambda$. Output $\mathsf{ots}' = (\mathsf{ots}^1, \ldots, \mathsf{ots}^\lambda)$ and $Y_0 = (\mathsf{y}_0^1, \ldots, \mathsf{y}_0^\lambda)$, $Y_1 = (\mathsf{y}_1^1, \ldots, \mathsf{y}_1^\lambda)$.
- $Y \stackrel{\$}{\leftarrow} \mathsf{OT}'_3(\mathsf{ots}', \mathsf{st})$: Parse $\mathsf{ots}' = (\mathsf{ots}^1, \ldots, \mathsf{ots}^\lambda)$. Let $\mathsf{y}_i \stackrel{\$}{\leftarrow} \mathsf{OT}_3(\mathsf{ots}^i, \mathsf{st})$ for $i = 1, \ldots, \lambda$. Output $Y = (\mathsf{y}_1, \ldots, \mathsf{y}_\lambda)$.

Theorem 5.2. *If Π is an elementary OT then Π' described above is a search OT.*

Proof. Assume there is some adversary $\mathcal{A}' = (\mathcal{A}'_1, \mathcal{A}'_2)$ that breaks the search OT security of Π'. That is, there exists some polynomial $p(\cdot)$ and an infinite set of values $\mathsf{Good} \subseteq \mathbb{N}$ such that for all $\lambda \in \mathsf{Good}$:

$$\Pr_{\mathsf{crs},\mathsf{r}}[\Pr[\mathsf{Exp}_{\mathsf{sOT}}^{\mathsf{crs},\mathsf{r},0}(\mathcal{A}') = 1] > 1/p(\lambda) \text{ and } \Pr[\mathsf{Exp}_{\mathsf{sOT}}^{\mathsf{crs},\mathsf{r},1}(\mathcal{A}') = 1] > 1/p(\lambda)] > 1/p(\lambda),$$

where $\mathsf{crs} \xleftarrow{\$} \mathsf{Setup}(1^\lambda)$ and $\mathsf{r} \xleftarrow{\$} \{0,1\}^\lambda$.

Let us define the set Good^+ to consist of values $(\mathsf{crs}, \mathsf{r})$ for which $\Pr[\mathsf{Exp}_{\mathsf{sOT}}^{\mathsf{crs},\mathsf{r},0}(\mathcal{A}') = 1] > 1/p(\lambda)$ and $\Pr[\mathsf{Exp}_{\mathsf{sOT}}^{\mathsf{crs},\mathsf{r},1}(\mathcal{A}') = 1] > 1/p(\lambda)$. Let us fix any such values in Good^+. Note that the choice of $(\mathsf{crs}, \mathsf{r}) \in \mathsf{Good}^+$ also implicitly fixes $(\mathsf{otr}, \mathsf{st}) = \mathcal{A}'_1(\mathsf{crs}; \mathsf{r})$. Therefore, by expanding the definition of $\mathsf{Exp}_{\mathsf{sOT}}^{\mathsf{crs},\mathsf{r},0}(\mathcal{A}')$, for this choice of values, we have that for $w \in \{0,1\}$:

$$\Pr[\mathcal{A}'_2(\mathsf{st}, \mathsf{ots}^1, \ldots, \mathsf{ots}^\lambda, w) = (\mathsf{y}_w^1, \ldots, \mathsf{y}_w^\lambda)] \geq 1/p(\lambda).$$

Let Amp be the success amplification algorithm form Lemma 5.1 with the polynomial p given above and with $\delta = 3/4$. For $w \in \{0,1\}$, let $(\mathsf{puzzle}, \mathsf{solution}) \xleftarrow{\$} G_w$ be the distribution that samples $\mathsf{puzzle} = \mathsf{ots}, \mathsf{solution} = \mathsf{y}_w$ with $(\mathsf{ots}, \mathsf{y}_0, \mathsf{y}_1) \xleftarrow{\$} \mathsf{OT}_2(\mathsf{crs}, \mathsf{otr})$ and let $\mathsf{WS}_w(\mathsf{puzzle}_1, \ldots, \mathsf{puzzle}_\lambda) = \mathcal{A}'_2(\mathsf{st}, \mathsf{puzzle}_1, \ldots, \mathsf{puzzle}_\lambda, w)$ be the weak solver. Then, by applying Lemma 5.1, we have:

$$\Pr[\mathsf{Amp}^{\mathsf{WS}_w, G_w}(\mathsf{ots}) = \mathsf{y}_w \quad : \quad (\mathsf{ots}, \mathsf{y}_0, \mathsf{y}_1) \xleftarrow{\$} \mathsf{OT}_2(\mathsf{crs}, \mathsf{otr})] \geq 3/4.$$

Finally, define $\mathcal{A}_2(\mathsf{st}, \mathsf{ots})$ to run $\mathsf{y}_w \xleftarrow{\$} \mathsf{Amp}^{\mathsf{WS}_w, G_w}(\mathsf{ots})$ for $w \in \{0,1\}$ and output $\mathsf{y}_0, \mathsf{y}_1$. Then for any fixed choice of values in Good^+ we have:

$$\begin{aligned}
&\Pr[\mathcal{A}_2(\mathsf{st}, \mathsf{ots}) = (\mathsf{y}_0, \mathsf{y}_1) \ : \ (\mathsf{ots}, \mathsf{y}_0, \mathsf{y}_1) \xleftarrow{\$} \mathsf{OT}_2(\mathsf{crs}, \mathsf{otr})] \\
\geq & 1 - \sum_w \Pr[\mathsf{Amp}^{\mathsf{WS}_w, G_w}(\mathsf{ots}) \neq \mathsf{y}_w \ : \ (\mathsf{ots}, \mathsf{y}_0, \mathsf{y}_1) \xleftarrow{\$} \mathsf{OT}_2(\mathsf{crs}, \mathsf{otr})] \\
\geq & \frac{1}{2}
\end{aligned}$$

where the second line follows by the union bound.

Let $\mathcal{A} = (\mathcal{A}'_1, \mathcal{A}_2)$. Then for all $\lambda \in \mathsf{Good}$:

$$\begin{aligned}
&\Pr[\mathsf{Exp}_{\mathsf{eOT}}^{\lambda}(\mathcal{A}) = 1] \\
\geq & \Pr_{\mathsf{crs},\mathsf{r}}[(\mathsf{crs}, \mathsf{r}) \in \mathsf{Good}^+] \Pr[\mathcal{A}_2(\mathsf{st}, \mathsf{ots}) = (\mathsf{y}_0, \mathsf{y}_1) | (\mathsf{crs}, \mathsf{r}) \in \mathsf{Good}^+] \\
\geq & \frac{1}{p(\lambda)} \cdot \frac{1}{2},
\end{aligned}$$

where $\mathsf{crs} \xleftarrow{\$} \mathsf{Setup}(1^\lambda)$, $\mathsf{r} \xleftarrow{\$} \{0,1\}^\lambda$, $(\mathsf{otr}, \mathsf{st}) \xleftarrow{\$} \mathcal{A}'_1(\mathsf{crs}; \mathsf{r})$, $(\mathsf{ots}, \mathsf{y}_0, \mathsf{y}_1) \xleftarrow{\$} \mathsf{OT}_2(\mathsf{crs}, \mathsf{otr})$. This shows that $\mathcal{A}$ breaks the elementary security of Π and therefore concludes the proof of the theorem. $\square$

5.2 From Search OT to Bit iOT

We rely on the following standard result that unpredictability implies indistinguishability.

Lemma 5.3 (Distinguishing Implies Predicting)**.** *There exists a PPT algorithm $\mathcal{P}$ such that the following holds. Let $(z, b) \xleftarrow{\$} D$ be some distribution with $b \in \{0,1\}$ and let $\mathcal{A}$ be an algorithm such that*

$$|\Pr[\mathcal{A}(z,b) = 1] - \Pr[\mathcal{A}(z,b') = 1]| \geq \varepsilon,$$

where $(z,b) \xleftarrow{\$} D$ and $b' \xleftarrow{\$} \{0,1\}$. Then

$$\Pr[\mathcal{P}^{\mathcal{A}}(z) = b] \geq \frac{1}{2} + \varepsilon.$$

Proof. Define $\mathcal{P}^{\mathcal{A}}(z)$ to choose $b \xleftarrow{\$} \{0,1\}$ and call $\mathcal{A}(z,b)$ to get b'. If $b' = 1$, output b, else output $1 - b$. A simple calculation of probabilities shows that $\mathcal{P}$ satisfies the claim of the lemma. □

We also rely on the Goldreich-Levin theorem [GL89]. The following is the key component of the theorem, which shows that there is an efficient local decoder for the Hadamard code.

Lemma 5.4 (Goldreich-Levin Decoding [GL89])**.** *There exists a PPT algorithm* GLDec *and a polynomial $q(\cdot,\cdot)$ such that for any n, ℓ, any $y \in \{0,1\}^n$ and any function $\mathcal{P} : \{0,1\}^n \to \{0,1\}$ satisfying*

$$\Pr_{s \xleftarrow{\$} \{0,1\}^n}[\mathcal{P}(s) = \langle y, s\rangle] \geq \frac{1}{2} + \frac{1}{\ell}$$

we have:

$$\Pr[\mathsf{GLDec}^{\mathcal{P}}(1^n, 1^\ell) = y] \geq \frac{1}{q(n,\ell)}.$$

Let $\Pi = (\mathsf{Setup}, \mathsf{OT}_1, \mathsf{OT}_2, \mathsf{OT}_3)$ be a *search OT* with message length $n = n(\lambda)$. We construct an iOT scheme $\Pi' = (\mathsf{Setup}, \mathsf{OT}'_1, \mathsf{OT}'_2, \mathsf{OT}'_3)$ with 1-bit message as follows:

- $(\mathsf{otr}', \mathsf{st}') \xleftarrow{\$} \mathsf{OT}'_1(\mathsf{crs}, b)$: Let $(\mathsf{otr}, \mathsf{st}) \xleftarrow{\$} \mathsf{OT}_1(\mathsf{crs}, b)$. Output $\mathsf{otr}' = \mathsf{otr}, \mathsf{st}' = (\mathsf{st}, b)$.
- $\mathsf{ots}' \xleftarrow{\$} \mathsf{OT}'_2(\mathsf{otr}', \mathsf{m}_0, \mathsf{m}_1)$: Sample $(\mathsf{ots}, \mathsf{y}_0, \mathsf{y}_1) \xleftarrow{\$} \mathsf{OT}_2(\mathsf{crs}, \mathsf{otr})$. Choose $s_0, s_1 \xleftarrow{\$} \{0,1\}^n$. For $b \in \{0,1\}$, let $c_b = \langle \mathsf{y}_b, s_b\rangle \oplus \mathsf{m}_b$. Output $\mathsf{ots}' = (\mathsf{ots}, s_0, s_1, c_0, c_1)$.
- $M \xleftarrow{\$} \mathsf{OT}'_3(\mathsf{st}', \mathsf{ots}')$: Parse $\mathsf{ots}' = (\mathsf{ots}, s_0, s_1, c_0, c_1)$, $\mathsf{st}' = (\mathsf{st}, b)$. Let $\mathsf{y} \xleftarrow{\$} \mathsf{OT}_3(\mathsf{ots}, \mathsf{st})$. Output $M = c_b \oplus \langle \mathsf{y}, s_b\rangle$.

Theorem 5.5. *If Π is a search OT then Π' is an iOT with 1-bit messages.*

Proof. Assume there is some adversary $\mathcal{A}' = (\mathcal{A}'_1, \mathcal{A}'_2)$ that breaks the iOT security of Π'. That is, there exists some polynomial $p(\cdot)$ and an infinite set $\mathsf{Good} \subseteq \mathbb{N}$ such that for all $\lambda \in \mathsf{Good}$ we have:

$$\Pr_{\mathsf{crs},\mathsf{r}}[\mathsf{Adv}_{\Pi'}^{\mathsf{crs},\mathsf{r},0}(\mathcal{A}') > 1/p(\lambda) \text{ and } \mathsf{Adv}_{\Pi'}^{\mathsf{crs},\mathsf{r},1}(\mathcal{A}') > 1/p(\lambda)] > 1/p(\lambda),$$

Let us define the set Good^+ to consist of values $(\lambda, \mathsf{crs}, \mathsf{r})$ for which $\mathsf{Adv}_{\Pi'}^{\mathsf{crs},\mathsf{r},0}(\mathcal{A}') > 1/p(\lambda)$ and $\mathsf{Adv}_{\Pi'}^{\mathsf{crs},\mathsf{r},1}(\mathcal{A}') > 1/p(\lambda)$. For any $\lambda \in \mathsf{Good}$ we have

$$\Pr_{\mathsf{crs},\mathsf{r}}[(\lambda, \mathsf{crs}, \mathsf{r}) \in \mathsf{Good}^+] \geq 1/p(\lambda). \tag{1}$$

Note that any such choice of $(\lambda, \mathsf{crs}, \mathsf{r}) \in \mathsf{Good}^+$ also implicitly fixes $(\mathsf{m}_0, \mathsf{m}_1, \mathsf{otr}, \mathsf{st}) = \mathcal{A}_1(\mathsf{crs}; \mathsf{r})$. Therefore, by expanding the definition of the advantage, for any choice of such values in Good^+, we have:

$$
\begin{aligned}
|\Pr[\mathcal{A}'_2(\mathsf{st}, (\mathsf{ots}, s_0, s_1, c_0, c_1), w = 0) = 1] - \Pr[\mathcal{A}'_2(\mathsf{st}, (\mathsf{ots}, s, c'_0, c_1), w = 0) = 1]| \geq &\quad 1/p(\lambda) \\
|\Pr[\mathcal{A}'_2(\mathsf{st}, (\mathsf{ots}, s_0, s_1, c_0, c_1), w = 1) = 1] - \Pr[\mathcal{A}'_2(\mathsf{st}, (\mathsf{ots}, s, c_0, c'_1), w = 1) = 1]| \geq &\quad 1/p(\lambda)
\end{aligned}
$$

where the probability is over $(\mathsf{ots}, \mathsf{y}_0, \mathsf{y}_1) \xleftarrow{\$} \mathsf{OT}_2(\mathsf{crs}, \mathsf{otr})$, $s_0, s_1 \xleftarrow{\$} \{0,1\}$ and we define $c_b = \langle \mathsf{y}_b, s_b \rangle \oplus \mathsf{m}_b$ and $c'_b \xleftarrow{\$} \{0,1\}$.

We can use the fact that distinguishing implies predicting (Lemma 5.3) to argue that the above means there is a PPT predictor $\mathcal{P}$ such that for any choice of values in Good^+:

$$\Pr[\mathcal{P}(\mathsf{st}, (\mathsf{ots}, s_0, s_1, c_1), w) = \langle \mathsf{y}_0, s_0 \rangle] \geq \frac{1}{2} + 1/(2p(\lambda)) \tag{2}$$

$$\Pr[\mathcal{P}(\mathsf{st}, (\mathsf{ots}, s_0, s_1, c_0), w) = \langle \mathsf{y}_1, s_1 \rangle] \geq \frac{1}{2} + 1/(2p(\lambda)) \tag{3}$$

where the probabilities are over $(\mathsf{ots}, \mathsf{y}_0, \mathsf{y}_1)$, s_0, s_1 as above.

Let us define the set Good_0^{++} to consist of values $v = (\lambda, \mathsf{crs}, \mathsf{r}, (\mathsf{ots}, \mathsf{y}_0, \mathsf{y}_1, s_1))$ such that the probability in the left-hand side of equation (2) with the fixed choice of the values v, is $\geq \frac{1}{2} + 1/(4p(\lambda))$. We define Good_1^{++} analogously. By an averaging argument(Lemma 3.1), for any $(\lambda, \mathsf{crs}, \mathsf{r}) \in \mathsf{Good}^+$ we have

$$\Pr_{(\mathsf{ots}, \mathsf{y}_0, \mathsf{y}_1, s_1)}[v \in \mathsf{Good}_0^{++}] \geq 1/(4p(\lambda)) \quad , \quad \Pr_{(\mathsf{ots}, \mathsf{y}_0, \mathsf{y}_1, s_0)}[v \in \mathsf{Good}_1^{++}] \geq 1/(4p(\lambda)). \tag{4}$$

By the Goldreich-Levin lemma (Lemma 5.4) there is then some PPT decoder Dec and some polynomial q such that for any fixing of values in Good_0^{++} and Good_1^{++} respectively we have:

$$
\begin{aligned}
&\Pr[\mathsf{Dec}(\mathsf{st}, (\mathsf{ots}, s_1, c_1), w = 0) = \mathsf{y}_0] \geq 1/q(\lambda) \\
&\Pr[\mathsf{Dec}(\mathsf{st}, (\mathsf{ots}, s_0, c_0), w = 1) = \mathsf{y}_1] \geq 1/q(\lambda))
\end{aligned}
$$

where the probability is only over the internal coins of Dec.

We can define an adversary $\mathcal{A}_2(\mathsf{st}, \mathsf{ots}, w)$ that chooses $s_{1-w} \xleftarrow{\$} \{0,1\}^n$ and $c_{1-w} \xleftarrow{\$} \{0,1\}$ and outputs $\mathsf{Dec}(\mathsf{st}, (\mathsf{ots}, s_{1-w}, c_{1-w}), w)$. We write $\mathcal{A}_2(\mathsf{st}, \mathsf{ots}, w; s_{1-w})$ to denote a run over a fixed choice of s_{1-w}. Then for any fixing of values in Good_0^{++} and Good_1^{++} respectively we have

$$
\begin{aligned}
&\Pr[\mathcal{A}_2(\mathsf{st}, \mathsf{ots}, w = 0; s_1) = \mathsf{y}_0] \geq 1/(2q(\lambda)) \\
&\Pr[\mathcal{A}_2(\mathsf{st}, \mathsf{ots}, w = 1; s_0) = \mathsf{y}_1] \geq 1/(2q(\lambda))
\end{aligned}
$$

where the above probabilities are only over the internal coins of $\mathcal{A}_2$ with s_{1-w} fixed; the only difference between $\mathcal{A}_2$ and Dec is that $\mathcal{A}_2$ has to guess the correct bit c_{1-w} and therefore loses a factor of $\frac{1}{2}$ in the success probability. In particular, for any choice of $(\lambda, \mathsf{crs}, \mathsf{r}) \in \mathsf{Good}^+$, there exists a polynomial $q'(\lambda) = (2q(\lambda))(4p(\lambda))$

$$
\begin{aligned}
&\Pr[\mathcal{A}_2(\mathsf{st}, \mathsf{ots}, w = 0) = \mathsf{y}_0] \geq 1/q'(\lambda) \\
&\Pr[\mathcal{A}_2(\mathsf{st}, \mathsf{ots}, w = 1) = \mathsf{y}_1] \geq 1/q'(\lambda)
\end{aligned}
$$

where the probability is now over $(\mathsf{ots}, \mathsf{y}_0, \mathsf{y}_1) \xleftarrow{\$} \mathsf{OT}_2(\mathsf{crs}, \mathsf{otr})$ and all randomness of $\mathcal{A}_2$. This follows by equation (4), which shows that once we fix a choice of values in Good^+ then the probability of ending up in $\mathsf{Good}_0^{++}, \mathsf{Good}_1^{++}$ respectively is $\geq 1/(4p(\lambda))$.

Finally, we define the adversary $\mathcal{A} = (\mathcal{A}_1', \mathcal{A}_2)$. Then for all infinitely many $\lambda \in \mathsf{Good}$ we have, by equation (1), that:

$$\Pr_{\mathsf{crs},\mathsf{r}}[\Pr[\mathcal{A}_2(\mathsf{st}, \mathsf{ots}, w = 0) = \mathsf{y}_0] \geq 1/q'(\lambda) \wedge \Pr[\mathcal{A}_2(\mathsf{st}, \mathsf{ots}, w = 1) = \mathsf{y}_1] \geq 1/q'(\lambda)] \geq 1/p(\lambda),$$

where the inner probability is over $(\mathsf{ots}, \mathsf{y}_0, \mathsf{y}_1) \xleftarrow{\$} \mathsf{OT}_2(\mathsf{crs}, \mathsf{otr})$ and all randomness of $\mathcal{A}_2$, and we define $(\mathsf{otr}, \mathsf{st}) = \mathcal{A}_1(\mathsf{crs}; \mathsf{r})$.

But the above is equivalent to saying that for all infinitely many $\lambda \in \mathsf{Good}$ we have:

$$\Pr_{\mathsf{crs},\mathsf{r}}[\mathsf{Adv}_{\Pi}^{\mathsf{crs},\mathsf{r},0}(\mathcal{A}) > 1/q'(\lambda) \text{ and } \mathsf{Adv}_{\Pi}^{\mathsf{crs},\mathsf{r},1}(\mathcal{A}) > 1/q'(\lambda)] > 1/p(\lambda),$$

in the search OT security game, and therefore $\mathcal{A}$ breaks the search OT security of Π. This completes the proof. □

5.3 From Bit iOT to String iOT

Let $\Pi = (\mathsf{Setup}, \mathsf{OT}_1, \mathsf{OT}_2, \mathsf{OT}_3)$ be an iOT scheme with 1 bit messages. Then, we construct an iOT scheme $\Pi' = (\mathsf{Setup}, \mathsf{OT}_1', \mathsf{OT}_2', \mathsf{OT}_3')$ with message length $n = n(\lambda)$ as follows:

- $(\mathsf{otr}', \mathsf{st}') \xleftarrow{\$} \mathsf{OT}_1'(\mathsf{crs}, b)$: Let $(\mathsf{otr}, \mathsf{st}) \xleftarrow{\$} \mathsf{OT}_1(\mathsf{crs}, b)$. Output $\mathsf{otr}' = \mathsf{otr}, \mathsf{st}' = \mathsf{st}$.
- $\mathsf{ots}' \xleftarrow{\$} \mathsf{OT}_2'(\mathsf{otr}', \mathsf{m}_0, \mathsf{m}_1)$: For each $i \in [n]$, sample $\mathsf{ots}^{(i)} \xleftarrow{\$} \mathsf{OT}_2(\mathsf{crs}, \mathsf{otr}, \mathsf{m}_0^{(i)}, \mathsf{m}_1^{(i)})$, where $\mathsf{m}_0^{(i)}$ and $\mathsf{m}_1^{(i)}$ are the i^{th} bits of m_0 and m_1, respectively. Output $\mathsf{ots}' = \{\mathsf{ots}^{(i)}\}_{i \in [n]}$.
- $M \xleftarrow{\$} \mathsf{OT}_3'(\mathsf{ots}', \mathsf{st}')$: Parse $\mathsf{ots}' = \{\mathsf{ots}^{(i)}\}$, $\mathsf{st}' = (\mathsf{st}, b)$. Let $M^{(i)} \xleftarrow{\$} \mathsf{OT}_3(\mathsf{ots}^{(i)}, \mathsf{st})$ and output M.

Theorem 5.6. *If Π is iOT with 1-bit messages then Π' is an iOT with messages of length n.*

Proof. The receiver's security follows straightforwardly since only otr can reveal the choice bit b and otr is identical in the string and bit iOT.

For sender's indistinguishable security, we need to ensure that a malicious receiver cannot distinguish both m_0^i and m_1^j from a uniform message for any choice of $i, j \in [n]$. We first define $2(n+1)$ hybrids $\mathcal{H}_{1,0}^{\mathsf{crs},\mathsf{r}}, \ldots, \mathcal{H}_{n+1,0}^{\mathsf{crs},\mathsf{r}}, \mathcal{H}_{1,1}^{\mathsf{crs},\mathsf{r}}, \ldots, \mathcal{H}_{n+1,1}^{\mathsf{crs},\mathsf{r}}$. For $j \in [n+1]$ and $w \in \{0,1\}$, $\mathcal{H}_{i,w}^{\mathsf{crs},\mathsf{r}}$ is indexed by a a common reference string crs and random coins $\mathsf{r} \in \{0,1\}^\lambda$ and has the description:

- Run $(\mathsf{m}_0, \mathsf{m}_1, \mathsf{otr}, \mathsf{st}) \xleftarrow{\$} \mathcal{A}_1(1^\lambda, \mathsf{crs}; \mathsf{r})$
- For $0 < j < i$ compute $\mathsf{ots}^{(j)} \xleftarrow{\$} \mathsf{OT}_2(\mathsf{crs}, \mathsf{otr}, \mathsf{m}_0^{(j)}, \mathsf{m}_1^{(j)})$
- For $n \geq j \geq i$ compute $\mathsf{ots}^{(j)} \xleftarrow{\$} \mathsf{OT}_2(\mathsf{crs}, \mathsf{otr}, \hat{M}_0^{(j)}, \hat{M}_1^{(j)})$ where $\hat{M}_w^{(j)} \xleftarrow{\$} \{0,1\}$ and $\hat{M}_{1-w}^{(j)} := \mathsf{m}_{1-w}^{(j)}$.
- Compute and output $s \xleftarrow{\$} \mathcal{A}_2(\mathsf{st}, \mathsf{ots})$, where $\mathsf{ots} = (\mathsf{ots}^{(1)}, \ldots, \mathsf{ots}^{(n)})$.

Notice that $\mathcal{H}^{\mathsf{crs},\mathsf{r}}_{1,w}$ is identical with $\mathsf{Exp}^{\mathsf{crs},\mathsf{r},w,0}_{\mathsf{iOT}}$ and $\mathcal{H}^{\mathsf{crs},\mathsf{r}}_{n+1,w}$ is identical with $\mathsf{Exp}^{\mathsf{crs},\mathsf{r},w,1}_{\mathsf{iOT}}$. Therefore, if there is an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ against the string iOT security of the above constructed OT $\Pi' = (\mathsf{Setup}, \mathsf{OT}'_1, \mathsf{OT}'_2, \mathsf{OT}'_3)$, i.e. there exist a non-negligible function ϵ such that

$$\Pr_{\mathsf{crs},\mathsf{r}}[\mathsf{Adv}^{\mathsf{crs},\mathsf{r},0}_{\mathsf{iOT}}(\mathcal{A}) > \epsilon \text{ and } \mathsf{Adv}^{\mathsf{crs},\mathsf{r},1}_{\mathsf{iOT}}(\mathcal{A}) > \epsilon] > \epsilon,$$

where $\mathsf{crs} \xleftarrow{\$} \mathsf{Setup}(1^\lambda)$ and $\mathsf{r} \xleftarrow{\$} \{0,1\}^\lambda$, then there is a $i, j \in [n]$ such that

$$\Pr_{\mathsf{crs},\mathsf{r}}[|\Pr[\mathcal{H}^{\mathsf{crs},\mathsf{r}}_{i,0}(\mathcal{A}) = 1] - \Pr[\mathcal{H}^{\mathsf{crs},\mathsf{r}}_{i+1,0}(\mathcal{A}) = 1]| > \epsilon' \text{ and } |\Pr[\mathcal{H}^{\mathsf{crs},\mathsf{r}}_{j,1}(\mathcal{A}) = 1] - \Pr[\mathcal{H}^{\mathsf{crs},\mathsf{r}}_{j+1,1}(\mathcal{A}) = 1]| > \epsilon'] > \epsilon,$$

where $\epsilon' > \frac{\epsilon}{n}$. This implies that $\mathcal{A}$ breaks the sender's indistinguishable security of the bit iOT Π for non-negligible function ϵ'. □

6 Weak Secure Function Evaluation

In this section, we will define our notion of weak secure function evaluation and provide instantiations of the new notion.

6.1 Definitions

Definition 6.1. *A weak secure function evaluation scheme* wSFE *for a function class* $\mathcal{F}$ *consists of four PPT algorithms* $(\mathsf{Setup}, \mathsf{Receiver}_1, \mathsf{Sender}, \mathsf{Receiver}_2)$ *with the following syntax.*

$\mathsf{Setup}(1^\lambda)$**:** *Takes as input a security parameter and outputs a common reference string* crs

$\mathsf{Receiver}_1(\mathsf{crs}, x)$: *Takes as input a common reference string* crs *and an input* x *and outputs a message* z_1 *and a state* st

$\mathsf{Sender}(\mathsf{crs}, f, \mathsf{z}_1)$: *Takes as input a common reference string* crs, *a function* $f \in \mathcal{F}$ *and a receiver message* z_1 *and outputs a sender message* z_2

$\mathsf{Receiver}_2(\mathsf{st}, \mathsf{z}_2)$**:** *Takes as input a state* st *and a sender message* z_2 *and outputs a value* y.

We require the following properties.

- ***Correctness:*** *It holds for any* λ, *any* $f \in \mathcal{F}$ *and any* x *in the domain of* f *that*

$$\mathsf{Receiver}_2(\mathsf{st}, \mathsf{Sender}(\mathsf{crs}, f, \mathsf{z}_1)) = f(x),$$

  where $\mathsf{crs} \xleftarrow{\$} \mathsf{Setup}(1^\lambda)$ *and* $(\mathsf{z}_1, \mathsf{st}) \xleftarrow{\$} \mathsf{Receiver}_1(\mathsf{crs}, x)$

- ***Receiver Privacy:*** *Let* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *be an adversary where* $\mathcal{A}_2$ *outputs a bit and let the experiment* $\mathsf{Exp}_{RP}(\mathcal{A})$ *be defined as follows:*

  – *Compute* $\mathsf{crs} \xleftarrow{\$} \mathsf{Setup}(1^\lambda)$

  – *Compute* $(x_0, x_1) \xleftarrow{\$} \mathcal{A}_1(\mathsf{crs})$

  – *Choose* $b \xleftarrow{\$} \{0,1\}$

– *Compute* $\mathsf{z}_1^* \stackrel{\$}{\leftarrow} \mathsf{Receiver}_1(\mathsf{crs}, x_b)$

– *Compute* $b' \stackrel{\$}{\leftarrow} \mathcal{A}_2(\mathsf{crs}, \mathsf{z}_1^*)$

– *If* $b' = b$ *output 1, otherwise 0*

Define $\mathsf{Adv}_{RP}(\mathcal{A}) = |\Pr[\mathsf{Exp}_{RP}(\mathcal{A}) = 1] - 1/2|$. *We say that* wSFE *has computational receiver privacy, if it holds for all PPT adversaries* $\mathcal{A}$ *that* $\mathsf{Adv}_{RP}(\mathcal{A}) < \mathsf{negl}(\lambda)$. *Likewise, we say that* wSFE *has statistical receiver privacy, if it holds for all unbounded (non-uniform) adversaries* $\mathcal{A}$ *that* $\mathsf{Adv}_{RP}(\mathcal{A}) < \mathsf{negl}(\lambda)$.

- ***Sender Privacy:*** *Let* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *be an adversary where* $\mathcal{A}_2$ *outputs a bit and let the experiment* $\mathsf{Exp}_{SP}(\mathcal{A})$ *be defined as follows:*

  – *Compute* $\mathsf{crs} \stackrel{\$}{\leftarrow} \mathsf{Setup}(1^\lambda)$

  – *Compute* $(f_0, f_1, \mathsf{z}_1) \stackrel{\$}{\leftarrow} \mathcal{A}_1(\mathsf{crs})$

  – *Choose* $b \stackrel{\$}{\leftarrow} \{0, 1\}$

  – *Compute* $\mathsf{z}_2^* \stackrel{\$}{\leftarrow} \mathsf{Sender}(\mathsf{crs}, f_b, \mathsf{z}_1)$

  – *Compute* $b' \stackrel{\$}{\leftarrow} \mathcal{A}_2(\mathsf{crs}, \mathsf{z}_2^*)$

  – *If* $b' = b$ *output 1, otherwise 0*

  Define $\mathsf{Adv}_{SP}(\mathcal{A}) = |\Pr[\mathsf{Exp}_{SP}(\mathcal{A}) = 1] - 1/2|$. *We say that* wSFE *has computational sender privacy, if it holds for all PPT adversaries* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *which output equivalent functions* $f_0 \equiv f_1$ *in the first stage that* $\mathsf{Adv}_{SP}(\mathcal{A}) < \mathsf{negl}(\lambda)$. *Likewise, we say that* wSFE *has statistical sender privacy, if it holds for all unbounded (non-uniform) adversaries* $\mathcal{A}$ *which output equivalent functions* $f_0 \equiv f_1$ *in the first stage that* $\mathsf{Adv}_{SP}(\mathcal{A}) < \mathsf{negl}(\lambda)$.

6.2 wSFE for all Circuits from iOT and Garbled Circuits

Let $\mathsf{iOT} = (\mathsf{Setup}, \mathsf{OT}_1, \mathsf{OT}_2, \mathsf{OT}_3)$ be an iOT protocol and let $(\mathsf{Garble}, \mathsf{Eval})$ be a garbling scheme. Overloading notation, assume that if $\vec{x} = (x_1, \ldots, x_n) \in \{0,1\}^n$ is an input vector, then $\mathsf{OT}_1(\mathsf{crs}, \vec{x}) = (\mathsf{OT}_1(\mathsf{crs}, x_1), \ldots, \mathsf{OT}_1(\mathsf{crs}, x_n))$. Similarly, if $\vec{m}_0 = (m_{0,1}, \ldots, m_{0,n})$ and $\vec{m}_1 = (m_{1,1}, \ldots, m_{1,n})$ are two vectors of messages, then denote

$$\mathsf{OT}_2(\mathsf{crs}, \vec{\mathsf{otr}}, \vec{m}_0, \vec{m}_1) = (\mathsf{OT}_2(\mathsf{crs}, \mathsf{otr}^1, m_{0,1}, m_{1,1}), \ldots, \mathsf{OT}_2(\mathsf{crs}, \mathsf{otr}^n, m_{0,n}, m_{1,n}))$$

The scheme wSFE is given as follows.

$\mathsf{Setup}(1^\lambda)$**:** Compute and output $\mathsf{crs} \stackrel{\$}{\leftarrow} \mathsf{iOT.Setup}(1^\lambda)$

$\mathsf{Receiver}_1(\mathsf{crs}, \vec{x} \in \{0,1\}^n)$**:** Compute $(\vec{\mathsf{otr}}, \vec{\mathsf{st}}') \stackrel{\$}{\leftarrow} \mathsf{iOT.OT}_1(\mathsf{crs}, \vec{x})$. Output $\mathsf{z}_1 \stackrel{\$}{\leftarrow} \vec{\mathsf{otr}}$ and $\mathsf{st} \stackrel{\$}{\leftarrow} \vec{\mathsf{st}}'$.

$\mathsf{Sender}(\mathsf{crs}, \mathsf{z}_1 = \vec{\mathsf{otr}}, \mathsf{C})$:

- Compute $(\widehat{\mathsf{C}}, \vec{\mathsf{lb}}^0, \vec{\mathsf{lb}}^1) \stackrel{\$}{\leftarrow} \mathsf{Garble}(1^\lambda, \mathsf{C})$
- Compute $\vec{\mathsf{ots}} \stackrel{\$}{\leftarrow} \mathsf{iOT.OT}_2(\mathsf{crs}, \vec{\mathsf{otr}}, \vec{\mathsf{lb}}^0, \vec{\mathsf{lb}}^1)$.

- Output $\mathsf{z}_2 \xleftarrow{\$} (\vec{\mathsf{ots}}, \widehat{\mathsf{C}})$.

$\mathsf{Receiver}_2(\mathsf{st} = \vec{\mathsf{st}}', \mathsf{z}_2)$:

- Parse $\mathsf{z}_2 = (\vec{\mathsf{ots}}, \widehat{\mathsf{C}})$.
- Compute $\vec{\mathsf{lb}} \xleftarrow{\$} \mathsf{iOT.OT}_3(\vec{\mathsf{st}}', \vec{\mathsf{ots}})$
- Compute $\mathsf{m} \xleftarrow{\$} \mathsf{Eval}(\widehat{\mathsf{C}}, \vec{\mathsf{lb}})$.
- Output m

6.2.1 Correctness

We will briefly argue that the scheme is correct. Thus, let $\mathsf{crs} \xleftarrow{\$} \mathsf{iOT.Setup}(1^\lambda)$ and $(\vec{\mathsf{otr}}, \vec{\mathsf{st}}) \xleftarrow{\$} \mathsf{iOT.OT}_1(\mathsf{crs}, \vec{x})$. Further let $(\widehat{\mathsf{C}}, \vec{\mathsf{lb}}^0, \vec{\mathsf{lb}}^1) \xleftarrow{\$} \mathsf{Garble}(1^\lambda, \mathsf{C})$ and $\vec{\mathsf{ots}} \xleftarrow{\$} \mathsf{iOT.OT}_2(\mathsf{crs}, \vec{\mathsf{otr}}, \vec{\mathsf{lb}}^0, \vec{\mathsf{lb}}^1)$. By the correctness of iOT it holds that

$$\vec{\mathsf{lb}} = \mathsf{iOT.OT}_3(\vec{\mathsf{st}}, \vec{\mathsf{ots}}) = \mathsf{GarbleInput}(\vec{\mathsf{lb}}^0, \vec{\mathsf{lb}}^1, \vec{x}).$$

Furthermore, by the correctness of the garbling scheme $(\mathsf{Garble}, \mathsf{Eval})$ it holds that

$$m = \mathsf{Eval}(\widehat{\mathsf{C}}, \vec{\mathsf{lb}}) = \mathsf{Eval}(\widehat{\mathsf{C}}, \mathsf{GarbleInput}(\vec{\mathsf{lb}}^0, \vec{\mathsf{lb}}^1, \vec{x})) = \mathsf{C}(\vec{x}),$$

and we get that wSFE is correct.

6.2.2 Receiver Privacy

We will first establish receiver privacy of wSFE.

Theorem 6.2. *Assume that* iOT *has receiver indistinguishability security. The* wSFE *has receiver privacy.*

The proof of Theorem 6.2 follows via standard techniques.

Proof. Let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be a PPT adversary against the receiver privacy of wSFE with advantage ϵ. Consider the following hybrids.

- Hybrid $\mathcal{H}_0$: This is the real receiver privacy experiment with choice bit $b = 0$, i.e. we compute $\vec{\mathsf{otr}}$ by $(\vec{\mathsf{otr}}, \vec{\mathsf{st}}) \xleftarrow{\$} \mathsf{iOT.OT}_1(\mathsf{crs}, \vec{x}_0)$.
- Hybrid $\mathcal{H}_i$ (for $i = 1, \ldots, n$): This is the same as hybrid $\mathcal{H}_{i-1}$, except that we compute otr_i by $(\mathsf{otr}_i, \mathsf{st}_i) \xleftarrow{\$} \mathsf{iOT.OT}_1(\mathsf{crs}, x_{1,i})$ instead of $(\mathsf{otr}_i, \mathsf{st}_i) \xleftarrow{\$} \mathsf{iOT.OT}_1(\mathsf{crs}, x_{0,i})$.

Observe that in $\mathcal{H}_n$ we compute $\vec{\mathsf{otr}}$ by $(\vec{\mathsf{otr}}, \vec{\mathsf{st}}) \xleftarrow{\$} \mathsf{iOT.OT}_1(\mathsf{crs}, \vec{x}_1)$. Thus $\mathcal{H}_n$ is the real receiver privacy experiment with choice bit $b = 1$. Thus, it holds that

$$|\Pr[\mathcal{H}_n(\mathcal{A}) = 1] - \Pr[\mathcal{H}_0(\mathcal{A}) = 1]| \geq \epsilon,$$

Consequently, there must be an $i^* \in [n]$ such that

$$|\Pr[\mathcal{H}_{i^*}(\mathcal{A}) = 1] - \Pr[\mathcal{H}_{i^*-1}(\mathcal{A}) = 1]| \geq \epsilon/n.$$

We will now construct a PPT adversary $\mathcal{B}$ with advantage ϵ/n against the receiver privacy of iOT. For simplicity, we will use an equivalent notion of iOT receiver privacy where the the adversary outputs two bits (β_0, β_1) and the experiment returns otr^* computed by $(\mathsf{otr}^*, \mathsf{st}^*) \stackrel{\$}{\leftarrow} \mathsf{iOT.OT}_1(\mathsf{crs}, \beta_b)$.

$\mathcal{B}_1(1^\lambda, \mathsf{crs})$:

- Run $\mathcal{H}_{i^*}(\mathcal{A})$ until before otr_i is computed. Output $(x_{i,0}, x_{i,1})$.

$\mathcal{B}_2(1^\lambda, \mathsf{crs}, \mathsf{otr}^*)$:

- Set $\mathsf{otr}_i \stackrel{\$}{\leftarrow} \mathsf{otr}^*$ and continue the simulation.
- Output whatever the simulated $\mathcal{H}_{i^*}(\mathcal{A})$ outputs.

We will now analyse the advantage of $\mathcal{B}$.

1. Assume first that otr^* was computed by $(\mathsf{otr}^*, \mathsf{st}^*) \stackrel{\$}{\leftarrow} \mathsf{iOT.OT}_1(\mathsf{crs}, x_{0,i})$. In this case $\mathcal{B}$ perfectly simulates $\mathcal{H}_{i^*-1}(\mathcal{A})$ and we get that $\Pr[\mathcal{B}(1^\lambda, \mathsf{crs}, \mathsf{otr}^*) = 1] = \Pr[\mathcal{H}_{i^*-1}(\mathcal{A}) = 1]$.

2. On the other hand, if otr^* was computed by $(\mathsf{otr}^*, \mathsf{st}^*) \stackrel{\$}{\leftarrow} \mathsf{iOT.OT}_1(\mathsf{crs}, x_{1,i})$, then $\mathcal{B}$ perfectly simulates $\mathcal{H}_{i^*}(\mathcal{A})$ and we get $\Pr[\mathcal{B}(1^\lambda, \mathsf{crs}, \mathsf{otr}^*) = 1] = \Pr[\mathcal{H}_{i^*}(\mathcal{A}) = 1]$.

Consequently, we get that

$$|\Pr[\mathsf{Exp}^1(\mathcal{B}) = 1] - \Pr[\mathsf{Exp}^0(\mathcal{B}) = 1]| = |\Pr[\mathcal{H}_{i^*}(\mathcal{A}) = 1] - \Pr[\mathcal{H}_{i^*-1}(\mathcal{A}) = 1]| \geq \epsilon/n,$$

which concludes the proof. □

6.2.3 Sender Privacy

We will now proceed to show sender privacy of wSFE against malicious receivers.

Theorem 6.3. *Assuming that* iOT *has indistinguishability sender privacy and that* $(\mathsf{Garble}, \mathsf{Eval})$ *is a simulation secure garbling scheme, it holds that* wSFE *has sender privacy.*

Proof. Assume towards contradiction that there exists a PPT adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ with non-negligible advantage ϵ against the sender privacy of wSFE, i.e.

$$\mathsf{Adv}_{SP}(\mathcal{A}) = |\Pr[\mathsf{Exp}_{SP}(\mathcal{A}) = 1] - 1/2| = \epsilon.$$

We will henceforth only consider λ for which $\epsilon(\lambda) > 1/p(\lambda)$, that is we assume that $1/\epsilon = \mathsf{poly}(\lambda)$ without further mention. Assume that the circuits C_0, C_1 output by $\mathcal{A}$ have at most $n = n(\mathsf{crs}) = \mathsf{poly}(\lambda)$ input wires. In the following denote $\epsilon' = \frac{\epsilon}{8(n+1)}$.

Denote by $\mathsf{Exp}_{SP}(\mathcal{A}; \mathsf{crs}, \mathsf{r}_\mathcal{A})$ the sender privacy experiment.

$\mathsf{Exp}_{SP}(\mathcal{A})$:

- Choose uniformly random coins $\mathsf{r}_\mathcal{A}$ for $\mathcal{A}$
- Compute $\mathsf{crs} \stackrel{\$}{\leftarrow} \mathsf{iOT.Setup}(1^\lambda; \mathsf{crs})$.

- Compute $(C_0, C_1, \mathsf{z}_1) \xleftarrow{\$} \mathcal{A}_1(\mathsf{crs}; \mathsf{r}_\mathcal{A})$
- Choose $b \xleftarrow{\$} \{0,1\}$
- Compute $(\widehat{\mathsf{C}}, \vec{\mathsf{lb}}^0, \vec{\mathsf{lb}}^1) \xleftarrow{\$} \mathsf{Garble}(1^\lambda, C_b)$
- For $i = 1, \dots, n$
  - Compute $\mathsf{ots}_i \xleftarrow{\$} \mathsf{iOT.OT}_2(\mathsf{crs}, \mathsf{otr}_i, \mathsf{lb}_i^0, \mathsf{lb}_i^1)$.
- Compute $b' \xleftarrow{\$} \mathcal{A}_2(\mathsf{crs}, (\widehat{\mathsf{C}}, \vec{\mathsf{ots}}); \mathsf{r}_\mathcal{A})$
- If $b' = b$ output 1, otherwise 0

In the following, we will need to generate samples of random variables $\mathsf{Exp}_i(\mathcal{A}, \mathsf{crs}, \mathsf{r}_\mathcal{A}, \bar{x})$ which themselves depend on the adversary $\mathcal{A}$, a common reference string crs, random coins $\mathsf{r}_\mathcal{A}$ for $\mathcal{A}$ and an additional input $\bar{x} \in \{0,1\}^i$. $\mathsf{Exp}_i(\mathcal{A}, \mathsf{crs}, \mathsf{r}_\mathcal{A}, \bar{x})$ is sampled by the following algorithm.

$\mathsf{Exp}_i(\mathcal{A}, \mathsf{crs}, \mathsf{r}_\mathcal{A}, \bar{x})$:

- Compute $(C_0, C_1, \mathsf{z}_1) \xleftarrow{\$} \mathcal{A}_1(\mathsf{crs}; \mathsf{r}_\mathcal{A})$
- Choose $b \xleftarrow{\$} \{0,1\}$
- Compute $(\widehat{\mathsf{C}}, \vec{\mathsf{lb}}^0, \vec{\mathsf{lb}}^1) \xleftarrow{\$} \mathsf{Garble}(1^\lambda, C_b)$
- For $j = 1, \dots, i$
  - Set $\mathsf{lb}'^{\bar{x}_j}_j \xleftarrow{\$} \mathsf{lb}^{\bar{x}_j}_j$
  - Choose $\mathsf{lb}'^{1-\bar{x}_j}_j \xleftarrow{\$} \{0,1\}^\lambda$
  - Compute $\mathsf{ots}_j \xleftarrow{\$} \mathsf{iOT.OT}_2(\mathsf{crs}, \mathsf{otr}_i, \mathsf{lb}'^0_j, \mathsf{lb}'^1_j)$.
- For $j = i+1, \dots, n$
  - Compute $\mathsf{ots}_j \xleftarrow{\$} \mathsf{iOT.OT}_2(\mathsf{crs}, \mathsf{otr}_i, \mathsf{lb}^0_j, \mathsf{lb}^1_j)$.
- Compute $b' \xleftarrow{\$} \mathcal{A}_2(\mathsf{crs}, (\widehat{\mathsf{C}}, \vec{\mathsf{ots}}); \mathsf{r}_\mathcal{A})$
- If $b' = b$ output 1, otherwise 0

Input Extractor We will now construct an input extractor $\mathsf{Extract}$, which takes an index i, an adversary $\mathcal{A}$, a common reference string crs, random coins $\mathsf{r}_\mathcal{A}$ for $\mathcal{A}$ and additional random coins $\mathsf{r}_{\mathsf{Extract}}$ as inputs and outputs a string $\bar{x} \in \{0,1\}^i$ or $\perp$.

We will use the following notation. For an efficiently sampleable random variable $T \in \{0,1\}$ we will use the shorthand "Compute an approximation $\tilde{\mu}$ of $\mathsf{E}[T]$ with error δ" to denote the following algorithm which computes a sample average:

- Set $N = \lceil \lambda / \delta^2 \rceil$
- For $j = 1, \dots, N$ sample $t_j \xleftarrow{\$} T$
- Output $\tilde{\mu} \xleftarrow{\$} \frac{1}{N} \sum_{j=1}^N t_j$

Recall that $\epsilon' = \frac{\epsilon}{8(n+1)}$. The algorithm $\mathsf{Extract}$ is given as follows.

$\mathsf{Extract}_i(\mathcal{A}, \mathsf{crs}, \mathsf{r}_A, \mathsf{r}_{\mathsf{Extract}} = (\mathsf{r}_{\mathsf{Extract},1}, \ldots, \mathsf{r}_{\mathsf{Extract},i}))$:

- If $i > 0$ compute $\bar{x}' \xleftarrow{\$} \mathsf{Extract}_{i-1}(\mathcal{A}, \mathsf{crs}, \mathsf{r}_A, (\mathsf{r}_{\mathsf{Extract},1}, \ldots, \mathsf{r}_{\mathsf{Extract},i-1}))$, otherwise set $\bar{x}' \xleftarrow{\$} \emptyset$
- Parse $\bar{x}' = (\bar{x}_1, \ldots, \bar{x}_{i-1})$
- Use random tape $\mathsf{r}_{\mathsf{Extract},i}$ for the following 3 steps.
- Compute an approximation $\tilde{\mu}_i$ of $\mathsf{E}[\mathsf{Exp}_{i-1}(\mathcal{A}, \mathsf{crs}, \mathsf{r}_{\mathcal{A}}, (\bar{x}_1, \ldots, \bar{x}_{i-1}))]$ with error $\epsilon'/2$
- Compute an approximation $\tilde{\mu}_{i,0}$ of $\mathsf{E}[\mathsf{Exp}_i(\mathcal{A}, \mathsf{crs}, \mathsf{r}_{\mathcal{A}}, (\bar{x}_1, \ldots, \bar{x}_{i-1}, 0))]$ with error $\epsilon'/2$
- Compute an approximation $\tilde{\mu}_{i,1}$ of $\mathsf{E}[\mathsf{Exp}_i(\mathcal{A}, \mathsf{crs}, \mathsf{r}_{\mathcal{A}}, (\bar{x}_1, \ldots, \bar{x}_{i-1}, 1))]$ with error $\epsilon'/2$
- Set $\tilde{\delta}_{i,0} \xleftarrow{\$} |\tilde{\mu}_{i,0} - \tilde{\mu}_i|$
- Set $\tilde{\delta}_{i,1} \xleftarrow{\$} |\tilde{\mu}_{i,1} - \tilde{\mu}_i|$
- If $\tilde{\delta}_{i,0} > 2\epsilon'$ and $\tilde{\delta}_{i,1} > 2\epsilon$ abort and output $\perp$.
- else if $\tilde{\delta}_{i,1} > 2\epsilon'$ set $\bar{x}_i \xleftarrow{\$} 0$
- Otherwise set $\bar{x}_i \xleftarrow{\$} 1$
- Set $\bar{x} \xleftarrow{\$} (\bar{x}_1, \ldots, \bar{x}_i)$
- Output $\bar{x}$

Observe that since $\mathcal{A}$ is a PPT algorithm the Exp_i can be simulated efficiently. Thus, every iteration of $\mathsf{Extract}_i$ is efficent. As $\mathsf{Extract}_i$ runs for i iterations, we conclude that $\mathsf{Extract}_i$ is efficient.

Hybrids We will now define a sequence of adversary-dependent hybrid experiments.

- Hybrid $\mathcal{H}_0(\mathcal{A})$: This is the real experiment $\mathsf{Exp}_{SP}(\mathcal{A})$.

For $i = 1, \ldots, n$ define the following sequence of hybrids.

- $\mathcal{H}_i(\mathcal{A})$:
  - Choose uniformly random coins $\mathsf{r}_{\mathcal{A}}$ for $\mathcal{A}$
  - Compute $\mathsf{crs} \xleftarrow{\$} \mathsf{iOT.Setup}(1^\lambda)$
  - Compute $\bar{x} \xleftarrow{\$} \mathsf{Extract}_i(\mathcal{A}, \mathsf{crs}, \mathsf{r}_A, \mathsf{r}_{\mathsf{Extract}})$
  - Compute $(C_0, C_1, \mathsf{z}_1) \xleftarrow{\$} \mathcal{A}_1(\mathsf{crs}; \mathsf{r}_{\mathcal{A}})$
  - Choose $b \xleftarrow{\$} \{0, 1\}$
  - Compute $(\widehat{\mathsf{C}}, \vec{\mathsf{lb}}^0, \vec{\mathsf{lb}}^1) \xleftarrow{\$} \mathsf{Garble}(1^\lambda, C_b)$
  - For $j = 1, \ldots, i$
    - Set $\mathsf{lb}'^{\bar{x}_j}_j \xleftarrow{\$} \mathsf{lb}^{\bar{x}_j}_j$
    - Choose $\mathsf{lb}'^{1-\bar{x}_j}_j \xleftarrow{\$} \{0, 1\}^\lambda$
    - Compute $\mathsf{ots}_j \xleftarrow{\$} \mathsf{iOT.OT}_2(\mathsf{crs}, \mathsf{otr}_j, \mathsf{lb}'^0_j, \mathsf{lb}'^1_j)$.
  - For $j = i + 1, \ldots, n$

    * Compute $\mathsf{ots}_j \xleftarrow{\$} \mathsf{iOT.OT}_2(\mathsf{crs}, \mathsf{otr}_j, \mathsf{lb}_j^0, \mathsf{lb}_j^1)$.
  - Compute $b' \xleftarrow{\$} \mathcal{A}_2(\mathsf{crs}, (\widehat{\mathsf{C}}, \vec{\mathsf{ots}}); \mathsf{r}_{\mathcal{A}})$
  - If $b' = b$ output 1, otherwise 0

- Hybrid $\mathcal{H}_{n+1}(\mathcal{A})$: This is the same as hybrid $\mathcal{H}_n(\mathcal{A}; \mathsf{crs}, \mathsf{r}_{\mathcal{A}})$, except that the garbled circuit $\widehat{\mathsf{C}}$ and the labels $\vec{\mathsf{lb}} = (\mathsf{lb}^i_{x_i^*})$ are computed via $(\widehat{\mathsf{C}}, \vec{\mathsf{lb}}) \xleftarrow{\$} \mathsf{GCSim}(1^\lambda, C_b(x^*))$. That is

  $\mathcal{H}_{n+1}(\mathcal{A})$:
  - Choose uniformly random coins $\mathsf{r}_{\mathcal{A}}$ for $\mathcal{A}$
  - Compute $\mathsf{crs} \xleftarrow{\$} \mathsf{iOT.Setup}(1^\lambda)$
  - Compute $\bar{x} \xleftarrow{\$} \mathsf{Extract}_n(\mathcal{A}, \mathsf{crs}, \mathsf{r}_A, \mathsf{r}_{\mathsf{Extract}})$
  - Compute $(C_0, C_1, \mathsf{z}_1) \xleftarrow{\$} \mathcal{A}_1(\mathsf{crs}; \mathsf{r}_{\mathcal{A}})$
  - Choose $b \xleftarrow{\$} \{0,1\}$
  - $(\widehat{\mathsf{C}}, \vec{\mathsf{lb}}) \xleftarrow{\$} \mathsf{GCSim}(1^\lambda, C_b(\bar{x}))$
  - For $j = 1, \ldots, n$
    * Set $\mathsf{lb}'^{\bar{x}_j}_j \xleftarrow{\$} \mathsf{lb}_j$
    * Choose $\mathsf{lb}'^{1-\bar{x}_j}_j \xleftarrow{\$} \{0,1\}^\lambda$
    * Compute $\mathsf{ots}_j \xleftarrow{\$} \mathsf{iOT.OT}_2(\mathsf{crs}, \mathsf{otr}_j, \mathsf{lb}'^0_j, \mathsf{lb}'^1_j)$.
  - Compute $b' \xleftarrow{\$} \mathcal{A}_2(\mathsf{crs}, (\widehat{\mathsf{C}}, \vec{\mathsf{ots}}); \mathsf{r}_{\mathcal{A}})$
  - If $b' = b$ output 1, otherwise 0

Observe that since C_0 and C_1 are functionally equivalent it holds that $C_0(\bar{x}) = C_1(\bar{x})$. Consequently, in hybrid $\mathcal{H}_{n+1}$ the view of the adversary $\mathcal{A}$ is independent of the challenge bit b and we conclude that $\mathsf{Adv}_{\mathcal{H}_{n+1}}(\mathcal{A}) = 0$.

We claim there must exist an $i^* \in [n+1]$ such that

$$|\Pr[\mathcal{H}_{i^*}(\mathcal{A}) = 1] - \Pr[\mathcal{H}_{i^*-1}(\mathcal{A}) = 1]| \geq \epsilon/(n+1) = 8\epsilon'.$$

If this was not the case, we would get that

$$\begin{aligned}
\mathsf{Adv}_{SP}(\mathcal{A}) &= |\Pr[\mathcal{H}_0(\mathcal{A}) = 1] - 1/2| \\
&= |\sum_{i=1}^{n+1} (\Pr[\mathcal{H}_{i-1}(\mathcal{A}) = 1] - \Pr[\mathcal{H}_i(\mathcal{A}) = 1]) + \Pr[\mathcal{H}_{n+1}(\mathcal{A}) = 1] - 1/2| \\
&\leq \sum_{i=1}^{n+1} \underbrace{|\Pr[\mathcal{H}_i(\mathcal{A}) = 1] - \Pr[\mathcal{H}_{i-1}(\mathcal{A}) = 1]|}_{<\epsilon/(n+1) \text{ by assumption}} + \underbrace{|\Pr[\mathcal{H}_{n+1}(\mathcal{A}) = 1] - 1/2|}_{=0} \\
&< (n+1) \cdot \epsilon/(n+1) \\
&= \epsilon,
\end{aligned}$$

which contradicts $\mathsf{Adv}_{SP}(\mathcal{A}) = \epsilon$.

We will show in Lemma 6.5 that if $i^* \in \{1, \ldots, n\}$, then we get a contradiction against the indistinguishability sender privacy of iOT. On the other hand, we will show in Lemma 6.6 that $i^* = n+1$ will lead to a contradiction against the security of $(\mathsf{Garble}, \mathsf{Eval})$. □

We will first establish that the approximations $\tilde{\delta}_{i^*,0}$ and $\tilde{\delta}_{i^*,1}$ computed by $\mathsf{Extract}_{i^*}(\mathcal{A}, \mathsf{crs}, \mathsf{r}_A, \mathsf{r}_{\mathsf{Extract}})$ are close to the true advantages between Exp_{i^*} and Exp_{i^*-1}, except with negligible probability over the coins used to compute the approximations. We establish this by a routine application of the Hoeffding bound.

Lemma 6.4. *Assume that* $\mathsf{r}_{\mathsf{Extract}} = (\mathsf{r}_{\mathsf{Extract},1}, \ldots, \mathsf{r}_{\mathsf{Extract},i^*})$*. Now fix* crs*,* $\mathsf{r}_{\mathcal{A}}$ *and* $(\mathsf{r}_{\mathsf{Extract},1}, \ldots, \mathsf{r}_{\mathsf{Extract},i^*-1})$ *such that* $\mathsf{Extract}_{i^*-1}(\mathcal{A}, \mathsf{crs}, \mathsf{r}_{\mathcal{A}}, (\mathsf{r}_{\mathsf{Extract},1}, \ldots, \mathsf{r}_{\mathsf{Extract},i^*-1})) \neq \perp$*. Let*

$$\bar{x}' \xleftarrow{\$} \mathsf{Extract}_{i^*-1}(\mathcal{A}, \mathsf{crs}, \mathsf{r}_{\mathcal{A}}, (\mathsf{r}_{\mathsf{Extract},1}, \ldots, \mathsf{r}_{\mathsf{Extract},i^*-1})).$$

Then it holds that

$$|\tilde{\delta}_{i^*,0} - |\mathsf{E}[\mathsf{Exp}_{i^*}(\mathcal{A}, \mathsf{crs}, \mathsf{r}_{\mathcal{A}}, (\bar{x}', 0))] - \mathsf{E}[\mathsf{Exp}_{i^*-1}(\mathcal{A}, \mathsf{crs}, \mathsf{r}_{\mathcal{A}}, \bar{x}')]|| \leq \epsilon'$$
$$|\tilde{\delta}_{i^*,1} - |\mathsf{E}[\mathsf{Exp}_{i^*}(\mathcal{A}, \mathsf{crs}, \mathsf{r}_{\mathcal{A}}, (\bar{x}', 1))] - \mathsf{E}[\mathsf{Exp}_{i^*-1}(\mathcal{A}, \mathsf{crs}, \mathsf{r}_{\mathcal{A}}, \bar{x}')]|| \leq \epsilon'$$

except with probability $2^{-\lambda}$ *over the choice of* $\mathsf{r}_{\mathsf{Extract},i^*}$*.*

Proof. The random variable $\tilde{\mu}_{i^*}$ is the average of $N = \lceil \lambda/\epsilon'^2 \rceil = \lceil \frac{\lambda}{(\epsilon/(8(n+1)))^2} \rceil$ samples of $\mathsf{Exp}_{i^*-1}(\mathcal{A}, \mathsf{crs}, \mathsf{r}_{\mathcal{A}}, \bar{x}')$. Consequently, it holds by the Hoeffding inequality (Theorem 3.2) that

$$\Pr_{\mathsf{r}_{\mathsf{Extract},i}}[|\tilde{\mu}_{i^*} - \mathsf{E}[\mathsf{Exp}_{i^*-1}(\mathcal{A}, \mathsf{crs}, \mathsf{r}_{\mathcal{A}}, \bar{x}')]| > \epsilon'/2] \leq 2e^{-2N(\epsilon'/2)^2} \leq 2e^{-\lambda}$$

Analogously, we obtain that

$$\Pr_{\mathsf{r}_{\mathsf{Extract},i^*}}[|\tilde{\mu}_{i^*,0} - \mathsf{E}[\mathsf{Exp}_{i^*}(\mathcal{A}, \mathsf{crs}, \mathsf{r}_{\mathcal{A}}, (\bar{x}', 0))]| > \epsilon'/2] \leq 2e^{-2N(\epsilon'/2)^2} \leq 2e^{-\lambda}$$

and

$$\Pr_{\mathsf{r}_{\mathsf{Extract},i^*}}[|\tilde{\mu}_{i^*,1} - \mathsf{E}[\mathsf{Exp}_{i^*}(\mathcal{A}, \mathsf{crs}, \mathsf{r}_{\mathcal{A}}, (\bar{x}', 1))]| > \epsilon'/2] \leq 2e^{-2N(\epsilon'/2)^2} \leq 2e^{-\lambda}.$$

Given that

$$|\tilde{\mu}_{i^*} - \mathsf{E}[\mathsf{Exp}_{i^*-1}(\mathcal{A}, \mathsf{crs}, \mathsf{r}_{\mathcal{A}}, \bar{x}')]| \leq \epsilon'/2$$
$$|\tilde{\mu}_{i^*,0} - \mathsf{E}[\mathsf{Exp}_{i^*}(\mathcal{A}, \mathsf{crs}, \mathsf{r}_{\mathcal{A}}, (\bar{x}', 0))]| \leq \epsilon'/2$$
$$|\tilde{\mu}_{i^*,1} - \mathsf{E}[\mathsf{Exp}_{i^*}(\mathcal{A}, \mathsf{crs}, \mathsf{r}_{\mathcal{A}}, (\bar{x}', 1))]| \leq \epsilon'/2$$

and using that

$$\tilde{\delta}_{i^*,0} = |\tilde{\mu}_{i^*,0} - \tilde{\mu}_{i^*}|$$
$$\tilde{\delta}_{i^*,1} = |\tilde{\mu}_{i^*,1} - \tilde{\mu}_{i^*}|$$

we get that

$$|\tilde{\delta}_{i^*,0} - |\mathsf{E}[\mathsf{Exp}_{i^*}(\mathcal{A}, \mathsf{crs}, \mathsf{r}_{\mathcal{A}}, (\bar{x}', 0))] - \mathsf{E}[\mathsf{Exp}_{i^*-1}(\mathcal{A}, \mathsf{crs}, \mathsf{r}_{\mathcal{A}}, \bar{x}')]|| \leq \epsilon'$$

and

$$|\tilde{\delta}_{i^*,1} - |\mathsf{E}[\mathsf{Exp}_{i^*}(\mathcal{A}, \mathsf{crs}, \mathsf{r}_{\mathcal{A}}, (\bar{x}', 1))] - \mathsf{E}[\mathsf{Exp}_{i^*-1}(\mathcal{A}, \mathsf{crs}, \mathsf{r}_{\mathcal{A}}, \bar{x}')]|| \leq \epsilon'.$$

Consequently, it holds by a union-bound that

$$\Pr_{\mathsf{r}_{\mathsf{Extract},i^*}}\left[\begin{array}{ll} & |\tilde{\delta}_{i^*,0} - |\mathsf{E}[\mathsf{Exp}_{i^*}(\mathcal{A}, \mathsf{crs}, \mathsf{r}_{\mathcal{A}}, (\bar{x}', 0))] - \mathsf{E}[\mathsf{Exp}_{i^*-1}(\mathcal{A}, \mathsf{crs}, \mathsf{r}_{\mathcal{A}}, \bar{x}')]|| > \epsilon' \\ \text{or} & |\tilde{\delta}_{i^*,1} - |\mathsf{E}[\mathsf{Exp}_{i^*}(\mathcal{A}, \mathsf{crs}, \mathsf{r}_{\mathcal{A}}, (\bar{x}', 1))] - \mathsf{E}[\mathsf{Exp}_{i^*-1}(\mathcal{A}, \mathsf{crs}, \mathsf{r}_{\mathcal{A}}, \bar{x}')]|| > \epsilon' \end{array}\right] \leq 6 \cdot e^{-\lambda} \leq 2^{-\lambda}$$

which concludes the proof. $\square$

Lemma 6.5. *Assume that* $|\Pr[\mathcal{H}_{i^*}(\mathcal{A}) = 1] - \Pr[\mathcal{H}_{i^*-1}(\mathcal{A}) = 1]| \geq 8 \cdot \epsilon'$ *for an* $i^* \in [n]$*. Then there exists a PPT adversary* $\mathcal{B}$ *which breaks the indistinguishability sender security of* iOT*.*

Proof. We will first slightly reformulate $\mathcal{H}_{i^*}$, leaving the actual experiment unchanged. First, instead of computing crs and sampling $\mathsf{r}_{\mathcal{A}}$ and $\mathsf{r}_{\mathsf{Extract}}$ itself, it takes these values as explicit inputs. Second and more importantly, once we have extracted $\bar{x}$, what is computed in the remaining steps is identical to $\mathsf{Exp}_{i^*}(\mathcal{A}, \mathsf{crs}, \mathsf{r}_{\mathcal{A}}, \bar{x})$. Consequently, we can rewrite $\mathcal{H}_{i^*}$ as follows, where we assume that $\mathsf{crs} \xleftarrow{\$} \mathsf{iOT.Setup}(1^\lambda)$ and $\mathsf{r}_{\mathcal{A}}$ and $\mathsf{r}_{\mathsf{Extract}}$ are uniformly random coins.

$\mathcal{H}_{i^*}(\mathcal{A}, \mathsf{crs}, \mathsf{r}_{\mathcal{A}}, \mathsf{r}_{\mathsf{Extract}})$:

- Compute $\bar{x} \xleftarrow{\$} \mathsf{Extract}_{i^*}(\mathcal{A}, \mathsf{crs}, \mathsf{r}_A, \mathsf{r}_{\mathsf{Extract}})$
- Compute and output $\mathsf{Exp}_{i^*}(\mathcal{A}, \mathsf{crs}, \mathsf{r}_{\mathcal{A}}, \bar{x})$

To make things more readable in the following, we will bundle crs, $\mathsf{r}_{\mathcal{A}}$ and $\mathsf{r}_{\mathsf{Extract}}$ in a variable aux. That is, we will set $\mathsf{aux} = (\mathsf{crs}, \mathsf{r}_{\mathcal{A}}, \mathsf{r}_{\mathsf{Extract}})$. Furthermore, we will assume that the output $\bar{x} \in \{0,1\}^{i^*}$ of $\mathsf{Extract}_{i^*}(\mathcal{A}, \mathsf{crs}, \mathsf{r}_A, \mathsf{r}_{\mathsf{Extract}})$ is of the form $\bar{x} = (\bar{x}', \bar{x}_{i^*})$, where $\bar{x}' \in \{0,1\}^{i^*-1}$ and $\bar{x}_{i^*} \in \{0,1\}$.

We will now define three events $\mathsf{GAP}(\mathsf{aux})$, $\mathsf{APPROX}(\mathsf{aux})$ and $\mathsf{GOOD}(\mathsf{aux})$ which only depend on aux.

- $\mathsf{GAP}(\mathsf{aux})$ holds, if and only if

$$|\Pr[\mathcal{H}_{i^*}(\mathcal{A}; \mathsf{aux}) = 1] - \Pr[\mathcal{H}_{i^*-1}(\mathcal{A}; \mathsf{aux}) = 1]| > 4\epsilon'.$$

- Let $\tilde{\delta}_{i^*,0}$ and $\tilde{\delta}_{i^*,1}$ be the values computed during the execution of $\mathsf{Extract}_{i^*}(\mathcal{A}, \mathsf{crs}, \mathsf{r}_A, \mathsf{r}_{\mathsf{Extract}})$. $\mathsf{APPROX}(\mathsf{aux})$ holds, if and only if

$$|\tilde{\delta}_{i^*,0} - |\mathsf{E}[\mathsf{Exp}_{i^*}(\mathcal{A}, \mathsf{crs}, \mathsf{r}_{\mathcal{A}}, (\bar{x}', 0))] - \mathsf{E}[\mathsf{Exp}_{i^*-1}(\mathcal{A}, \mathsf{crs}, \mathsf{r}_{\mathcal{A}}, \bar{x}')]|| \leq \epsilon'$$
$$|\tilde{\delta}_{i^*,1} - |\mathsf{E}[\mathsf{Exp}_{i^*}(\mathcal{A}, \mathsf{crs}, \mathsf{r}_{\mathcal{A}}, (\bar{x}', 1))] - \mathsf{E}[\mathsf{Exp}_{i^*-1}(\mathcal{A}, \mathsf{crs}, \mathsf{r}_{\mathcal{A}}, \bar{x}')]|| \leq \epsilon'$$

- $\mathsf{GOOD}(\mathsf{aux})$ holds if and only if

$$|\mathsf{E}[\mathsf{Exp}_{i^*}(\mathcal{A}, \mathsf{crs}, \mathsf{r}_{\mathcal{A}}, (\bar{x}', 0))] - \mathsf{E}[\mathsf{Exp}_{i^*-1}(\mathcal{A}, \mathsf{crs}, \mathsf{r}_{\mathcal{A}}, \bar{x}')]| > \epsilon'$$
$$|\mathsf{E}[\mathsf{Exp}_{i^*}(\mathcal{A}, \mathsf{crs}, \mathsf{r}_{\mathcal{A}}, (\bar{x}', 1))] - \mathsf{E}[\mathsf{Exp}_{i^*-1}(\mathcal{A}, \mathsf{crs}, \mathsf{r}_{\mathcal{A}}, \bar{x}')]| > \epsilon'.$$

We will first elaborate on the events in more detail. The event $\mathsf{GAP}(\mathsf{aux})$ characterizes that for *the same* choice of aux, the hybrids $\mathcal{H}_{i^*}(\mathcal{A}, \mathsf{aux})$ and $\mathcal{H}_{i^*-1}(\mathcal{A}, \mathsf{aux})$ have distance at least $4\epsilon'$. Notice that the extracted prefix $(\bar{x}_1, \ldots, \bar{x}_{i^*-1})$ is identical in both experiments $\mathcal{H}_{i^*}(\mathcal{A}, \mathsf{aux})$ and $\mathcal{H}_{i^*-1}(\mathcal{A}, \mathsf{aux})$. Consequently, $\mathsf{GAP}(\mathsf{aux})$ immediately implies that $\mathsf{Extract}_{i^*-1}(\mathcal{A}, \mathsf{crs}, \mathsf{r}_A, \mathsf{r}_{\mathsf{Extract}})$ does not output $\perp$, as this would imply that the two experiments are identically distributed.

The event $\mathsf{APPROX}(\mathsf{aux})$ ensures that the approximations $\tilde{\delta}_{i^*,0}$ and $\tilde{\delta}_{i^*,1}$ are sufficiently close to the true advantages.

Finally, the event $\mathsf{GOOD}(\mathsf{aux})$ ensures that aux is such that we will be able to mount a successful attack against indistinguishability sender security of iOT. Our first goal will be to show that the event $\mathsf{GOOD}(\mathsf{aux})$ holds with reasonably high probability over the choice of aux. Once this is established, we will construct an adversary $\mathcal{B}$ against the indistinguishability sender security of iOT.

Observe that by Lemma 6.4 it holds that

$$\Pr_{\mathsf{aux}}[\neg\mathsf{APPROX}(\mathsf{aux})] \leq 2^{-\lambda}. \tag{5}$$

As

$$|\Pr_{\mathsf{aux}}[\mathcal{H}_{i^*}(\mathcal{A}, \mathsf{aux}) = 1] - \Pr_{\mathsf{aux}}[\mathcal{H}_{i^*-1}(\mathcal{A}, \mathsf{aux}) = 1]| = |\Pr[\mathcal{H}_{i^*}(\mathcal{A}) = 1] - \Pr[\mathcal{H}_{i^*-1}(\mathcal{A}) = 1]| \geq 8 \cdot \epsilon'$$

it holds by the Markov inequality for advantages (Lemma 3.1) that

$$\Pr_{\mathsf{aux}}[\mathsf{GAP}(\mathsf{aux})] = \Pr_{\mathsf{aux}}[|\Pr[\mathcal{H}_{i^*}(\mathcal{A}; \mathsf{aux}) = 1] - \Pr[\mathcal{H}_{i^*-1}(\mathcal{A}; \mathsf{aux}) = 1]| > 4\epsilon'] \geq 4\epsilon'. \tag{6}$$

We will now show that if $\mathsf{GAP}(\mathsf{aux})$ holds, then it must either hold $\mathsf{GOOD}(\mathsf{aux})$ or not $\mathsf{APPROX}(\mathsf{aux})$. We will establish this by showing that $\neg\mathsf{GOOD}(\mathsf{aux})$ and $\mathsf{APPROX}(\mathsf{aux})$ imply $\neg\mathsf{GAP}(\mathsf{aux})$. Thus, fix $\mathsf{aux} = (\mathsf{crs}, \mathsf{r}_{\mathcal{A}}, \mathsf{r}_{\mathsf{Extract}})$ with $\neg\mathsf{GOOD}(\mathsf{aux})$ and $\mathsf{APPROX}(\mathsf{aux})$.

From $\neg\mathsf{GOOD}(\mathsf{aux})$ it follows that there is a $\beta \in \{0, 1\}$ such that

$$|\mathsf{E}[\mathsf{Exp}_{i^*}(\mathcal{A}, \mathsf{crs}, \mathsf{r}_{\mathcal{A}}, (\bar{x}', \beta))] - \mathsf{E}[\mathsf{Exp}_{i^*-1}(\mathcal{A}, \mathsf{crs}, \mathsf{r}_{\mathcal{A}}, \bar{x}')]| \leq \epsilon'.$$

We will now show that $\mathsf{Extract}_{i^*}(\mathcal{A}, \mathsf{crs}, \mathsf{r}_A, \mathsf{r}_{\mathsf{Extract}})$ will be able to identify the correct $\bar{x}_{i^*}$. Observe that since it holds that $\mathsf{APPROX}(\mathsf{aux})$, we get that

$$\tilde{\delta}_{i^*,\beta} \leq |\mathsf{E}[\mathsf{Exp}_{i^*}(\mathcal{A}, \mathsf{crs}, \mathsf{r}_{\mathcal{A}}, (\bar{x}', \beta))] - \mathsf{E}[\mathsf{Exp}_{i^*-1}(\mathcal{A}, \mathsf{crs}, \mathsf{r}_{\mathcal{A}}, \bar{x}')]| + \epsilon' \leq 2\epsilon'.$$

Consequently, $\mathsf{Extract}_{i^*}(\mathcal{A}, \mathsf{crs}, \mathsf{r}_A, \mathsf{r}_{\mathsf{Extract}})$ will not output $\perp$. We will distinguish two cases.

Case 1 : In this case it holds that

$$|\mathsf{E}[\mathsf{Exp}_{i^*}(\mathcal{A}, \mathsf{crs}, \mathsf{r}_{\mathcal{A}}, (\bar{x}', 1-\beta))] - \mathsf{E}[\mathsf{Exp}_{i^*-1}(\mathcal{A}, \mathsf{crs}, \mathsf{r}_{\mathcal{A}}, \bar{x}')]| \leq 4\epsilon'.$$

It follows immediately that

$$|\mathsf{E}[\mathsf{Exp}_{i^*}(\mathcal{A}, \mathsf{crs}, \mathsf{r}_{\mathcal{A}}, (\bar{x}', \bar{x}_{i^*}))] - \mathsf{E}[\mathsf{Exp}_{i^*-1}(\mathcal{A}, \mathsf{crs}, \mathsf{r}_{\mathcal{A}}, \bar{x}')]| \leq 4\epsilon',$$

regardless which $\bar{x}_{i^*} \in \{0, 1\}$ is chosen.

Case 2 : In this case it holds that

$$|\mathsf{E}[\mathsf{Exp}_{i^*}(\mathcal{A}, \mathsf{crs}, \mathsf{r}_{\mathcal{A}}, (\bar{x}', 1-\beta))] - \mathsf{E}[\mathsf{Exp}_{i^*-1}(\mathcal{A}, \mathsf{crs}, \mathsf{r}_{\mathcal{A}}, \bar{x}')]| > 4\epsilon'.$$

Again since it holds that $\mathsf{APPROX}(\mathsf{aux})$, we get that

$$\tilde{\delta}_{i^*,1-\beta} \geq |\mathsf{E}[\mathsf{Exp}_{i^*}(\mathcal{A}, \mathsf{crs}, \mathsf{r}_{\mathcal{A}}, (\bar{x}', 1-\beta))] - \mathsf{E}[\mathsf{Exp}_{i^*-1}(\mathcal{A}, \mathsf{crs}, \mathsf{r}_{\mathcal{A}}, \bar{x}')]| - \epsilon' \geq 3\epsilon' > 2\epsilon'.$$

Consequently, $\mathsf{Extract}_{i^*}(\mathcal{A}, \mathsf{crs}, \mathsf{r}_A, \mathsf{r}_{\mathsf{Extract}})$ will set $\bar{x}_{i^*} \xleftarrow{\$} \beta$ and again we can conclude

$$|\mathsf{E}[\mathsf{Exp}_{i^*}(\mathcal{A}, \mathsf{crs}, \mathsf{r}_{\mathcal{A}}, (\bar{x}', \bar{x}_{i^*}))] - \mathsf{E}[\mathsf{Exp}_{i^*-1}(\mathcal{A}, \mathsf{crs}, \mathsf{r}_{\mathcal{A}}, \bar{x}')]| \leq 4\epsilon',$$

Observe that we can write

$$\begin{aligned}\mathsf{E}[\mathsf{Exp}_{i^*-1}(\mathcal{A},\mathsf{crs},\mathsf{r}_{\mathcal{A}},\bar{x}')] &= \Pr[\mathsf{Exp}_{i^*-1}(\mathcal{A},\mathsf{crs},\mathsf{r}_{\mathcal{A}},\bar{x}') = 1]\\ \mathsf{E}[\mathsf{Exp}_{i^*}(\mathcal{A},\mathsf{crs},\mathsf{r}_{\mathcal{A}},(\bar{x}',0))] &= \Pr[\mathsf{Exp}_{i^*}(\mathcal{A},\mathsf{crs},\mathsf{r}_{\mathcal{A}},(\bar{x}',0)) = 1]\\ \mathsf{E}[\mathsf{Exp}_{i^*}(\mathcal{A},\mathsf{crs},\mathsf{r}_{\mathcal{A}},(\bar{x}',1))] &= \Pr[\mathsf{Exp}_{i^*}(\mathcal{A},\mathsf{crs},\mathsf{r}_{\mathcal{A}},(\bar{x}',1)) = 1].\end{aligned}$$

Further observe that since $\mathsf{Extract}_{i^*}(\mathcal{A},\mathsf{crs},\mathsf{r}_A,\mathsf{r}_{\mathsf{Extract}})$ will not output $\perp$, the output of $\mathcal{H}_{i^*}(\mathcal{A};\mathsf{aux})$ is distributed according to $\mathsf{Exp}_{i^*}(\mathcal{A},\mathsf{crs},\mathsf{r}_{\mathcal{A}},(\bar{x}',\bar{x}_{i^*}))$. We also know that $\mathcal{H}_{i^*-1}(\mathcal{A};\mathsf{aux})$ is distributed according to $\mathsf{Exp}_{i^*-1}(\mathcal{A},\mathsf{crs},\mathsf{r}_{\mathcal{A}},\bar{x}')$. This implies that

$$\begin{aligned}&|\Pr[\mathcal{H}_{i^*}(\mathcal{A};\mathsf{aux}) = 1] - \Pr[\mathcal{H}_{i^*-1}(\mathcal{A};\mathsf{aux}) = 1]| =\\ &\qquad |\mathsf{E}[\mathsf{Exp}_{i^*}(\mathcal{A},\mathsf{crs},\mathsf{r}_{\mathcal{A}},(\bar{x}',\bar{x}_{i^*}))] - \mathsf{E}[\mathsf{Exp}_{i^*-1}(\mathcal{A},\mathsf{crs},\mathsf{r}_{\mathcal{A}},\bar{x}')]| \le 4\epsilon',\end{aligned} \tag{7}$$

which in turn implies that $\neg\mathsf{GAP}(\mathsf{aux})$.

Thus, we have established that

$$\mathsf{GAP}(\mathsf{aux}) \Rightarrow \mathsf{GOOD}(\mathsf{aux}) \text{ or } \neg\mathsf{APPROX}(\mathsf{aux}). \tag{8}$$

From (6), (8) and (5) we obtain that

$$\begin{aligned}4\epsilon' &\le \Pr[\mathsf{GAP}(\mathsf{aux})]\\ &\le \Pr[\mathsf{GOOD}(\mathsf{aux}) \text{ or } \neg\mathsf{APPROX}(\mathsf{aux})]\\ &\le \Pr[\mathsf{GOOD}(\mathsf{aux})] + \Pr[\neg\mathsf{APPROX}(\mathsf{aux})]\\ &\le \Pr[(\mathsf{GOOD}(\mathsf{aux})] + 2^{-\lambda},\end{aligned}$$

where the third inequality follows by the union-bound. This implies that

$$\Pr_{\mathsf{aux}}[\mathsf{GOOD}(\mathsf{aux})] \ge 4\epsilon' - 2^{-\lambda} > \epsilon'.$$

We are now ready to construct an adversary $\mathcal{B}$ against the sender privacy of iOT. The adversary $\mathcal{B} = (\mathcal{B}_1,\mathcal{B}_2)$ is given as follows. In abuse of notation, we assume that $\mathcal{B}$ is stateful, i.e. the second stage $\mathcal{B}_2$ remembers all variables of the first stage $\mathcal{B}_1$.

$\mathcal{B}_1(\mathsf{crs};\mathsf{r}_{\mathcal{B}} = (\mathsf{r}_{\mathcal{A}},\mathsf{r}_{\mathsf{Extract}}))$:

- Compute $\bar{x} \xleftarrow{\$} \mathsf{Extract}_{i^*}(\mathcal{A},\mathsf{crs},\mathsf{r}_A,\mathsf{r}_{\mathsf{Extract}})$
- Compute $(C_0, C_1, \mathsf{z}_1) \xleftarrow{\$} \mathcal{A}_1(\mathsf{crs};\mathsf{r}_{\mathcal{A}})$
- Choose $b \xleftarrow{\$} \{0,1\}$
- Compute $(\widehat{\mathsf{C}}, \vec{\mathsf{lb}}^0, \vec{\mathsf{lb}}^1) \xleftarrow{\$} \mathsf{Garble}(1^\lambda, C_b)$
- For $j = 1,\ldots,i^*-1$
  - Set $\mathsf{lb}'^{\bar{x}_j}_j \xleftarrow{\$} \mathsf{lb}^{\bar{x}_j}_j$
  - Choose $\mathsf{lb}'^{1-\bar{x}_j}_j \xleftarrow{\$} \{0,1\}^\lambda$
  - Compute $\mathsf{ots}_j \xleftarrow{\$} \mathsf{iOT.OT}_2(\mathsf{crs},\mathsf{otr}_j,\mathsf{lb}'^0_j,\mathsf{lb}'^1_j)$.

- Output $(\mathsf{lb}^0_{i^*}, \mathsf{lb}^1_{i^*}, \mathsf{otr}_{i^*})$

$\mathcal{B}_2(\mathsf{crs}, \mathsf{r}_\mathcal{B}, \mathsf{ots}^*)$:

- Set $\mathsf{ots}_{i^*} \xleftarrow{\$} \mathsf{ots}^*$
- For $j = i^* + 1, \ldots, n$
  - Compute $\mathsf{ots}_j \xleftarrow{\$} \mathsf{iOT.OT}_2(\mathsf{crs}, \mathsf{otr}_i, \mathsf{lb}^0_j, \mathsf{lb}^1_j)$.
- Compute $b' \xleftarrow{\$} \mathcal{A}_2(\mathsf{crs}, (\widehat{\mathsf{C}}, \vec{\mathsf{ots}}); \mathsf{r}_\mathcal{A})$
- If $b' = b$ output 1, otherwise 0

Now fix crs and $\mathsf{r}_\mathcal{B}$. We will distinguish 3 cases.

1. In the first case, the challenge message ots^* is computed via $\mathsf{ots}^* \xleftarrow{\$} \mathsf{iOT.OT}_2(\mathsf{crs}, \mathsf{otr}^*_i, \mathsf{lb}^0_{i^*}, \mathsf{lb}^1_{i^*})$. It follows by inspection that in this case the output of $\mathcal{B}$ is distributed according to $\mathsf{Exp}_{i^*-1}(\mathcal{A}, \mathsf{crs}, \mathsf{r}_\mathcal{A}, \bar{x}')$.

2. In the second case, the challenge message ots^* is computed via $\mathsf{ots}^* \xleftarrow{\$} \mathsf{iOT.OT}_2(\mathsf{crs}, \mathsf{otr}^*_i, \mathsf{lb}^0_{i^*}, \tilde{\mathsf{lb}})$ for a uniformly random $\tilde{\mathsf{lb}} \xleftarrow{\$} \{0,1\}^\lambda$. It follows by inspection that in this case the output of $\mathcal{B}$ is distributed according to $\mathsf{Exp}_{i^*}(\mathcal{A}, \mathsf{crs}, \mathsf{r}_\mathcal{A}, (\bar{x}', 0))$.

3. In the third case, the challenge message ots^* is computed via $\mathsf{ots}^* \xleftarrow{\$} \mathsf{iOT.OT}_2(\mathsf{crs}, \mathsf{otr}^*_i, \tilde{\mathsf{lb}}, \mathsf{lb}^1_{i^*})$ for a uniformly random $\tilde{\mathsf{lb}} \xleftarrow{\$} \{0,1\}^\lambda$. It follows by inspection that in this case the output of $\mathcal{B}$ is distributed according to $\mathsf{Exp}_{i^*}(\mathcal{A}, \mathsf{crs}, \mathsf{r}_\mathcal{A}, (\bar{x}', 1))$.

We conclude that

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{crs},\mathsf{r}_\mathcal{B},0}_{\mathsf{iOT}}(\mathcal{B}) &= |\Pr[\mathsf{Exp}_{i^*}(\mathcal{A}, \mathsf{crs}, \mathsf{r}_\mathcal{A}, (\bar{x}, 0)) = 1] - \Pr[\mathsf{Exp}_{i^*-1}(\mathcal{A}, \mathsf{crs}, \mathsf{r}_\mathcal{A}, \bar{x}) = 1]| \\
\mathsf{Adv}^{\mathsf{crs},\mathsf{r}_\mathcal{B},1}_{\mathsf{iOT}}(\mathcal{B}) &= |\Pr[\mathsf{Exp}_{i^*}(\mathcal{A}, \mathsf{crs}, \mathsf{r}_\mathcal{A}, (\bar{x}, 1)) = 1] - \Pr[\mathsf{Exp}_{i^*-1}(\mathcal{A}, \mathsf{crs}, \mathsf{r}_\mathcal{A}, \bar{x}) = 1]|.
\end{aligned}$$

This implies that

$$\Pr_{\mathsf{crs},\mathsf{r}_\mathcal{B}}\left[\begin{array}{ll} & \mathsf{Adv}^{\mathsf{crs},\mathsf{r}_\mathcal{B},0}_{\mathsf{iOT}}(\mathcal{B}; \mathsf{crs}, \mathsf{r}_\mathcal{B}) > \epsilon' \\ \text{and} & \mathsf{Adv}^{\mathsf{crs},\mathsf{r}_\mathcal{B},1}_{\mathsf{iOT}}(\mathcal{B}; \mathsf{crs}, \mathsf{r}_\mathcal{B}) > \epsilon' \end{array}\right] = \Pr_{\mathsf{crs},\mathsf{r}_\mathcal{A},\mathsf{r}_{\mathsf{Extract}}}[\mathsf{GOOD}(\mathsf{crs}, \mathsf{r}_\mathcal{A}, \mathsf{r}_{\mathsf{Extract}})] > \epsilon',$$

which contradicts the sender privacy of iOT. □

Lemma 6.6. *Assume that* $|\Pr[\mathcal{H}_{n+1}(\mathcal{A}) = 1] - \Pr[\mathcal{H}_n(\mathcal{A}) = 1]| \geq 8\epsilon'$. *Then there exists a PPT adversary* $\mathcal{B}$ *with advantage* $8\epsilon'$ *against the security of* $(\mathsf{Garble}, \mathsf{Eval})$.

Proof. Consider the following reduction $\mathcal{B}^\mathcal{A}(1^\lambda)$ which is an adversary against the security of the garbling scheme $(\mathsf{Garble}, \mathsf{Eval})$.

$\mathcal{B}(1^\lambda)$:

- Simulate $\mathcal{H}_{n+1}$ faithfully until the challenge bit $b \xleftarrow{\$} \{0,1\}$ is chosen.
- Send C_b and $\bar{x}$ to the garbling experiment. Let $(\widehat{\mathsf{C}}, \vec{\mathsf{lb}})$ be the output of the garbling experiment.

- Continue the simulation of $\mathcal{H}_{n+1}$ faithfully using $(\widehat{\mathsf{C}}, \vec{\mathsf{lb}})$ and output whatever the simulated $\mathcal{H}_{n+1}$ outputs.

First consider the case that the garbling experiment generates $(\widehat{\mathsf{C}}, \vec{\mathsf{lb}})$ by $(\widehat{\mathsf{C}}, \vec{\mathsf{lb}}^0, \vec{\mathsf{lb}}^1) \xleftarrow{\$} \mathsf{Garble}(1^\lambda, \mathsf{C}_b)$ and $\mathsf{lb}_i \xleftarrow{\$} \mathsf{lb}_i^{\bar{x}_i}$ for all $i \in [n]$. In this case $\mathcal{B}$ faithfully simulates $\mathcal{H}_n$ and consequently the output of $\mathcal{B}$ is distributed identically to $\mathcal{H}_n$.

On the other hand, if the garbling experiment generates $(\widehat{\mathsf{C}}, \vec{\mathsf{lb}})$ by $(\widehat{\mathsf{C}}, \vec{\mathsf{lb}}) \xleftarrow{\$} \mathsf{GCSim}(1^\lambda, \mathsf{C}_b(\bar{x}))$, then $\mathcal{B}$ faithfully simulates $\mathcal{H}_{n+1}$ and consequently the output of $\mathcal{B}$ is distributed identically to $\mathcal{H}_{n+1}$.

We conclude that

$$\mathsf{Adv}(\mathcal{B}) = |\Pr[\mathcal{H}_n(\mathcal{A}) = 1] - \Pr[\mathcal{H}_{n+1}(\mathcal{A}) = 1]| \geq 8\epsilon',$$

which contradicts the security of $(\mathsf{Garble}, \mathsf{Eval})$. $\square$

7 Sender-UC OT from wSFE

In this section we will provide a two-round OT protocol with sender's UC security and receiver's indistinguishability security from any CPA-secure PKE and a two-round wSFE for a specific class of functions.

Let $\mathsf{PKE} := (\mathsf{KeyGen}, \mathsf{E}, \mathsf{Dec})$ be a PKE scheme and let $\mathsf{wSFE} := (\mathsf{Setup}, \mathsf{Receiver}_1, \mathsf{Sender}, \mathsf{Receiver}_2)$ be a two-round wSFE for a function class $\mathcal{F}$ defined as follows: any function in this class is of the form $\mathsf{C}[\mathsf{pk}, \mathsf{ct}, \mathsf{m}_0, \mathsf{m}_1]$, parameterized over a public key pk, a ciphertext ct and two messages m_0 and m_1, and is defined as follows:

$\mathsf{C}[\mathsf{pk}, \mathsf{ct}, \mathsf{m}_0, \mathsf{m}_1](b, \mathsf{r})$: If $\mathsf{PKE.E}(\mathsf{pk}, b; \mathsf{r}) = \mathsf{ct}$, output m_b; otherwise $\perp$.

Construction 7.1 (Sender-UC OT)**.** *The OT-protocol is based on the above two primitives* PKE *and* wSFE*, and is described as follows.*

$\mathsf{Setup}(1^\lambda)$**:** *Compute* $\mathsf{crs}' \xleftarrow{\$} \mathsf{wSFE.Setup}(1^\lambda)$ *and* $(\mathsf{pk}, \mathsf{sk}) \xleftarrow{\$} \mathsf{PKE.KeyGen}(1^\lambda)$*. Output* $\mathsf{crs} := (\mathsf{crs}', \mathsf{pk})$*.*

$\mathsf{OT}_1(\mathsf{crs} = (\mathsf{crs}', \mathsf{pk}), b)$**:** *Choose* $\mathsf{r} \xleftarrow{\$} \{0,1\}^\lambda$ *and compute* $\mathsf{ct} \xleftarrow{\$} \mathsf{PKE.E}(\mathsf{pk}, b; \mathsf{r})$*. Set* $\vec{x} := (b, r)$ *and compute* $(\mathsf{z}_1, \mathsf{st}) \xleftarrow{\$} \mathsf{wSFE.Receiver}_1(\mathsf{crs}', \vec{x})$*. Output* $\mathsf{otr} := (\mathsf{ct}, \mathsf{z}_1)$ *as the OT message and* st *as the private state.*

$\mathsf{OT}_2(\mathsf{crs} = (\mathsf{crs}', \mathsf{pk}), \mathsf{otr} = (\mathsf{ct}, \mathsf{z}_1), \mathsf{m}_0, \mathsf{m}_1)$**:** *Compute* $\mathsf{z}_2 \xleftarrow{\$} \mathsf{wSFE.Sender}(\mathsf{crs}', \mathsf{C}[\mathsf{pk}, \mathsf{ct}, \mathsf{m}_0, \mathsf{m}_1], \mathsf{z}_1)$*. Output* $\mathsf{ots} := \mathsf{z}_2$*.*

$\mathsf{OT}_3(\mathsf{st}, \mathsf{ots})$**:** *Let* $\mathsf{z}_2 := \mathsf{ots}$*. Compute and output* $\mathsf{Receiver}_2(\mathsf{st}, \mathsf{z}_2)$*.*

Theorem 7.2. *Assuming* PKE *is CPA-secure and perfectly correct (Definition 3.3), and that* wSFE *satisfies correctness, receiver privacy and sender privacy (Definition 6.1), then the OT given in Construction 7.1 provides receiver's indistinguishability security and sender's UC security.*

We now give the proof for each part of the theorem.

7.1 Correctness

The correctness of the OT protocol follows immediately from the perfect correctness of the underlying PKE scheme (Definition 3.3) and the correctness of the wSFE scheme (Definition 6.1). $\square$

7.2 Receiver's Indistinguishability Security

We will prove receiver's indistinguishability security for the constructed OT, assuming CPA-security for PKE and receiver privacy for wSFE. We need to show

$$(\mathsf{crs}', \mathsf{pk}, \mathsf{ct}, \mathsf{z}_1) \stackrel{c}{\equiv} (\mathsf{crs}', \mathsf{pk}, \mathsf{ct}', \mathsf{z}_1'), \tag{9}$$

where $\mathsf{crs}' \stackrel{\$}{\leftarrow} \mathsf{wSFE.Setup}(1^\lambda)$ and $(\mathsf{pk}, \mathsf{sk}) \stackrel{\$}{\leftarrow} \mathsf{PKE.KeyGen}(1^\lambda)$, $\mathsf{r} \stackrel{\$}{\leftarrow} \{0,1\}^\lambda$, $\mathsf{ct} \stackrel{\$}{\leftarrow} \mathsf{PKE.E}(\mathsf{pk}, 0; \mathsf{r})$, $(\mathsf{z}_1, *) \stackrel{\$}{\leftarrow} \mathsf{wSFE.Receiver}_1(\mathsf{crs}', (0, r))$, $\mathsf{ct}' \stackrel{\$}{\leftarrow} \mathsf{PKE.E}(\mathsf{pk}, 1; \mathsf{r})$ and $(\mathsf{z}_1', *) \stackrel{\$}{\leftarrow} \mathsf{wSFE.Receiver}_1(\mathsf{crs}', (1, r))$. To this end consider the following sequence of distributions:

- Dist_0: As in $(\mathsf{crs}', \mathsf{pk}, \mathsf{ct}, \mathsf{z}_1)$, corresponding to the lefthand side of Equation 9.
- Dist_1: Return $(\mathsf{crs}', \mathsf{pk}, \mathsf{ct}, \mathsf{z}_1^*)$, sampled same as Dist_0, except we use "fresh input" for generating z_1^*: sample $\mathsf{z}_1^* \stackrel{\$}{\leftarrow} \mathsf{wSFE.Receiver}_1(\mathsf{crs}', (0, \mathsf{r}'))$, for $\mathsf{r}' \stackrel{\$}{\leftarrow} \{0,1\}^\lambda$.
- Dist_2: Return $(\mathsf{crs}', \mathsf{pk}, \mathsf{ct}', \mathsf{z}_1^*)$, sampled as in Dist_1, except we switch the plaintext bit of the ciphertext: sample $\mathsf{ct}' \stackrel{\$}{\leftarrow} \mathsf{PKE.E}(\mathsf{pk}, 1; \mathsf{r})$.
- Dist_3: Return $(\mathsf{crs}', \mathsf{pk}, \mathsf{ct}', \mathsf{z}_1')$, sampled as in the righthand side of Equation 9.

By the receiver privacy of wSFE we have $\mathsf{Dist}_0 \stackrel{c}{\equiv} \mathsf{Dist}_1$. By the CPA security of PKE we have $\mathsf{Dist}_1 \stackrel{c}{\equiv} \mathsf{Dist}_2$. Finally, by the receiver privacy of the wSFE scheme we have $\mathsf{Dist}_2 \stackrel{c}{\equiv} \mathsf{Dist}_3$. The proof is now complete. $\square$

7.3 Sender's UC-Security

We will now show that our protocol provides sender's UC-security.

Let C^* be a boolean circuit of the same size and topology as C (that is, only differing in hardwired inputs) computing the following function.

- $\mathsf{C}^*[\mathsf{pk}, \mathsf{ct}, b^*, \mathsf{m}](b, \mathsf{r})$: Check if $b = b^*$ and $\mathsf{PKE.E}(\mathsf{pk}, b; \mathsf{r}) = \mathsf{ct}$. If so output m, otherwise $\perp$.

Simulating the receiver. The simulator $\mathcal{S}$ in the ideal model, which simulates an adversary $\mathcal{A}$ in the real model, acts as follows. First, $\mathcal{S}$ generates $\mathsf{crs}' \stackrel{\$}{\leftarrow} \mathsf{wSFE.Setup}(1^\lambda)$ and $(\mathsf{pk}, \mathsf{sk}) \stackrel{\$}{\leftarrow} \mathsf{PKE.KeyGen}(1^\lambda)$, and sets $\mathsf{crs} := (\mathsf{crs}', \mathsf{pk})$. When the parties call the ideal functionality $\mathcal{F}_{\mathrm{CRS}}$, then $\mathcal{S}$ return crs. Whenever $\mathcal{A}$ (corrupting the receiver) submits a protocol message $(\mathsf{sid}, (\mathsf{ct}, \mathsf{z}_1))$, then:

1. $\mathcal{S}$ first runs $\mathsf{PKE.Dec}(\mathsf{sk}, \mathsf{ct})$ to get $b^* \in \{0,1\}$;
2. $\mathcal{S}$ send $(\mathsf{sid}, \mathsf{receiver}, b^*)$ to the ideal functionality $\mathcal{F}_{\mathrm{OT}}$ to get m; then $\mathcal{S}$ stores the values of sid and m.

3. Whenever the dummy sender is activated for the same session sid, the simulator $\mathcal{S}$ sends the adversary $\mathcal{A}$ the message

$$\mathsf{z}_2 \xleftarrow{\$} \mathsf{wSFE.Sender}(\mathsf{crs}', \mathsf{C}^*[\mathsf{pk}, \mathsf{ct}, b^*, \mathsf{m}], \mathsf{z}_1).$$

Notice that for pk, ct, m and b^* formed as above, and for any pair $(\mathsf{m}'_0, \mathsf{m}'_1)$ such that $\mathsf{m}'_{b^*} = \mathsf{m}$, we have $\mathsf{C}^*[\mathsf{pk}, \mathsf{ct}, b^*, \mathsf{m}] \equiv \mathsf{C}[\mathsf{pk}, \mathsf{ct}, \mathsf{m}'_0, \mathsf{m}'_1]$. Thus, by the sender privacy of wSFE

$$\mathsf{IDEAL}_{\mathcal{F}_{\mathrm{OT}}, \mathcal{S}, \mathcal{Z}} \stackrel{c}{\equiv} \mathsf{EXEC}_{\mathsf{OT}, \mathcal{A}, \mathcal{Z}},$$

and the proof is complete. □

Finally, we mention that the OT protocol constructed in Construction 7.1 satisfies a receiver-extractability property, which was (implicitly) used in the proof of sender's UC security. Since we will use this definition later, we formalize it below.

Definition 7.3. *We say that an OT protocol* $(\mathsf{Setup}, \mathsf{OT}_1, \mathsf{OT}_2, \mathsf{OT}_3)$ *has receiver extractability if the setup algorithm* $\mathsf{Setup}(1^\lambda)$ *in addition to* crs *also outputs a trapdoor key* σ *and if there is a PPT algorithm* $\mathsf{Extract}$*, for which the following holds: for any stateful PPT adversary* $\mathcal{A} := (\mathcal{A}_1, \mathcal{A}_2)$*, assuming* $(\mathsf{m}_0, \mathsf{m}_1, \mathsf{otr}) \xleftarrow{\$} \mathcal{A}_1(\mathsf{crs})$ *and* $b = \mathsf{Extract}(\sigma, \mathsf{otr})$*, then* $\mathcal{A}_2$ *cannot distinguish between the outputs of* $\mathsf{OT}_2(\mathsf{crs}, \mathsf{otr}, (\mathsf{m}_0, \mathsf{m}_1))$ *and* $\mathsf{OT}_2(\mathsf{crs}, \mathsf{otr}, (\mathsf{m}_b, \mathsf{m}_b))$*.*

8 2-Round ZK from Sender-UC OT and Σ-protocols

In this section we give a two-round (statement-independent) ZK protocol against malicious verifiers in the CRS model based on a special type of Σ-protocols and an OT with sender's UC-security and receiver's indistinguishability security.

We first start by defining the properties we require of our Σ-protocol, and will then define the notion of statement-independent ZK protocols that we would like to achieve. Our notion of Σ-protocols is what Holmgren and Lombardi [HL18] called *extractable Σ-protocols*, defined as follows.

Definition 8.1 (Extractable Σ-protocols [HL18])**.** *A CRS-based Σ-protocol* $(\mathsf{Setup}, \mathsf{P}, \mathsf{V}, \mathsf{Extract}, \mathsf{Sim})$ *for a language* $\mathsf{L} \in \mathsf{NP}$ *is a three-round argument system between a prover* $\mathsf{P} := (\mathsf{P}_1, \mathsf{P}_2)$ *and a verifier* V*, where the prover is the initiator of the protocol and where the verifier's only message is a random bit* $b \in \{0,1\}$*. The setup algorithm* $(\mathsf{crs}, \sigma) \xleftarrow{\$} \mathsf{Setup}(1^\lambda)$ *returns a CRS value* crs *together with an associated trapdoor key* σ*. The trapdoor key* σ *will only play a role in the extractability requirement. We require the following properties:*

- *Completeness: For all* λ*, all* $(\mathsf{x}, \mathsf{w}) \in \mathsf{R}$ *(where* R *is the underlying relation), we have* $\Pr[\mathsf{V}(\mathsf{crs}, \mathsf{x}, \mathsf{a}, b, \mathsf{z}) = 1] = 1$*, where the probability is taken over* $(\mathsf{crs}, \sigma) \xleftarrow{\$} \mathsf{Setup}(1^\lambda)$*,* $(\mathsf{a}, \mathsf{st}) \xleftarrow{\$} \mathsf{P}_1(\mathsf{crs}, \mathsf{x}, \mathsf{w})$*,* $b \xleftarrow{\$} \{0,1\}$ *and* $\mathsf{z} \xleftarrow{\$} \mathsf{P}_2(\mathsf{st}, b)$*.*

- *Special soundness and extractability: For any value* crs *generated as* $(\mathsf{crs}, \sigma) \xleftarrow{\$} \mathsf{Setup}(1^\lambda)$*, any* $\mathsf{x} \notin \mathsf{L}$ *and any (possibly malicious) first-round message* a*, there exists at most one* $b \in \{0,1\}$ *for which there exists* z *such that* $\mathsf{V}(\mathsf{crs}, \mathsf{x}, \mathsf{a}, b, \mathsf{z}) = 1$*. Moreover, for such parameters, this unique value of* b *(if any) can be computed efficiently as* $\mathsf{Extract}(\sigma, \mathsf{x}, \mathsf{a})$*.*

- *Honest-verifier zero knowledge: For any value* crs *generated as* $(\mathsf{crs}, \sigma) \xleftarrow{\$} \mathsf{Setup}(1^\lambda)$*, any* $b \in \{0,1\}$ *and any* $(\mathsf{x}, \mathsf{w}) \in \mathsf{R}$*:*

$$(\mathsf{crs}, \mathsf{x}, \mathsf{a}, b, \mathsf{z}) \stackrel{c}{\equiv} (\mathsf{crs}, \mathsf{x}, \mathsf{a}', b, \mathsf{z}'), \tag{10}$$

where $(\mathsf{a}, \mathsf{st}) \xleftarrow{\$} \mathsf{P}_1(\mathsf{crs}, \mathsf{x}, \mathsf{w})$*,* $\mathsf{z} \xleftarrow{\$} \mathsf{P}_2(\mathsf{st}, b)$ *and* $(\mathsf{a}', \mathsf{z}') \xleftarrow{\$} \mathsf{Sim}(\mathsf{crs}, \mathsf{x}, b)$*.*

We will now define out notion of CRS-based two-round statement-independent ZK. Informally, a two-round ZK protocol is statement-independent if the verifier's message in the protocol is independent of the statement being proven.

Definition 8.2 (Two-round statement-independent zero knowledge)**.** *A two-round zero-knowledge argument system for a language* $\mathsf{L} \in \mathsf{NP}$ *with a corresponding relation* R *in the CRS model consists of four PPT algorithms* $\mathsf{ZK} = (\mathsf{Setup}, \mathsf{P}, \mathsf{V} := (\mathsf{V}_1, \mathsf{V}_2), \mathsf{Sim} := (\mathsf{Sim}_1, \mathsf{Sim}_2))$*, defined as follows. The setup algorithm* Setup *on input* 1^λ *outputs a value* crs*. The verifier algorithm* $\mathsf{V}_1(\mathsf{crs})$ *on input* crs *returns a message* msgv *together with a private state* st*. We stress that the verifier does not take as input any statement* x*, hence the "statement-independent" name. The prover algorithm* $\mathsf{P}(\mathsf{crs}, \mathsf{x}, \mathsf{w}, \mathsf{msgv})$ *on input* crs*, a statement* x *with a corresponding witness* w *and a verifier's message* msgv*, outputs a message* msgp*. Finally, the algorithm* $\mathsf{V}_2(\mathsf{st}, \mathsf{x}, \mathsf{msgp})$ *outputs a bit* b*. We require the following properties.*

- *Completeness: For all* $(\mathsf{x}, \mathsf{w}) \in \mathsf{L}$ *we have* $\Pr[\mathsf{V}_2(\mathsf{st}, \mathsf{x}, \mathsf{msgp}) = 1] = 1$*, where* $\mathsf{crs} \xleftarrow{\$} \mathsf{Setup}(1^\lambda)$*,* $(\mathsf{msgv}, \mathsf{st}) \xleftarrow{\$} \mathsf{V}_1(\mathsf{crs})$ *and* $\mathsf{msgp} \xleftarrow{\$} \mathsf{P}(\mathsf{crs}, \mathsf{x}, \mathsf{w}, \mathsf{msgv})$*.*

- *Adaptive soundness: No PPT malicious prover can convince an honest verifier of a false statement, even if the statement is chosen adaptively after seeing* crs *and the verifier's (statement-independent) message. Formally, for any PPT adversary* P^* *the following holds:* $\Pr[\mathsf{V}_2(\mathsf{st}, \mathsf{x}, \mathsf{msgp}) = 1 \wedge \mathsf{x} \notin \mathsf{L}] = \mathsf{negl}(\lambda)$*, where* $\mathsf{crs} \xleftarrow{\$} \mathsf{Setup}(1^\lambda)$*,* $(\mathsf{msgv}, \mathsf{st}) \xleftarrow{\$} \mathsf{V}_1(\mathsf{crs})$*,* $(\mathsf{x}, \mathsf{msgp}) \xleftarrow{\$} \mathsf{P}^*(\mathsf{crs}, \mathsf{msgv})$*.*

- *Adaptive Malicious Zero-Knowledge (ZK): Let* $\mathsf{V}^* = (\mathsf{V}_1^*, \mathsf{V}_2^*)$ *be a stateful two-phase adversary where* V_2^* *outputs a bit. Let the experiment* $\mathsf{Exp}_{\mathrm{ZK}}(\mathsf{V}^*)$ *be defined as follows:*

  1. *Choose* $b \xleftarrow{\$} \{0,1\}$
  2. *If* $b = 0$*, sample* $\mathsf{crs} \xleftarrow{\$} \mathsf{Setup}(1^\lambda)$*. Else, sample* $(\mathsf{crs}, \sigma) \xleftarrow{\$} \mathsf{Sim}_1(1^\lambda)$*.*
  3. *Let* $(\mathsf{x}, \mathsf{w}, \mathsf{msgv}) \xleftarrow{\$} \mathsf{V}_1^*(\mathsf{crs})$*. If* $\mathsf{R}(\mathsf{x}, \mathsf{w}) = 0$*, then halt.*
  4. *If* $b = 0$*, let* $\mathsf{msgp} \xleftarrow{\$} \mathsf{P}(\mathsf{crs}, \mathsf{x}, \mathsf{w}, \mathsf{msgv})$*. Else, let* $\mathsf{msgp} \xleftarrow{\$} \mathsf{Sim}_2(\sigma, \mathsf{x}, \mathsf{msgv})$*.*
  5. *Compute* $b' \xleftarrow{\$} \mathsf{V}_2^*(\mathsf{msgp})$*.*
  6. *If* $b' = b$ *output 1, otherwise 0.*

  Define $\mathsf{Adv}_{\mathrm{ZK}}(\mathsf{V}^*) = |\Pr[\mathsf{Exp}_{\mathrm{ZK}}(\mathsf{V}^*) = 1] - 1/2|$*. We say that the scheme is zero-knwoledge if for all PPT adversaries* V^**,* $\mathsf{Adv}_{\mathrm{ZK}}(\mathsf{V}^*) = \mathsf{negl}(\lambda)$*.*

Construction 8.3 (Two-round ZK)**.** *Let* $\mathsf{OT} := (\mathsf{Setup}, \mathsf{OT}_1, \mathsf{OT}_2, \mathsf{OT}_3)$ *be an* OT *protocol and let* $\mathsf{SIGM} := (\mathsf{Setup}, \mathsf{P}, \mathsf{V}, \mathsf{Extract}, \mathsf{Sim})$ *be an extractable* Σ*-protocol for a language* $\mathsf{L} \in \mathsf{NP}$ *(Definition 8.1). We give a two-round ZK protocol* $\mathsf{ZK} := (\mathsf{Setup}, \mathsf{P}, \mathsf{V} := (\mathsf{V}_1, \mathsf{V}_2))$ *for* L *as follows. The construction is parameterized over a polynomial* $r := r(\lambda)$*, which we will instantiate in the soundness proof.*

- $\mathsf{ZK.Setup}(1^\lambda)$: *Run* $\mathsf{crs}_{\mathrm{ot}} \xleftarrow{\$} \mathsf{OT.Setup}(1^\lambda)$ *and* $(\mathsf{crs}_{\mathrm{sig}}, \sigma) \xleftarrow{\$} \mathsf{SIGM.Setup}(1^\lambda)$. *Return* $\mathsf{crs} := (\mathsf{crs}_{\mathrm{ot}}, \mathsf{crs}_{\mathrm{sig}})$.

- $\mathsf{ZK.V}_1(\mathsf{crs} := (\mathsf{crs}_{\mathrm{ot}}, \mathsf{crs}_{\mathrm{sig}}))$: *For each* $i \in [r]$, *sample* $b_i \xleftarrow{\$} \{0,1\}$. *Let* $(\vec{\mathsf{otr}}, \vec{\mathsf{st}}_{\mathrm{ot}}) \xleftarrow{\$} \mathsf{OT}_1(\mathsf{crs}_{\mathrm{ot}}, \vec{b})$, *where* $\vec{b} := (b_1, \ldots, b_r)$. *Return* $(\mathsf{msgv}, \mathsf{st})$, *where* $\mathsf{msgv} := \vec{\mathsf{otr}}$ *is the message to the prover* P, *and* $\mathsf{st} := (b_1, \ldots, b_r, \vec{\mathsf{st}}_{\mathrm{ot}})$ *is the private state.*

- $\mathsf{ZK.P}(\mathsf{crs} := (\mathsf{crs}_{\mathrm{ot}}, \mathsf{crs}_{\mathrm{sig}}), \mathsf{x}, \mathsf{w}, \mathsf{msgv})$: *For each* $i \in [r]$ *sample* $(\mathsf{a}_i, \mathsf{sts}_i) \xleftarrow{\$} \mathsf{SIGM.P}_1(\mathsf{crs}_{\mathrm{sig}}, \mathsf{x}, \mathsf{w})$. *For each* $i \in [r]$ *and* $b \in \{0,1\}$, *form* $\mathsf{z}_{i,b} \xleftarrow{\$} \mathsf{SIGM.P}_2(\mathsf{sts}_i, b)$, *which is the prover's last message in the* Σ*-protocol when his first message was* a_i *and when the verifier's challenge bit is* b. *Return* $\mathsf{msgp} := (\vec{\mathsf{a}}, \mathsf{OT}_2(\mathsf{crs}_{\mathrm{ot}}, \vec{\mathsf{otr}}, \vec{\mathsf{z}_0}, \vec{\mathsf{z}_1}))$, *where* $\vec{\mathsf{a}} := (\mathsf{a}_1, \ldots, \mathsf{a}_r)$, $\vec{\mathsf{z}_0} := (\mathsf{z}_{1,0}, \ldots, \mathsf{z}_{r,o})$ *and* $\vec{\mathsf{z}_1} := (\mathsf{z}_{1,1}, \ldots, \mathsf{z}_{r,1})$.

- $\mathsf{ZK.V}_2(\mathsf{st}, \mathsf{x}, \mathsf{msgp})$: *Parse* $\mathsf{st} := (b_1, \ldots, b_r, \vec{\mathsf{st}}_{\mathrm{ot}})$, $\mathsf{msgp} := (\vec{\mathsf{a}}, \vec{\mathsf{ots}})$ *and* $\vec{\mathsf{a}} := (\mathsf{a}_1, \ldots, \mathsf{a}_r)$. *Let* $(\mathsf{z}_1, \ldots, \mathsf{z}_r) = \mathsf{OT}_3(\vec{\mathsf{st}}_{\mathrm{ot}}, \vec{\mathsf{ots}})$. *Return* 1 *if for all* $i \in [r]$: $\mathsf{SIGM.V}(\mathsf{crs}_{\mathrm{sig}}, \mathsf{x}, \mathsf{a}_i, b_i, \mathsf{z}_i) = 1$. *Otherwise, return* 0.

Theorem 8.4. *Assuming that* $\mathsf{OT} := (\mathsf{Setup}, \mathsf{OT}_1, \mathsf{OT}_2, \mathsf{OT}_3)$ *provides sender's UC-security and receiver's indistinguishability security, and that* $\mathsf{SIGM} := (\mathsf{Setup}, \mathsf{P}, \mathsf{V}, \mathsf{Extract}, \mathsf{Sim})$ *is an extractable* Σ*-protocol for a language* L *(Definition 8.1), then the protocol* ZK *given in Construction 8.3 satisfies completeness, adaptive soundness and adaptive malicious zero knowledge for* L.

Before proving the theorem, since CPA-secure PKE schemes imply the existence of extractable Σ-protocols (see [HL18] for the construction) we have the following corollary.

Corollary 8.5. *Assuming the existence of two-round OT with sender's UC security and receiver's indistinguishability security, and CPA-secure PKE with perfect correctness, there exists a two-round ZK protocol (in the sense of Definition 8.2) for any language* $\mathsf{L} \in \mathsf{NP}$.

Proof of completeness. The proof follows in a straightforward way from the completeness of the underlying OT and the Σ-protocol. □

Proof of adaptive soundness. We show that there does not exist a prover P^* for which the following holds with a non-negligible probability: the prover $\mathsf{P}^*(\mathsf{crs}, \mathsf{msgv})$, after seeing both crs and the verifier's (statement-independent) message msgv, manages to convince the verifier on a statement $\mathsf{x} \notin \mathsf{L}$. We prove this via a reduction to the receiver's indistinguishability security of the underlying OT scheme $\mathsf{OT} := (\mathsf{Setup}, \mathsf{OT}_1, \mathsf{OT}_2, \mathsf{OT}_3)$, through an adversary $\mathcal{A}$ as follows.

$\mathcal{A}(\mathsf{crs}_{\mathrm{ot}}, \vec{\mathsf{otr}} := (\mathsf{otr}_1, \ldots, \mathsf{otr}_r))$:

1. Sample $(\mathsf{crs}_{\mathrm{sig}}, \sigma) \xleftarrow{\$} \mathsf{SIGM.Setup}(1^\lambda)$. Set $\mathsf{crs} := (\mathsf{crs}_{\mathrm{ot}}, \mathsf{crs}_{\mathrm{sig}})$.
2. Invoke $\mathsf{P}^*(\mathsf{crs}, \vec{\mathsf{otr}})$ to get $(\mathsf{x}, \mathsf{msgp} := (\vec{\mathsf{a}}, \vec{\mathsf{ots}}))$. Parse $\vec{\mathsf{a}} := (\mathsf{a}_1, \ldots, \mathsf{a}_r)$. For $i \in [r]$ let $b_i := \mathsf{Extract}(\sigma, \mathsf{x}, \mathsf{a}_i)$.
3. Return $(b_1, \ldots, b_r)$ as the guess bits for the receiver's r bits.

To see why the reduction works, suppose $b'_1, \ldots, b'_r$ are the OT-receiver's challenge choice bits, namely for $i \in [r]$ the verifier sampled otr_i as $\mathsf{OT}_1(\mathsf{crs}_{\rm ot}, b'_i)$. Let $(\mathsf{x}, \mathsf{msgp} := (\vec{\mathsf{a}}, \vec{\mathsf{ots}})) \xleftarrow{\$} \mathsf{P}^*(\mathsf{crs}, \vec{\mathsf{otr}})$ and suppose $\mathsf{x} \notin \mathsf{L}$. (This happens with non-negligible probability.) If the verifier accepts the proof msgp on input x, then since $\mathsf{x} \notin \mathsf{L}$, by the completeness of the base OT scheme and the extractability property of SIGM we must have $b'_i = \mathsf{SIGM.Extract}(\sigma, \mathsf{x}, \mathsf{a}_i)$. □

Proof of malicious zero-knowledge. We now show that the protocol is malicious zero-knowledge, assuming the Σ-protocol is honest-verifier zero knowledge and the base OT has sender's UC security. For simplicity of exposition, we assume that the base OT scheme has the receiver-extractability property (Definition 7.3), which is anyway provided by the OT scheme given in Construction 7.1. We mention that we do not need this property and we can prove zero knowledge by assuming sender's UC security instead of receiver extractability, but giving the proof based on this property makes the presentation simpler.

In the following, let $\mathsf{OT.Extract}$ be the extraction algorithm for the receiver's input bit, guaranteed by receiver extractability. We define $\mathsf{ZK.Sim} := (\mathsf{ZK.Sim}_1, \mathsf{ZK.Sim}_2)$ as follows.

$\mathsf{ZK.Sim}_1(1^\lambda)$

1. Sample $(\mathsf{crs}_{\rm ot}, \sigma_{\rm ot}) \xleftarrow{\$} \mathsf{OT.Setup}(1^\lambda)$ and $\mathsf{crs}_{\rm sig} \xleftarrow{\$} \mathsf{SIGM.Setup}(1^\lambda)$. Let $\mathsf{crs} := (\mathsf{crs}_{\rm ot}, \mathsf{crs}_{\rm sig})$. Return crs as the CRS and $(\mathsf{crs}_{\rm ot}, \mathsf{crs}_{\rm sig}, \sigma_{\rm ot})$ as the private state.

$\mathsf{ZK.Sim}_2(\mathsf{crs}_{\rm ot}, \mathsf{crs}_{\rm sig}, \sigma_{\rm ot}, \mathsf{x}, \mathsf{msgv})$

1. Parse $\mathsf{msgv} := (\mathsf{otr}_1, \ldots, \mathsf{otr}_r)$.
2. For $i \in [r]$, extract $b_i := \mathsf{OT.Extract}(\sigma_{\rm ot}, \mathsf{otr}_i)$.
3. For $i \in [r]$ let $(\mathsf{a}_i, \mathsf{z_i}) \xleftarrow{\$} \mathsf{SIGM.Sim}(\mathsf{crs}_{\rm sig}, \mathsf{x}, b_i)$. Set $\vec{\mathsf{a}} := (\mathsf{a}_1, \ldots, \mathsf{a}_r)$, and $\vec{\mathsf{z}} := (\mathsf{z}_1, \ldots, \mathsf{z}_r)$.
4. Return $(\vec{\mathsf{a}}, \mathsf{OT}_2(\mathsf{crs}_{\rm ot}, \vec{\mathsf{z}}, \vec{\mathsf{z}}, \vec{\mathsf{otr}}))$.

The fact that the above simulation $\mathsf{ZK.Sim}$ provides a computationally indistinguishable view (in the sense of Definition 8.2) follows immediately from receiver-extractability of OT as well as the zero-knowledge property of the underlying Σ-protocol. □

9 UC-Secure OT from Sender-UC OT and Zero Knowledge

We will now show how to build a UC-secure OT scheme (with both receiver's and sender's UC security) from the combination of a CPA-secure PKE scheme, a CRS-based two-round statement-independent ZK protocol, and a two-round OT scheme with sender's UC-security and receiver's indistinguishability security.

Let $(\mathsf{Setup}, \mathsf{OT}_1, \mathsf{OT}_2, \mathsf{OT}_3)$ be the base two-round OT scheme, $\mathsf{PKE} := (\mathsf{KeyGen}, \mathsf{E}, \mathsf{Dec})$ be the PKE scheme and $\mathsf{ZK} = (\mathsf{Setup}, \mathsf{P}, \mathsf{V} := (\mathsf{V}_1, \mathsf{V}_2), \mathsf{Sim} := (\mathsf{Sim}_1, \mathsf{Sim}_2))$ be a two-round statement-independent ZK protocol for the language $\mathsf{L}_{\mathsf{pk}, \mathsf{crs}_{\rm ot}, \mathsf{otr}} \in \mathsf{NP}$, parameterized over a public key pk of the PKE scheme, a CRS value $\mathsf{crs}_{\rm ot}$ of the OT scheme and an OT-receiver's message otr, defined as follows:

$$\mathsf{L}_{\mathsf{pk},\mathsf{crs}_{\mathrm{ot}},\mathsf{otr}} = \big\{(\mathsf{ct}_0, \mathsf{ct}_1, \mathsf{ots}) \mid \exists(\mathsf{m}_0, \mathsf{m}_1, \mathsf{r}_0, \mathsf{r}_1, \mathsf{r}) \text{ s.t. } \mathsf{ct}_0 = \mathsf{E}(\mathsf{pk}, \mathsf{m}_0; \mathsf{r}_0), \mathsf{ct}_1 = \mathsf{E}(\mathsf{pk}, \mathsf{m}_1; \mathsf{r}_1) \\ \mathsf{ots} = \mathsf{OT}_2(\mathsf{crs}_{\mathrm{ot}}, \mathsf{otr}, \mathsf{m}_0, \mathsf{m}_1; \mathsf{r})\big\}. \quad (11)$$

Construction 9.1 (UC-secure OT). *We build* $\mathsf{OT}' := (\mathsf{Setup}', \mathsf{OT}'_1, \mathsf{OT}'_2, \mathsf{OT}'_3)$ *from the above primitives as follows.*

$\mathsf{Setup}'(1^\lambda)$: *Sample* $(\mathsf{pk}, \mathsf{sk}) \xleftarrow{\$} \mathsf{PKE.Gen}(1^\lambda)$, $\mathsf{crs}_{\mathrm{ot}} \xleftarrow{\$} \mathsf{OT.Setup}(1^\lambda)$ *and* $\mathsf{crs}_{\mathrm{zk}} \xleftarrow{\$} \mathsf{ZK.Setup}(1^\lambda)$. *Output* $\mathsf{crs} := (\mathsf{pk}, \mathsf{crs}_{\mathrm{ot}}, \mathsf{crs}_{\mathrm{zk}})$.

$\mathsf{OT}'_1(\mathsf{crs}, b)$: *Parse* $\mathsf{crs} := (\mathsf{pk}, \mathsf{crs}_{\mathrm{ot}}, \mathsf{crs}_{\mathrm{zk}})$. *Sample* $(\mathsf{otr}, \mathsf{st}_{\mathrm{ot}}) \xleftarrow{\$} \mathsf{OT}_1(\mathsf{crs}_{\mathrm{ot}}, b)$ *and* $(\mathsf{msgv}, \mathsf{st}_{\mathrm{zk}}) \xleftarrow{\$} \mathsf{ZK.V}_1(\mathsf{crs}_{\mathrm{zk}})$. *Output* $\mathsf{otr}' := (\mathsf{otr}, \mathsf{msgv})$ *as the message to the sender and output* $\mathsf{st} := (\mathsf{st}_{\mathrm{ot}}, \mathsf{st}_{\mathrm{zk}})$ *as the private state.*

$\mathsf{OT}'_2(\mathsf{crs}, \mathsf{otr}', \mathsf{m}_0, \mathsf{m}_1)$: *Parse* $\mathsf{crs} := (\mathsf{pk}, \mathsf{crs}_{\mathrm{ot}}, \mathsf{crs}_{\mathrm{zk}})$ *and* $\mathsf{otr}' := (\mathsf{otr}, \mathsf{msgv})$. *Sample* $\mathsf{r}, \mathsf{r}_0, \mathsf{r}_1 \xleftarrow{\$} \{0,1\}^*$. *Let* $\mathsf{ct}_0 := \mathsf{E}(\mathsf{pk}, \mathsf{m}_0; \mathsf{r}_0)$, $\mathsf{ct}_1 = \mathsf{E}(\mathsf{pk}, \mathsf{m}_1; \mathsf{r}_1)$, *and* $\mathsf{ots} = \mathsf{OT}_2(\mathsf{crs}_{\mathrm{ot}}, \mathsf{otr}, \mathsf{m}_0, \mathsf{m}_1; \mathsf{r})$. *Set* $\mathsf{x} := (\mathsf{ct}_0, \mathsf{ct}_1, \mathsf{ots})$ *and* $\mathsf{w} := (\mathsf{m}_0, \mathsf{m}_1, \mathsf{r}_0, \mathsf{r}_1, \mathsf{r})$. *Output* $\mathsf{ots}' := (\mathsf{ct}_0, \mathsf{ct}_1, \mathsf{ots}, \mathsf{msgp})$, *where* $\mathsf{msgp} \xleftarrow{\$} \mathsf{ZK.P}(\mathsf{crs}_{\mathrm{zk}}, \mathsf{x}, \mathsf{w}, \mathsf{msgv})$.

$\mathsf{OT}'_3(\mathsf{st}, \mathsf{ots}')$: *Parse* $\mathsf{st} := (\mathsf{st}_{\mathrm{ot}}, \mathsf{st}_{\mathrm{zk}})$, $\mathsf{ots}' := (\mathsf{ct}_0, \mathsf{ct}_1, \mathsf{ots}, \mathsf{msgp})$ *and let* $\mathsf{x} := (\mathsf{ct}_0, \mathsf{ct}_1, \mathsf{ots})$. *If* $\mathsf{ZK.V}_2(\mathsf{st}_{\mathrm{zk}}, \mathsf{x}, \mathsf{msgp}) \neq 1$, *then return* $\perp$. *Otherwise, return* $\mathsf{OT}_3(\mathsf{st}_{\mathrm{ot}}, \mathsf{ots})$.

Theorem 9.2. *Assuming that* $\mathsf{OT} := (\mathsf{Setup}, \mathsf{OT}_1, \mathsf{OT}_2, \mathsf{OT}_3)$ *provides sender's UC-security and receiver's indistinguishability security, that* $\mathsf{PKE} := (\mathsf{KeyGen}, \mathsf{E}, \mathsf{Dec})$ *is a CPA-secure scheme, and that* ZK *is a two-round ZK protocol for the language* L *described in Equation 11, then the OT protocol* OT' *given in Construction 9.1 satisfies completeness and UC security.*

Correctness of the above constructed OT protocol follows immediately by the correctness of constituent primitives. We will now prove that the protocol is UC secure.

Proof of receiver's UC-security. We now focus on the case that the sender is corrupted. Fix the the real-world adversary $\mathcal{A}$. First, $\mathcal{S}$ samples $(\mathsf{pk}, \mathsf{sk}) \xleftarrow{\$} \mathsf{PKE.Gen}(1^\lambda)$, $\mathsf{crs}_{\mathrm{ot}} \xleftarrow{\$} \mathsf{OT.Setup}(1^\lambda)$ and $\mathsf{crs}_{\mathrm{zk}} \xleftarrow{\$} \mathsf{ZK.Setup}(1^\lambda)$, and sets $\mathsf{crs} := (\mathsf{pk}, \mathsf{crs}_{\mathrm{ot}}, \mathsf{crs}_{\mathrm{zk}})$. When the parties call the ideal functionality $\mathcal{F}_{\mathrm{CRS}}$, then $\mathcal{S}$ returns crs. Whenever the dummy receiver is activated on a session sid, the simulator $\mathcal{S}$ samples $(\mathsf{otr}' := (\mathsf{otr}, \mathsf{msgv}), \mathsf{st} := (\mathsf{st}_{\mathrm{ot}}, \mathsf{st}_{\mathrm{zk}})) \xleftarrow{\$} \mathsf{OT}'_1(\mathsf{crs}_{\mathrm{ot}}, 0)$, sends otr' to $\mathcal{A}$, and stores all these values with their corresponding session sid. When $\mathcal{A}$ replies with a message $(\mathsf{sid}, \mathsf{ots}' := (\mathsf{ct}_0, \mathsf{ct}_1, \mathsf{ots}, \mathsf{msgp}))$, then $\mathcal{S}$ computes $b = \mathsf{ZK.V}_2(\mathsf{st}_{\mathrm{zk}}, (\mathsf{ct}_0, \mathsf{ct}_1, \mathsf{ots}), \mathsf{msgp})$; if $b = 1$, then $\mathcal{S}$ sends $(\mathsf{PKE.Dec}(\mathsf{sk}, \mathsf{ct}_0), \mathsf{PKE.Dec}(\mathsf{sk}, \mathsf{ct}_1))$ to $\mathcal{F}_{\mathrm{OT}}$; otherwise, $\mathcal{S}$ sends $\perp$ to $\mathcal{F}_{\mathrm{OT}}$.

To prove $\mathsf{IDEAL}_{\mathcal{F}_{\mathrm{OT}},\mathcal{S},\mathcal{Z}} \stackrel{c}{\equiv} \mathsf{EXEC}_{\mathsf{OT}',\mathcal{A},\mathcal{Z}}$, consider a tweak $\mathcal{S}'$ of the simulator $\mathcal{S}$, where instead of sending $\mathsf{OT}'_1(\mathsf{crs}_{\mathrm{ot}}, 0)$, it sends $\mathsf{OT}'_1(\mathsf{crs}_{\mathrm{ot}}, b')$, where b' is the bit value of the dummy receiver. By receiver's indistinguishability security we have $\mathsf{IDEAL}_{\mathcal{F}_{\mathrm{OT}},\mathcal{S},\mathcal{Z}} \stackrel{c}{\equiv} \mathsf{IDEAL}_{\mathcal{F}_{\mathrm{OT}},\mathcal{S}',\mathcal{Z}}$. Finally, since the underlying PKE scheme PKE is perfectly correct, a distinguisher between $\mathsf{IDEAL}_{\mathcal{F}_{\mathrm{OT}},\mathcal{S}',\mathcal{Z}}$ and $\mathsf{EXEC}_{\mathsf{OT}',\mathcal{A},\mathcal{Z}}$ immediately translates into an adversary against the soundness of the scheme ZK. the proof is now complete. □

Proof of sender's UC-security. We show the proof for the case that the receiver is corrupted. Fix the the real-world adversary $\mathcal{A}$. Let $\mathcal{S}'$ be the simulator for the UC security of the base OT scheme OT against malicious receivers. First, $\mathcal{S}$ invokes $\mathcal{S}'$ to get $\mathsf{crs}_{\mathrm{ot}}$, and then $\mathcal{S}$ samples $(\mathsf{pk}, \mathsf{sk}) \xleftarrow{\$} \mathsf{PKE.Gen}(1^\lambda)$ and $(\mathsf{crs}_{\mathrm{zk}}, \sigma_{\mathrm{zk}}) \xleftarrow{\$} \mathsf{ZK.Sim}_1(1^\lambda)$, and sets $\mathsf{crs} := (\mathsf{pk}, \mathsf{crs}_{\mathrm{ot}}, \mathsf{crs}_{\mathrm{zk}})$. When the parties call the ideal functionality $\mathcal{F}_{\mathrm{CRS}}$, then $\mathcal{S}$ returns crs. Whenever $\mathcal{A}$ (corrupting the receiver) submits a protocol message $(\mathsf{sid}, (\mathsf{otr}, \mathsf{msgv}))$, then:

1. $\mathcal{S}$ extracts the bit b^* underlying otr via the simulator $\mathcal{S}'$;
2. $\mathcal{S}$ send $(\mathsf{sid}, \mathsf{receiver}, b^*)$ to the ideal functionality $\mathcal{F}_{\mathrm{OT}}$ to get m; then $\mathcal{S}$ stores the values of sid and m.
3. Whenever the dummy sender is activated for the same session sid, the simulator $\mathcal{S}$ forms $\mathsf{ots} \xleftarrow{\$} \mathsf{OT}_2(\mathsf{crs}_{\mathrm{ot}}, \mathsf{otr}, \mathsf{m}, \mathsf{m})$, $\mathsf{ct}_0 \xleftarrow{\$} \mathsf{PKE.E}(\mathsf{pk}, \mathsf{m})$, $\mathsf{ct}_1 \xleftarrow{\$} \mathsf{PKE.E}(\mathsf{pk}, \mathsf{m})$, and
$$\mathsf{msgp} \xleftarrow{\$} \mathsf{ZK.Sim}_2(\sigma_{\mathrm{zk}}, (\mathsf{ct}_0, \mathsf{ct}_1, \mathsf{ots}), \mathsf{msgv}).$$
Then $\mathcal{S}$ sends the adversary $\mathcal{A}$ the message $\mathsf{ots}' := (\mathsf{ots}, \mathsf{msgp})$.

To prove $\mathsf{IDEAL}_{\mathcal{F}_{\mathrm{OT}},\mathcal{S},\mathcal{Z}} \stackrel{c}{\equiv} \mathsf{EXEC}_{\mathsf{OT}',\mathcal{A},\mathcal{Z}}$ we define two modified versions of the constructed protocol OT', which we call them OT^* and OT^{**}. These two variations differ from the real protocol OT only in the output distribution of the sender's message in response to $(\mathsf{crs}, \mathsf{msgv}', (\mathsf{m}_0, \mathsf{m}_1))$.

- Protocol: OT^*: using the simulator to produce the ZK proof. The output message of the prover $\mathsf{ots}' := (\mathsf{ots}, \mathsf{msgp})$ is formed as follows: Form ots exactly as in $\mathsf{OT}_2(\mathsf{crs}, \mathsf{msgv}', (\mathsf{m}_0, \mathsf{m}_1))$, and form msgp as follows: $\mathsf{msgp} \xleftarrow{\$} \mathsf{ZK.Sim}_2(\sigma_{\mathrm{zk}}, (\mathsf{ct}_0, \mathsf{ct}_1, \mathsf{ots}), \mathsf{msgv}))$, where $\mathsf{ct}_0 \xleftarrow{\$} \mathsf{PKE.E}(\mathsf{pk}, \mathsf{m}_0)$ and $\mathsf{ct}_1 \xleftarrow{\$} \mathsf{PKE.E}(\mathsf{pk}, \mathsf{m}_1)$.
- Protocol: OT^{**}: exactly as in OT^*, except we form $\mathsf{ct}_0 \xleftarrow{\$} \mathsf{PKE.E}(\mathsf{pk}, \mathsf{m}_{b^*})$ and $\mathsf{ct}_1 \xleftarrow{\$} \mathsf{PKE.E}(\mathsf{pk}, \mathsf{m}_{b^*})$.

By the ZK property of ZK we have $\mathsf{EXEC}_{\mathsf{OT}',\mathcal{A},\mathcal{Z}} \stackrel{c}{\equiv} \mathsf{EXEC}_{\mathsf{OT}^*,\mathcal{A},\mathcal{Z}}$. By CPA security of PKE we have $\mathsf{EXEC}_{\mathsf{OT}^*,\mathcal{A},\mathcal{Z}} \stackrel{c}{\equiv} \mathsf{EXEC}_{\mathsf{OT}^{**},\mathcal{A},\mathcal{Z}}$. Finally, since the sender's strategy of OT^{**} works exactly like the simulating adversary $\mathcal{S}$, we have $\mathsf{EXEC}_{\mathsf{OT}^{**},\mathcal{A},\mathcal{Z}} \equiv \mathsf{IDEAL}_{\mathcal{F}_{\mathrm{OT}},\mathcal{S},\mathcal{Z}}$. The proof is now complete. $\square$

10 Instantiations from CDH and LPN

10.1 Instantiation from CDH

We first give a construction of elementary OT from CDH. In fact, we show that the construction also already directly satisfies the stronger notion of search OT security. The protocol is given in Figure 4.

Definition 10.1 (Computational Diffie-Hellman (CDH) assumption)**.** *Let G be a group-generator scheme, which on input 1^λ outputs $(\mathbb{G}, p, g)$, where $\mathbb{G}$ is the description of a group, p is the order of the group which is always a prime number and g is a generator of the group. We say that G is CDH-hard if for any PPT adversary $\mathcal{A}$: $\Pr[\mathcal{A}(\mathbb{G}, p, g, g^{a_1}, g^{a_2}) = g^{a_1 a_2}] = \mathsf{negl}(\lambda)$, where $(\mathbb{G}, p, g) \xleftarrow{\$} \mathsf{G}(1^\lambda)$ and $a_1, a_2 \xleftarrow{\$} \mathbb{Z}_p$.*

$$
\begin{array}{lcl}
\text{Sender}(X): & \text{CRS: } X := g^x & \text{Receiver}(X, c): \\
 & & r \leftarrow \mathbb{Z}_p \\
 & \xleftarrow{\quad \mathsf{otr} := h_0 \quad} & h_0 := g^r X^{-c} \\
h_1 := h_0 X & & \\
s \leftarrow \mathbb{Z}_p & & \\
S := g^s & \xrightarrow{\quad \mathsf{ots} := S \quad} & \\
\text{output } y_0 := h_0^s, y_1 := h_1^s & & \text{output } y_c := h_c^s = S^r
\end{array}
$$

Figure 4: Elementary and Search OT from CDH.

Lemma 10.2. *The protocol in Figure 4 satisfies statistical receiver's indistinguishability security.*

Proof. The distribution of the receiver's message $h_0 = g^r X^{-c}$ is uniformly random over the group $\mathbb{G}$ no matter that the receiver's bit c is. □

Lemma 10.3. *The protocol in Figure 4 satisfies sender's elementary security based on the CDH assumption.*

Proof. Let there be a PPT adversary $\mathcal{A}$ that breaks the elementary security of the sender. Then we are able to construct a PPT adversary $\mathcal{B}$ that breaks the CDH assumption. Recall that $\mathcal{A}$ receives a CRS $X = g^x$, sends a group element h_0, receives $S = g^s$ for a uniform s, and succeeds if he outputs $y_0 = h_0^s$, $y_1 = h_1^s = (h_0 X)^s$. Our adversary against the CDH assumption receives $\mathbb{G}$, p, g, $A_1 := g^{a_1}$, $A_2 := g^{a_2}$ from his challenger, gives CRS $X := A_1$ to $\mathcal{A}$, receives h_0, gives $S := A_2$ to $\mathcal{A}$, receives y_0, y_1 and outputs y_1/y_0. If $\mathcal{A}$ succeeds then $y_0 = h_0^s = h_0^{a_2}$, $y_1 = h_1^s = (h_0X)^s = h_0^b A_1^{a_2} = h_0^{a_2} g^{a_1 a_2}$ and therefore $y_1/y_0 = g^{a_1 a_2}$, meaning that $\mathcal{B}$ succeeds in solving CDH. □

The above two lemmas already show that the scheme in Figure 4 is a elementary OT scheme and we can then rely on our black-box transformations from the previous sections to then get UC secure OT under CDH assumption. Therefore, the following Theorem follows as a corollary.

Theorem 10.4. *Under the CDH assumption there exists a 2-round UC OT.*

Although the above lemmas already suffice to show the above corollary, we note that we can actually show something stronger about the scheme in Figure 4. Not only does it satisfy sender's elementary security, it already also satisfies the stronger notion of sender's search security. To show this, we implicitly rely on the random self-reducibility of the CDH problem.

Lemma 10.5. *The protocol in Figure 4 satisfies sender's search security based on the CDH assumption.*

Proof. Let there be an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ with

$$\Pr_{\mathsf{crs},\mathsf{r}}[\Pr[\mathsf{Exp}^{\mathsf{crs},\mathsf{r},0}_{\mathsf{sOTiOT}}(\mathcal{A}) = 1] > \epsilon \text{ and } \Pr[\mathsf{Exp}^{\mathsf{crs},\mathsf{r},1}_{\mathsf{sOTiOT}}(\mathcal{A}) = 1] > \epsilon] > \epsilon,$$

the we can construct an adversary $\mathcal{A}'$ that solves CDH at least with probability ϵ^3. $\mathcal{A}'$ receives a CDH challenge $\mathbb{G}, p, g, A_1, A_2$. It sets crs $X := A_1$, chooses random coins r and invokes $\mathcal{A}_1$ which outputs a state st and OT message $\mathsf{otr} = h_0$. $\mathcal{A}'$ samples $d_1, d_2 \leftarrow \mathbb{Z}_p$, defines $S_0 :=$

$A_2 \cdot g^{d_1}$, $S_1 := A_2 \cdot g^{d_2}$ and invokes for $i \in \{0,1\}$ $\mathcal{A}_2(\mathsf{st}, S_i, i)$ which outputs y_i. $\mathcal{A}'$ returns solution $(h_0^{d_1} \cdot y_1)/(h_0^{d_2} \cdot y_0 \cdot A_1^{d_2})$ to the CDH challenger.

With probability ϵ, crs X and random coins r are good, i.e. $\Pr[\mathsf{Exp}^{\mathsf{crs},\mathsf{r},0}_{\mathsf{sOTiOT}}(\mathcal{A}) = 1] > \epsilon$ and $\Pr[\mathsf{Exp}^{\mathsf{crs},\mathsf{r},1}_{\mathsf{sOTiOT}}(\mathcal{A}) = 1] > \epsilon$. We condition on that being the case. Since S_0 and S_1 are independent, it holds with probability ϵ^2 that $\mathcal{A}_2$ is successful for input $(\mathsf{st}, S_0, 0)$ and input $(\mathsf{st}, S_1, 1)$. Conditioned on that being the case, $y_0 = h_0^{s_0} = h_0^{a_2+d_1}$ and $y_1 = h_1^{s_1} = (h_0 \cdot A_1)^{d_2+a_2}$. Therefore it holds that the submitted CDH solution is

$$\frac{h_0^{d_1} \cdot y_1}{h_0^{d_2} \cdot y_0 \cdot A_1^{d_1}} = \frac{h_0^{d_1} \cdot (h_0 \cdot A_1)^{d_2+a_2}}{h_0^{d_2} \cdot h_0^{a_2+d_1} \cdot A_1^{d_2}} = A_1^{a_2}.$$

Hence, $\mathcal{A}'$ solves CDH with at least probability ϵ^3. □

10.2 Instantiation from LPN

We now give an instantiation of an elementary OT under the *learning parity with noise* (LPN) assumption with noise rate $\rho = n^{-\varepsilon}$ for $\varepsilon > \frac{1}{2}$. This protocol only achieves imperfect correctness, with an inverse-polynomial failure probability, but we argue that this is sufficient to get UC OT with negligible error probability.

Definition 10.6 (Learning Parity with Noise)**.** *For a uniform $s \in \mathbb{Z}_2^n$, oracle $\mathcal{O}_{\mathsf{LPN}}$ outputs samples of the form a, $z = as+e$, where $a \xleftarrow{\$} \mathbb{Z}_2^n$ and Bernoulli distributed noise term $e \xleftarrow{\$} \mathcal{B}_\rho$ for parameter ρ. Oracle $\mathcal{O}_{\mathsf{uniform}}$ outputs uniform samples $a, z \in \mathbb{Z}_2^n \times \mathbb{Z}_2$. We say Learning with Parity (LPN) for dimension n and noise distribution $\mathcal{B}_\rho$ is hard iff for any ppt adversary $\mathcal{A}$,*

$$|\Pr[\mathcal{A}^{O_{\mathsf{LPN}}}(1^n) = 1] - \Pr[\mathcal{A}^{O_{\mathsf{uniform}}}(1^n) = 1]| \leq \mathsf{negl}.$$

In the following, we will use a variant of LPN, where the secret is sampled from the noise distribution rather than the uniform distribution and the first sample is errorless. This variant is known to be as hard as standard LPN. The two following lemmata give a more precise relation between LPN and its above described variant.

Lemma 10.7 ([BLP+13], Lemma 4.3)**.** *There is an efficient reduction from LPN with dimension n and noise distribution $\mathcal{B}_\rho$ to LPN where the first sample is errorless with dimension $n-1$ and noise distribution $\mathcal{B}_\rho$ that reduces the advantage by at most probability 2^{-n}.*

Lemma 10.8 ([ACPS09] Adaptation of Lemma 2)**.** *LPN samples of the from $a, as+e$ with uniform $a, s \in \mathbb{Z}_2^n$ and $e \xleftarrow{\$} \mathcal{B}_\rho$ can be efficiently transformed into samples $a', a's' + e$, where $s' \xleftarrow{\$} \mathcal{B}_\rho^n$ and uniform $a' \in \mathbb{Z}_2^n$. This also holds when $e = 0$, i.e. first is errorless LPN. The same transformation maintains the uniformity of samples in $\mathbb{Z}_2^n \times \mathbb{Z}_2$.*

Proof Sketch. The transformation queries LPN samples $A, z_A = As+e_s$ until $A \in \mathbb{Z}_2^{n\times n}$ is invertible. Then, $A^{-1}, A^{-1}z_A = s + A^{-1}e_s$ will allow mapping LPN samples $a, z = as + e$ to samples with secret $s' = e_s$ by computing the new sample $a' = aA^{-1}$, $z + aA^{-1}z_A = a's' + e$. In the case where $e = 0$, i.e. an errorless LPN sample, the resulting sample will also be errorless. □

Lemma 10.9. *The protocol in Figure 5 satisfies receiver's indistinguishability security based on the LPN assumption with dimension n and noise distribution $\mathcal{B}_\rho$.*

$$
\begin{array}{lcl}
\text{Sender}(A, v): & \text{CRS: } (A, v) \in \mathbb{Z}_2^{n\times(n+1)} & \text{Receiver}(A, v, c): \\
 & & x, e \leftarrow \mathcal{B}_\rho^n \\
 & \xleftarrow{\quad \mathsf{otr} := h_0 \quad} & h_0 := Ax + e + cv \\
h_1 := h_0 + v & & \\
S, E \leftarrow \mathcal{B}_\rho^{\lambda\times n} & & \\
Z := SA + E & \xrightarrow{\quad \mathsf{ots} := Z \quad} & \\
\text{output } y_0 := Sh_0, y_1 := Sh_1 & & \text{output } y_c := Zx
\end{array}
$$

Figure 5: Elementary OT from LPN with imperfect correctness.

Proof. The receiver's bit c is masked by an LPN sample $Ax + e$. Therefore, distinguishing the case $c = 0$ versus $c = 1$ is equivalent to breaking LPN. $\square$

Lemma 10.10. *The protocol in Figure 5 satisfies sender's elementary OT security based on the LPN assumption with dimension $n-1$ and noise distribution $\mathcal{B}_\rho$.*

Proof. We use a hybrid version of first is errorless LPN with a secret sampled from the noise distribution which is hard based on standard LPN with the same noise distribution and dimension $n-1$, see Lemma 10.7 and Lemma 10.8. Hybrid LPN is as hard as standard LPN losing a factor $\frac{1}{\lambda}$ in the advantage.

Let there be a malicious receiver that outputs y_0, y_1 with probability $\epsilon > \mathsf{negl}$ then there is a LPN distinguisher $\mathcal{A}$ that breaks hybrid first is errorless LPN with advantage ϵ. $\mathcal{A}$ operates as follows. It receives a LPN challenge v, A, z_v, Z and sets CRS to A, v. After receiving h_0, it sends Z to the malicious receiver and obtains y_0, y_1. If $y_0 + y_1 = z_v$ it outputs 1 otherwise 0.

Let $Z = SA + E$, $z_v = Sv$, then $\mathcal{A}$ faithfully simulates the actual protocol. With probability ϵ, the malicious receiver will output $(y_0, y_1) = (Sh_0, Sh_1)$. In this case $y_0 + y_1 = Sv$ equals z_v and $\mathcal{A}$ will output 1. In the uniform case, i.e. Z_A and z_v are uniform, hence the malicious receiver can output y_0, y_1 such that $y_0 + y_1 = z_v$ at most with probability $2^{-\lambda}$. Hence $\mathcal{A}$ breaks LPN with advantage $\frac{\epsilon}{\lambda} - 2^{-\lambda} > \mathsf{negl}$. $\square$

Lemma 10.11 (Imperfect Correctness)**.** *Let a sender and a receiver interact in the protocol in Figure 5 with parameter $\rho \leq \frac{1}{n^\epsilon}$, for constant $1 > \epsilon > \frac{1}{2}$. Then with overwhelming probability $1 - \mathsf{negl}(\lambda)$ over the coins of the receiver (i.e., x, e) we have the following probability of correctness over the coins of the sender (i.e., S, E):*

$$\Pr_{S,E}[Sh_c = Zx] \geq 1 - 4\lambda n^{1-2\epsilon},$$

where $4\lambda n^{1-2\epsilon}$ can be an arbitrary $\frac{1}{\mathsf{poly}(\lambda)}$ for a suitable choice of $n = \mathsf{poly}(\lambda)$.

Proof. The protocol is correct iff the receivers output Zx matches the senders output Sh_c. By construction, $Zx = SAx + Ex$, whereas $Sh_c = SAx + Se$. Hence correctness holds when $Ex - Se = 0$.

By Chernoff,

$$\Pr[|x| > 2\rho n \vee |e| > 2\rho n] \leq 2e^{-\frac{\rho n}{3}},$$

which is negligible for $\epsilon < 1$. Given that $|x| \leq 2\rho n$, for all rows e_i of E, $e_i x$ is distributed as the sum of at most $2\rho n$ Bernoulli variables with parameter ρ. Hence, by a union bound over the $2\rho n$ variables

$\Pr_{e_i}[e_i x = 1] \leq 2\rho^2 n$. Using another union bound over all λ rows yields $\Pr_E[Ex \neq 0 \in \mathbb{Z}_2^\lambda] \leq 2\lambda\rho^2 n$. Because of symmetry,

$$\Pr_{E,S}[Ex - Se = 0] \geq 1 - 4\lambda\rho^2 n.$$

□

10.2.1 Dealing with Imperfect Correctness

The above gives us an elementary OT scheme with imperfect correctness under LPN: with overwhelming probability over the coins of the receiver, we have a a $1/p(\lambda)$ error-probability over the coins of the sender, where we can choose $p(\lambda)$ to be an arbitrary polynomial. For concreteness we set $p(\lambda) = \lambda^2$, so the error probability is $1/\lambda^2$. We outline how to leverage the series of generic transformations from the previous sections to get UC OT with a negligible correctness error. This requires only minor modifications throughout.

Elementary OT → Search OT (Theorem 5.2): This transformation performs a λ-wise parallel repetition on the sender message and therefore, by the union bound, increases the correctness error from $1/\lambda^2$ to $1/\lambda$. Security is unaffected.

Search OT → bit-iOT (Theorem 5.5): This transformation preserves the correctness error of $1/\lambda$. Security is unaffected.

bit-iOT → string iOT (Theorem 5.6): Here, we can modify the transformation slightly and first encode the strings using an error-correcting code and have the receiver apply error correction. Since each bit has an independent error probability of $1/\lambda$, we can set the parameters of the error-correcting code to get an exponentially small error probability, say $2^{-2\lambda}$. Security is unaffected by this modification.

Imperfect → Perfect Correctness: The above gives a scheme where, with overwhelming probability over the receiver's coins, we have a $2^{-2\lambda}$ error probability over the sender's coins. However, our definition of OT correctness in Section 4.1 requires a stronger notion of *perfect correctness*: with overwhelming over the receiver's coins and the CRS, *all* choices of the sender coins yield the correct output. This is needed in two places: (1) In the construction of 2-round ZK arguments (Theorem 8.4), we rely on extractable commitments, which in turn require a PKE with perfect correctness (Definition 3.3). Constructing PKE from OT requires the same perfect correctness for the OT. (2) In the construction of UC OT from Sender-UC OT and ZK (Theorem 9.2) we also need the underlying Sender-UC OT to have perfect correctness. This is because we rely on the fact that if a malicious sender computes the second-round OT message correctly with some choice of random coins (which he proves via the ZK argument), then the receiver gets the correct value.

We can generically achieve such perfect correctness, using an idea similar to the one behind Naor's commitments [Nao90]. We add an additional random value r^* to the CRS. The sender computes his second-round OT message by relying on a pseudorandom generator G and setting the random coins to be $G(s) \oplus r^*$ where s is small seed of length (e.g.,) λ. By a counting argument, with overwhelming probability over r^* and the receiver's random coins, there is no choice of the sender's coins s that results in an error. Security is preserved by relying on the security of the PRG.

Combining the above, the following theorem follows as a corollary.

Theorem 10.12. *Under the LPN assumption with noise rate $\rho = n^{-\varepsilon}$ for $\varepsilon > \frac{1}{2}$ there exists a 2-round UC OT.*

References

[ACPS09] Benny Applebaum, David Cash, Chris Peikert, and Amit Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In Shai Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 595–618, Santa Barbara, CA, USA, August 16–20, 2009. Springer, Heidelberg, Germany. 8, 39

[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, 3

[Ale03] Michael Alekhnovich. More on average case vs approximation complexity. In *44th FOCS*, pages 298–307, Cambridge, MA, USA, October 11–14, 2003. IEEE Computer Society Press. 3

[BD18] Zvika Brakerski and Nico Döttling. Two-message statistically sender-private OT from LWE. In *TCC 2018, Part II*, LNCS, pages 370–390. Springer, Heidelberg, Germany, March 2018. 2, 3

[BGI+17] Saikrishna Badrinarayanan, Sanjam Garg, Yuval Ishai, Amit Sahai, and Akshay Wadia. Two-message witness indistinguishability and secure computation in the plain model from new assumptions. In Tsuyoshi Takagi and Thomas Peyrin, editors, *ASIACRYPT 2017, Part III*, volume 10626 of *LNCS*, pages 275–303, Hong Kong, China, December 3–7, 2017. Springer, Heidelberg, Germany. 10

[BL18] Fabrice Benhamouda and Huijia Lin. k-round multiparty computation from k-round oblivious transfer via garbled interactive circuits. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part II*, volume 10821 of *LNCS*, pages 500–532, Tel Aviv, Israel, April 29 – May 3, 2018. Springer, Heidelberg, Germany. 2, 3

[BLP+13] Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. Classical hardness of learning with errors. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *45th ACM STOC*, pages 575–584, Palo Alto, CA, USA, June 1–4, 2013. ACM Press. 39

[BM90] Mihir Bellare and Silvio Micali. Non-interactive oblivious transfer and applications. In Gilles Brassard, editor, *CRYPTO'89*, volume 435 of *LNCS*, pages 547–557, Santa Barbara, CA, USA, August 20–24, 1990. Springer, Heidelberg, Germany. 2, 8

[Can01] Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *42nd FOCS*, pages 136–145, Las Vegas, NV, USA, October 14–17, 2001. IEEE Computer Society Press. 1, 10

[CCM98] Christian Cachin, Claude Crépeau, and Julien Marcil. Oblivious transfer with a memory-bounded receiver. In *39th FOCS*, pages 493–502, Palo Alto, CA, USA, November 8–11, 1998. IEEE Computer Society Press. 2

[CHS05] Ran Canetti, Shai Halevi, and Michael Steiner. Hardness amplification of weakly verifiable puzzles. In Joe Kilian, editor, *TCC 2005*, volume 3378 of *LNCS*, pages 17–33, Cambridge, MA, USA, February 10–12, 2005. Springer, Heidelberg, Germany. 13

[CLOS02] Ran Canetti, Yehuda Lindell, Rafail Ostrovsky, and Amit Sahai. Universally composable two-party and multi-party secure computation. In *34th ACM STOC*, pages 494–503, Montréal, Québec, Canada, May 19–21, 2002. ACM Press. 10

[CR03] Ran Canetti and Tal Rabin. Universal composition with joint state. In Dan Boneh, editor, *CRYPTO 2003*, volume 2729 of *LNCS*, pages 265–281, Santa Barbara, CA, USA, August 17–21, 2003. Springer, Heidelberg, Germany. 11

[DHRS04] Yan Zong Ding, Danny Harnik, Alon Rosen, and Ronen Shaltiel. Constant-round oblivious transfer in the bounded storage model. In Moni Naor, editor, *TCC 2004*, volume 2951 of *LNCS*, pages 446–472, Cambridge, MA, USA, February 19–21, 2004. Springer, Heidelberg, Germany. 2

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

[GL89] Oded Goldreich and Leonid A. Levin. A hard-core predicate for all one-way functions. In *21st ACM STOC*, pages 25–32, Seattle, WA, USA, May 15–17, 1989. ACM Press. 5, 15

[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

[GS18] Sanjam Garg and Akshayaram Srinivasan. Two-round multiparty secure computation from minimal assumptions. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part II*, volume 10821 of *LNCS*, pages 468–499, Tel Aviv, Israel, April 29 – May 3, 2018. Springer, Heidelberg, Germany. 2, 3

[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

[HL18] Justin Holmgren and Alex Lombardi. Cryptographic hashing from strong one-way functions (or: One-way product functions and their applications). In *59th FOCS*, pages 850–858. IEEE Computer Society Press, 2018. 7, 32, 34

[JKKR17] Abhishek Jain, Yael Tauman Kalai, Dakshita Khurana, and Ron Rothblum. Distinguisher-dependent simulation in two rounds and its applications. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part II*, volume 10402 of *LNCS*, pages 158–189, Santa Barbara, CA, USA, August 20–24, 2017. Springer, Heidelberg, Germany. 6

[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. 1

[Nao90] Moni Naor. Bit commitment using pseudo-randomness. In Gilles Brassard, editor, *CRYPTO'89*, volume 435 of *LNCS*, pages 128–136, Santa Barbara, CA, USA, August 20–24, 1990. Springer, Heidelberg, Germany. 41

[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, 3

[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, 3, 10

[Rab05] Michael O. Rabin. How to exchange secrets with oblivious transfer. Cryptology ePrint Archive, Report 2005/187, 2005. http://eprint.iacr.org/2005/187. 1

[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, 2