Two-Message Witness Indistinguishability and Secure Computation in the Plain Model from New Assumptions

Saikrishna Badrinarayanan * Sanjam Garg † Yuval Ishai ‡ Amit Sahai §
Akshay Wadia ¶

Abstract

We study the feasibility of two-message protocols for secure two-party computation in the plain model, for functionalities that deliver output to one party, with security against malicious parties. Since known impossibility results rule out polynomial-time simulation in this setting, we consider the common relaxation of allowing super-polynomial simulation.

We first address the case of zero-knowledge functionalities. We present a new construction of two-message zero-knowledge protocols with super-polynomial simulation from any (sub-exponentially hard) game-based two-message oblivious transfer protocol, which we call Weak OT. As a corollary, we get the first two-message WI arguments for NP from (sub-exponential) DDH. Prior to our work, such protocols could only be constructed from assumptions that are known to imply non-interactive zero-knowledge protocols (NIZK), which do not include DDH.

We then extend the above result to the case of general single-output functionalities, showing how to construct two-message secure computation protocols with quasi-polynomial simulation from Weak OT. This implies protocols based on sub-exponential variants of several standard assumptions, including Decisional Diffie Hellman (DDH), Quadratic Residuosity Assumption, and N^{th} Residuosity Assumption. Prior works on two-message protocols either relied on some trusted setup (such as a common reference string) or were restricted to special functionalities such as blind signatures. As a corollary, we get three-message protocols for two-output functionalities, which include coin-tossing as an interesting special case. For both types of functionalities, the number of messages (two or three) is optimal.

Finally, motivated by the above, we further study the Weak OT primitive. On the positive side, we show that Weak OT can be based on any semi-honest 2-message OT with a short second message. This simplifies a previous construction of Weak OT from the N^{th} Residuosity Assumption. We also present a construction of Weak OT from Witness Encryption (WE) and injective one-way functions, implying the first construction of two-message WI arguments from WE. On the negative side, we show that previous constructions of Weak OT do not satisfy simulation-based security even if the simulator can be computationally unbounded.

*University of California, Los Angeles. Email: saikrishna@cs.ucla.edu
†University of California, Berkeley. Email: sanjamg@berkeley.edu
‡University of California, Los Angeles and Technion. Email: yuvali@cs.technion.ac.il
§University of California, Los Angeles. Email: sahai@cs.ucla.edu
¶University of California, Los Angeles. Email: akshaywadia@gmail.com

Contents

1 Introduction **3**
- 1.1 Our Contribution . . . 4
- 1.2 Discussion and Related Work . . . 6
- 1.3 Technical Overview . . . 7

2 Preliminaries **10**

3 Difficulties in constructing Two Message Secure Computation protocols **13**

4 Zero-Knowledge from Weak OT **15**
- 4.1 Security Proof . . . 17
  - 4.1.1 Argument of Knowledge . . . 17
  - 4.1.2 Zero-Knowledge . . . 19

5 Two Message Secure Computation from Weak OT **21**
- 5.1 Security Proof . . . 23
  - 5.1.1 Malicious Sender . . . 23
  - 5.1.2 Malicious Receiver . . . 25

6 Weak OT from High Rate Semi-Honest OT **27**
- 6.1 Weak OT from High Rate Linear Homomorphic Encryption . . . 29
- 6.2 Weak OT from N^{th} Residuosity Assumption . . . 31

7 Weak OT from Witness Encryption **31**

8 Acknowledgements **32**

A Required Primitives **37**
- A.1 One-Way Functions. . . . 37
- A.2 Non-interactive Commitment Scheme. . . . 37
- A.3 Yao's garbled circuit. . . . 38
- A.4 Other Definitions . . . 39
- A.5 Witness Encryption . . . 39
- A.6 High Rate Oblivious Transfer . . . 40
- A.7 Linear Homomorphic Encryption . . . 40

1 Introduction

There has been a long line of work on minimizing the round complexity of protocols for secure two-party computation (see, e.g., [CCKM00, HK07, KO04, GMPP16, HV16] and references therein). In the present work we continue the study of this question, focusing on protocols in the "plain model," which do not rely on any form of set-up, and where security is based on standard cryptographic assumptions.

We will start by addressing the case of computing functions that depend on the inputs of the two parties and deliver an output to one party. (The general case will be discussed later.) For such single-output functions, it is clear that two messages are necessary: the first by the "receiver" who receives the output and the second by the "sender." The main question we ask is under what assumptions two messages are also sufficient. Two-message protocols, also referred to as "non-interactive secure computation" (NISC) protocols [IKO+11, AMPR14], have the qualitative advantage of allowing one party to go offline (after sending its message) while waiting for the other party to respond.

For security against semi-honest parties, the situation is well understood: such general two-message protocols exist if a two-message oblivious transfer (OT) protocol with security against semi-honest parties exists [Yao86, CCKM00]. This assumption is also necessary, since OT is a simple special case of general secure computation.

The situation is far more complex when considering security against malicious parties. For protocols with black-box simulation, four messages are necessary and are also sufficient under standard assumptions [GK96, KO04]. This can be improved to two messages by using standard setup assumptions such as a common reference string [CCKM00, HK07, IKO+11]. In the plain model, however, two-message protocols that satisfy the standard notion of security are known not to exist, even when allowing non-black-box simulation and even for the special of zero-knowledge [GO94, BLV03]. To get around this impossibility, Pass [Pas03] suggested considering simulation whose running time is super-polynomial, but not necessarily unbounded, and realized two-message zero-knowledge in this model. General secure computation with super-polynomial simulation was first studied by Prabhakaran and Sahai [PS04] and by Barak and Sahai [BS05] in the context of concurrent security (with protocols requiring multiple rounds of interaction).

Secure computation with super-polynomial simulation is motivated by the fact that it captures the desirable security goals for the typical case of computing "non-cryptographic" functions, where even an unbounded simulator does not get a meaningful advantage. Moreover, using complexity leveraging, such protocols can be "as good" as standard protocols even for computing cryptographic functions, provided that the level of security of the primitives or other protocols with which they interact is sufficient to protect against an adversary with the same running time as the simulator. See Section 1.2 for further details.

The above discussion motivates the following question:

Under what assumptions can we construct two-message secure computation protocols with super-polynomial simulation in the plain model?

A natural first step is to study the above question for the special case of zero-knowledge, which captures functions that take input from only one party. Zero-knowledge protocols with unbounded simulation are equivalent to *witness indistinguishable* (WI) protocols. Two-message WI protocols for NP (also called private coin ZAPs) can be constructed from non-interactive zero-knowledge (NIZK) protocols [DN07]. These were used in [Pas03] to obtain 2-message zero-knowledge arguments with quasi-polynomial simulation. They were further used in [GRS+11] to obtain two-

message blind signatures in the plain model, which can be viewed as another instance of general secure two-party computation.

While it is known that NIZK can be based on standard assumptions such as trapdoor permutations and bilinear maps [BFM88, FLS99, CHK03, GOS06b], there are several other well studied assumptions, such as the DDH assumption or even a strong assumption such as Witness Encryption [GGSW13], that are not known to imply NIZK or even 2-message WI arguments for NP. As far as we know, all non-trivial instances of 2-message protocols in the plain model appearing in the literature (even ones with unbounded simulation) require either NIZK or bilinear maps [GOS06a].

1.1 Our Contribution

We essentially settle the above question, showing that general two-message secure computation in the plain model with super-polynomial simulation is implied by any (sub-exponentially secure) "game-based" two-message OT protocol. Such a protocol is required to be secure with super-polynomial simulation against a malicious receiver, and is only required to satisfy indistinguishability-based security against the sender (i.e., the sender cannot distinguish between the two possible selection bits of the receiver). From here on, we refer to such an OT protocol as *Weak OT*. Weak OT protocols can be easily constructed from the DDH assumption [NP01, AIR01] (which is not known to imply NIZK) and are also known to follow from the Quadratic Residuosity Assumption and the N^{th} Residuosity Assumption (i.e., the security of the Paillier cryptosystem) [HK12].

The above result essentially settles our main question, since Weak OT can be viewed as the simplest analogue of two-message semi-honest OT for the case of security against malicious parties. As a corollary of our main result, Weak OT implies 3-message protocols with super-polynomial simulation in the plain model for functions that deliver outputs to both parties. This includes (multi-output) coin-tossing as an important special case. Motivated by the usefulness of Weak OT, we further study this primitive, obtaining several new positive and negative results.

We now give a more detailed account of our results.

1. We start by studying the Weak OT primitive described above (and formally defined in Section 2) and explore the feasibility of using it for secure computation with super-polynomial simulation. We show that Weak OT protocols may not even be secure with unbounded simulation. We demonstrate this by constructing a protocol (only a slight modification of the protocol in [NP01, AIR01]) that achieves the game based notion but suffers from a real attack. Concretely, we show a malicious sender strategy for this protocol such that even a single instance of execution of the protocol with the malicious sender would suffer from the attack. This is counter-intuitive because in a single instance of OT, any probabilistic mapping from the receiver's input to its output can be realized by a malicious sender in the ideal model, and so simulation seems easy. However, in our attack, the receiver's output becomes a value that *cannot be known* to the sender. This attack not only violates the intuitive notion of correctness and security, but it provably cannot be simulated even by an unbounded simulator. This impossibility result shows that proving security using a super-polynomial simulator, which is the setting in the rest of our work, is non-trivial and interesting.

2. Based on any (sub-exponentially secure) Weak OT, we construct a secure protocol for two-message zero knowledge argument of knowledge with quasi-polynomial simulation in the plain model. This implies the first such protocols, and even the first 2-message WI protocols, under assumptions that are not known to imply NIZK. More precisely, we prove the following:

   Theorem 1. *Assuming the existence of sub-exponentially secure Weak OT, there exist two-message zero knowledge arguments (with argument of knowledge) for NP in the plain model*

with quasi-polynomial simulation.

In particular, we get the following new corollary:

Theorem 2. *Two-message witness indistinguishable arguments for NP can be based on the sub-exponentially hard Decisional Diffie-Hellman Assumption.*

3. Using a variant of the "GMW paradigm" [GMW87], we extend the above result to the case of general secure computation. Concretely, we prove the following theorem:

   Theorem 3. *Two Message Secure Computation protocols with quasi-polynomial simulation in the plain model for general single-output functionalities can be based on any sub-exponentially secure Weak OT.*

   As a corollary, we get the first general 2-message protocols in the plain model.

   Corollary 1. *Two Message Secure Computation protocols with quasi-polynomial simulation for general single-output functionalities can be based on any of the following sub-exponentially hard assumptions: (1) Decisional Diffie-Hellman Assumption; (2) Quadratic Residuosity Assumption; (3) N^{th} Residuosity Assumption.*

   While such protocols are not very hard to obtain from previous two-message zero-knowledge protocols with super-polynomial simulation, we are not aware of such a result in the literature. Moreover, the DDH-based construction crucially depends on our new construction of 2-message zero-knowledge protocols.

   Secure two-message protocols for single-output functionalities imply secure three-message protocols for two-output functionalities. Concretely, we get the following corollary.

   Corollary 2. *Three-message secure protocols with quasi-polynomial simulation for general two-output functionalities (satisfying "security with abort") can be based on sub-exponentially secure Weak OT, and in particular on sub-exponential DDH.*

   A particularly useful special case is that of (multi-bit) coin-tossing where neither party has any input and both parties get a uniformly random string as output. Here quasi-polynomial simulation seems enough for all natural applications. Despite the large body of work on coin-tossing, we are not aware of any previous multi-bit coin-tossing protocol in the plain model that provides a meaningful notion of security (even a game-based notion) with only 3 messages under standard assumptions. Our coin-tossing result should be compared with the 5-message protocol from [KO04] (which is optimal for standard black-box simulation) and a recent 4-message protocol from [HV16] which is secure with inverse-polynomial simulation error.

4. To further expand the class of assumptions on which we can base our general protocols, we provide new constructions of Weak OT satisfying the game based notion.

   The first construction is based on any high rate semi-honest secure OT which in turn can be reduced to any high rate additively homomorphic encryption. Concretely, we need a semi-honest one-out-of-two string-OT protocol in which the output length on a pair of strings of length ℓ is smaller than $c\ell$ for some constant $c < 2$. As a corollary, by instantiating the high rate homomorphic encryption scheme using a construction of Damgård and Jurik

(DJ) [DJ01], we simplify the construction and analysis of Weak OT from the N^{th} Residuosity Assumption of Halevi and Kalai [HK12]. In particular, our construction only relies on the semantic security of the DJ cryptosystem and simple "syntactic" properties (homomorphism and ciphertext length) and does not involve smooth projective hash functions. This general new construction of Weak OT could potentially lead to basing our general protocols on other assumptions, such as lattice-based assumptions. The construction is presented in Section 6.

Our second construction of Weak OT builds on Witness Encryption (WE) [GGSW13] and any injective one way function. This is described in Section 7. As a corollary, all of the results discussed above can also be based on WE (and injective one-way functions).

At the heart of our two-message secure computation protocols is a two-message protocol for zero-knowledge from sub-exponential security of game based OT. Note that this is contrast to the construction of Pass [Pas03] who gave a construction based on NIZKs. Our alternative new construction avoids the use of NIZKs and is what enables our new results that provide constructions under alternative assumptions. This construction of zero-knowledge is provided in Section 4. The construction of two-message secure computation using this zero-knowledge protocol is provided in Section 5.

1.2 Discussion and Related Work

In this section we discuss the two key features of our protocols: super-polynomial simulation and security in the plain model, and survey some related work.

What good is super-polynomial simulation? Intuitively speaking, the notion of super-polynomial simulation (SPS) guarantees that the real world adversary does not learn anything more than an ideal world adversary running in super-polynomial time. So, what does the SPS ideal world adversary learn? For information theoretic functionalities (example, Yao's millionaire problem), the running time of the ideal-world simulator does not affect security in any sense. In particular the computational power awarded to the ideal world adversary is useless for learning anything about the input of the honest party. It does not rule out the possibility that the adversary learns some super-polynomial function of its view but this is irrelevant for the task at hand. On the other hand, for cryptographic functionalities, the adversary's ability to run in super-polynomial time is indeed problematic as it could potentially harm the security of the functionality itself. However, at an often small cost to efficiency, it is almost always possible to choose a higher security parameter for the cryptographic operations performed by the functionality such that meaningful security can be obtained (see e.g. [GRS+11] for the example of blind signatures). SPS is commonly used in cryptography. In fact, any zero knowledge protocol with super polynomial simulation is a witness indistinguishable protocol.

Relation to concurrently secure computation. The notion of concurrently secure super-polynomial simulation [PS04, BS05, GGJS12] and its variants [CLP10] have been extensively studied in the literature. This notion is known to be impossible [Lin04, BPS06, Goy12, AGJ+12, GKOV12] to achieve with polynomial-time simulation. The notion of two-message secure computation that we study implies the notion of concurrently secure computation, in the restricted setting where the adversary is allowed to play as a sender or as a receiver across all concurrent sessions (the so-called "fixed-roles" setting). This improves on the round complexity of known solutions.

Recently, Dottling et al. [DFKS16] constructed two round two-party computation protocols for certain functionalities that is secure against semi-honest senders and malicious receivers. However,

they consider a game-based security notion against a malicious receiver and this is incomparable to our setting.

Concurrent and subsequent work. Concurrent to our work, Jain et al. [JKKR17] construct protocols that are similar to our two-round protocols. While their focus is on polynomial time distinguisher-dependent simulation, we focus on super-polynomial simulation. Therefore, the only result in common between the two papers is two-round witness indistinguishability for NP from Weak OT. Our proof of WI is significantly simpler than theirs, because our analysis is via super-polynomial simulation. Our paper also contains additional results on Weak OT (both negative and positive) that simplify previous constructions and extend the set of assumptions on which both our and their round-optimal protocols can be based.

Subsequent to our work, Khurana and Sahai [KS17] use our two-message secure computation protocol crucially to build two-message non-malleable commitments with respect to commitment from sub-exponentially hard DDH.

Even though we have a straight line simulation, our protocol doesnt extend to the UC/concurrent setting because that requires non-malleability. A very recent follow-up work by Badrinarayanan et al. [BGJ+17] achieves concurrent security in the MPC setting by building on our techniques (along with other techniques), using 3 rounds of simultaneous message exchange. Note that, in contrast, our protocols use only 2 rounds of unidirectional message exchange.

1.3 Technical Overview

The new 2-round SPS-Zero Knowledge protocol from 2-round Weak OT. The technical heart of our result is a new 2-round super-polynomial simulation secure zero knowledge protocol(SPS-ZK) from a 2-round weak OT. The weak OT protocol we use has statistical sender's security but only T-chooser's security. That is, the receiver's choice bit is secure against all adversaries running in time $T.\text{poly}(\lambda)$. Additionally, we will also use a T-time extractable commitment protocol. To ease the exposition, let's allow the simulator to run in exponential time. Then, by running the protocol with an appropriately smaller security parameter, we can rely on just quasi-polynomial simulation.

The main idea behind the new zero knowledge protocol is to "squash" a parallelized version of Blum's 3-round zero knowledge protocol for Hamiltonicity, by making use of the 2-round Weak OT protocol. Our technique applies more generally to parallelized Σ-protocols with "special soundness", but here we will focus on Blum's protocol for clarity. Recall that in Blum's protocol, the prover generates an initial message α, and prepares two responses γ_0, γ_1. The verifier then sends a random bit $\beta \in \{0, 1\}$, and the prover responds with γ_β.

To squash this protocol to two rounds, we first have the verifier choose β at the start, and then use β as its input in the role of receiver in the Weak OT protocol. Note that this intuitively keeps β hidden from the prover. Then, the prover sends α separately as part of its message, but also uses γ_0 and γ_1 as its inputs in the role of sender in the Weak OT protocol. Thus, the verifier learns only α and γ_β and can then verify the original Blum proof. This protocol can be repeated in parallel to boost soundness. We will now discuss how to establish SPS zero knowledge and computational soundness separately.

Zero Knowledge: No rewinding allowed. First, observe that we can't directly use the same proof strategy as in Blum's protocol as we can not rewind the adversary here. In our protocol, since the verifier sends just one message, if we try rewinding the malicious verifier, it could just keep sending the same message over and over. Thus, there is nothing to be gained from rewinding.

To establish zero-knowledge, we will use complexity leveraging to construct a super-polynomial simulator running in time $T_1 \cdot \text{poly}(\lambda)$, where $T_1 > T$, that can extract β from the verifier's first message in the Weak OT protocol. Now that the simulator knows β, simulation at first glance appears to be straightforward, since it needs to place a correct value only in γ_β. This can be done by just invoking the zero knowledge simulator of Blum's protocol. However, there is a subtle flaw in this argument due to the Weak OT protocol, as we discuss now in further detail.

Before we can see the flaw, we have to briefly discuss soundness. In order for soundness to hold, we need that the prover cannot somehow "malleate" the verifier's first OT message into a successful second OT response by the prover. The way we will achieve such non-malleability is by adding a weak commitment that can be extracted in time $T < T_1$. Recall that it is impossible for an adversary to take as input a T_1-strong commitment $C_{T_1}(m)$, and produce a T-weak commitment $C_T(m')$ in a way that causes m' to depend on m. This is easy to see: An adversary running in time T can anyway break $C_T(m')$ and recover m'. It could then use m' to predict m, thereby breaking the T_1-secure commitment $C_{T_1}(m)$ – a contradiction to the stronger security of C_{T_1} since $T_1 > T$.

In the case of zero knowledge, recall that our simulator runs in time $T_1 \cdot \text{poly}(\lambda)$, where $T_1 > T$. Note that the OT protocol does not have receiver's security against adversaries running in time $T_1 \cdot \text{poly}(\lambda)$, since it needs to extract β from the first message of the weak OT. But then, it can anyway break the commitment scheme since $T_1 > T$. Therefore, now, in order for the commitment to be stronger than the OT, we need $T_1 < T$, whereas for proving soundness, we require that $T_1 > T$. (Since we require that the time taken to break the commitment is lesser than the time taken to break the chooser's security in the OT protocol.) This a fundamental contradiction that suggests that perhaps our goal is impossible to achieve!

We fix this by exploiting the special structure of our protocol: Recall our observation that a cheating verifier, which is without loss of generality deterministic, if rewound, would just keep sending the same first message. Now, we want to exploit this fact to keep $T_1 > T$ as needed by soundness, and argue zero knowledge in a different way: The simulation strategy itself is the same as before. That is, the simulator runs in time $T_1 \cdot \text{poly}(\lambda)$ and extracts β from the first message of the weak OT. It then invokes the simulator of Blum's protocol and produces the prover's message. This second phase runs in polynomial time. Now, let's consider the reduction that breaks the commitment scheme by interacting with the malicious PPT verifier. The reduction, given the commitment to be broken as an external challenge, includes it as part of the prover's message (more specifically, includes it as a commitment to $\gamma_{1-\beta}$ in the string α). Now, based on the PPT verifier's guess it breaks the commitment. The only stage in the reduction that runs in super-polynomial time is when it breaks the initial message of the verifier to extract β. Therefore, let's consider the malicious verifier with the "best possible" initial message and fix this message. The value β^* extracted from this can just be given as auxiliary input (non-uniform advice) to the reduction! So, now, the reduction is a non-uniform PPT machine. Therefore, if the PPT reduction can now break the commitment scheme, we will achieve a contradiction. Note that the auxiliary input is also given to the external challenger of the commitment scheme.

Soundness. To establish soundness of the protocol, we in fact prove a stronger property: that our protocol is an argument of knowledge. (We will anyway need this later when we construct the two message secure computation protocol for any general functionality). We will construct an extractor, that, running in super-polynomial time $T \cdot \text{poly}(\lambda)$, can extract out γ_0 and γ_1 from the prover's initial message by running the commitment extractor. Blum's protocol is designed so that α, γ_0, and γ_1 together yield knowledge of the Hamiltonian cycle in the original graph and hence the extractor learns the witness. We will then show that if the extractor fails, but the malicious prover succeeds in giving a correct proof, we can use this prover to break the T-chooser's security of the OT protocol by using the external OT challenge in the verifier's first message against this

malicious prover. Several challenges arise when trying to establish soundness. We discuss them now.

Recall that the aim is to show that if the malicious prover succeeds in giving a valid proof but the extractor fails, then the reduction will break the T-chooser's security of the OT protocol. Note that the reduction can run in time $T.\text{poly}(\lambda)$. The idea here was that the reduction interacts with the malicious prover and embeds the external OT challenge (of the OT receiver) in one of the indices as part of the verifier's first message. After checking that the proof is valid, the reduction can extract both γ_0 and γ_1 from α by running the T-time commitment extractor and then run the $\mathsf{BlumExt}$ to obtain the choice bit of the OT challenge. However, there is a subtle issue here that in order to check that the proof is valid, the reduction needs to run the third stage OT algorithm to recover γ_β. But, since it did not generate the first OT message, the reduction does not have the associated state that was used in that generation and hence cannot validate the proof (the state output by the first OT algorithm will be needed as input for the third stage).

We fix this by using a simple combinatorial argument. We consider a new verifier strategy where the verifier checks the proof at all indices except one and this choice is not revealed to the prover. It can be easily seen that the success probability of the malicious prover is as much, if not more, against this new verifier as well. Now, the reduction no longer needs to verify the proof at the index where the OT challenge was embedded. Also, if the malicious prover has to produce a valid proof, with probability close to 1, it still needs to produce a valid proof at every index since it can guess the missed out index with very small probability. Therefore, the Blum extraction would still work correctly on the embedded index and the reduction can break the OT receiver's security.

Two message secure computation. Given any weak OT protocol and the two message secure zero knowledge protocol from above, we compile them together using Yao's garbled circuits construction to produce a two message secure computation protocol for any general functionality. In fact, we don't need the full power of the zero knowledge protocol from above. In this construction, we will only need the weaker notion of witness indistinguishability(WI) which is anyway implied by SPS zero knowledge.

Consider a sender with input x and a receiver with input y and let the function they're computing be f. In the first round, the receiver, using each bit of his input, computes the first message of the weak OT protocol and sends this across. In addition, he also initiates a WI protocol with the sender and sends the first message of the verifier. Finally, he also sends the output of a one way function OWF that is not invertible in time $T.\text{poly}(\lambda)$ (but is invertible in time $T_1.\text{poly}(\lambda)$ where $T_1 > T$). Looking ahead, this value will help the simulator against a cheating receiver to generate a proof using the trapdoor statement. In response, the sender computes a garbled circuit that has his input hardwired into it and then runs the OT algorithm using the garbled keys as his input to the OT. Also, he computes a commitment c_1 to his input and another commitment c_2 to 0 which will prove to be useful for the simulator. He then computes a WI proof that he computed the commitment c_1 correctly, ran the OT algorithm correctly and computed the garbled circuit correctly. It is easy to see that the receiver, after checking the validity of the proof, can recover the garbled keys corresponding to his input using the OT and evaluate the garbled circuit to obtain the output of the function. The trapdoor statement in the WI proof will basically say that the prover knows the pre-image to the output of the OWF and the commitment c_2 is a commitment to this pre-image. Notice that we don't need the full expressiveness of the zero knowledge property. It is enough to have just witness indistinguishability and the simulator against a malicious receiver, just extracts the pre-image of the one-way function OWF and uses the trapdoor statement to prove that the pre-image is correct.

Similar to the proof of the zero knowledge protocol, the key tool in order to prove security is complexity leveraging. The main obstacle we face is very similar to the one faced in the case of

the zero knowledge protocol. In particular, for proving security against a malicious receiver, we will need to break the chooser's security of the OT protocol and then reduce the security of our protocol to the hiding of the commitment scheme. Therefore, we will need $T_1 < T$. However, to prove security against a malicious sender, we will require that $T < T_1$, following a similar argument as in the case of the soundness of the zero knowledge protocol. As in the case of our zero knowledge protocol, we fix this issue by considering an intermediate hybrid where non-uniform advice can provide key information embedded in the malicious receiver's fixed first message. This advice allows us to consider experiments that do not incur the running time needed to actually extract the information that was present in the first message of the receiver.

2 Preliminaries

Let λ denote the security parameter. We say that a function is *negligible* in the security parameter λ, if it is asymptotically smaller than the inverse of any fixed polynomial. Otherwise, the function is said to be *non-negligible* in λ. We say that an event happens with *overwhelming* probability if it happens with a probability $p(\lambda) = 1 - \nu(\lambda)$ where $\nu(\lambda)$ is a negligible function of λ. In this section, we define the primitives studied in this paper. We will start by defining a weaker indistinguishability based notion for oblivious transfer and then subsequently describe the simulation based notion for general functionalities.

We write $y = A(x; r)$ when the algorithm A on input x and randomness r, outputs y. We write $y \leftarrow A(x)$ for the process of picking r at random and setting $y = A(x; r)$. We also write $y \leftarrow S$ for sampling y uniformly at random from the set S. Some more primitives are defined in Appendix A.

Weak OT. In this paper, we consider a 1-out-of-2 *Oblivious Transfer* protocol (similar to [CCM98, NP01, AIR01, DHRS04, HK12]) where one party, the *sender*, has input composed of two strings (M_0, M_1) and the input of the second party, the *chooser*, is a bit c. The chooser should learn M_c and nothing regarding M_{1-c} while the sender should gain no information about c. We give a definition for the setting where the sender is protected information theoretically while the chooser is protected only computationally.

Definition 1 (Weak OT)**.** *The chooser runs the algorithm* OT_1 *which takes* 1^λ *and a choice bit* $c \in \{0,1\}$ *as input and outputs* $(\mathsf{ot}_1, state)$*. Chooser then sends* ot_1 *to the sender, who obtains* ot_2 *by evaluating* $\mathsf{OT}_2(1^\lambda, \mathsf{ot}_1, M_0, M_1)$*, where* M_0 *and* M_1 *(such that* $M_0, M_1 \in \{0,1\}^\lambda$*) are its inputs. The sender then sends* ot_2 *to the chooser who obtains* M_c *by evaluating* $\mathsf{OT}_3(1^\lambda, \mathsf{ot}_2, state)$*.*

- ***Perfect correctness.*** *For every choice bit* $c \in \{0,1\}$ *of the chooser and input messages* M_0 *and* M_1 *of the sender we require that, if* $(\mathsf{ot}_1, state) \leftarrow \mathsf{OT}_1(1^\lambda, c)$*,* $\mathsf{ot}_2 \leftarrow \mathsf{OT}_2(1^\lambda, \mathsf{ot}_1, M_0, M_1)$*, then* $\mathsf{OT}_3(1^\lambda, \mathsf{ot}_2, state) = M_c$ *with probability* 1*. We speak of statistical correctness if this probability is overwhelming in* λ*.*

- ***Chooser's security.*** *We require that for every non-uniform polynomial-time adversary* $\mathcal{A}$*,* $|\Pr[\mathcal{A}(\mathsf{OT}_1(1^\lambda, 0)) = 1] - \Pr[\mathcal{A}(\mathsf{OT}_1(1^\lambda, 1)) = 1]|$ *is negligible in* λ*.*

  We speak of T*-chooser's security if the above condition holds against all non-uniform adversaries* $\mathcal{A}$ *running in time* $T \cdot \mathsf{poly}(\lambda)$*.*

- ***Statistical sender's security.*** *We define an unbounded* [1] *time extractor* OTExt *such that* OTExt *on any input* ot_1 *outputs* 0 *if there exists some random coins such that* $\mathsf{OT}_1(1^\lambda, 0)$ *outputs* ot_1*, and* 1 *otherwise.*

[1] Note that fixing the parameters of the scheme, we can bound the running time of the extractor by some sub-exponential function but we avoid it to keep notation simple and avoid unnecessary parameters.

Then for any value of ot_1*, and any* K_0, K_1, L_0, L_1 *with* $K_{\mathsf{OTExt}(\mathsf{ot}_1)} = L_{\mathsf{OTExt}(\mathsf{ot}_1)}$*, we have that* $\mathsf{OT}_2(1^\lambda, \mathsf{ot}_1, K_0, K_1)$ *and* $\mathsf{OT}_2(1^\lambda, \mathsf{ot}_1, L_0, L_1)$ *are statistically indistinguishable. We speak of computational sender's security if for all non-uniform polynomial time adversaries* $\mathcal{A}$ *we have that* $|\Pr[\mathcal{A}(\mathsf{OT}_2(1^\lambda, \mathsf{ot}_1, K_0, K_1)) = 1] - \Pr[\mathcal{A}(\mathsf{OT}_2(1^\lambda, \mathsf{ot}_1, L_0, L_1)) = 1]|$ *is negligible in* λ*.*

T-secure Weak OT. Finally, we define T-secure Weak OT to be a Weak OT protocol with T-chooser's security. Note that we can claim that any Weak OT protocol with chooser's security based on a set of assumptions Υ, is also a T-secure Weak OT protocol if each assumption in Υ is additionally assumed to be secure against all non-uniform adversaries running in time $T \cdot \mathsf{poly}(\lambda)$. Note that this additionally relies on the fact that the security reduction for proving chooser's security of the underlying protocol is tight up to a multiplicative polynomial factor, in the security parameter.

Naor-Pinkas and Aiello et al. [NP01, AIR01] provided a construction of a Weak OT protocol based on the Decisional Diffie-Hellman assumption. Subsequently, Halevi and Kalai [HK12] provided an instantiation based on any smooth projective hash function. Further, note that the above definition is not a simulation-based definition but rather an indistinguishability-based one. Although it is a meaningful notion and is sufficient for some applications, it is still weaker than the simulation-based (described next) notion.

Two Message Secure Computation via super-polynomial simulation. The simulation-based definition compares the "real world," where the parties (the sender and the receiver) execute the protocol, to an "ideal world," where no message is exchanged between the parties; rather, there is a trusted party that takes an input from both parties, computes the output of the functionality on these inputs, and sends the corresponding output to each party. Loosely speaking, the simulation (resp., super-polynomial simulation)-based definition asserts that for every efficient adversary $\mathcal{A}$ (controlling either the sender or the receiver) in the real world there exists an efficient (resp., super-polynomial) simulator $\mathcal{S}$, controlling the same party in the "ideal world," so that the outputs of the parties in the ideal world are computationally indistinguishable from their outputs in the real world. In particular, the simulator $\mathcal{S}$ needs to simulate the view of the adversary $\mathcal{A}$ in a computationally indistinguishable manner.

Next, we formally define a Two Message Secure Computation protocol $\langle S, R \rangle$, between a *sender* S with input x and a *receiver* R with input y. The receiver should learn $f(x, y)$ and nothing else[2] while the sender should gain no information about y. More formally we will define this notion by comparing a two-round realization in the real-world with an ideal world scenario.

Real World. A Two Message Secure Computation protocol $\langle S, R \rangle$ is defined by three probabilistic algorithms $(\mathsf{NISC}_1, \mathsf{NISC}_2, \mathsf{NISC}_3)$ as follows. The receiver runs the algorithm NISC_1 which takes the receiver's input $y \in \{0,1\}^\lambda$ as input and outputs $(\mathsf{nisc}_1, \mathit{state})$. The receiver then sends nisc_1 to the sender, who obtains nisc_2 by evaluating $\mathsf{NISC}_2(\mathsf{nisc}_1, x)$, where $x \in \{0,1\}^\lambda$ is the sender's input.[3] The sender then sends nisc_2 to the receiver who obtains $f(x, y)$ by evaluating $\mathsf{NISC}_3(\mathsf{nisc}_2, \mathit{state})$.

At the onset of the computation the *real world adversary* $\mathcal{A}$ corrupting either the sender S or the receiver R, receives some auxiliary information z. Next, the computation proceeds as described above where the honest party sends messages as prescribed by the protocol and the adversary $\mathcal{A}$ sends arbitrary messages on behalf on the corrupted party. At the end of the computation the uncorrupted party outputs whatever is specified in the protocol. The corrupted

[2] Unlike Weak OT in which the sender is protected information theoretically this notion will provide only computational security for the sender.

[3] For simplicity of notation we denote the lengths of the inputs of the sender and the receiver by λ. In general they could be arbitrary polynomials in the security parameter λ.

party outputs any arbitrary PPT function of the view of $\mathcal{A}$. The overall output of the real-world experiment consists of all the values output by all parties at the end of the protocol, and this random variable is denoted by $\mathsf{REAL}_{\mathcal{A}}^{\langle S,R\rangle}(1^k, x, y, z)$. Let $\mathsf{REAL}_{\mathcal{A}}^{\langle S,R\rangle}$ denote the ensemble $\{\mathsf{REAL}_{\mathcal{A}}^{\langle S,R\rangle}(1^k, x, y, z)\}_{k\in\mathbb{N},x,y\in\{0,1\}^\lambda,z\in\{0,1\}^*}$.

Ideal World. In the ideal world experiment, the sender S and the receiver R interact with a *trusted party* for computing a function $f : \{0,1\}^\lambda \times \{0,1\}^\lambda \to \{0,1\}^\lambda$. The ideal world computation in presence of the *ideal world adversary* $\mathcal{S}$ corrupting either the sender S or the receiver R, and an (incorruptible) *trusted party* $\mathcal{F}$, proceeds as follows. First, as in the real-life model, $\mathcal{S}$ gets auxiliary information z. Next, the ideal world adversary $\mathcal{S}$ generates any arbitrary input on behalf of the corrupted party, which it sends to the trusted party $\mathcal{F}$. The honest party sends its input to the trusted party $\mathcal{F}$. At this point the ideal functionality evaluates the output and sends it to the receiver. The honest receiver outputs this value. The adversarial receiver $\mathcal{S}$ outputs an arbitrary value. Note that $\mathcal{S}$ is allowed to run in super-polynomial time. In this work, we will focus by default on simulators running in quasi-polynomial time - i.e $n^{\mathsf{poly}(\log(n))}$ where n is the security parameter. (See Definition 6 in [Pas03] for a definition of quasi-polynomial simulation in the context of zero-knowledge protocols.)

The overall output of the ideal process consists of all the values that are output by all parties at the end of the protocol. Let this random variable be denoted by $\mathsf{IDEAL}_{\mathcal{S}}^{\mathcal{F}}(1^k, x, y, z)$ and let $\mathsf{IDEAL}_{\mathcal{S}}^{\mathcal{F}}$ denote the ensemble $\{\mathsf{IDEAL}_{\mathcal{S}}^{\mathcal{F}}(1^k, x, y, z)\}_{k\in\mathbb{N},x,y\in\{0,1\}^\lambda,z\in\{0,1\}^*}$.

Equivalence of Computations. Informally, we require that executing a protocol $\langle S, R\rangle$ in the real world roughly emulates the ideal process for evaluating f.

Definition 2. *Let f be any polynomial time computable function on two inputs and let $\langle S, R\rangle$ be a protocol between a sender S and a receiver R. We say that $\langle S, R\rangle$* two-message securely evaluates *f if for every PPT real world adversary $\mathcal{A}$ there exists an ideal world adversary $\mathcal{S}$, such that $\mathsf{REAL}_{\mathcal{A}}^{\langle S,R\rangle} \overset{c}{\approx} \mathsf{IDEAL}_{\mathcal{S}}^{\mathcal{F}}$.*

Stricter Simulation. As described above the ideal world adversary is allowed to execute in super-polynomial time. We will consider a stricter notion of simulation under which a simulator is allowed to execute in super-polynomial time prior to its interaction with the ideal functionality. The simulator is subsequently restricted to be polynomial time. More formally, the *ideal world adversary* is modeled by two machines $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2)$. First, as in the real-life model, $\mathcal{S}_1$ gets auxiliary information z. Next, the ideal world adversary $\mathcal{S}_1$ gets to execute for a super-polynomial amount of time, and generates any arbitrary input on behalf of the corrupted party, which it sends to the trusted party $\mathcal{F}$, and a state of polynomial length. The honest party sends its input to the trusted party $\mathcal{F}$. At this point the ideal functionality evaluates the output and sends it to the receiver. The honest receiver outputs this value. The adversarial receiver $\mathcal{S}_2$ outputs an arbitrary PPT function of the received value and state. All the results in this paper obtain this stricter notion described above, and throughout this paper we will use the term Two Message Secure Computation to always imply this stricter notion.

Now, we define the notion of Two Message Secure Computation for two specific functionalities, namely, zero-knowledge and Parallel OT.

Zero-Knowledge. A Two Message Secure Computation for the zero-knowledge functionality can be described by three PPT algorithms $(\mathsf{ZK}_1, \mathsf{ZK}_2, \mathsf{ZK}_3)$. Consider a prover, who holds a witness w for some statement $x \in L$ and wants to convince a verifier about it. The verifier first computes $(\mathsf{zk}_1, \mathsf{zkst}) \leftarrow \mathsf{ZK}_1(1^\lambda)$ and sends zk_1 to the prover (zkst is the verifier's state). Then, the prover

computes $\mathsf{zk}_2 \leftarrow \mathsf{ZK}_2(1^\lambda, \mathsf{zk}_1, x, w)$ and sends zk_2 to the verifier. Finally, the verifier accepts if the the computation $\mathsf{ZK}_3(1^\lambda, \mathsf{zk}_2, x, \mathsf{zkst})$ outputs 1. We require that the first message of the verifier depends just on the size of the theorem being proven[4] and not on the theorem itself. This allows the prover to choose its theorem after receiving the first message of the verifier.

We stress that while a witness indistinguishable argument scheme is already a zero knowledge scheme with *unbounded* simulation, it is not necessarily a zero knowledge scheme with a super-polynomial time simulator with running time bounded by a quasi-polynomial function of the security parameter. Thus, zero knowledge with quasi-polynomial simulation is a strictly more powerful notion than witness indistinguishability. This follows immediately from the fact that the notion of witness indistinguishability is meaningless in the setting of unique witness, but the notion of quasi-polynomial simulation is still meaningful.

Parallel OT. A Two Message Secure Computation protocol for the functionality that performs k-parallel invocations of oblivious transfer where $k = \mathsf{poly}(\lambda)$ will be denoted by Parallel OT throughout this paper.

3 Difficulties in constructing Two Message Secure Computation protocols

Goldreich and Oren [GO94] showed that it is impossible to construct 2-round zero-knowledge arguments for languages outside BPP. As explained in [DJKL12], this result extends in a straightforward manner to show the impossibility of constructing 2-round T-zero-knowledge[5] arguments for T-hard languages, that are sound against cheating provers running in time $T \cdot \mathsf{poly}(\lambda)$. More recent works [DJKL12, CLMP12] gave a black-box impossibility result ruling out 2-round zero-knowledge sound against polynomial-time cheating provers (based on T-hard falsifiable assumptions). Note that since Two Message Secure Computation for OT implies 2-round zero-knowledge arguments we can obtain analogous impossibility results for Two Message Secure Computation for OT. It is also interesting to note that the positive result for Two Message Secure Computation for OT obtained in this paper assume that the underlying assumption is T'-hard for T' that is strictly more than the running time of the distinguisher. Thus, these results are essentially tight.

The aforementioned impossibility result only rules out black-box reductions to falsifiable assumptions. As a starting point, based on the premise that known instances of Weak OT protocols such as [NP01, AIR01, HK12] are not known to be susceptible to any attacks in the simulation setting, one may conjecture that that in fact all Weak OT protocols can be proven secure under a simulation based definition when unbounded simulation is allowed and we are willing to make strong (possibly non-falsifiable) assumptions.

In fact, it is argued in [HK12] (Section 3) that any Weak OT protocol provides simulation-based security in the standard sense for the case of a malicious sender, assuming that the simulator is allowed to reset the sender. This is argued as follows. The simulator (who does not know the choice bit of the actual receiver) simulates the (honest) receiver first with choice bit $b = 0$, and then it resets the sender and simulates the honest receiver with choice bit $b = 1$. This way the simulator extracts both messages M_0 and M_1 from the corrupted sender. It then gives (M_0, M_1) to the trusted party. Then the simulator uses the view of the cheating sender in the first execution (with the choice-bit $b = 0$). They argue that this view is indistinguishable from the "real world"

[4] We can write ZK_1 to take an additional input specifying the size of the theorem proven. To keep notation simple we avoid it and upper bound the size of the theorem proven by λ.

[5] Recall that in T-zero-knowledge protocols the simulator and the distinguisher run in time $T \cdot \mathsf{poly}(\lambda)$.

view based on the receiver's security of the Weak OT protocol, and from the fact that the sender does not receive any output from the trusted party. On the other hand, they argue that Weak OT does not give standard simulation-based guarantee in the case that the receiver is corrupted, because a malicious receiver is not guaranteed to "know its own choice bit b," and therefore the simulator does not know which input bit to send to the trusted party. However, they point out that this does guarantee an exponential time simulation of the receiver's view of the interaction.

Note that the argument from [HK12] is only applicable when the view of a cheating sender or the view of a cheating receiver is considered by itself. We show that if (as per standard definitions) joint distributions of the outputs of both parties are considered, then proving simulation based security for a Weak OT protocol even when unbounded simulation is allowed is very problematic. We demonstrate this by constructing a protocol that can be proved to be Weak OT under reasonable assumptions but suffers for a real attack under a simulation based definition. In particular, we show a malicious sender strategy such that even a single instance of execution of the protocol with a malicious sender can not be simulated by any unbounded simulator.

The protocol that we construct is only a slight modification of known protocols [NP01, AIR01] and highlights at the very least, the obstacles that we face even in proving security of specific protocols. We start by recalling the ElGamal encryption scheme abstractly. Let G be a multiplicative subgroup of Z_q^* of order p, where p and q are primes and p is of length λ that divides $q-1$. Let g be the generator of this group G. The public key for ElGamal encryption is generated by sampling $x \leftarrow Z_p^*$ and setting the public key to be (p, q, g, h) where $h = g^x$. The encryption procedure $Enc((p, q, g, h), m)$ is defined follows: Choose $r \in \mathbb{Z}_p^*$ and output $(g^r, m \cdot h^r)$. The decryption procedure $Dec((u, v), x)$ outputs $\frac{v}{u^x}$. Let $e(\cdot)$ be some invertible encoding function mapping Z_p to G. Then *circular security* of ElGamal implies that the encryption scheme remains semantically secure even when an encryption of $e(x)$ is given to the adversary. In particular semantic security is preserved when $Enc((p, q, g, h), e(x))$ is included in the public key. As pointed out in [BHHO08], it is unlikely that it would follow from the DDH assumption.

Lemma 1. *Assuming that ElGamal is circularly secure, there exists a Weak OT protocol and a real world cheating sender S^* strategy for this protocol such that it can not be simulated by any (unbounded) ideal world simulator.*

Proof. We will start by giving the protocol. The protocol used in our counter example is very similar to the DDH based Weak OT protocols from [NP01, AIR01]. The only difference being that our protocol includes an encryption E of $e(a)$ along with its first message. This value is not used by the protocol itself but however will be useful for the malicious sender that we will construct.

1. $(\mathsf{ot}_1, state) \leftarrow \mathsf{OT}_1(c)$: Sample $a, b \leftarrow \mathbb{Z}_p$. Compute $x := g^a$, $y := g^b$, $z := g^{ab+c}$ and $E = (g^r, e(a) \cdot g^{br})$. The output ot_1 is then the tuple (x, y, z, E).

2. $\mathsf{ot}_2 \leftarrow \mathsf{OT}_2(\mathsf{ot}_1, M_0, M_1)$: sample $t_0, s_0, t_1, s_1 \leftarrow \mathbb{Z}_p$. For each $i \in \{0, 1\}$, compute $w_i := x^{s_i} g^{t_i}$ and $u_i := \left(z \cdot g^{-i}\right)^{s_i} y^{t_i} M_i$. The output ot_2 is then the tuple (w_0, u_0, w_1, u_1).

3. $\mathsf{OT}_3(\mathsf{ot}_2, state)$: Compute M_c as $u_c \cdot w_c^{-b}$.

The above protocol is a Weak OT protocol. The argument follows directly from the proof of [NP01, AIR01] except that in our case the chooser's security will be based on the circular security of ElGamal.

We will now provide an attack that specifies a particular cheating strategy for a malicious sender (in a single instance of execution of the protocol) that can not be simulated by any unbounded simulator. In particular we will provide an efficient malicious sender strategy such that for every

unbounded simulator we have that the joint distributions of the sender's view and the receiver's output in the real world and the ideal world are efficiently distinguishable.

Our cheating sender proceeds as follows: On receiving the message (x, y, z, E) it proceeds by setting ot_2 to be the tuple (E, E). On receiving this message R, regardless of the value of b, outputs $e(a)$. Note that in the real world the joint distribution of the view of the sender and the output of the honest receiver $((x, y, z, E), e(a))$ is sufficient for the distinguisher to efficiently compute the input of the honest receiver. The distinguisher computes the honest receiver's input c as follows:

- Given : (x, y, z, E, a)
- Compute g^{ab} as y^a. (since $y = g^b$)
- Then, compute c as $\frac{z}{g^{ab}}$ (since $z = g^{ab+c}$).

However no unbounded simulator can simulate this distribution in the ideal world. □

The protocol used in describing the above attack is a simple modification of the Naor-Pinkas/Aiello-Ishai-Reingold protocol where the honest receiver with its first message includes an encryption of the secret key. It is reasonable to assume that ElGamal is indeed circularly secure. Furthermore if it was possible to efficiently obtain an encryption of the secret key given the public key then the counterexample presented above would extend to the Naor-Pinkas/Aiello-Ishai-Reingold protocol. We do not believe that such a procedure exists. But it seems likely that the existence of such a procedure that efficiently concocts an encryption of the secret key can not be ruled out under the DDH assumption alone. Based on this conjecture we can claim that Naor-Pinkas/Aiello-Ishai-Reingold protocol can not be proved secure with an unbounded simulator under the DDH assumption. We stress that we do not make any of these speculative assumptions elsewhere in the paper. We use them here just to possibly explain obstacles in coming up with proofs for known protocols under reasonable assumptions.

4 Zero-Knowledge from Weak OT

In the following two sections, we will prove that a secure realization of T-secure Weak OT protocol (Weak OT with T-chooser's security), where T is an appropriate super-polynomial function in the security parameter, suffices for realizing Two Message Secure Computation for any functionality. We will provide this construction in two steps. First, in this section, we will show that a T-secure Weak OT protocol suffices for constructing a Two Message Secure Computation protocol for the zero-knowledge functionality. In the next section, we will show that zero-knowledge and Weak OT suffice for realizing Two Message Secure Computation for any functionality. More efficient constructions can be obtained by first realizing Parallel OT and then using it to instantiate the protocols of Ishai et al. [IPS08, IKO+11].

Before we describe the protocol, let's list the primitives used. Let T, T_1 be some super-polynomial functions in the security parameter λ with $T < T_1$.

Parameters:

- $(\mathsf{OT}_1, \mathsf{OT}_2, \mathsf{OT}_3)$ be functions corresponding to a T-secure Weak OT protocol. That is, it is secure against all adversaries running in time $T.\text{poly}(\lambda)$, but can be broken by adversaries running in time $T_1.\text{poly}(\lambda)$.
- $\mathcal{C} = (\mathsf{Com}, \mathsf{Open})$ be a non-interactive T-extractable commitment scheme with non-uniform hiding. (see Definition 6 in Appendix A).

The construction of the protocol appears in Figure 1.

Notation for (modified) Blum's Hamiltonicity Protocol.

- The distribution $\mathcal{D}(\cdot,\cdot)$ on input x and witness w generates $(\alpha, \gamma^0, \gamma^1)$ as follows. Sample (a, b^0, b^1) such that a is the first message of Blum's Hamiltonicity protocol (instantiated with commitment $\mathcal{C}$) and b^0 and b^1 are the response on challenges 0 and 1 respectively. Let $\alpha = (a, c^0, c^1)$ where $c^0 = \mathsf{Com}(b^0; r^0)$ and $c^1 = \mathsf{Com}(b^1; r^1)$ and let $\gamma^0 = (b^0, r^0)$ and $\gamma^1 = (b^1, r^1)$.
- Let $\mathcal{V}_{Blum}$ be the (modified) verification algorithm for the Blum's Hamiltonicity protocol. More specifically, $\mathcal{V}_{Blum}$ on input $(x, \alpha, \beta, \gamma)$ outputs 1 if it the underlying transcript is an accepting transcript of the Blum's Hamiltonicity protocol.
- $\mathcal{S}_{Blum}$ on input (x, β) generates a simulated accepting transcript $(\alpha, \beta, \gamma^0, \gamma^1)$ such that it is computationally indistinguishable from a real transcript.
- Finally, we will also use the extractor for Blum's Hamiltonicity protocol, denoted by $\mathsf{BlumExt}$. The extractor on input x, α, γ^0 and γ^1 outputs the Hamiltonian cycle in x or $(\perp, \beta)$ such that for no value of γ, $\mathcal{V}_{Blum}(x, \alpha, 1-\beta, \gamma) = 1$. The extractor $\mathsf{BlumExt}$ runs in time $T.\mathrm{poly}(\lambda)$.

π_{ZK}

Common Input: A graph x.
Auxiliary Input for P: w such that w is a Hamiltonian cycle in the graph x.

1. $(\mathsf{zk}_1, \mathsf{zkst}) \leftarrow \mathsf{ZK}_1(1^\lambda)$:
   - For each $i \in [\lambda]$, V samples $\beta_i \leftarrow \{0,1\}$ and generates $(\mathsf{ot}_{1,i}, state_i) \leftarrow \mathsf{OT}_1(\beta_i)$.
   - It sets $\mathsf{zk}_1 := \{\mathsf{ot}_{1,i}\}_{i\in[\lambda]}$ and $\mathsf{zkst} := \{state_i\}_{i\in[\lambda]}$.
2. $\mathsf{zk}_2 \leftarrow \mathsf{ZK}_2(1^\lambda, \mathsf{zk}_1, x, w)$:
   - P parses zk_1 as $\{\mathsf{ot}_{1,i}\}_{i\in[\lambda]}$.
   - For each $i \in [\lambda]$, P generates $(\alpha_i, \gamma_i^0, \gamma_i^1) \leftarrow \mathcal{D}(x, w)$, $\mathsf{ot}_{2,i} \leftarrow \mathsf{OT}_2(\mathsf{ot}_{1,i}, \gamma_i^0, \gamma_i^1)$.
   - It then sets $\mathsf{zk}_2 := \{\mathsf{ot}_{2,i}, \alpha_i\}_{i\in[\lambda]}$.
3. $\mathsf{ZK}_3(1^\lambda, \mathsf{zk}_2, x, \mathsf{zkst})$:
   - V parses zkst as $\{state_i\}_{i\in[\lambda]}$ and zk_2 as $\{\mathsf{ot}_{2,i}, \alpha_i\}_{i\in[\lambda]}$.
   - For each $i \in [\lambda]$, V obtains $\gamma_i^{\beta_i}$ as $\mathsf{OT}_3(\mathsf{ot}_{2,i}, state_i)$ and checks to see if $\mathcal{V}_{Blum}(x, \alpha_i, \beta_i, \gamma_i^{\beta_i}) = 1$.
   - Output 1 if all the checks pass and 0 otherwise.

Figure 1: Two Message Secure Computation for Zero-Knowledge

Lemma 2. *Assuming that* $(\mathsf{OT}_1, \mathsf{OT}_2, \mathsf{OT}_3)$ *is a* 2^{λ^ϵ}*-secure Weak OT protocol and* $\mathcal{C} = (\mathsf{Com}, \mathsf{Open})$ *is a* 2^{λ^ϵ}*-extractable and non-uniformly hiding non-interactive*[6] *commitment scheme for some con-*

[6]Note that we present our protocol in terms of non-interactive commitments (implied by one-to-one OWFs) just for simplicity of notation. We can instead use Naor's two round commitment scheme [Nao91] that can be based on one-way functions. The only difference being that Naor's commitment is statistically binding instead of being perfectly binding. All claims which rely on this lemma will be stated keeping this simplification in mind.

stant $0 < \epsilon < 1$*, we have that the protocol* π_{ZK} *described in Figure 1 is a two message zero-knowledge argument for NP with quasi-polynomial simulation.*

This lemma immediately implies the following theorem.

Theorem 4. *Two round protocols for the zero-knowledge functionality with quasi-polynomial simulation can be based on sub-exponentially hard Decisional Diffie-Hellman Assumption.*

4.1 Security Proof

The correctness of the scheme follows from the correctness of Blum's Hamiltonicity protocol. We will now give proofs for the simulation of the prover (argument of knowledge) and of the verifier (zero-knowledge).

Remark: In the security proofs in this section and the next, the simulator will run in time $T_1 \cdot \mathsf{poly}(\lambda)$. Notice that when we instantiate the primitives, $T = 2^{\lambda^\epsilon}$ and $T_1 = 2^\lambda$. This corresponds to an exponential time simulator whereas we require the simulator to only run in quasi-polynomial time. We will use the standard trick of using a smaller security parameter to address this. Let $\lambda = \log^2(k)$. We will now use k as the security parameter in our protocols. Note that the assumptions are still sub-exponentially secure with respect to λ. However, the simulator now runs in time $2^{\log^2(k)} = k^{\log(k)}$ which is quasi-polynomial in the security parameter k.

4.1.1 Argument of Knowledge

First, we note that arguing argument of knowledge also implicitly captures soundness of the protocol. In order to argue the argument of knowledge property, we need to construct an extractor Ext with the following property: we require that for any PPT malicious prover P^* such that $(\mathsf{zk}_2, x^*) \leftarrow \mathsf{P}^*(1^\lambda, \mathsf{zk}_1)$ and $\mathsf{ZK}_3(1^\lambda, \mathsf{zk}_2, x^*, \mathsf{zkst}) = 1$ where $(\mathsf{zk}_1, \mathsf{zkst}) \leftarrow \mathsf{ZK}_1(1^\lambda)$ we have that the extractor algorithm Ext running in time $T \cdot \mathsf{poly}(\lambda)$ on input $(\mathsf{zk}_2, x^*, \mathsf{zkst})$ outputs a Hamiltonian cycle in the graph x^*. The extractor is described in Figure 2.

Input: $(\mathsf{zk}_2, x^*, \mathsf{zkst})$.
The extractor does the following:

- For each $i \in [\lambda]$, recall that $\alpha_i = (a_i, c_i^0, c_i^1)$ where c_i^0 and c_i^1 are commitments to strings γ_i^0 and γ_i^1 respectively. Run the T-time extractor ComExt on inputs c_i^0 and c_i^1 to obtain γ_i^0 and γ_i^1.
- For each $i \in [\lambda]$, execute the T-time extractor $\mathsf{BlumExt}$ on input $(\alpha, \gamma_i^0, \gamma_i^1)$. If $\mathsf{BlumExt}$ outputs a Hamiltonian cycle in graph x^*, then abort everything else and output the extracted cycle. On the other hand, if $\mathsf{BlumExt}$ outputs $(\perp, \cdot)$ for every $i \in [\lambda]$, then output $\perp$.

Figure 2: Extraction strategy against a malicious prover

Now we will argue that this extraction procedure described above successfully extracts a cycle in x^* with overwhelming probability. We will prove this by reaching a contradiction. Lets assume that there exists a PPT cheating prover P^* such that it succeeds in generating accepting proofs even though the extraction of the witness fails. More formally, lets P^* be a PPT adversary such

that, $\epsilon = \Pr[\mathsf{ZK}_3(1^\lambda, \mathsf{zk}_2, x^*, \mathsf{zkst}) = 1 \bigwedge \mathsf{Ext}(\mathsf{zk}_2, x^*, \mathsf{zkst}) = \perp : (\mathsf{zk}_1, \mathsf{zkst}) \leftarrow \mathsf{ZK}_1(1^\lambda), (\mathsf{zk}_2, x^*) \leftarrow \mathsf{P}^*(1^\lambda, \mathsf{zk}_1)]$ is non-negligible.

Then we will use such an adversarial prover P^* and construct an adversary contradicting T-chooser's security of the Weak OT protocol. We proceed with the following hybrids:

- H_0: This is the real game with the guarantee that ϵ is non-negligible.
- H_1: Recall than in H_0 ZK_3 outputs 1 only if $\mathcal{V}_{Blum}(x, \alpha_i, \beta_i, \gamma_i^{\beta_i}) = 1$ for every $i \in [\lambda]$. In H_1 we modify ZK_3 and denote it by ZK'_3. ZK'_3 samples a random subset $S \subset [\lambda]$ such that $|S| = (\lambda - 1)$ and check $\mathcal{V}_{Blum}(x, \alpha_i, \beta_i, \gamma_i^{\beta_i}) = 1$ for all $i \in S$ (as opposed to all $i \in [\lambda]$). We have that, $\Pr[\mathsf{ZK}'_3(1^\lambda, \mathsf{zk}_2, x^*, \mathsf{zkst}) = 1 \bigwedge \mathsf{Ext}(\mathsf{zk}_2, x^*, \mathsf{zkst}) = \perp : (\mathsf{zk}_1, \mathsf{zkst}) \leftarrow \mathsf{ZK}_1(1^\lambda), (\mathsf{zk}_2, x^*) \leftarrow P^*(1^\lambda, \mathsf{zk}_1)]$ is at least ϵ.

Let $R \subseteq [\lambda] \backslash S$, be a set such that $j \in R$ if $\mathcal{V}_{Blum}(x, \alpha_j, \beta_j, \gamma_j^{\beta_j}) = 1$. Clearly, $0 \leq |R| \leq 1$ since there is only one index not in S. Further, let E be the event such that $|R| = 1$. Now, its easy to see that the only way $|R|$ could be 0 is if the malicious prover P^* was able to guess S correctly. This can happen with probability at most $\frac{1}{\lambda}$ (i.e probability that P^* correctly guesses which random index was not part of S). Therefore, $\Pr[\neg E] = \frac{1}{\lambda}$ and so, $\Pr[E] = (1 - \frac{1}{\lambda})$. That is, probability that the other index not part of set S belongs to set R is at least $(1 - \frac{1}{\lambda})$.

Using this malicious prover P^*, we will construct an adversary $\mathcal{A}$ that contradicts the T-chooser's security of the Weak OT protocol. Note that commitments can be broken in time $T.\text{poly}(\lambda)$ but the chooser's bit in the OT protocol is assumed to be secure against adversaries running in time $T.\text{poly}(\lambda)$. The adversary $\mathcal{A}$ obtains an external challenge $\mathsf{OT}_1(b)$ for a random $b \in \{0, 1\}$, and it needs to guess b. It does the following:

- Invoke P^* and embed the challenge in one of the random locations $i^* \leftarrow [\lambda]$. That is, as part of the first message of the verifier, for index i^*, set the external challenge $\mathsf{OT}_1(b)$ as ot_{1,i^*} and $state_{i^*} = i^*$. For all other indices $i \in [\lambda]$, choose a random bit β_i and compute $(\mathsf{ot}_{1,i}, state_i) \leftarrow \mathsf{OT}_1(\beta_i)$.
- Set $\mathsf{zkst} := \{state_i\}_{i \in [\lambda]}$ and send $\mathsf{zk}_1 = \{\mathsf{ot}_{1,i}\}_{i \in [\lambda]}$.
- Obtain the message zk_2 from P^* and run the algorithm ZK'_3 on input $(1^\lambda, x^*, \mathsf{zk}_2, \mathsf{zkst})$ using the random set S to be $[\lambda] \backslash \{i^*\}$.
- If the algorithm ZK'_3 outputs 0 then $\mathcal{A}$ outputs a random bit.
- On the other hand if ZK'_3 outputs 1, similar to the extractor described above, $\mathcal{A}$ first runs ComExt to extract $\gamma_{i^*}^0$ and $\gamma_{i^*}^1$. Then, it outputs b' where $(\perp, b')$ is the output of $\mathsf{BlumExt}$ on input $(\alpha_i^*, \gamma_{i^*}^0, \gamma_{i^*}^1)$.

Analysis :
When ZK'_3 outputs 0 (which happens with probability $1 - \epsilon$) then $\mathcal{A}$'s guess about b will be correct with probability at least $\frac{1}{2}$. On the other hand when ZK'_3 outputs 1, we will have that with probability at least $(1 - \frac{1}{\lambda})$ $i^* \in R$ and hence b' where $(\perp, b')$ is the output of $\mathsf{BlumExt}$ on input $(\alpha, \gamma_{i^*}^0, \gamma_{i^*}^1)$ will be the correct guess for b. Compiling together the two cases we have that $\mathcal{A}$ guesses the bit b correctly with probability at least $(1-\epsilon).1/2 + \epsilon(1 - \frac{1}{\lambda}) = \frac{1}{2} + \frac{\epsilon}{2} - \frac{\epsilon}{\lambda}$ which is non-negligible if ϵ is non-negligible. This contradicts the T-chooser's security of the OT protocol.

This completes the proof of argument of knowledge.

Remark: The reason we had to switch to a new hybrid where for one index we don't run the Blum verifier (essentially the target index where we embed the external OT challenge) is that for the index where we embed the external challenge, in order to run the verifier, the adversary $\mathcal{A}$ needs the state that was generated when the algorithm OT_1 was run to generate the first message. However, since $\mathcal{A}$ got that externally, it doesn't have the state in its possession and hence can't run the Blum verifier for this index.

4.1.2 Zero-Knowledge

In order to show zero-knowledge (or simulating a malicious verifier V^*) we need to construct a simulator $\mathcal{S}$ satisfying Definition 2. Let's consider a malicious verifier V^* described using a pair of algorithms $(\mathsf{V}_1^*, \mathsf{V}_2^*)$. The simulation strategy is described in Figure 3. Note that the simulator runs in time $T_1.\text{poly}(\lambda)$.

Common Input: A graph x.

1. $(\mathsf{zk}_1, \mathsf{zkst}) \leftarrow \mathsf{V}_1^*(x, 1^\lambda)$: The malicious verifier runs V_1^* computes its first message zk_1 to be sent to the prover and some associated state zkst.
2. The simulator $\mathcal{S}$ does the following:
   - Parse zk_1 as $\{\mathsf{ot}_{1,i}\}_{i\in[\lambda]}$.
   - For each $i \in \lambda$, run $\mathsf{OTExt}(\mathsf{ot}_{1,i})$ to extract the challenge bit β_i of the verifier. By the sender's security in the OT protocol, we know that the extraction succeeds with non negligible probability. Note that this requires time $T_1.\text{poly}(\lambda)$.
   - For each $i \in \lambda$, run $\mathcal{S}_{Blum}$ on input (x, β_i) to produce $(\alpha_i, \gamma_i^0, \gamma_i^1)$. (where γ_i^{1-b} is in fact a dummy message)
   - Compute $\mathsf{ot}_{2,i} \leftarrow \mathsf{OT}_2(\mathsf{ot}_{1,i}, \gamma_i^0, \gamma_i^1)$.
   - Set $\mathsf{zk}_2 := \{\mathsf{ot}_{2,i}, \alpha_i\}_{i\in[\lambda]}$ and send it to the verifier.
3. $\mathsf{V}_2^*(x, \lambda, \mathsf{zkst}, \mathsf{zk}_2)$: The malicious verifier runs V_2^* outputs either 0 or 1.

Figure 3: Simulation strategy against a malicious verifier

Claim 1. *The simulation strategy described in Figure 3 is secure against a malicious verifier.*

Proof. Using a series of hybrid arguments, we will show that the view of the malicious verifier in the ideal world is computationally indistinguishable from its view in the real world.

Let's assume to the contrary that there exists a PPT malicious verifier $\mathsf{V}^* = (\mathsf{V}_1^*, \mathsf{V}_2^*)$ that has a non negligible probability ϵ of distinguishing its view in the real world from the ideal world. Let's consider the "best possible" initial message of the verifier - i.e the output of the algorithm V_1^* that produces the highest distinguishing probability between the views in the real and ideal worlds. Let's fix this message as the initial message zk_1^* of the verifier. That is, consider V_1^* to be a deterministic algorithm that takes as input the randomness used to output this best possible message.

Essentially, given any PPT malicious adversary $\widehat{\mathsf{R}^*}$ that can distinguish the two views with non-negligible probability ϵ, we are transforming it into a new deterministic adversary R^* such that the randomness used to produce this best possible initial message is hardwired inside it. Therefore, even R^* can distinguish the two views with probability at least ϵ.

Using this malicious verifier, we can construct a non-uniform PPT adversary $\mathcal{A}$ that breaks either the hiding property of the commitment scheme or the sender's security of the OT protocol. Note that the commitment scheme is secure against all PPT adversaries (it is only assumed to be broken by an adversary running in time $T.\text{poly}(\lambda)$) and the OT protocol in fact has statistical security and hence is secure against all PPT adversaries. Thus, this would lead to a contradiction. In our reduction, the non-uniform advice (or auxiliary input) given to the adversary $\mathcal{A}$ is the set of challenge bits $\{\beta_i^*\}_{i\in[\lambda]}$ of the verifier V_1^* that was used to generate the fixed first message (observe that this is exactly what the simulator in the ideal world extracts in the first step by running the OTExt algorithm). These challenge bits are also accessible to the second stage verifier V_2^* as part of the state - zkst that is output by V_1^*. We will now describe the reduction. $\mathcal{A}$ acts as the prover in its interaction with the malicious verifier V^*.

1. Hybrid 0: This is the real experiment where the message sent to the verifier $\mathsf{zk}_2 = \{\mathsf{ot}_{2,i}, \alpha_i\}_{i\in[\lambda]}$ is computed using the algorithm $\mathsf{ZK}_2(1^\lambda, \mathsf{zk}_1^*, x, w)$. Here, $\alpha_i = (a_i, c_i^0, c_i^1)$ where $c_i^0 = \mathsf{Com}(b_i^0; r_i^0)$, $c_i^1 = \mathsf{Com}(b_i^1; r_i^1)$, $\gamma_i^0 = (b_i^0, r_i^0)$ and $\gamma_i^1 = (b_i^1, r_i^1)$. Also, $\mathsf{ot}_{2,i} \leftarrow \mathsf{OT}_2(\mathsf{ot}_{1,i}, \gamma_i^0, \gamma_i^1)$.

2. Hybrid 1: For each $i \in \lambda$, compute $\gamma_i^{(1-\beta_i^*)} = (\perp, \perp)$.

3. Hybrid 2: For each $i \in \lambda$, compute $c_i^{(1-\beta_i^*)} = \mathsf{Com}(\perp; r_i^{(1-\beta_i^*)})$. Observe that this is same as the ideal world experiment since the simulator would do exactly this: replace the entries corresponding to the positions not challenged by the verifier (i.e positions $(1-\beta_i^*)$) using $\perp$.

We will now argue indistinguishability of the hybrids. Note that the auxiliary input is also given to the external challenger of the primitive we are trying to break in every pair of hybrids.

Lemma 3. *Assuming statistical sender's security of the OT protocol, Hybrid 0 is indistinguishable from Hybrid 1.*

Proof. In both hybrids, the receiver's choice bits in the OT protocol are the set of β_i^*'s. (which was fixed as the non-uniform advice given to the adversary $\mathcal{A}$ generating the hybrids). The only difference in the two hybrids is the way in which the values $(\gamma_i^{(1-\beta_i^*)})$ that are part of the sender's input to the OT protocol are generated. In hybrid 0, they are generated as $\gamma_i^{=}(b_i^{(1-\beta_i^*)}, r_i^{(1-\beta_i^*)})$ while in hybrid 1, they are generated as $\gamma_i^{(1-\beta_i^*)} = (\perp, \perp)$. Therefore, if the malicious verifier V^* can distinguish between the two hybrids with non-negligible probability ϵ, the non-uniform PPT adversary $\mathcal{A}$ can break the sender's security of the OT protocol with non-negligible probability ϵ which would be a contradiction. (In fact, it is enough to assume that the OT protocol has just computational security instead of statistical.) □

Lemma 4. *Assuming computational hiding of the commitment scheme, Hybrid 1 is indistinguishable from Hybrid 2.*

Proof. In both hybrids, the receiver's choice bits in the OT protocol are the set of β_i^*'s. (which was fixed as the non-uniform advice given to the adversary $\mathcal{A}$ generating the hybrids). The only difference in the two hybrids is the way in which the commitments $(c_i^{(1-\beta_i^*)})$ are generated. In hybrid

1, they are generated as $c_i^{(1-\beta_i^*)} = \mathsf{Com}(b_i^{(1-\beta_i^*)}; r_i^{(1-\beta_i^*)})$ while in hybrid 2, they are generated as $c_i^{(1-\beta_i^*)} = \mathsf{Com}(\bot; r_i^{(1-\beta_i^*)})$. Therefore, if the malicious verifier V^* can distinguish between the two hybrids with non-negligible probability ϵ, the non-uniform PPT adversary $\mathcal{A}$ can break the hiding property of the commitment with non-negligible probability ϵ which would be a contradiction. □

This completes the proof of zero knowledge. □

5 Two Message Secure Computation from Weak OT

In this section, we show that Weak OT together with two message witness indistinguishability gives an immediate construction of Two Message Secure Computation for the general functionality. This construction is obtained by compiling the Yao's garbled circuit construction (see Section A.3) and Weak OT protocol with our zero-knowledge protocol. Note that the zero knowledge protocol from Section 4 already satisfies this requirement. (and gives a much stronger functionality).

Let's consider two parties : the sender S with input x and the receiver R with input y that wish to securely compute any general function f. Before we describe the protocol, let's list the primitives used. Let T, T_1 be some super-polynomial functions in the security parameter λ with $T < T_1$.

Parameters:

- $(\mathsf{OT}_1, \mathsf{OT}_2, \mathsf{OT}_3)$ be functions corresponding to a T-secureWeak OT protocol. That is, its secure against all adversaries running in time $T.\text{poly}(\lambda)$, but can be broken by adversaries running in time $T_1.\text{poly}(\lambda)$.
- $(\mathsf{WI}_1, \mathsf{WI}_2, \mathsf{WI}_3)$ be a two message secure computation protocol for the witness indistinguishability functionality. This protocol is secure against all adversaries running in time $T.\text{poly}(\lambda)$, but can be broken by adversaries running in time $T_1.\text{poly}(\lambda)$.
- $(\mathsf{Garble}, \mathsf{GCEval})$ be the algorithms corresponding to Yao's garbled circuit construction that is secure against all adversaries running in time $T.\text{poly}(\lambda)$, but can be broken by adversaries running in time $T_1.\text{poly}(\lambda)$.
- com be a commitment scheme that is extractable in time $T.\text{poly}(\lambda)$.
- let OWF be a one-way function that is not invertible in time $T.\text{poly}(\lambda)$ but can be inverted by an attacker $\mathcal{A}_{\mathsf{OWF}}$ running in time $T_1.\text{poly}(\lambda)$.

In Figure 4 we describe the construction of our Two Message Secure Computation protocol. We will next prove its security.

Lemma 5. *If $(\mathsf{OT}_1, \mathsf{OT}_2, \mathsf{OT}_3)$ is a sub-exponentially secure Weak OT protocol, $(\mathsf{WI}_1, \mathsf{WI}_2, \mathsf{WI}_3)$ is a sub-exponentially secure two message witness indistinguishable argument for NP and sub-exponentially secure one-way functions exist, the protocol presented in Figure 4 is a Two Message Secure Computation protocol with quasi-polynomial simulation for any general function.*

This lemma immediately implies the following theorem.

Theorem 5. *Two Message Secure Computation protocols with quasi-polynomial simulation for general functionalities can be based on any of the following sub-exponential assumptions: (1) Decisional Diffie-Hellman Assumption; (2) Quadratic Residuosity Assumption; (3) N^{th} Residuosity Assumption; or (4) Witness Encryption (together with one-to-one one-way functions).*

$\langle S, R \rangle$

Inputs: The sender S gets input x and the receiver R gets input y. Both S and R get the function f they want to evaluate as input.
Output: R expects to receive $f(x, y)$ as output.

1. $(\mathsf{nisc}_1, state) \leftarrow \mathsf{NISC}_1(1^\lambda, y)$:
   - R generates $(\mathsf{ot}_{1,i}, state_i) \leftarrow \mathsf{OT}_1(y_i)$ for each $i \in [\lambda]$ and $(\mathsf{wi}_1, \mathsf{wist}) \leftarrow \mathsf{WI}_1(1^\lambda)$.
   - R chooses a random string $z \leftarrow \{0,1\}^\lambda$ and computes $Z = \mathsf{OWF}(z)$.
   - It sets $\mathsf{nisc}_1 := (\mathsf{ot}_{1,1}, \ldots, \mathsf{ot}_{1,\lambda}, \mathsf{wi}_1, Z)$, $state = (state_1, \ldots, state_\lambda, \mathsf{wist}, z, Z)$ and sends nisc_1 to S.
2. $\mathsf{nisc}_2 \leftarrow \mathsf{NISC}_2(\mathsf{nisc}_1, x)$:
   - S parses nisc_1 as $(\mathsf{ot}_{1,1}, \ldots, \mathsf{ot}_{1,\lambda}, \mathsf{wi}_1, Z)$.
   - It computes $c_1 = \mathsf{com}(x; r_1)$ and $c_2 = \mathsf{com}(0; r_2)$ using randomness r_1, r_2 respectively.
   - S samples a 2λ key tuple $\overline{K} = \{K_{i,b}\}_{i\in[\lambda], b\in\{0,1\}}$, where $K_{i,b} \in \{0,1\}^\lambda$ and generates a garbled circuit $GC := \mathsf{Garble}(\overline{K}, C; \omega)$ where $C_x(y)$ is a circuit that evaluates $f(x, y)$ on input y.
   - Then, S generates $\mathsf{ot}_{2,i} := \mathsf{OT}_2(\mathsf{ot}_{1,i}, K_{i,0}, K_{i,1}; \omega_i)$ for each $i \in [\lambda]$.
   - After that, S computes $\mathsf{wi}_2 \leftarrow \mathsf{WI}_2(1^\lambda, \mathsf{wi}_1, (\mathsf{ot}_{2,1}, \ldots, \mathsf{ot}_{2,\lambda}, GC, c_1, c_2, Z), (\omega, \{\omega_i\}_{i\in[\lambda]}, x, r_1, \perp, \perp))$ for the statement $(\mathsf{ot}_{2,1}, \ldots, \mathsf{ot}_{2,\lambda}, GC, c_1, c_2, Z) \in L$, where L contains tuples for which there exists either a witness $\Omega = (\overline{K}, C, \omega, \omega_1, \ldots, \omega_\lambda, x, r_1)$ such that:

     $$(1) \quad GC := \mathsf{Garble}(\overline{K}, C; \omega) \bigwedge c_1 = \mathsf{com}(x; r_1) \bigwedge \forall i \in [\lambda], \mathsf{ot}_{2,i} = \mathsf{OT}_2(\mathsf{ot}_{1,i}, K_{i,0}, K_{i,1}; \omega_i)$$

     (OR) there exists a witness $\Omega_2 = (z, r_2)$ such that:

     $$(2) \quad \mathsf{OWF}(z) = Z \bigwedge c_2 = \mathsf{com}(z; r_2)$$

   - Finally, S sets $\mathsf{nisc}_2 := (\mathsf{ot}_{2,1}, \ldots, \mathsf{ot}_{2,\lambda}, GC, \mathsf{wi}_2, c_1, c_2)$ and sends it to R.
3. $\mathsf{NISC}_3(\mathsf{nisc}_2, state)$:
   - R parses nisc_2 as $(\mathsf{ot}_{2,1}, \ldots, \mathsf{ot}_{2,\lambda}, GC, \mathsf{wi}_2, c_1, c_2)$ and $state$ as $(state_1, \ldots, state_\lambda, \mathsf{wist}, z, Z)$.
   - If $\mathsf{WI}_3(1^\lambda, \mathsf{wi}_2, (\mathsf{ot}_{2,1}, \ldots, \mathsf{ot}_{2,\lambda}, GC, c_1, c_2), \mathsf{wist}) = 0$ then R outputs $\perp$.
   - Otherwise, for each $i \in [\lambda]$ it obtains $K_{i,y_i} = \mathsf{OT}_3(\mathsf{ot}_{2,i}, state_i)$ and outputs $\mathsf{GCEval}(\overline{K}_y, GC)$.

Figure 4: Two Message Secure Computation for a general function f

We can easily transform the above protocol to a setting where both parties are required to receive outputs by adding an extra round. Now, for the special case of coin tossing, we get the following corollary:

Corollary 3. *Three round secure coin tossing protocols with quasi-polynomial simulation can be based on any of the following sub-exponential assumptions: (1) Decisional Diffie-Hellman Assump-*

tion; (2) Quadratic Residuosity Assumption; (3) N^{th} Residuosity Assumption; or (4) Witness Encryption (together with one-to-one one-way functions)

In order to get better efficiency we can use the Two Message Secure Computation protocol in Figure 4 and obtain a protocol (Parallel OT) for realizing the functionality that allows for $\mathsf{poly}(\lambda)$-parallel oblivious transfer invocations. We can then use this protocol in order to instantiate the protocols of Ishai et al. [IPS08, IKO+11].

5.1 Security Proof

In this section, we prove *Lemma* 5. That is, we prove the security of our two message secure computation protocol from Figure 4.

The correctness of the scheme easily follows from the correctness of the OT protocol, the commitment scheme, the one way function, the zero knowledge proof and the garbled circuit. We will now give proofs for the simulation of the malicious sender and the malicious receiver respectively.

Remark: Note that the one way function f used in the protocol is assumed to be secure against all adversaries running in time T.poly(λ) whereas, the commitment scheme used (which is built on some other one way function) is assumed to be secure only against non-uniform PPT adversaries and in particular, can be broken by an adversary running in time T.poly(λ).

5.1.1 Malicious Sender

We need to construct a simulator $\mathcal{S}$ satisfying Definition 2. Let's consider a malicious sender S^*. The simulation strategy is described in Figure 5. Note that the simulator runs only in time T.poly(λ) and not T_1.poly(λ).

1. $\mathcal{S}$ does the following:
   - Generate $(\mathsf{ot}_{1,i}, state_i) \leftarrow \mathsf{OT}_1(0)$ for each $i \in [\lambda]$ and $(\mathsf{wi}_1, \mathsf{wist}) \leftarrow \mathsf{WI}_1(1^\lambda)$.
   - Choose a random string $z \leftarrow \{0,1\}^\lambda$ and computes $Z = \mathsf{OWF}(z)$.
   - Set $\mathsf{nisc}_1 := (\mathsf{ot}_{1,1}, \ldots, \mathsf{ot}_{1,\lambda}, \mathsf{wi}_1, Z)$,
     $state = (state_1, \ldots, state_\lambda, \mathsf{wist}, z, Z)$ and send nisc_1 to S.
2. The malicious party S^* sends nisc_2 to $\mathcal{S}$.
3. $\mathcal{S}$ does the following:
   - Parse nisc_2 as $(\mathsf{ot}_{2,1}, \ldots, \mathsf{ot}_{2,\lambda}, GC, \mathsf{wi}_2, c_1, c_2)$.
   - If $\mathsf{WI}_3(1^\lambda, \mathsf{wi}_2, (\mathsf{ot}_{2,1}, \ldots, \mathsf{ot}_{2,\lambda}, GC, c_1, c_2), \mathsf{wist}) = 0$ then, output $\perp$.
   - Else, run the T-time extractor ComExt on input c_1 to obtain x^*.
   - Send x^* to the ideal functionality as the input of the malicious sender.

Figure 5: Simulation strategy against a malicious sender

Claim 2. *The simulation strategy described in Figure 5 is secure against a malicious sender.*

Proof. From the chooser's security of the OT protocol, the view of the malicious prover is indistinguishable in both the real and ideal worlds. Therefore, let's now focus on the output of the honest receiver R. If we show that the simulator $\mathcal{S}$ extracts the input of the malicious sender correctly with overwhelming probability, then the output of R in the real and ideal worlds would be the same.

First, observe that if the algorithm WI_3 outputs 0 in step 3 of the simulation (the simulator outputs $\perp$), the proof didn't verify and so even in the real world, the receiver would output $\perp$ and hence the views are identical.

Now, let's analyze the probability that the value x^* sent by the simulator $\mathcal{S}$ to the ideal functionality is not equal to the malicious sender's input x, given that the algorithm WI_3 output 1 (i.e verified that the proof wi_2 is correct). That is, let's assume there exists a cheating sender S^* that can cause the simulator to output a value x^* with non-negligible probability ϵ such that $f(x^*, y) \neq f(x, y)$ where y is the input of the honest receiver. That is, $\epsilon = \Pr[\mathsf{WI}_3(1^\lambda, \mathsf{wi}_2, (\mathsf{ot}_{2,1}, \ldots, \mathsf{ot}_{2,\lambda}, GC, c_1, c_2), \mathsf{wist}) = 1 \bigwedge f(\mathcal{S}(\mathsf{nisc}_2, state), y) \neq f(x, y)]$ is non-negligible.

First, observe that from the argument of knowledge property proved in Section 4 about the proof system, if the proof verifies correctly (i.e the algorithm WI_3 outputs 1), then with overwhelming probability, there exists a valid witness. In other words, similar to the argument of knowledge property proved in *Section* 4, we can break the T-chooser's security of the OT scheme if the malicious sender is able to generate a valid proof, but the extractor is unable to extract a witness. Now, we will use the malicious sender S^* and construct an adversary $\mathcal{A}$ that can invert the one-way function OWF in time $T.\text{poly}(\lambda)$ which would contradict our assumption that OWF is not invertible in time $T.\text{poly}(\lambda)$. The adversary $\mathcal{A}$ works as follows:

- Receives as an external challenge a value Z^* that is the output of the one-way function OWF.
- Interacts with the malicious sender S^*. Sends the first message of the protocol nisc_1 - where the value Z that is part of nisc_1 is set to be Z^*.
- Receives $\mathsf{nisc}_2 = (\mathsf{ot}_{2,1}, \ldots, \mathsf{ot}_{2,\lambda}, GC, \mathsf{wi}_2, c_1, c_2)$ from S^*. (Note : we are given that the algorithm WI_3 outputs 1).
- Runs the extractor of the witness indistinguishable proof system to obtain a valid witness. This takes time $T.\text{poly}(\lambda)$.
- Note that if the output of the extractor is a valid witness for statement (1) in protocol Figure 4, then by the correctness of the extractor ComExt, the garbling scheme, the OT protocol and the witness indistinguishable proof, $\mathsf{ComExt}(c_1) = x^*$ such that $f(x^*, y) = f(x, y)$ which would contradict the assumption we started with.
- Therefore, the witness obtained from the witness indistinguishable proof system extractor is a valid witness for statement (2). In particular, the witness is (z, r_2) such that $c_2 = \mathsf{com}(z; r_2)$ and $\mathsf{OWF}(z) = Z^*$.
- Output z as the preimage of Z^*.

Therefore, the adversary $\mathcal{A}$ runs in time $T.\text{poly}(\lambda)$ and inverts the one way function OWF leading to a contradiction.

□

5.1.2 Malicious Receiver

We need to construct a simulator $\mathcal{S}$ satisfying Definition 2. Let's consider a malicious party R^* described using a pair of algorithms $(\mathsf{R}_1^*, \mathsf{R}_2^*)$. The simulation strategy is described in Figure 6. Note that the simulator runs in time $T_1.\text{poly}(\lambda)$.

1. $(\mathsf{nisc}_1, state) \leftarrow \mathsf{R}_1^*(y, 1^\lambda)$: The malicious party runs R_1^* to compute its first message nisc_1 to be sent to S and some associated state *state*.
2. The simulator $\mathcal{S}$ does the following:
   - Parse nisc_1 as $(\mathsf{ot}_{1,1}, \ldots, \mathsf{ot}_{1,\lambda}, \mathsf{wi}_1, Z)$.
   - For each $i \in \lambda$, run $\mathsf{OTExt}(\mathsf{ot}_{1,i})$ to extract the challenge bit y_i of R^*. By the sender's security in the OT protocol, we know that the extraction succeeds with non negligible probability. Concatenate all of them to recover input y of R^*.
   - Run the attacker of the one way function $\mathcal{A}_{\mathsf{OWF}}$ on input Z to invert it and produce z. That is, $\mathsf{OWF}(z) = Z$. For the sake of simplicity, let's assume that the one way function is onto - i.e every element in its co-domain has a pre-image. Therefore, a malicious receiver can't send a string in the co-domain that doesn't have an inverse. We know that this succeeds with overwhelming probability since the attacker, running in time $T_1.\text{poly}(\lambda)$ succeeds with overwhelming probability.
   - Send the input y to the ideal functionality to receive back the output of the function - $f(x, y)$.
   - Compute $c_1 = \mathsf{com}(0; r_1)$ and $c_2 = \mathsf{com}(z; r_2)$ using randomness r_1, r_2 respectively.
   - Sample a 2λ key tuple $\overline{K} = \{K_{i,b}\}_{i\in[\lambda], b\in\{0,1\}}$, where $K_{i,b} \in \{0,1\}^\lambda$.
   - Run the simulator of the garbled circuit construction $\mathcal{S}_{GC}$ on input $(\overline{K}, f(x, y))$ to generate a simulated garbled circuit GC.
   - Compute $\mathsf{ot}_{2,i} := \mathsf{OT}_2(\mathsf{ot}_{1,i}, K_{i,0}, K_{i,1}; \omega_i)$ for each $i \in [\lambda]$.
   - Run the algorithm WI_2 to compute wi_2 for the statement $(\mathsf{ot}_{2,1}, \ldots, \mathsf{ot}_{2,\lambda}, GC, c_1, c_2, Z) \in L$, using the witness $(\bot, \bot, \bot, \bot, z, r_2)$. That is, use the witness to prove that the statement (2) is true.
   - Finally, set $\mathsf{nisc}_2 := (\mathsf{ot}_{2,1}, \ldots, \mathsf{ot}_{2,\lambda}, GC, \mathsf{zk}_2, c_1, c_2)$ and send it to R^*.
3. $\mathsf{R}_2^*(\mathsf{nisc}_2, state))$: The malicious party runs the algorithm R_2^* and produces some output.

Figure 6: Simulation strategy against a malicious receiver

Claim 3. *The simulation strategy described in Figure 6 is secure against a malicious receiver.*

Proof. Since the honest sender has no output from the ideal functionality, it is enough to focus on just the view of the malicious receiver. Using a series of hybrid arguments, we will show that the view of the malicious receiver in the ideal world is computationally indistinguishable from its view in the real world.

Let's assume to the contrary that there exists a PPT malicious receiver $\mathsf{R}^* = (\mathsf{R}_1^*, \mathsf{R}_2^*)$ that has a non negligible probability ϵ of distinguishing its view in the real world from the ideal world. Let's

consider the "best possible" initial message of the receiver - i.e the output of the algorithm R_1^* that produces the highest distinguishing probability between the views in the real and ideal worlds. Let's fix this message as the initial message nisc_1^* of the receiver. That is, consider R_1^* to be a deterministic algorithm that takes as input the randomness used to output this best possible message.

Essentially, given any PPT malicious adversary $\widehat{\mathsf{R}^*}$ that can distinguish the two views with non-negligible probability ϵ, we are transforming it into a new deterministic adversary R^* such that the randomness used to produce this best possible initial message is hardwired inside it. Therefore, even R^* can distinguish the two views with probability at least ϵ.

Using this malicious party R^*, we can construct a non-uniform PPT adversary $\mathcal{A}$ that breaks one amongst the hiding property of the commitment scheme, the security of the garbling scheme, the witness indistinguishability property of the proof system and the sender's security of the OT protocol. Note that all the above listed primitives are secure against even non-uniform PPT adversaries and thus, this would lead to a contradiction. In our reduction, the non-uniform advice (or auxiliary input) given to the adversary $\mathcal{A}$ is the set of challenge bits $\{y_i^*\}_{i\in[\lambda]}$ of the party R_1^* that was used to generate the $\mathsf{ot}_{1,i}$'s as part of the fixed first message and the preimage z of the value Z sent as part of the initial message. (observe that this is exactly what the simulator in the ideal world extracts by running the OTExt algorithm and the one-way function inverter $\mathcal{A}_{\mathsf{OWF}}$). These challenge bits are also accessible to the second stage algorithm R_2^* as part of the state - *state* that is output by R_1^*. We will now describe the reduction. $\mathcal{A}$ acts as the party Alice in its interaction with the malicious party R^*.

1. Hybrid 0: This is the real experiment where the message sent to R^* is $(\mathsf{ot}_{2,1}, \ldots, \mathsf{ot}_{2,\lambda}, GC, \mathsf{wi}_2, c_1, c_2)$. Here, wi_2 is computed using the algorithm WI_2 for the statement (1).

2. Hybrid 1: In this hybrid, compute $c_2 = \mathsf{com}(z; r_2)$.

3. Hybrid 2: In this hybrid, compute wi_2 by running WI_2 using the witnes (z, r_2) to prove that statement (2) is true.

4. Hybrid 3: Recall that till now, $\mathsf{ot}_{2,i}$ is computed as $\mathsf{ot}_{2,i} := \mathsf{OT}_2(\mathsf{ot}_{1,i}, K_{i,0}, K_{i,1}; \omega_i)$ for each $i \in [\lambda]$. In this hybrid, for each $i \in \lambda$, compute $K_{i,(1-y_i^*)}$ uniformly at random.

5. Hybrid 4: In this hybrid, the garbled circuit GC is computed as the output of the simulator $\mathcal{S}_{GC}$ on input $(\overline{K}, f(x, y^*))$.

6. Hybrid 5: In this hybrid, compute $c_1 = \mathsf{com}(0; r_1)$. Observe that this is same as the ideal world experiment.

We now prove the indistinguishability of the hybrids. Note that the auxiliary input is also given to the external challenger of the primitive we are trying to break in every pair of hybrids.

Lemma 6. *Assuming computational hiding of the commitment scheme, Hybrid 0 is indistinguishable from Hybrid 1.*

Proof. The only difference in the two hybrids is the way in which the commitment c_2 is generated. Therefore, if the malicious party R^* can distinguish between the two hybrids with non-negligible probability ϵ, the non-uniform PPT adversary $\mathcal{A}$ can break the hiding property of the commitment with non-negligible probability ϵ which would be a contradiction. □

Lemma 7. *Assuming that* $(\mathsf{WI}_1, \mathsf{WI}_2, \mathsf{WI}_3)$ *is a secure protocol for witness indistinguishability, Hybrid 1 is indistinguishable from Hybrid 2.*

Proof. The only difference in the two hybrids is the way in which the second message of the witness indistinguishability protocol wi_2 is generated. In hybrid 1, it is generated by proving that statement (1) is correct, while in hybrid 2, it is generated by proving that statement (2) is correct.

Therefore, if the malicious party R^* can distinguish between the two hybrids with non-negligible probability ϵ, the non-uniform PPT adversary $\mathcal{A}$ can break the witness indistinguishability property of the proof system with non-negligible probability ϵ which would be a contradiction. □

Lemma 8. *Assuming statistical sender's security of the OT protocol, Hybrid 2 is indistinguishable from Hybrid 3.*

Proof. In both hybrids, the receiver's choice bits in the OT protocol are the set of y_i^*'s. (which was fixed as the non-uniform advice given to the adversary $\mathcal{A}$ generating the hybrids). The only difference in the two hybrids is the way in which the values $(K_{i,(1-y_i^*)})$ that are part of the sender's input to the OT protocol are generated. Therefore, if the malicious verifier V^* can distinguish between the two hybrids with non-negligible probability ϵ, the non-uniform PPT adversary $\mathcal{A}$ can break the sender's security of the OT protocol with non-negligible probability ϵ which would be a contradiction. (in fact, it is enough to assume that the OT protocol has just computational security instead of statistical). □

Lemma 9. *Assuming that* $(\mathsf{Garble}, \mathsf{GCEval})$ *is a secure garbling scheme, Hybrid 3 is indistinguishable from Hybrid 4.*

Proof. The only difference in the two hybrids is the way in which the garbled circuit is generated. In hybrid 3, it is generated honestly by running the Garble algorithm, while in hybrid 4, it is generated using the simulator $\mathcal{S}_{GC}$. Therefore, if the malicious party R^* can distinguish between the two hybrids with non-negligible probability ϵ, the non-uniform PPT adversary $\mathcal{A}$ can break the security of the garbling scheme with non-negligible probability ϵ which would be a contradiction. □

Lemma 10. *Assuming computational hiding of the commitment scheme, Hybrid 4 is indistinguishable from Hybrid 5.*

Proof. The only difference in the two hybrids is the way in which the commitment c_1 is generated. Therefore, if the malicious party R^* can distinguish between the two hybrids with non-negligible probability ϵ, the non-uniform PPT adversary $\mathcal{A}$ can break the hiding property of the commitment with non-negligible probability ϵ which would be a contradiction. □

This completes the proof of security against a malicious receiver.

□

6 Weak OT from High Rate Semi-Honest OT

In this section, we first give a generic construction of two message Weak OT from any high rate two message semi-honest OT.

As defined earlier, let λ be the security parameter. Consider a sender $\mathcal{S}$ with inputs $(\mathsf{m}_0, \mathsf{m}_1) \in \{0,1\}^n$ where $n = \mathsf{poly}(\lambda)$ and a receiver $\mathcal{R}$ with choice bit b who wish to run a Weak OT protocol. Let $\mathsf{OT}^{sh} = (\mathsf{OT}_1^{sh}, \mathsf{OT}_2^{sh}, \mathsf{OT}_3^{sh})$ be a two message semi-honest secure OT protocol with high rate c (> 0.5). Recall that the rate of the OT protocol is defined as the ratio of the size of one of

the sender's input strings to the size of the sender's message. That is, $\mathsf{rate} = \frac{|\mathsf{m}_0|}{|\mathsf{OT}_2^{sh}(\mathsf{m}_0,\mathsf{m}_1,\mathsf{OT}_1^{sh}(\mathsf{b}))|}$. Let $\mathsf{Ext} : \{0,1\}^{s(n)} \times \{0,1\}^d \to \{0,1\}^n$ be a (k,ϵ) strong seeded randomness extractor as defined in Appendix A, where $d = O(\log(s(n)))$ and $s(n) = \frac{c*n}{(2c-1)*(1-O(1))}$. Recall that we know how to construct such a strong seeded extractor for $n = k*(1-O(1))$. We prove the following theorem.

Theorem 6. *Assuming* $\mathsf{OT}^{sh} = (\mathsf{OT}_1^{sh}, \mathsf{OT}_2^{sh}, \mathsf{OT}_3^{sh})$ *is a high rate (*> 0.5*) two message semi-honest OT protocol, the protocol in Figure 7 securely realizes the Weak OT functionality.*

The construction of the Weak OT protocol is as follows.

π_{WeakOT}

1. $(\mathsf{ot}_1, state) \leftarrow \mathsf{OT}_1(\mathsf{b})$:
   - Compute $(\mathsf{ot}_1^{sh}, state) = \mathsf{OT}_1^{sh}(\mathsf{b})$.
   - Set the output $\mathsf{ot}_1 = \mathsf{ot}_1^{sh}$.

2. $\mathsf{ot}_2 \leftarrow \mathsf{OT}_2(\mathsf{ot}_1, \mathsf{m}_0, \mathsf{m}_1)$:
   - Pick two random strings $(\mathsf{r}_0, \mathsf{r}_1) \in \{0,1\}^{4n}$ and two random seeds $\mathsf{s}_0, \mathsf{s}_1 \in \{0,1\}^d$.
   - Compute $\mathsf{ot}_2^{sh} = \mathsf{OT}_2^{sh}(\mathsf{ot}_1, \mathsf{r}_0, \mathsf{r}_1)$.
   - Compute $\mathsf{S}_0 = \mathsf{m}_0 \oplus \mathsf{Ext}(\mathsf{r}_0, \mathsf{s}_0)$ and $\mathsf{S}_1 = \mathsf{m}_1 \oplus \mathsf{Ext}(\mathsf{r}_1, \mathsf{s}_1)$.
   - Output $\mathsf{ot}_2 = (\mathsf{ot}_2^{sh}, \mathsf{S}_0, \mathsf{s}_0, \mathsf{S}_1, \mathsf{s}_1)$.

3. $\mathsf{OT}_3(\mathsf{ot}_2, state)$:
   - Parse ot_2 as $(\mathsf{ot}_2^{sh}, \mathsf{S}_0, \mathsf{s}_0, \mathsf{S}_1, \mathsf{s}_1)$
   - Compute $\mathsf{r}_\mathsf{b} = \mathsf{OT}_3^{sh}(\mathsf{ot}_2^{sh}, state)$.
   - Compute the output $\mathsf{m}_b = (\mathsf{S}_\mathsf{b} \oplus \mathsf{Ext}(\mathsf{r}_\mathsf{b}, \mathsf{s}_\mathsf{b}))$.

Figure 7: Weak OT from semi-honest OT

Correctness: The correctness of the protocol can be observed easily from the correctness of the semi-honest secure OT scheme OT^{sh} and the correctness of the extractor Ext.

Chooser's Security We need to show that for every non-uniform polynomial-time adversary $\mathcal{A}$, $|\Pr[\mathcal{A}(\mathsf{OT}_1(1^\lambda, 0)) = 1] - \Pr[\mathcal{A}(\mathsf{OT}_1(1^\lambda, 1)) = 1]|$ is negligible in λ. Recall that the output of $\mathsf{OT}_1(1^\lambda, \mathsf{b}))$ is $(\mathsf{OT}_1^{sh}(1^\lambda, \mathsf{b}))$. We will now show if such an adversary $\mathcal{A}$ exists, then we can use $\mathcal{A}$ to break the chooser's security of the semi-honest secure OT protocol OT^{sh} which will be a contradiction.

Consider an adversarial sender $\mathcal{A}_{\mathsf{OT}^{sh}}$ for the semi-honest secure protocol that interacts with a challenger $\mathcal{C}_{\mathsf{OT}^{sh}}$. $\mathcal{A}_{\mathsf{OT}^{sh}}$ plays the role of the receiver in the Weak OT protocol with the adversary $\mathcal{A}$. $\mathcal{A}_{\mathsf{OT}}^{sh}$ receives the challenge ot_1^{sh} from the challenger. It now forwards this to $\mathcal{A}$ as the first message ot_1 of the OT protocol. $\mathcal{A}_{\mathsf{OT}}^{sh}$ receives a guess b' as the output of $\mathcal{A}$ and forwards this as its output to $\mathcal{C}_{\mathsf{OT}}^{sh}$. It is easy to see that $\mathcal{A}_{\mathsf{OT}}^{sh}$ has broken the chooser's security of the semi-honest secure OT protocol OT^{sh} which is a contradiction.

Remark: Note that we can also give a simulation-based guarantee if we allow a resetting simulator. The simulator (who does not know the choice bit of the actual receiver) first simulates

the (honest) receiver with choice bit $\mathsf{b} = 0$. Then, it resets the sender and simulates the honest receiver with choice bit $\mathsf{b} = 1$. This way, the simulator can extract both messages m_0 and m_1 from the corrupted sender and then give them to the trusted party.

Statistical Sender's Security We now construct an unbounded time Simulator Sim that interacts with a malicious receiver $\mathcal{R}^*$ in the ideal world and show that the receiver's view is statistically indistinguishable in the real and ideal worlds. Let the honest sender have inputs $(\mathsf{m}_0, \mathsf{m}_1)$. The strategy for the simulator $\mathsf{Sim}_{\mathcal{R}}$ against a malicious receiver is described in Figure 8.

Upon receiving a message ot_1^* from $\mathcal{R}^*$, Sim does the following:

1. Consider two random variables $\mathsf{r}_0, \mathsf{r}_1$ such that the support of each of them consists of all strings of length $s(n)$.
2. Let $\mathsf{ot}_2^{sh} = \mathsf{OT}_2^{sh}(\mathsf{ot}_1^*, \mathsf{r}_0, \mathsf{r}_1)$.
3. Since OT^{sh} is a high rate OT, we know that $\frac{|\mathsf{r}_0|}{|\mathsf{ot}_2^{sh}|} = c$ where $c > 0.5$.
4. Therefore, given $(\mathsf{ot}_2^{sh}, \mathsf{ot}_1^*)$, there must exist a b such that $H_\infty(\mathsf{r}_\mathsf{b}) > (2s(n) - s(n)/c)$.
5. Query the ideal functionality $\mathcal{F}_{\mathsf{ot}}$ with $(1 - \mathsf{b})$ and receives as output $\mathsf{m}_{1-\mathsf{b}}$.
6. Pick two random seeds $\mathsf{s}_0, \mathsf{s}_1 \in \{0,1\}^d$.
7. Compute S_b as a uniformly random string of length n and $\mathsf{S}_{1-\mathsf{b}} = \mathsf{m}_{1-\mathsf{b}} \oplus \mathsf{Ext}(\mathsf{r}_{1-\mathsf{b}}, \mathsf{s}_{1-\mathsf{b}})$.
8. Output $\mathsf{ot}_2 = (\mathsf{ot}_2^{sh}, \mathsf{S}_0, \mathsf{s}_0, \mathsf{S}_1, \mathsf{s}_1))$.

Figure 8: Simulation strategy against a malicious receiver.

The view of the adversary is the tuple ot_2. Notice that the only difference in the real and ideal world is the way the string S_b is computed. In the real world, $\mathsf{S}_\mathsf{b} = \mathsf{m}_\mathsf{b} \oplus \mathsf{Ext}(\mathsf{r}_\mathsf{b}, \mathsf{s}_\mathsf{b})$ whereas in the ideal world, it is computed as a uniformly random string. Observe that on expanding $s(n)$, $H_\infty(\mathsf{r}_\mathsf{b}) = n/(1 - O(1))$. Therefore, from the definition of strong seeded extractors, recall that since $H_\infty(\mathsf{r}_\mathsf{b}) = n/(1 - O(1))$ and s_b is a uniformly random string of length (d), $SD\left((\mathsf{Ext}(\mathsf{r}_\mathsf{b}, U_d), U_d), (U_n, U'_d)\right) \leq \epsilon$ where ϵ is a negligible function in λ. Thus, $\mathsf{Ext}(\mathsf{r}_\mathsf{b}, \mathsf{s}_\mathsf{b})$ is statistically indistinguishable from a uniformly random string and hence S_b in the real and ideal worlds are statistically indistinguishable and this completes the proof.

6.1 Weak OT from High Rate Linear Homomorphic Encryption

In this section, we first show how to construct two message semi-honest OT from any linear homomorphic encryption. Let $\mathsf{LHE} = (\mathsf{Setup}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Add}, \mathsf{Const.Mul})$ be any linear homomorphic encryption scheme as defined in Appendix A.7. We prove the following theorem.

Theorem 7. *Assuming* LHE *is a high rate* (> 0.5) *linear homomorphic encryption scheme, the protocol in Figure 9 securely realizes the OT functionality with semi-honest security.*

We now describe the construction of the two message semi-honest OT protocol.

Correctness: The correctness of the protocol can be observed easily from the correctness of the encryption scheme LHE. Also, observe that if the encryption scheme has high rate, so does the OT protocol.

$\pi_{\mathsf{OT}^{sh}}$

1. $(\mathsf{ot}_1^{sh}, \mathit{state}) \leftarrow \mathsf{OT}_1^{sh}(\mathsf{b})$:
   - Generate $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{Setup}(1^\lambda)$ for the encryption scheme.
   - Compute the output $\mathsf{ot}_1^{sh} := (\mathsf{pk}, \mathsf{Enc}_{\mathsf{pk}}(\mathsf{b}; \mathsf{r}))$ usign randomness r. Let *state* be $(\mathsf{b}, \mathsf{pk}, \mathsf{sk})$.

2. $\mathsf{ot}_2^{sh} \leftarrow \mathsf{OT}_2^{sh}(\mathsf{ot}_1, \mathsf{m}_0, \mathsf{m}_1)$:
   - Compute $\mathsf{ct}_0 = \mathsf{Const.Mul}(\mathsf{ot}_1, \mathsf{m}_1)$.
   - Let $\mathsf{ct}_{1,1} = \mathsf{Const.Mul}(\mathsf{Enc}_{\mathsf{pk}}(1; \mathsf{r}'), \mathsf{m}_0)$ where r' is a random string. Let $\mathsf{ct}_{1,2} = \mathsf{Const.Mul}(\mathsf{ot}_1, -1)$ and $\mathsf{ct}_{1,3} = \mathsf{Const.Mul}(\mathsf{ct}_{1,2}, \mathsf{m}_1)$.
   - Compute $\mathsf{ct}_1 = \mathsf{Add}(\mathsf{ct}_{1,1}, \mathsf{ct}_{1,3})$.
   - Compute $\mathsf{ct} = \mathsf{Add}(\mathsf{ct}_0, \mathsf{ct}_1)$. That is, $\mathsf{ct} = \mathsf{Enc}_{\mathsf{pk}}(\mathsf{b} \cdot \mathsf{m}_1 + (1 - \mathsf{b}) \cdot \mathsf{m}_0)$.
   - Output $\mathsf{ot}_2 = \mathsf{ct}$.

3. $\mathsf{OT}_3(\mathsf{ot}_2, \mathit{state})$:
   - Parse *state* as $(\mathsf{b}, \mathsf{pk}, \mathsf{sk})$ and ot_2 as (ct)
   - Compute the output $\mathsf{m}_\mathsf{b} = \mathsf{Dec}_{\mathsf{sk}}(\mathsf{ct})$.

Figure 9: OT^{sh} from linear homomorphic encryption

Chooser's Security Recall that $\mathsf{OT}_1(1^\lambda, \mathsf{b})) = (\mathsf{pk}, \mathsf{Enc}_{\mathsf{pk}}(\mathsf{b}))$. The simulation strategy in the ideal is pretty simple: the simulator runs the setup of the encryption algorithm and just outputs $(\mathsf{pk}, \mathsf{Enc}_{\mathsf{pk}}(0))$ as his first message. We will now show if there exists an adversarial sender $\mathcal{A}$ that can distinguish between the real and ideal worlds, then we can use $\mathcal{A}$ to break the security of the encryption scheme LHE which will be a contradiction.

First, observe that if the receiver's input $\mathsf{b} = 0$, then the views are identical in both the worlds and hence can't be distinguished. Therefore, let's consider the scenario where $\mathsf{b} = 1$. Consider an adversary $\mathcal{A}_{\mathsf{LHE}}$ for the encryption scheme that interacts with a challenger $\mathcal{C}_{\mathsf{LHE}}$. $\mathcal{A}_{\mathsf{LHE}}$ plays the role of the receiver in the Weak OT protocol with the adversary $\mathcal{A}$. $\mathcal{A}_{\mathsf{LHE}}$ sends $(0, 1)$ to $\mathcal{C}_{lhe}$ and receives the public key pk and the challenge ciphertext $\mathsf{Enc}_{\mathsf{pk}}(\mathsf{b}'))$. It now forwards both of these $(\mathsf{pk}, \mathsf{Enc}_{\mathsf{pk}}(\mathsf{b}')))$to $\mathcal{A}$ as the first message ot_1 as the OT protocol. Notice that $\mathsf{b}' = 0$ corresponds to the ideal world and $\mathsf{b}' = 1$ corresponds to the real world. $\mathcal{A}_{\mathsf{LHE}}$ receives a guess as the output of $\mathcal{A}$ and forwards this as its output to $\mathcal{C}_{\mathsf{LHE}}$. Thus, $\mathcal{A}_{\mathsf{LHE}}$ has broken the security of the encryption scheme which is a contradiction.

Statistical Sender's Security We now construct a simulator Sim that interacts with a semi-honest receiver $\mathcal{R}^*$ with input b in the ideal world and show that the receiver's view is statistically indistinguishable in the real and ideal worlds. Let the honest sender have inputs $(\mathsf{m}_0, \mathsf{m}_1)$. The strategy for the simulator $\mathsf{Sim}_{\mathcal{R}}$ against a malicious receiver is described in Figure 10.

The view of the adversary is the tuple ot_2. Notice that in both worlds, it is computed identically as an encryption of m_b. Hence, the view in the real and ideal worlds are statistically indistinguishable and this completes the proof.

Finally, as a corollary of the Theorem 6 and Theorem 7, we get a construction of Weak OT from any high rate linear homomorphic encryption scheme. Formally:

Input: $(\mathsf{b}, \mathsf{m}_\mathsf{b})$
Upon receiving a message ot_1^{sh} from $\mathcal{R}^*$, Sim does the following:

1. If $\mathsf{b} = 1$, output $\mathsf{ot}_2^{sh} = \mathsf{Add}(\mathsf{Const.Mul}(\mathsf{ot}_1, \mathsf{m}_1), \mathsf{Enc}_{\mathsf{pk}}(0; \mathsf{r}_0))$ where r_0 is a random string.

2. If $\mathsf{b} = 0$, let $\mathsf{ct}_{1,1} = \mathsf{Const.Mul}(\mathsf{Enc}_{\mathsf{pk}}(1; \mathsf{r}'), \mathsf{m}_0)$, $\mathsf{ct}_{1,2} = \mathsf{Const.Mul}(\mathsf{ot}_1, -1)$ and $\mathsf{ct}_{1,3} = \mathsf{Const.Mul}(\mathsf{ct}_{1,2}, \mathsf{r}_1)$ where $\mathsf{r}', \mathsf{r}_1$ are random strings. Output $\mathsf{ot}_2^{sh} = \mathsf{Add}(\mathsf{ct}_{1,1}, \mathsf{ct}_{1,3})$.

Figure 10: Simulation strategy against a malicious receiver.

Corollary 4. *Assuming* LHE *is a high rate (> 0.5) linear homomorphic encryption scheme, the protocol in Figure 7 securely realizes the Weak OT functionality.*

6.2 Weak OT from N^{th} Residuosity Assumption

Finally, we instantiate the high rate linear homomorphic encryption scheme using a construction where the size of the ciphertext is λ more than the size of the plaintext. Such an encryption scheme can be built based on the N^{th} Residuosity Assumption [DJ01, IP07]. As a result, we get the following corollary:

Corollary 5. *Assuming the N^{th} Residuosity Assumption holds, the protocol in Figure 7 securely realizes the Weak OT functionality.*

An earlier construction of Weak OT based on the N^{th} Residuosity Assumption appeared in [HK12]. In that construction, they first construct Weak OT from any smooth projective hash function which is then instantiated based on the N^{th} Residuosity Assumption using a complex transformation. Our construction and analysis are arguably simpler.

7 Weak OT from Witness Encryption

In this section, we give a new construction of Weak OT based on any injective one-way function along with *witness encryption*.

In this subsection we show that a T-extractable non-interactive commitment scheme $\mathcal{C} = (\mathsf{Com}, \mathsf{Open})$ with non-uniform hiding and a non-uniform witness encryption scheme[7] $(\mathsf{Enc}_{\mathsf{WE}}, \mathsf{Dec}_{\mathsf{WE}})$ (defined in Appendix A.5) can be used to realize a Weak OT protocol. Recall that the non-interactive commitment scheme can be based on injective one way functions. We next provide the construction of Weak OT which is very similar to the construction from indistinguishability obfuscation given in [SW14].

Lemma 11. *Assuming injective one-way functions exist and a non-uniform witness encryption scheme exists, the protocol described above is a secure Weak OT protocol.*

Proof. The proof is very similar to the one in Section 5.7 of [SW14] and hence is not written in more detail here. □

[7] The construction of Garg et. al. [GGSW13] is secure even against non-uniform adversaries.

π_{WeakOT}

- $(\mathsf{ot}_1, state) \leftarrow \mathsf{OT}_1(c)$: Let the output $\mathsf{ot}_1 := \mathsf{Com}(c; r)$, where r are the random coins. Let *state* be (c, r).
- $\mathsf{ot}_2 \leftarrow \mathsf{OT}_2(\mathsf{ot}_1, M_0, M_1)$: For each $\beta \in \{0, 1\}$, compute u_β as $\mathsf{Enc}_{\mathsf{WE}}(1^\lambda, x_\beta, M_\beta)$ where x_β if the following NP-statement:

  $$\exists s \text{ such that } \mathsf{ot}_1 = \mathsf{Com}(\beta; s)$$

  The output ot_2 is then the tuple (u_0, u_1).
- $\mathsf{OT}_3(\mathsf{ot}_2, state)$: Parse *state* as (c, r) and compute M_c as $\mathsf{Dec}_{\mathsf{WE}}(u_c, r)$.

Figure 11: Weak OT from Witness Encryption

8 Acknowledgements

The second author's research is supported in part from DARPA/ARL SAFEWARE Award W911NF15C0210, AFOSR Award FA9550-15-1-0274, NSF CRII Award 1464397, AFOSR YIP Award and research grants by the Okawa Foundation and Visa Inc. The views expressed are those of the author and do not reflect the official policy or position of the funding agencies.

The third author's research is supported in part by a DARPA/ARL SAFEWARE award, NSF Frontier Award 1413955, NSF grants 1619348, 1228984, 1136174, and 1065276, NSF-BSF grant 2015782, ISF grant 1709/14, BSF grant 2012378, a Xerox Faculty Research Award, a Google Faculty Research Award, an equipment grant from Intel, and an Okawa Foundation Research Grant. This material is based upon work supported by the Defense Advanced Research Projects Agency through the ARL under Contract W911NF-15-C-0205. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense, the National Science Foundation, or the U.S. Government.

The fourth author's research is supported in part from a DARPA/ARL SAFEWARE award, NSF Frontier Award 1413955, NSF grants 1619348, 1228984, 1136174, and 1065276, BSF grant 2012378, a Xerox Faculty Research Award, a Google Faculty Research Award, an equipment grant from Intel, and an Okawa Foundation Research Grant. This material is based upon work supported by the Defense Advanced Research Projects Agency through the ARL under Contract W911NF-15-C-0205. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense, the National Science Foundation, or the U.S. Government.

References

[AGJ+12] Shweta Agrawal, Vipul Goyal, Abhishek Jain, Manoj Prabhakaran, and Amit Sahai. New impossibility results for concurrent composition and a non-interactive completeness theorem for secure computation. In Reihaneh Safavi-Naini and Ran Canetti, editors, *Advances in Cryptology – CRYPTO 2012*, volume 7417 of *Lecture Notes in Computer Science*, pages 443–460, Santa Barbara, CA, USA, August 19–23, 2012. Springer, Berlin, Germany. 6

[AIR01] William Aiello, Yuval Ishai, and Omer Reingold. Priced oblivious transfer: How to sell digital goods. pages 119–135, 2001. 4, 10, 11, 13, 14

[AMPR14] Arash Afshar, Payman Mohassel, Benny Pinkas, and Ben Riva. Non-interactive secure computation based on cut-and-choose. In *EUROCRYPT*, 2014. 3

[BFM88] Manuel Blum, Paul Feldman, and Silvio Micali. Non-interactive zero-knowledge and its applications (extended abstract). In *STOC*, 1988. 4

[BGJ+17] Saikrishna Badrinarayanan, Vipul Goyal, Abhishek Jain, Dakshita Khurana, and Amit Sahai. Round optimal concurrent mpc via strong simulation. In *TCC*, 2017. 7

[BHHI10] Boaz Barak, Iftach Haitner, Dennis Hofheinz, and Yuval Ishai. Bounded key-dependent message security. In Henri Gilbert, editor, *Advances in Cryptology – EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 423–444, French Riviera, May 30 – June 3, 2010. Springer, Berlin, Germany. 38

[BHHO08] Dan Boneh, Shai Halevi, Michael Hamburg, and Rafail Ostrovsky. Circular-secure encryption from decision diffie-hellman. In David Wagner, editor, *Advances in Cryptology – CRYPTO 2008*, volume 5157 of *Lecture Notes in Computer Science*, pages 108–125, Santa Barbara, CA, USA, August 17–21, 2008. Springer, Berlin, Germany. 14

[Blu82] Manuel Blum. Coin flipping by telephone. In *Proc. IEEE Spring COMPCOM*, pages 133–137, 1982. 38

[BLV03] Boaz Barak, Yehuda Lindell, and Salil P. Vadhan. Lower bounds for non-black-box zero knowledge. In *44th Annual Symposium on Foundations of Computer Science*, pages 384–393, Cambridge, Massachusetts, USA, October 11–14, 2003. IEEE Computer Society Press. 3

[BPS06] Boaz Barak, Manoj Prabhakaran, and Amit Sahai. Concurrent non-malleable zero knowledge. In *47th Annual Symposium on Foundations of Computer Science*, pages 345–354, Berkeley, CA, USA, October 21–24, 2006. IEEE Computer Society Press. 6

[BS05] Boaz Barak and Amit Sahai. How to play almost any mental game over the net - Concurrent composition via super-polynomial simulation. In *46th Annual Symposium on Foundations of Computer Science*, pages 543–552, Pittsburgh, PA, USA, October 23–25, 2005. IEEE Computer Society Press. 3, 6

[CCKM00] Christian Cachin, Jan Camenisch, Joe Kilian, and Joy Müller. One-round secure computation and secure autonomous mobile agents. In *Automata, Languages and Programming, 27th International Colloquium, ICALP 2000, Geneva, Switzerland, July 9-15, 2000, Proceedings*, pages 512–523, 2000. 3

[CCM98] Christian Cachin, Claude Crépeau, and Julien Marcil. Oblivious transfer with a memory-bounded receiver. In *39th Annual Symposium on Foundations of Computer Science*, pages 493–502, Palo Alto, California, USA, November 8–11, 1998. IEEE Computer Society Press. 10

[CHK03] Ran Canetti, Shai Halevi, and Jonathan Katz. A forward-secure public-key encryption scheme. In Eli Biham, editor, *Advances in Cryptology – EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 255–271, Warsaw, Poland, May 4–8, 2003. Springer, Berlin, Germany. 4

[CLMP12] Kai-Min Chung, Edward Lui, Mohammad Mahmoody, and Rafael Pass. Unprovable security of two-message zero knowledge. Cryptology ePrint Archive, Report 2012/711, 2012. 13

[CLP10] Ran Canetti, Huijia Lin, and Rafael Pass. Adaptive hardness and composable security in the plain model from standard assumptions. In *51st Annual Symposium on Foundations of Computer Science*, pages 541–550, Las Vegas, Nevada, USA, October 23–26, 2010. IEEE Computer Society Press. 6

[CLT13] Jean-Sébastien Coron, Tancrède Lepoint, and Mehdi Tibouchi. Practical multilinear maps over the integers. In *CRYPTO*, 2013. 39

[DFKS16] Nico Döttling, Nils Fleischhacker, Johannes Krupp, and Dominique Schröder. Two-message, oblivious evaluation of cryptographic functionalities. In *Annual Cryptology Conference*, pages 619–648. Springer, 2016. 6

[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: 1st Theory of Cryptography Conference*, volume 2951 of *Lecture Notes in Computer Science*, pages 446–472, Cambridge, MA, USA, February 19–21, 2004. Springer, Berlin, Germany. 10

[DJ01] Ivan Damgård and Mads Jurik. A generalisation, a simpli. cation and some applications of paillier's probabilistic public-key system. In *International Workshop on Public Key Cryptography*, 2001. 6, 31, 40

[DJKL12] Dana Dachman-Soled, Abhishek Jain, Yael Tauman Kalai, and Adriana Lopez-Alt. On the (in)security of the fiat-shamir paradigm, revisited. Cryptology ePrint Archive, Report 2012/706, 2012. 13

[DKSS13] Zeev Dvir, Swastik Kopparty, Shubhangi Saraf, and Madhu Sudan. Extensions to the method of multiplicities, with applications to kakeya sets and mergers. *SIAM J. Comput.*, 42(6):2305–2328, 2013. 39

[DN07] Cynthia Dwork and Moni Naor. Zaps and their applications. *SIAM Journal on Computing*, 36(6):1513–1543, 2007. 3

[FLS99] Uriel Feige, Dror Lapidot, and Adi Shamir. Multiple noninteractive zero knowledge proofs under general assumptions. *SIAM J. Comput.*, 1999. 4

[GGH13] Sanjam Garg, Craig Gentry, and Shai Halevi. Candidate multilinear maps from ideal lattices. In *EUROCRYPT*, 2013. 39

[GGJS12] Sanjam Garg, Vipul Goyal, Abhishek Jain, and Amit Sahai. Concurrently secure computation in constant rounds. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology – EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*, pages 99–116, Cambridge, UK, April 15–19, 2012. Springer, Berlin, Germany. 6

[GGSW13] Sanjam Garg, Craig Gentry, Amit Sahai, and Brent Waters. Witness encryption and its applications. In *STOC*, 2013. 4, 6, 31, 39

[GK96] Oded Goldreich and Hugo Krawczyk. On the composition of zero-knowledge proof systems. *SIAM J. Comput.*, 25(1):169–192, 1996. 3

[GKOV12] Sanjam Garg, Abishek Kumarasubramanian, Rafail Ostrovsky, and Ivan Visconti. Impossibility results for static input secure computation. In Reihaneh Safavi-Naini and Ran Canetti, editors, *Advances in Cryptology – CRYPTO 2012*, volume 7417 of *Lecture Notes in Computer Science*, pages 424–442, Santa Barbara, CA, USA, August 19–23, 2012. Springer, Berlin, Germany. 6

[GL89] Oded Goldreich and Leonid A. Levin. A hard-core predicate for all one-way functions. In *21st Annual ACM Symposium on Theory of Computing*, pages 25–32, Seattle, Washington, USA, May 15–17, 1989. ACM Press. 38

[GMPP16] Sanjam Garg, Pratyay Mukherjee, Omkant Pandey, and Antigoni Polychroniadou. The exact round complexity of secure computation. In *EUROCRYPT*, 2016. 3

[GMW87] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game or A completeness theorem for protocols with honest majority. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computing, 1987, New York, New York, USA*, pages 218–229, 1987. 5

[GO94] Oded Goldreich and Yair Oren. Definitions and properties of zero-knowledge proof systems. *Journal of Cryptology*, 7(1):1–32, 1994. 3, 13

[GOS06a] Jens Groth, Rafail Ostrovsky, and Amit Sahai. Non-interactive zaps and new techniques for NIZK. In *Advances in Cryptology - CRYPTO 2006, 26th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 2006, Proceedings*, pages 97–111, 2006. 4

[GOS06b] Jens Groth, Rafail Ostrovsky, and Amit Sahai. Perfect non-interactive zero knowledge for NP. In Serge Vaudenay, editor, *Advances in Cryptology – EUROCRYPT 2006*, volume 4004 of *Lecture Notes in Computer Science*, pages 339–358, St. Petersburg, Russia, May 28 – June 1, 2006. Springer, Berlin, Germany. 4

[Goy12] Vipul Goyal. Positive results for concurrently secure computation in the plain model. In *FOCS*, 2012. 6

[GRS+11] Sanjam Garg, Vanishree Rao, Amit Sahai, Dominique Schröder, and Dominique Unruh. Round optimal blind signatures. In Phillip Rogaway, editor, *Advances in Cryptology – CRYPTO 2011*, volume 6841 of *Lecture Notes in Computer Science*, pages 630–648, Santa Barbara, CA, USA, August 14–18, 2011. Springer, Berlin, Germany. 3, 6, 38

[GUV09] Venkatesan Guruswami, Christopher Umans, and Salil P. Vadhan. Unbalanced expanders and randomness extractors from parvaresh–vardy codes. *J. ACM*, 56(4), 2009. 39

[HK07] Omer Horvitz and Jonathan Katz. Universally-composable two-party computation in two rounds. In *Advances in Cryptology - CRYPTO 2007, 27th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2007, Proceedings*, pages 111–129, 2007. 3

[HK12] Shai Halevi and Yael Tauman Kalai. Smooth projective hashing and two-message oblivious transfer. *J. Cryptology*, 25(1), 2012. 4, 6, 10, 11, 13, 14, 31

[HV16] Carmit Hazay and Muthuramakrishnan Venkitasubramaniam. What security can we achieve within 4 rounds? In *Security and Cryptography for Networks - 10th International Conference, SCN 2016, Amalfi, Italy, August 31 - September 2, 2016, Proceedings*, pages 486–505, 2016. 3, 5

[IKO+11] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, Manoj Prabhakaran, and Amit Sahai. Efficient non-interactive secure computation. In Kenneth G. Paterson, editor, *Advances in Cryptology – EUROCRYPT 2011*, volume 6632 of *Lecture Notes in Computer Science*, pages 406–425, Tallinn, Estonia, May 15–19, 2011. Springer, Berlin, Germany. 3, 15, 23

[ILL89] Russell Impagliazzo, Leonid A. Levin, and Michael Luby. Pseudo-random generation from one-way functions. In *21st Annual ACM Symposium on Theory of Computing*, pages 12–24, Seattle, Washington, USA, May 15–17, 1989. ACM Press. 39

[IP07] Yuval Ishai and Anat Paskin. Evaluating branching programs on encrypted data. In *TCC*, 2007. 31

[IPS08] Yuval Ishai, Manoj Prabhakaran, and Amit Sahai. Founding cryptography on oblivious transfer - efficiently. In David Wagner, editor, *Advances in Cryptology – CRYPTO 2008*, volume 5157 of *Lecture Notes in Computer Science*, pages 572–591, Santa Barbara, CA, USA, August 17–21, 2008. Springer, Berlin, Germany. 15, 23

[JKKR17] Abhishek Jain, Yael Tauman Kalai, Dakshita Khurana, and Ron Rothblum. Distinguisher-dependent simulation in two rounds and its applications. *IACR Cryptology ePrint Archive*, 2017:330, 2017. 7

[KO04] Jonathan Katz and Rafail Ostrovsky. Round-optimal secure two-party computation. In *CRYPTO*, 2004. 3, 5

[KS17] Dakshita Khurana and Amit Sahai. Two-message non-malleable commitments from standard sub-exponential assumptions. *IACR Cryptology ePrint Archive*, 2017:291, 2017. 7

[Lin04] Yehuda Lindell. Lower bounds for concurrent self composition. In Moni Naor, editor, *TCC 2004: 1st Theory of Cryptography Conference*, volume 2951 of *Lecture Notes in Computer Science*, pages 203–222, Cambridge, MA, USA, February 19–21, 2004. Springer, Berlin, Germany. 6

[Nao91] Moni Naor. Bit commitment using pseudorandomness. *Journal of Cryptology*, 4(2):151–158, 1991. 16

[NP01] Moni Naor and Benny Pinkas. Efficient oblivious transfer protocols. In *12th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 448–457, Washington, DC, USA, January 7–9, 2001. ACM-SIAM. 4, 10, 11, 13, 14

[Pas03] Rafael Pass. Simulation in quasi-polynomial time, and its application to protocol composition. In Eli Biham, editor, *Advances in Cryptology – EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 160–176, Warsaw, Poland, May 4–8, 2003. Springer, Berlin, Germany. 3, 6, 12

[PS04] Manoj Prabhakaran and Amit Sahai. New notions of security: Achieving universal composability without trusted setup. In László Babai, editor, *36th Annual ACM Symposium on Theory of Computing*, pages 242–251, Chicago, Illinois, USA, June 13–16, 2004. ACM Press. 3, 6

[SW14] Amit Sahai and Brent Waters. How to use indistinguishability obfuscation: deniable encryption, and more. In *STOC 2014*, pages 475–484, 2014. 31

[TU12] Amnon Ta-Shma and Christopher Umans. Better condensers and new extractors from parvaresh-vardy codes. In *CCC*, pages 309–315, 2012. 39

[Yao86] Andrew Chi-Chih Yao. How to generate and exchange secrets (extended abstract). In *27th Annual Symposium on Foundations of Computer Science, Toronto, Canada, 27-29 October 1986*, pages 162–167, 1986. 3

A Required Primitives

In this section we review some of the required primitives. Most definitions are standard, except that in some cases we require security against superpolynomial adversaries. In these cases we write, e.g., T-one-wayness and mean one-wayness against adversaries running in time $T \cdot \mathsf{poly}(\lambda)$. We will now review those primitives and security definitions that are not standard.

A.1 One-Way Functions.

The standard notion of one-wayness of functions is defined as follows. Let $\mathcal{A}$ be an adversary and define OW-advantage for a function $f : \{0,1\}^\lambda \mapsto \{0,1\}^\lambda$ as

$$\mathbf{Adv}^{ow}_{f,\mathcal{A}}(\lambda) = \Pr\left[\ f(z) = y : x \leftarrow \{0,1\}^\lambda\ ;\ y \leftarrow f(x)\ ;\ z \leftarrow \mathcal{A}(y)\ \right]\ .$$

Definition 3. *A function f is* non-uniform one-way *if $\mathbf{Adv}^{ow}_{f,\mathcal{A}}$ is negligible for any non-uniform PPT algorithm $\mathcal{A}$.*

A.2 Non-interactive Commitment Scheme.

A commitment scheme consists of a pair of efficient algorithms $\mathcal{C} = (\mathsf{Com}, \mathsf{Open})$ where: Com takes as input 1^λ and $m \in \{0,1\}^\lambda$ and outputs $(\mathsf{decom}, \mathsf{com}) \leftarrow \mathsf{Com}(1^\lambda, m)$, where decom and com are both of length $\{0,1\}^\lambda$; the algorithm $\mathsf{Open}(\mathsf{decom}, \mathsf{com})$ outputs a message m or $\perp$ if com is not a valid commitment to any message.

It is assumed that the commitment scheme is complete, i.e., for any message $m \in \{0,1\}^\lambda$ and $(\mathsf{decom}, \mathsf{com}) \leftarrow \mathsf{Com}(1^\lambda, m)$, we have $\mathsf{Open}(\mathsf{decom}, \mathsf{Com}(m)) = m$ with overwhelming probability in $\lambda \in \mathbb{N}$.

Commitment schemes must satisfy two properties: hiding and binding. Hiding means that no adversary can distinguish which of two messages are locked in the commitment. Let $\mathcal{A}$ be a non-uniform adversary against $\mathcal{C}$ and define its hiding-advantage as

$$\mathbf{Adv}^{\text{hid}}_{\mathcal{C},\mathcal{A}}(\lambda) = 2 \cdot \Pr\left[b = b' \;\middle|\; \begin{array}{l}(m_0, m_1, \mathsf{st}) \leftarrow \mathcal{A}(1^\lambda); b \leftarrow \{0,1\}; \\ (\mathsf{decom}, \mathsf{com}) \leftarrow \mathsf{Com}(m_b); b' \leftarrow \mathcal{A}(\mathsf{com}, \mathsf{st})\end{array}\right] - 1 \,.$$

Definition 4. *$\mathcal{C}$ is* non-uniform hiding *if the advantage function $\mathbf{Adv}^{hid}_{\mathcal{C},\mathcal{A}}$ is a negligible function for all non-uniform adversaries $\mathcal{A}$ running in time $\mathsf{poly}(\lambda)$.*

Binding says that the adversary cannot open the commitment in two different ways. Here, we define the strongest variant known as perfectly binding.

Definition 5. *$\mathcal{C}$ is* perfectly binding *if there exist no values $(\mathsf{com}, m_0, m_1, \mathsf{decom}_0, \mathsf{decom}_1)$ with distinct messages $m_0 \neq m_1$ such that $\mathsf{Open}(\mathsf{com}, \mathsf{decom}_0) = m_0$ and $\mathsf{Open}(\mathsf{com}, \mathsf{decom}_1) = m_1$.*

In addition to these requirements we assume that the commitment scheme is extractable in superpolynomial time T, i.e., there exists an algorithm that gets as input a commitment and outputs the contained message. More formally:

Definition 6. *A commitment scheme $\mathcal{C}$ is* extractable *in time T, if there exists an algorithm* ComExt *running in time $T \cdot \mathsf{poly}(\lambda)$ such that for any $\mathsf{com}, \mathsf{decom}$ with $\mathsf{Open}(\mathsf{com}, \mathsf{decom}) = m \neq \perp$, we have $\mathsf{ComExt}(\mathsf{com}) = m$ with overwhelming probability.*

Such a commitment scheme can be instantiated [GL89] by any permutation f that is T-one-way and a hard-core predicate B of f. To commit to a bit b, the sender computes $(f(U_\lambda), B(U_\lambda) \oplus b)$, where U_λ is the uniform distribution over $\{0,1\}^\lambda$. It can also be based on injective one way functions[Blu82]

A.3 Yao's garbled circuit.

We start by recalling the Yao's construction of garbled circuit as formalized in Theorem 4 in [BHHI10]. Notice that we state their theorem for non-uniform security (as in [GRS+11]) while it has been stated for uniform security in [BHHI10].

Theorem 8 (Garbled Circuits [BHHI10])**.** *Suppose that non-uniformly secure one-way functions exist. Then there is a pair of polynomial-time randomized algorithms $(\mathsf{Garble}, \mathsf{GCEval})$ that for security/input parameter λ, output parameter m, and circuit size parameter s satisfy the following:*

- ***Syntax.*** *Garble takes a 2λ key tuple $\overline{K} = \{K_{i,b}\}_{i\in[\lambda],b\in\{0,1\}}$, where $K_{i,b} \in \{0,1\}^\lambda$, and a size s circuit describing a function $h : \{0,1\}^\lambda \to \{0,1\}^m$, and outputs a "garbled circuit" GC. GCEval takes an input $x \in \{0,1\}^\lambda$, a λ key tuple, and a garbled circuit GC and outputs $y \in \{0,1\}^m$.*
- ***Correctness.*** *We require that if $GC = \mathsf{Garble}(\overline{K}, h)$ then $\mathsf{GCEval}(\overline{K}_x, GC) = h(x)$, where we define $\overline{K}_x = \{(x_i, K_{i,x_i})\}_{i\in[\lambda]}$.*[8]

[8] For ease of notation we assume that the input x is included in the description of $\overline{K}_x$. This is needed to guarantee correctness even when a pair of keys happen to be identical. Alternatively, we could avoid giving x as input to GCEval by either settling for statistical correctness or allowing the keys to be correlated.

- ***Security against receiver.*** *For every polynomials $s(\lambda), m(\lambda)$, every $x \in \{0,1\}^\lambda$ and every $h : \{0,1\}^\lambda \to \{0,1\}^{m(\lambda)}$ of size $s(\lambda)$ there exists a PPT simulator $\mathcal{S}_{GC}$ such that if $\overline{K}$ is chosen at random then*

$$\overline{K}_x \| \mathsf{Garble}(\overline{K}, h) \overset{c}{\approx} \overline{K}_x \| \mathcal{S}_{GC}(\overline{K}, h(x))$$

Here $\overset{c}{\approx}$ denotes non-uniform computational indistinguishability.

- ***Security against outsider.*** *For every polynomials $s(\lambda)$, $m(\lambda)$ and every $h, h' : \{0,1\}^\lambda \to \{0,1\}^{m(\lambda)}$ of size $s(\lambda)$, if $\overline{K}$ is chosen at random then*

$$\mathsf{Garble}(\overline{K}, h) \overset{c}{\approx} \mathsf{Garble}(\overline{K}, h')$$

A.4 Other Definitions

Definition 7 (Statistical Difference)**.** *Let X and Y be two distributions with supports in $\{0,1\}^k$. The* statistical difference *between X and Y, $SD(X,Y)$ is given by,*

$$SD(X,Y) = \frac{1}{2} \sum_{x \in \{0,1\}^k} \left| \text{Prob}[X = x] - \text{Prob}[Y = x] \right|.$$

Definition 8 (Min-Entropy)**.** *For a random variable X, the* min-entropy *of X, $H_\infty(X)$ is defined as,*

$$H_\infty(X) = \min_x \left\{ \frac{1}{\log(\text{Prob}[X = x])} \right\}.$$

*A random variable with min-entropy at least k is called a k-*source*.*

Definition 9 (Extractors)**.** *A function $\mathsf{Ext} : \{0,1\}^m \times \{0,1\}^d \to \{0,1\}^n$ is called a (k,ϵ)-*strong extractor *if for every k-source X with support in $\{0,1\}^m$,*

$$SD\left((\mathsf{Ext}(X, U_d), U_d), (U_n, U'_d)\right) \leq \epsilon,$$

where U_d, U'_d are independent uniform distributions over $\{0,1\}^d$, and U_n is uniform over $\{0,1\}^n$.

In particular, we know [ILL89, GUV09, TU12, DKSS13] how to construct strong seeded extractors with negligible ϵ for any entropy $k >= \log^2(m)$ with seed length $d = O(\log m)$ and $n = (1 - O(1)) * k$ output bits. We use them in our constructions in Section 6.

A.5 Witness Encryption

Very recently, Garg et. al. introduced the notion of witness encryption [GGSW13] and provided a construction based on the Decision Multilinear No-Exact-Cover Assumption instantiated with multilinear maps [GGH13, CLT13].

Definition 10. *A* witness encryption *scheme for an* **NP** *language L (with corresponding witness relation R) consists of the following two polynomial-time algorithms:*

> **Encryption.** *The algorithm $\mathsf{Enc}_{\mathsf{WE}}(1^\lambda, x, M)$ takes as input a security parameter 1^λ, an unbounded-length string x, and a message $M \in \mathcal{M}$ for some message space $\mathcal{M}$, and outputs a ciphertext CT.*

Decryption. *The algorithm* $\mathsf{Dec}_{\mathsf{WE}}(\mathsf{CT}, w)$ *takes as input a ciphertext* CT *and an unbounded-length string* w*, and outputs a message* M *or the symbol* $\perp$*.*

These algorithms satisfy the following two conditions:

- ***Correctness.*** *For any security parameter* λ*, for any* $M \in \mathcal{M}$*, and for any* $x \in L$ *such that* $R(x, w)$ *holds, we have that*

$$\Pr\Big[\mathsf{Dec}_{\mathsf{WE}}\big(\mathsf{Enc}_{\mathsf{WE}}(1^\lambda, x, M), w\big) = M\Big] = 1 - neg(\lambda)$$

- ***Soundness Security.*** *For any* $x \notin L$*, for any PPT adversary* A *and messages* $M_0, M_1 \in \mathcal{M}$*, there exists a negligible function* $neg(\cdot)$*, such that:*

$$\Big|\Pr\big[A(\mathsf{Enc}_{\mathsf{WE}}(1^\lambda, x, M_0)) = 1\big] - \Pr\big[A(\mathsf{Enc}_{\mathsf{WE}}(1^\lambda, x, M_1)) = 1\big]\Big| < neg(\lambda)$$

A.6 High Rate Oblivious Transfer

Let $(\mathsf{ot}_1, \mathsf{ot}_2, \mathsf{ot}_3)$ be a two message 1-out-of-2 oblivious transfer protocol between a sender $\mathcal{S}$ with inputs (M_0, M_1) and a receiver $\mathcal{R}$ with choice bit b. We say that the protocol has a rate c if, for any security parameter λ, there exists d such that for all senders's inputs (M_0, M_1) where each M_i is of length $s(\lambda) > \lambda^d$, $\frac{|M_i|}{|\mathsf{ot}_2(M_0, M_1, \mathsf{ot}_1(b))|} > c$. Here, $i \in \{1, 2\}$.

In this paper, we are interested in two message semi-honest OT protocols with high (> 0.5) rate.

A.7 Linear Homomorphic Encryption

A linear homomorphic encryption scheme $\mathsf{LHE} = (\mathsf{Setup}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Add}, \mathsf{Const.Mul})$ consists of the following 5 algorithms.

- $\mathsf{Setup}(1^\lambda)$: The setup algorithm takes as input a security parameter λ and outputs a public key-private key pair $(\mathsf{pk}, \mathsf{sk})$.
- $\mathsf{Enc}_{\mathsf{pk}}(\mathsf{m})$: The encryption algorithm takes as input the public key pk and a plaintext message m with size $n = \mathsf{poly}(\lambda)$. It outputs a ciphertext ct.
- $\mathsf{Dec}_{\mathsf{sk}}(\mathsf{ct})$: The decryption algorithm takes as input the secret key sk and a ciphertext ct. It outputs a string y or $\perp$.
- $\mathsf{Add}(\mathsf{ct}_1, \mathsf{ct}_2)$: The evaluation algorithm takes as input two ciphertexts ct_1 and ct_2. It outputs a new ciphertext ct that is an encryption of the addition of the two plaintexts underlying in ct_1 and ct_2. That is, $\mathsf{ct} = \mathsf{Enc}_{\mathsf{pk}}(\mathsf{m}_1 + \mathsf{m}_2)$ where $\mathsf{ct}_1 = \mathsf{Enc}_{\mathsf{pk}}(\mathsf{m}_1)$ and $\mathsf{ct}_2 = \mathsf{Enc}_{\mathsf{pk}}(\mathsf{m}_2)$.
- $\mathsf{Const.Mul}(\mathsf{ct}_1, c)$: The constant multiply algorithm takes as input a ciphertext ct_1 and a constant c. It outputs a new ciphertext ct such that $\mathsf{ct} = \mathsf{Enc}_{\mathsf{pk}}(c \cdot \mathsf{m})$ where $\mathsf{ct}_1 = \mathsf{Enc}_{\mathsf{pk}}(\mathsf{m})$.

Correctness and security are defined similarly as in any public key encryption scheme.

Definition 11. ***(Rate:)*** *We say that a linear homomorphic encryption encryption scheme has a rate* c *if, for any security parameter* λ*, there exists* d *such that for all plaintext messages* m *of length* $s(\lambda) > \lambda^d$, $\frac{|\mathsf{m}|}{|\mathsf{Enc}_{\mathsf{pk}}(\mathsf{m})|} > c$.

We will be interested in high rate (> 0.5) linearly homomorphic encryption schemes. Damgard and Jurik [DJ01] show how to build such an encryption scheme based on the N^{th} Residuosity Assumption where the size of the ciphertext is only λ more than the size of the plaintext being encrypted.