Distinguisher-Dependent Simulation in Two Rounds and its Applications

Abhishek Jain * Yael Tauman Kalai † Dakshita Khurana ‡ Ron Rothblum §

Abstract

We devise a novel simulation technique that makes black-box use of the adversary as well as the distinguisher. Using this technique we construct several round-optimal protocols, many of which were previously unknown even using non-black-box simulation techniques:

- Two-round witness indistinguishable (WI) arguments for NP from different assumptions than previously known.
- Two-round arguments and three-round arguments of knowledge for NP that achieve strong WI, witness hiding (WH) and distributional weak zero knowledge (WZK) properties in a setting where the instance is only determined by the prover in the last round of the interaction. The soundness of these protocols is guaranteed against adaptive provers.
- Three-round two-party computation satisfying input-indistinguishable security as well as a weaker notion of simulation security against malicious adversaries.
- Three-round extractable commitments with guaranteed correctness of extraction from polynomial hardness assumptions.

Our three-round protocols can be based on DDH or QR or N^{th} residuosity and our two-round protocols require quasi-polynomial hardness of the same assumptions. In particular, prior to this work, two-round WI arguments for NP were only known based on assumptions such as the existence of trapdoor permutations, hardness assumptions on bilinear maps, or the existence of program obfuscation; we give the first construction based on (quasi-polynomial) DDH or QR or N^{th} residuosity.

Our simulation technique bypasses known lower bounds on black-box simulation [Goldreich-Krawcyzk'96] by using the distinguisher's output in a meaningful way. We believe that this technique is likely to find additional applications in the future.

*Department of Computer Science, Johns Hopkins University, Baltimore, USA. Supported in part by a DARPA/ARL Safeware Grant W911NF-15-C-0213.

†Microsoft Research, Cambridge, USA.

‡Department of Computer Science, UCLA, USA.

§Department of Computer Science, MIT, Cambridge, USA. Partially supported by the grants: NSF MACS - CNS-1413920, DARPA IBM - W911NF-15-C-0236 and SIMONS Investigator award Agreement Dated 6-5-12.

Contents

1 Introduction

The notion of zero-knowledge (ZK) proofs [39] is fundamental to cryptography. Intuitively, zero-knowledge proofs guarantee that the proof of a statement does not reveal anything beyond the validity of the statement. This seemingly paradoxical requirement is formalized via the *simulation* paradigm, namely, by requiring the existence of an efficient simulator that simulates the view of a malicious verifier, without access to any witness for the statement.

Over the years, ZK proofs (and arguments) have been integral to the design of numerous cryptographic protocols, most notably general-purpose secure computation [37], as well as specific tasks such as coin-tossing, equivocal and/or extractable commitments and non-malleable protocols [25]. Even protocols satisfying weaker notions of ZK such as strong witness indistinguishability and witness hiding (WH)[29], are typically constructed only via a ZK protocol[1]. In particular, the round complexity of ZK determines the round complexity of known constructions for these tasks.

Goldreich and Krawcyzk (GK) [36] established that three round ZK arguments for NP with black-box simulation do not exist for languages outside BPP. Furthermore, all known non-black-box simulation techniques [3] require more than three rounds.[2] This has acted as a barrier towards achieving round-efficient protocols for many of the aforementioned tasks. In this work, we investigate the possibility of overcoming this barrier.

(When) Is ZK Necessary? ZK proofs are typically used to enforce "honest behaviour" for participants of a cryptographic protocol. The zero-knowledge property additionally ensures privacy of the inputs of honest parties. However, many applications of ZK described above do not themselves guarantee simulation-based security but only weaker indistinguishability-based security. As such, it is not immediately clear whether the "full" simulation power of ZK is necessary for such applications.

For example, *strong witness indistinguishability* requires that for two indistinguishable statement distributions $\mathcal{X}_1, \mathcal{X}_2$, a proof (or argument) for statement $x_1 \leftarrow \mathcal{X}_1$ must be indistinguishable from a proof (or argument) for statement $x_2 \leftarrow \mathcal{X}_2$. All known constructions of strong witness indistinguishable protocols rely on ZK arguments with standard simulation – and therefore end up requiring at least as many rounds as ZK arguments. Similar issues arise in constructing input-hiding/input-indistinguishable secure computation, witness hiding arguments and proofs, and extractable (or other sophisticated) commitment schemes. However, it is unclear whether ZK is actually *necessary* in these settings.

This raises the question of whether it is possible to devise "weaker" simulation strategies in three rounds or less that can be used to recover several applications of ZK. In this work, we implement such a black-box simulation strategy in only *two* rounds.

Distinguisher-Dependent Simulation. Our starting observation is that for any cryptographic protocol that only aims to achieve indistinguishability-based security, the security reduction has access to an efficient *distinguisher*. In such scenarios, one can hope to argue security via a (weaker) simulation strategy that potentially makes use of the distinguisher in a non-trivial manner.

The idea of distinguisher-dependent simulation is not new and has previously been studied in the context of interactive proofs, where it is referred to as weak zero knowledge (WZK) [28][3]. Informally, WZK says that any bit of information that can be learned by the verifier by interacting with the prover can be simulated given only the instance. As such, WZK suffices for many applications of ZK, and in particular, implies meaningful weaker notions such as WH and WI [29].

[1] The work of Bitansky and Paneth [10] constructing 3 round witness-hiding and weak zero-knowledge from variants of auxiliary-input point obfuscation, is an exception.

[2] Here we only refer to *explicit* simulation, and not non-explicit simulation via knowledge assumptions [42, 5].

[3] Recall that standard ZK requires that for any adversarial verifier, there exists a simulator that can produce a view that is indistinguishable from the real one to every distinguisher. WZK relaxes this notion by reversing the order of quantifiers, and allowing the simulator to depend on the distinguisher.

The immediate question is whether distinguisher-dependent simulation can be realized in three rounds or less. At first, the answer seems to be negative since the lower bound of GK also extends to WZK (this was already noted in [10]).

A key insight in our work is that in many applications of ZK proofs, the statement being proven is chosen by the prover from a (public) distribution. Suppose that the proof system is *delayed-input* [49], namely, where the instance and witness are only required for computing the last prover message. In this case, it is to an honest prover's advantage to reveal the instance to the verifier only in the last round. This does not violate correctness due to the delayed input property, but "weakens" a malicious verifier, and in particular, ensures that a malicious verifier's messages are independent of the instance. Interestingly, we observe that the lower bound of GK no longer holds in this case[4].

At a high-level, this is because in this setting, a simulator may be able to learn non-trivial information about the distinguisher's behavior by observing its output on different samples created using possibly different instances from the same distribution. This observation is, in fact, not limited to delayed-input proofs and extends to a large class of important two-party functionalities including coin-tossing, generating common reference strings and oblivious PRFs.

This observation opens doors to the possibility of constructing proof systems and secure computation in three rounds or less with meaningful simulation-based and indistinguishability-based security guarantees.

A New Black-box Simulation Technique. We devise a new distinguisher-dependent black-box simulation technique that only requires two-rounds of communication. Roughly, we show that a single bit of information (of whether the proof is accepted or rejected by the distinguisher) can be used to learn information about the (possibly) malicious verifier and distinguisher, in a bit-by-bit fashion, and that this information can later be used to efficiently simulate the proof.

We remark that the ability to learn a bit of information based on whether the protocol execution is accepted or rejected has in the past been viewed as a source of insecurity in cryptographic protocols. For example, in the delegation of computation schemes of [33, 18], an adversarial prover can successfully cheat if it is able to observe the verifier's output over multiple executions. For similar reasons, special care is taken to prevent "input-dependent aborts" in the design of many secure computation protocols.

In this work, we turn this apparent weakness into a positive by using it to devise a new black-box simulation strategy. Using this strategy, we obtain several new results on proof systems and secure computation. Most of our results were previously unknown even using non-black-box simulation techniques.

Our Setting. In order to prove privacy, we must sometimes restrict ourselves to a setting where the prover has the *flexibility* to sample instances and witnesses in the last round of the argument. More specifically, our simulator will require knowledge of any witnesses that are fixed (implicitly or explicitly) before the last message is sent; however, it will not require knowledge of witnesses fixed in the last round.

1.1 Our Results

We now proceed to describe our results. We start with our results on interactive proof systems and then describe their applications to secure two-party computation and extractable commitment schemes. All of these results rely on our new black-box simulation strategy.

I. Delayed-Input Interactive Proofs. We study two and three round *delayed-input* interactive proof systems where the instance to be proven can be chosen by the prover in the last round, and soundness holds even against adaptive cheating provers who choose the instance depending upon the

[4]Indeed, the GK proof strategy crucially uses a verifier that chooses its protocol message as a function of the instance. See Section 1.2 for further discussion.

verifier's message. First studied by [49], delayed-input protocols have found numerous applications over the years in the design of round-efficient cryptographic protocols for a variety of tasks such as secure computation [48, 31, 45], resettable security [24, 61], non-malleable commitments [59, 20], improved Σ-protocols [21, 22, 51], and so on.

In the context of establishing various privacy notions, we consider both *adaptive* verifiers, who receive the instance at the beginning of the protocol, and hence may choose their message based on this instance, and *non-adaptive* verifiers, who receive the instance only in the last round of the protocol, and hence their message is independent of the instance. As we discuss later, guaranteeing privacy against non-adaptive verifiers suffices for many natural applications of delayed-input proof systems.

(i). Two Round Argument Systems. Our first contribution is a two-round delayed-input argument system that achieves witness-indistinguishability (WI) against *adaptive* verifiers, and strong WI, witness hiding (WH) and distributional weak zero-knowledge (WZK) against *non-adaptive* verifiers.

Theorem 1 (Informal)**.** *Assuming the existence of two-round oblivious transfer that is secure against malicious PPT receivers and quasi-polynomial time semi-honest senders, there exists a two-round delayed-input interactive argument system for* NP *with adaptive soundness and the following privacy guarantees:*

- *WI against adaptive verifiers.*
- *Strong WI, WH and distributional WZK against non-adaptive verifiers.*

Oblivious transfer (OT) protocols as required in the above theorem can be constructed based on quasi-polynomial hardness of Decisional Diffie-Hellman (DDH) [52] or N'th Residuosity or Quadratic Residuosity [46, 44].

Comparison with Prior Work. If we know an a priori super-polynomial bound on the hardness of the language, then two-round WH can be obtained from two-round ZK with super-polynomial time simulators (SPS) [54]. However, no constructions of two-round WH or distributional WZK for NP against non-uniform verifiers were previously known. (We refer the reader to Section 1.3 for a more thorough discussion.)

WI proofs in two rounds (or less) were previously only known based on either trapdoor permutations[5] [27], or the decision linear assumption on bilinear groups [41], or indistinguishability obfuscation [11]. Our result in Theorem 1 substantially adds to the set of standard assumptions that suffice for two-round WI. We remark that unlike previous protocols, our WI protocol is not publicly verifiable.

Privacy Amplification via Round Compression. We obtain Theorem 1 by "compressing" any Σ-protocol[6] [23] into a two-round private-coin argument using OT. Our compiler follows the approach of [1, 47], except that we use a maliciously secure OT as opposed to a computational PIR [17].

Interestingly, our approach of compressing a Σ-protocol into a two-round argument results in amplifying its privacy guarantees. Indeed, standard Σ-protocols are not known to be WZK. Furthermore, [43, 55] proved that such protocols cannot be proven WH using black-box reductions.

Avoiding NP *Reductions.* An added benefit of our approach is that given a Σ-protocol for a language L, we obtain a two-round private-coin argument system with the security guarantees stated in Theorem 1 for the same language L, *without* using expensive NP reductions. To the best of our knowledge, no such two-round argument system was previously known.

[5] Presently, the only known candidates for trapdoor permutations are based on factoring or indistinguishability obfuscation [12, 32].

[6] Very roughly, a Σ-protocol is a three round protocol that is honest verifier zero-knowledge, and has a strong soundness guarantee. We refer the reader to Definition 1.

(ii). Three Round Arguments of Knowledge. Our second contribution is a three-round delayed-input interactive *argument of knowledge* system that achieves WH and distributional WZK against non-adaptive verifiers. This protocol uses only polynomial assumptions, but requires an extra round.

Theorem 2 (Informal)**.** *Assuming the existence of two-round oblivious transfer (OT) that is secure against malicious PPT receivers and semi-honest PPT senders, as well as dense cryptosystems, there exists a three-round interactive argument of knowledge for* NP *that achieves soundness against adaptive (unbounded) provers and Strong WI, WH and distributional WZK against non-adaptive PPT verifiers.*

Comparison with Prior Work. Three-round ZK arguments are known either based on non-standard "knowledge assumptions" [42, 5], or against adversaries with *bounded* non-uniformity [9, 7]. In this work, we consider security against adversaries with non-uniform advice of arbitrarily large polynomial length, based on standard cryptographic assumptions. Prior to our work, three-round WH and WZK arguments for NP were known from non-black-box techniques that rely on auxiliary input point obfuscation assumptions [10]. These protocols, unlike ours, guarantee privacy also against adaptive verifiers. However, some of their underlying assumptions have recently been shown to be implausible [14, 6]. (See Section 1.3 for a more detailed discussion.)

II. Secure Two-Party Computation. We next study two-party computation against malicious adversaries in the plain model without trusted setup assumptions. In this setting, the state of the art result is due to Katz and Ostrovsky [48] who constructed a four-round protocol for general functions in the setting where only one party receives the output. We refer to the output recipient as the *receiver* and the other party as the *sender*.

As an application of our new simulation technique, we obtain two new results on two-party computation in *three* rounds. Our first result achieves input-indistinguishable security [50] against malicious receivers, while our second result achieves distinguisher-dependent simulation security against malicious receivers. In both of these results, we achieve standard simulation security against malicious senders. We elaborate on these results below.

(i). Three Round Input-Indistinguishable Computation. The notion of input-indistinguishable computation (IIC) was introduced by Micali, Pass and Rosen [50] as a weakening of standard simulation-based security notion for secure computation while still providing meaningful security. (See also [30, 53].) Roughly, input-indistinguishable security against malicious receivers guarantees[7] that for any function f and a pair of inputs (x_1, x_2) for the sender, a malicious receiver cannot distinguish whether the sender's input is x_1 or x_2 as long as the receiver's "implicit input" y in the execution is such that $f(x_1, y) = f(x_2, y)$.[8]

We construct the first three-round IIC protocol for general functions based on polynomial hardness assumptions. In fact, our protocol achieves standard simulation-based security against malicious senders and input-indistinguishable security against malicious receivers.

Theorem 3 (Informal)**.** *Assuming the existence of two-round oblivious transfer that is secure against malicious PPT receivers and semi-honest PPT senders, along with dense cryptosystems, there exists a three-round secure two-party computation protocol for general functions between a sender and a receiver, where only the receiver obtains the output, with standard simulation security against malicious senders and input-indistinguishable security against malicious receivers.*

(ii). Three Round Two-Party Computation with Distinguisher Dependent Simulation. We also consider a weak simulation-based security notion for two-party computation that is defined

[7] Security against malicious senders can be defined analogously.

[8] The formal security definition of IIC is much more delicate, and we refer the reader to the technical sections for details.

analogously to distributional WZK by allowing the simulator to depend (non-uniformly) upon the distinguisher and the distribution over the public input to the adversary. We refer to this as distributional distinguisher-dependent simulation secure two-party computation. While this generalizes the notion of distributional WZK, it also implies distinguisher-dependent simulation security for all functionalities where the honest party's input can be efficiently sampled (without the need for non-uniform advice) even if the input of the malicious party and any common input is already fixed.

We show that the same protocol as in Theorem 3 also satisfies distributional distinguisher-dependent security for all functionalities. In particular, we obtain three round distinguisher-dependent simulation secure two party computation for inherently distributional functionalities such as coin-tossing, generating common *reference* strings and oblivious PRFs.

Theorem 4 (Informal)**.** *Assuming the existence of two-round oblivious transfer that is secure against malicious PPT receivers and semi-honest PPT senders, as well as dense cryptosystems, there exists a three-round protocol for secure two-party computation for any function between a sender and receiver, where only the receiver obtains the output, with standard simulation security against a malicious sender and distributional distinguisher-dependent simulation security against a malicious receiver. This implies distinguisher-dependent simulation secure two-party computation for any function where the sender's input can be efficiently sampled even if the receiver's input (and any common input) is already fixed.*

A Two-round Protocol. We also remark that our three-round two-party computation protocol can be downgraded to a two-round protocol that achieves distributional distinguisher-dependent simulation security or input-indistinguishable security against malicious receivers and quasi-polynomial time simulation security against malicious senders (or polynomial-time simulation security against semi-honest senders).

Outputs for Both Parties. Theorem 3 and Theorem 4 consider the case where only one party, namely the receiver, learns the output. As observed in [48], such a protocol can be easily transformed into one where both parties receive the output by computing a modified functionality that outputs signed values. Now the output recipient can simply forward the output to the other party who accepts it only if the signature verifies.

This adds a round of communication, making the protocol four rounds in total. Because we consider distinguisher-dependent simulation security (or input-indistinguishable security), this bypasses the lower bound of [48] who proved that coin-tossing cannot be realized with standard simulation-based security in less than five rounds when both parties receive output.

III. Extractable Commitments. We finally discuss application of our techniques to *extractable* commitments. A commitment scheme is said to be extractable if there exists a PPT extractor that can extract the committed value with *guaranteed correctness of extraction.* In particular, if the commitment is not "well-formed" (i.e., not computed honestly), then the extractor must output $\perp$, while if the commitment is well-formed, then the extractor must output the correct committed value. Extractable commitments are very useful in the design of advanced cryptographic protocols, in particular, to facilitate the extraction of the adversary's input in tasks such as secure computation, non-malleable commitments, etc.

A standard way to construct extractable commitment schemes is to "compile" a standard commitment scheme with a ZKAoK, namely, by having a committer commit to its value using a standard commitment and additionally give a ZKAoK to prove knowledge of the decommitment value. The soundness property of ZKAoK guarantees the well-formedness of commitment, which in turn guarantees correctness of extraction of the committed value using the AoK extractor for ZKAoK, while the ZK property preserves the hiding of the underlying commitment. This approach yields a four round extractable commitment scheme starting from any four round ZKAoK. However, in the absence

of three-round ZKAoK, constructing three-round extractable commitments from *polynomial* hardness assumptions have so far proven to be elusive.[9]

The main challenge here is to enforce honest behavior on a malicious committer, while at the same time guaranteeing privacy for honest committers. Indeed, natural variations of the above approach (e.g., using weaker notions such as WIPOK that are known in three rounds) seem to only satisfy one of these two requirements, but not both.

As an application of Theorem 2, we construct the first three-round extractable commitment scheme based on standard polynomial-time hardness assumptions.

Theorem 5 (Informal)**.** *Assuming the existence of two-round oblivious transfer that is secure against malicious PPT receivers and semi-honest PPT senders, as well as dense cryptosystems, there exists a three-round extractable commitment scheme.*

Roughly, our construction of extractable commitments follows the same approach as described above. Our main observation is that the hiding property of the extractable commitment can be argued if the AoK system satisfies a *strong* WI property (instead of requiring full-fledged ZK).

1.2 Discussion

Non-adaptive Verifiers. Our results on distributional WZK, WH and strong WI are w.r.t. non-adaptive verifiers who learn the statement in the last round of the protocol. To the best of our knowledge, privacy against non-adaptive verifiers has not been studied before, and therefore, it is natural to ask whether it is a meaningful notion of privacy.

We argue that privacy against non-adaptive verifiers is very useful. Our main observation is that in many applications of delayed-input proof systems, the verifier is already non-adaptive, or can be made non-adaptive by design. Two concrete examples follow:

- We construct a three-round extractable commitment scheme by combining a standard commitment with a three-round delayed-input strong WIAoK of correctness of the committed value, that achieves security against non-adaptive verifiers. By sending the commitment in the last round, we automatically make the verifier non-adaptive.

- In secure computation using garbled circuits (GCs) [60], a malicious sender must prove correctness of its GC. In this case, the instance (i.e., the GC) can simply be sent together with the last prover message, which automatically makes the verifier non-adaptive. This does not affect the security of the receiver if the proof system achieves adaptive soundness (which is true for our constructions). Indeed, our construction uses exactly this approach.

We anticipate that the notion of privacy against non-adaptive verifiers will find more applications in the future.

Bypassing GK and GO Lower Bounds. We now elaborate on the reasons why we are able to bypass the lower bounds of [36] and [38]. The black-box impossibility result of [36] for three-round ZK crucially uses an adaptive verifier. More specifically, they consider a verifier that has a random seed to a pseudo-random function hard-wired into it, and for any instance and first message sent by the prover, it uses its PRF seed, to answer honestly with fresh-looking randomness. It is then argued that

[9] All known constructions of three-round extractable commitments from polynomial-hardness assumptions (such as [56, 57]) only satisfy a weak extraction property where either the extractor outputs (with non-negligible probability) a non $\perp$ value when the commitment is not well-formed, or it fails to output the correct value when the commitment is well-formed. It is, however, possible to construct extractable commitments using quasi-polynomial hardness [40] or using three round zero-knowledge with super-polynomial simulation [54].

a black-box simulator can be used to break soundness. Very roughly, this is because a cheating prover can simply run the black-box simulator; if the simulator rewinds the verifier, then the cheating prover answers it with a random message on behalf of the verifier. This proof also extends to WZK because any query made by the simulator to the distinguisher can simply be answered with "reject."

Note, however, that in the non-adaptive setting, the verifier is not allowed to generate different messages for different instances, and hence the simulator has more power than a cheating prover, since it can fix the first message of the prover and then test whether the distinguisher accepts or not with various instances and various third round messages. Indeed, we exploit exactly this fact to design a distinguisher-dependent simulator for our protocols.

We next explain why we are able to overcome the lower bound of [38] for two-round ZK. A key argument in the proof of [38] is that no (possibly non-black-box) simulator can simulate the prover's message for a false statement (even when the protocol is privately verifiable). For ZK, this is argued by setting the verifier's auxiliary input to be an honestly generated first message and providing the corresponding private randomness to the distinguisher, who is chosen *after* the simulator. Now, if the simulator succeeds, then we can break soundness of the protocol. However, in WZK, since the distinguisher is fixed in advance, the above approach does not work. In particular, if the distinguisher is given the private randomness then the simulator is given it as well (and hence can simulate), and otherwise, the simulator can succeed by simulating a rejecting transcript.

1.3 Related Work

Concurrent Work. Concurrent to our work, Badrinarayanan et al. [2] construct protocols that are similar to our two-round protocols. However their focus is on super-polynomial simulation, whereas we focus on polynomial time distinguisher-dependent simulation. They also give other instantiations of two-round OT, which can be combined with our results to obtain two-round delayed-input distributional weak zero-knowledge from additional assumptions.

Proof Systems. We mention two related works on two-round ZK proofs that overcome the lower bound of [38] in different ways. A recent work of [19] constructs a two-round (T, t, ϵ)-ZK proof system for languages in statistical zero-knowledge, where roughly, (T, t, ϵ) ZK requires the existence of a simulator that simulates the view of the verifier for any distinguisher running in time t and distinguishing probability ϵ. The running time T of the simulator depends upon t and ϵ. In another recent work, [9] construct a two-round ZK argument system against verifiers with auxiliary inputs of a priori bounded size.

Three-round ZK proofs are known either based on non-standard "knowledge assumptions" [42, 5], or against adversaries that receive auxiliary inputs of a priori bounded size [9, 7]. In contrast, in this work, we consider security against adversaries with non-uniform advice of arbitrarily polynomial size, based on standard cryptographic assumptions.

Finally, we discuss WI, WH and WZK in three rounds. While three round WI is known from injective one-way functions [29], WH and WZK are non-trivial to realize even in three rounds. In particular, [43] proved a lower bound for three-round public-coin WH w.r.t. a natural class of black-box reductions. More recently, [55] extended their result to rule out all black-box reductions. Presently, the only known constructions of three-round WH and WZK for NP require either "knowledge assumptions" [42, 5], or rely on the assumption of auxiliary-input point obfuscation (AIPO) and auxiliary-input multi-bit point obfuscation (AIMPO), respectively, with an additional "recognizability" property [10]. For general auxiliary inputs, however, AIMPO was recently proven to be impossible w.r.t. general auxiliary inputs [14], assuming the existence of indistinguishability obfuscation [4]. Further, one of the assumptions used by [10] to build recognizable AIPO, namely, strong DDH assumption [15], was recently shown to be

impossible w.r.t. general auxiliary inputs [6], assuming the existence of virtual grey-box obfuscation [8].

Secure Computation. Katz and Ostrovsky [48] constructed a four-round two-party computation protocol for general functions where only one party receives the output. A recent work of Garg et al. [31] extends their result to the simultaneous-message model to obtain a four-round protocol where both parties receive the outputs.

The notion of input-indistinguishable computation (IIC) was introduced by Micali, Pass and Rosen [50] as a weakening of standard simulation-based security notion for secure computation while still providing meaningful security. (See also [30, 53].) We provide the first three-round protocol that provides input-indistinguishable security.

A recent work of Döttling et al. [26] constructs a two-round two-party computation protocol for oblivious computation of cryptographic functionalities. They consider semi-honest senders and malicious receivers, and prove game-based security against the latter. We remark that our three-round two-party computation protocol can be easily downgraded to a two-round protocol that achieves weak simulation security against malicious receivers and super-polynomial time simulation security against malicious senders (or polynomial-time simulation against semi-honest senders). We note that our result is incomparable to [26], because we consider a restricted class of distributions (such as product distributions), albeit any functionality, whereas [26] considers the class of cryptographic functionalities.

1.4 Organization

The rest of this paper is organized as follows. We begin with an overview of our techniques in Section 2. In Section 3, we describe important relevant preliminaries including Σ-protocols and oblivious transfer. In Section 4, we recall definitions of adaptive soundness, witness indistinguishability, distributional weak-ZK and witness hiding against non-adaptive verifiers. In Section 5, we describe our two-round protocol, which uses any Σ-protocol with a special structure, together with 2-message OT. In the same section, we describe how to modify our protocol so as to rely on *any* Σ-protocol, and also show how to base security on polynomial hardness assumptions at the cost of adding an extra round. Due to lack of space, we defer additional details of our three round protocols and their applications to the full version of the paper.

2 Technical Overview

We now give an overview of our main ideas and techniques.

2.1 Argument Systems

We construct a two-round argument system, which we prove is both witness indistinguishable (against all malicious verifiers), and is distributional ϵ-weak zero-knowledge (against non-adaptive malicious verifiers). Our protocol makes use of two components:

- Any Σ-protocol consisting of three messages (a, e, z) that is secure against unbounded provers,
- Any two-message oblivious transfer protocol, denoted by $(\mathsf{OT}_1, \mathsf{OT}_2)$, which is secure against malicious PPT receivers, and malicious senders running in time at most $2^{|z|}$. For receiver input b and sender input messages (m_0, m_1), we denote the two messages of the OT protocol as $\mathsf{OT}_1(b)$ and $\mathsf{OT}_2(m_0, m_1)$. We note that $\mathsf{OT}_2(m_0, m_1)$ also depends on the message $\mathsf{OT}_1(b)$ sent by the receiver. For the sake of simplicity, we omit this dependence from the notation.

For simplicity, throughout most of the paper, we assume that the Σ-protocol is a parallel repetition of Σ-protocols with a single-bit challenge and constant soundness[10]. Namely, we assume that the Σ-protocol contains three messages, denoted by (a, e, z) and that these messages can be parsed as $a = (a_1, \ldots, a_\kappa)$, $e = (e_1, \ldots, e_\kappa)$, and $z = (z_1, \ldots, z_\kappa)$, where for each $i \in [\kappa]$, the triplet (a_i, e_i, z_i) are messages corresponding to an underlying Σ-protocol with a single-bit challenge (i.e., where $e_i \in \{0, 1\}$). We denote by f_1 and f_2 the functions that satisfy $a_i = f_1(x, w; r_i)$ and $z_i = f_2(x, w, r_i, e_i)$, for answers provided by the honest prover, and where r_i is uniformly chosen randomness.

We show how to convert any such Σ-protocol into a two-round protocol (P, V) using OT. Our transformation is essentially the same as the one suggested by Aeillo et. al. [1], and used by Kalai and Raz [47], to reduce rounds in interactive protocols, except that we use an OT scheme rather than a computational PIR scheme (since as opposed to [1, 47] we are not concerned with compressing the length of the messages). Specifically, given any such Σ-protocol and OT protocol, our two-round protocol (P, V), proceeds as follows.

- For $i \in [\kappa]$, V picks $e_i \xleftarrow{\$} \{0, 1\}$, and sends $\mathsf{OT}_{1,i}(e_i)$ in parallel. Each e_i is encrypted with a fresh OT instance.
- For $i \in [\kappa]$, P computes $a_i = f_1(x, w; r_i), z_i^{(0)} = f_2(x, w, r_i, 0), z_i^{(1)} = f_2(x, w, r_i, 1)$. The prover P then sends $a_i, \mathsf{OT}_{2,i}(z_i^{(0)}, z_i^{(1)})$ in parallel for all $i \in [\kappa]$.
- The verifier V recovers $z_i^{(e_i)}$ from the OT, and accepts if and only if for every $i \in [\kappa]$, the transcript $(a_i, e_i, z_i^{(e_i)})$ is an accepting transcript of the underlying Σ-protocol.

Soundness. It was proven in [47] that such a transformation from any public-coin interactive proof to a two-round argument preserves soundness against PPT provers. We extend their proof to show that the resulting two-round protocol also satisfies *adaptive* soundness, i.e., is sound against cheating provers that may adaptively choose some instance x as a function of the verifier message.

To prove soundness, we rely on the following special-soundness property of Σ-protocols: There exists a polynomial-time algorithm A that given any instance x of some NP language L with witness relation R_L, and a pair of accepting transcripts $(a, e, z), (a, e', z')$ for x with the same first prover message, where $e \neq e'$, outputs w such that $w \in R_L(x)$. In particular, this means that for any $x \notin L$, for any fixed message a, there exists at most *one* unique value of receiver challenge e, for which there exists z such that (a, e, z) is an accepting transcript (as otherwise the algorithm A would output a witness $w \in R_L(x)$, which is impossible).

Going back to our protocol – suppose a cheating prover, on input the verifier message $\mathsf{OT}_1(e^*)$, outputs $x^* \notin L$, together with messages $a^*, \mathsf{OT}_2(z^*)$, such that the verifier accepts with non-negligible probability. Since, for any $x^* \notin L$ and any a^*, there exists at most one unique value of receiver challenge e, for which there exists a z that causes the verifier to accept – intuitively, this means that a^* encodes the receiver challenge e^*.

Thus, for fixed a^*, a reduction can enumerate over all possible values of z (corresponding to all possible e), and check which single e results in an accepting transcript. Then, this would allow a reduction to break receiver security of the oblivious transfer. Since such a reduction would require time at least $2^{|z|}$, we need the underlying oblivious transfer to be $2^{|z|}$-secure (or, sub-exponentially secure). If z can be scaled down to be of size poly-logarithmic in the security parameter, we can rely on an oblivious transfer protocol which is quasi-polynomially secure against malicious receivers.

A New Extraction Technique for Proving Weaker Notions of Zero-Knowledge. We now proceed to describe our main ideas for proving the privacy guarantees of our protocol. For simplicity,

[10] We later describe how garbled circuits can be used in order to modify our construction to work with any Σ-protocol.

consider a single repetition of the protocol outlined above. That is, consider a protocol where the verifier picks a random **bit** $e \xleftarrow{\$} \{0,1\}$ and sends $r = \mathsf{OT}_1(e)$ to the prover. The prover then sends $a, \mathsf{OT}_2(z^{(0)}, z^{(1)})$ to the verifier, where $(a, z^{(0)}, z^{(1)})$ are computed similarly as before.

By the security of the underlying OT scheme against malicious receivers (see Definition 2 and discussion therein), the following holds: For any malicious verifier (i.e. malicious receiver of the OT scheme) there exists a (possibly inefficient) simulator that interacts with an ideal OT functionality and is able to simulate the view of the verifier. This means that for any PPT distinguisher $\mathcal{D}_\mathcal{V}$ (that obtains as input the view of the verifier and additional auxiliary information), its output distribution when the prover sends $(a, \mathsf{OT}_2(z^{(0)}, z^{(1)}))$ is indistinguishable from one of the following:

- Its output distribution when the prover sends $(a, \mathsf{OT}_2(z^{(0)}, z^{(0)}))$ (implicitly corresponding to receiver choice bit 0).
- Its distribution output when the prover sends $(a, \mathsf{OT}_2(z^{(1)}, z^{(1)}))$ (implicitly corresponding to receiver choice bit 1).

Suppose the message of the verifier, $\mathsf{OT}_1(e)$ is generated independently of the instance x, and suppose that the instance x is generated according to some distribution $\mathcal{D}$. Then an extractor $\mathcal{E}$, given the message $\mathsf{OT}_1(e)$, can guess e (if the distinguisher "knows" e), up to ϵ-error in time $\mathsf{poly}(1/\epsilon)$, as follows: The extractor will generate $\mathsf{poly}(1/\epsilon)$ many instance-witness pairs $(x, w) \in R_L$, where each x is distributed independently from $\mathcal{D}$ ($\mathcal{E}$ will have these instance-witness pairs hardwired if they are hard to sample). Then for each such instance-witness pair the extractor will generate $(a, z^{(0)}, z^{(1)})$, and will observe the distinguisher's output corresponding to the prover's message $(a, \mathsf{OT}_2(z^{(0)}, z^{(0)}))$, $(a, \mathsf{OT}_2(z^{(1)}, z^{(1)}))$, and $(a, \mathsf{OT}_2(z^{(0)}, z^{(1)}))$. If the distinguisher cannot distinguish between these three distributions then the extractor outputs $\perp$ (indicating that the distinguisher does not know e). If the extractor outputs $\perp$, the distinguisher is (distributionally) insensitive to the prover's response, so we can behave as if it was approximated to 0.

However, if the distinguisher can distinguish between $(a, \mathsf{OT}_2(z^{(0)}, z^{(1)}))$ and $(a, \mathsf{OT}_2(z^{(b)}, z^{(b)}))$, then the distinguisher will guess $e = 1 - b$. In this way, the extractor can approximate (up to ϵ-error) whether the implicit receiver choice bit is 0 or 1, while running in time $\mathsf{poly}(1/\epsilon)$. This idea forms the basis of our new extraction technique.

Witness Indistinguishability. Since witness indistinguishability is known to compose under parallel repetition, it suffices to prove WI for a single repetition of the protocol outlined above. In fact, we will try to prove something even stronger.

As explained above, there exists a distinguisher-dependent simulator $\mathsf{Sim}_{\mathcal{D}_V}$, that, given a fixed receiver message r, can try to approximate the verifier's implicit challenge bit e, by observing the distinguisher's output corresponding to various sender messages, up to error ϵ. Once $\mathsf{Sim}_{\mathcal{D}_V}$ has successfully extracted the verifier's challenge, it can use the honest-verifier zero-knowledge simulator of the underlying Σ-protocol.

Of course, to even begin the extraction process, $\mathsf{Sim}_{\mathcal{D}_V}$ needs to observe the output of the distinguisher on $(a, \mathsf{OT}_2(z^{(0)}, z^{(1)}))$. However, even computing $(a, \mathsf{OT}_2(z^{(0)}, z^{(1)}))$ correctly, requires access to a witness! This is because a correctly compute tuple $(a, z^{(0)}, z^{(1)})$ actually *encodes a witness*.

In the case of witness indistinguishability, this is not a problem – since an "intermediate" simulator for witness indistinguishability has access to both witnesses in question, and therefore *can* generate valid messages $(a, \mathsf{OT}_2(z^0, z^1))$ using both witnesses. It can use these transcripts to learn the verifier's challenge bit, and then use the bit it learned, to generate a simulated transcript for the same receiver message r (where the simulated transcript uses neither of the two witnesses). We mainly rely on OT security to show that the distinguisher $\mathcal{D}_V$ cannot distinguish between the view generated by such a

simulator $\mathsf{Sim}_{\mathcal{D}_V}$ and the real view of the verifier, when he interacts with an honest prover that uses only one of the witnesses.

There are additional subtleties in the proof, for instance, in ensuring that the extracted values when the simulator uses one particular witness for learning, do not contradict the values extracted when it uses the other witness. We refer the reader to Section 5.3 for a detailed proof.

Distributional Weak Zero-Knowledge. We prove that the same protocol satisfies distributional weak zero-knowledge against non-adaptive verifiers (which can also be easily seen to imply witness-hiding against non-adaptive verifiers). Distributional weak zero-knowledge is a "distributional" relaxation of the standard notion of zero-knowledge where the simulator is additionally allowed to depend on the distribution of instances, and on the distinguisher. This notion roughly requires that for every distribution $\mathcal{X}$ over instances, every verifier V and distinguisher $\mathcal{D}_V$ that obtains the view of V, every $\epsilon = \frac{1}{\mathsf{poly}(\kappa)}$ for some polynomial $\mathsf{poly}(\cdot)$, there exists a simulator $\mathsf{Sim}_{\mathcal{D}_V}$ that runs in time $\mathsf{poly}(1/\epsilon)$ and outputs a view, such that the distinguisher $\mathcal{D}_V$ has at most ϵ-advantage in distinguishing the real view of V from the simulated view.

Fix the first message of the verifier (since the verifier is non-adaptive, this is fixed independently of the instance). The simulator $\mathsf{Sim}_{\mathcal{D}_V}$ obtains as (non-uniform) advice, $\mathsf{poly}(1/\epsilon)$ *randomly chosen* instance-witness pairs from the distribution in question.[11] It then uses these pairs together with the extraction strategy $\mathcal{E}$ described above, to "learn" an approximation to the verifier's implicit challenge string in the fixed verifier message. However, distributional weak zero-knowledge is not known to be closed under parallel composition. Therefore, we modify the simple extraction strategy described previously for a single repetition, so as to extract *all* bits of the verifier's challenge, while still remaining efficient in $\mathsf{poly}(1/\epsilon)$.

This is done inductively: at any time-step $i \in [\kappa]$, the simulator $\mathsf{Sim}_{\mathcal{D}_V}$ has extracted an approximation for the first $(i-1)$ bits of the verifier's challenge, and is now supposed to extract the i^{th} bit. At a high level, the extraction strategy of $\mathsf{Sim}_{\mathcal{D}_V}$ is as follows:

- It generates a "fake" output for the first $(i-1)$ parallel repetitions as follows: for $j \in [i-1]$, if the j^{th} bit of the verifier's challenge was approximated to 0, respond with $a_j, (z_j^0, z_j^0)$ in the j^{th} repetition (and similarly, if it was approximated to 1, respond with $a_j, (z_j^1, z_j^1)$).
- For all $j \in [i+1, \kappa]$ it responds honestly with $a_j, (z_j^0, z_j^1)$ in the j^{th} repetition.
- With outputs for all $j < i$ set to "fake" according to approximated challenge, and for all $j > i$ set to honest, at $j = i$, $\mathsf{Sim}_{\mathcal{D}_V}$ uses the extraction strategy $\mathcal{E}$ described above. That is, for $j = i$, it sets the output to $a_i, (z_i^0, z_i^1)$, $a_i, (z_i^0, z_i^0)$, and $a_i, (z_i^1, z_i^1)$, and checks whether the output of the distinguisher when given inputs corresponding to $a_i, \mathsf{OT}_{2,i}(z_i^0, z_i^1)$ is close to its output when given inputs corresponding to $a_i, \mathsf{OT}_{2,i}(z_i^0, z_i^0)$ or to $a_i, \mathsf{OT}_{2,i}(z_i^i, z_i^i)$. It uses this to approximate the i^{th} bit of the verifier's challenge.

Via an inductive hybrid argument, we prove that with high probability, the approximation computed by $\mathsf{Sim}_{\mathcal{D}_V}$ has at most $\Theta(\epsilon)$-error when $\mathsf{Sim}_{\mathcal{D}_V}$ runs in time $\mathsf{poly}(1/\epsilon)$. Once $\mathsf{Sim}_{\mathcal{D}_V}$ has successfully extracted the verifier's challenge, it can use the honest-verifier zero-knowledge simulator of the underlying Σ-protocol as before.

Note that in order to perform extraction, the simulator is required to generate various $a_i, \mathsf{OT}_{2,i}(z_i^0, z_i^1)$ tuples, which it does using the instance-witness pairs it sampled or obtained as advice. $\mathsf{Sim}_{\mathcal{D}_V}$ then uses the challenge it extracted to generate fake proofs for various other $x \leftarrow \mathcal{X}$. Non-adaptivity of the verifier ensures that the simulator can, for a fixed verifier messages, generate proofs for several other

[11] In most cryptographic applications, and in all our applications, it is possible for the simulator to efficiently sample random instance-witness pairs from the distribution on its own, without the need for any non-uniform advice.

statements in the distribution while observing the output of the distinguisher. We refer the reader to Section 5.4 for a complete proof.

Three Round Protocols from Polynomial Hardness Assumptions. We also describe how quasi-polynomial assumptions can be avoided at the cost of an extra round. The need for quasi-polynomial assumptions in our two-round protocols is to guarantee soundness: roughly, we require that a cheating prover should be unable to "maul" the receiver's challenge while providing his message. In the two-round setting, this is achieved by ensuring (via complexity leveraging) that the security of receiver OT message is stronger than the security of the prover's response. Three rounds, however, give an opportunity to *rewind* and extract the value inside the prover's message, while relying on (polynomial) hiding of the receiver OT message.

We assume here that the first round of the Σ-protocol consists of commitments to certain values, and the third round consists of decommitments to a subset of these commitments, together with additional auxiliary information (for instance, the Blum protocol for Graph Hamiltonicity satisfies this requirement). We modify the protocol to have the prover send extractable commitments (instead of standard commitments) to commit to the values needed for the first round of the Σ-protocol.

Consider a PPT cheating prover that generates a proof for $x \notin L$. A reduction can obtain the receiver OT message externally as an encryption of some n-bit challenge, and then extract the values committed by the prover. Because the underlying Σ-protocol is special-sound against unbounded provers, any accepting proof for $x \notin L$, will allow recovering the receiver challenge directly by observing the values committed by the prover. We must, however, ensure that adding such extractable commitments does not harm privacy – since our simulator is required to generate several actual proofs before it is able to output a simulated proof. To accomplish this, we design a special kind of (weakly) extracting commitments, details of which can be found in Section 6. We note here that over-extraction suffices, in particular, we only care about extracting the values committed by provers that generate accepting transcripts.

2.2 Applications

We now describe some applications of our proof systems. As a first step, we describe a transformation from our three-round distributional WZK argument system to an *argument of knowledge*[12] (that retains the distributional weak ZK/strong WI property against non-adaptive verifiers).

Weak ZK/Strong WI Argument of Knowledge. We begin with the following simple idea for a distributional weak ZKAoK, for instances $x \leftarrow \mathcal{X}$: Let us use a delayed-input witness indistinguishable adaptive proof of knowledge (WIPoK), for instance the Lapidot-Shamir proof [49], to prove the following statement:

$$\text{Either } x \in L, \text{ OR, } \exists \text{ randomness } r \text{ such that } c = \mathsf{com}(1^\kappa; r).$$

Here, the commitment string c is also chosen and sent in the last round, together with instance x. Furthermore, to ensure that a (cheating) prover indeed uses the witness for $x \in L$, the prover must also give a weak ZK proof, for the same string c that $\exists r$ such that $c = \mathsf{com}(0^\kappa; r)$. The argument of knowledge property of this protocol now follows from the proof of knowledge property of WIPoK, the soundness of the weak ZK argument, and the statistical binding property of the commitment scheme. Specifically, by adaptive soundness of the weak ZK proof, c must indeed be constructed as a commitment to 0^κ; moreover, by the statistical binding property of the commitment scheme, the same string c cannot be a commitment to 1^κ. Therefore, the only possible witness that can be extracted from the WIPoK is indeed a witness for the instance x.

[12] Despite using a variant of extractable commitments, the three-round argument described in the previous section not a standard AoK in the delayed-input setting.

To prove weak ZK/strong WI property for the same protocol, we would ideally like to have the following sequence of hybrid arguments: First, we start simulating the weak ZK proof, by observing the output of the distinguisher on several different instances from the distribution $\mathcal{X}$, while using correct witnesses for these instances. We then use the information learned to simulate the weak ZK proof for c obtained externally in the main transcript. Since the string c is not used in the main thread at all, we change it so that $\mathsf{com}(0^\kappa; r)$ for uniformly random r. Next, we must begin using (c, r) as witnesses in the WIPoK, instead of using the witness for x.

It is in this step that there arises a subtle issue, because of the way our simulator works. In each experiment, before it can generate a simulated proof, it must first generate several real proofs for other random instances. We require the WIPoK to maintain witness indistinguishability, *even when the simulator provides multiple proofs for different instances using the same first two messages.* This is in general, not true for proof systems such as Lapidot-Shamir [49]. This is also not as strong a requirement as resettable-WI [16] since the verifier's message is fixed and remains the same for all proofs.

We refer to this property as *reusable* WI and construct an adaptively sound argument of knowledge satisfying this property. The argument of knowledge works by the prover sending two three-round extractable commitments (with "over" extraction) [56, 57] to random strings, encrypting the witness with each of these strings using standard private key encryption, and sending a three-round delayed-input reusable WI argument (this does not need to be an argument of knowledge, and could be instantiated with a ZAP, or with our three round arguments) to establish that one of the two commitments is a valid extractable commitment, and the corresponding ciphertext correctly encrypts the witness. The use of private key encryption gives us the additional desired property of reusability.

Extractable Commitments. Given the weak ZK argument of knowledge, our construction of three-round extractable commitments simply consists of sending a non-interactive statistically binding commitment to the message in the last round, together with a (distributional) weak ZK argument of knowledge to establish knowledge of the committed message and randomness. The weak ZK property helps prove hiding of this scheme, while the proof of knowledge property guarantees correct polynomial-time extraction, with overwhelming probability. We refer the reader to the full version for details.

Three Round, Two Party, Input-Indistinguishable Secure Computation. We begin by considering the following two-round protocol for two-party computation: The receiver generates OT messages corresponding to his inputs, together with the first message of a two-round weak ZK argument. Then, the sender generates garbled circuits corresponding to his own input labels, together with the second message of the two-round weak ZK argument.

This protocol already satisfies input-indistinguishable security against malicious receivers, as well as distinguisher-dependent security against malicious receivers, when an honest sender's input is sampled from some public distribution. Even though our weak ZK proof guarantees hiding against malicious receivers, security is not immediate. Indeed, we must first *extract* an adversarial receiver's input from his OT messages, and weak ZK does not help with that. Thus, apart from simulating the weak ZK, we must use our extraction strategy in this context, in order to (distributionally) learn the receiver's input.

In the full version, we describe applications of our techniques to obtaining input-indistinguishable secure computation, as well as distributional distinguisher-dependent secure computation in three rounds from polynomial assumptions. In particular, we also note that a large class of functionalities such as coin tossing, generating common *reference* strings, oblivious PRFs, etc. (that we call *independent-input functions*) are distributional by definition, and can be realized with distinguisher-dependent polynomial simulation security in three rounds.

3 Preliminaries

Throughout this paper, we will use κ to denote the security parameter, and $\mathsf{negl}(\kappa)$ to denote any function that is asymptotically smaller than $\frac{1}{\mathsf{poly}(\kappa)}$ for any polynomial $\mathsf{poly}(\cdot)$.

Definition 1 (Σ-protocols)**.** *Let $L \in \mathsf{NP}$ with corresponding witness relation R_L, and let x denote an instance with corresponding witness $w(x)$. A protocol $\Pi = (P, V)$ is a Σ-protocol for relation R_L if it is a three-round public-coin protocol, and the following requirements hold:*

- **Completeness:** $\Pr[\langle P(x, w(x)), V(x)\rangle = 1] = 1 - \mathsf{negl}(\kappa)$*, assuming P and V follow the protocol honestly.*
- **Special Soundness:** *There exists a polynomial-time algorithm A that given any x and a pair of accepting transcripts $(a, e, z), (a, e', z')$ for x with the same first prover message, where $e \neq e'$, outputs w such that $w \in R_L(x)$.*
- **Honest verifier zero-knowledge:** *There exists a probabilistic polynomial time simulator S_Σ such that*

  $$\{S_\Sigma(x, e)\}_{x \in L, e \in \{0,1\}^\kappa} \approx_c \{\langle P(x, w(x)), V(x, e)\rangle\}_{x \in L, e \in \{0,1\}^\kappa}$$

  where $S_\Sigma(x, e)$ denotes the output of simulator S upon input x and e, and $\langle P(x, w(x)), V(x, e)\rangle$ denotes the output transcript of an execution between P and V, where P has input (x, w), V has input x and V's random tape (determining its query) is e.

Definition 2 (Oblivious Transfer)**.** *Oblivious transfer is a protocol between two parties, a sender S with messages (m_0, m_1) and receiver R with input a choice bit b, such that R obtains output m_b at the end of the protocol. We let $\langle S(m_0, m_1), R(b)\rangle$ denote an execution of the OT protocol with sender input (m_0, m_1) and receiver input bit b. It additionally satisfies the following properties.*
Receiver Security. *For any sender S^*, all auxiliary inputs $z \in \{0,1\}^*$, and all $(b, b') \in \{0,1\}$, $\mathsf{View}_{S^*}(\langle S^*(z), R(b)\rangle) \approx_c \mathsf{View}_{S^*}(\langle S^*(z), R(b')\rangle)$.*
Sender Security. *This is defined using the real-ideal paradigm, and requires that for all auxiliary inputs $z \in \{0,1\}^*$, every distribution on the inputs (m_0, m_1) and any* adversarial *receiver R^*, there exists a (possibly unbounded) simulator Sim_{R^*} that interacts with an ideal functionality $\mathcal{F}_{\mathsf{ot}}$ on behalf of R^*. Here $\mathcal{F}_{\mathsf{ot}}$ is an oracle that obtains the inputs (m_0, m_1) from the sender and b from the Sim_{R^*} (simulating the malicious receiver), and outputs m_b to Sim_{R^*}. Then $\mathsf{Sim}_{R^*}^{\mathcal{F}_{\mathsf{ot}}}$ outputs a receiver view V_{Sim} that is computationally indistinguishable from the real view of the malicious receiver $\mathsf{View}_{R^*}(\langle S(m_0, m_1, z), R^*\rangle)$.*

We will make use of **two-message** oblivious-transfer protocols with security against malicious receivers and semi-honest senders. Such protocols have been constructed based on the DDH assumption [52], and a stronger variant of smooth-projective hashing, which can be realized from DDH as well as the N^{th}-residuosity and Quadratic Residuosity assumptions [46, 44]. Such protocols can also be based on indistinguishability obfuscation (iO) together with one-way functions [58].

We will use the following sender security property in our protocols (which is implied by the definition of sender security in Definition 2 above). For any fixed first message generated by a malicious receiver R^*, we require that either of the following statements is true:

- For all m_0, m_1, $\mathsf{View}_{R^*}\langle S(m_0, m_1, z), R^*\rangle) \approx_c \mathsf{View}_{R^*}(\langle S(m_0, m_0, z), R^*\rangle)$
- Or, for all m_0, m_1, $\mathsf{View}_{R^*}(\langle S(m_0, m_1, z), R^*\rangle) \approx_c \mathsf{View}_{R^*}(\langle S(m_1, m_1, z), R^*\rangle)$

This follows from the (unbounded) simulation property, i.e., there exists a simulator that extracts some receiver input b from the first message of R^*, sends it to the ideal functionality, obtains m_b and generates

an indistinguishable receiver view. Then, by the definition of sender security, the simulated view must be close to both $\mathsf{View}_{R^*}(\langle S(m_0, m_1, z), R^*\rangle)$, and $\mathsf{View}_{R^*}(\langle S(m_b, m_b, z), R^*\rangle)$.

We also note that all the aforementioned instantiations of two-message oblivious-transfer are additionally secure against *unbounded* malicious receivers.

4 Definitions

4.1 Proof Systems

Delayed-Input Interactive Protocols. An n-round delayed-input interactive protocol (P, V) for deciding a language L with associated relation R_L proceeds in the following manner:

- At the beginning of the protocol, P and V receive the size of the instance and execute the first $n-1$ rounds.
- At the start of the last round, P receives an input $(x, w) \in R_L$ and V receives x. Upon receiving the last round message from P, V outputs 1 or 0.

An execution of (P, V) with instance x and witness w is denoted as $\langle P, V\rangle(x, w)$. Whenever clear from context, we also use the same notation to denote the output of V. **Delayed-Input Interactive**

Arguments. An n-round delayed-input interactive argument for a language L must satisfy the standard notion of completeness as well as *adaptive soundness*, where the soundness requirement holds even against malicious PPT provers who choose the statement adaptively, depending upon the first $n-1$ rounds of the protocol.

Definition 3 (Delayed-Input Interactive Arguments)**.** *An n-round delayed-input interactive protocol (P, V) for deciding a language L is an interactive argument for L if it satisfies the following properties:*

- **Completeness:** *For every $(x, w) \in R_L$,*

$$\Pr\big[\langle P, V\rangle(x, w) = 1\big] \geq 1 - \mathsf{negl}(\kappa),$$

  where the probability is over the random coins of P and V.

- **Adaptive Soundness:** *For every $z \in \{0,1\}^*$, every PPT prover P^* that chooses $x \in \{0,1\}^\kappa \setminus L$ adaptively, depending upon the first $n-1$ rounds,*

$$\Pr\big[\langle P^*(z), V\rangle(x) = 1\big] \leq \mathsf{negl}(\kappa),$$

  where the probability is over the random coins of V.

Witness Indistinguishability. A proof system is witness indistinguishable if for any statement with at least two witnesses, proofs computed using different witnesses are indistinguishable.

Definition 4 (Witness Indistinguishability)**.** *A delayed-input interactive argument (P, V) for a language L is said to be* witness-indistinguishable *if for every non-uniform PPT verifier V^*, every $z \in \{0,1\}^*$, and every sequence (x, w_1, w_2) such that $w_1, w_2 \in R_L(x)$, the following two ensembles are computationally indistinguishable:*

$$\big\{\langle P, V^*(z)\rangle(x, w_1)\big\} \text{ and } \big\{\langle P, V^*(z)\rangle(x, w_2)\big\}$$

Non-adaptive Distributional Weak Zero Knowledge. Zero knowledge (ZK) requires that for any adversarial verifier, there exists a simulator that can produce a view that is indistinguishable from the real one to *every* distinguisher. Weak zero knowledge (WZK) relaxes the standard notion of ZK by reversing the order of quantifiers, and allowing the simulator to depend on the distinguisher.

We consider a variant of WZK, namely, distributional WZK [34, 28], where the instances are chosen from some hard distribution over the language. Furthermore, we allow the simulator's running time to depend upon the distinguishing probability of the distinguisher. We refer to this as distributional ϵ-WZK, which says that for every distinguisher D with distinguishing probability ϵ (where ϵ is an inverse polynomial) there exists a simulator with running time polynomial in ϵ. This notion was previously considered in [28, 19].

We define distributional ϵ-WZK property against *non-adaptive* malicious verifiers that receive the instance only in the last round of the protocol.

Definition 5 (Non-adaptive Distributional ϵ-Weak Zero Knowledge)**.** *A delayed-input interactive argument (P, V) for a language L is said to be* distributional ϵ-weak zero knowledge against non-adaptive verifiers *if for every efficiently samplable distribution $(\mathcal{X}_\kappa, \mathcal{W}_\kappa)$ on R_L, i.e., $\mathsf{Supp}(\mathcal{X}_\kappa, \mathcal{W}_\kappa) = \{(x, w) : x \in L \cap \{0,1\}^\kappa, w \in R_L(x)\}$, every non-adaptive PPT verifier V^*, every $z \in \{0,1\}^*$, every PPT distinguisher $\mathcal{D}$, and every $\epsilon = 1/\mathsf{poly}(\kappa)$, there exists a simulator $\mathcal{S}$ that runs in time $\mathsf{poly}(\kappa, \epsilon)$ such that:*

$$\left| \Pr_{(x,w)\leftarrow(\mathcal{X}_\kappa,\mathcal{W}_\kappa)} \left[\mathcal{D}(x, z, \mathsf{View}_{V^*}[\langle P, V^*(z)\rangle(x, w)] = 1\right] \right.$$

$$\left. - \Pr_{(x,w)\leftarrow(\mathcal{X}_\kappa,\mathcal{W}_\kappa)} \left[\mathcal{D}(x, z, \mathcal{S}^{V^*,D}(x, z)) = 1\right] \right| \leq \epsilon(\kappa),$$

where the probability is over the random choices of (x, w) as well as the random coins of the parties.

Non-adaptive Witness Hiding. Let L be an NP language and let $(\mathcal{X}, \mathcal{W})$ be a distribution over the associated relation R_L. A proof system is witness hiding w.r.t. $(\mathcal{X}, \mathcal{W})$ if for any $(x, w) \leftarrow (\mathcal{X}, \mathcal{W})$, a proof for x is "one-way" in the sense that no verifier can extract a witness for x from its interaction with the prover. Note that in order for WH to be non-trivial, it is necessary that $(\mathcal{X}, \mathcal{W})$ be a "hard" distribution.

Below, we define witness hiding property against *non-adaptive* malicious verifiers that receive the instance only in the last round of the protocol.

Definition 6 (Hard Distributions)**.** *Let $(\mathcal{X}, \mathcal{W}) = (\mathcal{X}_\kappa, \mathcal{W}_\kappa)_{\kappa\in\mathbb{N}}$ be an efficiently samplable distribution on R_L, i.e., $\mathsf{Supp}(\mathcal{X}_\kappa, \mathcal{W}_\kappa) = \{(x, w) : x \in L \cap \{0,1\}^\kappa, w \in R_L(x)\}$. We say that $(\mathcal{X}, \mathcal{W})$ is hard if for any poly-size circuit family $\{C_\kappa\}$, it holds that:*

$$\Pr_{(x,w)\leftarrow(\mathcal{X}_\kappa,\mathcal{W}_\kappa)} \left[C_\kappa(x) \in R_L(x)\right] \leq \mathsf{negl}(\kappa).$$

Definition 7 (Non-adaptive Witness Hiding)**.** *A delayed-input interactive argument (P, V) for a language L is said to be* witness hiding against non-adaptive verifiers *w.r.t. a hard distribution $(\mathcal{X}_\kappa, \mathcal{W}_\kappa)$ if for every non-adaptive PPT verifier V^*, every $z \in \{0,1\}^*$, it holds that:*

$$\Pr_{(x,w)\leftarrow(\mathcal{X}_\kappa,\mathcal{W}_\kappa)} \left[\langle P, V^*(z)\rangle(x) \in R_L(x)\right] \leq \mathsf{negl}(\kappa).$$

Non-adaptive Strong Witness Indistinguishability

Definition 8 (Non-adaptive Strong Witness Indistinguishability)**.** *A delayed-input interactive argument (P, V) for a language L is said to be* strong witness indistinguishable against non-adaptive verifiers *w.r.t. a pair of indistinguishable distributions $(\mathcal{X}_{1,\kappa}, \mathcal{W}_{1,\kappa}), (\mathcal{X}_{2,\kappa}, \mathcal{W}_{2,\kappa})$ if for every non-adaptive PPT verifier V^*, every $z \in \{0,1\}^*$, it holds that:*

$$\left| \Pr_{(x,w)\leftarrow(\mathcal{X}_{1,\kappa},\mathcal{W}_{1,\kappa})} \left[\mathcal{D}(x, z, \mathsf{View}_{V^*}[\langle P, V^*(z)\rangle(x, w)] = 1\right] \right.$$

$$\left. - \Pr_{(x,w)\leftarrow(\mathcal{X}_{2,\kappa},\mathcal{W}_{2,\kappa})} \left[\mathcal{D}(x, z, \mathsf{View}_{V^*}[\langle P, V^*(z)\rangle(x, w)] = 1\right] \right| \leq \mathsf{negl}(\kappa).$$

Definition 9 (Weak Resettable Non-adaptive Distributional ϵ-Weak Zero Knowledge)**.** *A three round delayed-input interactive argument (P, V) for a language L is said to be* weak resettable distributional weak zero-knowledge*, if for every efficiently samplable distribution $(\mathcal{X}_\kappa, \mathcal{W}_\kappa)$ on R_L, i.e., $\mathsf{Supp}(\mathcal{X}_\kappa, \mathcal{W}_\kappa) = \{(x, w) : x \in L \cap \{0,1\}^\kappa, w \in R_L(x)\}$, every non-adaptive PPT verifier V^*, every $z \in \{0,1\}^*$, every PPT distinguisher $\mathcal{D}$, and every $\epsilon = 1/\mathsf{poly}(\kappa)$, there exists a simulator $\mathcal{S}$ that runs in time $\mathsf{poly}(\kappa, \epsilon)$ and generates a simulated proof for instance $x \xleftarrow{\$} \mathcal{X}_\kappa$, such that over the randomness of sampling $(x, w) \leftarrow (\mathcal{X}_\kappa, \mathcal{W}_\kappa)$, $\Pr[b' = b] \leq \frac{1}{2} + \epsilon + \mathsf{negl}(\kappa)$ in the following experiment, where the challenger C plays the role of the prover:*

- *At the beginning, (C, V^*) receive the size of the instance, V^* receives auxiliary input z, and they execute the first 2 rounds. Let us denote these messages by τ_1, τ_2.*
- *Next, (C, V^*) run $\mathsf{poly}(\kappa)$ executions, with the same fixed first message τ_1, but different second messages chosen potentially maliciously by V^*. In each execution, C picks a fresh sample $(x, w) \leftarrow (\mathcal{X}_\kappa, \mathcal{W}_\kappa)$, and generates a proof for it according to honest verifier strategy.*
- *Next, C samples bit $b \xleftarrow{\$} \{0,1\}$ and if $b = 0$, for $x \xleftarrow{\$} \mathcal{X}_\kappa$ it generates an honest proof with first two messages τ_1, τ_2, else if $b = 1$, it generates a simulated proof with first two messages τ_1, τ_2 using simulator $\mathcal{S}$.*
- *Finally, V^* sends its view to a distinguisher $\mathcal{D}$ that outputs b.*

Remark 1. *A non-adaptive distributional weak ZK argument of knowledge is an argument of knowledge that satisfies the distributional weak ZK property against non-adaptive verifiers. Similarly, a non-adaptive strong WI argument of knowledge is an argument of knowledge that satisfies the strong WI property against non-adaptive verifiers. Finally, a non-adaptive witness hiding argument of knowledge can be defined similarly as an argument of knowledge that satisfies the witness hiding property against non-adaptive verifiers.*

Definition 10 (Reusable Witness Indistinguishable Argument of Knowledge)**.** *A three round delayed-input interactive argument of knowledge (P, V) for a language L is said to be* reusable witness indistinguisable*, if for every PPT verifier V^*, every $z \in \{0,1\}^*$, every $k = \mathsf{poly}(\kappa)$ and every sequence $(x^1, w^1), (x^2, w^2), \ldots (x^{k-1}, w^{k-1}), (x^k, w_1^k, w_2^k)$, $\Pr[b' = b] \leq \frac{1}{2} + \mathsf{negl}(\kappa)$ in the following experiment:*

- *At the beginning, (P, V^*) receive the size of the instance, and execute the first 2 rounds.*
- *Next, P receives inputs $(x^1, w^1), (x^2, w^2), \ldots (x^{k-1}, w^{k-1}), (x^k, w_1^k, w_2^k)$ and V^* receives $(x^1, x^2 \ldots x^k)$.*
- *Next P samples bit $b \xleftarrow{\$} \{0,1\}$ and generates the third message of the delayed-input witness indistinguishable argument of knowledge for instances $(x^1, x^2 \ldots x^k)$ using witnesses $(w^1, w^2, \ldots w^{k-1}, w_b^k)$*
- *Finally, V^* outputs b'.*

4.2 Two Party Computation

We define two party computation with distinguisher-dependent simulation. Following the terminology of [28], we call this weak two-party computation. This can also be naturally extended to weak multi-party computation.

We consider malicious adversaries who may arbitrarily deviate from the specified protocol. Also, we consider a model where parties send messages one by one. We consider the standard real-ideal definition where, very roughly, we require that any adversary interacting in the real world does not learn significantly more than an adversary that interacts with a simulator in an ideal world – except, that the simulator for a malicious receiver can depend upon the distinguisher.

We now give the formal definitions of two party computation. Parts of the definition are taken verbatim from [35].

A two-party functionality $F : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^* \times \{0,1\}^*$, where $F = (F_1, F_2)$, is such that for each pair of inputs (x, y), the output pair is a random variable $F_1(x, y), F_2(x, y)$ ranging over pairs of strings. The first party (with input x) wishes to obtain $F_1(x, y)$ and the second party (with input y) wishes to obtain $F_2(x, y)$.

Ideal model execution. The ideal model execution proceeds as follows:

- Inputs. Each party obtains an input, denoted w ($w = x$ for P_1 and $w = y$ for P_2).
- Send inputs to trusted party. An honest party always sends w to the trusted party. A malicious party may, depending on w, either abort or send some $w' \in \{0,1\}^{|w|}$ to the trusted party.
- Trusted party answers first party. In case it has obtained an input pair (x, y), the trusted party replies to the first party with $F_1(x, y)$. Otherwise (in case it didn't receive two valid inputs), the trusted party replies to both parties with a special symbol $\perp$.
- Trusted party answers second party. In case the first party is malicious, it may, dependeing on its input and the trusted party's answer, decide to stop the trusted party by sending it $\perp$ after receiving its output. In this case the trusted party sends $\perp$ to the second party. Otherwise (i.e., if not stopped), the trusted party sends $F_2(x, y)$ to the second party.
- Outputs. An honest party always outputs the message it obtained from the trusted party. A malicious party may output an arbitrary (PPT) function of its initial input and the message obtained from the trusted party.

Let $\mathcal{S}(\mathcal{S}_1, \mathcal{S}_2)$ be a pair of non-uniform PPT machines (representing parties in the ideal model). Such a pair is admissible if for at least one $i \in \{1, 2\}$ we have that $\mathcal{S}_i$ is honest (i.e., follows the honest party instructions in the above-described ideal execution). Then, the joint execution of F under $\mathcal{S}$ in the ideal model (on input pair (x, y) and security parameters κ), denoted $\mathsf{IDEAL}_{F,\mathcal{S}}(\kappa, x, y)$ is defined as the output pair of $\mathcal{S}_1$ and $\mathcal{S}_2$ from the above ideal execution.

Real model execution. We next consider the real model in which a real two-party protocol is executed (and there exists no trusted third party). In this case, a malicious party may follow an arbitrary feasible strategy; that is, any stratgy implementable by non-uniform PPT machines. In particular, the malicious party may abort the execution at any point in time (and when this happens prematurely, the other party is left with no output). Let F be as above and let Π be a two-party protocol for computing F. Furthermore, let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be a pair of non-uniform PPT mahcines (representing parties in the real model). Such a pair is admissible if for at least one $i \in \{1, 2\}$, $\mathcal{A}_i$ is honest (i.e., follows the strategy specified by the protocol). Then, the joint execution of Π under $\mathcal{A}$ in the real model, denoted by $\mathsf{REAL}_{\Pi,\mathcal{A}}(\kappa, x, y)$, is defined as the output pair of $\mathcal{A}_1$ and $\mathcal{A}_2$ resulting from the protocol interaction.

Definition 11 (Weak Secure Two Party Computation with Black-Box Simulation)**.** *Let F and Π be as described above. Protocol Π is said to securely compute F (in the malicious model) with weak security or distinguisher-dependent security, if for every pair of admissible non-uniform PPT machines $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ in the real model, for every error ϵ, and every distinguisher $\mathcal{D}$, there exists a pair of admissible PPT machines $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2)$ in the ideal model that run in time $\mathsf{poly}(\frac{1}{\epsilon})$, such that:*

$$\left|\Pr\left[\mathcal{D}\left(\mathsf{IDEAL}_{F,\mathcal{S}}(\kappa, x, y)_{\kappa\in\mathbb{N},(x,y)s.t.|x|=|y|}\right)\right]\right.$$

$$\left. - \Pr\left[\mathcal{D}\left(\mathsf{REAL}_{\Pi,\mathcal{A}}(\kappa, x, y)_{\kappa\in\mathbb{N},(x,y)s.t.|x|=|y|}\right)\right]\right| \leq \epsilon + \mathsf{negl}(\kappa)$$

Definition 12 (Distributional Weak Secure Two Party Computation with Black-Box Simulation)**.** *Let F and Π be as described above. Protocol Π is said to securely compute F (in the malicious model) with distributional weak security or distributional distinguisher-dependent security, if for every adversary $\mathcal{A}$ with fixed public input, and every pair of admissible non-uniform PPT machines $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ in the real model, for every error ϵ, and every distinguisher $\mathcal{D}$, there exists a pair of admissible PPT machines $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2)$ in the ideal model that run in time $\mathsf{poly}(\frac{1}{\epsilon})$, such that:*

$$\left|\Pr\left[\mathcal{D}\left(\mathsf{IDEAL}_{F,\mathcal{S}}(\kappa, x, y)_{\kappa\in\mathbb{N},(x,y)s.t.|x|=|y|}\right)\right]\right.$$

$$\left. - \Pr\left[\mathcal{D}\left(\mathsf{REAL}_{\Pi,\mathcal{A}}(\kappa, x, y)_{\kappa\in\mathbb{N},(x,y)s.t.|x|=|y|}\right)\right]\right| \leq \epsilon + \mathsf{negl}(\kappa)$$

Note that this definition weakens the previous definition by allowing the simulator to (non-uniformly) depend on the public input.

Definition 13 (Independent-Input Functionalities)**.** *An independent-input functionality is defined as a functionality between two parties, Alice and Bob. Let $(\mathcal{Q}, \mathcal{R}, \mathcal{U})$ denote the joint distribution over inputs of both parties, where Alice's private input can be sampled efficiently from* public *distribution $\mathcal{Q}$, Bob's private input is sampled from (possibly private) distribution $\mathcal{R}$, and $\mathcal{U}$ denotes their common public input. Then, a functionality F over $(\mathcal{X} = (\mathcal{Q}, \mathcal{U})) \times (\mathcal{Y} = (\mathcal{R}, \mathcal{U}))$, is independent-input for Alice, if $\mathcal{Q}$ is independent of $(\mathcal{R}, \mathcal{U})$. We denote the class of all two-party independent-input functionalities by $\mathcal{F}_{\mathsf{IIF}}$.*

Definition 14 (Weak Secure Computation for $\mathcal{F}_{\mathsf{IIF}}$ Functionalities)**.** *A protocol Π is said to securely compute $\mathcal{F}_{\mathsf{IIF}}$ with weak security for Alice (in the malicious model) and standard security for Bob (in the malicious model) if for every pair of admissible non-uniform PPT machines $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ (representing Alice and Bob respecitvely) computing $\mathcal{F}_{\mathsf{IIF}}$ in the real model, every error ϵ and every distinguisher $\mathcal{D}$ that obtains Bob's view, there exists a pair of admissible PPT machines $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2)$, representing Alice and Bob respectively in the ideal model (where $\mathcal{S}_2$ runs in time $\mathsf{poly}(\frac{1}{\epsilon})$, such that for every distinguisher $\mathcal{D}$ that obtains Alice's view:*

$$\left|\Pr\left[\mathcal{D}\left(\mathsf{IDEAL}_{F,\mathcal{S}}(\kappa, x, y)_{\kappa\in\mathbb{N},(x,y)s.t.|x|=|y|}\right)\right]\right.$$

$$\left. - \Pr\left[\mathcal{D}\left(\mathsf{REAL}_{\Pi,\mathcal{A}}(\kappa, x, y)_{\kappa\in\mathbb{N},(x,y)s.t.|x|=|y|}\right)\right]\right| \leq \mathsf{negl}(\kappa)$$

We emphasize that the simulator for a malicious Bob is distinguisher-dependent, whereas the simulator for malicious Alice satisfies the standard simulation security definition, without distinguisher-dependence.

Examples of independent-input functionalities (for which the above definition implies distinguisher-dependent simulation security) include: coin-tossing, generating common reference strings, evaluating oblivious PRFs, etc. We note that functionalities such as standard ZK and blind signatures do not satisfy this property because Alice's input (witness for ZK instance or signing key for signatures) is correlated with a public instance/verification key, and is not efficiently samplable given the public input.

4.2.1 Input-Indistinguishable Computation

We recall the notion of input-indistinguishable secure computation as defined by Micali, Pass and Rosen [50]. While they gave a definition for the concurrent setting, below, we provide a stand-alone version of their definition.

We first recall the notion of implicit input from their work, which is then used to formalize input-indistinguishable security.

Definition 15 (Implicit Input)**.** *Let $(\mathcal{A}_1, \mathcal{A}_2)$ be a k-round protocol, and let $\mathcal{A}_2^*$ be the adversary. Consider a function in_R that maps the full view of $\mathcal{A}_2^*$, denoted by $\mathsf{View}_1^*(\tau)$ in an execution τ of $(\mathcal{A}_1, \mathcal{A}_2^*)$ into an input $y^* \in (\mathcal{Y} \cup \perp)$. The function is said to be receiver implicit input for $(\mathcal{A}_1, \mathcal{A}_2^*)$ if $y^* = \perp$ whenever the receiver aborts, and otherwise y^* is equal to the unique input used by the receiver in execution τ.*

We also require a designated output delivery message in the protocol (before which no information on the output of the protocol is revealed. For simplicity, we assume that output delivery occurs in the last round of the protocolm and define boolean variable $\mathsf{output}_1(\tau)$ to be true if and only if the output delivery message has been sent to party $\mathcal{A}_1$ in τ.

Definition 16 ((Stand-alone) Input-indistinguishable computation)**.** *Let $f : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^* \times \{0,1\}^*$ be a (deterministic) function, and let Π be a two-party protocol. We say that Π securely computes f with respect to the sender and implicit input function in_R mapping a transcript of the execution to implicit input y^* of Bob, if the following conditions hold:*

- *Completeness: For every $x, y \in \mathcal{X} \times \mathcal{Y}$, every $\kappa \in \mathbb{N}$:*

$$\Pr[P_1(\mathsf{View}_1(\tau)) = f_1(x, y)] = 1$$

  where $\tau \stackrel{\$}{\leftarrow} \mathsf{REAL}_{\Pi,\mathcal{A}}(\kappa, x, y)_{\kappa \in \mathbb{N}, (x,y) s.t. |x|=|y|}$ We note that this is only for the case where sender also obtains an output.

- *Implicit Computation: For every efficient $\mathcal{A}_2^*$, for every $(x, y) \in \mathcal{X} \times \mathcal{Y}$, if $\mathsf{output}_1(\tau) = \mathsf{true}$, then $\Pr[\mathcal{A}_1(\mathsf{View}_1(\tau)) = f(x, y^*)] > 1 - \mathsf{negl}(n)$. Else if $\mathsf{output}_1(\tau) = \mathsf{false}$, then $\Pr[\mathcal{A}_1(\mathsf{View}_1(\tau)) = \perp] > 1 - \mathsf{negl}(n)$. Here $\tau \stackrel{\$}{\leftarrow} \mathsf{REAL}_{\Pi,\mathcal{A}}(\kappa, x, y)_{\kappa \in \mathbb{N}, (x,y) s.t. |x|=|y|}$ and $y^* \leftarrow \mathsf{in}_R(\mathsf{View}_2^*(\tau))$.*

- *Input Indistinguishability and Independence: For every efficient $\mathcal{A}_2^*$, every $x_1, x_2 \in \mathcal{X}$, and every $y \in \mathcal{Y}$, the following ensembles are computationally indistinguishable:*

  - $\mathsf{Expt}^{\mathcal{A}_1, \mathcal{A}_2^*}(x_1, x_2, y; \kappa)$
  - $\mathsf{Expt}^{\mathcal{A}_1, \mathcal{A}_2^*}(x_2, x_1, y; \kappa)$

  where the random variable $\mathsf{Expt}^{\mathcal{A}_1, \mathcal{A}_2^*}(x_1, x_2, y; \kappa)$ is defined as follows:

  1. *$\tau \stackrel{\$}{\leftarrow} \mathsf{REAL}_{\Pi,\mathcal{A}}(\kappa, x_1, y)_{\kappa \in \mathbb{N}}$*
  2. *$y^* \leftarrow \mathsf{in}_R(\mathsf{View}_2^*(\tau))$*
  3. *If $\mathsf{output}(\tau)$ is true, and $f_2(x_1, y^*) \neq f_2(x_2, y^*)$ then output $\perp$.*
  4. *Else, output $y^*, \mathsf{View}_2(\tau)$.*

4.3 Extractable Commitments

A commitment scheme allows a party to commit to a secret value x by publishing $C = \mathsf{com}(x; r)$ with randomnesss r, in such a way that $\mathsf{com}(x; r) \approx_c \mathsf{com}(0; r)$. The player can later decommit C to reveal x, by publishing x and a decommitment string r': then it is required that the player cannot open C to reveal $x' \neq x$ in a way that is acceptable to the verifier. In this paper, we are only interested in commitments where the binding property is statistical.

Definition 17 (Extractable Commitments)**.** *In addition to the standard properties of binding and hiding, a commitment is extractable if additionally, for any committer $\mathcal{C}$ that generates a commitment transcript C, there exists an efficient algorithm, called an extractor, which extracts x, such that with probability $1 - \mathsf{negl}(\kappa)$ over the randomness of the extractor and the transcript, there exists randomness r such that $C = \mathsf{com}(x; r)$.*

We say that the commitment is black-box extractable if the extractor works with black-box access to the committer.

Extractable Commitments with Over-Extraction. We note that simple three round constructions of extractable commitments are known [56, 57], if we only require correctness of the extracted value when the commitment is generated to a valid value. Otherwise (if the commitment is invalid), the extractor is allowed to output any (possibly valid) value. These are called extractable commitments with over-extraction.

5 Two Round Argument Systems

5.1 Construction

We show how to use two-message malicious-secure oblivious transfer (OT) to convert any three-message Σ-protocol according to Definition 1, into a two-message argument system. We then prove soundness of the resulting argument system, assuming sub-exponential security of oblivious transfer. We also prove that this protocol is witness indistinguishable, satisfies distributional weak zero-knowledge, strong WI and witness hiding against non-adaptive verifiers.

Let $\mathsf{OT} = (\mathsf{OT}_1, \mathsf{OT}_2)$ denote a two-message bit oblivious transfer protocol according to Definition 2. Let $\mathsf{OT}_1(b)$ denote the first message of the OT protocol with receiver input b, and let $\mathsf{OT}_2(m_0, m_1)$ denote the second message of the OT protocol with sender input bits m_0, m_1.

Let $\Sigma = (a, e, z)$ denote the three messages of a Σ-protocol. For most of this paper, we consider Σ-protocols that are a parallel composition of individual protocols with a single-bit challenge and constant soundness, i.e., the Σ-protocol contains three messages, denoted by (a, e, z) and that these messages can be parsed as $a = (a_1, \ldots, a_\kappa)$, $e = (e_1, \ldots, e_\kappa)$, and $z = (z_1, \ldots, z_\kappa)$, where for each $i \in [\kappa]$, the triplet (a_i, e_i, z_i) are messages corresponding to an underlying Σ-protocol with a single-bit challenge (i.e., where $e_i \in \{0, 1\}$). We denote by f_1 and f_2 the functions that satisfy $a_i = f_1(x, w; r_i)$ and $z_i = f_2(x, w, r_i, e_i)$, where r_i is uniformly chosen randomness.

Examples of such Σ-protocols are the parallel Blum proof of Graph Hamiltonicity [13], and the Lapidot-Shamir [49] three round WI proof. By a Karp reduction to Graph Hamiltonicity, there exists such a Σ-protocol for all of NP.

5.2 Adaptive Soundness

The protocol in Figure 1 compiles a three-round public coin proof to a two-round argument using oblivious transfer. Kalai-Raz [47] proved that such a compiler, applied to any public-coin proof system

Witness Indistinguishable and Weak Distributional Zero-Knowledge Argument
Prover Input: Instance $x \in L$, witness w such that $R_L(x, w) = 1$.
Verifier Input: Instance x, language L.

- **Verifier Message:** The verifier picks challenge $e \xleftarrow{\$} \{0,1\}^\kappa$ for the Σ-protocol, and for $i \in [\kappa]$, sends $\mathsf{OT}_{1,i}(e_i)$ in parallel. Each bit e_i is encrypted with a fresh OT instance.
- **Prover Message:** For $i \in [\kappa]$, the prover sends $a_i, \mathsf{OT}_{2,i}(z_i^0, z_i^1)$ in parallel.
- **Verifier Output:** The verifier V recovers z_i as the output of OT_i for $i \in [\kappa]$, and outputs accept if for all $i \in [\kappa]$, $(a_i, e_i, z_i)_{i\in[\kappa]}$ is an accepting transcript of the underlying Σ-protocol.

Figure 1: Two Round Argument System for NP

preserves soundness. Specifically, the following theorem in [47] proves (static) soundness of the above protocol, assuming sub-exponential oblivious transfer.

Imported Theorem 1. *(Rephrased) Let $\Sigma = (a, e, z)$ denote a Σ-protocol, and let $\ell = \mathsf{poly}(\kappa, s)$ be the size of z, where κ is the security parameter, and s is an upper bound on the length of allowed instances. Assuming the existence of an oblivious transfer protocol secure against probabilistic senders running in time at most 2^ℓ, the protocol in Figure 1 is sound.*

We observe that the proof in Kalai-Raz [47] can be extended to prove adaptive soundness, i.e., soundness against malicious provers that can adaptively choose $x \notin L$ based on the verifier's input message.

Lemma 1. *Let $\Sigma = (a, e, z)$ denote a Σ-protocol, and let ℓ be the size of z. Assuming the existence of an oblivious transfer protocol secure against probabilistic senders running in time at most 2^ℓ, the protocol in Figure 1 is adaptively sound.*

Proof. We will use a prover that breaks soundness to break sub-exponential receiver security of the underlying oblivious transfer. The reduction samples two random challenge strings e_0, e_1 and reduction sends them to an external OT challenger. The external OT challenger picks $b \xleftarrow{\$} \{0,1\}$, and outputs $\mathsf{OT}_1(e_{i,b})$ for $i \in [\kappa]$, which the reduction forwards to the cheating prover P^*.

P^* outputs $x \notin L$, together with messages a_i, $\mathsf{OT}_2(z_i^0, z_i^1)$ for $i \in [\kappa]$. Next, the reduction R does a brute-force search over all possible values of z, checking whether (a, e_0, z) is an accepting transcript for any $z \in \{0,1\}^\ell$ and whether (a, e_1, z') is an accepting transcript for any $z' \in \{0,1\}^\ell$.

Suppose a cheating prover breaks soundness with probability $p = \frac{1}{\mathsf{poly}(\kappa)}$ over the randomness of the experiment. Since the reduction chooses prover messages e_0, e_1 uniformly at random, with probability p, the prover P^* outputs a_i^*, $\mathsf{OT}_2(z_i^0, z_i^1)$ for $i \in [\kappa]$ that cause the verifier to accept.

Thus, with probability p, R finds at least one z such that (a^*, e_b, z) is an accepting transcript.

Since $e_{\bar{b}}$ was picked uniformly at random and independent of e_b, we argue that with at most $\mathsf{negl}(\kappa)$ probability, R finds one or more z' such that $(a^*, e_{\bar{b}}, z')$ is an accepting transcript. Note that with probability $1 - 2^{-\kappa}$, we have that $e_b \neq e_{\bar{b}}$. By special-soundness of the underlying Σ-protocol, if there exists z' such that $(a^*, e_{\bar{b}}, z')$ is an accepting transcript, conditioned on $e_b \neq e_{\bar{b}}$, this would allow obtaining a witness w from (a, e_b, z) and $(a, e_{\bar{b}}, z')$, which is a contradiction since $x \notin L$.

Therefore, if R finds z such that (a^*, e_b, z) is an accepting transcript, R outputs e_b as its guess for the first OT message, and this guess is correct with probability at least $p - \mathsf{negl}(\kappa)$. Since R runs in time 2^ℓ and guesses the OT message with non-negligible probability, this is a contradiction to the security of OT against 2^ℓ-time malicious senders. □

Observing the Verifier's output. The protocol is not sound when the prover is allowed to generate a-priori unbounded arguments using the same verifier message, as an adaptive function of the *verifier's accept/reject outputs on prior arguments*. Looking ahead, such a prover can use the simulation strategy from Section 5.4 to explicitly break soundness.

However, the protocol is sound when the prover is only allowed to generate an *a-priori bounded* arguments that adaptively depend on the verifier's accept/reject outputs on prior arguments. This can be ensured via simply having the verifier output a longer challenge string – to obtain adaptive soundness for B executions, the protocol requires the verifier to generate $e \xleftarrow{\$} \{0,1\}^{\kappa \cdot B}$, and encrypt it using $\kappa \cdot B$ OT instances. The prover uses the first κ instances for the first argument, the second set of κ instances for the second, and so forth. It is easy to see then that the argument of Lemma 1 easily extends to the bounded execution case.

5.3 Witness Indistinguishability

Theorem 6. *Assuming two-round oblivious transfer (OT) secure against malicious PPT receivers, the two-round protocol in Figure 1 is witness-indistinguishable against PPT verifiers.*

Recall that witness indistinguishability (WI) is closed under parallel composition [29], therefore it suffices to prove WI for a single repetition (i.e., for some $i \in [\kappa]$) of the protocol in Figure 1. That is, we consider the following protocol:

Witness Indistinguishable Argument
Prover Input: Instance x, witness w such that $R(x, w) = 1$.
Verifier Input: Instance x

- **Verifier Message:** The verifier picks challenge $e_r \xleftarrow{\$} \{0,1\}$ for the Σ-protocol, and sends $r = \mathsf{OT}_1(e_r)$ in parallel.
- **Prover Message:** The prover verifies that r is a valid message according to the underlying OT scheme. Then the prover computes $a = f_1(x, w, r), z_{r,0} = f_2(x, w, r, 0)$ and $z_{r,1} = f_2(x, w, r, 1)$ and sends $a, \mathsf{OT}_2(z_{r,0}, z_{r,1})$ in parallel.
- **Verifier Output:** The verifier recovers $z_{r,e}$ and verifies that $(a, e, z_{r,e})$ is a valid accepting transcript of the Σ-protocol.

Figure 2: A Single Repetition of a Two Round Argument System for NP

Our proof proceeds via a sequence of hybrid arguments, where, in an intermediate hybrid, we construct a distinguisher-dependent simulator, that learns (using both witnesses w_1 and w_2), an approximation for the verifier's challenge e. Upon learning the challenge, the simulator uses the honest-verifier ZK property to generate a simulated proof, without using any of the witnesses.

5.3.1 Proof via Hybrid Experiments

For an NP language L with corresponding relation R_L, consider an instance $x \in L$ and let w_1, w_2 be two witnesses such that $R_L(x, w_1) = 1$ and $R(x, w_2) = 1$. We prove witness indistinguishability by contradiction: suppose there exists a distinguisher $\mathcal{D}_\mathcal{V}$ that distinguishes between experiments where the prover generates a proof using witness w_1 versus an experiment where the prover generates a proof using witness w_2, with advantage greater than ϵ'. We then consider a sequence of 6 hybrid experiments, indexed by error parameter $\epsilon = \epsilon'/7$, and by the previous statement, $\mathcal{D}_\mathcal{V}$ must distinguish two consecutive

hybrids in the sequence with advantage greater than $\epsilon'/6$. But this is a contradiction, because we prove that the advantage of the distinguisher $\mathcal{D}_{\mathcal{V}}$ between every two consecutive hybrids (indexed by ϵ) is at most $\epsilon + \mathsf{negl}(\kappa)$.

Hybrid_{w_1} :

This hybrid corresponds to an honest prover that generates a proof for x using witness w_1. That is, the challenger computes $a = f_1(x, w_1, r), z^0 = f_2(x, w_1, r, e = 0), z^1 = f_2(x, w_1, r, e = 1)$, and sends the prover message according to Figure 2.

The output of this hybrid denoted by $\mathcal{D}_V(\mathsf{Hybrid}_{w_1})$ is the output of the distinguisher on input the view of the verifier in this experiment.

$\mathsf{Hybrid}_{1,\epsilon}$:

In this hybrid, with probability at least $1 - 2^{-\kappa}$, the view of the verifier is the same as Hybrid_{w_1}, and with probability at most $2^{-\kappa}$, the output view is $\perp$. This ensures that the advantage of the distinguisher between the previous hybrid and this hybrids is at most $2^{-\kappa}$.

This hybrid is indexed by a small error parameter $\epsilon = \frac{1}{\mathsf{poly}(\kappa)}$ for some polynomial $\mathsf{poly}(\cdot)$, and proceeds as follows. The challenger sets a counter $\mathsf{count} = 0$ and while $\mathsf{count} \leq \kappa$, repeats the following two steps:

Step_1 : The first step of this experiment is the same as Hybrid_{w_1}, that is, first compute $a = f_1(x, w_1, r), z^0 = f_2(x, w_1, r, e = 0), z^1 = f_2(x, w_1, r, e = 1)$, and send prover message according to Figure 2. Denote the view of the verifier at the end of this step, by View_1.

Step_2 : Additionally, (unlike Hybrid_{w_1}), guess $e_{\mathsf{guess}} \xleftarrow{\$} \{0, 1\}$. Then, run the algorithm in Figure 3 with oracle access to the V and distinguisher $\mathcal{D}$, and error parameter ϵ, to obtain e_{approx}. This corresponds, roughly, to approximating the verifier's challenge e, with error at most ϵ (this approximation is called e_{approx}).

If $e_{\mathsf{guess}} = e_{\mathsf{approx}}$, set the output of the distinguisher on input the view View_1, as the output of the experiment, and stop.

Else, set $\mathsf{count} = \mathsf{count} + 1$ and continue (go to start of while loop).

We will add a more detailed explanation of the approximating algorithm in the next hybrid. In this hybrid, it suffices to note that independently with probability at least $\frac{1}{2}$ in any iteration, $e_{\mathsf{guess}} = e_{\mathsf{approx}}$. Conditioned on $e_{\mathsf{guess}} = e_{\mathsf{approx}}$ in at least one iteration, the view of the distinguisher in this hybrid remains the same as Hybrid_{w_1}.

If $\mathsf{count} > \kappa$, abort and output 0 as the output of the experiment.

Lemma 2. $|\Pr[\mathcal{D}_{\mathcal{V}}(\mathsf{Hybrid}_{w_1}) = 1] - \Pr[\mathcal{D}_{\mathcal{V}}(\mathsf{Hybrid}_{1,\epsilon}) = 1]| \leq \mathsf{negl}(\kappa)$

Proof. The experiments are identical conditioned on the challenger not aborting. Since e_{guess} is sampled independently at random from e_{approx}, $\Pr[e_{\mathsf{guess}} = e_{\mathsf{approx}}] = \frac{1}{2}$ independently in every iteration. Thus, the advantage of the distinguisher is at most the probability of abort, which is $\frac{1}{2^\kappa}$. □

$\mathsf{Hybrid}_{2,\epsilon}$: In this hybrid, at an intuitive level, the challenger approximates the receiver's challenge (i.e., the bit e_r), and replaces the sender's oblivious transfer messages with simulated messages, corresponding to the approximated value of e_r.

That is, the (malicious) receiver sends message r, that could possibly correspond to $\mathsf{OT}_1(e_r)$ for some challenge bit e_r (or to no e_r at all). The challenger verifies that r is a valid message according to the underlying OT scheme. By security of the underlying OT against malicious receivers (refer Definition 2), for any fixed r sent by a malicious receiver that the challenger verifies to be a valid OT message, and any auxiliary input z, the following statement is true: Conditioned on r being the first message of R, either the distribution of receiver views $\mathsf{View}_R(\langle S(m_0, m_1), R(z)\rangle) \approx_c \mathsf{View}_R(\langle S(m_0, m_0), R(z)\rangle$ for all

Algorithm $\mathcal{M}^{V,\mathcal{D}_V}$ to approximate the verifier's challenge.

1. Set $p = 1/\epsilon^3$.
2. For $w \in \{w_1, w_2\}$, and for the same fixed first message of the verifier, repeat the following:
   - Set $j = 1, \mathcal{D}_{0,w} = 0$ and repeat:
     (a) If $j = p$, then halt.
     (b) Sample fresh randomness r_j, set $a = f_1(x, w, r_j), z^0 = z^1 = f_2(x, w, e = 0, r_j)$, and send the prover message according to Figure 2.
     Set $\mathcal{D}_{0,w} = \mathcal{D}_{0,w} + \frac{1}{p}$ if the output of the distinguisher $\mathcal{D}_V = 1$ (w.l.o.g., we assume that the distinguisher $\mathcal{D}_V$ outputs either 0 or 1).
   - Set $j = 1, \mathcal{D}_{1,w} = 0$ and repeat:
     (a) If $j = p$, then halt.
     (b) Sample fresh randomness r_j, set $a = f_1(x, w, r_j), z^0 = z^1 = f_2(x, w, a, e = 1, r_j)$, and send the prover message according to Figure 2.
     Set $\mathcal{D}_{1,w} = \mathcal{D}_{1,w} + \frac{1}{p}$ if the output of the distinguisher $\mathcal{D}_V = 1$ (w.l.o.g., we assume that the distinguisher $\mathcal{D}_V$ outputs either 0 or 1).
   - Set $j = 1, \mathcal{D}_w = 0$ and repeat:
     (a) If $j = p$, then halt.
     (b) Sample fresh randomness r_j, set $a = f_1(x, w, r_j), z^0 = f_2(x, w, a, e = 0, r_j), z^1 = f_2(x, w, a, e = 1, r_j)$, and send the prover message according to Figure 2.
     Set $\mathcal{D}_w = \mathcal{D}_w + \frac{1}{p}$ if the output of the distinguisher $\mathcal{D}_V = 1$ (w.l.o.g., we assume that the distinguisher $\mathcal{D}_V$ outputs either 0 or 1).
3. If $|\mathcal{D}_{1,w_2} - \mathcal{D}_{w_2}| \geq |\mathcal{D}_{0,w_2} - \mathcal{D}_{w_2}| + \epsilon$, set $e_{\mathsf{approx}} = 0$.
4. Else if $|\mathcal{D}_{0,w_2} - \mathcal{D}_{w_2}| \geq |\mathcal{D}_{1,w_2} - \mathcal{D}_{w_2}| + \epsilon$, set $e_{\mathsf{approx}} = 1$.
5. Else if $|\mathcal{D}_{1,w_1} - \mathcal{D}_{w_1}| \geq |\mathcal{D}_{0,w_1} - \mathcal{D}_{w_1}| + \epsilon$, set $e_{\mathsf{approx}} = 0$.
6. Else set $e_{\mathsf{approx}} = 1$.

Figure 3: Approximately Learning the Verifier's Challenge

(m_0, m_1), or, $\mathsf{View}_R(\langle S(m_0, m_1), R(z)\rangle) \approx_c \mathsf{View}_R(\langle S(m_1, m_1), R(z)\rangle)$ for all (m_0, m_1). That is, every r generated by a malicious receiver that verifies as a valid OT message, *behaves* like $\mathsf{OT}_1(e_r)$ for some bit e_r.

In other words, for any distinguisher that has input the view of the verifier, at least one out of $\mathsf{View}_R(\langle S(m_0, m_0), R(z)\rangle)$ and $\mathsf{View}_R(\langle S(m_1, m_1), R(z)\rangle)$ is $\mathsf{negl}(\kappa)$- close to the correct distribution $\mathsf{View}_R(\langle S(m_0, m_1), R(z)\rangle)$ (or, both could be $\mathsf{negl}(\kappa)$-close, which we do not discuss here because the distinguisher is a trivial distinguisher, and the proof becomes easier). When only one of the distributions $\mathsf{View}_R(\langle S(m_0, m_0), R(z)\rangle)$ and $\mathsf{View}_R(\langle S(m_1, m_1), R(z)\rangle)$ is close to the correct distribution, the challenger computes which distribution is close by sending *many* randomly chosen sender messages to the distinguisher, according to all three distributions, and learning whether the output of the distinguisher on $\mathsf{View}_R(\langle S(m_0, m_0), R(z)\rangle)$ or the output of the distinguisher on input $\mathsf{View}_R(\langle S(m_1, m_1), R(z)\rangle)$ is close to the output of the distinguisher on input $\mathsf{View}_R(\langle S(m_0, m_1, z)\rangle)$, upto error $\epsilon = \frac{1}{\mathsf{poly}(\kappa)}$.

Formally, the experiment is indexed by an error parameter $\epsilon = \frac{1}{\mathsf{poly}(\kappa)}$, and proceeds as follows.

Step_1 : First, guess $e_{\mathsf{guess}} \xleftarrow{\$} \{0,1\}$. Next, compute $a = f_1(x, w_1, r), z^0 = f_2(x, w_1, r, e_{\mathsf{guess}}), z^1 = f_2(x, w_1, r, e_{\mathsf{guess}})$, and send prover message according to Figure 2.

Step_2 : Then, run the protocol in Figure 3 with error parameter ϵ to compute e_{approx}. If $e_{\mathsf{guess}} = e_{\mathsf{approx}}$, set the output of the distinguisher on input the view of the verifier in Step_1 of this experiment, as the output of the experiment $\mathcal{D}_V(\mathsf{Hybrid}_{2,\epsilon})$, and stop.

Else, set $\mathsf{count} = \mathsf{count} + 1$ and continue (go to start of while loop).

If $\mathsf{count} > \kappa$, abort and output 0 as the output of the experiment.

Lemma 3.

$$\left|\Pr[\mathcal{D}_{\mathcal{V}} = 1|\mathsf{Hybrid}_{1,\epsilon}] - \Pr[\mathcal{D}_{\mathcal{V}} = 1|\mathsf{Hybrid}_{2,\epsilon}]\right| \leq \epsilon + \mathsf{negl}(\kappa)$$

Proof. For the fixed verifier message $\mathsf{OT}_1(e)$ corresponding to the receiver challenge bit e_r, and witness $w \in \{w_1, w_2\}$,

- Let $\mathcal{D}_{\mathsf{correct},0,w}$ denote the actual distribution output by the distinguisher when the challenger samples fresh randomness r_j, sets $a = f_1(x, w, r_j), z^0 = z^1 = f_2(x, w, e = 0, r_j)$, and send the prover message according to Figure 2. We will abuse notation and also use $\mathcal{D}_{\mathsf{correct},0,w}$ to denote the probability that the distinguisher outputs 1 in this situation.
- Let $\mathcal{D}_{\mathsf{correct},1,w}$ denote the actual distribution output by the distinguisher when the challenger samples fresh randomness r_j, sets $a = f_1(x, w, r_j), z^0 = z^1 = f_2(x, w, e = 1, r_j)$, and send the prover message according to Figure 2. We will abuse notation and also use $\mathcal{D}_{\mathsf{correct},1,w}$ to denote the probability that the distinguisher outputs 1 in this situation.
- Let $\mathcal{D}_{\mathsf{correct},w}$ denote the actual distribution output by the distinguisher when the challenger samples fresh randomness r_j, sets $a = f_1(x, w, r_j), z^0 = f_2(x, w, e = 0, r_j), z^1 = f_2(x, w, a, e = 1, r_j)$, and send the prover message according to Figure 2. We will abuse notation and also use $\mathcal{D}_{\mathsf{correct},w}$ to denote the probability that the distinguisher outputs 1 in this situation.
- We note that $\mathcal{D}_{0,w_1}, \mathcal{D}_{1,w_1}, \mathcal{D}_{0,w_2}, \mathcal{D}_{1,w_2}, \mathcal{D}_{w_1}, \mathcal{D}_{w_2}$ denote the approximate distributions that the simulator lerans (refer Figure 3), while $\mathcal{D}_{\mathsf{correct},0,w}, \mathcal{D}_{\mathsf{correct},1,w}$ and $\mathcal{D}_{\mathsf{correct},w}$ denote the actual distributions (output by the distinguisher) themselves.

Claim 1. *Either of the following statements is true:*

- *For all witnesses w,*

$$|\Pr[\mathcal{D}_{\mathsf{correct},0,w} = 1] - \Pr[\mathcal{D}_{\mathsf{correct},w} = 1]| \leq \mathsf{negl}(\kappa)$$

- *For all witnesses w,*

$$|\Pr[\mathcal{D}_{\mathsf{correct},1,w} = 1] - \Pr[\mathcal{D}_{\mathsf{correct},w} = 1]| \leq \mathsf{negl}(\kappa)$$

Proof. Assume, for contradiction, that there exist $\mathcal{V}$ and $\mathcal{D}_{\mathcal{V}}$ for which the claim is not true. We will use them to break sender security of the underlying OT. Consider a reduction $\mathcal{R}$ that obtains the first OT message from $\mathcal{V}$ and forwards this message to the OT challenger.

The reduction sets $a = f_1(x, w, r_j), z^0 = f_2(x, w, e = 0, r_j), z^1 = f_2(x, w, e = 1, r_j)$, and sends (z^0, z^1) to the OT challenger.

The OT challenger generates either the real message $\mathsf{OT}_2(z^0, z^1)$, or a simulated message $\mathsf{OT}_2(z^*, z^*)$, for some (fixed) $z^* \in \{z^0, z^1\}$. The reduction forwards this message to the OT.

The reduction mirrors the output of $\mathcal{D}_{\mathcal{V}}$ and it holds that,
$|\Pr[\mathcal{D}_{\mathcal{V}} = 1|\text{real OT message}] - \Pr[\mathcal{D}_{\mathcal{V}} = 1|\text{simulated OT message}]| \geq \frac{1}{\mathsf{poly}(\kappa)}$ for some polynomial $\mathsf{poly}(\cdot)$, for both $z^* = z^0$ and $z^* = z^1$, which is a contradiction. □

This claim establishes that *at least one* of the distributions $\mathcal{D}_{\mathsf{correct},0,w}$ and $\mathcal{D}_{\mathsf{correct},1,w}$ is negligibly close to $\mathcal{D}_{\mathsf{correct},w}$.

If both $\mathcal{D}_{\mathsf{correct},0,w}$ and $\mathcal{D}_{\mathsf{correct},1,w}$ are ϵ-close to $\mathcal{D}_{\mathsf{correct},w}$ for $w = w_1$, then for any value of e_{guess} in $\{0,1\}$, $\left|\Pr[\mathcal{D}_{\mathcal{V}} = 1|\mathsf{Hybrid}_{1,\epsilon}] - \Pr[\mathcal{D}_{\mathcal{V}} = 1|\mathsf{Hybrid}_{2,\epsilon}]\right| \leq \epsilon + \mathsf{negl}(\kappa)$ and we are done.

Therefore, for the rest of this lemma, we restrict ourselves to the case where for $w = w_1$, one and only one out of $\mathcal{D}_{\mathsf{correct},0,w}$ and $\mathcal{D}_{\mathsf{correct},1,w}$ is ϵ-close to $\mathcal{D}_{\mathsf{correct},w}$. In particular, this also implies that $|\mathcal{D}_{\mathsf{correct},0,w} - \mathcal{D}_{\mathsf{correct},1,w}| > \epsilon$ for $w = w_1$.

If the challenger could "magically" set e_{guess} to 0 if $\mathcal{D}_{\mathsf{correct},0,w}$ was close to $\mathcal{D}_{\mathsf{correct},w}$, and to 1 if $\mathcal{D}_{\mathsf{correct},0,w}$ was close to $\mathcal{D}_{\mathsf{correct},w}$, then again we would have that

$$\left|\Pr[\mathcal{D}_{\mathcal{V}} = 1|\mathsf{Hybrid}_{1,\epsilon}] - \Pr[\mathcal{D}_{\mathcal{V}} = 1|\mathsf{Hybrid}_{2,\epsilon}]\right| \leq \epsilon.$$

Unfortunately, the challenger cannot magically know which distributions are close, and will therefore have to approximate these distributions to obtain an answer. We now bound the probability that the challenger's approximation e_{approx} is incorrect conditioned on $|\mathcal{D}_{\mathsf{correct},0,w} - \mathcal{D}_{\mathsf{correct},1,w}| > \epsilon$, i.e., we show:

Claim 2.

$$\Pr\left[(e_{\mathsf{approx}} = b)\middle|(|\mathcal{D}_{\mathsf{correct},1,w} - \mathcal{D}_{\mathsf{correct},0,w}| > \epsilon) \wedge (|\mathcal{D}_{\mathsf{correct},w} - \mathcal{D}_{\mathsf{correct},b,w}| > \epsilon)\right] \leq \mathsf{negl}(\kappa) \text{ where } w = w_1.$$

Proof. We note that for $w \in \{w_1, w_2\}, \mathcal{D}_0, \mathcal{D}_1, \mathcal{D}_w$ consist of p random samples from the distributions: $\mathcal{D}_{\mathsf{correct},0,w}, \mathcal{D}_{\mathsf{correct},1,w}, \mathcal{D}_{\mathsf{correct},w}$.

Then, using a simple Chernoff bound, we have that for $w \in \{w_1, w_2\}$:

- $\Pr[(\mathcal{D}_0 > \mathcal{D}_{\mathsf{correct},0,w}(1+\alpha)) \vee (\mathcal{D}_0 < \mathcal{D}_{\mathsf{correct},0,w}(1-\alpha))] \leq 2\exp^{-\frac{\alpha^2 p \mathcal{D}_{\mathsf{correct},0,w}}{2}}$
- $\Pr[(\mathcal{D}_1 > \mathcal{D}_{\mathsf{correct},1,w}(1+\alpha)) \vee (\mathcal{D}_1 < \mathcal{D}_{\mathsf{correct},1,w}(1-\alpha))] \leq 2\exp^{-\frac{\alpha^2 p \mathcal{D}_{\mathsf{correct},1,w}}{2}}$
- $\Pr[(\mathcal{D}_w > \mathcal{D}_{\mathsf{correct},w}(1+\alpha)) \vee (\mathcal{D}_w < \mathcal{D}_{\mathsf{correct},w}(1-\alpha))] \leq 2\exp^{-\frac{\alpha^2 p \mathcal{D}_{\mathsf{correct},w}}{2}}$

Setting $\alpha = \frac{\epsilon}{2}$, by a simple union bound we have that for $w \in \{w_1, w_2\}$,

$$\Pr\left[\left(|\mathcal{D}_{\mathsf{correct},0,w} - \mathcal{D}_0| > \frac{\epsilon}{2}\right) \vee \left(|\mathcal{D}_{\mathsf{correct},1,w} - \mathcal{D}_1| > \frac{\epsilon}{2}\right) \vee \left(|\mathcal{D}_{\mathsf{correct},w} - \mathcal{D}_w| > \frac{\epsilon}{2}\right)\right] \leq 6\exp^{-\frac{1}{2\epsilon}}$$

Since ϵ will always be set to $\frac{1}{\mathsf{poly}(\kappa)}$ for some polynomial $\mathsf{poly}(\cdot)$, for $w \in \{w_1, w_2\}$,

$$\Pr\left[\left(|\mathcal{D}_{\mathsf{correct},0,w} - \mathcal{D}_0| > \frac{\epsilon}{2}\right) \vee \left(|\mathcal{D}_{\mathsf{correct},1,w} - \mathcal{D}_1| > \frac{\epsilon}{2}\right) \vee \left(|\mathcal{D}_{\mathsf{correct},w} - \mathcal{D}_w| > \frac{\epsilon}{2}\right)\right] \leq 6\exp^{-\frac{1}{8\epsilon}}$$

We consider the event that the approximation e_{approx} is incorrect, and perform a case-analysis of this event.

- **Case I:** Suppose that the value e_{approx} was fixed in Step 5 or Step 6 (i.e., by using witness w_1 to approximate). Recall that one of $\mathcal{D}_{\mathsf{correct},0,w}$ and $\mathcal{D}_{\mathsf{correct},1,w}$ is at least ϵ-far from $\mathcal{D}_{\mathsf{correct},w}$, and the other is at most $\mathsf{negl}(\kappa)$-far, for $w = w_1$. The bit b is estimated via $\mathcal{D}_0, \mathcal{D}_1, \mathcal{D}_w$ which each have error at most $\frac{\epsilon}{2}$, from the corresponding distributions $\mathcal{D}_{\mathsf{correct},0,w}, \mathcal{D}_{\mathsf{correct},1,w}, \mathcal{D}_{\mathsf{correct},0,w}$. Thus, $\Pr[e_{\mathsf{approx}} \text{ is incorrect in Case I}] \leq \mathsf{negl}(\kappa)$.
- **Case II:** Suppose that the value e_{approx} was fixed in Step 3 or Step 4 of Figure 3 (i.e., by using witness w_2 to approximate). Recall that there exists a bit $\bar{b}$ such that $\mathcal{D}_{\mathsf{correct},\bar{b},x}$ is at least ϵ-far from $\mathcal{D}_{\mathsf{correct},w}$, and $\mathcal{D}_{\mathsf{correct},b,w}$ is at most $\mathsf{negl}(\kappa)$-far, for $w = w_1$. By Claim 1, even for $w = w_2$, $\mathcal{D}_{\mathsf{correct},b,w}$ is at most $\mathsf{negl}(\kappa)$-far from $\mathcal{D}_{\mathsf{correct},w}$.

Then, e_{approx} is incorrect if Step 3 and Step 4 result in output $\overline{b} = 1-b$, which happens if and only if $|\mathcal{D}_{b,w_2} - \mathcal{D}_{w_2}| > |\mathcal{D}_{\overline{b},w_2} - \mathcal{D}_{w_2}| + \epsilon$. However, note that $\Pr[|\mathcal{D}_{b,w_2} - \mathcal{D}_{w_2}| > \epsilon \big| |\mathcal{D}_{\mathsf{correct},b,w} - \mathcal{D}_{\mathsf{correct},w}| = \mathsf{negl}(\kappa)] \leq \mathsf{negl}(\kappa)$ by the Chernoff bounds above. Therefore, Steps 3 and 4 result in incorrect output e_{approx} with probability at most $\mathsf{negl}(\kappa)$.

Summing up, $\Pr\left[e_{\mathsf{approx}} = b \big| |(\mathcal{D}_{\mathsf{correct},1,w} - \mathcal{D}_{\mathsf{correct},0,w}| > \epsilon) \wedge (|\mathcal{D}_{\mathsf{correct},w} - \mathcal{D}_{\mathsf{correct},b,w}| > \epsilon)\right] \leq \mathsf{negl}(\kappa)$ for $w = w_1$. □

This completes the proof of the lemma. □

$\mathsf{Hybrid}_{3,\epsilon}$: In this experiment, the challenger approximates the verifier challenge and conditions on $e_{\mathsf{guess}} = e_{\mathsf{approx}}$ as before. In $\mathsf{Hybrid}_{2,\epsilon}$, the challenger response $\mathsf{OT}_2(z_{e_{\mathsf{approx}}}, z_{e_{\mathsf{approx}}})$ was fixed and did not encode the witness, but the message a still possibly encoded witness w_1. In this hybrid, instead of sampling $(a, z_{e_{\mathsf{guess}}})$ using the witness w_1, the challenger simulates $(a, z_{e_{\mathsf{guess}}})$ without any witness, instead relying on the honest-verifier ZK simulator of the underlying Σ-protocol.

Formally, the experiment is indexed by an error parameter $\epsilon = \frac{1}{\mathsf{poly}(\kappa)}$, and proceeds as follows.

Step_1 : First, guess $e_{\mathsf{guess}} \xleftarrow{\$} \{0,1\}$. Next, compute without using the witness w_1, $a = f_1(x, r, e_{\mathsf{guess}}), z^0 = z^1 = f_2(x, r, e_{\mathsf{guess}})$, and send prover message according to Figure 2.

Step_2 : Then, run the protocol in Figure 3 with error parameter ϵ to compute e_{approx}. If $e_{\mathsf{guess}} = e_{\mathsf{approx}}$, set the output of the distinguisher on input the view of the verifier in Step_1 of this experiment, as the output of the experiment $\mathcal{D}_V(\mathsf{Hybrid}_{3,\epsilon})$, and stop.

Else, set $\mathsf{count} = \mathsf{count} + 1$ and continue (go to start of while loop).

If $\mathsf{count} > \kappa$, then abort and output 0 as the output of the experiment.

Lemma 4. $|\Pr[\mathcal{D}_{\mathcal{V}}(\mathsf{Hybrid}_{1,\epsilon}) = 1] - \Pr[\mathcal{D}_{\mathcal{V}}(\mathsf{Hybrid}_{2,\epsilon}) = 1]| \leq \mathsf{negl}(\kappa)$

Proof. Assume, for contradiction, that there exist $\mathcal{V}$ and $\mathcal{D}_{\mathcal{V}}$ for which the claim is not true. We will use them to break honest-verifier zero-knowledge of the underlying Σ-protocol.

Consider a reduction $\mathcal{R}$ that in all iterations of Step 1, does the following: $\mathcal{R}$ first sets $e_{\mathsf{guess}} \xleftarrow{\$} \{0,1\}$. $\mathcal{R}$ then sends e_{guess} to the honest-verifier ZK challenger, and obtains (a^*, z^*), that is either sampled honestly using the witness w_1 and verifier challenge e_{guess}, or sampled using the honest-verifier ZK simulator and verifier challenge e_{guess}.

The reduction $\mathcal{R}$ then sends $a^*, \mathsf{OT}_2(z^*, z^*)$ to the distinguisher $\mathcal{D}_{\mathcal{V}}$ as the output of the challenger between $\mathsf{Hybrid}_{1,\epsilon}$ and $\mathsf{Hybrid}_{2,\epsilon}$. Note that the experiment corresponds to $\mathsf{Hybrid}_{1,\epsilon}$ if (a^*, z^*) is sampled honestly using the witness w_1, and to $\mathsf{Hybrid}_{2,\epsilon}$ if it is sampled using the honest-verifier ZK simulator. Then, $\mathcal{R}$ can just mirror the output of the distinguisher $\mathcal{D}_{\mathcal{V}}$ such that, $\Pr[\mathcal{D}_{\mathcal{V}}|\text{real } (a^*, z^*) = 1] - \Pr[\mathcal{D}_{\mathcal{V}}|\text{simulated } (a^*, z^*)] \geq \frac{1}{\mathsf{poly}(\kappa)}$ for some polynomial $\mathsf{poly}(\cdot)$, which is a contradiction. □

$\mathsf{Hybrid}_{4,\epsilon}$:

This hybrid is identical to $\mathsf{Hybrid}_{2,\epsilon}$ except that in Step_1, $a = f_1(x, w_2, r), z^0 = f_2(x, w_2, r, e_{\mathsf{guess}}), z^1 = f_2(x, w_2, r, e_{\mathsf{guess}})$. That is, the challenger starts using witness w_2 to compute $(a, z_{e_{\mathsf{guess}}})$.

Lemma 5. $|\Pr[\mathcal{D}_{\mathcal{V}}(\mathsf{Hybrid}_{3,\epsilon}) = 1] - \Pr[\mathcal{D}_{\mathcal{V}}(\mathsf{Hybrid}_{4,\epsilon}) = 1]| \leq \mathsf{negl}(\kappa)$

Proof. The proof of this lemma follows in the same way as the proof of Lemma 4. □

$\mathsf{Hybrid}_{5,\epsilon}$:

This is identical to $\mathsf{Hybrid}_{1,\epsilon}$, except that in Step_1, $a = f_1(x, w_2, r), z^0 = f_2(x, w_2, r, e = 0), z^1 =$

$f_2(x, w_2, r, e = 1)$. That is, the challenger now starts using the witness w_2 to compute (a, z^0, z^1), and the experiment is identical to an honest challenger using w_2 to generate the proof, except it aborts with probability $\frac{1}{2^\kappa}$.

Lemma 6. $|\Pr[\mathcal{D}_\mathcal{V}(\mathsf{Hybrid}_{4,\epsilon}) = 1] - \Pr[\mathcal{D}_\mathcal{V}(\mathsf{Hybrid}_{5,\epsilon}) = 1]| \leq \epsilon + \mathsf{negl}(\kappa)$

Proof. The proof of this lemma follows in the same way as the proof of Lemma 3. □

Hybrid_{w_2} :
This is the real experiment corresponding to generating the proof with witness w_2, where the challenger computes $a = f_1(x, w_2, r), z^0 = f_2(x, w_2, r, e = 0), z^1 = f_2(x, w_2, r, e = 1)$ and sends the prover message according to Figure 2.

Lemma 7. $|\Pr[\mathcal{D}_\mathcal{V}(\mathsf{Hybrid}_{5,\epsilon}) = 1] - \Pr[\mathcal{D}_\mathcal{V}(\mathsf{Hybrid}_{w_2}) = 1]| \leq \mathsf{negl}(\kappa)$

Proof. The proof of this lemma follows in the same way as the proof of Lemma 2. □

Suppose there exists a verifier V, a distinguisher $\mathcal{D}_V$, and a polynomial $p(\cdot)$ such that $\Pr[\mathcal{D}_V(\mathsf{Hybrid}_{w_1}) = 1] - \Pr[\mathcal{D}_V(\mathsf{Hybrid}_{w_2}) = 1] = \epsilon' \geq \frac{1}{p(\cdot)}$. Consider the family of hybrids parameterized by $\epsilon = \frac{\epsilon'}{7}$.

Then, the distinguisher must necessarily have advantage at least $\frac{\epsilon'}{6}$ in distinguishing one pair of consecutive hybrids between the six consecutive pairs Hybrid_{w_1} and Hybrid_{w_2}, which is a contradiction, since the distinguisher can have advantage at most $\epsilon + \mathsf{negl}(\kappa) = \frac{\epsilon'}{7} + \mathsf{negl}(\kappa)$ between each pair of consecutive hybrids. This completes the proof of witness indistinguishability.

5.4 Distributional Weak Zero Knowledge

In this section, we have the following theorem:

Theorem 7. *Assuming oblivious transfer (OT) secure against malicious PPT receivers, the protocol in Figure 1 is distributional weak zero-knowledge against non-adaptive verifiers.*

Proof. (Overview) The proof of weak zero-knowledge is more involved that WI, because weak ZK is not closed under parallel composition. We develop an inductive analysis and a simulation strategy that learns the receiver's challenge bit-by-bit.

Fix any PPT V^*, any distinguisher $\mathcal{D}$, any distribution $(\mathcal{X}, \mathcal{W}, \mathcal{Z})$, and any $\epsilon > 0$. We construct a simulator Sim_ϵ that obtains non-uniform advice z, $p_\epsilon = \mathsf{poly}(1/\epsilon)$ random instance-witness samples $(x_1^*, w_1^*), (x_2^*, w_2^*), \ldots (x_{p_\epsilon}^*, w_{p_\epsilon}^*)$ from the distribution $(\mathcal{X}, \mathcal{W})$. Or, if the distribution $(\mathcal{X}, \mathcal{W})$ is efficiently samplable, Sim_ϵ samples $(x_1^*, w_1^*), (x_2^*, w_2^*), \ldots (x_{p_\epsilon}^*, w_{p_\epsilon}^*)$ these on its own.

At a high level, the simulator uses these instances to approximately-learn the verifier's challenge string e (call this approximation e_{approx}), and then generates a transcript corresponding to a random $x \sim \mathcal{X}$, by using the honest-verifier ZK simulation strategy of the underlying Σ-protocol, corresponding to verifier challenge e_{approx}.

We now describe a sequence of hybrid experiments, where hybrid $\mathsf{Hybrid}_{\mathsf{Sim}_\epsilon}$ corresponds to our simulator Sim_ϵ.

5.4.1 Proof via Hybrid Experiments

$\mathsf{Hybrid}_0 := \mathsf{Hybrid}_{0,\epsilon}$:
This hybrid corresponds to an honest prover in the real world. That is, for $i \in [\kappa]$, the challenger samples $(x, w) \xleftarrow{\$} (\mathcal{X}, \mathcal{W})$ and sends $a_i = f_1(x, w, r_i), z_i^0 = f_2(x, w, r_i, e_i = 0), z_i^1 = f_2(x, w, r_i, e_i = 1)$ to the verifier.

$\mathsf{Hybrid}_{1,\epsilon}$:
This hybrid is indexed by a small error parameter $\epsilon = \frac{1}{\mathsf{poly}(\kappa)}$ for some polynomial $\mathsf{poly}(\cdot)$, and proceeds as follows. Fix the first message r of the verifier.

1. Run the algorithm in Figure 4 parameterized by $I = 1$ with oracle access to the distinguisher $\mathcal{D}$, and error parameter ϵ, to obtain guess $e_{\mathsf{approx},1}$ for the first bit of the verifier challenge.
2. Next, compute $a_1 = f_1(x, w, r_1), z_1^0 = f_2(x, w, r_1, e_{\mathsf{approx},1}), z_1^1 = f_2(x, w, r_1, e_{\mathsf{approx},1})$.
3. For $i \in [2, \kappa]$, compute (a_i, z_i^0, z_i^1) honestly.
4. Send prover message according to Figure 1 using the a_i, z_i computed for $i \in [\kappa]$.

$\mathsf{Hybrid}_{I,\epsilon}$ for $I \in [2, \kappa]$:
This hybrid is indexed by a small error parameter $\epsilon = \frac{1}{\mathsf{poly}(\kappa)}$ for some polynomial $\mathsf{poly}(\cdot)$, and proceeds as follows.

1. Run the algorithm in Figure 4 parameterized by I with oracle access to the verifier V, distinguisher $\mathcal{D}$, and error parameter ϵ, to obtain guess e_{approx} for the first I bits of the verifier challenge.
2. Next, for $i \in [I]$, compute $a_i = f_1(x, w, r_i), z_i^0 = f_2(x, w, r^i, e_{\mathsf{approx},i}), z_i^1 = f_2(x, w, r^i, e_{\mathsf{approx},i})$.
3. For $i \in [I + 1, \kappa]$, compute (a_i, z_i^0, z_i^1) honestly.
4. Send prover message according to Figure 1 using the a_i, z_i computed for $i \in [\kappa]$.

Lemma 8. *For all $I \in [0, \kappa - 1]$,*

$$\left|\Pr[\mathcal{D}_{\mathcal{V}} = 1 | \mathsf{Hybrid}_{I,\epsilon}] - \Pr[\mathcal{D}_{\mathcal{V}} = 1 | \mathsf{Hybrid}_{I+1,\epsilon}]\right| \leq \frac{\epsilon}{\kappa + 1}$$

Proof. The only difference between $\mathsf{Hybrid}_{I,\epsilon}$ and $\mathsf{Hybrid}_{I+1,\epsilon}$ is that in Hybrid_{I+1}, $e_{\mathsf{approx},I+1}$ is computed according to the algorithm in Figure 4 and the challenger sets $a_{I+1} = f_1(x, w, r_{I+1}), z_{I+1}^0 = z_{I+1}^1 = f_2(x, w, r_{I+1}, e_{\mathsf{guess},I+1})$, and then sends prover message according to Figure 1.

For the fixed verifier message OT_1, for $i \in [\kappa]$ and a fixed prefix $e_{\mathsf{prefix}} = e_{\mathsf{approx},[I]}$, denoting the first I bits of e_{approx},

- Let $\mathcal{D}_{e_{\mathsf{prefix}},0,x}$ denote the actual distribution output by the distinguisher when the challenger samples random $(x, w) \xleftarrow{\$} (\mathcal{X}, \mathcal{W})$,
  - For $j \leq I$, sets $a_j = f_1(x, w, r_j), z_j^0 = z_j^1 = f_2(x, w, r_j, e_j = e_{\mathsf{prefix},j})$, and using these sends prover message according to Figure 1. Here, $e_{\mathsf{prefix},j}$ denotes the j^{th} bit of e_{prefix}.
  - For $j = I + 1$, sets $a_j = f_1(x, w, r_j), z_j^0 = z_j^1 = f_2(x, w, r_j, e_j = 0)$, and using these sends prover message according to Figure 1.

Algorithm $\mathcal{M}^{V,\mathcal{D}_V}$ to approximate the verifier's challenge upto the I^{th} bit.

- Set $p = \kappa^2/\epsilon^3, i = 1, e_{\mathsf{approx}} = \perp$. For fixed verifier message r,
- While $i \leq I$, repeat:
  - Set $\mathcal{D}_0 = 0$ and for $j \in [p]$, repeat:
    1. For $k < i$, sample fresh randomness r_k and set $a_k = f_1(x_j^*, w_j^*, r_k), z_k^0 = z_k^1 = f_2(x_j^*, w_j^*, r_k, e = e_{\mathsf{approx},k})$.
    2. Sample fresh r_i, set $a_i = f_1(x_j^*, w_j^*, r_i), z_i^0 = z_i^1 = f_2(x_j^*, w_j^*, a, \mathbf{e} = \mathbf{0}, r_i)$.
    3. For $k \in [i+1, \kappa]$, sample fresh randomness r_k and honestly set $a_k = f_1(x_j^*, w_j^*, r_k), z_k^0 = f_2(x_j^*, w_j^*, a, e = 0, r_k), z_k^1 = f_2(x_j^*, w_j^*, a, e = 1, r_k)$
    4. Using (a, z) computed above, send prover message according to Figure 1, together with the instance x_j^*.
       Set $\mathcal{D}_0 = \mathcal{D}_0 + \frac{1}{p}$ if the output of the distinguisher $\mathcal{D}_V = 1$ (w.l.o.g., we assume that the distinguisher $\mathcal{D}_V$ outputs either 0 or 1).
  - Set $\mathcal{D}_1 = 0$ and for $j \in [p]$, repeat:
    1. For $k < i$, sample fresh randomness r_k and set $a_k = f_1(x_j^*, w_j^*, r_k), z_k^0 = z_k^1 = f_2(x_j^*, w_j^*, r_k, e = e_{\mathsf{approx},k})$.
    2. Sample fresh r_i, set $a_i = f_1(x_j^*, w_j^*, r_i), z_i^0 = z_i^1 = f_2(x_j^*, w_j^*, a, \mathbf{e} = \mathbf{1}, r_i)$.
    3. For $k \in [i+1, \kappa]$, sample fresh randomness r_k and honestly set $a_k = f_1(x_j^*, w_j^*, r_k), z_k^0 = f_2(x_j^*, w_j^*, a, e = 0, r_k), z_k^1 = f_2(x_j^*, w_j^*, a, e = 1, r_k)$
    4. Using (a, z) computed above, send prover message according to Figure 1, together with the instance x_j^*.
       Set $\mathcal{D}_1 = \mathcal{D}_1 + \frac{1}{p}$ if the output of the distinguisher $\mathcal{D}_V = 1$.
  - Set $\mathcal{D}_w = 0$ and for $j \in [p]$, repeat:
    1. For $k < i$, sample fresh randomness r_k and set $a_k = f_1(x_j^*, w_j^*, r_k), z_k^0 = z_k^1 = f_2(x_j^*, w_j^*, r_k, e = e_{\mathsf{approx},k})$.
    2. For $k \in [i, \kappa]$, sample fresh randomness r_k and honestly set $a_k = f_1(x_j^*, w_j^*, r_k), z_k^0 = f_2(x_j^*, w_j^*, a, e = 0, r_k), z_k^1 = f_2(x_j^*, w_j^*, a, e = 1, r_k)$.
    3. Using (a, z) computed above, send prover message according to Figure 1, together with the instance x_j^*.
       Set $\mathcal{D}_w = \mathcal{D}_w + \frac{1}{p}$ if the output of the distinguisher $\mathcal{D}_V = 1$.
  - If $|\mathcal{D}_1 - \mathcal{D}_w| \leq |\mathcal{D}_0 - \mathcal{D}_w|$, set $e_{\mathsf{approx},i} = 1$, else set $e_{\mathsf{approx},i} = 0$.
  - Set $i = i + 1$ and go to beginning of the while loop.
- Output e_{approx}.

Figure 4: Approximately Learning the Verifier's Challenge

- For $j \in [I+2, \kappa]$, sets $a_j = f_1(x, w, r_j), z_j^0 = f_2(x, w, r_j, e_j = 0)$, $z_j^1 = f_2(x, w, r_j, e_j = 1)$, and using these sends prover message according to Figure 1.

  We will abuse notation and also use $\mathcal{D}_{e_{\mathsf{prefix}},0,x}$ to denote the probability that the distinguisher outputs 1 in this situation.

- Let $\mathcal{D}_{e_{\mathsf{prefix}},1,x}$ denote the actual distribution output by the distinguisher when the challenger samples random $(x,w) \xleftarrow{\$} (\mathcal{X},\mathcal{W})$ and fresh randomness r,
  - For $j \leq I$, sets $a_j = f_1(x,w,r_j), z_j^0 = z_j^1 = f_2(x,w,r_j,e_j = e_{\mathsf{prefix},j})$, and using these sends prover message according to Figure 1.
  - For $j = I+1$, sets $a_j = f_1(x,w,r_j), z_j^0 = z_j^1 = f_2(x,w,r_j,e_j = 1)$, and using these sends prover message according to Figure 1.
  - For $j \in [I+2,\kappa]$, sets $a_j = f_1(x,w,r_j), z_j^0 = f_2(x,w,r_j,e_j = 0)$, $z_j^1 = f_2(x,w,r,e_j = 1,r_j)$, and using these sends prover message according to Figure 1.

    We will abuse notation and also use $\mathcal{D}_{e_{\mathsf{prefix}},1,x}$ to denote the probability that the distinguisher outputs 1 in this situation.
- Let $\mathcal{D}_{e_{\mathsf{prefix}},w,x}$ denote the actual distribution output by the distinguisher when the challenger samples random $(x,w) \xleftarrow{\$} (\mathcal{X},\mathcal{W})$ and fresh randomness r,
  - For $j \leq I$, sets $a = f_1(x,w,r_j), z_j^0 = z_j^1 = f_2(x,w,r_j,e_j = e_{\mathsf{prefix},j})$, and using these sends prover message according to Figure 1.
  - For $j \in [I+1,\kappa]$, sets $a = f_1(x,w,r_j), z_j^0 = f_2(x,w,r_j,e_j = 0)$, $z_j^1 = f_2(x,w,r_j,e_j = 1)$, and using these sends prover message according to Figure 1.

    We will abuse notation and also use $\mathcal{D}_{e_{\mathsf{prefix}},w,x}$ to denote the probability that the distinguisher outputs 1 in this situation.

Claim 3. *Either of the following statements is true:*

- *For any prefix* $e_{\mathsf{prefix}} \in \{0,1\}^I$, e $|\Pr[\mathcal{D}_{e_{\mathsf{prefix}},0,x} = 1] - \Pr[\mathcal{D}_{e_{\mathsf{prefix}},w,x} = 1]| \leq \mathsf{negl}(\kappa)$
- *For any prefix* $e_{\mathsf{prefix}} \in \{0,1\}^I$, $e|\Pr[\mathcal{D}_{e_{\mathsf{prefix}},1,x} = 1] - \Pr[\mathcal{D}_{e_{\mathsf{prefix}},w,x} = 1]| \leq \mathsf{negl}(\kappa)$

Proof. This claim follows from security of the OT. Assume, for contradiction, that there exist $\mathcal{V}$ and $\mathcal{D}_{\mathcal{V}}$ for which the claim is not true. We will use them to break receiver security of the underlying OT. Consider a reduction $\mathcal{R}$ that obtains the first OT message from $\mathcal{V}$ and forwards this message to the OT challenger.

The reduction picks $(x,w) \xleftarrow{\$} (\mathcal{X},\mathcal{W})$, $r \xleftarrow{\$} \{0,1\}^*$ and sets $a_{I+1} = f_1(x,w,r), z_{I+1}^0 = f_2(x,w,r,e = 0), z_{I+1}^1 = f_2(x,w,r,e = 1)$, and sends (z_{I+1}^0, z_{I+1}^1) to the OT challenger.

The OT challenger generates either the real message $\mathsf{OT}_2(z_{I+1}^0, z_{I+1}^1)$ corresponding to verifier input, or a simulated message $\mathsf{OT}_2(z^*,z^*)$, for some $z^* \in \{z_0, z_1\}$. The reduction sets all other (a^i, z_0^i, z_1^i) for $i \neq (I+1)$ according to Hybrid_I, and generates sender message accordingly.

Then, the output of distinguisher $\mathcal{D}_{\mathcal{V}}$ on input the simulated message is either distributed identically to $\mathcal{D}_{e_{\mathsf{prefix}},0,x}$ or $\mathcal{D}_{e_{\mathsf{prefix}},1,x}$ (depending upon whether z^* is 0 or 1). The reduction mirrors the output of $\mathcal{D}_{\mathcal{V}}$ and it holds that, $\Pr[\mathcal{D}_{\mathcal{V}} = 1|\text{real OT message}] - \Pr[\mathcal{D}_{\mathcal{V}} = 1|\text{simulated OT message}] \geq \frac{1}{\mathsf{poly}(\kappa)}$ for some polynomial $\mathsf{poly}(\cdot)$, for both $z^* = z_{I+1}^0$ and $z^* = z_{I+1}^1$, which is a contradiction. □

This claim establishes that for any prefix, *at least one* of the distributions $\mathcal{D}_{e_{\mathsf{prefix}},0,x}$ and $\mathcal{D}_{e_{\mathsf{prefix}},1,x}$ is negligibly close to $\mathcal{D}_{e_{\mathsf{prefix}},w,x}$.

If both $\mathcal{D}_{e_{\mathsf{prefix}},0,x}$ and $\mathcal{D}_{e_{\mathsf{prefix}},1,x}$ are $\epsilon/(\kappa+1)$-close to $\mathcal{D}_{e_{\mathsf{prefix}},w,x}$, then for any value of $e_{\mathsf{approx},I+1} \in \{0,1\}$, $\left|\Pr[\mathcal{D}_{\mathcal{V}} = 1|\mathsf{Hybrid}_{I,\epsilon}] - \Pr[\mathcal{D}_{\mathcal{V}} = 1|\mathsf{Hybrid}_{I+1,\epsilon}]\right| \leq \epsilon/(\kappa+1)$ and we are done.

Therefore, for the rest of this lemma, we restrict ourselves to the case where one and only one out of $\mathcal{D}_{e_{\mathsf{prefix}},0,x}$ and $\mathcal{D}_{e_{\mathsf{prefix}},1,x}$ is $\frac{\epsilon}{\kappa+1}$-close to $\mathcal{D}_{e_{\mathsf{prefix}},w,x}$. In particular, this also implies that $|\mathcal{D}_{e_{\mathsf{prefix}},0,x} - \mathcal{D}_{e_{\mathsf{prefix}},1,x}| > \frac{\epsilon}{\kappa+1}$.

If the challenger could "magically" set $e_{\mathsf{approx},I+1}$ to 0 if $\mathcal{D}_{e_{\mathsf{prefix}},0,x}$ was close to $\mathcal{D}_{e_{\mathsf{prefix}},w,x}$, and to 1 if $\mathcal{D}_{e_{\mathsf{prefix}},0,x}$ was close to $\mathcal{D}_{e_{\mathsf{prefix}},w,x}$, then again we would have that

$$\left|\Pr[\mathcal{D}_{\mathcal{V}}=1|\mathsf{Hybrid}_{I,\epsilon}]-\Pr[\mathcal{D}_{\mathcal{V}}=1|\mathsf{Hybrid}_{I+1,\epsilon}]\right|\leq\epsilon/(\kappa+1)$$

Unfortunately, the challenger cannot magically know which distributions are close, and will therefore have to approximate these distributions to obtain an answer. We now bound the probability that the challenger's approximation $e_{\mathsf{approx},I}$ is incorrect conditioned on $|\mathcal{D}_{e_{\mathsf{prefix}},0,x}-\mathcal{D}_{e_{\mathsf{prefix}},1,x}|>\frac{\epsilon}{\kappa+1}$, i.e., we show:

Claim 4.

$$\Pr\left[(e_{\mathsf{approx},I}=b)\middle|(\mathcal{D}_{e_{\mathsf{prefix}},1,x}-\mathcal{D}_{e_{\mathsf{prefix}},0,x}>\frac{\epsilon}{\kappa+1})\wedge(|\mathcal{D}_{\mathsf{correct},w}-\mathcal{D}_{\mathsf{correct},b,w}|>\frac{\epsilon}{\kappa+1})\right]\leq\mathsf{negl}(\kappa)$$

Proof. We note that for the $(I+1)^{th}$ iteration of Figure 4, $\mathcal{D}_0$ just consists of p random samples of a distribution with mean $\mathcal{D}_{e_{\mathsf{prefix}},0,x}$, $\mathcal{D}_1$ just consists of p random samples of a distribution with mean $\mathcal{D}_{e_{\mathsf{prefix}},1,x}$, and $\mathcal{D}_w$ just consists of p random samples of a distribution with mean $\mathcal{D}_{e_{\mathsf{prefix}},w,x}$.

Then, using a simple Chernoff bound, we have:

- $\Pr[\big(\mathcal{D}_0>\mathcal{D}_{e_{\mathsf{prefix}},0,x}(1+\alpha)\big)\vee\big(\mathcal{D}_0<\mathcal{D}_{e_{\mathsf{prefix}},0,x}(1-\alpha)\big)\leq 2\exp^{-\frac{\alpha^2 p\mathcal{D}_0}{2}}]$
- $\Pr[\big(\mathcal{D}_1>\mathcal{D}_{e_{\mathsf{prefix}},1,x}(1+\alpha)\big)\vee\big(\mathcal{D}_1<\mathcal{D}_{e_{\mathsf{prefix}},1,x}(1-\alpha)\big)\leq 2\exp^{-\frac{\alpha^2 p\mathcal{D}_1}{2}}]$
- $\Pr[\big(\mathcal{D}_w>\mathcal{D}_{e_{\mathsf{prefix}},w,x}(1+\alpha)\big)\vee\big(\mathcal{D}_w<\mathcal{D}_{e_{\mathsf{prefix}},w,x}(1-\alpha)\big)\leq 2\exp^{-\frac{\alpha^2 p\mathcal{D}_1}{2}}]$

Setting $\alpha=\frac{\epsilon}{2\kappa}$, and since $p=\frac{\kappa^2}{\epsilon^3}$, by a simple union bound we have that

$$\Pr\left[\left(|\mathcal{D}_{e_{\mathsf{prefix}},0,x}-\mathcal{D}_0|>\frac{\epsilon}{2\kappa}\right)\vee\left(|\mathcal{D}_{e_{\mathsf{prefix}},1,x}-\mathcal{D}_1|>\frac{\epsilon}{2\kappa}\right)\vee\left(|\mathcal{D}_{e_{\mathsf{prefix}},w,x}-\mathcal{D}_w|>\frac{\epsilon}{2\kappa}\right)\right]$$

$\leq 6\exp^{-\frac{1}{8\epsilon}}$. Since ϵ will always be set to $\frac{1}{\mathsf{poly}(\kappa)}$ for some polynomial $\mathsf{poly}(\cdot)$,

$$\Pr\left[\left(|\mathcal{D}_{e_{\mathsf{prefix}},0,x}-\mathcal{D}_0|>\frac{\epsilon}{2\kappa}\right)\vee\left(|\mathcal{D}_{e_{\mathsf{prefix}},1,x}-\mathcal{D}_1|>\frac{\epsilon}{2\kappa}\right)\vee\left(|\mathcal{D}_{e_{\mathsf{prefix}},w,x}-\mathcal{D}_w|>\frac{\epsilon}{2\kappa}\right)\right]$$

$\leq\mathsf{negl}(\kappa)$.

Recall that one of $\mathcal{D}_{e_{\mathsf{prefix}},0,x}$ and $\mathcal{D}_{e_{\mathsf{prefix}},w,x}$ is at least $\epsilon/(\kappa+1)$-far from $\mathcal{D}_{e_{\mathsf{prefix}},w,x}$, and the other is at most $\mathsf{negl}(\kappa)$-far. The bit $e_{\mathsf{approx},I}$ is estimated via $\mathcal{D}_0,\mathcal{D}_1,\mathcal{D}_w$ which each have error at most $\frac{\epsilon}{2\kappa}$, from the corresponding
$\mathcal{D}_{e_{\mathsf{prefix}},0,x},\mathcal{D}_{e_{\mathsf{prefix}},1,x},\mathcal{D}_{e_{\mathsf{prefix}},w,x}$. Thus,

$$\Pr\Big[e_{\mathsf{approx},I}=b\big|\ |(\mathcal{D}_{e_{\mathsf{prefix}},1,x}-\mathcal{D}_{e_{\mathsf{prefix}},0,x}|>\epsilon/(\kappa+1))\bigwedge$$

$$(|\mathcal{D}_{e_{\mathsf{prefix}},w,x}-\mathcal{D}_{e_{\mathsf{prefix}},b,x}|>\epsilon/(\kappa+1))\Big]\leq\mathsf{negl}(\kappa).$$

□

This completes the proof of the lemma.

□

$\mathsf{Hybrid}_{\mathsf{Sim},\epsilon}$: This hybrid corresponds to the interaction of the simulator with the verifier and distinguisher. It is indexed by a small error parameter $\epsilon = \frac{1}{\mathsf{poly}(\kappa)}$ for some polynomial $\mathsf{poly}(\cdot)$, and proceeds as follows.

1. Run the algorithm in Figure 4 parameterized by κ with oracle access to the verifier V, distinguisher $\mathcal{D}$, and error parameter ϵ, to obtain guess e_{approx} for the entire verifier challenge (all κ bits).
2. Next, for $i \in [\kappa]$, compute (without using the witness), $a_i = f_1(x, w, e_{\mathsf{approx},i}, r_i), z_i^0 = z_i^1 = f_2(x, w, e_{\mathsf{approx},i}, r_i)$ and send prover message according to Figure 1.

Lemma 9. $\left|\Pr[\mathcal{D}_{\mathcal{V}}(\mathsf{Hybrid}_{\kappa,\epsilon}) = 1] - \Pr[\mathcal{D}_{\mathcal{V}}(\mathsf{Hybrid}_{\mathsf{Sim},\epsilon}) = 1]\right| \leq \mathsf{negl}(\kappa)$

Proof. Assume, for contradiction, that there exist $\mathcal{V}$ and $\mathcal{D}_{\mathcal{V}}$ for which the claim is not true. We will use them to break honest-verifier zero-knowledge of the underlying Σ-protocol.

Consider a reduction $\mathcal{R}$ that does the following: $\mathcal{R}$ computes e_{approx} using Figure 4. $\mathcal{R}$ then sends e_{approx} to the honest-verifier ZK challenger, and obtains (a^*, z^*), that is either sampled honestly using the instance x and witness w, and the verifier challenge e_{approx}, or sampled using the honest-verifier ZK simulator and verifier challenge e_{approx}.

The reduction $\mathcal{R}$ then sends $a^*, \mathsf{OT}_2(z^*, z^*)$ to the distinguisher $\mathcal{D}_{\mathcal{V}}$ as the output of the challenger between $\mathsf{Hybrid}_{\kappa,\epsilon}$ and $\mathsf{Hybrid}_{\mathsf{Sim},\epsilon}$. Note that the experiment corresponds to $\mathsf{Hybrid}_{\kappa,\epsilon}$ if (a^*, z^*) is sampled honestly using the instance x and witness w, and to $\mathsf{Hybrid}_{\mathsf{Sim},\epsilon}$ if it is sampled using the honest-verifier ZK simulator. Then, $\mathcal{R}$ can just mirror the output of the distinguisher $\mathcal{D}_{\mathcal{V}}$ such that, $\Pr[\mathcal{D}_{\mathcal{V}} = 1|\text{real } (a^*, z^*)] - \Pr[\mathcal{D}_{\mathcal{V}} = 1|\text{simulated } (a^*, z^*)] \geq \frac{1}{\mathsf{poly}(\kappa)}$ for some polynomial $\mathsf{poly}(\cdot)$, which is a contradiction. □

Suppose the distinguisher $\mathcal{D}_V$ has a distinguishing advantage ϵ between Hybrid_0 and $\mathsf{Hybrid}_{\mathsf{Sim}_\epsilon}$, then it necessarily has advantage at least $\epsilon/(\kappa+1)$ in distinguishing one consecutive pair of hybrids between Hybrid_0 and $\mathsf{Hybrid}_{\mathsf{Sim}_\epsilon}$, which is a contradiction. This completes our proof. □

5.5 Strong Witness Indistinguishability

We note that the simulator's learning is monotone for two distributions, i.e., given two distributions $\mathcal{X}_1, \mathcal{X}_2$, then the view generated by a simulator Sim_ϵ that learns using samples from both distributions, $\mathcal{X}_1 \cup \mathcal{X}_2$, but outputs the simulation for a sample from $\mathcal{X}_1$, is indistinguishable from the view generated by a simulator Sim_ϵ that learns using samples from only $\mathcal{X}_1$ and then outputs the simulation for a sample from $\mathcal{X}_1$.

In other words, learning using additional distributions can only provide "more" information to the simulator. This observation coupled with the proof of weak ZK, directly implies strong witness indistinguishability, when the instances are sampled either from distribution $\mathcal{X}_1$ or from (an indistinguishable) distribution $\mathcal{X}_2$. This is because, the simulator can learn (in all hybrids) using instances from $\mathcal{X}_1 \cup \mathcal{X}_2$, and use these to simulate external samples generated according to either $\mathcal{X}_1$ or $\mathcal{X}_2$.

Corollary 8. *Assuming oblivious transfer (OT) secure against malicious PPT receivers, the protocol in Figure 1 is strong witness-indistinguishable against non-adaptive verifiers.*

5.6 Witness Hiding

It is easy to see that distributional weak zero-knowledge implies witness hiding. Suppose there exists a distribution $\mathcal{X}_\kappa$ and a PPT verifier V^* with auxiliary input z, that interacts with prover P. P samples random $X \sim \mathcal{X}_\kappa$ together with some witness $W(X)$ and generates a proof for V^* – such that V^* outputs a witness for $X \in \mathcal{X}$ with probability $\gamma = \frac{1}{\mathsf{poly}(\kappa)}$ for some polynomial $\mathsf{poly}(\cdot)$. Then, by the distributional weak zero-knowledge property, there exists a non-uniform simulator Sim_ϵ that uses V^* to output a witness for $X \sim \mathcal{X}$ with probability at least $\gamma - \epsilon$. Setting $\epsilon = \frac{\gamma}{2}$, we obtain a non-uniform polynomial size circuit $(\mathsf{Sim}_\epsilon, V^*)$ that outputs a witness for $X \sim \mathcal{X}$ with probability at least $\gamma/2$, which is a contradiction to the assumption in Definition 7. This implies the following corollary.

Corollary 9. *Assuming two-message oblivious transfer (OT) secure against malicious PPT receivers, the protocol in Figure 1 is witness-hiding against non-adaptive verifiers.*

5.7 Extensions

In this section, we sketch some simple extensions of our main results.

So far, we assumed that the Σ-protocol contains three messages, denoted by (a, e, z) and that these messages can be parsed as $a = (a_1, \ldots, a_\kappa)$, $e = (e_1, \ldots, e_\kappa)$, and $z = (z_1, \ldots, z_\kappa)$, where for each $i \in [\kappa]$, the triplet (a_i, e_i, z_i) are messages corresponding to an underlying Σ-protocol with a single-bit challenge (i.e., where $e_i \in \{0, 1\}$). We denote by f_1 and f_2 the functions that satisfy $a_i = f_1(x, w; r_i)$ and $z_i = f_2(x, w, r_i, e_i)$, where r_i is uniformly chosen randomness.

However, there is a large class of Σ-protocols that do not have this special structure. In Figure 5, we describe how any Σ-protocol can be compiled into 2-message WI and 2-message distributional weak ZK, assuming 2-message malicious-secure OT and garbled circuits. Our protocol is described in Figure 5.

Witness Indistinguishable and Distributional Weak Zero-Knowledge Argument
Prover Input: Instance $x \in L$, witness w such that $R_L(x, w) = 1$.
Verifier Input: Instance x, language L.

- **Verifier Message:** The verifier picks challenge $e \xleftarrow{\$} \{0,1\}^\kappa$ for the Σ-protocol, and for $i \in [\kappa]$, sends $\mathsf{OT}_{1,i}(e_i)$ in parallel. Each e_i is encrypted with a fresh OT instance.
- **Prover Message:** The prover samples a, and then constructs a garbled circuit $\mathsf{GC}(a, \cdot)$ for a function that on input e (the verifier challenge), outputs the corresponding message z of the underlying Σ-protocol. Let $(\mathsf{label}_i^0, \mathsf{label}_i^1)_{i \in [\kappa]}$ denote the labels of the garbled circuit. The prover sends $a, \mathsf{GC}(a, \cdot)$, together with $\mathsf{OT}_{2,i}(\mathsf{label}_i^0, \mathsf{label}_i^1)$ for all $i \in [\kappa]$.
- **Verifier Output:** The verifier V recovers z as the output of the garbled circuit on the labels obtained via OT, and outputs accept if (a, e, z) is an accepting transcript of the underlying Σ-protocol.

Figure 5: Two Round Argument System for NP from any Σ-Protocol

6 Three Round Protocols from Polynomial Assumptions.

Our three round protocol from polynomial assumptions is described in Figure 6. We denote the three messages of a Σ-protocol by (a, e, z), and assume that the Σ-protocol is a parallel repetition of protocols

with a single bit receiver challenge, and where the third message consists only of subsets of decommitments to messages committed in the first round. We further assume that a consists of a string of commitments, and z contains decommitment information for some of these commitments. We denote the i^{th} set of commitments (in the i^{th} parallel repetition of the Σ-protocol) by $a_i = \mathsf{commit}(h_i)$. We will implement this commitment differently in our protocol in Figure 6. We let com denote a non-interactive statistically binding commitment scheme, and let $\mathsf{wi} = (\mathsf{wi}_1, \mathsf{wi}_2, \mathsf{wi}_3)$ denote the messages of a 3-message delayed-input WI argument for NP. We also assume the existence of dense cryptosystems which are known based on DDH, QR, RSA, etc.

Theorem 10. *There exists a 3-message argument that satisfies distributional weak zero-knowledge, strong witness indistinguishability, witness hiding and witness indistinguishability against non-adaptive malicious verifiers, assuming either polynomially-hard DDH, N^{th}-residuosity or Quadratic Residuosity.*

6.1 Construction

Distributional Weak Zero-Knowledge Argument
Prover Input: Instance $x \in L$, witness w such that $R_L(x, w) = 1$.
Verifier Input: Instance x, language L.

- **Prover Message:** Pick $r_1, r_2, r_1', r_2' \xleftarrow{\$} \{0,1\}^*$, send $c_1 = \mathsf{com}(r_1; r_1'), c_2 = \mathsf{com}(r_2; r_2')$ using non-interactive statistically binding commitment com. Also, send wi_1 as the first message of the WI argument.
- **Verifier Message:** Pick challenge $e \xleftarrow{\$} \{0,1\}^\kappa$ for the Σ-protocol, and for $i \in [\kappa]$, send $\mathsf{OT}_{1,i}(e_i)$ in parallel. Each e_i is encrypted with a fresh OT instance. Additionally send $\widetilde{r}_1, \widetilde{r}_2 \xleftarrow{\$} \{0,1\}^*$, and send wi_2 as the second message of the WI argument.
- **Prover Message:** Send r_1, r_2 with wi_3 as the third message of the WI argument proving that $\exists r_1'$ such that $c_1 = \mathsf{com}(r_1; r_1')$ OR $\exists r_2'$ such that $c_2 = \mathsf{com}(r_2; r_2')$. Set $\mathsf{pk}_1 = r_1 \oplus \widetilde{r}_1, \mathsf{pk}_2 = r_2 \oplus \widetilde{r}_2$ as public keys for a dense cryptosystem.

  Define $\mathsf{commit}(M; R) = \mathsf{enc}_{\mathsf{pk}_1}(M; s_1), \mathsf{enc}_{\mathsf{pk}_2}(M; s_2)$ and $R = s_1 || s_2$, which is decommitted by revealing R. For $i \in [\kappa]$, and send $\mathsf{commit}(h_i), \mathsf{OT}_{2,i}(z_i^0, z_i^1)$ in parallel using the scheme commit. The decommitment information in z_i^0, z_i^1 corresponding to any commitment, only consists of the randomness R used to generate the commitment using commit.
- **Verifier Output:** The verifier V recovers z_i as the output of OT_i for $i \in [\kappa]$, and outputs accept if and only if wi is an accepting transcript and $(a_i, e_i, z_i)_{i \in [\kappa]}$ is an accepting transcript of the underlying Σ-protocol, according to the commitment scheme commit.

Figure 6: Three Round Argument System for NP

6.2 Adaptive Soundness

Theorem 11 (Adaptive Soundness)**.** *The protocol in Figure 6 satisfies adaptive soundness against malicious PPT provers.*

Proof. We will use any prover P^* that breaks soundness to break receiver security of the underlying oblivious transfer.

The prover P^* generates the first message of the protocol (including the commitments to r_1, r_2, and wi_1). The reduction then samples two strings $e_0, e_1 \xleftarrow{\$} \{0,1\}^*$ and then sends them to an external OT challenger. The external OT challenger picks $b \xleftarrow{\$} \{0,1\}$, and outputs message $\mathsf{ch} = \{\mathsf{OT}_1(e_{i,b})\}_{i\in[\kappa]}$, which the reduction forwards to the cheating prover P^* as the OT receiver challenge for the second round of the protocol. The reduction also picks $\tilde{r} \xleftarrow{\$} \{0,1\}^*$ and sends it together with honestly sampled wi_2 to P^*.

We assume, for the sake of contradiction, that the cheating prover P^* generates an accepting transcript with probability at least $p(\kappa) = \frac{1}{\mathsf{poly}(\kappa)}$: thus, it reveals r_1 and r_2 together with wi_3 that one out of r_1 and r_2 was correctly decommitted. If P^* aborts or generates a non-accepting transcript, then the reduction aborts, and if it doesn't, then the reduction records (r_1, r_2) and then rewinds the prover to the end of the first message. We will condition the rest of the experiment on extraction being successful at the end of this step, which occurs with probability at least $p - \mathsf{negl}(\kappa)$.

Next, the reduction does the following: Sample $\mathsf{pk}_1, \mathsf{pk}_2 \xleftarrow{\$} \{0,1\}^*$, together with their secret keys $\mathsf{sk}_1, \mathsf{sk}_2$. Then, set $\widetilde{r}_1, \widetilde{r}_2 = (\mathsf{pk}_1 \oplus r_1, \mathsf{pk}_2 \oplus r_2)$. Send verifier message using the same (externally obtained) OT challenge ch.

Suppose the prover responds and generates a third message such that wi_3 verifies.The reduction will use $\mathsf{sk}_1, \mathsf{sk}_2$ to extract the values inside the commitments using commit. By our assumption, in any such attempt, the prover generates an accepting transcript with probability at least $p(\kappa)$ for some $x \notin L$, and hence the extractor's success probability is at least $p(\kappa) - \mathsf{negl}(\kappa)$. If the extractor did not succeed, it outputs $\perp$. On the other hand, if the extractor succeeds in extracting the committed value, it proceeds as follows.

By special soundness of the underlying Σ-protocol against unbounded provers, in any accepting transcript, the values extracted from the commitments necessarily *encode* the verifier challenge according to the Σ-protocol, or, encode the witness itself. For example, when using the Blum Σ-protocol for Hamiltonicity (where the prover commits to the entire hamiltonian cycle and the permutation graph in within the first message of the Σ-protocol), when $x \notin L$, in any accepting transcript the values within the commitments can either correspond to a Hamiltonian cycle or a permutation depending on whether the verifier challenge bit for that index is 0 or 1. If the reduction extracts a witness (then $x \in L$), and the reduction outputs $\perp$.

If it does not extract a witness, then the reduction checks if the extracted value corresponds to verifier message being e_0 or e_1. If the value is neither e_0 or e_1, the reduction outputs $\perp$. Then, since the prover generates accepting transcripts with probability at least $p(\kappa) - \mathsf{negl}(\kappa)$, such a reduction still necessarily outputs e_b with probability at least $p(\kappa) - \mathsf{negl}(\kappa)$. On the other hand, e_{1-b} is information-theoretically hidden from the prover and the prover cannot guess e_{1-b} with probability greater than $\mathsf{negl}(\kappa)$. Since such a reduction directly contradicts receiver input-hiding security of the two-message oblivious transfer against PPT adversaries, this proves that the protocol satisfies adaptive soundness. □

6.3 Witness Indistinguishability

Theorem 12 (Witness Indistinguishability)**.** *The protocol in Figure 6 is witness-indistinguishable against malicious PPT verifiers.*

Proof. Recall that witness indistinguishability (WI) is closed under parallel composition [29], therefore it suffices to prove WI for a single repetition (i.e., for some $i \in [\kappa]$) of the protocol in Figure 6.

Our proof proceeds via a sequence of hybrid arguments, where, in an intermediate hybrid, we construct a distinguisher-dependent simulator, that learns (using both witnesses w_1 and w_2), an approximation for the verifier's challenge bit e. Upon learning the challenge, the simulator uses the honest-verifier ZK property to generate a simulated proof, without using any of the witnesses.

6.3.1 Proof via Hybrid Experiments

For an NP language L with corresponding relation R_L, consider an instance $x \in L$ and let w_1, w_2 be two witnesses such that $R_L(x, w_1) = 1$ and $R(x, w_2) = 1$.

We prove witness indistinguishability by contradiction: suppose there exists a distinguisher $\mathcal{D}_\mathcal{V}$ that distinguishes between experiments where the prover generates a proof using witness w_1 versus an experiment where the prover generates a proof using witness w_2, with advantage greater than ϵ'. We then consider a sequence of 6 hybrid experiments, indexed by error parameter $\epsilon = \epsilon'/7$, and by the previous statement, $\mathcal{D}_\mathcal{V}$ must distinguish two consecutive hybrids in the sequence with advantage greater than $\epsilon'/6$. But this is a contradiction, because we prove that the advantage of the distinguisher $\mathcal{D}_\mathcal{V}$ between every two consecutive hybrids (indexed by ϵ) is at most $\epsilon + \mathsf{negl}(\kappa)$.

Hybrid_{w_1} :
This hybrid corresponds to an honest prover that generates a proof for $x \in L$ using witness w_1. That is, the challenger computes $a = f_1(x, w_1, r), z^0 = f_2(x, w_1, r, e = 0), z^1 = f_2(x, w_1, r, e = 1)$, and sends the prover message according to Figure 6.

The output of this hybrid denoted by $\mathcal{D}_V(\mathsf{Hybrid}_{w_1})$ is the output of the distinguisher on input the view of the verifier in this experiment.

$\mathsf{Hybrid}_{1,\epsilon}$:
In this hybrid, with probability at least $1-2^{-\kappa}$, the view of the verifier will be identical to Hybrid_{w_1}, and with probability at most $2^{-\kappa}$, the output view is $\perp$. This ensures that the advantage of the distinguisher between the previous hybrid and this hybrids is at most $2^{-\kappa}$.

This hybrid is indexed by a small error parameter $\epsilon = \frac{1}{\mathsf{poly}(\kappa)}$ for some polynomial $\mathsf{poly}(\cdot)$, and proceeds as follows. The challenger sets a counter $\mathsf{count} = 0$ and while $\mathsf{count} \leq \kappa$, repeats the following two steps:

Step_1 : The first step of this experiment is the same as Hybrid_{w_1}, that is, first compute $a = f_1(x, w_1, r), z^0 = f_2(x, w_1, r, e = 0), z^1 = f_2(x, w_1, r, e = 1)$, and send prover message according to Figure 6. Denote the view of the verifier at the end of this step, by View_1.

Step_2 : Additionally, (unlike Hybrid_{w_1}), guess $e_{\mathsf{guess}} \xleftarrow{\$} \{0,1\}$. Then, run the algorithm in Figure 7 with oracle access to the V and distinguisher $\mathcal{D}$, and error parameter ϵ, to obtain e_{approx}. This corresponds, roughly, to approximating the verifier's challenge e, with error at most ϵ (this approximation is called e_{approx}).

If $e_{\mathsf{guess}} = e_{\mathsf{approx}}$, set the output of the distinguisher on input the view View_1, as the output of the experiment, and stop.

Else, set $\mathsf{count} = \mathsf{count} + 1$ and continue (go to start of while loop).

We will add a more detailed explanation of the approximating algorithm in the next hybrid. In this hybrid, it suffices to note that independently with probability at least $\frac{1}{2}$ in any iteration, $e_{\mathsf{guess}} = e_{\mathsf{approx}}$. Conditioned on $e_{\mathsf{guess}} = e_{\mathsf{approx}}$ in at least one iteration, the view of the distinguisher in this hybrid remains the same as Hybrid_{w_1}.

If $\mathsf{count} > \kappa$, abort and output 0 as the output of the experiment.

Lemma 10. $|\Pr[\mathcal{D}_\mathcal{V}(\mathsf{Hybrid}_{w_1}) = 1] - \Pr[\mathcal{D}_\mathcal{V}(\mathsf{Hybrid}_{1,\epsilon}) = 1]| \leq \mathsf{negl}(\kappa)$

Proof. The experiments are identical conditioned on the challenger not aborting. Since e_{guess} is sampled independently at random from e_{approx}, $\Pr[e_{\mathsf{guess}} = e_{\mathsf{approx}}] = \frac{1}{2}$ independently in every iteration. Thus, the advantage of the distinguisher is at most the probability of abort, which is $\frac{1}{2^\kappa}$. □

Algorithm $\mathcal{M}^{V,\mathcal{D}_V}$ to approximate the verifier's challenge.

1. Set $p = 1/\epsilon^3$.
2. For $w \in \{w_1, w_2\}$, and for the same fixed first two messages, repeat the following changing the last message each time:
   - Set $j = 1, \mathcal{D}_{0,w} = 0$ and repeat:
     (a) If $j = p$, then halt.
     (b) Sample fresh randomness r_j, set $a = f_1(x, w, r_j), z^0 = z^1 = f_2(x, w, e = 0, r_j)$, and send the prover message according to Figure 6.
     Set $\mathcal{D}_{0,w} = \mathcal{D}_{0,w} + \frac{1}{p}$ if the output of the distinguisher $\mathcal{D}_V = 1$ (w.l.o.g., we assume that the distinguisher $\mathcal{D}_V$ outputs either 0 or 1).
   - Set $j = 1, \mathcal{D}_{1,w} = 0$ and repeat:
     (a) If $j = p$, then halt.
     (b) Sample fresh randomness r_j, set $a = f_1(x, w, r_j), z^0 = z^1 = f_2(x, w, a, e = 1, r_j)$, and send the prover message according to Figure 6.
     Set $\mathcal{D}_{1,w} = \mathcal{D}_{1,w} + \frac{1}{p}$ if the output of the distinguisher $\mathcal{D}_V = 1$ (w.l.o.g., we assume that the distinguisher $\mathcal{D}_V$ outputs either 0 or 1).
   - Set $j = 1, \mathcal{D}_w = 0$ and repeat:
     (a) If $j = p$, then halt.
     (b) Sample fresh randomness r_j, set $a = f_1(x, w, r_j), z^0 = f_2(x, w, a, e = 0, r_j), z^1 = f_2(x, w, a, e = 1, r_j)$, and send the prover message according to Figure 6.
     Set $\mathcal{D}_w = \mathcal{D}_w + \frac{1}{p}$ if the output of the distinguisher $\mathcal{D}_V = 1$ (w.l.o.g., we assume that the distinguisher $\mathcal{D}_V$ outputs either 0 or 1).
3. If $|\mathcal{D}_{1,w_2} - \mathcal{D}_{w_2}| \geq |\mathcal{D}_{0,w_2} - \mathcal{D}_{w_2}| + \epsilon$, set $e_{\mathsf{approx}} = 0$.
4. Else if $|\mathcal{D}_{0,w_2} - \mathcal{D}_{w_2}| \geq |\mathcal{D}_{1,w_2} - \mathcal{D}_{w_2}| + \epsilon$, set $e_{\mathsf{approx}} = 1$.
5. Else if $|\mathcal{D}_{1,w_1} - \mathcal{D}_{w_1}| \geq |\mathcal{D}_{0,w_1} - \mathcal{D}_{w_1}| + \epsilon$, set $e_{\mathsf{approx}} = 0$.
6. Else set $e_{\mathsf{approx}} = 1$.

Figure 7: Approximately Learning the Verifier's Challenge

$\mathsf{Hybrid}_{2,\epsilon}$: In this hybrid, at an intuitive level, the challenger approximates the receiver's challenge (i.e., the bit e_r), and replaces the sender's oblivious transfer messages with simulated messages, corresponding to the approximated value of e_r.

That is, the challenger sends the first round message honestly, after which the (malicious) receiver sends the second message consisting of oblivious transfer message r, that could possibly correspond to $\mathsf{OT}_1(e_r)$ for some challenge bit e_r (or to no e_r at all). The challenger verifies that r is a valid message according to the underlying OT scheme. By security of the underlying OT against malicious receivers (refer Definition 2), for any fixed r sent by a malicious receiver that the challenger verifies to be a valid OT message, and any auxiliary input z, the following statement is true: Conditioned on r being the first message of R, either the distribution of receiver views $\mathsf{View}_R(\langle S(m_0, m_1), R(z)\rangle) \approx_c \mathsf{View}_R(\langle S(m_0, m_0), R(z)\rangle)$ for all (m_0, m_1), or, $\mathsf{View}_R(\langle S(m_0, m_1), R(z)\rangle) \approx_c \mathsf{View}_R(\langle S(m_1, m_1), R(z)\rangle)$

for all (m_0, m_1). That is, every r generated by a malicious receiver that verifies as a valid OT message, *behaves* like $\mathsf{OT}_1(e_r)$ for some bit e_r.

In other words, for any distinguisher that has input the view of the verifier, at least one out of $\mathsf{View}_R(\langle S(m_0, m_0), R(z)\rangle)$ and $\mathsf{View}_R(\langle S(m_1, m_1), R(z)\rangle)$ is $\mathsf{negl}(\kappa)$- close to the correct distribution $\mathsf{View}_R(\langle S(m_0, m_1), R(z)\rangle)$ (or, both could be $\mathsf{negl}(\kappa)$-close, which we do not discuss here because the distinguisher is a trivial distinguisher, and the proof becomes easier). When only one of the distributions $\mathsf{View}_R(\langle S(m_0, m_0), R(z)\rangle)$ and $\mathsf{View}_R(\langle S(m_1, m_1), R(z)\rangle)$ is close to the correct distribution, the challenger computes which distribution is close by sending *many* randomly chosen sender messages to the distinguisher, according to all three distributions, and learning whether the output of the distinguisher on $\mathsf{View}_R(\langle S(m_0, m_0), R(z)\rangle)$ or the output of the distinguisher on input $\mathsf{View}_R(\langle S(m_1, m_1), R(z)\rangle)$ is close to the output of the distinguisher on input $\mathsf{View}_R(\langle S(m_0, m_1, z)\rangle)$, upto error $\epsilon = \frac{1}{\mathsf{poly}(\kappa)}$.

Formally, the experiment is indexed by an error parameter $\epsilon = \frac{1}{\mathsf{poly}(\kappa)}$, and proceeds as follows.

Step_1 : First, guess $e_{\mathsf{guess}} \xleftarrow{\$} \{0, 1\}$. Next, compute $a = f_1(x, w_1, r), z^0 = f_2(x, w_1, r, e_{\mathsf{guess}}), z^1 = f_2(x, w_1, r, e_{\mathsf{guess}})$, and send prover message according to Figure 6.

Step_2 : Then, run the protocol in Figure 7 with error parameter ϵ to compute e_{approx}. If $e_{\mathsf{guess}} = e_{\mathsf{approx}}$, set the output of the distinguisher on input the view of the verifier in Step_1 of this experiment, as the output of the experiment $\mathcal{D}_V(\mathsf{Hybrid}_{2,\epsilon})$, and stop.

Else, set $\mathsf{count} = \mathsf{count} + 1$ and continue (go to start of while loop).

If $\mathsf{count} > \kappa$, abort and output 0 as the output of the experiment.

Lemma 11.

$$\left|\Pr[\mathcal{D}_{\mathcal{V}} = 1|\mathsf{Hybrid}_{1,\epsilon}] - \Pr[\mathcal{D}_{\mathcal{V}} = 1|\mathsf{Hybrid}_{2,\epsilon}]\right| \leq \epsilon + \mathsf{negl}(\kappa)$$

Proof. For the fixed verifier message $\mathsf{OT}_1(e)$ corresponding to the receiver challenge bit e_r, and witness $w \in \{w_1, w_2\}$,

- Let $\mathcal{D}_{\mathsf{correct},0,w}$ denote the actual distribution output by the distinguisher when the challenger samples fresh randomness r_j, sets $a = f_1(x, w, r_j), z^0 = z^1 = f_2(x, w, e = 0, r_j)$, and send the prover message according to Figure 6. We will abuse notation and also use $\mathcal{D}_{\mathsf{correct},0,w}$ to denote the probability that the distinguisher outputs 1 in this situation.

- Let $\mathcal{D}_{\mathsf{correct},1,w}$ denote the actual distribution output by the distinguisher when the challenger samples fresh randomness r_j, sets $a = f_1(x, w, r_j), z^0 = z^1 = f_2(x, w, e = 1, r_j)$, and send the prover message according to Figure 6. We will abuse notation and also use $\mathcal{D}_{\mathsf{correct},1,w}$ to denote the probability that the distinguisher outputs 1 in this situation.

- Let $\mathcal{D}_{\mathsf{correct},w}$ denote the actual distribution output by the distinguisher when the challenger samples fresh randomness r_j, sets $a = f_1(x, w, r_j), z^0 = f_2(x, w, e = 0, r_j), z^1 = f_2(x, w, a, e = 1, r_j)$, and send the prover message according to Figure 6. We will abuse notation and also use $\mathcal{D}_{\mathsf{correct},w}$ to denote the probability that the distinguisher outputs 1 in this situation.

- We note that $\mathcal{D}_{0,w_1}, \mathcal{D}_{1,w_1}, \mathcal{D}_{0,w_2}, \mathcal{D}_{1,w_2}, \mathcal{D}_{w_1}, \mathcal{D}_{w_2}$ denote the approximate distributions that the simulator lerans (refer Figure 7), while $\mathcal{D}_{\mathsf{correct},0,w}, \mathcal{D}_{\mathsf{correct},1,w}$ and $\mathcal{D}_{\mathsf{correct},w}$ denote the actual distributions (output by the distinguisher) themselves.

Claim 5. *Either of the following statements is true:*

- *For all witnesses w,*

$$|\Pr[\mathcal{D}_{\mathsf{correct},0,w} = 1] - \Pr[\mathcal{D}_{\mathsf{correct},w} = 1]| \leq \mathsf{negl}(\kappa)$$

- *For all witnesses w,*

$$|\Pr[\mathcal{D}_{\mathsf{correct},1,w} = 1] - \Pr[\mathcal{D}_{\mathsf{correct},w} = 1]| \leq \mathsf{negl}(\kappa)$$

Proof. Assume, for contradiction, that there exist $\mathcal{V}$ and $\mathcal{D}_{\mathcal{V}}$ for which the claim is not true. We will use them to break sender security of the underlying OT. Consider a reduction $\mathcal{R}$ that behaves honestly in the first round, then obtains the OT receiver message from $\mathcal{V}$ and forwards this message to the OT challenger.

The reduction sets $a = f_1(x, w, r_j), z^0 = f_2(x, w, e = 0, r_j), z^1 = f_2(x, w, e = 1, r_j)$, and sends (z^0, z^1) to the OT challenger.

The OT challenger generates either the real message $\mathsf{OT}_2(z^0, z^1)$, or a simulated message $\mathsf{OT}_2(z^*, z^*)$, for some (fixed) $z^* \in \{z^0, z^1\}$. The reduction forwards this message to the distinguisher.

The reduction mirrors the output of $\mathcal{D}_{\mathcal{V}}$ and it holds that,
$|\Pr[\mathcal{D}_{\mathcal{V}} = 1|\text{real OT message}] - \Pr[\mathcal{D}_{\mathcal{V}} = 1|\text{simulated OT message}]| \geq \frac{1}{\mathsf{poly}(\kappa)}$ for some polynomial $\mathsf{poly}(\cdot)$, for both $z^* = z^0$ and $z^* = z^1$, which is a contradiction. □

This claim establishes that *at least one* of the distributions $\mathcal{D}_{\mathsf{correct},0,w}$ and $\mathcal{D}_{\mathsf{correct},1,w}$ is negligibly close to $\mathcal{D}_{\mathsf{correct},w}$.

If both $\mathcal{D}_{\mathsf{correct},0,w}$ and $\mathcal{D}_{\mathsf{correct},1,w}$ are ϵ-close to $\mathcal{D}_{\mathsf{correct},w}$ for $w = w_1$, then for any value of e_{guess} in $\{0, 1\}$, $\left|\Pr[\mathcal{D}_{\mathcal{V}} = 1|\mathsf{Hybrid}_{1,\epsilon}] - \Pr[\mathcal{D}_{\mathcal{V}} = 1|\mathsf{Hybrid}_{2,\epsilon}]\right| \leq \epsilon + \mathsf{negl}(\kappa)$ and we are done.

Therefore, for the rest of this lemma, we restrict ourselves to the case where for $w = w_1$, one and only one out of $\mathcal{D}_{\mathsf{correct},0,w}$ and $\mathcal{D}_{\mathsf{correct},1,w}$ is ϵ-close to $\mathcal{D}_{\mathsf{correct},w}$. In particular, this also implies that $|\mathcal{D}_{\mathsf{correct},0,w} - \mathcal{D}_{\mathsf{correct},1,w}| > \epsilon$ for $w = w_1$.

If the challenger could "magically" set e_{guess} to 0 if $\mathcal{D}_{\mathsf{correct},0,w}$ was close to $\mathcal{D}_{\mathsf{correct},w}$, and to 1 if $\mathcal{D}_{\mathsf{correct},0,w}$ was close to $\mathcal{D}_{\mathsf{correct},w}$, then again we would have that

$$\left|\Pr[\mathcal{D}_{\mathcal{V}} = 1|\mathsf{Hybrid}_{1,\epsilon}] - \Pr[\mathcal{D}_{\mathcal{V}} = 1|\mathsf{Hybrid}_{2,\epsilon}]\right| \leq \epsilon.$$

Unfortunately, the challenger cannot magically know which distributions are close, and will therefore have to approximate these distributions to obtain an answer. We now bound the probability that the challenger's approximation e_{approx} is incorrect conditioned on $|\mathcal{D}_{\mathsf{correct},0,w} - \mathcal{D}_{\mathsf{correct},1,w}| > \epsilon$, i.e., we show:

Claim 6.

$$\Pr\left[(e_{\mathsf{approx}} = b)\middle|(|\mathcal{D}_{\mathsf{correct},1,w} - \mathcal{D}_{\mathsf{correct},0,w}| > \epsilon) \wedge (|\mathcal{D}_{\mathsf{correct},w} - \mathcal{D}_{\mathsf{correct},b,w}| > \epsilon)\right] \leq \mathsf{negl}(\kappa) \text{ where } w = w_1.$$

Proof. We note that for $w \in \{w_1, w_2\}, \mathcal{D}_0, \mathcal{D}_1, \mathcal{D}_w$ consist of p random samples from the distributions: $\mathcal{D}_{\mathsf{correct},0,w}, \mathcal{D}_{\mathsf{correct},1,w}, \mathcal{D}_{\mathsf{correct},w}$.

Then, using a simple Chernoff bound, we have that for $w \in \{w_1, w_2\}$:

- $\Pr[(\mathcal{D}_0 > \mathcal{D}_{\mathsf{correct},0,w}(1+\alpha)) \vee (\mathcal{D}_0 < \mathcal{D}_{\mathsf{correct},0,w}(1-\alpha))] \leq 2\exp^{-\frac{\alpha^2 p \mathcal{D}_{\mathsf{correct},0,w}}{2}}$
- $\Pr[(\mathcal{D}_1 > \mathcal{D}_{\mathsf{correct},1,w}(1+\alpha)) \vee (\mathcal{D}_1 < \mathcal{D}_{\mathsf{correct},1,w}(1-\alpha))] \leq 2\exp^{-\frac{\alpha^2 p \mathcal{D}_{\mathsf{correct},1,w}}{2}}$
- $\Pr[(\mathcal{D}_w > \mathcal{D}_{\mathsf{correct},w}(1+\alpha)) \vee (\mathcal{D}_w < \mathcal{D}_{\mathsf{correct},w}(1-\alpha))] \leq 2\exp^{-\frac{\alpha^2 p \mathcal{D}_{\mathsf{correct},w}}{2}}$

Setting $\alpha = \frac{\epsilon}{2}$, by a simple union bound we have that for $w \in \{w_1, w_2\}$,

$$\Pr\left[\left(|\mathcal{D}_{\mathsf{correct},0,w} - \mathcal{D}_0| > \frac{\epsilon}{2}\right) \vee \left(|\mathcal{D}_{\mathsf{correct},1,w} - \mathcal{D}_1| > \frac{\epsilon}{2}\right) \vee \left(|\mathcal{D}_{\mathsf{correct},w} - \mathcal{D}_w| > \frac{\epsilon}{2}\right)\right] \leq 6\exp^{-\frac{1}{2\epsilon}}$$

Since ϵ will always be set to $\frac{1}{\mathsf{poly}(\kappa)}$ for some polynomial $\mathsf{poly}(\cdot)$, for $w \in \{w_1, w_2\}$,

$$\Pr\left[\left(|\mathcal{D}_{\mathsf{correct},0,w} - \mathcal{D}_0| > \frac{\epsilon}{2}\right) \vee \left(|\mathcal{D}_{\mathsf{correct},1,w} - \mathcal{D}_1| > \frac{\epsilon}{2}\right) \vee \left(|\mathcal{D}_{\mathsf{correct},w} - \mathcal{D}_w| > \frac{\epsilon}{2}\right)\right] \leq 6\exp^{-\frac{1}{8\epsilon}}$$

We consider the event that the approximation e_{approx} is incorrect, and perform a case-analysis of this event.

- **Case I:** Suppose that the value e_{approx} was fixed in Step 5 or Step 6 (i.e., by using witness w_1 to approximate). Recall that one of $\mathcal{D}_{\mathsf{correct},0,w}$ and $\mathcal{D}_{\mathsf{correct},1,w}$ is at least ϵ-far from $\mathcal{D}_{\mathsf{correct},w}$, and the other is at most $\mathsf{negl}(\kappa)$-far, for $w = w_1$. The bit b is estimated via $\mathcal{D}_0, \mathcal{D}_1, \mathcal{D}_w$ which each have error at most $\frac{\epsilon}{2}$, from the corresponding distributions $\mathcal{D}_{\mathsf{correct},0,w}, \mathcal{D}_{\mathsf{correct},1,w}, \mathcal{D}_{\mathsf{correct},0,w}$. Thus, $\Pr[e_{\mathsf{approx}}$ is incorrect in Case I$] \leq \mathsf{negl}(\kappa)$.

- **Case II:** Suppose that the value e_{approx} was fixed in Step 3 or Step 4 of Figure 7 (i.e., by using witness w_2 to approximate). Recall that there exists a bit $\bar{b}$ such that $\mathcal{D}_{\mathsf{correct},\bar{b},x}$ is at least ϵ-far from $\mathcal{D}_{\mathsf{correct},w}$, and $\mathcal{D}_{\mathsf{correct},b,w}$ is at most $\mathsf{negl}(\kappa)$-far, for $w = w_1$. By Claim 5, even for $w = w_2$, $\mathcal{D}_{\mathsf{correct},b,w}$ is at most $\mathsf{negl}(\kappa)$-far from $\mathcal{D}_{\mathsf{correct},w}$.

  Then, e_{approx} is incorrect if Step 3 and Step 4 result in output $\bar{b} = 1-b$, which happens if and only if $|\mathcal{D}_{b,w_2} - \mathcal{D}_{w_2}| > |\mathcal{D}_{\bar{b},w_2} - \mathcal{D}_{w_2}| + \epsilon$. However, note that $\Pr[|\mathcal{D}_{b,w_2} - \mathcal{D}_{w_2}| > \epsilon \big| |\mathcal{D}_{\mathsf{correct},b,w} - \mathcal{D}_{\mathsf{correct},w}| = \mathsf{negl}(\kappa)] \leq \mathsf{negl}(\kappa)$ by the Chernoff bounds above. Therefore, Steps 3 and 4 result in incorrect output e_{approx} with probability at most $\mathsf{negl}(\kappa)$.

Summing up, $\Pr\left[e_{\mathsf{approx}} = b \big| |(\mathcal{D}_{\mathsf{correct},1,w} - \mathcal{D}_{\mathsf{correct},0,w}| > \epsilon) \wedge (|\mathcal{D}_{\mathsf{correct},w} - \mathcal{D}_{\mathsf{correct},b,w}| > \epsilon)\right] \leq \mathsf{negl}(\kappa)$ for $w = w_1$. □

This completes the proof of the lemma. □

$\mathsf{Hybrid}_{3,\epsilon}$: In this experiment, the challenger approximates the verifier challenge and conditions on $e_{\mathsf{guess}} = e_{\mathsf{approx}}$ as before. In $\mathsf{Hybrid}_{2,\epsilon}$, the challenger response $\mathsf{OT}_2(z_{e_{\mathsf{approx}}}, z_{e_{\mathsf{approx}}})$ was fixed and did not encode the witness, but the message a still possibly encoded witness w_1. In this hybrid, instead of sampling $(a, z_{e_{\mathsf{guess}}})$ using the witness w_1, the challenger simulates $(a, z_{e_{\mathsf{guess}}})$ without any witness, instead essentially relying on the honest-verifier ZK simulator of the underlying Σ-protocol.

Formally, the experiment is indexed by an error parameter $\epsilon = \frac{1}{\mathsf{poly}(\kappa)}$, and proceeds as follows.

Step_1 : First, guess $e_{\mathsf{guess}} \xleftarrow{\$} \{0,1\}$. Next, compute without using the witness w_1, $a = f_1(x, r, e_{\mathsf{guess}})$, $z^0 = z^1 = f_2(x, r, e_{\mathsf{guess}})$, and send prover message according to Figure 6.

Step_2 : Then, run the protocol in Figure 7 with error parameter ϵ to compute e_{approx}. If $e_{\mathsf{guess}} = e_{\mathsf{approx}}$, set the output of the distinguisher on input the view of the verifier in Step_1 of this experiment, as the output of the experiment $\mathcal{D}_V(\mathsf{Hybrid}_{3,\epsilon})$, and stop.

Else, set $\mathsf{count} = \mathsf{count} + 1$ and continue (go to start of while loop).

If $\mathsf{count} > \kappa$, then abort and output 0 as the output of the experiment.

Lemma 12. $|\Pr[\mathcal{D}_{\mathcal{V}}(\mathsf{Hybrid}_{1,\epsilon}) = 1] - \Pr[\mathcal{D}_{\mathcal{V}}(\mathsf{Hybrid}_{2,\epsilon}) = 1]| \leq \mathsf{negl}(\kappa)$

Proof. Assume, for contradiction, that there exist $\mathcal{V}$ and $\mathcal{D}_{\mathcal{V}}$ for which the claim is not true. We will use them to break hiding of the non-interactive commitment scheme com, or the IND-CPA security of the the dense cryptosystem (which essentially is responsible for honest-verifier ZK of the underlying Σ-protocol).

In the first sub-hybrid, the challenger $\mathcal{C}$ first conducts the experiment identically to $\mathsf{Hybrid}_{1,\epsilon}$ except that it obtains a public key pk_2 externally and sets the opening of the second commitment to $r_2' = \mathsf{pk}_2 \oplus \widetilde{r}_2$

(which is different from the value r_2 that it committed to in the first round). It uses r_1 as witness for wi. The view of a verifier in this experiment remains computationally indistinguishable from the view in $\mathsf{Hybrid}_{1,\epsilon}$ because of hiding of the commitment scheme com.

In the next sub-hybrid, only for Step_1, it changes the second set of encryptions $\mathsf{enc}_{\mathsf{pk}_2}$ to be computed without using the witness, corresponding to e_{guess}. Since these are never opened, this experiment remains indistinguishable by the IND-CPA security of the dense public key encryption scheme.

In the next sub-hybrid, the challenger opens r_2 honestly (instead of setting it as $\mathsf{pk}_2 \oplus \widetilde{r}_2$ for externally obtained pk_2. The view of a verifier in this experiment remains computationally indistinguishable from the view in $\mathsf{Hybrid}_{1,\epsilon}$ because of hiding of the commitment scheme com.

In the next sub-hybid, the challenger uses r_2 as witness for wi instead of using r_1. Since the experiment is conducted with a single set of (fixed) first two messages, or in other words, since the challenger is never rewound, this remains indistinguishable by the witness indistinguishability of wi.

Next, following the same sequence of sub-hybrids again, again, only for Step_1, the challenger also changes the first set of encryptions $\mathsf{enc}_{\mathsf{pk}_1}$ to be computed without using the witness, corresponding to verifier challenge e_{guess} for the Σ-protocol. At this point, the challenger computes the prover message in Step_1 without using the witness, exactly as in $\mathsf{Hybrid}_{2,\epsilon}$. This proves the lemma. □

$\mathsf{Hybrid}_{4,\epsilon}$:
This hybrid is identical to $\mathsf{Hybrid}_{2,\epsilon}$ except that in Step_1, $a = f_1(x, w_2, r), z^0 = f_2(x, w_2, r, e_{\mathsf{guess}}), z^1 = f_2(x, w_2, r, e_{\mathsf{guess}})$. That is, the challenger starts using witness w_2 to compute $(a, z_{e_{\mathsf{guess}}})$.

Lemma 13. $|\Pr[\mathcal{D}_{\mathcal{V}}(\mathsf{Hybrid}_{3,\epsilon}) = 1] - \Pr[\mathcal{D}_{\mathcal{V}}(\mathsf{Hybrid}_{4,\epsilon}) = 1]| \leq \mathsf{negl}(\kappa)$

Proof. The proof of this lemma follows in the same way as the proof of Lemma 12. □

$\mathsf{Hybrid}_{5,\epsilon}$:
This is identical to $\mathsf{Hybrid}_{1,\epsilon}$, except that in Step_1, $a = f_1(x, w_2, r), z^0 = f_2(x, w_2, r, e = 0), z^1 = f_2(x, w_2, r, e = 1)$. That is, the challenger now starts using the witness w_2 to compute (a, z^0, z^1), and the experiment is identical to an honest challenger using w_2 to generate the proof, except it aborts with probability $\frac{1}{2^\kappa}$.

Lemma 14. $|\Pr[\mathcal{D}_{\mathcal{V}}(\mathsf{Hybrid}_{4,\epsilon}) = 1] - \Pr[\mathcal{D}_{\mathcal{V}}(\mathsf{Hybrid}_{5,\epsilon}) = 1]| \leq \epsilon + \mathsf{negl}(\kappa)$

Proof. The proof of this lemma follows in the same way as the proof of Lemma 11. □

Hybrid_{w_2} :
This is the real experiment corresponding to generating the proof with witness w_2, where the challenger computes $a = f_1(x, w_2, r), z^0 = f_2(x, w_2, r, e = 0), z^1 = f_2(x, w_2, r, e = 1)$ and sends the prover message according to Figure 6.

Lemma 15. $|\Pr[\mathcal{D}_{\mathcal{V}}(\mathsf{Hybrid}_{5,\epsilon}) = 1] - \Pr[\mathcal{D}_{\mathcal{V}}(\mathsf{Hybrid}_{w_2}) = 1]| \leq \mathsf{negl}(\kappa)$

Proof. The proof of this lemma follows in the same way as the proof of Lemma 10. □

Suppose there exists a verifier V, a distinguisher $\mathcal{D}_V$, and a polynomial $p(\cdot)$ such that $\Pr[\mathcal{D}_V(\mathsf{Hybrid}_{w_1}) = 1] - \Pr[\mathcal{D}_V(\mathsf{Hybrid}_{w_2}) = 1] = \epsilon' \geq \frac{1}{p(\cdot)}$. Consider the family of hybrids parameterized by $\epsilon = \frac{\epsilon'}{7}$.

Then, the distinguisher must necessarily have advantage at least $\frac{\epsilon'}{6}$ in distinguishing one pair of consecutive hybrids between the six consecutive pairs Hybrid_{w_1} and Hybrid_{w_2}, which is a contradiction,

since the distinguisher can have advantage at most $\epsilon + \mathsf{negl}(\kappa) = \frac{\epsilon'}{7} + \mathsf{negl}(\kappa)$ between each pair of consecutive hybrids. This completes the proof of witness indistinguishability. Furthermore, the same protocol is also reusable witness indistinguishable, that is, it remains witness indistinguishable even when proofs of several instances are provided using the same first two messages – this follows by a simple sequence of hybrid experiments going over all instances. □

6.4 Distributional Weak Zero-Knowledge

Theorem 13 (Distributional Weak Zero-Knowledge)**.** *The protocol in Figure 6 is distributional weak zero-knowledge against malicious PPT verifiers.*

Proof. Fix any PPT V^*, any distinguisher $\mathcal{D}$, any distribution $(\mathcal{X}, \mathcal{W}, \mathcal{Z})$, and any $\epsilon > 0$. We construct a simulator Sim_ϵ that obtains non-uniform advice z, $p_\epsilon = \mathsf{poly}(1/\epsilon)$ random instance-witness samples $(x_1^*, w_1^*), (x_2^*, w_2^*), \ldots (x_{p_\epsilon}^*, w_{p_\epsilon}^*)$ from the distribution $(\mathcal{X}, \mathcal{W})$. Or, if the distribution $(\mathcal{X}, \mathcal{W})$ is efficiently samplable, Sim_ϵ samples $(x_1^*, w_1^*), (x_2^*, w_2^*), \ldots (x_{p_\epsilon}^*, w_{p_\epsilon}^*)$ these on its own.

At a high level, the simulator uses these instances to approximately-learn the verifier's challenge string e (call this approximation e_{approx}), and then generates a transcript corresponding to a random $x \sim \mathcal{X}$, by using the honest-verifier ZK simulation strategy of the underlying Σ-protocol, corresponding to verifier challenge e_{approx}. We now describe a sequence of hybrid experiments, where hybrid $\mathsf{Hybrid}_{\mathsf{Sim}_\epsilon}$ corresponds to our simulator Sim_ϵ.

6.4.1 Proof via Hybrid Experiments

$\mathsf{Hybrid}_0 := \mathsf{Hybrid}_{0,\epsilon}$:
This hybrid corresponds to an honest prover in the real world. That is, for $i \in [\kappa]$, the challenger samples $(x, w) \xleftarrow{\$} (\mathcal{X}, \mathcal{W})$ and sends $a_i = f_1(x, w, r_i), z_i^0 = f_2(x, w, r_i, e_i = 0), z_i^1 = f_2(x, w, r_i, e_i = 1)$ to the verifier.

$\mathsf{Hybrid}_{1,\epsilon}$:
This hybrid is indexed by a small error parameter $\epsilon = \frac{1}{\mathsf{poly}(\kappa)}$ for some polynomial $\mathsf{poly}(\cdot)$, and proceeds as follows. Fix the first message r of the verifier.

1. Run the algorithm in Figure 8 parameterized by $I = 1$ with oracle access to the distinguisher $\mathcal{D}$, and error parameter ϵ, to obtain guess $e_{\mathsf{approx},1}$ for the first bit of the verifier challenge.
2. Next, compute $a_1 = f_1(x, w, r_1), z_1^0 = f_2(x, w, r_1, e_{\mathsf{approx},1}), z_1^1 = f_2(x, w, r_1, e_{\mathsf{approx},1})$.
3. For $i \in [2, \kappa]$, compute (a_i, z_i^0, z_i^1) honestly.
4. Send prover message according to Figure 6 using the a_i, z_i computed for $i \in [\kappa]$.

$\mathsf{Hybrid}_{I,\epsilon}$ for $I \in [2, \kappa]$:
This hybrid is indexed by a small error parameter $\epsilon = \frac{1}{\mathsf{poly}(\kappa)}$ for some polynomial $\mathsf{poly}(\cdot)$, and proceeds as follows.

1. Run the algorithm in Figure 8 parameterized by I with oracle access to the verifier V, distinguisher $\mathcal{D}$, and error parameter ϵ, to obtain guess e_{approx} for the first I bits of the verifier challenge.
2. Next, for $i \in [I]$, compute $a_i = f_1(x, w, r_i), z_i^0 = f_2(x, w, r^i, e_{\mathsf{approx},i}), z_i^1 = f_2(x, w, r^i, e_{\mathsf{approx},i})$.

3. For $i \in [I+1, \kappa]$, compute (a_i, z_i^0, z_i^1) honestly.

4. Send prover message according to Figure 6 using the a_i, z_i computed for $i \in [\kappa]$.

Algorithm $\mathcal{M}^{V,\mathcal{D}_V}$ to approximate the verifier's challenge upto the I^{th} bit.

- Set $p = \kappa^2/\epsilon^3, i = 1, e_{\mathsf{approx}} = \perp$. For fixed verifier message r,
- While $i \leq I$, repeat:
  - Set $\mathcal{D}_0 = 0$ and for $j \in [p]$, repeat:
    1. For $k < i$, sample fresh randomness r_k and set $a_k = f_1(x_j^*, w_j^*, r_k), z_k^0 = z_k^1 = f_2(x_j^*, w_j^*, r_k, e = e_{\mathsf{approx},k})$.
    2. Sample fresh r_i, set $a_i = f_1(x_j^*, w_j^*, r_i), z_i^0 = z_i^1 = f_2(x_j^*, w_j^*, a, \mathbf{e} = \mathbf{0}, r_i)$.
    3. For $k \in [i+1, \kappa]$, sample fresh randomness r_k and honestly set $a_k = f_1(x_j^*, w_j^*, r_k), z_k^0 = f_2(x_j^*, w_j^*, a, e = 0, r_k), z_k^1 = f_2(x_j^*, w_j^*, a, e = 1, r_k)$
    4. Using (a, z) computed above, send prover message according to Figure 6, together with the instance x_j^*.
       Set $\mathcal{D}_0 = \mathcal{D}_0 + \frac{1}{p}$ if the output of the distinguisher $\mathcal{D}_V = 1$ (w.l.o.g., we assume that the distinguisher $\mathcal{D}_V$ outputs either 0 or 1).
  - Set $\mathcal{D}_1 = 0$ and for $j \in [p]$, repeat:
    1. For $k < i$, sample fresh randomness r_k and set $a_k = f_1(x_j^*, w_j^*, r_k), z_k^0 = z_k^1 = f_2(x_j^*, w_j^*, r_k, e = e_{\mathsf{approx},k})$.
    2. Sample fresh r_i, set $a_i = f_1(x_j^*, w_j^*, r_i), z_i^0 = z_i^1 = f_2(x_j^*, w_j^*, a, \mathbf{e} = \mathbf{1}, r_i)$.
    3. For $k \in [i+1, \kappa]$, sample fresh randomness r_k and honestly set $a_k = f_1(x_j^*, w_j^*, r_k), z_k^0 = f_2(x_j^*, w_j^*, a, e = 0, r_k), z_k^1 = f_2(x_j^*, w_j^*, a, e = 1, r_k)$
    4. Using (a, z) computed above, send prover message according to Figure 6, together with the instance x_j^*.
       Set $\mathcal{D}_1 = \mathcal{D}_1 + \frac{1}{p}$ if the output of the distinguisher $\mathcal{D}_V = 1$.
  - Set $\mathcal{D}_w = 0$ and for $j \in [p]$, repeat:
    1. For $k < i$, sample fresh randomness r_k and set $a_k = f_1(x_j^*, w_j^*, r_k), z_k^0 = z_k^1 = f_2(x_j^*, w_j^*, r_k, e = e_{\mathsf{approx},k})$.
    2. For $k \in [i, \kappa]$, sample fresh randomness r_k and honestly set $a_k = f_1(x_j^*, w_j^*, r_k), z_k^0 = f_2(x_j^*, w_j^*, a, e = 0, r_k), z_k^1 = f_2(x_j^*, w_j^*, a, e = 1, r_k)$.
    3. Using (a, z) computed above, send prover message according to Figure 6, together with the instance x_j^*.
       Set $\mathcal{D}_w = \mathcal{D}_w + \frac{1}{p}$ if the output of the distinguisher $\mathcal{D}_V = 1$.
  - If $|\mathcal{D}_1 - \mathcal{D}_w| \leq |\mathcal{D}_0 - \mathcal{D}_w|$, set $e_{\mathsf{approx},i} = 1$, else set $e_{\mathsf{approx},i} = 0$.
  - Set $i = i + 1$ and go to beginning of the while loop.
- Output e_{approx}.

Figure 8: Approximately Learning the Verifier's Challenge

Lemma 16. *For all $I \in [0, \kappa - 1]$,*

$$\left|\Pr[\mathcal{D}_{\mathcal{V}} = 1 | \mathsf{Hybrid}_{I,\epsilon}] - \Pr[\mathcal{D}_{\mathcal{V}} = 1 | \mathsf{Hybrid}_{I+1,\epsilon}]\right| \leq \frac{\epsilon}{\kappa + 1}$$

Proof. The only difference between $\mathsf{Hybrid}_{I,\epsilon}$ and $\mathsf{Hybrid}_{I+1,\epsilon}$ is that in Hybrid_{I+1}, $e_{\mathsf{approx},I+1}$ is computed according to the algorithm in Figure 8 and the challenger sets $a_{I+1} = f_1(x, w, r_{I+1}), z^0_{I+1} = z^1_{I+1} = f_2(x, w, r_{I+1}, e_{\mathsf{guess},I+1})$, and then sends prover message according to Figure 6.

For the fixed prover first message and fixed verifier message (which fixes OT_1), for $i \in [\kappa]$ and a fixed prefix $e_{\mathsf{prefix}} = e_{\mathsf{approx},[I]}$, denoting the first I bits of e_{approx},

- Let $\mathcal{D}_{e_{\mathsf{prefix}},0,x}$ denote the actual distribution output by the distinguisher when the challenger samples random $(x, w) \xleftarrow{\$} (\mathcal{X}, \mathcal{W})$,
  - For $j \leq I$, sets $a_j = f_1(x, w, r_j), z^0_j = z^1_j = f_2(x, w, r_j, e_j = e_{\mathsf{prefix},j})$, and using these sends prover message according to Figure 6. Here, $e_{\mathsf{prefix},j}$ denotes the j^{th} bit of e_{prefix}.
  - For $j = I + 1$, sets $a_j = f_1(x, w, r_j), z^0_j = z^1_j = f_2(x, w, r_j, e_j = 0)$, and using these sends prover message according to Figure 6.
  - For $j \in [I + 2, \kappa]$, sets $a_j = f_1(x, w, r_j), z^0_j = f_2(x, w, r_j, e_j = 0)$, $z^1_j = f_2(x, w, r_j, e_j = 1)$, and using these sends prover message according to Figure 6.

    We will abuse notation and also use $\mathcal{D}_{e_{\mathsf{prefix}},0,x}$ to denote the probability that the distinguisher outputs 1 in this situation.
- Let $\mathcal{D}_{e_{\mathsf{prefix}},1,x}$ denote the actual distribution output by the distinguisher when the challenger samples random $(x, w) \xleftarrow{\$} (\mathcal{X}, \mathcal{W})$ and fresh randomness r,
  - For $j \leq I$, sets $a_j = f_1(x, w, r_j), z^0_j = z^1_j = f_2(x, w, r_j, e_j = e_{\mathsf{prefix},j})$, and using these sends prover message according to Figure 6.
  - For $j = I + 1$, sets $a_j = f_1(x, w, r_j), z^0_j = z^1_j = f_2(x, w, r_j, e_j = 1)$, and using these sends prover message according to Figure 6.
  - For $j \in [I + 2, \kappa]$, sets $a_j = f_1(x, w, r_j), z^0_j = f_2(x, w, r_j, e_j = 0)$, $z^1_j = f_2(x, w, r, e_j = 1, r_j)$, and using these sends prover message according to Figure 6.

    We will abuse notation and also use $\mathcal{D}_{e_{\mathsf{prefix}},1,x}$ to denote the probability that the distinguisher outputs 1 in this situation.
- Let $\mathcal{D}_{e_{\mathsf{prefix}},w,x}$ denote the actual distribution output by the distinguisher when the challenger samples random $(x, w) \xleftarrow{\$} (\mathcal{X}, \mathcal{W})$ and fresh randomness r,
  - For $j \leq I$, sets $a = f_1(x, w, r_j), z^0_j = z^1_j = f_2(x, w, r_j, e_j = e_{\mathsf{prefix},j})$, and using these sends prover message according to Figure 6.
  - For $j \in [I + 1, \kappa]$, sets $a = f_1(x, w, r_j), z^0_j = f_2(x, w, r_j, e_j = 0)$, $z^1_j = f_2(x, w, r_j, e_j = 1)$, and using these sends prover message according to Figure 6.

    We will abuse notation and also use $\mathcal{D}_{e_{\mathsf{prefix}},w,x}$ to denote the probability that the distinguisher outputs 1 in this situation.

Claim 7. *Either of the following statements is true:*

- *For any prefix $e_{\mathsf{prefix}} \in \{0,1\}^I$, e $|\Pr[\mathcal{D}_{e_{\mathsf{prefix}},0,x} = 1] - \Pr[\mathcal{D}_{e_{\mathsf{prefix}},w,x} = 1]| \leq \mathsf{negl}(\kappa)$*
- *For any prefix $e_{\mathsf{prefix}} \in \{0,1\}^I$, $e|\Pr[\mathcal{D}_{e_{\mathsf{prefix}},1,x} = 1] - \Pr[\mathcal{D}_{e_{\mathsf{prefix}},w,x} = 1]| \leq \mathsf{negl}(\kappa)$*

Proof. This claim follows from security of the OT. Assume, for contradiction, that there exist $\mathcal{V}$ and $\mathcal{D}_{\mathcal{V}}$ for which the claim is not true. We will use them to break receiver security of the underlying OT. Consider a reduction $\mathcal{R}$ that obtains the first OT message from $\mathcal{V}$ and forwards this message to the OT challenger.

The reduction picks $(x, w) \xleftarrow{\$} (\mathcal{X}, \mathcal{W})$, $r \xleftarrow{\$} \{0,1\}^*$ and sets $a_{I+1} = f_1(x, w, r), z^0_{I+1} = f_2(x, w, r, e = 0), z^1_{I+1} = f_2(x, w, r, e = 1)$, and sends (z^0_{I+1}, z^1_{I+1}) to the OT challenger.

The OT challenger generates either the real message $\mathsf{OT}_2(z^0_{I+1}, z^1_{I+1})$ corresponding to verifier input, or a simulated message $\mathsf{OT}_2(z^*, z^*)$, for some $z^* \in \{z_0, z_1\}$. The reduction sets all other (a^i, z^i_0, z^i_1) for $i \neq (I+1)$ according to Hybrid_I, and generates sender message accordingly.

Then, the output of distinguisher $\mathcal{D}_{\mathcal{V}}$ on input the simulated message is either distributed identically to $\mathcal{D}_{e_{\mathsf{prefix}},0,x}$ or $\mathcal{D}_{e_{\mathsf{prefix}},1,x}$ (depending upon whether z^* is 0 or 1). The reduction mirrors the output of $\mathcal{D}_{\mathcal{V}}$ and it holds that, $\Pr[\mathcal{D}_{\mathcal{V}} = 1|\text{real OT message}] - \Pr[\mathcal{D}_{\mathcal{V}} = 1|\text{simulated OT message}] \geq \frac{1}{\mathsf{poly}(\kappa)}$ for some polynomial $\mathsf{poly}(\cdot)$, for both $z^* = z^0_{I+1}$ and $z^* = z^1_{I+1}$, which is a contradiction. □

This claim establishes that for any prefix, *at least one* of the distributions $\mathcal{D}_{e_{\mathsf{prefix}},0,x}$ and $\mathcal{D}_{e_{\mathsf{prefix}},1,x}$ is negligibly close to $\mathcal{D}_{e_{\mathsf{prefix}},w,x}$.

If both $\mathcal{D}_{e_{\mathsf{prefix}},0,x}$ and $\mathcal{D}_{e_{\mathsf{prefix}},1,x}$ are $\epsilon/(\kappa+1)$-close to $\mathcal{D}_{e_{\mathsf{prefix}},w,x}$, then for any value of $e_{\mathsf{approx},I+1} \in \{0,1\}$, $\left|\Pr[\mathcal{D}_{\mathcal{V}} = 1|\mathsf{Hybrid}_{I,\epsilon}] - \Pr[\mathcal{D}_{\mathcal{V}} = 1|\mathsf{Hybrid}_{I+1,\epsilon}]\right| \leq \epsilon/(\kappa+1)$ and we are done.

Therefore, for the rest of this lemma, we restrict ourselves to the case where one and only one out of $\mathcal{D}_{e_{\mathsf{prefix}},0,x}$ and $\mathcal{D}_{e_{\mathsf{prefix}},1,x}$ is $\frac{\epsilon}{\kappa+1}$-close to $\mathcal{D}_{e_{\mathsf{prefix}},w,x}$. In particular, this also implies that $|\mathcal{D}_{e_{\mathsf{prefix}},0,x} - \mathcal{D}_{e_{\mathsf{prefix}},1,x}| > \frac{\epsilon}{\kappa+1}$.

If the challenger could "magically" set $e_{\mathsf{approx},I+1}$ to 0 if $\mathcal{D}_{e_{\mathsf{prefix}},0,x}$ was close to $\mathcal{D}_{e_{\mathsf{prefix}},w,x}$, and to 1 if $\mathcal{D}_{e_{\mathsf{prefix}},0,x}$ was close to $\mathcal{D}_{e_{\mathsf{prefix}},w,x}$, then again we would have that

$$\left|\Pr[\mathcal{D}_{\mathcal{V}} = 1|\mathsf{Hybrid}_{I,\epsilon}] - \Pr[\mathcal{D}_{\mathcal{V}} = 1|\mathsf{Hybrid}_{I+1,\epsilon}]\right| \leq \epsilon/(\kappa+1)$$

Unfortunately, the challenger cannot magically know which distributions are close, and will therefore have to approximate these distributions to obtain an answer. We now bound the probability that the challenger's approximation $e_{\mathsf{approx},I}$ is incorrect conditioned on $|\mathcal{D}_{e_{\mathsf{prefix}},0,x} - \mathcal{D}_{e_{\mathsf{prefix}},1,x}| > \frac{\epsilon}{\kappa+1}$, i.e., we show:

Claim 8.

$$\Pr\left[(e_{\mathsf{approx},I} = b)\middle|(\mathcal{D}_{e_{\mathsf{prefix}},1,x} - \mathcal{D}_{e_{\mathsf{prefix}},0,x} > \frac{\epsilon}{\kappa+1}) \wedge (|\mathcal{D}_{\mathsf{correct},w} - \mathcal{D}_{\mathsf{correct},b,w}| > \frac{\epsilon}{\kappa+1})\right] \leq \mathsf{negl}(\kappa)$$

Proof. We note that for the $(I+1)^{th}$ iteration of Figure 8, $\mathcal{D}_0$ just consists of p random samples of a distribution with mean $\mathcal{D}_{e_{\mathsf{prefix}},0,x}$, $\mathcal{D}_1$ just consists of p random samples of a distribution with mean $\mathcal{D}_{e_{\mathsf{prefix}},1,x}$, and $\mathcal{D}_w$ just consists of p random samples of a distribution with mean $\mathcal{D}_{e_{\mathsf{prefix}},w,x}$.

Then, using a simple Chernoff bound, we have:

- $\Pr[\big(\mathcal{D}_0 > \mathcal{D}_{e_{\mathsf{prefix}},0,x}(1+\alpha)\big) \vee \big(\mathcal{D}_0 < \mathcal{D}_{e_{\mathsf{prefix}},0,x}(1-\alpha)\big) \leq 2\exp^{-\frac{\alpha^2 p\mathcal{D}_0}{2}}]$
- $\Pr[\big(\mathcal{D}_1 > \mathcal{D}_{e_{\mathsf{prefix}},1,x}(1+\alpha)\big) \vee \big(\mathcal{D}_1 < \mathcal{D}_{e_{\mathsf{prefix}},1,x}(1-\alpha)\big) \leq 2\exp^{-\frac{\alpha^2 p\mathcal{D}_1}{2}}]$
- $\Pr[\big(\mathcal{D}_w > \mathcal{D}_{e_{\mathsf{prefix}},w,x}(1+\alpha)\big) \vee \big(\mathcal{D}_w < \mathcal{D}_{e_{\mathsf{prefix}},w,x}(1-\alpha)\big) \leq 2\exp^{-\frac{\alpha^2 p\mathcal{D}_1}{2}}]$

Setting $\alpha = \frac{\epsilon}{2\kappa}$, and since $p = \frac{\kappa^2}{\epsilon^3}$, by a simple union bound we have that

$$\Pr\left[\left(|\mathcal{D}_{e_{\mathsf{prefix}},0,x} - \mathcal{D}_0| > \frac{\epsilon}{2\kappa}\right) \vee \left(|\mathcal{D}_{e_{\mathsf{prefix}},1,x} - \mathcal{D}_1| > \frac{\epsilon}{2\kappa}\right) \vee \left(|\mathcal{D}_{e_{\mathsf{prefix}},w,x} - \mathcal{D}_w| > \frac{\epsilon}{2\kappa}\right)\right]$$

$\leq 6\exp^{-\frac{1}{8\epsilon}}$. Since ϵ will always be set to $\frac{1}{\mathsf{poly}(\kappa)}$ for some polynomial $\mathsf{poly}(\cdot)$,

$$\Pr\left[\left(|\mathcal{D}_{e_{\mathsf{prefix}},0,x} - \mathcal{D}_0| > \frac{\epsilon}{2\kappa}\right) \vee \left(|\mathcal{D}_{e_{\mathsf{prefix}},1,x} - \mathcal{D}_1| > \frac{\epsilon}{2\kappa}\right) \vee \left(|\mathcal{D}_{e_{\mathsf{prefix}},w,x} - \mathcal{D}_w| > \frac{\epsilon}{2\kappa}\right)\right]$$

$\leq \mathsf{negl}(\kappa)$.

Recall that one of $\mathcal{D}_{e_{\mathsf{prefix}},0,x}$ and $\mathcal{D}_{e_{\mathsf{prefix}},w,x}$ is at least $\epsilon/(\kappa+1)$-far from $\mathcal{D}_{e_{\mathsf{prefix}},w,x}$, and the other is at most $\mathsf{negl}(\kappa)$-far. The bit $e_{\mathsf{approx},I}$ is estimated via $\mathcal{D}_0, \mathcal{D}_1, \mathcal{D}_w$ which each have error at most $\frac{\epsilon}{2\kappa}$, from the corresponding
$\mathcal{D}_{e_{\mathsf{prefix}},0,x}, \mathcal{D}_{e_{\mathsf{prefix}},1,x}, \mathcal{D}_{e_{\mathsf{prefix}},w,x}$. Thus,

$$\Pr\Big[e_{\mathsf{approx},I} = b \Big| \; (|\mathcal{D}_{e_{\mathsf{prefix}},1,x} - \mathcal{D}_{e_{\mathsf{prefix}},0,x}| > \epsilon/(\kappa+1)) \bigwedge$$
$$(|\mathcal{D}_{e_{\mathsf{prefix}},w,x} - \mathcal{D}_{e_{\mathsf{prefix}},b,x}| > \epsilon/(\kappa+1))\Big] \leq \mathsf{negl}(\kappa).$$

□

This completes the proof of the lemma.

□

$\mathsf{Hybrid}_{\mathsf{Sim},\epsilon}$: This hybrid corresponds to the interaction of the simulator with the verifier and distinguisher. It is indexed by a small error parameter $\epsilon = \frac{1}{\mathsf{poly}(\kappa)}$ for some polynomial $\mathsf{poly}(\cdot)$, and proceeds as follows.

1. Run the algorithm in Figure 8 parameterized by κ with oracle access to the verifier V, distinguisher $\mathcal{D}$, and error parameter ϵ, to obtain guess e_{approx} for the entire verifier challenge (all κ bits).
2. Next, for $i \in [\kappa]$, compute (without using the witness), $a_i = f_1(x, w, e_{\mathsf{approx},i}, r_i), z_i^0 = z_i^1 = f_2(x, w, e_{\mathsf{approx},i}, r_i)$ and send prover message according to Figure 6.

Lemma 17. $\left|\Pr[\mathcal{D}_{\mathcal{V}}(\mathsf{Hybrid}_{\kappa,\epsilon}) = 1] - \Pr[\mathcal{D}_{\mathcal{V}}(\mathsf{Hybrid}_{\mathsf{Sim},\epsilon}) = 1]\right| \leq \mathsf{negl}(\kappa)$

Proof. Assume, for contradiction, that there exist $\mathcal{V}$ and $\mathcal{D}_{\mathcal{V}}$ for which the claim is not true. We will use them to break hiding of the commitment scheme com or the IND-CPA security of the dense public-key cryptosystem. In the first sub-hybrid, the challenger $\mathcal{C}$ first conducts the experiment identically to $\mathsf{Hybrid}_{\kappa,\epsilon}$ except that it obtains a public key pk_2 externally and sets the opening r_2 to $\mathsf{pk}_2 \oplus \widetilde{r}_2$. It uses r_1 as witness for wi. The view of a verifier in this experiment remains computationally indistinguishable from the view in $\mathsf{Hybrid}_{\kappa,\epsilon}$ because of hiding of the commitment scheme com.

In the next sub-hybrid, *after* computing e_{approx} using Figure 8, it changes the second set of encryptions $\mathsf{enc}_{\mathsf{pk}_2}$ (only in the simulated transcript) to be computed without using the witness, corresponding to e_{approx}. Since these are never opened, this experiment remains indistinguishable by the IND-CPA security of the dense public key encryption scheme.

In the next sub-hybrid, the challenger opens r_2 honestly (instead of setting it as $\mathsf{pk}_2 \oplus \widetilde{r}_2$ for externally obtained pk_2). The view of a verifier in this experiment remains computationally indistinguishable from the view in $\mathsf{Hybrid}_{1,\epsilon}$ because of hiding of the commitment scheme com.

In the next sub-hybid, the challenger uses r_2 as witness for wi instead of using r_1. Since the experiment is conducted with a single set of (fixed) first two messages, or in other words, since the challenger is never rewound, this remains indistinguishable by the witness indistinguishability of wi.

Next, following the same sequence of sub-hybrids again, again, only for the final simulated transcript, the challenger also changes the first set of encryptions $\mathsf{enc}_{\mathsf{pk}_1}$ to be computed without using the witness, corresponding to verifier challenge e_{approx} for the Σ-protocol. This proves the lemma. □

Suppose the distinguisher $\mathcal{D}_V$ has a distinguishing advantage ϵ between Hybrid_0 and $\mathsf{Hybrid}_{\mathsf{Sim}_\epsilon}$, then it necessarily has advantage at least $\epsilon/(\kappa+1)$ in distinguishing one consecutive pair of hybrids between Hybrid_0 and $\mathsf{Hybrid}_{\mathsf{Sim}_\epsilon}$, which is a contradiction. This completes our proof. □

Remark 2 (Weak Resettable Security.)**.** *The protocol in Figure 6 satisfies weak resettable security according to Definition 9, when the WI proof* $(\mathsf{wi}_1, \mathsf{wi}_2, \mathsf{wi}_3)$ *is instantiated with an appropriate resettable WI (eg, a resettable reusable ZAP based on trapdoor permutations), and when the OT protocol is instantiated with an appropriate 2-message resettable secure OT (i.e. the construction in [52, 44] modified a là [16] such that the sender uses a PRF on the receiver's message to compute the randomness for sender message).*

This requires that the real transcript in the main execution, be ϵ-indistinguishable from the simulated transcript in the main execution, even in the presence of a verifier that obtains multiple "lookahead" transcripts, where all lookahead transcripts contain honestly generated proofs, using the same first message of the argument. Note that this first message only consists of non-interactive commitments $\mathsf{com}(r_1), \mathsf{com}(r_2)$. Thus in fact, the same sequence of hybrids, $\mathsf{Hybrid}_0, \ldots \mathsf{Hybrid}_{\kappa,\epsilon}$ goes through as before, where the hybrids remain indistinguishable because of resettable security of the ZAP and two-message OT. Finally, $\mathsf{Hybrid}_{\kappa,\epsilon}$ and $\mathsf{Hybrid}_{\mathsf{Sim},\epsilon}$ remain indistinguishable by the same proof as that of Lemma 17, except that in the first sub-hybrid corresponding to Lemma 17, the challenger sets the opening of the second commitment to $r_2 = (\mathsf{pk}_2 \oplus \widetilde{r}_2)$ for the main execution, and use the same opening for all look-ahead executions. Again, in the final sub-hybrid, this step is reversed and the challenger opens r_2 correctly, for the main as well as all look-ahead executions.

We also note that a malicious verifier that first observes the openings r_1, r_2 of the prover, and then rewinds and chooses $\widetilde{r}_1, \widetilde{r}_2$ for the main execution based on the values r_1, r_2, can directly break simulation security by extracting from commit. However, note that such a verifier is disallowed by our definition of weak resettable security, where the verifier is strongly non-adaptive, that is, it is committed to the second message of the main execution, *before* it even observes any lookahead executions. Indeed, our proof breaks down against such a verifier since we can no longer consistently set the opening of the second commitment to $r_2 = (\mathsf{pk}_2 \oplus \widetilde{r}_2)$ for $\widetilde{r}_2$ that was sent in the main execution.

Strong WI and Witness Hiding. The proof of strong witness indistinguishability and witness hiding, even in the weak resettable setting, follows from distributional weak ZK with extended simulation as in Section 5.4.

7 Three Round Extractable Commitments

We use three round adaptively sound weak ZK arguments against non-adaptive verifiers, according to Definition 5 to construct three-round extractable commitments.

We begin by modifying standard constructions of WIPoK to obtain a 3-round delayed-input reusable witness indistinguishable argument of knowledge, that ensures witness indistinguishability, even when the verifier obtains $\mathsf{poly}(1/\epsilon)$ third round messages, possibly corresponding to multiple different instances and witnesses. This is described in Figure 9.

7.1 Reusable Witness Indistinguishable Argument of Knowledge

We now prove that the protocol in Figure 9 is a reusable witness indistinguishable argument of knowledge.

Lemma 18. *The protocol in Figure 9 is an adaptive argument of knowledge.*

Reusable Witness Indistinguishable Argument of Knowledge.
Input: Prover $\mathcal{P}$ has input x and witness w such that $R(x, w) = 1$.

- Let $\mathsf{com} = \mathsf{overext\text{-}com}_1, \mathsf{overext\text{-}com}_2, \mathsf{overext\text{-}com}_3$ denote the three messages of a three round extractable commitment scheme, with over-extraction.
- Let $\mathsf{wi} = \mathsf{wi}_1, \mathsf{wi}_2, \mathsf{wi}_3$ denote the messages of an adaptively sound 3-round delayed-input reusable witness indistinguishable argument (this need not be an argument of knowledge).
- Let com denote a non-interactive statistically binding commitment scheme, which can be based on one-one one-way functions.

1. The prover $\mathcal{P}$ samples random $r_1, r_2 \xleftarrow{\$} \{0,1\}^{2\kappa}$ and sends $\mathsf{overext\text{-}com}_1(r_1), \mathsf{overext} - \mathsf{com}_1(r_2)$ to $\mathcal{V}$, together with wi_1.
2. $\mathcal{V}$ sends wi_2 to $\mathcal{P}$, together with $\mathsf{overext\text{-}com}_2$ for both extractable commitments.
3. $\mathcal{P}$ sends $\mathsf{overext\text{-}com}_3(r_1), \mathsf{overext\text{-}com}_3(r_2)$ to $\mathcal{V}$, together with instance x. $\mathcal{P}$ samples random $(a_1, a_2) \xleftarrow{\$} \{0,1\}^{2\kappa}$, and sends $a_1, x_1 = w \oplus \mathsf{PRF}(r_1, a_1), a_2, x_2 = w \oplus \mathsf{PRF}(r_2, a_2)$. $\mathcal{P}$ also sends wi_3 proving that:

$$(\exists r_1, a_1 \text{ such that } x_1 = w \oplus \mathsf{PRF}(r_1, a_1) \wedge \mathsf{overext\text{-}com}(r_1)$$
$$\text{is correctly constructed for } R(x, w) = 1) \bigvee$$
$$(\exists r_2, a_2 \text{ such that } x_2 = w \oplus \mathsf{PRF}(r_2, a_2) \wedge \mathsf{overext\text{-}com}(r_2)$$
$$\text{is correctly constructed for } R(x, w) = 1)$$

4. $\mathcal{V}$ accepts if and only if $\mathsf{wi}, \mathsf{com}$ verify.

Figure 9: Reusable Witness Indistinguishable Argument of Knowledge

Proof. (Sketch) For any accepting transcript (main thread) generated by the prover, because of adaptive soundness of wi, the i^{th} extractable commitment is generated as a valid extractable commitment to randomness r_i, such that $\mathsf{PRF}(r_i, a_i) \oplus x_i$ yields a witness for the corresponding (distributional) statement x, for some $i \in \{1, 2\}$. When the prover is rewound to the end of the first message, then with overwhelming probability, the prover produces $O(\kappa)$ accepting transcripts within κ^2 rewinds. Again, by a simple probabilistic argument, with overwhelming probability at least one of the accepting transcripts (in the rewinding thread) produce a valid extractable commitment for the same index i as the main thread (even though they may use different witnesses w).

Thus, by the extraction property of the over-extractable commitments, such an extractor can use the underlying extractor for the overextracting commitments, to extract r_i, and therefore extract a valid witness for the main thread.

We note that when wi is instantiated by a 2-round ZAP, we obtain a proof of knowledge assuming ZAPs, and additionally DDH/QR/N^{th} residuosity. On the other hand, when it is instantiated by our 2-round WI system, we obtain an argument of knowledge based on quasi-polynomial hardness of DDH/QR/N^{th} residuosity. When instantiated by our 3-round (reusable) WI argument described in Section 5.7, we obtain an argument of knowledge based on polynomial hardness of DDH/QR/N^{th} residuosity. □

Lemma 19. *The protocol in Figure 9 is reusable witness indistinguishable according to Definition 10.*

Proof. (Sketch) Suppose we want to prove witness indistinguishability for a subset S of statements generated in the third round, using witness w_1 versus w_2. We consider the following sequence of simple hybrid experiments:

Hybrid_0 : This corresponds to the real experiment where the prover generates the protocol transcript of Figure 9 using a valid witness w_1 in both extractable commitments, and generates proofs for multiple statements (given fixed first and second messages), according to the strategy in Figure 9. The prover picks $b \xleftarrow{\$} \{0,1\}$, and uses r_b as witness for the wi.

Hybrid_1 : In this hybrid, the prover samples $r' \xleftarrow{\$} \{0,1\}^\kappa$ and generates the $3-b^{th}$ $\mathsf{overext\text{-}com}$ to r'. However, it still generates $x_{3-b}, r_{3-b}, a_{3-b}$ the same way as in Hybrid_0. It continues to use r_b as witness for the wi. This hybrid is indistinguishable from Hybrid_0 by hiding of the $3-b^{th}$ $\mathsf{overext\text{-}com}$, because the receiver challenge for the extractable commitment is fixed for multiple third round messages sent by the challenger.

Hybrid_2 : Here, the prover generates $x_{3-b} \xleftarrow{\$} \{0,1\}^\kappa$ for all statements in S, uniformly at random. This hybrid is indistinguishable from Hybrid_1 by the security of the PRF using key r_{3-b}.

Hybrid_3 : In this hybrid, the prover generates $x_{3-b} = \mathsf{PRF}(r_{3-b}, a_{3-b}) \oplus w_2$, for all statements in S. Again, this hybrid is indistinguishable from Hybrid_2 by the security of the PRF using key r_{3-b}.

Hybrid_4 : Now, the prover generates $r' = r_{3-b}$ while generating the $3-b^{th}$ $\mathsf{overext\text{-}com}$ to r'. This hybrid is indistinguishable from Hybrid_3 by hiding of the $3-b^{th}$ $\mathsf{overext\text{-}com}$, because the receiver challenge for the extractable commitment is fixed for multiple third round messages sent by the challenger.

Hybrid_5 : The prover generates wi_2 using $x_{3-b}, r_{3-b}, a_{3-b}$ as witness for **all statements**, both in and outside the set S. This hybrid is indistinguishable from Hybrid_4 by the security of wi_2 generated for multiple instances, given fixed receiver message. Note that statements outside the set S still use w_1 as witness.

Hybrid_6 : In this hybrid, the prover samples $r' \xleftarrow{\$} \{0,1\}^\kappa$ and generates the b^{th} $\mathsf{over\text{-}extcom}$ to r'. This is indistinguishable from Hybrid_5 by hiding of $\mathsf{over\text{-}ext\text{-}com}$, because the receiver challenge for the extractable commitment is fixed for multiple third round messages sent by the challenger.

Hybrid_7 : Here, the prover generates $x_b \xleftarrow{\$} \{0,1\}^\kappa$ for all statements in S, uniformly at random. This hybrid is indistinguishable from Hybrid_6 by the security of the PRF using key r_b.

Hybrid_8 : In this hybrid, the prover generates $x_b = \mathsf{PRF}(r_b, a_b) \oplus w_2$, for all statements in S. Again, this hybrid is indistinguishable from Hybrid_7 by the security of the PRF using key r_b.

Hybrid_9 : Now, the prover generates $r' = r_b$ while generating the b^{th} $\mathsf{overext\text{-}com}$ to r'. This hybrid is indistinguishable from Hybrid_8 by hiding of the b^{th} $\mathsf{overext\text{-}com}$, because the receiver challenge for the extractable commitment is fixed for multiple third round messages sent by the challenger. This is the final hybrid where w_2 is used as a witness for all statements in S, while w_1 continues to be used for all statements outside of the set S. □

Next, in Figure 10, we construct a 3-round weak ZK adaptive argument of knowledge against non-adaptive verifiers and extended simulation security, by composing our weak ZK argument with extended

simulation security, with the reusable delayed-input witness indistinguishable argument of knowledge.

7.2 Distributional Weak ZK/Strong WI Argument of Knowledge

We now prove that the protocol in Figure 10 is a distributional weak zero-knowledge argument of knowledge, and a distributional strong WI argument of knowledge.

Weak Zero-Knowledge Argument of Knowledge.
Input: Prover $\mathcal{P}$ has input a distribution $(\mathcal{X}, \mathcal{W})$.
Let $\mathsf{wzk} = \mathsf{wzk}_1, \mathsf{wzk}_2, \mathsf{wzk}_3$ denote an adaptively sound three round weak ZK argument with extended simulation security, against non-adaptive verifiers.
Let $\mathsf{wipok} = \mathsf{wipok}_1, \mathsf{wipok}_2, \mathsf{wipok}_3$ denote an adaptively sound delayed-input three round reusable witness indistinguishable argument of knowledge.
Let $\mathsf{com} = \mathsf{com}_1, \mathsf{com}_2$ denote a non-interactive statistically binding commitment scheme, which can be based on injective one-way functions.

1. The prover $\mathcal{P}$ sends $\mathsf{wipok}_1, \mathsf{wzk}_1$ to $\mathcal{V}$.
2. $\mathcal{V}$ sends $\mathsf{wipok}_2, \mathsf{wzk}_2$ to $\mathcal{P}$, together with com_1.
3. $\mathcal{P}$ samples $(x, w) \xleftarrow{\$} (\mathcal{X}, \mathcal{W})$, together with $c = \mathsf{com}_2(0; r)$ and computes $\mathsf{wzk}_3, \mathsf{wipok}_3$ where:

   $$\mathsf{wipok} \text{ proves that } \exists r \text{ such that } c = \mathsf{com}(1; r) \text{ OR } x \in L$$

   $$\mathsf{wzk} \text{ proves that } \exists r \text{ such that } c = \mathsf{com}(0; r)$$

4. $\mathcal{V}$ accepts if and only if $\mathsf{wipok}, \mathsf{com}$ and wzk verify.

Figure 10: Weak Zero-Knowledge Argument of Knowledge

Lemma 20. *The protocol in Figure 10 is an adaptive argument of knowledge.*

Proof. (Sketch) Suppose there exists an adversarial prover that adaptively picks a statement x and generates an argument using the protocol in Figure 10. Then, by the adaptive argument of knowledge property of wipok, there exists an extractor that extracts a witness for either $x \in L$ or $c = \mathsf{com}(1; r)$. Moreover, by the adaptive soundness of wzk against unbounded provers, with overwhelming probability in the real and rewinding executions, if the transcript is accepted by a verifier, then $c = \mathsf{com}(0; r)$ for some randomness r. Furthermore, by the statistical binding property of the commitment, with probability at least $1 - \mathsf{negl}(n)$, there cannot exist a string r such that $c = \mathsf{com}(1; r)$. Thus, the witness extracted by the extractor must necessarily be a witness for $x \in L$. That is, the extractor succeeds in extracting a witness for (possibly adaptively chosen) $x \in L$ with probability at least $1 - \mathsf{negl}(\kappa)$. □

Lemma 21. *The protocol in Figure 10 is weak zero knowledge and strong witness indistinguishable.*

Proof. (Sketch) The proof of weak zero-knowledge/strong WI of the protocol proceeds in the following sequence of hybrid experiments:

Hybrid_0 : This corresponds to real execution where the prover generates the protocol transcript of Figure 10 using valid witnesses for instances x.

Hybrid_1 : This experiment is the same as Hybrid_0, except that the simulator generates a simulated wzk proof by learning the receiver challenge, rewinding and observing the output of the distinguisher multiple times. The view of the verifier and distinguisher in this hybrid is indistinguishable from Hybrid_1, by the weak ZK/strong WI property of wzk.

Hybrid_2 : This experiment is the same as Hybrid_1, except that the simulator generates $c = \mathsf{com}(1; r)$ for $r \xleftarrow{\$} \{0,1\}$. The wzk proof is simulated the same way as Hybrid_1, i.e., by sending multiple proofs using multiple $c = \mathsf{com}(0; r')$ for third round messages, together with wiaok using $(\tilde{x}, \tilde{w})$ chosen from the distribution, and learning the output of the distinguisher. The view of the verifier and distinguisher in this hybrid is indistinguishable from Hybrid_1, by the hiding property of the commitment used to obtain string c.

Hybrid_3 : This experiment is the same as Hybrid_1, except that the simulator generates wiaok_3 in the main thread using $c = \mathsf{com}(1; r)$ as witness instead of using (x, w) as witness (but only in the main thread). The wzk proof is simulated the same way as Hybrid_1, i.e., by sending multiple proofs using multiple $c = \mathsf{com}(0; r')$ for third round messages, together with wiaok using $(\tilde{x}, \tilde{w})$ chosen from the distribution, and learning the output of the distinguisher. The view of the verifier and distinguisher in this hybrid is indistinguishable from Hybrid_1, by the reusable WI property of wiaok.

This hybrid also describes the simulation strategy for the simulator of the WZKAoK. Since the actual witness for the instance x in the main thread is no longer required in this hybrid, in order to prove strong WI, the same sequence of hybrids can be repeated in reverse order, after (indistinguishably) changing the instance, similar to Section 5.5. □

7.3 Extractable Commitments

We construct three round extractable commitments in Figure 11, where an extractor extracts the value committed by a (possibly adversarial) committer with overwhelming probability.

Three Round Extractable Commitment Scheme.
Committer Input. Committer C has input message m.
Let $\mathsf{wzkaok} = \mathsf{wzkaok}_1, \mathsf{wzkaok}_2, \mathsf{wzkaok}_3$ denote a three round weak ZK adaptive argument of knowledge with extended simulation security, against non-adaptive verifiers.
Let com denote a non-interactive statistically binding commitment scheme, which can be based on one-one one-way functions.

Commit Phase.

1. The committer $\mathcal{C}$ sends wzkaok_1 to the receiver $\mathcal{R}$.
2. $\mathcal{R}$ sends wzkaok_2 to $\mathcal{C}$.
3. $\mathcal{C}$ sends $c = \mathsf{com}(m; r)$ and computes wzkaok_3 proving $\exists(m, r)$ such that $c = \mathsf{com}(m; r)$.
4. $\mathcal{R}$ accepts the commitment if and only if wzkaok and com verify.

Decommit Phase. $\mathcal{C}$ sends (m, r) to $\mathcal{R}$, who accepts iff it is a valid decommitment of c.

Figure 11: Extractable Commitments

Lemma 22. *The scheme in Figure 11 is extractable, without over-extraction or under-extraction.*

Proof. (Sketch) Consider a main-thread transcript generated by a committer. By the argument of knowledge property of wzkaok, given any fixed main-thread transcript, a unique (m, r) can be extracted from the weak ZK argument of knowledge with overwhelming probability, from any transcript generated by a (possibly unbounded) committer. □

Lemma 23. *The protocol in Figure 11 is computationally hiding.*

Proof. (Sketch) The proof of hiding of the scheme follows from the distributional strong WI property of the underlying wzkaok and the hiding of com. This can be proved via the following sequence of hybrid experiments:
Hybrid_0: The challenger generates an honest commitment to message m according to the strategy in Figure 11.
Hybrid_1: The challenger replaces this with an honest commitment to message 0 according to the strategy in Figure 11. Since $\mathsf{com}(m)$ is indistinguishable from $\mathsf{com}(0)$, the computational hiding property follows from strong WI of the underlying wzkaok for indistinguishable instance distributions.

In some more detail (opening up the strong WI argument), the challenger can begin by simulating the wzk protocol using extended simulation, i.e., by using instances from both distributions $\mathsf{com}(m; r)$ and $\mathsf{com}(0; r)$ to honestly generate the WIPoK and learn the receiver's challenge. Next, in the main thread, the instance $\mathsf{com}(m; r)$ can be replaced with $\mathsf{com}(0; r)$ externally, while simulating the wzk argument6. Finally, this can be replaced by an honestly generated commitment to 0 according to Figure 11. □

8 Two-Party Computation

In this section, we construct three round two-party secure computation between two parties where only the receiver obtains the output, with distributional distinguisher-dependent simulation security for the receiver and (standard) simulation security for the other party. Our construction is described in Figure 12. We show that the same protocol is input-indistinguishable with respect to a malicious receiver.

We prove that this protocol satisfies (standard) simulation security against a malicious sender, and distributional distinguisher-dependent security against a malicious receiver. We also remark that a two-round version of the same protocol (with the three round WZKPoK replaced with a WZK argument), gives a way of performing secure two-party computation in two rounds, with (efficient) distributional distinguisher-dependent simulation against malicious receivers, and super-polynomial simulation against malicious senders (or polynomial-time simulation against semi-honest senders).

Theorem 14. *The protocol in Figure 12 is a secure protocol for two party computation with distributional distinguisher-dependent security against a malicious receiver and standard simulation security against a malicious sender.*

To prove Theorem 14, we describe the simulation strategy against a malicious sender in Figure 13 and the simulation strategy against a malicious receiver in Figure 14.

Lemma 24. *The view* $\mathsf{IDEAL}_{\mathcal{F},\mathsf{Sim}}$ *generated by the simulator* Sim *in Figure 13 is such that for all PPT distinguishers* $\mathcal{D}$*,*

$$\Big|\Pr\left[\mathcal{D}\left(\mathsf{IDEAL}_{\mathcal{F},\mathsf{Sim}}(\kappa, x, y)_{\kappa\in\mathbb{N},(x,y)s.t.|x|=|y|}\right)\right] - \Pr\left[\mathcal{D}\left(\mathsf{REAL}_{\Pi,\mathcal{S}}(\kappa, x, y)_{\kappa\in\mathbb{N},(x,y)s.t.|x|=|y|}\right)\right]\Big| \leq \mathsf{negl}(\kappa)$$

Three Round Secure Two Party Computation.
Sender Input. Sender $\mathcal{S}$ has (public) input distribution $\mathcal{Q}$.
Receiver Input. Receiver $\mathcal{R}$ has private input distribution $\mathcal{R}$.

- Let $\mathsf{wzkaok} = \mathsf{wzkaok}_1, \mathsf{wzkaok}_2, \mathsf{wzkaok}_3$ denote a three round weak ZK adaptive argument of knowledge against non-adaptive verifiers.
- Let $\mathsf{OT} = \mathsf{OT}_1, \mathsf{OT}_2$ denote the messages of a two-round OT, according to Definition 2.
- Let $\{\mathsf{GC}, (\mathsf{label}_i)_{i\in[\kappa]}\}(f)$ denote a garbled circuit with its labels generated corresponding to functionality f.

Protocol Description.

1. The sender $\mathcal{S}$ sends wzkaok_1 to the receiver $\mathcal{R}$.
2. $\mathcal{R}$ samples $y \leftarrow \mathcal{R}$ and sends wzkaok_2 to $\mathcal{C}$, together with $\mathsf{OT}_1(y)$.
3. $\mathcal{S}$ samples $x \leftarrow \mathcal{Q}$ and computes $\{\mathsf{GC}, (\mathsf{label}_i)_{i\in[\kappa]}\}(U(x,\cdot))$, where U is the universal function. $\mathcal{S}$ then sends $\mathsf{GC}(U(x,\cdot))$, together with $o = \mathsf{OT}_2(i)$ for $i \in [\kappa]$. Additionally, $\mathcal{S}$ sends wzkaok_3 proving: $\exists(x, r)$ such that $\{\mathsf{GC}, (\mathsf{label}_i)_{i\in[\kappa]}\}(U(x,\cdot))$ and $o = \mathsf{OT}_2(\mathsf{label}_i)$ for $i \in [\kappa]$.
4. If wzkaok verifies, $\mathcal{R}$ obtains labels label_{y_i} for $i \in [\kappa]$ corresponding to his input y, and outputs $z = \mathsf{GC}(\mathsf{label}_y)$. Optionally, if the sender requires the output, $\mathcal{R}$ sends z to $\mathcal{S}$.

Figure 12: Two Party Computation with Distributional Distinguisher-Dependent Security

Proof. (Sketch) By (computational) hiding of OT_1, the probability of abort between the real and simulated views, is at most $\mathsf{negl}(\kappa)$. Otherwise, upon successful extraction of the sender's input from wzkaok_3, Sim sends this input to the ideal functionality and obtains output z, which it (optionally) sends to $\mathcal{S}$. By hiding of OT_1, the real view is indistinguishable from the ideal view. □

Lemma 25. *For every PPT distinguisher $\mathcal{D}$ that obtains the view of the receiver, and every $\epsilon = \frac{1}{\mathsf{poly}(\kappa)}$, there exists a simulator Sim_ϵ where the view $\mathsf{IDEAL}_{\mathcal{F},\mathsf{Sim}_\epsilon}$ in Figure 14 is such that*

$$\Big|\Pr\left[\mathcal{D}\left(\mathsf{IDEAL}_{\mathcal{F},\mathsf{Sim}_\epsilon}(\kappa,x,y)_{\kappa\in\mathbb{N},(x,y)s.t.|x|=|y|}\right)\right]$$
$$-\Pr\left[\mathcal{D}\left(\mathsf{REAL}_{\Pi,\mathcal{R}}(\kappa,x,y)_{\kappa\in\mathbb{N},(x,y)s.t.|x|=|y|}\right)\right]\Big| \leq \epsilon$$

Proof. (Sketch) By distributional simulation security of wzkaok_3, when the weak ZK simulator is executed with error parameter ϵ, the output of the distinguisher remains $\frac{\epsilon}{2}$-close to its output in the real world. Furthermore, for fixed public input of the receiver, the learning strategy of Sim_ϵ is identical to that in Figure 4. Thus, by the analysis in Claim 3 and Claim 4, Sim_ϵ learns approximately the correct input of the receiver, corresponding to distinguisher $\mathcal{D}$. In other words, letting y_i denote the receiver input learned by Sim_ϵ, the distinguisher's output on input $\mathsf{OT}_2(\mathsf{label}_i^{y_i}, \mathsf{label}_i^{y_i})$ for $i \in [\kappa]$ remains $\frac{\epsilon}{2}$-close to the distinguisher output on input the correct set of labels. Given fixed receiver input y_i, by security of garbled circuits, the simulator can indistinguishably replace the garbled circuit with a constant circuit that generates the output of $\mathcal{F}$ on input y_i. In particular, this implies distinguisher-dependent security for $\mathcal{F}_{\mathsf{IIF}}$ functionalities where an honest sender's input can be sampled from an independent public distribution. □

Simulation strategy against Malicious Sender.
In this description, if the adversary $\mathcal{S}$ aborts at any stage in the main thread, Sim outputs $\perp$.

1. The simulator Sim obtains wzkaok_1 from the sender $\mathcal{S}$.
2. Sim samples 0^κ and sends wzkaok_2 to $\mathcal{C}$, together with $\mathsf{OT}_1(0^\kappa)$.
3. Sim then obtains GC, together with OT_2 and wzkaok_3. It aborts if wzkaok_3 doesn't verify.
4. Sim then rewinds and again sends the message in Step 2, with a different value for the challenge wzkaok_2. It obtains $\mathcal{S}$'s response, wzkaok_3. It continues rewinding this way, until it succeeds in extracting the witness for wzkaok for the main thread. If it does not succeed after κ^2 tries, it aborts. The extracted witness includes the input x of $\mathcal{S}$.
5. Sim sends x to the ideal functionality, and obtains the output z, which it (optionally) sends to $\mathcal{S}$, if the protocol demands.

Figure 13: Sender Simulation

Theorem 15. *The protocol in Figure 12 is a secure protocol for two party computation with input-indistinguishable security against a malicious receiver and standard simulation security against a malicious sender.*

Security against malicious senders is already proven in Lemma 25. We prove input-indistinguishable security against malicious receivers in the following lemma.

Lemma 26 (Input-Indistinguishable Security against Malicious Receivers)**.** *The protocol in Figure 12 satsifies input-indistinguishable security against malicious receivers according to Definition 16.*

Proof. (Sketch) We observe that implicit computation follows because the receiver message is statistically binding to the receiver's input. Moreover, independence of receiver input follows because receiver message is sent *before* the sender sends a message depending on his input.

To prove input indistinguishability, we consider a sequence of hybrids, where we gradually move from the real world execution with sender input x_1 to an execution with sender input x_2. We begin by simulating the weak ZK argument of knowledge: then by distributional simulation security (with extended simulation) of wzkaok_3, when the weak ZK simulator is executed with error parameter ϵ starts simulating wzkaok_3, the output of the distinguisher remains $\frac{\epsilon}{2}$-close to its output in the real world. Furthermore, the learning strategy of Sim_ϵ is identical to that in Figure 4. Thus, by the analysis in Claim 3 and Claim 4, Sim_ϵ learns approximately the correct input of the receiver, corresponding to distinguisher $\mathcal{D}$. In other words, letting y^* denote the receiver input learned by Sim_ϵ, the distinguisher's output on input $\mathsf{OT}_2(\mathsf{label}_i^{y_i^*}, \mathsf{label}_i^{y_i^*})$ for $i \in [\kappa]$ remains $\frac{\epsilon}{2}$-close to the distinguisher output on input the correct set of labels. Given fixed receiver input y^*, by security of garbled circuits, the simulator can indistinguishably replace the garbled circuit with a constant circuit that generates the output $f(x_1, y^*)$. Since, $f(x_2, y^*) = f(x_1, y^*)$, the sequence of hybrids described above can be repeated in reverse order such that in the final hybrid, the sender executes the protocol honestly with input x_2. □

References

[1] Aiello, W., Bhatt, S.N., Ostrovsky, R., Rajagopalan, S.: Fast verification of any remote procedure call: Short witness-indistinguishable one-round proofs for NP. In: Automata, Languages and

Simulation strategy against Malicious Receiver.
In this description, if the adversary $\mathcal{R}$ aborts at any stage in the main thread, Sim_ϵ outputs $\perp$.

1. The simulator Sim_ϵ sends wzkaok_1 to the receiver $\mathcal{R}$ (Note that wzkaok_1 is delayed-input, thus wzkaok_1 doesn't require any input).
2. Sim_ϵ then obtains OT_1 from $\mathcal{R}$.
3. Sim_ϵ then samples $x \leftarrow \mathcal{Q}$, and then computes $\{\mathsf{GC}, \mathsf{label}\}(U(x, \cdot))$, where U is the universal function. Sim_ϵ sends $\mathsf{GC}(U(x, \cdot))$, together with $o = \mathsf{OT}_2(\mathsf{label})$. Additionally, Sim_ϵ sends wzkaok_3 proving: $\exists(x, r)$ such that $\{\mathsf{GC}, \mathsf{label}\}(U(x, \cdot))$ and $o = \mathsf{OT}_2(\mathsf{label})$.
4. Sim_ϵ then uses several simulated wzkaok_3 messages with error $\epsilon/2$ (note that these can be simulated using the distribution $\mathcal{Q}$), in order to extract the receiver OT input via the following strategy:
   - Sample $x \leftarrow \mathcal{Q}$.
   - For bit of i the receiver input y, denote the garbled circuit labels by $(\mathsf{label}_i^0, \mathsf{label}_i^1)$. In the same way as in Figure 4, by running in time $\mathsf{poly}(\frac{1}{\epsilon})$, observe whether $\Pr[\mathcal{D} = 1|\mathsf{OT}_2(\mathsf{label}_i^0, \mathsf{label}_i^1)]$ is $\frac{\epsilon}{7}$2-close to $\Pr[\mathcal{D} = 1|\mathsf{OT}_2(\mathsf{label}_i^0, \mathsf{label}_i^0)]$ or $\Pr[\mathcal{D} = 1|\mathsf{OT}_2(\mathsf{label}_i^1, \mathsf{label}_i^1)]$.
     By OT security, for any PPT distinguisher that obtains the receiver's view, one of the two must be close, if the first is close, Sim_ϵ sets $y_i = 0$ otherwise it sets $y_i = 1$. Repeat inductively for indices from 1 to n, setting y_i for $i \in [1, i']$ before fixing $y_{i'+1}$.
   - On learning y, send it to the ideal functionality to obtain output z. Then, construct $\{\mathsf{GC}, \mathsf{label}\}(\mathcal{Z})$ where $\mathcal{Z}$ denotes the constant function that always outputs z. Send $\{\mathsf{GC}, \mathsf{label}\}$ to $\mathcal{R}$ together with the simulated third message wzkaok_3.

Figure 14: Receiver Simulation

Programming, 27th International Colloquium, ICALP 2000, Geneva, Switzerland, July 9-15, 2000, Proceedings. pp. 463–474 (2000)

[2] Badrinarayanan, S., Garg, S., Ishai, Y., Sahai, A., Wadia, A.: Two-message witness indistinguishability and secure computation in the plain model from new assumptions. IACR Cryptology ePrint Archive 2017, 433 (2017), http://eprint.iacr.org/2017/433

[3] Barak, B.: How to go beyond the black-box simulation barrier. In: FOCS. pp. 106–115 (2001)

[4] Barak, B., Goldreich, O., Impagliazzo, R., Rudich, S., Sahai, A., Vadhan, S.P., Yang, K.: On the (im)possibility of obfuscating programs. In: Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19-23, 2001, Proceedings. pp. 1–18 (2001)

[5] Bellare, M., Palacio, A.: The knowledge-of-exponent assumptions and 3-round zero-knowledge protocols. In: Advances in Cryptology - CRYPTO 2004, 24th Annual International Cryptology-Conference, Santa Barbara, California, USA, August 15-19, 2004, Proceedings. pp. 273–289 (2004)

[6] Bellare, M., Stepanovs, I., Tessaro, S.: Contention in cryptoland: Obfuscation, leakage and UCE. In: Theory of Cryptography - 13th International Conference, TCC 2016-A, Tel Aviv, Israel, January 10-13, 2016, Proceedings, Part II. pp. 542–564 (2016)

[7] Bitansky, N., Brakerski, Z., Kalai, Y.T., Paneth, O., Vaikuntanathan, V.: 3-message zero knowledge against human ignorance. In: TCC-B (2016)

[8] Bitansky, N., Canetti, R.: On strong simulation and composable point obfuscation. In: Advances in Cryptology - CRYPTO 2010, 30th Annual Cryptology Conference, Santa Barbara, CA, USA, August 15-19, 2010. Proceedings. pp. 520–537 (2010)

[9] Bitansky, N., Canetti, R., Paneth, O., Rosen, A.: On the existence of extractable one-way functions. In: Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014. pp. 505–514 (2014)

[10] Bitansky, N., Paneth, O.: Point obfuscation and 3-round zero-knowledge. In: Theory of Cryptography - 9th Theory of Cryptography Conference, TCC 2012, Taormina, Sicily, Italy, March 19-21, 2012. Proceedings. pp. 190–208 (2012)

[11] Bitansky, N., Paneth, O.: Zaps and non-interactive witness indistinguishability from indistinguishability obfuscation. In: Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part II. pp. 401–427 (2015)

[12] Bitansky, N., Paneth, O., Wichs, D.: Perfect structure on the edge of chaos - trapdoor permutations from indistinguishability obfuscation. In: Theory of Cryptography - 13th International Conference, TCC 2016-A, Tel Aviv, Israel, January 10-13, 2016, Proceedings, Part I. pp. 474–502 (2016)

[13] Blum, M.: How to prove a theorem so no one else can claim it. In: Proceedings of the International Congress of Mathematicians. pp. 1444–1451 (1987)

[14] Brzuska, C., Mittelbach, A.: Indistinguishability obfuscation versus multi-bit point obfuscation with auxiliary input. In: Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014, Proceedings, Part II. pp. 142–161 (2014)

[15] Canetti, R.: Towards realizing random oracles: Hash functions that hide all partial information. In: Advances in Cryptology - CRYPTO '97, 17th Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 1997, Proceedings. pp. 455–469 (1997)

[16] Canetti, R., Goldreich, O., Goldwasser, S., Micali, S.: Resettable zero-knowledge (extended abstract). In: Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing, May 21-23, 2000, Portland, OR, USA. pp. 235–244 (2000)

[17] Chor, B., Goldreich, O., Kushilevitz, E., Sudan, M.: Private information retrieval. In: 36th Annual Symposium on Foundations of Computer Science, Milwaukee, Wisconsin, 23-25 October 1995. pp. 41–50 (1995)

[18] Chung, K., Kalai, Y.T., Vadhan, S.P.: Improved delegation of computation using fully homomorphic encryption. In: Advances in Cryptology - CRYPTO 2010, 30th Annual Cryptology Conference, Santa Barbara, CA, USA, August 15-19, 2010. Proceedings. pp. 483–501 (2010)

[19] Chung, K., Lui, E., Pass, R.: From weak to strong zero-knowledge and applications. In: Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part I. pp. 66–92 (2015)

[20] Ciampi, M., Ostrovsky, R., Siniscalchi, L., Visconti, I.: Concurrent non-malleable commitments (and more) in 3 rounds. In: Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part III. pp. 270–299 (2016)

[21] Ciampi, M., Persiano, G., Scafuro, A., Siniscalchi, L., Visconti, I.: Improved or-composition of sigma-protocols. In: Theory of Cryptography - 13th International Conference, TCC 2016-A, Tel Aviv, Israel, January 10-13, 2016, Proceedings, Part II. pp. 112–141 (2016)

[22] Ciampi, M., Persiano, G., Scafuro, A., Siniscalchi, L., Visconti, I.: Online/offline OR composition of sigma protocols. In: Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part II. pp. 63–92 (2016)

[23] Cramer, R., Damgård, I., Schoenmakers, B.: Proofs of partial knowledge and simplified design of witness hiding protocols. In: Advances in Cryptology - CRYPTO '94, 14th Annual International Cryptology Conference, Santa Barbara, California, USA, August 21-25, 1994, Proceedings. pp. 174–187 (1994)

[24] Crescenzo, G.D., Persiano, G., Visconti, I.: Constant-round resettable zero knowledge with concurrent soundness in the bare public-key model. In: Advances in Cryptology - CRYPTO 2004, 24th Annual International CryptologyConference, Santa Barbara, California, USA, August 15-19, 2004, Proceedings. pp. 237–253 (2004)

[25] Dolev, D., Dwork, C., Naor, M.: Non-malleable cryptography (extended abstract). In: Proceedings of the 23rd Annual ACM Symposium on Theory of Computing, May 5-8, 1991, New Orleans, Louisiana, USA. pp. 542–552 (1991)

[26] Döttling, N., Fleischhacker, N., Krupp, J., Schröder, D.: Two-message, oblivious evaluation of cryptographic functionalities. In: Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part III. pp. 619–648 (2016)

[27] Dwork, C., Naor, M.: Zaps and their applications. In: 41st Annual Symposium on Foundations of Computer Science, FOCS 2000, 12-14 November 2000, Redondo Beach, California, USA. pp. 283–293 (2000)

[28] Dwork, C., Naor, M., Reingold, O., Stockmeyer, L.J.: Magic functions. In: 40th Annual Symposium on Foundations of Computer Science, FOCS '99, 17-18 October, 1999, New York, NY, USA. pp. 523–534 (1999)

[29] Feige, U., Shamir, A.: Witness indistinguishable and witness hiding protocols. In: Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, May 13-17, 1990, Baltimore, Maryland, USA. pp. 416–426 (1990)

[30] Garg, S., Goyal, V., Jain, A., Sahai, A.: Concurrently secure computation in constant rounds. In: Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings. pp. 99–116 (2012)

[31] Garg, S., Mukherjee, P., Pandey, O., Polychroniadou, A.: The exact round complexity of secure computation. In: Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part II. pp. 448–476 (2016)

[32] Garg, S., Pandey, O., Srinivasan, A., Zhandry, M.: Breaking the sub-exponential barrier in obfustopia. In: CRYPTO (2017)

[33] Gennaro, R., Gentry, C., Parno, B.: Non-interactive verifiable computing: Outsourcing computation to untrusted workers. In: Advances in Cryptology - CRYPTO 2010, 30th Annual Cryptology Conference, Santa Barbara, CA, USA, August 15-19, 2010. Proceedings. pp. 465–482 (2010)

[34] Goldreich, O.: A uniform-complexity treatment of encryption and zero-knowledge. J. Cryptology 6(1), 21–53 (1993)

[35] Goldreich, O.: Foundations of Cryptography: Volume 2, Basic Applications. Cambridge University Press, New York, NY, USA (2004)

[36] Goldreich, O., Krawczyk, H.: On the composition of zero-knowledge proof systems. SIAM J. Comput. 25(1), 169–192 (1996)

[37] Goldreich, O., Micali, S., Wigderson, A.: How to play any mental game. In: STOC (1987)

[38] Goldreich, O., Oren, Y.: Definitions and properties of zero-knowledge proof systems. J. Cryptology 7(1), 1–32 (1994)

[39] Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof-systems. In: STOC. pp. 291–304 (1985)

[40] Goyal, V., Pandey, O., Richelson, S.: Textbook non-malleable commitments. In: Wichs, D., Mansour, Y. (eds.) Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016. pp. 1128–1141. ACM (2016), http://doi.acm.org/10.1145/2897518.2897657

[41] Groth, J., Ostrovsky, R., Sahai, A.: 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. pp. 97–111 (2006)

[42] Hada, S., Tanaka, T.: On the existence of 3-round zero-knowledge protocols. In: Advances in Cryptology - CRYPTO '98, 18th Annual International Cryptology Conference, Santa Barbara, California, USA, August 23-27, 1998, Proceedings. pp. 408–423 (1998)

[43] Haitner, I., Rosen, A., Shaltiel, R.: On the (im)possibility of arthur-merlin witness hiding protocols. In: Theory of Cryptography, 6th Theory of Cryptography Conference, TCC 2009, San Francisco, CA, USA, March 15-17, 2009. Proceedings. pp. 220–237 (2009)

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

[45] Hazay, C., Venkitasubramaniam, M.: On the power of secure two-party computation. In: Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II. pp. 397–429 (2016)

[46] Kalai, Y.T.: Smooth projective hashing and two-message oblivious transfer. In: Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005, Proceedings. pp. 78–95 (2005)

[47] Kalai, Y.T., Raz, R.: Probabilistically checkable arguments. In: Advances in Cryptology - CRYPTO 2009, 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2009. Proceedings. pp. 143–159 (2009)

[48] Katz, J., Ostrovsky, R.: Round-optimal secure two-party computation. In: Advances in Cryptology - CRYPTO 2004, 24th Annual International CryptologyConference, Santa Barbara, California, USA, August 15-19, 2004, Proceedings. pp. 335–354 (2004)

[49] Lapidot, D., Shamir, A.: Publicly verifiable non-interactive zero-knowledge proofs. In: Menezes, A., Vanstone, S.A. (eds.) Advances in Cryptology - CRYPTO '90, 10th Annual International Cryptology Conference, Santa Barbara, California, USA, August 11-15, 1990, Proceedings. Lecture Notes in Computer Science, vol. 537, pp. 353–365. Springer (1990), http://dx.doi.org/10.1007/3-540-38424-3_26

[50] Micali, S., Pass, R., Rosen, A.: Input-indistinguishable computation. In: 2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06). pp. 367–378 (Oct 2006)

[51] Mittelbach, A., Venturi, D.: Fiat-shamir for highly sound protocols is instantiable. In: Security and Cryptography for Networks - 10th International Conference, SCN 2016, Amalfi, Italy, August 31 - September 2, 2016, Proceedings. pp. 198–215 (2016)

[52] Naor, M., Pinkas, B.: Efficient oblivious transfer protocols. In: Proceedings of the Twelfth Annual Symposium on Discrete Algorithms, January 7-9, 2001, Washington, DC, USA. pp. 448–457 (2001)

[53] Ostrovsky, R., Persiano, G., Visconti, I.: On input indistinguishable proof systems. In: Automata, Languages, and Programming - 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part I. pp. 895–906 (2014)

[54] Pass, R.: Simulation in quasi-polynomial time, and its application to protocol composition. In: Advances in Cryptology - EUROCRYPT 2003, International Conference on the Theory and Applications of Cryptographic Techniques, Warsaw, Poland, May 4-8, 2003, Proceedings. pp. 160–176 (2003)

[55] Pass, R.: Limits of provable security from standard assumptions. In: Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011. pp. 109–118 (2011)

[56] Prabhakaran, M., Rosen, A., Sahai, A.: Concurrent zero knowledge with logarithmic round-complexity. In: 43rd Symposium on Foundations of Computer Science (FOCS 2002), 16-19 November 2002, Vancouver, BC, Canada, Proceedings. pp. 366–375 (2002)

[57] Rosen, A.: A note on constant-round zero-knowledge proofs for NP. In: Theory of Cryptography, First Theory of Cryptography Conference, TCC 2004, Cambridge, MA, USA, February 19-21, 2004, Proceedings. pp. 191–202 (2004)

[58] Sahai, A., Waters, B.: How to use indistinguishability obfuscation: deniable encryption, and more. In: Shmoys, D.B. (ed.) Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014. pp. 475–484. ACM (2014), http://doi.acm.org/10.1145/2591796.2591825

[59] Wee, H.: Black-box, round-efficient secure computation via non-malleability amplification. In: 51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23-26, 2010, Las Vegas, Nevada, USA. pp. 531–540 (2010)

[60] Yao, A.C.: How to generate and exchange secrets (extended abstract). In: 27th Annual Symposium on Foundations of Computer Science, Toronto, Canada, 27-29 October 1986. pp. 162–167 (1986)

[61] Yung, M., Zhao, Y.: Generic and practical resettable zero-knowledge in the bare public-key model. In: Advances in Cryptology - EUROCRYPT 2007, 26th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Barcelona, Spain, May 20-24, 2007, Proceedings. pp. 129–147 (2007)