Abstract

Two-message witness indistinguishable protocols were first constructed by Dwork and Naor (FOCS 2000). They have since proven extremely useful in the design of several cryptographic primitives. However, so far no two-message arguments for NP provided statistical privacy against malicious verifiers. In this paper, we construct the first:

- Two-message statistical witness indistinguishable (SWI) arguments for NP.
- Two-message statistical zero-knowledge arguments for NP with super-polynomial simulation (Statistical SPS-ZK).
- Two-message statistical distributional weak zero-knowledge (SwZK) arguments for NP, where the instance is sampled by the prover in the second round.

These protocols are based on quasi-polynomial hardness of two-message oblivious transfer (OT) with game-based security against PPT senders and unbounded receivers, which in turn can be based on quasi-polynomial hardness of DDH or QR or \( N^{th} \) residuosity. We also demonstrate simple applications of these arguments to constructing more secure forms of oblivious transfer.

Along the way, we show that the Kalai and Raz (Crypto 09) transform compressing interactive proofs to two-message arguments can be generalized to compress certain types of interactive arguments. We introduce and construct a new technical tool, which is a variant of extractable two-message statistically hiding commitments, building on the recent work of Khurana and Sahai (FOCS 17). These techniques may be of independent interest.
## Contents

1 Introduction .......................... 1
  1.1 Our Results .......................... 3
  1.2 Other Related Work ................. 4

2 Overview of Techniques .......... 4
  2.1 First Attempt: Compressing the Blum Protocol via OT ............... 5
    2.1.1 Soundness .......................... 5
    2.1.2 Can we Achieve Statistical Privacy Against Malicious Verifiers? 6
  2.2 Compressing Interactive Arguments While Preserving Soundness ....... 7
    2.2.1 Main Tool: Statistically Hiding Extractable Commitments ....... 8
    2.2.2 Modifying Blum to use Statistically Hiding Extractable Commitments 10
    2.2.3 Arguing Soundness of the Compressed Protocol .................. 10
  2.3 Applications to OT .................. 11
  2.4 On the Relationship with Non-Malleability ....................... 13

3 Preliminaries ......................... 13
  3.1 Oblivious Transfer .................. 14
  3.2 Proof Systems ...................... 15

4 Extractable Commitments .......... 17
  4.1 Definitions ......................... 17
  4.2 Protocol ............................ 19

5 Two-Message Arguments with Statistical Privacy .............. 24
  5.1 Modified Blum Protocol ............. 24
  5.2 Compressing Four Message Argument to a Two Message Argument .. 26
  5.3 Proofs of Privacy .................. 29

6 Oblivious Transfer: Stronger Security and Reversal .......... 35
  6.1 Simulation-Secure Two-Message Oblivious Transfer ............. 35
  6.2 Reversing Oblivious Transfer .......... 37

References ......................... 41
1 Introduction

Witness indistinguishable (WI) protocols \cite{FS90} allow a prover to convince a verifier that some statement \( x \) belongs to an NP language \( L \), with the following privacy guarantee: if there are two witnesses \( w_1, w_2 \) that both attest to the fact that \( x \in L \), then a computationally bounded verifier should not be able to distinguish an honest prover using witness \( w_1 \) from an honest prover using witness \( w_2 \). WI is a relaxation of zero-knowledge that has proven to be surprisingly useful. Because WI is a relaxation, unlike zero-knowledge, there are no known lower bounds on the rounds of interaction needed to build WI protocols. Indeed, in an influential work, Dwork and Naor \cite{DN00} introduced WI protocols that only require two messages to be exchanged between prover and verifier, and these were further derandomized to non-interactive protocols by \cite{BOV03}. Due to this extremely low level of interaction, two-message WI protocols have proven to be very useful in the design of several cryptographic primitives. Later, \cite{GOS06, BP15, JKKR17, BGJ+17} achieved two message WI protocols from other assumptions, including bilinear maps, indistinguishability obfuscation, and quasi-polynomial DDH respectively.

**Two-message Statistical WI.** In this work, we revisit this basic question of constructing two-message WI protocols, to ask whether it is possible to upgrade the WI privacy guarantee to hold even against computationally unbounded verifiers. In other words, can we construct statistical WI (SWI) protocols for NP that require only 2 messages to be exchanged? This is the natural analog of one of the earliest questions studied in the context of zero-knowledge protocols: whether statistical zero-knowledge arguments \cite{BCC88} are possible for NP.

Indeed, statistical security is important because it allows for everlasting privacy against malicious verifiers, long after protocols have completed execution. On the other hand, soundness is usually necessary only in an online setting: in order to convince a verifier of a false statement, a cheating prover must find a way to cheat during the execution of the protocol.

The critical bottleneck to achieving two-message statistical WI has been proving soundness: for instance, the Dwork-Naor transformation from a non-interactive zero-knowledge (NIZK) protocol to two-message WI requires the underlying NIZK to be a proof system – that is, for the NIZK to be sound against computationally unbounded cheating provers. Of course, to achieve statistical privacy, we must necessarily sacrifice soundness against unbounded provers. Thus, remarkably, 17 years after the introduction of two-message WI protocols, until our work, there has been no construction of two-message statistical WI arguments. In fact, this question was open even for three-message protocols.

In our first result, we resolve this research question, constructing the first two-message statistical WI argument for NP, based on standard cryptographic hardness assumptions against quasi-polynomial time adversaries. Because two-message WI is so widely applicable, and statistical privacy is useful in many situations where computational privacy does not suffice, we expect our two-message SWI argument to be a useful new tool in the protocol designer’s toolkit.

**Stronger two-message statistically private arguments.** The techniques we use to build two-message SWI also allow us to solve other interesting protocol questions. As our next contribution, we construct arguments allowing super-polynomial statistical simulation in just two messages\footnote{Achieving such two-message arguments was claimed to be impossible in the work of \cite{CLMP12}, however the work of \cite{KS17} showed that the line of impossibility claims \cite{CLMP12} for super-polynomial simulation was incomplete.}. Note that in all these arguments, the simulator works by breaking soundness of the proof, so these are only sound against provers running in time smaller than that of the simulator.

Pass \cite{Pas03} first constructed two-message arguments with super-polynomial simulation, where simulated proofs were indistinguishable by distinguishers running in time significantly smaller than
that of the (uniform) simulator. Very recently, Khurana and Sahai \cite{KS17} constructed the first two-message arguments for NP achieving super-polynomial strong simulation, where the super-polynomial simulator runs in time $T$, and the output is indistinguishable from real executions of the protocol even against distinguishers that run in time $T'$ that is much larger than $T$.

However, an even stronger security property would be super-polynomial statistical simulation, where the output of the simulator is indistinguishable from real executions of the protocol even against distinguishers that run in an unbounded amount of time. In this paper, we construct the first two-message argument for NP achieving this property.

Moreover, recently \cite{JKKR17} recently constructed two-message arguments in the delayed-input distributional setting, with distinguisher-dependent polynomial simulation. However, these protocols also only satisfied computational privacy, and a natural open question was to achieve statistical privacy. We show that our techniques can be used to get two-message arguments for NP with distinguisher-dependent polynomial simulation, in the delayed-input distributional setting \cite{JKKR17}, while still maintaining statistical privacy.

**New Oblivious Transfer protocols.** Our techniques also have applicability to an intriguing question about oblivious transfer (OT): The works of Naor and Pinkas \cite{NP01} and Aiello et al \cite{AIR01} introduced influential two-message protocols for OT achieving a game-based notion of security, which offered security against computationally unbounded malicious receivers. A natural question is: Can we achieve a similar result offering security against computationally unbounded senders? Note that to achieve such a result, at least three messages must be exchanged in the OT protocol: indeed, suppose to the contrary that there was a two-message OT protocol with security against an unbounded sender. Then the first message of the protocol sent by the receiver must statistically hide the choice bit of the receiver in order for this message to provide security against an unbounded cheating sender. However, a non-uniform cheating receiver could begin the protocol with non-uniform advice consisting of a valid first message $m$ together with honest receiver randomness $r_0$ that explains $m$ with regard to the choice bit $b = 0$, and honest receiver randomness $r_1$ that explains $m$ with regard to choice bit $b = 1$. Now this receiver would be able to recover both inputs of the honest sender by using both random values $r_1, r_2$ in turn on the sender’s response message, violating OT security against a (bounded) malicious receiver.

Again remarkably, this basic question – constructing a 3-message OT protocol with security against unbounded sender – has been open since the works of \cite{NP01,AIR01} 16 years ago. We resolve this question, by exhibiting such a 3-message OT protocol, based on standard cryptographic hardness assumptions with security against quasi-polynomial time adversaries. Such an OT protocol can also be plugged into the constructions of \cite{JKKR17} to achieve proofs for NP (as opposed to arguments) achieving delayed-input distributional weak ZK, witness hiding and strong indistinguishability.

Our techniques even apply to other well-studied questions about OT, even in the two-message setting with security against unbounded receivers: It has long been known that the 2-message OT protocols of \cite{NP01,AIR01} do not rule out selective failure attacks; for instance if two OTs are run in parallel, we do not know how to rule out the possibility that the sender can cause the OTs to abort if and only if the receiver’s two choice bits are equal. Intuitively, this should not be possible in a secure OT, and the “gold standard” for preventing all such attacks for OT is to prove security via simulation. For 2-message OT protocols, however, only super-polynomial simulation is possible, and this was recently formally established by \cite{BGI+17} but at the cost of sacrificing security against unbounded receivers. This sacrifice seems inherent: if an OT protocol has a super-polynomial simulator, then it seems that an unbounded malicious receiver can just “run the simulator” to extract the inputs of the sender. This presents a conundrum; perhaps simulation
security and security against an unbounded malicious receiver are like oil and water and cannot mix.

In fact, we show that it is possible to construct a two-message OT protocol with both super-polynomial simulation security, and security against unbounded receivers.

**Our core technique.** Indeed, echoes of the dilemma we outlined above permeate all the problems that we address in this work. For this reason, we build a new core technique that is applicable to all the problems we posed above, from two-message statistical WI to two-message OT with both SPS security and security against unbounded receivers, and all the others in between. Our core technical challenge is to be able to push the privacy guarantees within the protocols of [KS17] all the way to statistical privacy.

To achieve this, we start with interactive arguments satisfying honest verifier statistical zero-knowledge, and compress them to two-message arguments by proving soundness of the [BMW98, KR09] heuristic. We note that the [BMW98, KR09] is known to be insecure when applied generally to interactive arguments (as opposed to proofs). Namely, there exists a (contrived) interactive argument such that reducing rounds via the [BMW98] -heuristic results in an insecure 2-message argument. Nevertheless, we construct a family of 4-message interactive arguments with statistical hiding guarantees, and prove that the [BMW98] -heuristic is sound when applied to such protocols.

At the heart of our technique is the following idea: We devise protocols that are almost always statistically private, but with negligible probability, they are statistically binding. Crucially, we show that a (computationally bounded) sender cannot distinguish between the case when the protocol ends up being statistically private (which happens most of the time), and the case when the protocol ends up being statistically binding (which happens very rarely). We then show how to leverage this rare statistical binding event to make our protocol behave like a “proof” instead of an argument, allowing the [BMW98, KR09] heuristic to go through. More generally, this rare event helps us achieve other extraction properties that we want in our applications. We elaborate on this intuition below in our technical overview, providing a detailed but still informal overview of our techniques and results. We stress that all the security properties that we establish in this paper can be based on standard cryptographic hardness assumptions with security against quasi-polynomial time adversaries.

### 1.1 Our Results

We construct several protocols with security properties assuming the existence of a quasi-poly secure OT, which can in turn be instantiated based on quasi-poly hardness of the DDH assumption [NP01], or based on the quasi-poly hardness of QR or the N’th residuosity assumption [Kal05, HK12a]. In particular, we first construct a 2-message argument for NP with statistical hiding guarantees, and we prove that our resulting 2-message protocol has the following statistical hiding guarantees:

1. Our 2-message argument is statistical witness indistinguishable. We note that prior to this work, we did not even know how to construct a 3-message statistical WI scheme.

2. Our 2-message argument is statistical zero-knowledge with super-polynomial time simulation.

3. Our 2-message argument is statistical weak zero-knowledge in the delayed input setting, where the instance is sampled from some distribution after the verifier sent the first message.

We also obtain the following results on oblivious transfer:

---

2 We note that prior to this work, this was believed to be impossible to achieve via black-box reductions [CLMPT12].
1. We construct a three-message OT protocol simultaneously satisfying super-polynomial simulation security, and security against a computationally unbounded sender.

2. We construct a two-message OT protocol simultaneously satisfying super-polynomial simulation security, and security against a computationally unbounded receiver.

1.2 Other Related Work

Two message statistical witness indistinguishable arguments were constructed for specific languages admitting hash proof systems, by [GOVW12]. However, no two-message statistical WI arguments were known for all of NP.

Our key technique consists of compressing an interactive protocol into a two-message protocol. In this context, two main approaches for reducing rounds in interactive protocols appeared in the literature. The first is due to Fiat and Shamir [FS86], and the second is due to [BMW98 ABOR00]. Kalai and Raz [KR09] proved that the [BMW98]-heuristic is sound when applied to a statistically sound interactive proof, assuming the existence of a super-polynomial OT (or computational PIR) scheme. Very recently, [KRR17] showed that the Fiat-Shamir heuristic is also sound when applied to to a statistically sound interactive proof, assuming the existence of a sub-exponentially secure indistinguishability obfuscation and assuming the existence of exponentially secure multi-bit point function obfuscation.

The works of [JJKR17 BGI+17] are closely related to our work. They assume the existence of a quasi-poly secure oblivious transfer (OT) scheme, and show how to convert any 3-message public-coin protocol which is honest-verifier zero-knowledge, into a 2-message protocol, while keeping (and even improving) the secrecy guarantees. However, their work does not yield statistical privacy, which is the focus of the present work. These works apply the the [BMW98]-heuristic to 3-message public-coin honest-verifier zero-knowledge proofs, to obtain their resulting 2-message protocols. We note that since they start with a statistical sound proof they obtain only computational hiding guarantees, and after applying the [BMW98]-heuristic, their resulting 2-message protocols are only computationally sound (in addition to being only computational hiding).

In contrast, in this work we construct 2-message arguments with statistical hiding guarantees. Moreover, we do this by constructing a 4-message interactive argument with statistical hiding guarantees, and converting it into a 2-message computationally sound protocol by applying the [BMW98]-heuristic to it.

2 Overview of Techniques

Our starting point is an approach initially explored by [BMW98 ABOR00 KR09], to compress public coin interactive proofs into two-message arguments. Recently, [JJKR17 BGI+17] proved that a variant of this compression also guarantees strong forms of privacy against malicious verifiers. We begin by reviewing these techniques with respect to a specific protocol from [JJKR17 BGI+17] in the following section.

2.1 First Attempt: Compressing the Blum Protocol via OT

The preliminary protocol makes use of two components:

- A three-message proof for Graph Hamiltonicity, due to Blum [Blu86]. Denote its three messages by \((a, e, z)\), which can be parsed as \(a = \{a_i\}_{i \in [\kappa]}\), \(e = \{e_i\}_{i \in [\kappa]}\) and \(z = \{z_i\}_{i \in [\kappa]}\). Here for each \(i \in [\kappa]\), the triplet \((a_i, e_i, z_i)\) are messages corresponding to an underlying Blum
any two-message oblivious transfer protocol, denoted by \((\text{OT}_1, \text{OT}_2)\), which is secure against malicious PPT receivers, and malicious senders running in time at most \(2^{\ell_1}\). For receiver input \(b\) and sender input messages \((M^0, M^1)\), we denote the two messages of the OT protocol as \(\text{OT}_1(b)\) and \(\text{OT}_2(M^0, M^1)\). We note that \(\text{OT}_2(M^0, M^1)\) also depends on the message \(\text{OT}_1(b)\) sent by the receiver. For the sake of simplicity, we omit this dependence from the notation.

Given these components, the two-message protocol \((P, V)\) is described in Figure 1.

**Preliminary Two-Message Protocol from [JKKR17, BGI+17]**

- For \(i \in [\kappa]\), \(V\) picks \(e_i \leftarrow \{0, 1\}\), and sends \(\text{OT}_{1,i}(e_i)\) 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 \(a_i, \text{OT}_{2,i}(z_i^{(0)}, z_i^{(1)})\) 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 \(\Sigma\)-protocol.

Figure 1: Preliminary two-message protocol

### 2.1.1 Soundness

It was proven in [KR09, JKKR17, BGI+17] that such a transformation from any public-coin interactive proof to a two-round argument preserves soundness against adaptive PPT provers, who may choose the instance adaptively depending upon the message sent by the verifier.

**Background.** Recall that in the original protocol of Blum, for each index \(i \in [\kappa]\), \(a_i\) consists of a statistically binding commitment to a random permutation \(\pi\) and \(\pi(G)\), where \(G\) denotes the input instance with Hamiltonian cycle \(H\). Then, if the verifier challenge \(e_i = 0\), the prover computes \(z_i\) as a decommitment \(\pi, \pi(G)\) and the verifier accepts if and only if the graph \(G\) was correctly permuted. On the other hand, if \(e_i = 1\), the prover computes \(z_i\) as a decommitment only to the edges of the Hamiltonian Cycle \(\pi(H)\) in \(\pi(G)\), and the verifier accepts if the revealed edges are indeed a Hamiltonian Cycle.

To prove soundness, [KR09, JKKR17, BGI+17] rely on the following special-soundness property of the Blum protocol: 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 \((a_i, e_i, z_i), (a_i, e_i', z_i')\) 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 \(e_i\), for which there exists \(z_i\) such that \((a_i, e_i, z_i)\) is an accepting transcript (as otherwise the algorithm \(A\) would output a witness \(w \in R_L(x)\), which is impossible).

**Analysis of the Protocol.** Going back to the protocol in Figure 1—suppose a cheating prover, on input the verifier message \(\text{OT}_1(e_i^*)\), outputs \(x^* \notin L\), together with messages \(a_i^*, \text{OT}_2(z_i^*)\), such
that the verifier accepts with probability $\frac{1}{2} + \negl(\kappa)$. Since, for any $x^* \notin L$ and any $a^*_i$, there exists at most one unique value of receiver challenge $e^*_i$, for which there exists a $z^*_i$ that causes the verifier to accept – intuitively, this means that $a^*_i$ consists of a commitment that encodes the receiver challenge $e^*_i$. More specifically, if $x^* \notin L$, then $a^*_i$ can either only consist of a commitment to $\pi$ and a correctly permuted graph $\pi(G)$ (corresponding to $e_i = 0$), or a commitment to some graph (that is not a permutation of $G$) but one that contains a hamiltonian cycle.

Thus, a reduction that can break (via brute-force) the commitment string $a^*_i$, can check if the committed value corresponds to a correctly permuted $\pi(G)$. If it does, the reduction guesses $e_i = 0$, otherwise it guesses $e_i = 1$. At the same time, we will jack up security of the OT to guarantee that no reduction running in time large enough to break the commitment scheme, should be able to guess the receiver challenge $e$ with probability greater than $\frac{1}{2} + \negl(\kappa)$. This would give a contradiction and prove that a single parallel execution of the protocol in Figure 1 has soundness $\frac{1}{2} + \negl(\kappa)$.

The same argument generalizes to prove that no adaptive PPT prover $P^*$ can cheat with non-negligible probability when we perform $\kappa$ parallel repetitions. More specifically, the reduction can use any prover that cheats with non-negligible probability to guess the $\kappa$-bit challenge $e$ with non-negligible probability, contradicting the security of $\kappa$ parallel repetitions of OT.

We now move to analyzing privacy of the protocol in Figure 1.

### 2.1.2 Can we Achieve Statistical Privacy Against Malicious Verifiers?

The work of [JKKR17, BGI+17] showed that the protocol in Figure 1 satisfies computational witness indistinguishability, as well as other stronger (computational) privacy guarantees against malicious verifiers. Their proofs rely on the security of OT against malicious receivers, as well as the honest verifier zero-knowledge property of the underlying Blum proof.

As we already described, the focus of this paper is achieving statistical privacy. To this end, we begin with the following observations about the protocol in Figure 1:

- The underlying Blum proof only satisfies computational honest verifier zero-knowledge. This is because it uses a statistically binding, computationally hiding commitment to generate the first message $\{a_i\}_{i \in [\kappa]}$. An unbounded malicious verifier that breaks the commitment in $\{a_i\}_{i \in [\kappa]}$ can in fact, extract $\pi$, and therefore obtain the witness (i.e., the Hamiltonian cycle) from any honest prover.

- The underlying OT protocols [NP01, HK12a] used in the protocol of Figure 1 are already statistically private against malicious receivers. This implies that the messages $\{z_i^{1-e_i}\}_{i \in [\kappa]}$ are statistically hidden from any malicious verifier.

As a result of (1) above, the protocol in Figure 1 is also only computationally private. At this point, it is clear that the main bottleneck towards achieving statistical privacy against malicious verifiers, is the computationally hiding commitment in the message $\{a_i\}_{i \in [\kappa]}$. A natural first idea is then to replace this commitment with a statistically hiding commitment.

That is, we consider a modified version of the underlying Blum protocol, which is the same as the original Blum protocol, except that it uses a statistically hiding, computationally binding commitment. Such a commitment must contain two-messages in order to satisfy binding against non-uniform PPT provers. Therefore, our modified version of Blum has four messages, where in the first message, for $i \in [\kappa]$, the verifier sends the first message $q_i$ of a statistically hiding, computationally binding commitment. Next, the prover responds with $a_i$ consisting of the committer message in response to $q_i$, committing to values $\pi_i, \pi_i(G)$. The next messages $\{e_i\}_{i \in [\kappa]}$ and $\{z_i\}_{i \in [\kappa]}$...
remain the same as before. It is not hard to see that the resulting four-message modified Blum protocol satisfies statistical honest verifier zero-knowledge.

Let us again compress this four-message protocol using the same strategy as before, via two-message OT. That is, the verifier sends in parallel \( \{ q_i, OT_{1,i}(e_i) \}_{i \in [k]} \), and the prover responds with \( \{ a_i, OT_{2,i}(z_{i}^{(0)}, z_{i}^{(1)}) \}_{i \in [k]} \). In this case, because of statistical hiding of the commitments and statistical sender security of OT, the proof in \[ JKKR17, BGI^+17 \] can be easily extended to achieve statistical witness indistinguishability.

One may now hope that the analysis in \[ KR09 \] can be used to prove that the resulting protocol also remains sound against PPT provers. Unfortunately, as we saw above, the proof of soundness \[ KR09 \] crucially relies on the fact that for any \( x \notin L \) and any \( a_i \), there exists a single bit \( e_i \) for which \( a_i \) can be opened in an acceptable way: therefore, \( a_i \) must “encode” \( e_i \). In other words, for this proof to work, we crucially require that the four-message protocol be sound against malicious unbounded provers to begin with.

This is no longer true for the four-message modified version of the Blum protocol described above, because an unbounded prover will be able to equivocate the openings of the computationally binding commitments. In fact, the problem runs deeper: note that what we seem to require is a proof that satisfies statistical honest-verifier ZK, however, such proofs are unlikely to exist for all of NP (see, e.g. \[ SV03 \]). Therefore, the only remaining option, if we follows this approach, is to find a way to compress some form of statistical honest verifier ZK argument while preserving soundness.

### 2.2 Compressing Interactive Arguments While Preserving Soundness

The problem of compressing general interactive arguments while preserving soundness has been a question of broader interest, especially in the context of delegating computation. In this paper, unlike the setting of delegation, we are not concerned with the succinctness of our arguments. Yet, there are no previously known approaches to compressing any types of interactive argument systems that are not also proofs.

In this paper, we develop one such approach. Our high-level idea is as follows: since we already ruled out constructing a proof that satisfies statistical honest-verifier ZK, we will instead construct an argument that satisfies statistical honest-verifier ZK. But this argument will have the property that with a small probability, it will in fact be a proof! Furthermore, no cheating prover will be able to differentiate the case when it is an argument from the case when it is a proof. In other words, we will ensure that any cheating prover that outputs \( x^* \notin L \) together with an accepting proof with non-negligible probability in the original protocol, will continue to do so with non-negligible probability even when it is in proof mode. Upon switching to proof mode, we can apply the techniques of \[ KR09 \] to argue soundness and obtain a contradiction.

Our main technical tool that will help us realize the above outline will be a two-message statistically-hiding extractable commitment scheme, which we now describe.

#### 2.2.1 Main Tool: Statistically Hiding Extractable Commitments

Our construction of statistically hiding, extractable commitments is obtained by building on the recent work of Khurana and Sahai \[ KS17 \].

They showed how to construct extractable computationally hiding commitments: which were completely insecure against unbounded malicious receivers. However, their constructions satisfied hiding against \( T \)-time receivers that interacted with any malicious committer, in (unbounded) polynomially many executions in the real world.
The “extractability” requirement was, additionally, that there exist a $T'$ (less than $T$) time extractor. This extractor may interact with the committer up to $T'$ times, and obtain multiple transcripts: but it should eventually be able to output one transcript, together with the message committed in such a transcript. Moreover, the distribution of transcripts output by such an extractor was required to be indistinguishable (from the committer’s point of view), from the uniform distribution over all transcripts output by the committer.

Their overall idea was to run two-message secure computation (with indistinguishability-based security guarantees) between the committer and the receiver, to ensure that with a negligible probability $2^{-m}$ for $m = \Omega(\log \kappa)$, the receiver just obtains the message being committed, in the clear. On the other hand, with overwhelming probability $1 - 2^{-m}$, the receiver obtains an actual ($T$-hiding) statistically binding commitment as the output of the two-party computation. The extractor of [KS17] worked by executing the committer roughly $2^m$ times playing the role of an honest receiver, waiting to obtain a transcript where it obtained the committed message in the clear. Moreover, the protocol was designed so that the choice of whether the message is revealed to the receiver or not (and therefore whether a transcript allowed for extraction or not), was made by a coin-flip. The output of this coin-flip would be hidden from any malicious committer. Naturally, the protocol of [KS17] was not statistically hiding: not only because of its reliance on only a computationally hiding commitment, but also because of other components of the two-message two-party computation protocol including computationally secure two-party computation.

In this paper, we push hiding security of these commitments against malicious receivers, all the way to statistical. The first question in this situation is: since the commitment is required to be statistically hiding and the committed message is information theoretically lost, what does it mean to “extract” the committed message? We observe that following the definition of [KS17], it suffices to consider an extractor that only extracts from some “special” transcripts, that occur with negligible probability $2^{-m}$. These transcripts will be statistically binding, and therefore there will exist the notion of a committed message in these specific transcripts. On the other hand, most transcripts (1 - $2^{-m}$ fraction of them) will be statistically hiding and the message will be completely lost.

**Basic Construction.** In order to obtain a construction that satisfies these properties, we go back to the drawing board. Instead of implementing a sophisticated two-party computation using garbled circuits, we consider the following basic commitment scheme (Figure 2) implemented using game-based oblivious transfer [NP01, AIR01, Kal05, HK12a], with statistical sender security. We make the following observations about this protocol:

- Assuming statistical sender security of OT, this scheme is 1/2-hiding against malicious receivers (i.e., $r \neq ch$ happens with probability $\frac{1}{2}$, and in this case the message is statistically hidden from any malicious receiver).

- Assuming computational receiver security of OT, this scheme is computationally binding. That is, no malicious PPT committer, upon generating a commitment transcript, can successfully decommit it to two different values $\tilde{M}_1 \neq \tilde{M}_2$, except with negligible probability. This is because given such a committer, the reduction can use this committer to deduce that $r \neq ch$, which should be impossible except with negligible probability.

A formal analysis of this property can be found in Section 4.2, Lemma 2.

---

3We note that this is different from guessing $ch$, which can be done with probability $\frac{1}{2}$: however, a cheating committer can not only guess $ch$ but also certify via two valid decommitments to different messages that it guessed $ch$ correctly, which is not allowed except with negligible probability.
Committer Input: Message $M \in \{0,1\}^p$, where $p = \text{poly}(\kappa)$.

Commit Stage:
Receiver Message.
- Pick challenge string $\text{ch} \leftarrow \{0,1\}$.
- Compute and send the first OT message $\text{OT}_1(\text{ch}, r_1)$ using uniform randomness $r_1$.

Committer Message.
- Sample a random string $r \leftarrow \{0,1\}$. Set $M^r = M$, $M^{1-r} \leftarrow \{0,1\}^p$.
- Compute $o_2 = \text{OT}_2(M^0, M^1; r_2)$ with uniform randomness $r_2$.
- Send $(r, o_2)$.

Reveal Stage: The committer reveals $M$, and both values $(M^0, M^1)$ as well as the randomness $r_2$. The receiver accepts the decommitment to message $M$ if and only if:
1. $o_2 = \text{OT}_2(M^0, M^1; r_2)$,
2. $M^r = M$.

Figure 2: Basic Construction of a Two-Message Statistically Hiding Commitment

Our Construction. Recall that we would like a scheme where most transcripts ($1 - 2^{-m}$ fraction of them) should be statistically hiding and the message should be completely lost. Moreover, we would like a $2^{-m}$ fraction of transcripts to be statistically binding; in fact, it will suffice to directly reveal the message being committed in these transcripts to the receiver. Starting with the basic construction above, a natural way to achieve this is to commit to an XOR secret sharing of the message $M$ via $m$ parallel executions of the basic scheme described above. Formally, our construction is described in Figure 3. This scheme satisfies the following properties:
- It remains computationally binding against malicious PPT committers, just like the basic scheme.
- Because the underlying OT is statistically hiding, our scheme is now $(1 - 2^{-m})$-statistically hiding against malicious receivers (i.e., it is not statistically hiding only in the case that $r \neq \text{ch}$, which happens with probability $2^{-m}$).
- Most importantly, because of receiver security of the OT, no malicious PPT committer can distinguish the case where $r = \text{ch}$ from the case where $r \neq \text{ch}$.

2.2.2 Modifying Blum to use Statistically Hiding Extractable Commitments

Now, instead of plugging in any statistically hiding commitment scheme, we plug in the extractable statistically hiding commitment scheme of Figure 3 to generate messages $\{q_i, a_i\}_{i \in [\kappa]}$, with $m = \Omega(\log \kappa)$. This is formally described in Section 5.1. By statistical hiding of the commitment,

---

This requires a more delicate argument, as well as reliance on $2^m$-security of the OT to ensure that a PPT cheating committer cannot bias $r$ away from $\text{ch}$ all the time.
Extraction parameter: $m$.
Committer Input: Message $M \in \{0, 1\}^p$.

Commit Stage:
Receiver Message.
- Pick challenge string $ch \leftarrow \{0, 1\}^m$.
- Sample uniform randomness $\{r_{1,i}\}_{i \in [m]}$.
- Compute and send $\{\text{OT}_1(ch_i, r_{1,i})\}_{i \in [m]}$ using $m$ instances of two-message OT.

Committer Message.
- Sample a random string $r \leftarrow \{0, 1\}^m$.
- For every $i \in [m]$ and every $b \in \{0, 1\}$, sample $M_i^b \leftarrow \{0, 1\}^p$ subject to $\bigoplus_{i \in [m]} M_i^b = M$.
- For every $i \in [m]$ compute $o_{2,i} = \text{OT}_2(M_i^0, M_i^1, r_{2,i})$ with uniform randomness $r_{2,i}$.
- Send $(r, \{o_{2,i}\}_{i \in [m]})$.

Reveal Stage: The committer reveals $M$, and all values $\{M_i^0, M_i^1\}_{i \in [m]}$ as well as the randomness $r_{2,i}$. The receiver accepts the decommitment to message $M$ if and only if:

1. For all $i \in [m]$, $o_{2,i} = \text{OT}_2(M_i^0, M_i^1, r_{2,i})$,
2. $\bigoplus_{i \in [m]} M_i^b = M$.

---

Figure 3: Our Extractable Commitments

the resulting protocol is a statistical honest verifier ZK argument. On the other hand, by the
eextractability of the commitment, (more specifically in the case where $r = ch$), the protocol, in
fact, becomes a proof. Furthermore, no cheating PPT prover can distinguish the case when $r = ch$
from when $r \neq ch$. Looking ahead, like we already alluded to at the beginning of the overview, we
will compress this while simultaneously ensuring that any malicious prover outputting an accepting
transcript corresponding to $x \not\in L$ with noticeable probability when $r \neq ch$, must continue to do so
even when $r \neq ch$. We will now analyze the soundness of the resulting protocol.

2.2.3 Arguing Soundness of the Compressed Protocol

We show that the resulting protocol remains sound against cheating PPT provers. While we also
achieve a variant of adaptive soundness, for the purposes of this overview we restrict ourselves to
proving soundness against non-adaptive provers that output the instance $x$ before the start of the
protocol.

At a high level, we will begin by noting that a cheating prover that first outputs $x \not\in L$ together
with an accepting proof with probability $p = \frac{1}{\text{poly}(\kappa)}$, cannot distinguish the case when $r = ch$ from
the case when $r \neq ch$ by the property of the extractable commitment. Moreover, such a prover
must continue to generate accepting transcripts for $x \not\in L$ with probability at least $\frac{1}{\text{poly}(\kappa)}$ even in
case $r = ch$. Although the event $r = ch$ only occurs with negligible probability, we use the extractor

\footnote{Ensuring this requires the decommit phase of the extractable commitment to be publicly verifiable, without the
of extcom to amplify this probability by making many queries to the prover. The extractor then outputs a transcript of the proof (corresponding to \( r = ch \)), together with the values committed in all messages corresponding to the extractable commitment. This requires the oblivious transfer used for such compression to be hard against adversaries running in time large enough to enable extraction from the extcom. Additional details of our construction can be found in Section 5.2.

In fact, we notice that our technique is more generally applicable. In particular, we focus on applications to some natural questions about oblivious transfer.

### 2.3 Applications to OT

**OT Secure against Unbounded Senders** While we have long known two-message OT protocols with game-based security against unbounded malicious receivers and PPT malicious senders [NP01, AIR01, Kal05, HK12a], the following natural, extremely related question has remained unanswered so far. Can we construct three-message oblivious transfer with game-based security against unbounded malicious senders and non-uniform PPT malicious receivers?

It is clear that a minimum of three rounds is required for this task, since in any two-message protocol in the plain model secure against non-uniform receivers, the first message must unconditionally bind a malicious receiver to a single choice bit (as otherwise a cheating receiver may obtain non-uniformly, a receiver message as well as randomness that allows opening this message to two different bits). In order to achieve such oblivious transfer, we explore a very natural approach: [WW06] suggested the following way to information-theoretically reverse any ideal OT protocol (with receiver message denoted by \( OT_R \) and sender message denoted \( OT_S \)), by adding single round (Refer to Figure 4).

<table>
<thead>
<tr>
<th>Sender Input: Message bits ( x_0, x_1 ). Receiver Input: Choice bit ( b ).</th>
</tr>
</thead>
<tbody>
<tr>
<td>◦ <strong>Sender Message.</strong> Sample ( x'_0, x'_1 \leftarrow {0, 1}^2 ) and ( r_S ) uniformly at random. Set ( c = x'_0 \oplus x'_1 ), and send ( m_S = OT_R(c; r_S) ).</td>
</tr>
<tr>
<td>◦ <strong>Receiver Message.</strong></td>
</tr>
<tr>
<td>‒ Sample input (single-bit) messages ( m_0, m_1 ) uniformly at random such that ( m_0 \oplus m_1 = b ).</td>
</tr>
<tr>
<td>‒ Send ( m_R = OT_S(m_0, m_1; r_R) ).</td>
</tr>
<tr>
<td>◦ <strong>Sender Message.</strong></td>
</tr>
<tr>
<td>‒ Obtain output ( a ) of the two-message OT using ( (m_R, r_S) ).</td>
</tr>
<tr>
<td>‒ Send ( z = a \oplus x'_0, z_0 = x'_0 \oplus x_0, z_1 = x'_1 \oplus x_1 ).</td>
</tr>
<tr>
<td>◦ <strong>Receiver Output:</strong> The receiver outputs ( y = (z \oplus z_b \oplus m_0) ).</td>
</tr>
</tbody>
</table>

**Figure 4: Oblivious Transfer Reversal**

If we did manage to somehow reverse the two-message OT protocols of [NP01, AIR01, Kal05, HK12a] using such a reversal, then clearly we would obtain a three-message protocol with game-receiver needing to maintain any state from the commit phase. This is for technical reasons, specifically, public verifiability of the decommit phase is required to check whether a transcript is accepting or rejecting even while obtaining the receiver message for the extractable commitment, externally.
based security against unbounded senders and malicious PPT receivers. However, surprisingly, proving game-based security of the protocol obtained by reversing [NP01 AIR01 Kal05 HK12a] appears highly non-trivial, and in fact it is not clear if such security can be proven at all. More specifically, the security reduction against a malicious receiver for the resulting 3 round protocol must make use of a cheating receiver to contradict an assumption. To do this, it must obtain the sender’s first message externally, but since the reduction no longer knows the randomness used for computing this message, it is unclear how such a reduction would be able to complete the third message of the protocol in Figure 4. Indeed, this problem occurs because the original OT lacks any form of simulation security against malicious senders.

Our solution is to strengthen security of the underlying OT in order to make this transformation go through. As we already noted, this also turns out to be related to the problem of preventing selective failure attacks in 2-message OT.

We construct a two-message simulatable variant of oblivious transfer, with security against unbounded receivers, as well as (super-polynomial) simulation security against both malicious senders and malicious receivers.

Given such a protocol, the security reduction described above is able to use the underlying simulator to extract the inputs of the adversary, in order to complete the three-message OT reversal described in Figure 4.

Simulation-Secure Two-Message Oblivious Transfer The first question is, whether it is even possible to obtain two-message oblivious transfer, with unbounded simulation security against malicious senders as well as malicious receivers, while preserving security against unbounded malicious receivers. We will achieve this by bootstrapping known protocols that already satisfy super-polynomial simulation security against malicious receivers, to also add simulation security against malicious senders.

At first, such a definition may appear self-contradictory: if there exists a black-box simulator against that is able to extract both inputs of the malicious sender, then in a two-message protocol, an unbounded receiver may also be able to learn both inputs of the sender by running such a simulator – thereby blatantly violating sender security.

Our key differentiation between the simulator and a malicious receiver, that will block the above intuition from going through, will again be that the simulator can access the sender super-polynomially many times, while an unbounded malicious receiver will only be able to participate in (unbounded, but) polynomially many interactions with the sender.

That is, our protocol will be designed such that, with a small probability $2^{-m}$, the sender will be forced to reveal both his inputs to the receiver. On the other hand, with probability $1 - 2^{-m}$, the sender message that does not correspond to the receiver’s choice bit, will remain statistically hidden. And again, most importantly, a malicious sender will not be able to distinguish between the case where he was forced to reveal both inputs, and the case where he was not.

As a result, the simulator against a malicious sender will run approximately $2^m$ executions with the malicious senders, waiting for an event where the sender is forced to reveal both inputs: and it will just use this execution to output the sender view. We will show, just like the case of statistically hiding extractable commitments, that a cheating sender will not be able to distinguish such views from views that did not allow extraction. Finally, when $m = \Omega(\log n)$, the resulting protocol will still satisfy statistical security against unbounded receivers, while simultaneously

---

6We note that existing two-message protocols [NP01 AIR01 Kal05 HK12a] with security against unbounded receivers do not satisfy simulation-based security against malicious senders.

7This will be achieved by having the sender send a statistically private argument described in the previous section, proving that he computed the message correctly. Such an argument will also enable extraction of the witness with probability $2^{-m}$.
allowing approximately $2^m$-time simulation. Please refer to Section 6 for formal details of our techniques.

2.4 On the Relationship with Non-Malleability

Another way to interpret some of our results is via the lens of non-malleability: in any two-message protocol between Alice and Bob, where Alice sends the first message and Bob sends the second, we show how to enforce that the input used by Bob to generate his message remain independent of the input used by Alice.

One way to accomplish such a task is to set parameters so that the security of Bob’s message is much weaker than that of Alice, in a way that it is possible to break security of Bob’s message via brute-force, and extract Bob’s input in time $T$, while arguing that Alice’s input remained computationally hidden, even against $T$-time adversaries. However, this would crucially require Bob’s message to only be computationally hidden, so that it would actually be recoverable via brute-force. This was used in several works, including [Pas03] which gave the first constructions of computational zero-knowledge with superpolynomial time simulation.

In this paper, building on the recent work of [KS17], we essentially prove that it is possible to achieve similar guarantees while keeping Bob’s message statistically hidden. Indeed, this is the main reason that our proofs of soundness go through.

3 Preliminaries

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

The statistical distance between two distributions $D_1, D_2$ is denoted by $\Delta(D_1, D_2)$ and defined as:

$$\Delta(D_1, D_2) = \frac{1}{2} \sum_{v \in V} |\Pr_{x \leftarrow D_1}[x = v] - \Pr_{x \leftarrow D_2}[x = v]|.$$

We say that two families of distributions $D_1 = \{D_{1,\kappa}\}, D_2 = \{D_{2,\kappa}\}$ are statistically indistinguishable if $\Delta(D_{1,\kappa}, D_{2,\kappa}) = \text{negl}(\kappa)$. We say that two families of distributions $D_1 = \{D_{1,\kappa}\}, D_2 = \{D_{2,\kappa}\}$ are computationally indistinguishable if for all non-uniform probabilistic polynomial time distinguishers $D$,

$$|\Pr_{r \leftarrow D_{1,\kappa}}[D(r) = 1] - \Pr_{r \leftarrow D_{2,\kappa}}[D(r) = 1]| = \text{negl}(\kappa).$$

Let $\Pi$ denote an execution of a protocol. We use $\text{View}_A(\Pi)$ denote the view, including the randomness and state of party $A$ in an execution $\Pi$. We also use $\text{Output}_A(\Pi)$ denote the output of party $A$ in an execution of $\Pi$.

Remark 1. In what follows we define several 2-party protocols. We note that in all these protocols both parties take as input the security parameter $1^\kappa$. We omit this from the notation for the sake of brevity.

Definition 1 ($\Sigma$-protocols). Let $L \in \text{NP}$ with corresponding witness relation $R_L$. A protocol $\Pi = \langle P, V \rangle$ is a $\Sigma$-protocol for relation $R_L$ if it is a three-round public-coin protocol which satisfies:

- Completeness: For all $(x, w) \in R_L$, $\Pr[\text{Output}_V(P(x, w), V(x)) = 1] = 1 - \text{negl}(\kappa)$, 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 \((x, w) \in RL\).

• **Honest verifier zero-knowledge:** There exists a probabilistic polynomial time simulator \(S_{\Sigma}\) such that for all \((x, w) \in RL\), the distributions \(\{S_{\Sigma}(x, e)\}\) and \(\{\text{View}_V(P(x, w(x)), V(x, e))\}\) are statistically indistinguishable. Here \(S_{\Sigma}(x, e)\) denotes the output of simulator \(S\) upon input \(x\) and \(e\), such that \(V\)'s random tape (determining its query) is \(e\).

### 3.1 Oblivious Transfer

**Definition 2** (Oblivious Transfer). Oblivious transfer is a protocol between two parties, a sender \(S\) with input messages \((m_0, m_1)\) and receiver \(R\) with input a choice bit \(b\). The correctness requirement is that \(R\) obtains output \(m_b\) at the end of the protocol (with probability 1). We let \((S(m_0, m_1), R(b))\) denote an execution of the OT protocol with sender input \((m_0, m_1)\) and receiver input bit \(b\). We require OT that satisfies the following properties:

• **Computational Receiver Security.** For any non-uniform PPT sender \(S^*\) and any \((b, b') \in \{0, 1\}\), the views \(\text{View}_{S^*}((S^*, R(b)))\) and \(\text{View}_{S^*}((S^*, R(b')))\) are computationally indistinguishable.

  We say that the OT scheme is \(T\)-secure if \(T\cdot \text{poly}(\kappa)\)-size malicious senders have distinguishing advantage less than \(1/\rho\).

• **Statistical Sender Security.** This is defined using the real-ideal paradigm, and requires that for any distribution on the inputs \((m_0, m_1)\) and any unbounded adversarial receiver \(R^*\), there exists a (possibly unbounded) simulator \(\text{Sim}_{R^*}\) that interacts with an ideal functionality \(\mathcal{F}_{\text{ot}}\) on behalf of \(R^*\). Here \(\mathcal{F}_{\text{ot}}\) is an oracle that obtains the inputs \((m_0, m_1)\) from \(S\) and \(b\) from \(\text{Sim}_{R^*}\) (simulating the malicious receiver), and outputs \(m_b\) to \(\text{Sim}_{R^*}\). Then \(\text{Sim}_{R^*}^\mathcal{F}_{\text{ot}}\) outputs a receiver view that is statistically indistinguishable from the real view of the malicious receiver \(\text{View}_{R^*}((S(m_0, m_1), R^*))\). We say that the OT protocol satisfies \((1 - \delta)\) statistical sender security if the statistical distance between the real and ideal distributions is at most \(\delta\).

Throughout the paper (except Section 6), we focus on two-message oblivious transfer. We now discuss an additional specific property of two-message OT protocols.

**Property 1.** We define an additional property of two-message OT protocols: for any “valid” receiver message \(m_R\) (that does not cause an honest sender to abort), there must exist randomness \(r\) and receiver choice bit \(b\) such that \(m_R\) corresponds to an honestly generated receiver OT message using input bit \(b\) and randomness \(r\).

Such two-message protocols have been constructed based on the DDH assumption [NP01], 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 [Kal05, HK12]. Such two-message protocols can also be based on witness encryption or indistinguishability obfuscation (iO) together with one-way permutations [SW14].

We use the following sender security property in our protocols (which follows by combining Property 1 with the definition of sender security in Definition 2 above).

**Claim 1.** For any two-message OT protocol satisfying [Definition 2 and Property 1], for every malicious receiver \(R^*\) and every “valid” first message \(m_{R^*}\) generated by \(R^*\), we require that there exists an unbounded machine \(E\) which extracts \(b\) such that either of the following statements is true:
○ For all \(m_0, m_1, m_2\), \(\text{View}_{R^*}\langle S(m_0, m_1), R^* \rangle\) and \(\text{View}_{R^*}\langle S(m_0, m_2), R^* \rangle\) are statistically indistinguishable and \(b = 0\), or,

○ For all \(m_0, m_1, m_2\), \(\text{View}_{R^*}\langle S(m_0, m_1), R^* \rangle\) and \(\text{View}_{R^*}\langle S(m_2, m_1), R^* \rangle\) are statistically indistinguishable and \(b = 1\).

**Proof.** From the (unbounded) simulation property of the two-message OT, there exists a simulator that extracts a receiver input bit \(b\) 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, when \(b = 0\), the simulated view must be close to both \(\text{View}_{R^*}\langle S(m_0, m_1), R^* \rangle\), and \(\text{View}_{R^*}\langle S(m_0, m_2), R^* \rangle\). Similarly, when \(b = 1\), the simulated view must be statistically close to both \(\text{View}_{R^*}\langle S(m_0, m_1), R^* \rangle\), and \(\text{View}_{R^*}\langle S(m_2, m_1), R^* \rangle\).

Finally, we define bit OT as oblivious transfer where the sender inputs bits instead of strings.

**Definition 3 (Bit Oblivious Transfer).** We say that an oblivious transfer protocol according to **Definition 2** is a bit oblivious transfer if the senders messages \(m_0, m_1\) are each in \(\{0, 1\}\).

### 3.2 Proof Systems

**Delayed-Input Interactive Protocols.** An \(n\)-message delayed-input interactive protocol 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 security parameter, and execute the first \(n - 1\) messages.

○ Before sending the last message, \(P\) receives input \((x, w) \in R_L\). \(P\) sends \(x\) to \(V\) together with the last message of the protocol. Upon receiving the last message from \(P\), \(V\) outputs 1 or 0.

An execution of this protocol with instance \(x\) and witness \(w\) is denoted by \(\langle P(x, w), V(x) \rangle\). A delayed-input interactive protocol is a protocol satisfying the completeness and soundness condition in the delayed input setting. One can consider both proofs – with soundness against unbounded provers, and arguments – with soundness against computationally bounded provers. In particular, a delayed-input interactive argument satisfies adaptive soundness against malicious PPT provers. That is, soundness is required to hold even against PPT provers who choose the statement adaptively (maliciously), depending upon the first \(n - 1\) messages of the protocol.

**Definition 4 (Delayed-Input Interactive Arguments).** An \(n\)-message 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 \((x, w) \in R_L\),

\[
\Pr[\text{Output}_V\langle P(x, w), V(x) \rangle = 1] = 1 - \text{negl}(\kappa),
\]

where the probability is over the random coins of \(P\) and \(V\), and where in the protocol \(V\) receives \(x\) together with the last message of the protocol.

○ **Adaptive Soundness:** For every (non-uniform) PPT prover \(P^*\) that given \(1^\kappa\) chooses an input length \(1^p\), and then chooses \(x \in \{0, 1\}^p \setminus L\) adaptively, depending upon the transcript of the first \(n - 1\) messages,

\[
\Pr[\text{Output}_V\langle P^*, V \rangle(x) = 1] = \text{negl}(\kappa),
\]

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. In this paper, we only consider statistical witness indistinguishability, which we formally define below.

**Definition 5** (Statistical Witness Indistinguishability). A delayed-input interactive argument \((P, V)\) for a language \(L\) is said to be statistical witness-indistinguishable if for every unbounded verifier \(V^*\), every polynomially bounded function \(n = n(\kappa) \leq \text{poly}(\kappa)\), and every \((x_n, w_{1, n}, w_{2, n})\) such that \((x_n, w_{1, n}) \in R_L\) and \((x_n, w_{2, n}) \in R_L\) and \(|x_n| = n\), the following two ensembles are statistically indistinguishable:

\[
\{\text{View}_{V^*}(P(x_n, w_{1, n}), V^*(x_n))\} \text{ and } \{\text{View}_{V^*}(P(x_n, w_{2, n}), V^*(x_n))\}
\]

Delayed-Input 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 \([\text{Go}93, \text{DNRS}99]\), where the instances are chosen from some 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 \(\epsilon\)-WZK, which says that for every \(T_D\)-time distinguisher \(D\) and every distinguishing advantage \(\epsilon\) (think of \(\epsilon\) as an inverse polynomial) there exists a simulator, that is an oracle machine running in time \(\text{poly}(\kappa, 1/\epsilon)\) with oracle access to the distinguisher, that generates a view that \(D\) cannot distinguish from the view generated by the real prover. This notion was previously considered in \([\text{DNRS}99, \text{CLP}15, \text{JKKR}17]\).

When considering delayed-input interactive protocols it is natural to consider a delayed input version of secrecy. In what follows, we define delayed-input distributional statistical \(\epsilon\)-WZK.

**Definition 6** (Delayed-Input Distributional Statistical \(\epsilon\)-Weak Zero Knowledge). A delayed-input interactive argument \((P, V)\) for a language \(L\) is said to be delayed-input distributional statistical \(\epsilon\)-weak zero knowledge if for every polynomially bounded function \(n = n(\kappa) \leq \text{poly}(\kappa)\), and for every efficiently samplable distribution \((X_\kappa, W_\kappa)\) on \(R_L\), i.e., \(\text{Supp}(X_\kappa, W_\kappa) = \{(x, w) \in R_L : x \in \{0, 1\}^{n(\kappa)}\}\), every unbounded verifier \(V^*\) that obtains the instance from the prover in the last message of the protocol, every unbounded distinguisher \(D\), and every \(\epsilon\) (which will usually be set to \(1/\text{poly}(\kappa)\) for some polynomial \(\text{poly}(\cdot)\)), there exists an oracle machine – a simulator \(S\) that runs in time \(\text{poly}(\kappa, 1/\epsilon)\) with oracle access to \(D\) such that:

\[
\Pr_{(x, w) \leftarrow (X_\kappa, W_\kappa)} \left[D(x, \text{View}_{V^*}(P(x, w), V^*(x))) = 1\right]
\]

\[
- \Pr_{(x, w) \leftarrow (X_\kappa, W_\kappa)} \left[D(x, S^{V^*, D}(x)) = 1\right] \leq \epsilon(\kappa),
\]

where the probability is over the random choices of \((x, w)\) as well as the random coins of the parties. Note that because the simulator makes oracle calls to \(D\), if the distinguisher’s running time is \(T_D\), the total running time of the simulator becomes \(T_D \cdot \text{poly}(\kappa, 1/\epsilon)\).

Zero-Knowledge with Super-polynomial Simulation. We now define zero-knowledge with super-polynomial simulation in the same way as \([\text{Pas}03]\), except we prove statistical security against malicious verifiers.
Definition 7 (Statistical ZK with Super-polynomial Simulation). We say that a two message protocol \((P, V)\) for an NP language \(L\) is statistical zero-knowledge with super-polynomial \(T_{Sim}\)-time simulation, if it satisfies the following properties:

- **Delayed-Input Completeness.** For every \((x, w) \in R_L\), \(\Pr[\text{Output}_V(P(x, w), V(x))] = 1 - \text{negl}(\kappa)\), where the probability is over the random coins of \(P\) and \(V\).

- **Adaptive Soundness.** For every non-uniform polynomial-size \(P^*\), that upon receiving a security parameter \(1^\kappa\) chooses an instance length \(1^p\), and chooses the instance \(x \in \{0, 1\}^p\) adaptively after observing the verifier’s message, it holds that

\[
\Pr[\text{Output}_V(P^*, V)(x) = 1 \land x \notin L] = \text{negl}(\kappa),
\]

where the probability is over the random coins of \(V\).

- **Statistical Zero-Knowledge.** There exists a (uniform) simulator \(S\) that runs in time \(T_{Sim}\), such that for every polynomial \(n = n(\kappa) \leq \text{poly}(\kappa)\), and for every \((x_n, w_n) \in R_L\) where each \(|x_n| = n\), and every unbounded verifier \(V^*\), the two distributions \(SV^*(x_n)\) and \(\text{View}_{V^*}(P(x_n, w_n), V^*(x_n))\) are statistically close.

We will later define an additional extractability property of statistical SPS-ZK in Property 2.

4 Extractable Commitments

4.1 Definitions

Our notion of extractable commitments tailors the definition in [KS17] to the setting of statistically hiding commitments. We begin by (re-)defining the notion of a commitment scheme. As before, we use \(\kappa\) to denote the security parameter, and we let \(p = \text{poly}(\kappa)\) be an arbitrary fixed polynomial such that the message space is \(\{0, 1\}^p\).

We restrict ourselves to commitments with non-interactive decommitment, and where the (honest) receiver is not required to maintain any state at the end of the commit phase in order to execute the decommit phase. Our construction will satisfy this property and this will be useful in our applications to constructing statistically private arguments.

Definition 8. [Statistically Hiding Commitment Scheme] A commitment \(\langle C, R \rangle\) is a two-phase protocol between a committer \(C\) and receiver \(R\), consisting of algorithms \(\text{Commit}, \text{Decommit}\) and \(\text{Verify}\). At the beginning of the protocol, \(C\) obtains as input a message \(M \in \{0, 1\}^p\). Next, \(C\) and \(R\) execute the commit phase, and obtain a commitment transcript, denoted by \(\tau\), together with a private state for \(C\), denoted by \(\text{state}_{C, \tau}\). We use the notation

\[
(\tau, \text{state}_{C, \tau}) \leftarrow \text{Commit}(C(M), R).
\]

Later, \(C\) and \(R\) possibly engage in a decommit phase, where the committer \(C\) computes and sends message \(y = \text{Decommit}(\tau, \text{state}_{C, \tau})\) to \(R\). At the end, \(R\) computes \(\text{Verify}(\tau, y)\) to output \(\bot\) or a message \(\tilde{M} \in \{0, 1\}^p\)\(^8\).

A statistically hiding commitment scheme is required to satisfy three properties:

\[^8\] We note that in our definition, \(R\) does not need to keep a state from the commitment phase in order to execute the decommitment phase.
(Perfect) Completeness. If $C, R$ honestly follow the protocol, then for every $M \in \{0,1\}^p$:

$$\Pr[\text{Verify}(\tau, \text{Decommit}(\tau, \text{state}_{C, \tau})) = M] = 1$$

where the probability is over $(\tau, \text{state}_{C, \tau}) \leftarrow \text{Commit}(C(M), R)$.

Statistical Hiding. For every two messages $M_1, M_2 \in \{0,1\}^{2p}$, every unbounded malicious receiver $R^*$ and honest committer $C$, a commitment is $\delta(\kappa)$-statistically hiding if the statistical distance between the distributions $\text{View}_{R^*}(\text{Commit}(C(M_1), R^*))$ and $\text{View}_{R^*}(\text{Commit}(C(M_2), R^*))$ is at most $\delta(\kappa)$. The scheme is statistically hiding if $\delta(\kappa) \leq \frac{1}{\text{poly}(\kappa)}$ for every polynomial $\text{poly}(\cdot)$.

Computational Binding. Consider any non-uniform PPT committer $C^*$ that produces $\tau \leftarrow \text{Commit}(C^*, R)$, and then outputs $y_1, y_2$. Let $\widetilde{M}_1 = \text{Verify}(\tau, y_1)$ and $\widetilde{M}_2 = \text{Verify}(\tau, y_2)$. Then, we require that

$$\Pr[(\widetilde{M}_1 \neq \perp) \land (\widetilde{M}_2 \neq \perp) \land (\widetilde{M}_1 \neq \widetilde{M}_2)] = \text{negl}(\kappa),$$

over the randomness of sampling $\tau \leftarrow \text{Commit}(C^*, R)$.

In the following, we define a PPT oracle-aided algorithm $\text{Samp}$ such that for all $C^*$, $\text{Samp}^{C^*}$ samples $\tau \leftarrow \text{Commit}(C^*, R)$ generated by a malicious committer $C^*$ using uniform randomness for the receiver.

We also define an extractor $E$ that given black-box access to $C^*$, outputs some transcript generated by $C^*$, and then without executing any decommitment phase with $C^*$, outputs message $\widetilde{M}_e$; we require “correctness” of this extracted message $\widetilde{M}_e$. We also require that for any non-uniform PPT $C^*$, the distribution of $\tau$ generated by $\text{Samp}^{C^*}$ is indistinguishable from the distribution output by $E^{C^*}$. This is formally defined in Definition 9.

Definition 9. [$T$-Extractable Commitment Scheme] We say that a statistically hiding commitment scheme is $T$-extractable if there exists a $T \cdot \text{poly}(\kappa)$-time uniform oracle machine $E$ such that the following holds. Let $C^*$ be any non-uniform PPT adversarial committer, that before starting the commitment phase, outputs auxiliary information denoted by $z$, and at the end of the commitment phase outputs auxiliary information denoted by aux. Then, the following holds.

- There exists a PPT oracle sampling algorithm $\text{Samp}^{C^*}$ that samples $(\tau_{C^*}, \text{aux}) \leftarrow \text{Commit}(C^*, R)$. Let $\text{Exp}_{\text{Samp}^{C^*}} = (\tau_{C^*}, \text{aux})$ be the output of $\text{Samp}^{C^*}$.

- $E^{C^*}$ outputs $(\tau_{C^*}, \text{aux}, \widetilde{M})$, while only making oracle calls to $C^*$ during the commit phase (without ever running the decommit phase). We denote by $\text{Exp}_{E^{C^*}} = (\tau_{C^*}, \text{aux})$.

We require that:

- Indistinguishability. The distributions $(\text{Exp}_{\text{Samp}^{C^*}}, z)$ and $(\text{Exp}_{E^{C^*}}, z)$ are computationally indistinguishable.

- Correctness of Extraction. Consider any non-uniform PPT $C^*$ and let $(\tau, \text{aux}, \widetilde{M})$ denote the output of $E^{C^*}$. Then for any string $y_1$, denoting $\widetilde{M}_1 = \text{Verify}(\tau, y_1)$,

$$\Pr[(\widetilde{M} \neq \perp) \land (\widetilde{M}_1 \neq \perp) \land (\widetilde{M} \neq \widetilde{M}_1)] = \text{negl}(\kappa),$$

where the probability is over $(\tau, \text{aux}, \widetilde{M}) \leftarrow E^{C^*}$.
4.2 Protocol

In this section, we construct two-message statistically hiding, extractable commitments according to Definition 9. Our construction is described in Figure 5.

Let \( OT = (OT_1, OT_2) \) denote a two-message string oblivious transfer protocol according to Definition 2 satisfying Property 1. Let \( OT_1(b; r_1) \) denote the first message of the OT protocol with receiver input \( b \) and randomness \( r_1 \), and let \( OT_2(M_0, M_1; r_2) \) denote the second message of the OT protocol with sender input strings \( M_0, M_1 \) and randomness \( r_2 \).

**Extraction parameter:** \( m \).

**Committer Input:** Message \( M \in \{0, 1\}^p \).

**Commit Stage:**

**Receiver Message.**

- Pick challenge string \( ch \leftarrow \{0, 1\}^m \).
- Sample uniform randomness \( \{r_{1,i}\}_{i \in [m]} \).
- Compute and send \( \{OT_1(ch_i, r_{1,i})\}_{i \in [m]} \) using \( m \) instances of two-message OT.

**Committer Message.**

- Sample a random string \( r \leftarrow \{0, 1\}^m \).
  - For every \( i \in [m] \) and every \( b \in \{0, 1\} \), sample \( M_i^b \leftarrow \{0, 1\}^p \) subject to \( \bigoplus_{i \in [m]} M_i^b = M \).
- For every \( i \in [m] \) compute \( o_{2,i} = OT_2(M_i^0, M_i^1; r_{2,i}) \) with uniform randomness \( r_{2,i} \).
- Send \( (r, \{o_{2,i}\}_{i \in [m]}) \).

**Reveal Stage:** The committer reveals \( M \), and all values \( \{M_i^0, M_i^1\}_{i \in [m]} \) as well as the randomness \( r_{2,i} \). The receiver accepts the decommitment to message \( M \) if and only if:

1. For all \( i \in [m] \), \( o_{2,i} = OT_2(M_i^0, M_i^1; r_{2,i}) \),
2. \( \bigoplus_{i \in [m]} M_i^b = M \).

The value \( m \) will determine the running time \( T = 2^m \cdot \kappa \log \kappa \) of the extractor. The protocol will have statistical receiver security \( 1 - 2^{-m} - \delta_{OT} \), when the underlying OT has statistical sender security \( 1 - \delta_{OT} \).

**Figure 5: Extractable Commitments**

We will now prove the following main theorem.

**Theorem 1.** Set \( T = (2^m \cdot \kappa \log \kappa) \). Assuming that the underlying OT protocol is \( T \)-secure against malicious senders, \((1 - \delta_{OT}) \) secure against malicious receivers according to Definition 2 and satisfies Property 1, the scheme in Figure 5 is a \((1 - 2^{-m} - \delta_{OT}) \) statistically hiding, \( T \)-extractable commitment scheme according to Definition 9.

We will prove this theorem by showing statistical hiding, computational binding, and extractability in Lemma 1, Lemma 2, and Lemma 3 below respectively.

\(^9\)Note that \( OT_2 \) also depends on \( OT_1 \). We omit this dependence in our notation for brevity.
Lemma 1. Assuming the underlying OT satisfies \((1 - \delta_{\text{OT}})\) statistical sender security according to [Definition 2] and satisfies [Property 1], the scheme in [Figure 5] is \((1 - 2^{-m} - \delta_{\text{OT}})\) statistically hiding according to [Definition 8].

**Proof.** Fix any (unbounded) malicious receiver \(R^*\). Let \(m_{R^*}\) be the message sent by \(R^*\) during the commit phase. By Property 1, \(m_{R^*}\) uniquely defines a receiver challenge \(\text{ch}\). With probability \(2^{-m} \neq \text{ch}\) for \(r\) chosen uniformly at random by an honest committer. Conditioned on \(r \neq \text{ch}\), there exists at least one index \(j \in [m]\) such that \(r_j \neq \text{ch}_j\). When \(r_j \neq \text{ch}_j\), by \((1 - \delta_{\text{OT}})\) statistical sender security of OT, \(M_j^\ell\) is \((1 - \delta_{\text{OT}})\)-statistically hidden from any malicious receiver. Since \(M_j^\ell\) is one of the shares in an XOR secret sharing of \(M\), the message \(M\) is \((1 - 2^{-m} - \delta_{\text{OT}})\) statistically hidden from any malicious receiver.

Lemma 2. Assuming the underlying OT satisfies receiver security according to [Definition 2] and satisfies [Property 1], the scheme in [Figure 5] is computationally binding against non-uniform PPT malicious committers, according to [Definition 8].

**Proof.** Suppose for contradiction that there exists a non-uniform malicious PPT cheating committer \(C^*\) and a polynomial \(p(\cdot)\) such that outputs transcript \(\tau \leftarrow \text{Commit}(C^*, R), y_1, y_2\) such that \(M_1 = \text{Verify}(\tau, y_1), M_2 = \text{Verify}(\tau, y_2)\), and

\[
\Pr[\langle M_1 \neq \bot \rangle \land \langle M_2 \neq \bot \rangle \land \langle M_1 \neq M_2 \rangle] = \frac{1}{p(\kappa)}
\]

We will construct a reduction \(A\) that has black-box access to such a committer \(C^*\), and breaks receiver OT security according to [Definition 2] \(A\) takes as input \(o' = \text{OT}_1(b)\) and is required to distinguish the case when \(b = 0\) from when \(b = 1\). \(A\) does the following:

1. Sample \(\ell \leftarrow \{0, 1\}^{m-1}\).
2. Sample \(r_{1,i}\) uniformly at random and compute \(\{o_{1,i} = \text{OT}_1(\text{ch}_i, r_{1,i})\}_{i \in [m] \setminus \{\ell\}}\). Set \(o_{1,\ell} = o'\).
3. Forward \(\{o_{1,i}\}_{i \in [m]}\) as the first message of the scheme in Figure 5 to the adversary \(C^*\).
4. Obtain \(r_j, o_{2,i}\) from \(C^*\). This is the end of the commit phase, denote transcript by \(\tau\).
5. Obtain \(y_1, y_2\) from \(C^*\). If \(\text{Verify}(\tau, y_1) = \text{Verify}(\tau, y_2)\) or if either of them are \(\bot\), abort and output \(\bot\). Else continue.
6. Parse \(y_1\) as \(M, \{M_i^h, r_{2,i}\}_{i \in [m] \setminus \{0, 1\}}\) and \(y_2\) as \(\tilde{M}, \{\tilde{M}_i^h, \tilde{r}_{2,i}\}_{i \in [m] \setminus \{0, 1\}}\).
7. Define \(S = \{j \in [m] : M_j^\ell \neq \tilde{M}_j^\ell\}\). Since \(M \neq \tilde{M}\), \(|S| > 1\). If \(\ell \in S\), output \(1 - r_\ell\), else output \(\bot\).

We prove the following claim, which will contradict OT security according to [Definition 2] and complete the proof of the lemma.

**Claim 2.** Let \(b'\) denote the output of \(A\). Then, \(\Pr[b' = b] \geq \frac{1}{m \cdot p(\kappa)}\).

**Proof of Claim.** By assumption, with probability at least \(\frac{1}{p(\kappa)}\), the following event \(E\) occurs: \(A\) proceeds to Step 5 and obtains \(M \neq \tilde{M}\).

After \(A\) proceeds to Step 5, it first creates the set \(S\) of indices \(j \in [m]\) where \(M_j^\ell \neq \tilde{M}_j^\ell\). Since \(M \neq \tilde{M}\), \(|S| > 1\). Since \(A\) samples \(\ell\) independently and uniformly at random, \(\Pr[\ell \in S | E] \geq \frac{1}{m}\). Thus, \(\Pr[\ell \in S] \geq \frac{1}{m \cdot p(\kappa)}\).
Let us now condition on \((\ell \in \mathcal{S}) \land \exists \). For any \(i \in [m]\), by correctness of the \(i^{th}\) parallel OT, the OT sender is bound to a unique input value \(M^{ch}_i\). Thus, for any \(i \in [m]\), the existence of \(M^{ch}_i \neq \tilde{M}^{ch}_i\) that reconstruct to \(a_{2,i}\) implies that \(ch_i \neq r_i\) (except with probability \(\negl(\kappa)\)). Since we conditioned on \(\ell \in \mathcal{S}\), we have that \(M^{ch}_i \neq \tilde{M}^{ch}_i\), and therefore \((b = ch) = (1 - r) = b'\).

Thus, \(\Pr[b' = b | (\ell \in \mathcal{S}) \land \exists \] = 1. This implies that \(\Pr[b' = b \land (\ell \in \mathcal{S}) \land \exists \] \geq \frac{1}{m \cdot p(\kappa)}\). Since \(\mathcal{A}\) outputs \(\bot\) if either of the events \((\ell \in \mathcal{S})\) and \(\exists\) did not occur, we have that \(\Pr[b' = b] \geq \frac{1}{m \cdot p(\kappa)}\), proving the claim.

**Lemma 3.** The scheme in Figure 6 is a \(T\)-extractable commitment scheme, where \(T = 2^m \cdot \kappa^{\log \kappa}\).

**Proof.** We begin by describing the extractor \(\mathcal{E}\) from Definition 9 in Figure 6. In the figure, we denote the first message of transcript \(\tau\) by \(\tau_1\) and the second message by \(\tau_2\). \(\mathcal{E}\) will obtain oracle access to \(C^*\), and the running time of \(\mathcal{E}^{C^*}\) will be \(T = 2^m \cdot \kappa^{\log \kappa}\).

\(\mathcal{E}^{C^*}\) repeats the following \(2^m \cdot \kappa^{\log \kappa}\) times. If it reaches the end of \(2^m \cdot \kappa^{\log \kappa}\) iterations, it outputs \(\bot\). We will call each iteration a *trial*.

1. Choose \(ch \leftarrow \{0,1\}^m\). Compute \(\tau_1 = \text{OT}_1(ch, R_i)\) using uniform randomness \(R = \{R_i\}_{i \in [m]}\).
2. Query the oracle \(C^*\) in the Commit phase with \(\tau_1\), and obtain response \((\tau_2, aux)\), where \(\tau_2\) also contains \(r\). If \(C^*\) aborts or sends an invalid message, do the following.
   - If this is the first iteration, output \((\tau_1, \tau_2, aux, \bot)\) and stop.
   - If this is not the first iteration, go to Step 1 and start a new trial.
3. Else, \(C^*\) did not abort. If \(r \neq ch\), go to Step 1 and start a new trial.
4. Else, \(C^*\) did not abort and \(r = ch\) (this iteration is considered a success). Then use \(R\) to obtain \(\{M^{ch}_i\}_{i \in [m]}\). Next, compute \(\tilde{M} = \bigoplus_{i \in [m]} \{\tilde{M}^{ch}_i\}_{i \in [m]}\). Output \((R, \tau_1, \tau_2, aux, \tilde{M})\).

Figure 6: Description of the Extractor \(\mathcal{E}^{C^*}\)

We will now analyze the extractor \(\mathcal{E}\). We consider two cases:

1. Suppose \(\Pr[\langle C^*, R \rangle(\cdot) \text{ aborts}] = 1 - \negl(\kappa)\). In this case, observe that the extractor also outputs aborting views with \(\bot\) as the extracted message, with probability equal to \(1 - \negl(\kappa)\), and so does the sampler \(\text{Samp}\). Therefore, joint distribution output by the extractor is indistinguishable from that output by \(\text{Samp}\).

2. Else, suppose \(\Pr[\langle C^*, R \rangle(\cdot) \text{ aborts}] = 1 - \delta\), where \(\delta \geq \frac{1}{p(\kappa)}\) for some polynomial \(p(\cdot)\). If \(C^*\) aborts in the first iteration, then \(\mathcal{E}^{C^*}\) will also abort and output \(R\) together with \((\tau_1, \tau_2, aux, \bot)\), where the latter is trivially close to the distribution output by \(\text{Samp}\), conditioned on outputting an aborting transcript. On the other hand, if \(C^*\) does not abort, then \(\mathcal{E}\) will try \(2^m \cdot \kappa^{\log \kappa}\) iterations and attempt to extract the message.

In the rest of the proof, we will only analyze this case. We will prove that with overwhelming probability conditioned on a non-aborting trial being output in the first iteration, the extrac-
tor will successfully output a non-aborting trial together with an extracted message \( \tilde{M} \), such that the output distribution is indistinguishable from that of \textbf{Samp}.

We begin with the following observations.

- After \( q = (1/\delta) \log(1/\delta) \cdot m \cdot k^{\log \kappa} \) independent trials, the probability of obtaining at least one non-aborting view (in other words, \( E \) proceeding to Step 3) is at least \( 1 - (2m \cdot k^{\log \kappa}) \).

- Since the extractor performs \( 2m \cdot k^{\log \kappa} \) trials, by a union bound, except with probability \( \frac{\delta}{k^{\log \kappa}} \), the extractor obtains at least \( (2m \cdot \kappa) \) non-aborting views with independently chosen \( \text{ch} \).

Recall that extraction succeeds from a trial when the \( r \) chosen by the committer is equal to the \( \text{ch} \) chosen by the extractor in this trial. We now prove the following claim, which asserts that the event \( r = \text{ch} \) occurs with “good enough” probability. Here \( \delta_{\text{malS}} \) denotes the distinguishing advantage of a malicious sender in the OT protocol. Recall that we set \( \delta_{\text{malS}} \) to \( 1/T = 2^{-m} \cdot k^{-\log \kappa} \).

**Claim 3.** Whenever \( \delta_{\text{malS}} < 2^{-m} \), the following is true: In any individual trial, conditioned on the trial not aborting, the probability that \( r = \text{ch} \) is at least \( 2^{-m} (1 - \delta) \).

**Proof of Claim.** Suppose the claim is not true. We will use this to contradict the receiver security of the OT protocol. To do so, we describe our reduction algorithm \( \hat{A} \) and distinguisher \( \hat{D} \) for the receiver security game.

The reduction \( \hat{A} \) picks two challenges \( \text{ch}_1, \text{ch}_2 \leftarrow \{0,1\}^m \) at random, and creates auxiliary information consisting of these two challenges. We will use \( \text{OT}_1(\text{ch}; R) \) to denote \( \{\text{OT}_1(\text{ch}_i; R_i)\}_{i \in [m]} \).

Now, \( \hat{D} \) will obtain from the OT challenger as challenge either \( \tau_1 = \text{OT}_1(\text{ch}_1; R) \) or \( \tau_1 = \text{OT}_1(\text{ch}_2; R) \). \( \hat{D} \) runs the malicious sender \( \mathcal{C}^* \) on the message \( \tau_1 \), obtaining \( \tau_2 \). It finds \( r \) from \( \tau_2 \): if \( r = \text{ch}_1 \), it outputs 1. Otherwise, it aborts and outputs \( \perp \).

We will now analyze the probability that \( \hat{D} \) outputs 1 in the two cases where \( \text{ch} = \text{ch}_1 \) or \( \text{ch} = \text{ch}_2 \). If \( \text{ch} = \text{ch}_1 \), then by assumption, we have that, conditioned on a non-aborting trial, \( \Pr[\hat{D} = 1| \text{ch} = \text{ch}_1] < 2^{-m} (1 - \delta) \). On the other hand, if \( \text{ch} = \text{ch}_2 \), then no information about \( \text{ch}_1 \) is given to the distinguisher \( \hat{D} \). Therefore, conditioned on a non-aborting trial, \( \Pr[\hat{D} = 1| \text{ch} = \text{ch}_2] = 2^{-m} \).

Thus, we have that \( |\Pr[\hat{D} = 1| \text{ch} = \text{ch}_1| - \Pr[\hat{D} = 1| \text{ch} = \text{ch}_2]| \geq 2^{-m} \cdot \delta \), which is a contradiction if \( 2^{-m} \cdot \delta > \delta_{\text{malS}} \).

Then, the following claim proves that the extraction is “successful”, that is, the extractor outputs a transcript and correctly

**Claim 4.** Whenever \( \delta_{\text{malS}} \ll 2^{-m} \), when interacting with committer \( \mathcal{C}^* \) that does not abort with overwhelming probability, the extraction algorithm \( \mathcal{E}^{\mathcal{C}^*} \) outputs a transcript \((\tau_1, \tau_2, \text{aux})\) and \( \tilde{M} \) such that with probability at least \( 1 - \text{negl}(\kappa) \), the following holds:

- The transcript \((\tau_1, \tau_2)\) statistically binds \( \mathcal{C}^* \) to a single message \( M \), and,

- \( \tilde{M} = M \).

**Proof.** By [Claim 3] within \( 2m \cdot \kappa \) independent non-aborting trials, with probability \( 1 - \text{negl}(\kappa) \), \( r = \text{ch} \) in at least one non-aborting trial. Now, when \( r = \text{ch} \), the extractor obtains \( \{M_i^{\text{ch}_i}\}_{i \in [m]} \) as OT output, and outputs \( \tilde{M} = \bigoplus_{i \in [m]} M_i^{\text{ch}_i} \). By correctness of OT and since \( r = \text{ch} \), \( M = \bigoplus_{i \in [m]} M_i^{\text{ch}_i} \), correctness of extracted value follows. Moreover, by correctness of OT, the committer is statistically bound to a single input for each index \( i \in [m] \), and therefore to a single message in all transcripts where \( r = \text{ch} \).
Now we proceed to the proof that the extractor \( E \) outputs views that are indistinguishable from views that would be output by \( \text{Samp} \).

First, we note that aborting views output by the extractor (that is, the case when \( C^* \) aborts in Step 2 on the first iteration in Figure 6) are identically distributed to aborting views output by the sampler. Thus, no distinguisher can gain an advantage on observing aborting views. On the other hand, if \( C^* \) did not output an aborting view in the first iteration, then with probability \( 1 - \text{negl}(\kappa) \), \( E \) outputs a non-aborting view together with an extracted value.

We now suppose that there is a distinguisher that distinguishes between \((\tau_1, \tau_2, \text{aux})\) output by \( E^{C^*} \) and that output by \( \text{Samp} \). In other words, we have a committer \( C^* \) and distinguisher \( D \) such that:

\[
|\Pr[D(\text{Exp}_{\text{Samp}}, z) = 1] - \Pr[D(\text{Exp}_E, z) = 1]| > \delta'(\kappa)
\]

Without loss of generality, we can assume that there exists a probability \( p \) such that:

\[
\Pr[D(\text{Exp}_{\text{Samp}}, z) = 1] = p - \delta'
\]

and

\[
\Pr[D(\text{Exp}_E, z) = 1] = p
\]

where \( \delta' > \frac{1}{\text{poly}(\kappa)} \) (if \( \delta' < -\frac{1}{\text{poly}(\kappa)} \), we can flip the output of the distinguisher).

We will use this distinguisher to contradict the receiver security of the OT protocol. To do so, we describe our reduction algorithm \( \hat{A} \) and distinguisher \( \hat{D} \) for the receiver security game.

The reduction \( \hat{A} \) chooses two challenges \( ch_1, ch_2 \leftarrow \{0,1\}^m \) at random, and creates auxiliary information consisting of these two challenges. Next, \( \hat{A} \) obtains as input (from the OT challenger) either \( \tau_1 = \text{OT}_1(ch = ch_1; R) \) or \( \tau_1 = \text{OT}_1(ch = ch_2; R) \).

\( \hat{A} \) now runs the malicious sender \( C^* \) on the message \( \tau_1 \), obtaining \( \tau_2, \text{aux} \). It generates the joint distribution \((\tau_1, \tau_2, \text{aux})\). If \( r = ch_1 \), it outputs \( D(\tau_1, \tau_2, \text{aux}) \). Otherwise, it aborts and outputs \( \bot \).

We now analyze two cases.

- Suppose \( ch = ch_1 \). Observe that if \( r = ch_1 \), this is the distribution that corresponds to the output \( \text{Exp}_E \) in Definition 9.

  Formally, by Claim 4 the distribution output by \( E \) is \( 1 - \text{negl}(\kappa) \) statistically close to a distribution that exactly consists of \((\tau_1, \tau_2, \text{aux})\) corresponding to transcripts where \( r = ch \)

  In this case, conditioned on a non-aborting trial of \( C^* \) in the first iteration, by Claim \[3\] we have that \( \Pr[r' = ch_1 | ch = ch_1] \geq 2^{-m}(1 - \delta \cdot \text{negl}) \). Thus, \( \Pr[\hat{D} = 1 | ch = ch_1] \geq p \cdot 2^{-m}(1 - \delta \cdot \text{negl}) \).

- Suppose \( ch = ch_2 \). Observe that this is the distribution that corresponds to the output of \( \text{Exp}_{\text{Samp}} \) in Definition 9. In this case, no information about \( ch_1 \) is given to the adversary. Thus, conditioned on a non-aborting trial of \( C^* \) in the first iteration, we have that \( \Pr[r' = ch_1 | ch = ch_2] \leq 2^{-m} \). Thus, \( \Pr[\hat{D} = 1 | ch = ch_2] \leq (p - \delta') \cdot (2^{-m}) \).

Since \( 2^{-m} \delta > \delta_{\text{malS}} \), we have that

\[
|\Pr[\hat{D} = 1 | ch = ch_1] - \Pr[\hat{D} = 1 | ch = ch_2]| > 2^{-m} \cdot (\delta' - \delta \cdot \text{negl}) > \delta_{\text{malS}}
\]

This is a contradiction, and therefore the lemma follows. \( \square \)

It is easy to observe that this commitment scheme composes, such that the joint distribution of values committed in parallel via \( n \) commitments is extractable in time \( 2^{nm} \cdot \kappa^{\log \kappa} \). Furthermore,
exactly as in [KS17], this lemma can be generalized for non-uniform malicious committers that run in super polynomial time \( T_C \): the extractor’s running time as well as the receiver security of the underlying OT will need to be higher by a factor of \( T_C \). Furthermore, indistinguishability of the real samples and the extracted transcripts also generalizes to hold against non-uniform distinguishers that run in superpolynomial time \( T' \), by similarly strengthening the receiver security of OT.

**Lemma 4.** Assuming OT secure against \( \max(T', 2^m \cdot T_C \cdot \kappa^{\log \kappa}) \)-time malicious senders according to **Definition 2**, the protocol in Figure 7 is an extractable commitment according to **Definition 9**, secure against malicious committers running in time \( T_C \), with an extractor running in time \( 2^m \cdot T_C \cdot \kappa^{\log \kappa} \), such that the distributions \( \text{Exp}_{E^*} \) and \( \text{Samp}_{E^*} \) are indistinguishable by \( T' \)-size distinguishers.

5 Two-Message Arguments with Statistical Privacy

5.1 Modified Blum Protocol

We begin by describing a very simple modification to the Blum \( \Sigma \)-protocol for Graph Hamiltonicity. The protocol we describe will have soundness error \( \frac{1}{2} - \text{negl}(\kappa) \) against adaptive PPT provers, and will satisfy statistical zero-knowledge. Since Graph Hamiltonicity is NP-complete, this protocol can also be used to prove any statement in NP via a Karp reduction. This protocol is described in Figure 7.

We give an overview of the protocol here. Note that the only modification to the original protocol of Blum [Blu86] is that we use statistically hiding, extractable commitments instead of statistically binding commitments. The proofs of soundness and statistical zero-knowledge are fairly straightforward. They roughly follow the same structure as [Blu86], replacing statistically binding commitments with statistically hiding commitments.

**Lemma 5.** Assuming that \( \text{extcom} \) is computationally binding, the protocol in Figure 7 satisfies soundness against PPT provers that may choose \( x \) adaptively in the second round of the protocol.

**Proof.** The proof of soundness follows by the computational binding property of \( \text{extcom} \) and the soundness of the (original) Blum protocol.

Let \( L \) denote the language consisting of all graphs that have a Hamiltonian cycle. Consider a cheating prover \( P^* \) that convinces a malicious verifier of a statement \( x \notin L \) with probability \( \frac{1}{2} + h(n) \), where \( h(\cdot) > \frac{1}{\text{poly}(\cdot)} \) for some polynomial \( \text{poly}(\cdot) \). By a simple averaging argument, this means that there exists at least one transcript prefix \( \tau \) consisting of the first two messages of the protocol, where for \( G \notin L \) sent by the prover in the third message, \( \Pr[V \text{ accepts } | \tau, G \notin L] > \frac{1}{2} \). This implies that there exists a cheating prover that generates a transcript prefix \( \tau \), for which it provides an accepting opening corresponding to both \( b = 0 \) and \( b = 1 \), with probability at least \( h(n) \). Next, we argue that such a cheating prover must break the (computational) binding of \( \text{com} \).

Since \( G \notin L \), it is information theoretically impossible for any cheating prover to generate a commitment to a unique string \( \pi, \pi(G) \) such that there exists a Hamiltonian cycle in \( \pi(G) \). Therefore, any prover that opens a transcript prefix \( \tau, G \) corresponding to both \( b = 0 \) and \( b = 1 \) for \( G \notin L \), must open at least one commitment in the set \( \{ \text{extcom}_P, \{ \text{extcom}_{i,j} \}_{i,j \in \mathbb{P} \times \mathbb{P}} \} \) to two different values, thereby giving a contradiction to the binding of the commitment scheme.

**Lemma 6.** Assuming that \( \text{extcom} \) is statistically hiding, the protocol in Figure 7 satisfies statistical zero-knowledge.

**Proof.** The simulation strategy is identical to that of [Blu86]. The simulator \( \text{Sim} \) first guesses the challenge bit \( c' \). It begins an interaction with the malicious verifier. On obtaining the first message
Modified Blum Argument

1. **Verifier Message:** The verifier does the following:
   - Send the first message \( \text{extcom}_{1,i,j} \) for independent instances of the extractable commitment, where \( i, j \in [p(\kappa)] \times [p(\kappa)] \).
   - Send an additional first message \( \text{extcom}_{1,P} \) for another independent instance of the extractable commitment.

2. **Prover Message:** The prover gets input graph \( G \in \{0, 1\}^{p(\kappa) \times p(\kappa)} \) represented as an adjacency matrix, with \((i, j)^{th}\) entry denoted by \( G[i][j] \), Hamiltonian cycle \( H \subseteq G \). Here \( p(\cdot) \) is an a-priori fixed polynomial. The prover does the following:
   - Sample a random permutation \( \pi \) on \( p(\kappa) \) nodes, and compute \( c_P = \text{extcom}_{2,P}(\pi) \) as a commitment to \( \pi \) using \( \text{extcom} \).
   - Compute \( \pi(G) \), which is the adjacency matrix corresponding to the graph \( G \) when its nodes are permuted according to \( \pi \). Compute \( c_{i,j} = \text{extcom}_{2,i,j}(\pi(G)[i][j]) \) for \((i, j) \in [p(\kappa)] \times [p(\kappa)] \).
   - Send \( G, c_P, c_{i,j} \) for \((i, j) \in [p(\kappa)] \times [p(\kappa)] \).

3. **Verifier Message:** Sample and send \( c \leftarrow \{0, 1\} \) to the prover.

4. **Prover Message:** The prover does the following:
   - If \( c = 0 \), send \( \pi \) and the decommitments of \( \text{extcom}_P, \text{extcom}_{i,j} \) for \((i, j) \in [p(\kappa)] \times [p(\kappa)] \).
   - If \( c = 1 \), send the decommitment of \( \text{extcom}_{i,j} \) for all \((i, j) \) such that \( \pi(H)[i][j] = 1 \).

5. **Verifier Output:** The verifier does the following:
   - If \( c = 0 \), accept if and only if all \( \text{extcom} \) openings were accepted and \( \pi(G) \) was computed correctly by applying \( \pi \) on \( G \).
   - If \( c = 1 \), accept if and only if all \( \text{extcom} \) openings were accepted and all the opened commitments form a Hamiltonian cycle.

Figure 7: Modified Blum SZK Argument

from the verifier, if \( c' = 0 \), it samples \( \pi \) uniformly at random and generates a commitment to \( \pi, \pi(G) \) following honest prover strategy to generate the commitment. If \( c' = 1 \), it samples \( \pi, H' \) uniformly at random where \( H' \) is an arbitrary hamiltonian cycle, and generates a commitment to \( \pi, \pi(H') \) following honest prover strategy to generate the commitment. Next, it waits for the verifier to send \( c \), and if \( c \neq c' \), it aborts and repeats the experiment. If \( c = c' \), then it decommits to the commitments according to honest prover strategy.

Note that when \( c = c' = 1 \), the resulting simulation is perfect zero-knowledge since the simulated view of the verifier is identical to the view generated by an honest prover. On the other hand when \( c = c' = 0 \), it follows from the statistical hiding property of the commitment \( \text{extcom} \) that the verifier cannot distinguish the case where \( \text{extcom} \) is a commitment to \( \pi, \pi(G) \) and a hamiltonian cycle is opened in \( \pi(G) \), from the case where \( \text{extcom} \) is not a commitment to \( \pi(G) \), but instead to some
$\pi(H')$ for a hamiltonian cycle $H'$.

Since honest-verifier zero-knowledge composes under parallel repetition, we also have the following lemma:

**Lemma 7.** Assuming that $\text{extcom}$ is statistically hiding, the protocol in Figure 7 satisfies honest verifier statistical zero-knowledge, even under parallel repetition.

### 5.2 Compressing Four Message Argument to a Two Message Argument

In Figure 8, we describe the construction of a two-message argument, using extractable commitments (with two messages denoted by $\text{ext-com}_1$, $\text{ext-com}_2$) according to Definition 9. This essentially consists of compressing the modified Blum argument from Figure 7 into a two-message argument.

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

Let $\Sigma = (q, a, e, z)$ denote the four messages of a the modified Blum protocol from Figure 7. Here $(q, a)$ denote the messages of the extractable commitment. We will perform a parallel repetition of this protocol, thus for each $i \in [\kappa]$, $(q_i, a_i, e_i, z_i)$ are messages corresponding to an underlying modified Blum 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.

---

**Two-Message Argument**

- **Verifier Message:**
  - Pick $\{q_i\}_{i \in [\kappa]}$ and pick challenge $\{e_i\}_{i \in [\kappa]}$ for the modified Blum Protocol.
  - Compute $\{o_{1,i} = \text{OT}_{1,i}(e_i)\}_{i \in [\kappa]}$.
  - Send $\{q_i, o_{1,i}\}_{i \in [\kappa]}$ in parallel.

- **Prover Message:**
  - Obtain input $x \in L$, witness $w$ such that $R_L(x, w) = 1$.
  - Compute $\{a_i\}_{i \in [\kappa]}$ according to the strategy in Figure 7.
  - Compute $\{z^0_i\}_{i \in [\kappa]}$ according to the strategy in Figure 7 using $(q_i, a_i, e'_i)$ corresponding to verifier challenge bit $e'_i = 0$.
  - Compute $\{z^1_i\}_{i \in [\kappa]}$ according to the strategy in Figure 7 using $(q_i, a_i, e'_i)$ and corresponding to verifier challenge bit $e'_i = 1$.
  - Compute $o_{2,i} = \text{OT}_{2,i}(z^0_i, z^1_i)$ and send $\{a_i, o_{2,i}\}_{i \in [\kappa]}$.

- **Verifier Output:** The verifier $V$ recovers $z_i$ as the output of $\text{OT}_{1,i}, \text{OT}_{2,i}$ for $i \in [\kappa]$, and outputs accept if for all $i \in [\kappa]$, $(q_i, a_i, e_i, z_i)_{i \in [\kappa]}$ is an accepting transcript of the underlying modified Blum protocol.

---

Figure 8: Two Message Argument System for NP
Lemma 8. Assuming that extcom is a $2^m \cdot \kappa^{\log \kappa}$-extractable commitment scheme according to Definition 9 and that OT is $2^m \cdot \kappa^{\log \kappa}$-secure, the protocol in Figure 8 satisfies soundness against PPT malicious provers.

Furthermore, assuming that the distributions $\text{Exp}_{E^*}$ and $\text{Exp}_{\text{ Samp.c}^*}$ corresponding to extcom, Definition 9 are indistinguishable by $T'$-size distinguishers (refer Lemma 4), the protocol in Figure 8 satisfies adaptive soundness against all PPT provers, when the instance is chosen from a language that is decidable by $T$-size circuits.

Proof. We will prove soundness by contradiction. Suppose there exists a cheating prover $P^*$ that with probability at least $p(\kappa) = \frac{1}{\text{poly}(\kappa)}$, outputs $x \notin L$ together with an accepting transcript. We will use the prover $P^*$ to construct an adversarial committer $C^*$ for $\kappa$ copies of the extractable commitment. We will finally use this to break the receiver security of the underlying OT protocol.

Recall that the messages $\{(q_i,a_i)\}_{i \in [\kappa]}$ correspond to $\kappa$ parallel repetitions of the extractable commitment scheme according to Definition 9. $C^*$ does the following:

- Obtain the receiver message $\{q_i\}_{i \in [\kappa]}$ for extcom externally.
- Sample uniform randomness $R$. Generate $ch_i = \{\text{OT}_1(e_i)\}_{i \in [\kappa]}$ using honest verifier strategy and randomness $R$, according to Figure 8.
- Send $\{q_i,o_{1,i}\}_{i \in [\kappa]}$ to $P^*$, and obtain $\{a_i,o_{2,i}\}_{i \in [\kappa]}$, $x$ from $P^*$.
- Set $\text{aux} = x,R,\{o_{1,i},o_{2,i}\}_{i \in [\kappa]}$.
- Output $\{a_i\}_{i \in [\kappa]}$ as committer message. Also output aux.

The following claim follows by description of the protocol (Figure 8).

Claim 5. The distribution $(\tau,\text{ aux})$ output by $C^*$ is such that with probability at least $p(\kappa)$, $x \notin L$ (this $x$ is part of aux) and the values $\{(q,i,a_i,e_i,z_i)\}_{i \in [\kappa]}$ corresponding to $(\tau,\text{ aux})$ constitute an accepting transcript of the modified Blum protocol.

Next, run the $2^m \cdot \kappa^{\log \kappa}$-time extractor $E$, that is guaranteed for $C^*$. By Definition 9 $E$ outputs $(\tilde{\tau},\tilde{\text{ aux}},\text{ val})$ such that:

- The joint distribution $(\tilde{\tau},\tilde{\text{ aux}})$ is indistinguishable by $T'$-time non-uniform machines from the joint distribution $(\tau,\text{ aux})$ output by $C^*$ above. In particular, since $L \in \text{DTIME}(T')$, this implies that with probability at least $p(\kappa) - \text{negl}(\kappa)$ over the randomness of the extractor, the joint distribution $(\tilde{\text{ aux}},\tilde{\text{ val}})$ also satisfy Claim 5 above.
- The transcript of $\tilde{\tau}$ is statistically binding to the unique value $\text{ val}$.

We will now describe a reduction $A$ that uses such an extractor $E$ to break the receiver security of OT according to Definition 2. $A$ does the following:

- First sample $ch_1, ch_2 \xleftarrow{\$} \{0,1\}^*$, and send to the OT challenger.
- Obtain in parallel $\{\text{OT}_1(e_i)\}_{i \in [\kappa]}$ externally as encoding either $e = ch_1$ or $e = ch_2$, and generate $\{q_i\}_{i \in [\kappa]}$ for extcom on its own using honest receiver strategy.
- Next, run committer $C^*$, and obtain output $\{a_i\}_{i \in [\kappa]}$. 

27
Then, run extractor $\mathcal{E}$ on $\mathcal{C}^*$ in time $2^{m-p(\kappa)\cdot \kappa}$ to obtain output $(\tilde{\text{aux}}, \text{extcom}, \text{val})$. Parse $\text{val}$ as $\kappa$ random variables, $v_1, v_2, \ldots, v_\kappa$, corresponding to $\kappa$ parallel repetitions of the modified Blum protocol. If $x$ is output adaptively by $P^*$, run in time $T'$ to decide if $x \in L$, and if $x \not\in L$, abort and output 0.

Else, compute string $\text{ch}'$ as follows, where $\text{ch}_i$ denotes the $i^{th}$ bit of $\text{ch}'$:

1. For all $i \in [\kappa]$, if $v_i$ consists of $\pi, G'$ such that $G' = \pi(G)$, set $\text{ch}_i' = 0$.
2. Else set $\text{ch}_i' = 1$.

Output $\text{ch}'$.

With probability at least $p(\kappa) - \text{negl}(\kappa)$, $(\tilde{\tau}, \tilde{\text{aux}})$ output by $\mathcal{E}$ is an accepting transcript with instance $x \not\in L$. Since the commitment transcript $\text{extcom}$ output by $\mathcal{E}$ is such that no unbounded $\tilde{C}$ can decommit $\tilde{\tau}$ to $\tilde{\text{value}} \neq \text{val}$. Thus, no (unbounded) cheating prover can open the transcript $\text{extcom}$ to two different values, except with probability $1 - \text{negl}(\kappa)$.

Moreover, since $x \not\in L$ (that is, the corresponding Graph $G$ does not contain any Hamiltonian Cycle), it must be the case that a committed value at any position of the parallel repetition $i \in [\kappa]$, can either consist of the correctly permuted graph $G$ that is missing a Hamiltonian cycle (which will verify correctly iff receiver challenge bit is 0), or a different graph $G'$ that does not correspond to a correctly permuted $G$ but instead has a Hamiltonian cycle (which will verify correctly iff receiver challenge bit is 1). It follows that the reduction outputs $\text{ch}' = e$ with probability at least $p(\kappa) - \text{negl}(\kappa)$, which directly contradicts receiver input-hiding security of OT$_1$ against machines running in time $2^{m^2 \cdot \kappa} \cdot \kappa^\log \kappa$.

**Remark 2.** This proof exactly generalizes to executing only $\Omega(\log \kappa)$ parallel executions of the Blum protocol, while still yielding negligible soundness error. Furthermore, we will see that statistical privacy guarantees will hold even when $m = \Omega(\log \kappa)$. Therefore, the protocol in [Figure 8] can be realized only relying on quasi-polynomially secure oblivious transfer according to Definition 2 and satisfying Property 1.

Similar to the extractability of commitments, we also define an additional property of two-message arguments, that we call extractability. This property will be useful in our applications to obtaining stronger forms of OT, and we believe will also be useful for other future applications.

**Property 2.** An argument is said to be $T$-extractable if there exists a $T$-poly($\kappa$) time uniform oracle machine $\mathcal{E}$ such that the following holds. Let $P^*$ be any non-uniform PPT adversarial adaptive prover, that before starting the commitment phase, may output some auxiliary information denoted by $z$, and at the end of the commitment phase may output some auxiliary information denoted by $\text{aux}$. Then, the following holds:

- There exists a PPT sampling oracle $\text{Samp}^{P^*}$ that samples transcript $(\tau_{P^*}, \text{aux}) \leftarrow (P^*, V)$.
- Let $\text{Exp}_{\text{Samp}^{P^*}} = (\tau_{P^*}, \text{aux})$ be the output of $\text{Samp}^{P^*}$.
- $\mathcal{E}^{P^*}$ outputs $(\tau_{P^*}, \text{aux}, w)$, while only making oracle access to $P^*$ during the commit phase (without ever running the decommit phase). $\mathcal{E}^{P^*}$ also outputs an honest verifier state corresponding to $\tau$. We denote by $\text{Exp}_{\mathcal{E}^{P^*}} = (\tau_{P^*}, \text{aux})$.

We require that:

- The distributions $(\text{Exp}_{\text{Samp}^{P^*}}, z)$ and $(\text{Exp}_{\mathcal{E}^{P^*}}, z)$ are computationally indistinguishable.
If the transcript $\tau$ contains an instance $x$, and is an accepting transcript, then $R(x, w) = 1$. 

We show that the scheme in Figure 8 is also extractable according to Property 2 where the extractor for the argument can extract a transcript with a witness, from any prover, by relying on the extractor of the commitment scheme $\text{extcom}$.

Lemma 9. The scheme in Figure 8 is $2^{\alpha \kappa} \cdot \kappa^{1 \log \kappa}$ extractable according to Property 2.

Proof. Suppose there exists a cheating prover $P^*$ that generates accepting transcripts with probability $p = \frac{1}{\text{poly}(\kappa)}$ for some polynomial $\text{poly}(\cdot)$.

The extractor $E'$ for the argument system directly the extractor $E$ for the $\kappa$ parallel repetitions of $\text{extcom}$ to obtain the joint distribution $\kappa$ committed values $\text{val}$, together with a transcript $\tilde{\tau}$ and $\tilde{\text{aux}}$. By indistinguishability of transcripts and auxiliary information output by $E$, $\tilde{\text{aux}}$ must also contain an accepting transcript of the proof with probability $p$.

In other words, for each $i$, $E'$ obtains the unique decommitment information for the messages $\{q_i, a_i\}_{i \in [\kappa]}$: note that by the structure of the modified Blum protocol, this corresponds exactly to $\{z_{0,i}\}_{i \in [\kappa]}$. For every index $i \in [\kappa]$ such that $e_i = 1$, it obtains decommitment information $z_{1,i}$ from $\tilde{\text{aux}}$ (this must be present and must verify accurately since $E'$ outputs an accepting transcript). For these indices, $E'$ attempts to compute the witness $H_i = \pi_i^{-1}(H'_i)$ using the permutation $\pi_i$ recovered from $z_{0,i}$ and the Hamiltonian cycle $H'_i$ recovered from $z_{1,i}$. If for any $i \in [\kappa]$, $H_i$ corresponds to a correct witness for the graph $G$, $E$ outputs $H_i$.

If $\tilde{\tau}$ is an accepting transcript (this must be true with probability $p = \text{negl}(\kappa)$ by the indistinguishability of extracted and real transcripts of $\text{extcom}$), and with probability $q = \frac{1}{\text{poly}(\kappa)}$, none of the indices $i \in [\kappa]$ helped extract the witness, then the soundness proof in Lemma 8 implies that $\text{ch'}$ can be guessed using $\tilde{\text{val}}$ and this must equal $e$ with probability at least $p \cdot q - \text{negl}(\kappa)$, which is a contradiction.

5.3 Proofs of Privacy

Lemma 10. The protocol in Figure 8 satisfies statistical zero-knowledge with superpolynomial simulation, according to Definition 7.

Proof. The simulation strategy is straightforward: the simulator obtains $\{q_i, o_1, i\}_{i \in [\kappa]}$ externally. It runs in super-polynomial time to break the receiver message $\text{O}T_{1,i}$ via brute-force to extract $\{e_i\}_{i \in [\kappa]}$. Given $\{e_i\}_{i \in [\kappa]}$, run the honest verifer ZK simulator for modified Blum on input $\{o_i, e_i\}_{i \in [\kappa]}$. Obtain $\{a_i, z_{i,e_i}\}_{i \in [\kappa]}$ from the honest verifier ZK simulator. Finally, it sends for $i \in [\kappa]$, $a_i$ together with $\text{O}T_{2,i}(z_{i,e_i}, z_{i,e_i})$.

Statistical zero-knowledge then follows because of statistical zero knowledge by Lemma 7 and from the statistical security of $\text{O}T$ against unbounded verifiers. We will argue this by a simple hybrid argument, where in the first hybrid, the challenger sends for $i \in [\kappa]$, honestly generated $a_i$ together with $\text{O}T_{2,i}(z_{i,e_i}, z_{i,e_i})$. The verifier’s view in this hybrid is statistically close to honestly generated proofs because of statistical security of $\text{O}T$ against unbounded malicious receivers. The verifier’s view in this intermediate hybrid is also statistically close to the simulator’s output because of statistical hiding of $\text{extcom}$ which ensure that honestly generated $\{a_i\}_{i \in [\kappa]}$ remain statistically indistinguishable from simulated values $\{a_i\}_{i \in [\kappa]}$.

This also yields the following lemma.

Lemma 11. The protocol in Figure 8 satisfies statistical witness indistinguishability against all malicious verifiers.
Proof. (Sketch.) This claim follows by a simple hybrid argument, where in an intermediate hybrid, the challenger generates the proof via the superpolynomial simulator of Lemma 10 (without using any witness). By Lemma 10, this intermediate hybrid is statistically close to any hybrid where a specific witness is used. This proves witness indistinguishability of the protocol. Refer to [BGI+17] for a more detailed proof.

Lemma 12. The protocol in Figure 8 satisfies distributional statistical delayed-input \( \epsilon \)-weak zero-knowledge according to Definition 6.

Proof. We develop an inductive analysis and a simulation strategy that learns the receiver’s challenge bit-by-bit.

Fix any unbounded \( V^* \), any distinguisher \( D \), any distribution \((X,W,Z)\), and any \( \epsilon > 0 \). We construct a simulator \( \text{Sim}_\epsilon \) that generates \( p_\epsilon = \text{poly}(1/\epsilon) \) samples \((x_1^*, w_1^*), (x_2^*, w_2^*), \ldots (x_p^*, w_p^*)\) from the distribution \((X,W)\).

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

We now define our simulator via a sequence of hybrid experiments, where hybrid \( \text{Hybrid}_{\text{Sim}_\epsilon} \) corresponds to our simulator \( \text{Sim}_\epsilon \).

Proof via Hybrid Experiments

\( \text{Hybrid}_0 := \text{Hybrid}_{0,\epsilon} \):
This hybrid corresponds to an honest prover in the real world. That is, for \( i \in [\kappa] \), the challenger samples \((x,w) \xleftarrow{\$} (X,W)\) and sends \( a_i = f_1(q_i, x, w, r_i), z_0^i = f_2(q_i, x, w, r_i, e_{ch}, i = 0), z_1^i = f_2(q_i, x, w, r_i, e_{ch}, i = 1) \) to the verifier.

\( \text{Hybrid}_{I,\epsilon} \) for \( I \in [2, \kappa] \):
This hybrid is indexed by a small error parameter \( \epsilon \), and proceeds as follows. Fix the first message \( r \) of the verifier.

1. Run the algorithm in Figure 9 parameterized by \( I = 1 \) with oracle access to the distinguisher \( D \), and error parameter \( \epsilon \), to obtain guess \( e_{ch,1} \) for the first bit of the verifier challenge.
2. Next, compute \( a_1 = f_1(q_i, x, w, r_i), z_0^i = f_2(q_i, x, w, r_i, e_{ch,1}), z_1^i = f_2(q_i, x, w, r_i, e_{ch,1}) \).
3. For \( i \in [2, \kappa] \), compute \((a_i, z_0^i, z_1^i)\) honestly.
4. Send prover message according to Figure 8 using the \( a_i, z_i \) computed for \( i \in [\kappa] \).

\( \text{Hybrid}_{I,\epsilon} \) for \( I \in [2, \kappa] \):
This hybrid is indexed by a small error parameter \( \epsilon \), and proceeds as follows.

1. Run the algorithm in Figure 9 parameterized by \( I \) with oracle access to the verifier \( V \), distinguisher \( D \), and error parameter \( \epsilon \), to obtain guess \( e_{ch} \) for the first \( I \) bits of the verifier challenge.
2. Next, for \( i \in [I] \), compute \( a_i = f_1(q_i, x, w, r_i), z_0^i = f_2(q_i, x, w, r_i, e_{ch,i}), z_1^i = f_2(q_i, x, w, r_i, e_{ch,i}) \).
3. For \( i \in [I + 1, \kappa] \), compute \((a_i, z_0^i, z_1^i)\) honestly.
4. Send prover message according to Figure 8 using the $a_i, z_i$ computed for $i \in [\kappa]$.

---

**Algorithm $\mathcal{M}^{V,D_V}$ to approximate the verifier's challenge up to the $I^{th}$ bit.**

- Set $p = \kappa^2/e^3$, $i = 1$, $e_{ch} = \bot$. For fixed verifier message $r$,
- While $i \leq I$, repeat:
  - Set $D_0 = 0$ and for $j \in [p]$, repeat:
    1. For $k < i$, sample fresh randomness $r_k$ and set $a_k = f_1(x_j^*, w_j^*, r_k), z_k^0 = z_k^1 = f_2(x_j^*, w_j^*, r_k, e = e_{ch,k})$.
    2. Sample fresh $r_i$, set $a_i = f_1(x_j^*, w_j^*, r_i), z_i^0 = z_i^1 = f_2(x_j^*, w_j^*, a_i = 0, r_i)$.
    3. For $k \in [i+1, \kappa]$, sample fresh randomness $r_k$ and honestly set $a_k = f_1(x_j^*, w_j^*, r_k), z_k^0 = z_k^1 = f_2(x_j^*, w_j^*, a_i = 1, r_k)$.
  - Using $(a, z)$ computed above, send prover message according to Figure 8 together with the instance $x_j^*$.
    - Set $D_0 = D_0 + \frac{1}{p}$ if the output of the distinguisher $D_V = 1$ (w.l.o.g., we assume that the distinguisher $D_V$ outputs either 0 or 1).
  - Set $D_1 = 0$ and for $j \in [p]$, repeat:
    1. For $k < i$, sample fresh randomness $r_k$ and set $a_k = f_1(x_j^*, w_j^*, r_k), z_k^0 = z_k^1 = f_2(x_j^*, w_j^*, r_k, e = e_{ch,k})$.
    2. Sample fresh $r_i$, set $a_i = f_1(x_j^*, w_j^*, r_i), z_i^0 = z_i^1 = f_2(x_j^*, w_j^*, a_i = 1, r_i)$.
    3. For $k \in [i+1, \kappa]$, sample fresh randomness $r_k$ and honestly set $a_k = f_1(x_j^*, w_j^*, r_k), z_k^0 = z_k^1 = f_2(x_j^*, w_j^*, a_i = 1, r_k)$.
  - Using $(a, z)$ computed above, send prover message according to Figure 8 together with the instance $x_j^*$.
    - Set $D_1 = D_1 + \frac{1}{p}$ if the output of the distinguisher $D_V = 1$.
  - Set $D_w = 0$ and for $j \in [p]$, repeat:
    1. For $k < i$, sample fresh randomness $r_k$ and set $a_k = f_1(x_j^*, w_j^*, r_k), z_k^0 = z_k^1 = f_2(x_j^*, w_j^*, r_k, e = e_{ch,k})$.
    2. For $k \in [i, \kappa]$, sample fresh randomness $r_k$ and honestly set $a_k = f_1(x_j^*, w_j^*, r_k), z_k^0 = z_k^1 = f_2(x_j^*, w_j^*, a_i = 0, r_k)$.
    3. Using $(a, z)$ computed above, send prover message according to Figure 8 together with the instance $x_j^*$.
    - Set $D_w = D_w + \frac{1}{p}$ if the output of the distinguisher $D_V = 1$.
    - If $|D_1 - D_w| \leq |D_0 - D_w|$, set $e_{ch, i} = 1$, else set $e_{ch, i} = 0$.
    - Set $i = i + 1$ and go to beginning of the while loop.

- Output $e_{ch}$.

---

**Figure 9:** Approximately Learning the Verifier’s Challenge
Lemma 13. For all $I \in [0, \kappa - 1]$,

$$\Delta(\text{Hybrid}_{I,\epsilon}, \text{Hybrid}_{I+1,\epsilon}) \leq \frac{\epsilon}{\kappa + 1}$$

Proof. The only difference between $\text{Hybrid}_{I,\epsilon}$ and $\text{Hybrid}_{I+1,\epsilon}$ is that in $\text{Hybrid}_{I+1,\epsilon}$, $\epsilon_{ch,I+1}$ is computed according to the algorithm in Figure 9 and the challenger sets $a_{I+1} = f_1(q_i, x, w, r_{I+1}), z_{I+1}^0 = z_{I+1}^1 = f_2(q_i, x, w, r_{I+1}, e_{\text{guess},I+1})$, and then sends prover message according to Figure 8.

For the fixed verifier message $\text{OT}_1$, for $i \in [\kappa]$ and a fixed prefix $e_{\text{prefix}} = \epsilon_{ch,[I]}$, denoting the first $I$ bits of $e_{\text{ch}},$

- Let $D_{e_{\text{prefix}},0,x}$ denote the actual distribution output by the distinguisher when the challenger samples random $(x, w) \overset{\$}{\leftarrow} (\mathcal{X}, \mathcal{W}),$
  - For $j \leq I$, sets $a_j = f_1(x, w, r_j), z_j^0 = z_j^1 = f_2(q_i, x, w, r_j, e_j = e_{\text{prefix},j})$, and using these sends prover message according to Figure 8. Here, $e_{\text{prefix},j}$ denotes the $j^{th}$ bit of $e_{\text{prefix}}$.
  - For $j = I + 1$, sets $a_j = f_1(x, w, r_j), z_j^0 = z_j^1 = f_2(q_i, x, w, r_j, e_j = 0)$, and using these sends prover message according to Figure 8.
  - For $j \in [I+2, \kappa]$, sets $a_j = f_1(x, w, r_j), z_j^0 = f_2(q_i, x, w, r_j, e_j = 0), z_j^1 = f_2(q_i, x, w, r_j, e_j = 1)$, and using these sends prover message according to Figure 8.
  - We will abuse notation and also use $D_{e_{\text{prefix}},0,x}$ to denote the probability that the distinguisher outputs 1 in this situation.

- Let $D_{e_{\text{prefix}},1,x}$ denote the actual distribution output by the distinguisher when the challenger samples random $(x, w) \overset{\$}{\leftarrow} (\mathcal{X}, \mathcal{W})$ and fresh randomness $r,$
  - For $j \leq I$, sets $a_j = f_1(x, w, r_j), z_j^0 = z_j^1 = f_2(q_i, x, w, r_j, e_j = e_{\text{prefix},j})$, and using these sends prover message according to Figure 8.
  - For $j = I + 1$, sets $a_j = f_1(x, w, r_j), z_j^0 = z_j^1 = f_2(q_i, x, w, r_j, e_j = 1)$, and using these sends prover message according to Figure 8.
  - For $j \in [I+2, \kappa]$, sets $a_j = f_1(x, w, r_j), z_j^0 = f_2(q_i, x, w, r_j, e_j = 0), z_j^1 = f_2(q_i, x, w, r_j, e_j = 1), r_j$, and using these sends prover message according to Figure 8.
  - We will abuse notation and also use $D_{e_{\text{prefix}},1,x}$ to denote the probability that the distinguisher outputs 1 in this situation.

- Let $D_{e_{\text{prefix}},w,x}$ denote the actual distribution output by the distinguisher when the challenger samples random $(x, w) \overset{\$}{\leftarrow} (\mathcal{X}, \mathcal{W})$ and fresh randomness $r,$
  - For $j \leq I$, sets $a = f_1(x, w, r_j), z_j^0 = z_j^1 = f_2(q_i, x, w, r_j, e_j = e_{\text{prefix},j})$, and using these sends prover message according to Figure 8.
  - For $j \in [I+1, \kappa]$, sets $a = f_1(x, w, r_j), z_j^0 = f_2(q_i, x, w, r_j, e_j = 0), z_j^1 = f_2(q_i, x, w, r_j, e_j = 1)$, and using these sends prover message according to Figure 8.
  - We will abuse notation and also use $D_{e_{\text{prefix}},w,x}$ to denote the probability that the distinguisher outputs 1 in this situation.

Claim 6. Either of the following statements is true:

- For any prefix $e_{\text{prefix}} \in \{0,1\}^I$, $\Delta(D_{e_{\text{prefix}},0,x}, D_{e_{\text{prefix}},w,x}) \leq \frac{\epsilon}{\kappa + 1}$
For any prefix \( e \text{prefix} \in \{0, 1\}^i \), \( \Delta(D_{\text{prefix}, 1,x}, D_{\text{prefix}, 0,x}) \leq \frac{\epsilon}{k+1} \)

Proof. This claim follows from \( \epsilon \)-statistical security of the OT. Assume, for contradiction, that there exists \( V \) for which the claim is not true. We will use such a verifier to break receiver security of the underlying OT. Consider a reduction \( R \) that obtains the first OT message from \( V \) and forwards this message to the OT challenger.

The reduction picks \( (x, w) \leftarrow (X, W) \), \( r \in \{0, 1\}^k \) and sets \( a_{I+1} = f_1(x, w, r), z_{I+1} = f_2(q_i, x, w, r, e = 0), z_{I+1} = f_2(q_i, x, w, r, e = 1) \), and sends \((z_0, z_1, z_{I+1})\) to the OT challenger.

The OT challenger generates either the real message \( OT_2(z_{I+1}^0, z_{I+1}^1) \) corresponding to verifier input, or a simulated message \( OT_2(z^*, z^*) \), 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 \( \text{Hybrid}_{I} \) and generates sender message accordingly.

Then, the output distribution on input the simulated message is either distributed identically to \( D_{\text{prefix}, 0,x} \) or \( D_{\text{prefix}, 1,x} \) (depending upon whether \( z^* \) is 0 or 1). Since the real and simulated message are \( \frac{\epsilon}{k+1} \)-close, this completes the proof of the claim.

This claim establishes that for any prefix, at least one of the distributions \( D_{\text{prefix}, 0,x} \) and \( D_{\text{prefix}, 1,x} \) is \( \frac{\epsilon}{k+1} \)-close to \( D_{\text{prefix}, w,x} \).

If both \( D_{\text{prefix}, 0,x} \) and \( D_{\text{prefix}, 1,x} \) are \( \epsilon/(k+1) \)-close to \( D_{\text{prefix}, w,x} \), then for any value of \( e_{\text{ch}, I+1} \in \{0, 1\} \), \( \Delta(\text{Hybrid}_{I}, \epsilon_i) \leq \epsilon/(k+1) \) and we are done.

Therefore, for the rest of this lemma, we restrict ourselves to the case where one and only one of \( D_{\text{prefix}, 0,x} \) and \( D_{\text{prefix}, 1,x} \) is \( \frac{\epsilon}{k+1} \)-close to \( D_{\text{prefix}, w,x} \). In particular, this also implies that \( \Delta(D_{\text{prefix}, 0,x} - D_{\text{prefix}, 1,x}) > \frac{\epsilon}{k+1} \).

If the challenger could “magically” set \( e_{\text{ch}, I+1} \) to 0 if \( D_{\text{prefix}, 0,x} \) was close to \( D_{\text{prefix}, w,x} \), and to 1 if \( D_{\text{prefix}, 0,x} \) was close to \( D_{\text{prefix}, w,x} \), then again we would have that

\[
\Pr[D_{V} = 1 | \text{Hybrid}_{I, I+1}] - \Pr[D_{V} = 1 | \text{Hybrid}_{I+1, I}] \leq \epsilon/(k+1)
\]

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 \( e_{\text{ch}, I} \) is incorrect conditioned on \( |D_{\text{prefix}, 0,x} - D_{\text{prefix}, 1,x}| > \frac{\epsilon}{k+1} \), i.e., we show:

Claim 7.

\[
\Pr[(e_{\text{ch}, I} = b) | (\Delta(D_{\text{prefix}, 1,x}, D_{\text{prefix}, 0,x}) > \frac{\epsilon}{k+1}) \land (\Delta(D_{\text{correct}, w}, D_{\text{correct}, b,w}) > \frac{\epsilon}{k+1})] \leq \text{negl}(\kappa)
\]

Proof. We note that for the \( (I + 1) \)th iteration of Figure 9, \( D_0 \) just consists of \( p \) random samples of a distribution with mean \( D_{\text{prefix}, 0,x} \), \( D_1 \) just consists of \( p \) random samples of a distribution with mean \( D_{\text{prefix}, 1,x} \), and \( D_w \) just consists of \( p \) random samples of a distribution with mean \( D_{\text{prefix}, w,x} \).

Then, using a simple Chernoff bound, we have:

\[
\Pr\left[\left(D_0 > D_{\text{prefix}, 0,x}(1 + \alpha)\right) \lor \left(D_0 < D_{\text{prefix}, 0,x}(1 - \alpha)\right) \leq 2 \exp^{-\frac{\alpha^2 \epsilon D_0}{2}}\right]
\]

\[
\Pr\left[\left(D_1 > D_{\text{prefix}, 1,x}(1 + \alpha)\right) \lor \left(D_1 < D_{\text{prefix}, 1,x}(1 - \alpha)\right) \leq 2 \exp^{-\frac{\alpha^2 \epsilon D_1}{2}}\right]
\]

\[
\Pr\left[\left(D_w > D_{\text{prefix}, w,x}(1 + \alpha)\right) \lor \left(D_w < D_{\text{prefix}, w,x}(1 - \alpha)\right) \leq 2 \exp^{-\frac{\alpha^2 \epsilon D_w}{2}}\right]
\]

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

\[
\Pr\left[\left(D_{\text{prefix}, 0,x} - D_0 > \frac{\epsilon}{2\kappa}\right) \lor \left(D_{\text{prefix}, 1,x} - D_1 > \frac{\epsilon}{2\kappa}\right) \lor \left(D_{\text{prefix}, w,x} - D_w > \frac{\epsilon}{2\kappa}\right) \right] \leq 6 \exp^{-\frac{1}{2\kappa}}.
\]
Since $\epsilon$ will always be set to $\frac{1}{\text{poly}(\kappa)}$ for some polynomial $\text{poly}(\cdot)$,

$$\Pr \left[ \left( |D_{\text{prefix},0,x} - D_0| > \frac{\epsilon}{2\kappa} \right) \lor \left( |D_{\text{prefix},1,x} - D_1| > \frac{\epsilon}{2\kappa} \right) \lor \left( |D_{\text{prefix},w,x} - D_w| > \frac{\epsilon}{2\kappa} \right) \right] \leq \text{negl}(\kappa).$$

Recall that one of $D_{\text{prefix},0,x}$ and $D_{\text{prefix},w,x}$ is at least $\epsilon/(\kappa + 1)$-far from $D_{\text{prefix},w,x}$, and the other is at most $\text{negl}(\kappa)$-far. The bit $e_{ch,1}$ is estimated via $D_0, D_1, D_w$ which each have error at most $\frac{\epsilon}{2\kappa}$, from the corresponding $D_{\text{prefix},0,x}, D_{\text{prefix},1,x}, D_{\text{prefix},w,x}$.

Thus, we have:

$$\Pr[(e_{ch,1} = b) \left( |\Delta(D_{\text{prefix},1,x}, D_{\text{prefix},0,x}) > \frac{\epsilon}{\kappa + 1} \right) \land (|\Delta(D_{\text{correct},w}, D_{\text{correct},b,w}) > \frac{\epsilon}{\kappa + 1}) \leq \text{negl}(\kappa)]$$

This completes the proof of the lemma.

$\text{Hybrid}_{\text{Sim},\epsilon}$: This hybrid corresponds to the interaction of the simulator with the verifier and distinguisher. It is indexed by a small error parameter $\epsilon$, and proceeds as follows.

1. Run the algorithm in Figure 9 parameterized by $\kappa$ with oracle access to the verifier $V$, distinguisher $D$, and error parameter $\epsilon$, to obtain guess $e_{ch}$ for the entire verifier challenge (all $\kappa$ bits).

2. Next, for $i \in [\kappa]$, execute the honest verifier ZK simulator of the modified Blum protocol to obtain $\{a_i, z_i\}_{i \in [\kappa]}$ and send prover message according to Figure 8.

Lemma 14. $\Delta(\text{Hybrid}_{k,\epsilon}, \text{Hybrid}_{\text{Sim},\epsilon}) \leq \text{negl}(\kappa)$

Proof. Assume, for contradiction, that there exist $V, D$ for which the lemma is not true. We will use $V, D$ to break honest-verifier statistical zero-knowledge of the underlying modified Blum protocol.

Consider a reduction $R$ that on input $\{q_i, a_i, i\}_{i \in [\kappa]}$ computes $e_{ch}$ using Figure 9. $R$ then sends $\{q_i, e_{ch}\}_{i \in [\kappa]}$ to the honest-verifier ZK challenger. It obtains $(a^*, z^*)$, that is either sampled honestly using the instance $x$ and witness $w$, or sampled using the honest-verifier ZK simulator on $x$.

The reduction $R$ then sends $a^*, \text{OT}_2(z^*, z^*)$ to the distinguisher $D_V$ as the output of the challenger between $\text{Hybrid}_{k,\epsilon}$ and $\text{Hybrid}_{\text{Sim},\epsilon}$. Note that the experiment corresponds to $\text{Hybrid}_{k,\epsilon}$ if $(a^*, z^*)$ is sampled honestly using the instance $x$ and witness $w$, and to $\text{Hybrid}_{\text{Sim},\epsilon}$ if it is sampled using the honest-verifier ZK simulator. Therefore, we have that $\Delta(\text{Hybrid}_{k,\epsilon}, \text{Hybrid}_{\text{Sim},\epsilon}) \leq \text{negl}(\kappa)$.

Therefore, we have the following main theorem.

Theorem 2. Assuming quasi-polynomially secure oblivious transfer according to Definition 2 and Property 1 there exists a two-message argument system that is extractable according to Property 2 satisfies statistical witness indistinguishability (Definition 5), statistical weak distributional $\epsilon$-zero-knowledge for delayed-input statements (Definition 7), and statistical zero-knowledge with super-polynomial simulation (Definition 6).
We also observe that our arguments can be made resettable statistical witness indistinguishable by applying \cite{BGGL01}.

6 Oblivious Transfer: Stronger Security and Reversal

6.1 Simulation-Secure Two-Message Oblivious Transfer

We first construct an oblivious transfer protocol with unbounded simulation-based security against both malicious receivers and malicious senders. We define this variant below.

**Definition 10** (Simulation-Secure Oblivious Transfer). As in Definition 2, we let \((S(m_0, m_1), R(b))\) denote an execution of the OT protocol with sender input \((m_0, m_1)\) and receiver input bit \(b\). We consider OT that satisfies the following properties (which are both defined using real-ideal security definitions):

- **Computational Receiver Security.** There exists a \(T_{\text{Sim}}\)-time oracle-aided simulator \(\text{Sim}^{S^*}\) that interacts with any non-uniform malicious PPT sender \(S^*\) and outputs \(\text{View}(\text{Sim}^{S^*})\). It also extracts and sends \(S^*\)'s inputs \(m_0, m_1\) to an ideal functionality \(\mathcal{F}_{\text{ot}}\), which obtains choice bit \(b\) from the honest receiver \(R\) and outputs \(\text{Output}_{\text{ideal}} = m_b\) to \(R\). Then, we require that for every non-uniform PPT \(S^*\), the joint distributions \((\text{View}(\text{Sim}^{S^*}), \text{Output}_{\text{ideal}})\) and \((\text{View}_{\text{Sps}}(S^*, R(b)), \text{Output}_{\text{Rsps}}(S^*, R(b)))\) are computationally indistinguishable.

- **Statistical Sender Security.** There exists a (possibly unbounded) oracle-aided simulator \(\text{Sim}^{R^*}\) that interacts with any unbounded adversarial receiver \(R^*\), and with an ideal functionality \(\mathcal{F}_{\text{ot}}\) on behalf of \(R^*\). Here \(\mathcal{F}_{\text{ot}}\) is an oracle that obtains the inputs \((m_0, m_1)\) from \(S\) and \(b\) from \(\text{Sim}^{R^*}\) (simulating the malicious receiver), and outputs \(m_b\) to \(\text{Sim}^{R^*}\). Then we require that for all \(m_0, m_1\), \(\text{Sim}^{R^*}\) outputs a receiver view that is statistically indistinguishable from the real view of the malicious receiver \(\text{View}_{\text{Rsps}}((S(m_0, m_1, z), R^*))\).

Our construction of two-message OT satisfying Definition 10 is described in Figure 10. It uses a two-message OT scheme according to Definition 2 and satisfying Property 1, whose messages are denoted by \(\text{OT}_1\) and \(\text{OT}_2\). It also uses a statistical SPS zero-knowledge according to \(\text{stat-sps-zk}\) according to Definition 7, whose messages are denoted by \(\text{stat-sps-zk}_1\) and \(\text{stat-sps-zk}_2\).

**Lemma 15.** The protocol in Figure 10 satisfies receiver security according to Definition 10.

**Proof.** Security against non-uniform PPT malicious senders according to Definition 10 can be proven based on the hiding security of OT against malicious senders according to Definition 2 and from the extractability of the statistical SPS-ZK according to Definition 7 and Property 2. The simulator \(\text{Sim}^{S^*}\) for Definition 10 is described in Figure 11.

We now prove that the joint distribution \((\text{View}(\text{Sim}^{S^*}), \text{Output}_{\text{ideal}})\) is indistinguishable from the real distribution, \((\text{View}_{\text{Sps}}(S^*, R(b)), \text{Output}_{\text{Rsps}}(S^*, R(b)))\).

In order to prove this, we consider an intermediate hybrid \(\text{Hybrid}_1\) where the challenger behaves the same way as the simulator, except that it generates and sends \(m_R\) the same way as an honest receiver, that is \(m_R = \text{OT}_1(b, r_R)\) for honest receiver input \(b\). However, it uses the extractor for the statistical SPS-ZK to obtain \((m_0, m_1)\) and send them to the ideal functionality. The output of \(\text{Hybrid}_1\) is the view of \(S^*\) interacting with the above challenger, together with \(m_b\) output by the ideal functionality. By the extractability of the argument \(\text{Property 2}\) and correctness of the extracted values, this joint distribution is indistinguishable from the joint distribution \((\text{View}_{\text{Sps}}(S^*, R(b)), \text{Output}_{\text{Rsps}}(S^*, R(b)))\).
Sender Input: Message bits $x_0, x_1$. Receiver Input: Choice bit $b$.

- **Receiver Message.**
  - Sample $r_R \leftarrow \{0, 1\}^*$ and send $m_R = \text{OT}_1(b; r_R)$.
  - Sample and send $\text{stat-sps-zk}_1$.

- **Sender Message.**
  - Send $m_S = \text{OT}_2(m_R, x_0, x_1; r_S)$.
  - Send $\text{stat-sps-zk}_2$ proving that $\exists x_0, x_1, r_S$ such that $m_S = \text{OT}_2(m_R, x_0, x_1; r_S)$.

- **Receiver Output.**
  - If $\text{stat-sps-zk}$ does not verify, output $\bot$ and abort.
  - Else obtain output $a$ of the two-message OT using $(m_S, r_R)$. Output $a$.

---

Figure 10: Simulation Secure Oblivious Transfer

The joint distribution output in Hybrid$_1$ is indistinguishable from the simulated distribution denoted by $(\text{View}((\text{Sim}^{S^*}), \text{Output}_{\text{Ideal}}))$, because $m_R$ hides $b$, by **Definition 2**. More formally, given $S^*$ and any distinguisher $D$ that distinguishes the output of the intermediate hybrid from the simulated distribution, we consider a reduction that uses $S^*$ and $D$ to break security of OT according to **Definition 2**. This reduction obtains the first message $m_R$ externally to either correspond to $\text{OT}_1(b; r_R)$ or $\text{OT}_1(0; r_R)$. It uses this message as the first message and runs the rest of the simulation strategy. Then the output distribution corresponds to either Hybrid$_1$ or $(\text{View}((\text{Sim}^{S^*}), \text{Output}_{\text{Ideal}}))$, proving that $D$ breaks OT security according to **Definition 2**, which is a contradiction. \(\square\)

**Lemma 16.** The protocol in Figure 10 satisfies statistical sender security according to **Definition 10**.

**Proof.** Security against unbounded malicious receivers according to **Definition 10** follows from the perfect security of the underlying oblivious transfer protocol OT against malicious receivers according to **Definition 2** and from the statistical SPS-ZK security of $\text{stat-sps-zk}$.

We now describe the simulator $\text{Sim}_{R^*}$ that interacts with any malicious receiver generating first message $m_{R^*}$. It parses $m_{R^*}$ as $m_{a, R^*}, m_{b, R^*}$, which are respectively the first messages of the underlying OT and of the statistical SPS-ZK. Next, it executes the unbounded simulator $\text{Sim}_{\text{OT}}$ of the underlying OT protocol against malicious receivers, and uses this to generate the sender message $m_S$. It also executes the $T_{\text{Sim}_{\text{zk}}}$-time simulator of the statistical SPS-ZK protocol, and uses this to generate the sender message of the statistical SPS-ZK.

To prove that the simulated transcript is indistinguishable from the real transcript, we consider an intermediate hybrid, where in the first hybrid, we consider a challenger that generates sender message $m_S$ using honest sender strategy, and executes $T_{\text{Sim}_{\text{zk}}}$-time simulator of the statistical SPS-ZK protocol to generate the sender message of the statistical SPS-ZK. The view of $R^*$ in this intermediate hybrid is statistically close to the view of $R^*$ interacting with an honest sender, by the security of statistical SPS-ZK. The view of $R^*$ in this intermediate hybrid is statistically close to the view of $R^*$ interacting with $\text{Sim}_{R^*}$ described above, by statistical simulation security of the underlying OT against malicious receivers. \(\square\)
1. Sample uniform randomness $r_R$, and compute $m_R = \text{OT}_1(0; r_R)$.

2. Construct prover $P^*$ using $S^*$ as follows:
   - On input receiver challenge $\text{stat-sps-zk}_1$, send $(m_R, \text{stat-sps-zk}_1)$ to $S^*$.
   - Obtain response $\text{stat-sps-zk}_2, m_S$ from $S^*$.
   - Send $\text{stat-sps-zk}_1, \text{stat-sps-zk}_2$ as proof transcript for the statement: $\exists x_0, x_1, r_S$ such that $m_S = \text{OT}_2(m_R, x_0, x_1; r_S)$. Send $m_R, m_S$ as auxiliary information.

3. Run the extractor for the statistical statistical SPS-ZK on $P^*$ to obtain transcript $\tilde{m}_R, \tilde{m}_S, \text{stat-sps-zk}_1, \text{stat-sps-zk}_2$, together with extracted witness $x_0, x_1$.

4. Output transcript $\tilde{m}_R, \tilde{m}_S, \text{stat-sps-zk}_1, \text{stat-sps-zk}_2$.

5. Send $x_0, x_1$ to $F_{\text{ot}}$.

Figure 11: Simulation Strategy Against Malicious Senders

6.2 Reversing Oblivious Transfer

We first construct an oblivious transfer protocol with unbounded simulation-based security against both malicious receivers and malicious senders. We define this variant below.

**Definition 11 (Simulation-Secure Oblivious Transfer Against Unbounded Senders).** As in **Definition 2**, we let $(S(m_0, m_1), R(b))$ denote an execution of the OT protocol with sender input $(m_0, m_1)$ and receiver input bit $b$. We consider OT that satisfies the following properties (which are both defined using real-ideal security definitions):

- **Computational Sender Security.** There exists a $T_{\text{Sim}}$-time oracle-aided simulator $\text{Sim}^{S^*}$ that interacts with any non-uniform malicious PPT receiver $R^*$ and interacts with the ideal functionality $F_{\text{ot}}$ on behalf of any malicious receiver $R^*$ Here $F_{\text{ot}}$ is an oracle that obtains the inputs $(m_0, m_1)$ from $S$ and $b$ from $\text{Sim}^{R^*}$ (simulating the malicious receiver), and outputs $m_b$ to $\text{Sim}^{R^*}$. Then we require that for all $m_0, m_1, \text{Sim}^{R^*}$ outputs a receiver view that is computationally indistinguishable from the real view of the malicious receiver $\text{View}_{R^*}((S(m_0, m_1), R^*))$.

- **Statistical Receiver Security.** There exists a (possibly unbounded) oracle-aided simulator $\text{Sim}^{S^*}$ that interacts with any unbounded adversarial sender $S^*$, and with an ideal functionality $F_{\text{ot}}$ on behalf of $S^*$. Here $F_{\text{ot}}$ is an oracle that obtains the inputs $(m_0, m_1)$ from $\text{Sim}^{S^*}$ and $b$ from $R$ and outputs $\text{Output}_{\text{ideal}} = m_b$ to $R$. Then, we require that for every unbounded $S^*$, the joint distributions $(\text{View}(\text{Sim}^{S^*}), \text{Output}_{\text{ideal}})$ and $(\text{View}_{S^*}(S^*, R(b)), \text{Output}_{S^*}(S^*, R(b)))$ are statistically indistinguishable.

We now describe a three-message (bit) oblivious transfer protocol with simulation-based security against malicious receivers and unbounded malicious senders, according to **Definition 11**.

This is obtained by reversing a two-message (bit) oblivious transfer protocol with simulation security against unbounded malicious receivers and PPT malicious senders, according to **Definition 10**, constructed in **Figure 10**. Let $\text{OT}_R(b; r_R)$ denote the receiver message of such an oblivious transfer protocol computed as a function of input bit $b$ and randomness $r_R$, and let $\text{OT}_S(m_R, x_0, x_1; r_S)$...
denote the sender message of such a protocol computed as a function of receiver message $m_R$, sender inputs $x_0, x_1$ and randomness $r_S$. Our protocol is described in Figure 12.

<table>
<thead>
<tr>
<th>Sender Input:</th>
<th>Message bits $x_0, x_1$.</th>
<th>Receiver Input:</th>
<th>Choice bit $b$.</th>
</tr>
</thead>
<tbody>
<tr>
<td>o Sender Message.</td>
<td>Sample $x'_0, x'_1 \leftarrow {0,1}^2$ and $r_S$ uniformly at random. Set $c = x'_0 \oplus x'_1$, and send $m_S = \text{OT}_R(c; r_S)$.</td>
<td>o Receiver Message.</td>
<td>- Sample input (single-bit) messages $m_0, m_1$ uniformly at random such that $m_0 \oplus m_1 = b$. - Send $m_R = \text{OT}_S(m_0, m_1; r_R)$.</td>
</tr>
<tr>
<td>o Receiver Output:</td>
<td>The receiver outputs $y = (z \oplus z_b \oplus m_0)$.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 12: Oblivious Transfer Reversal

Completeness It follows from inspection of the protocol (refer also [WW06]) that on sender input $m_0, m_1$ and receiver input $b$, the receiver obtains output $m_b$.

Lemma 17. The protocol in Figure 12 satisfies sender security according to Definition 11

Proof. (Sketch) The simulator $\text{Sim}^{R^*}$ runs the simulator $\text{Sim}^{S^*}$ of the underlying OT protocol to generate the first message of the protocol, to extract $m_0, m_1$ from the second message, sent by the malicious receiver $R^*$. It computes $b = m_0 \oplus m_1$ and sends this to the ideal functionality. On obtaining $x_b$ from the ideal functionality, it computes $z, z_0, z_1$ uniformly at random subject to $m_b = (z \oplus z_b \oplus m_0)$. It sends $z, z_0, z_1$ to $R^*$ on behalf of the honest sender. It follows by receiver security of the underlying OT, that $\text{Sim}^{S^*}$ performs correct extraction, and generates an indistinguishable view. Therefore, since $\text{Sim}^{R^*}$ generates the third message to be identically distributed as the real execution, $\text{Sim}^{R^*}$ also generates an indistinguishable view.

Lemma 18. The protocol in Figure 12 satisfies statistical receiver security according to Definition 11

Proof. (Sketch) The simulator $\text{Sim}^{S^*}$ runs the simulator $\text{Sim}^{R^*}$ of the underlying OT protocol to extract $c$ from the first message of the sender, generate the second message of the protocol, and obtain $m_c$. It sets $a = m_c$. Next, given the third message of the protocol generated by $S^*$, it sets $x_0 = z_0 \oplus z \oplus a$, $x_1 = z_1 \oplus c \oplus x_0 \oplus z_0$, and sends these to the ideal functionality. It follows by statistical sender security of the underlying OT, that the underlying simulator $\text{Sim}^{S^*}$ correctly extracts sender inputs, and therefore $\text{Sim}^{S^*}$ generates an indistinguishable view.

References

[ABOR00] William Aiello, Sandeep N. Bhatt, Rafail Ostrovsky, and Sivaramakrishnan Rajagopalan. Fast verification of any remote procedure call: Short witness-


[KRR17] Yael Tauman Kalai, Guy N. Rothblum, and Ron D. Rothblum. From obfuscation to the security of fiat-shamir for proofs. In *Advances in Cryptology - CRYPTO 2017 -


