Statistical Witness Indistinguishability (and more) in Two Messages

Yael Tauman Kalai[1], Dakshita Khurana[2(✉)], and Amit Sahai[2]

[1] Microsoft Research, Cambridge, USA
yaelism@gmail.com
[2] Department of Computer Science, UCLA, Los Angeles, USA
{dakshita,sahai}@cs.ucla.edu

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 simulator is a probabilistic polynomial time machine with oracle access to the distinguisher, and 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), which in turn can be based on quasi-polynomial hardness of DDH or QR or N^{th} residuosity. We also show how such protocols can be used to build 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.

1 Introduction

Witness indistinguishable (WI) protocols [16] 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

J. B. Nielsen and V. Rijmen (Eds.): EUROCRYPT 2018, LNCS 10822, pp. 34–65, 2018.
https://doi.org/10.1007/978-3-319-78372-7_2

the rounds of interaction needed to build WI protocols. Indeed, in an influential work, Dwork and Naor [14] introduced WI protocols that only require two messages to be exchanged between the prover and verifier, and these were further derandomized to non-interactive protocols by [6]. 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, [4,8,21,23] achieved two message or non-interactive WI protocols from other assumptions, namely assumptions on 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, and 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 two messages to be exchanged? This is the natural analog of one of the earliest questions studied in the context of zero-knowledge protocols: Are statistical zero-knowledge arguments [10] 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 arguments for NP, based on standard cryptographic hardness assumptions against quasi-polynomial time adversaries (such as quasi-poly hardness of DDH, or Quadratic Residuosity, or N'th Residuosity). 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 Protocols. The techniques we use to build two-message SWI also allow us to achieve other forms of statistical privacy.

One of the most popular notions of privacy in proof systems is that of zero-knowledge. This is usually formalized via simulation, by showing the existence of a *polynomial-time* simulator that simulates the view of any polynomial size (malicious) verifier. At an intuitive level, the existence of such a simulator means that any information that a polynomial size verifier learns from an honest prover,

he could have generated on his own (in and indistinguishable manner), without access to such a prover. It is known [20] that zero-knowledge is impossible to achieve in just two messages. However, other weaker variants have been shown to be achievable in this setting.

Pass [29] was the first to construct a two-message argument with quasi-polynomial time simulation. In his work, the simulated proofs were indistinguishable by distinguishers running in time significantly smaller than that of the (uniform) simulator. Very recently, [27] constructed the first two-message arguments for NP achieving super-polynomial *strong* simulation, where the simulated proofs remain indistinguishable by distinguishers running in time significantly larger than that of the (uniform) simulator. These capture the intuition that for any information that a quasi-polynomial size verifier learns from an honest prover, indistinguishable information could have been generated by the verifier in a similar amount of time.

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 arguments satisfying this property in two messages.[1] This improves upon the work of [27] by pushing their privacy guarantees all the way to statistical.

We note that in all these arguments, the simulator works by breaking soundness of the proof, so all of the above two-message arguments are only sound against provers running in time less than that of the simulator.

Recently, [23] showed that this caveat could be overcome, by weakening the ZK requirement. Specifically, they constructed two-message arguments in the delayed-input distributional setting, with distinguisher-dependent polynomial-time simulation. These protocols only satisfy *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 in the delayed-input distributional setting with distinguisher-dependent simulation, where the simulator runs in polynomial time with oracle access to the distinguisher, and achieving *statistical* privacy.

Our Core Technique. Our key technique consists of compressing an interactive protocol into a two-message protocol. Specifically, we start with an interactive argument satisfying honest-verifier *statistical zero-knowledge*, and compress it into a two-message argument by proving soundness of the [25] heuristic, which builds on [7]. Actually, to obtain a two-message protocol with statistically privacy, it does not suffice to start with an honest-verifier statistical ZK protocol, but rather we need the ZK property to hold against semi-malicious verifiers.[2] We gloss over this detail in this high-level overview.

[1] Achieving such two-message arguments was believed to be impossible [12], however the work of [27] showed that the line of impossibility claims [12] for super-polynomial simulation was surmountable.

[2] A semi-malicious verifier is one who follows the prescribed algorithm but with possibly malicious randomness.

This heuristic is believed to be *insecure* when applied generally to interactive arguments (as opposed to proofs). Nevertheless, we construct a family of 4-message interactive arguments with statistical hiding guarantees, and prove that the [25]-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 (and only computationally sound), but with negligible probability, they are statistically sound. Crucially, we show that a (computationally bounded) prover 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 sound (which happens very rarely). At the heart of our construction is a new special commitment scheme, which build upon and significantly extend commitment schemes from [27]. We then show how to leverage this rare statistical soundness event, to allow the soundness of the the [25]-heuristic to kick in.

This rare event helps us achieve other extraction properties that we require in our applications. We elaborate on this below in our technical overview, providing a detailed but still informal overview of our techniques and results. Our protocols are based on standard cryptographic hardness assumptions with security against quasi-polynomial time adversaries (such as the quasi-poly hardness of DDH, or Quadratic Residuosity, or N'th Residuosity).

New Oblivious Transfer Protocols. Our techniques also have applicability to an intriguing question about oblivious transfer (OT): The works of Naor and Pinkas [28] and Aiello et al. [2] introduced influential two-message protocols for OT achieving a game-based notion of security, which offers 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 the choice bit $b = 1$. Now this receiver would be able to recover both inputs of the honest sender by using both random values r_0 and r_1 on the sender's response message, violating OT security against a (bounded) malicious receiver.

Again remarkably, this basic question, of constructing a 3-message OT protocol with security against unbounded sender, has been open since the works of [2,28] 17 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 (same assumptions as before). Such an OT protocol can also be plugged into the constructions of [23] to achieve

three-message *proofs* for NP (as opposed to arguments) achieving delayed-input distributional weak ZK, witness hiding and strong witness indistinguishability.

Our techniques also 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 two-message OT protocols of [2,28] do not rule out selective failure attacks. For example, 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 two-message OT protocols, however, only super-polynomial simulation is possible, and this was recently formally established in [3] 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 cannot be simultaneously achieved.

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.

1.1 Summary of 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 [28], or based on the quasi-poly hardness of QR or the N'th residuosity assumption [22,24]. We first construct a two-message argument for NP with the following statistical hiding guarantees:

1. Our two-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 two-message argument is statistical zero-knowledge with super-polynomial time simulation.[3]
3. Our two-message argument is statistical weak zero-knowledge in the delayed input setting where the simulator has oracle access to the distinguisher, and where the instance is sampled from some distribution after the verifier sent the first message.

We also obtain the following results on oblivious transfer:

1. We construct a three-message OT protocol simultaneously satisfying super-polynomial simulation security, and security against a computationally unbounded sender.

[3] We note that prior to this work, this was believed to be impossible to achieve via black-box reductions [12].

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 [18]. However, no two-message statistical WI arguments were known for all of NP.

Two main approaches for reducing rounds in interactive proof systems have appeared in the literature. The first is due to Fiat and Shamir [17], and the second is due to [25] and is based on the [7]-heuristic for converting multi-prover interactive proofs to two-message arguments. The [25]-heuristic is sound when applied to a *statistically sound* interactive proof, assuming the existence of a super-polynomial OT (or super-polynomially secure computational PIR) scheme. Very recently, [11,26] showed that the Fiat-Shamir heuristic is also sound when applied to a *statistically sound* interactive proof, assuming the existence of a symmetric encryption scheme where the key cannot be recovered even with *exponentially* small probability (even after seeing encryptions of key-dependent messages).[4]

The works of [3,23] 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 zero-knowledge against semi-malicious verifiers, into a two-message protocol, while keeping (and even improving) the secrecy guarantees. However, these works do not yield statistical privacy, which is the focus of the present work. More specifically, these works apply the [25]-heuristic to 3-message public-coin proofs that are zero-knowledge against semi-malicious verifiers, to obtain their resulting two-message protocols. We note that since they start with a statistically sound proof they obtain only *computational hiding* guarantees, and after applying the [25]-heuristic, their resulting two-message protocols are only *computationally sound* (in addition to being only computational hiding).

In contrast, in this work we construct two-message arguments with *statistical hiding* guarantees. More specifically, we do this by constructing a 4-message interactive argument with statistical hiding guarantees, and converting it into a two-message computationally sound protocol by applying the [25]-heuristic to it.

2 Overview of Techniques

Our starting point is the [25]-heuristic, which shows how to compress public coin interactive *proofs* into two-message arguments. We note that this heuristic is based on the heuristic introduced in [7] (and explored in [1]), which converts

[4] Their actual assumption is a bit more complex and we refer to [11] for details.

multi-prover interactive proofs into two-message arguments. We note that the [25]-heuristic is only known to be sound when applies to interactive proofs (and believed not to be sound when applied to general interactive arguments).

Recently, [3,23] proved that this heuristic also preserves (and even enhances) privacy. Our strategy will be to follow this blueprint, but in the statistical setting. This becomes quite tricky in the statistical setting because we do not have interactive *proofs* for NP with statistical privacy guarantees. In particular, we do not have an interactive *proof* for NP which is statistical zero-knowledge against semi-malicious verifiers (which is the privacy guarantee needed in [3,23], but in the computational setting).

However, we do have an interactive *argument* which is statistical zero-knowledge against semi-malicious verifiers. We construct such an interactive argument of a specific form, and prove that the [25]-heuristic is sound when applied to this interactive argument.

We begin by reviewing the techniques from [3,23], where we take as a running example the Blum protocol for Graph Hamiltonicity, which is known to be (computational) zero-knowledge against semi-malicious verifiers.

2.1 First Attempt: Compressing the Blum Protocol via OT

In what follows, we recall the two-message protocol from [3,23] (with computational privacy guarantees), which makes use of the following two components:

- A three-message proof for Graph Hamiltonicity, due to Blum [9]. 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 protocol with a single-bit challenge (i.e., where $e_i \in \{0,1\}$). We also denote by f_1 and f_2 the functions that satisfy $a_i = f_1(x, w; r_i)$ and $z_i = f_2(x, w, r_i, e_i)$, for answers provided by the honest prover, and where r_i is uniformly chosen randomness.
- Any two-message oblivious transfer protocol, denoted by $(\mathsf{OT}_1, \mathsf{OT}_2)$, which is secure against malicious PPT receivers, and malicious senders running in time at most $2^{|z|}$. For receiver input b and sender input messages (M^0, M^1), we denote the two messages of the OT protocol as $\mathsf{OT}_1(b)$ and $\mathsf{OT}_2(M^0, M^1)$. We note that $\mathsf{OT}_2(M^0, M^1)$ also depends on the message $\mathsf{OT}_1(b)$ sent by the receiver. For the sake of simplicity, we omit this dependence from the notation.

Given these components, the two-message protocol $\langle P, V\rangle$ (from [3,23]) is described in Fig. 1.

Soundness. It was proven in [3,23,25] 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.

Preliminary Two-Message Protocol from [24,3]

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

Fig. 1. Preliminary two-message protocol

Can We Achieve Statistical Privacy Against Malicious Verifiers? Let us now analyze the privacy of the protocol in Fig. 1. The work of [3,23] showed that the protocol in Fig. 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 zero-knowledge property of the underlying Blum proof, when restricted to semi-malicious verifiers.

As we already described, the focus of this paper is achieving statistical privacy. To this end, we take a closer look at the Blum protocol.

Background. Recall that in the (parallel repetition of the) Blum protocol, for each index $i \in [\kappa]$, a_i consists of a statistically binding commitment to a random permutation π and the permuted graph $\pi(G)$, 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 to $(\pi, \pi(G))$, 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 and only if the revealed edges are indeed a Hamiltonian Cycle.

In an quest for statistical privacy, we notice the following properties about the protocol in Fig. 1:

1. A single parallel repetition of the underlying Blum proof only satisfies computational 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 π, and therefore obtain the witness (i.e., the Hamiltonian cycle) from any honest prover.
2. The underlying OT protocols [22,28] used in the protocol of Fig. 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 Fig. 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*.

To this end, 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 the Blum protocol 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 zero-knowledge* against semi-malicious verifiers.

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 $\{q_i, \mathsf{OT}_{1,i}(e_i)\}_{i\in[\kappa]}$, and the prover responds with $\{a_i, \mathsf{OT}_{2,i}(z_i^{(0)}, z_i^{(1)})\}_{i\in[\kappa]}$. In this case, because of the *statistical* hiding of the commitments and the *statistical* sender security of OT, the proof in [3,23] can be easily extended to achieve *statistical* witness indistinguishability.

One may now hope that the analysis in [3,23,25] can be used to prove that the resulting protocol also remains *sound* against PPT provers. Unfortunately, as we noted above, the proof of soundness [3,23,25] crucially relies on the fact that the starting protocol is a *proof* (as opposed to an argument). More specifically, the soundness proof in previous works goes through as follows: Consider for simplicity the case of a single repetition, and suppose a cheating prover, on input the verifier message $\mathsf{OT}_1(e^*)$, outputs $x^* \notin L$, together with a message $(a^*, \mathsf{OT}_2(z^*))$, such that the verifier accepts with probability $\frac{1}{2} + \frac{1}{\mathsf{poly}(\kappa)}$. Intuitively, since for any $x^* \notin L$ and any a^*, there exists at most one unique value of receiver challenge e^*, for which there exists a z^* that causes the verifier to accept, this means that a^* consists of a commitment that *encodes* the receiver challenge e^*. By using an OT scheme that is secure against adversaries that can break the commitment within a^*, a cheating prover can be used to contradict receiver security of OT. This proves that a single parallel execution of the protocol in Fig. 1 has soundness $\frac{1}{2} + \mathsf{negl}(\kappa)$. The same argument can be generalized to prove that no adaptive PPT prover P^* can cheat with non-negligible probability when we perform κ parallel repetitions. More specifically, the reduction can use any prover that cheats with non-negligible probability to guess the κ-bit challenge e with non-negligible probability, contradicting the security of κ parallel repetitions of OT.

This proof crucially relies on the fact that the commitment is statistically binding. This is no longer true for the four-message modified version of the Blum protocol described above. In fact, the problem runs deeper: Note that what we

seem to need for this approach to work is a *proof* that satisfies *statistical ZK* against semi-malicious verifiers, however, such proofs are unlikely to exist for all of NP (see, e.g. [30]). Therefore, the only remaining option, if we follow this approach, is to find a way to compress some form of statistical 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, even 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 ZK against semi-malicious verifiers, we will instead construct an *argument* that satisfies statistical ZK against semi-malicious verifiers. 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 [25] 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.

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 [27].

They construct an extractable *computationally hiding* commitment scheme, which is completely insecure against unbounded malicious receivers. The underlying idea behind their work, which we will share, is the following: In their commitment scheme, with a negligible probability, 2^{-m} for $m = \Omega(\log \kappa)$, the message being committed to is transmitted to the receiver. Otherwise, with overwhelming probability $1 - 2^{-m}$, the receiver obtains an actual (statistically-binding) commitment to the message. Crucially, the committer does not know which case occurs – whether its message was transmitted to the receiver or not. In this way, their commitment can be seen as an unusually noisy *erasure channel*.

Committer Input: Message $M \in \{0,1\}^p$, where $p = \mathsf{poly}(\kappa)$.
Commit Stage:
Receiver Message.

- Pick challenge string $\mathsf{ch} \xleftarrow{\$} \{0,1\}$.
- Compute and send the first OT message $\mathsf{OT}_1(\mathsf{ch}, r_1)$ using uniform randomness r_1.

Committer Message.

- Sample a random string $r \xleftarrow{\$} \{0,1\}$. Set $M^r = M, M^{1-r} \xleftarrow{\$} \{0,1\}^p$.
- Compute $o_2 = \mathsf{OT}_2(M^0, M^1; r_2)$ 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. $o_2 = \mathsf{OT}_2(M^0, M^1; r_2)$,
2. $M^r = M$.

Fig. 2. Basic construction of a two-message statistically hiding commitment

Our commitment will work to achieve the same goal, but crucially we will seek to achieve a statistically hiding commitment.

The reason why the work of [27] was inherently limited to achieving only computational hiding is because of the way they implement the erasure channel described above: In their work, this was implemented using a two-message secure computation protocol, that implemented a coin-flipping procedure to provide the randomness underlying the erasure channel. Such two-message secure computation protocols only achieve computational hiding. Therefore, in our work, we must depart fundamentally from this method of implementing the erasure channel.

Basic Construction. In order to obtain a construction that essentially implements the erasure channel described above, 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 (Fig. 2) implemented using game-based oblivious transfer [2,22,24,28], 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., $r \neq \mathsf{ch}$ 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 $\widetilde{M}_1 \neq \widetilde{M}_2$, 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[5]. A formal analysis can be found in the full version of the paper.

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 Fig. 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 $r \neq \mathsf{ch}$, which happens with probability 2^{-m}).
- Most importantly, because of receiver security of the OT, no malicious PPT committer can distinguish the case where $r = \mathsf{ch}$ from the case where $r \neq \mathsf{ch}$.[6]

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 Fig. 3 to generate messages $\{q_i, a_i\}_{i\in[\kappa]}$, with $m = \Omega(\log \kappa)$. This is formally described in Sect. 5.1. By statistical hiding of the commitment, the resulting protocol is a statistical ZK argument. On the other hand, by the extractability of the commitment, (more specifically in the case where $r = \mathsf{ch}$), the protocol, in fact, becomes a proof. Furthermore, no cheating PPT prover can distinguish the case when $r = \mathsf{ch}$ from when $r \neq \mathsf{ch}$. 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 $x \notin L$ with noticeable probability when $r \neq \mathsf{ch}$, must continue to do so even when $r \neq \mathsf{ch}$. We will now analyze the soundness of the resulting protocol.

Arguing Soundness of the Compressed Protocol. We show that the resulting protocol remains sound against cheating PPT provers. While we also achieve

[5] We 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.

[6] 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 ch all the time.

Extraction parameter: m.
Committer Input: Message $M \in \{0,1\}^p$.
Commit Stage:
Receiver Message.

- Pick challenge string $\mathsf{ch} \xleftarrow{\$} \{0,1\}^m$.
- Sample uniform randomness $\{r_{1,i}\}_{i\in[m]}$.
- Compute and send $\{\mathsf{OT}_1(\mathsf{ch}_i, r_{1,i})\}_{i\in[m]}$ using m instances of two-message OT.

Committer Message.

- Sample a random string $r \xleftarrow{\$} \{0,1\}^m$.
  For every $i \in [m]$ and every $b \in \{0,1\}$, sample $M_i^b \xleftarrow{\$} \{0,1\}^p$ subject to $\bigoplus_{i\in[m]} M_i^{r_i} = M$.
- For every $i \in [m]$ compute $o_{2,i} = \mathsf{OT}_2(M_i^0, M_i^1; r_{2,i})$ 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 $i \in [m]$, $o_{2,i} = \mathsf{OT}_2(M_i^0, M_i^1; r_{2,i})$,
2. $\bigoplus_{i\in[m]} M_i^{r_i} = M$.

Fig. 3. Our extractable commitments

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 $x \notin L$ together with an accepting proof with probability $p = \frac{1}{\mathsf{poly}(\kappa)}$, cannot distinguish the case when $r = \mathsf{ch}$ from the case when $r \neq \mathsf{ch}$ by the property of the extractable commitment. Moreover, such a prover must continue to generate accepting transcripts for $x \notin L$ with probability at least $\frac{1}{\mathsf{poly}(\kappa)}$ even in case $r = \mathsf{ch}$[7]. Although the event $r = \mathsf{ch}$ only occurs with negligible probability, we use the extractor of extcom to amplify this probability by making many queries to the prover. The extractor then outputs a transcript of the proof (corresponding to $r = \mathsf{ch}$), together with the values committed in all messages corresponding to the extractable commitment. This requires the oblivious trans-

[7] Ensuring this requires the decommit phase of the extractable commitment to be publicly verifiable, without the 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.

fer 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 Sect. 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 [2,22,24,28], 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: [32] 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 Fig. 4).

If we did manage to somehow reverse the two-message OT protocols of [2,22,24,28] using such a reversal, then clearly we would obtain a three-message protocol with game-based security against unbounded senders and malicious PPT receivers. However, surprisingly, proving game-based security of the protocol obtained by reversing [2,22,24,28] 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

Sender Input: Message bits x_0, x_1. **Receiver Input:** Choice bit b.

- **Sender Message.** Sample $x'_0, x'_1 \xleftarrow{\$} \{0,1\}^2$ and r_S uniformly at random. Set $c = x'_0 \oplus x'_1$, and send $m_S = \mathsf{OT}_R(c; r_S)$.
- **Receiver Message.**
  - Sample input (single-bit) messages m_0, m_1 uniformly at random such that $m_0 \oplus m_1 = b$.
  - Send $m_R = \mathsf{OT}_S(m_0, m_1; r_R)$.
- **Sender Message.**
  - Obtain output a of the two-message OT using (m_R, r_S).
  - Send $z = a \oplus x'_0, z_0 = x'_0 \oplus x_0, z_1 = x'_1 \oplus x_1$.
- **Receiver Output:** The receiver outputs $y = (z \oplus z_b \oplus m_0)$.

Fig. 4. Oblivious transfer reversal

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 Fig. 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[8].

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 Fig. 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 superpolynomially 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[9]. 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.

[8] We note that existing two-message protocols [2,22,24,28] with security against unbounded receivers do not satisfy simulation-based security against malicious senders.

[9] This 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}.

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 allowing approximately 2^m-time simulation. Please refer to Sect. 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 [29] which gave the first constructions of computational zero-knowledge with superpolynomial time simulation.

In this paper, building on the recent work of [27], 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 κ to denote the security parameter, and $\mathsf{negl}(\kappa)$ to denote any function that is asymptotically smaller than $\frac{1}{\mathsf{poly}(\kappa)}$ for any polynomial $\mathsf{poly}(\cdot)$.

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

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

We say that two families of distributions $D_1 = \{D_{1,\kappa}\}, D_2 = \{D_{2,\kappa}\}$ are statistically indistinguishable if $\Delta(D_{1,\kappa}, D_{2,\kappa}) = \mathsf{negl}(\kappa)$. 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 $\mathcal{D}$,

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

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

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^κ. We omit this from the notation for the sake of brevity.

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

- **Completeness:** *For all $(x,w) \in R_L$, $\Pr[\mathsf{Output}_V\langle P(x,w), V(x)\rangle = 1] = 1 - \mathsf{negl}(\kappa)$, assuming P and V follow the protocol honestly.*
- **Special Soundness:** *There exists a polynomial-time algorithm A that given any x and a pair of accepting transcripts $(a,e,z),(a,e',z')$ for x with the same first prover message, where $e \neq e'$, outputs w such that $(x,w) \in R_L$.*
- **Semi-malicious verifier zero-knowledge:** *There exists a probabilistic polynomial time simulator $\mathcal{S}_\Sigma$ such that for all $(x,w) \in R_L$, the distributions $\{\mathcal{S}_\Sigma(x,e)\}$ and $\{\mathsf{View}_V\langle P(x,w(x)), V(x,e)\rangle\}$ are statistically indistinguishable, where $\mathcal{S}_\Sigma(x,e)$ denotes the output of simulator $\mathcal{S}$ upon receiving input x and the verifier's random tape, denoted by 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 a 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 $\langle S(m_0,m_1), R(b)\rangle$ denote an execution of the OT protocol with sender input (m_0,m_1) and receiver input bit b. 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 $\mathsf{View}_{S^*}(\langle S^*, R(b)\rangle)$ and $\mathsf{View}_{S^*}(\langle S^*, R(b')\rangle)$ are computationally indistinguishable.*
  We say that the OT scheme is T-secure if any $\mathsf{poly}(T)$-size malicious sender S^* has a distinguishing advantage less than $\frac{1}{poly(T)}$.
- $(1-\delta)$-**Statistical Sender Security.** *For any receiver R^* that outputs receiver message m_{R^*}, there exists bit b such that for all m_0, m_1, the distribution $\mathsf{View}_{R^*}\langle S(m_0,m_1), R^*\rangle$ is $(1-\delta)$ statistically close to $\mathsf{View}_{R^*}\langle S(m_b,m_b), R^*\rangle$.*

Such two-message protocols have been constructed based on the DDH assumption [28], 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 [22,24]. Such two-message protocols can also be based on witness encryption or indistinguishability obfuscation (iO) together with one-way permutations [31].

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 (cheating) provers, and arguments – with soundness against computationally bounded (cheating) 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 $(x,w) \in R_L$,*

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

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^κ 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,*

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

where the probability is over the random coins of V.

Witness Indistinguishability. A proof system is witness indistinguishable if for any statement with at least two witnesses, proofs computed using different witnesses are indistinguishable. 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 $n = n(\kappa) \leq \mathsf{poly}(\kappa)$, 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:*

$$\left\{\mathsf{View}_{V^*}\langle P(x_n, w_{1,n}), V^*(x_n)\rangle\right\} \text{ and } \left\{\mathsf{View}_{V^*}\langle P(x_n, w_{2,n}), V^*(x_n)\rangle\right\}$$

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 [15,19], 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 ϵ-WZK, which says that for every $\mathcal{T}_{\mathcal{D}}$-time distinguisher $\mathcal{D}$ and every distinguishing advantage ϵ (think of ϵ as an inverse polynomial) there exists a simulator, that is an oracle machine running in time $\mathsf{poly}(\kappa, 1/\epsilon)$ with oracle access to the distinguisher, that generates a view that $\mathcal{D}$ cannot distinguish from the view generated by the real prover. This notion was previously considered in [13,15,23].

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 ϵ-WZK.

Definition 6 (Delayed-Input Distributional Statistical ϵ-Weak Zero Knowledge). *A delayed-input interactive argument (P,V) for a language L is said to be delayed-input distributional statistical* ϵ-weak zero knowledge *if for every polynomially bounded function $n = n(\kappa) \leq \mathsf{poly}(\kappa)$, and for every efficiently samplable distribution $(\mathcal{X}_\kappa, \mathcal{W}_\kappa)$ on R_L, i.e., $\mathsf{Supp}(\mathcal{X}_\kappa, \mathcal{W}_\kappa) = \{(x,w) \in R_L : x \in \{0,1\}^{n(\kappa)}\}$, every unbounded verifier V^* that obtains the instance from*

the prover in the last message of the protocol, every unbounded distinguisher $\mathcal{D}$, and every ϵ (which will usually be set to $1/\mathsf{poly}(\kappa)$ for some polynomial $\mathsf{poly}(\cdot)$), there exists a simulator $\mathcal{S}$ that runs in time $\mathsf{poly}(\kappa, 1/\epsilon)$ and has oracle access to $\mathcal{D}$ and V^*, such that:

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

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

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

Zero-Knowledge with Super-Polynomial Simulation. We now define zero-knowledge with super-polynomial simulation in the same way as [29], except that we define *statistical* security against malicious verifiers.

Definition 7 (Statistical ZK with Super-polynomial Simulation). *We say that a delayed input two message argument (P, V) for an NP language L is statistical zero-knowledge with super-polynomial T_{Sim}-time simulation, if there exists a (uniform) simulator $\mathcal{S}$ that runs in time T_{Sim}, such that for every polynomial $n = n(\kappa) \leq \mathsf{poly}(\kappa)$, and for every $(x_n, w_n) \in R_L$ where each $|x_n| = n$, and every unbounded verifier V^*, the two distributions $\mathcal{S}^{V^*}(x_n)$ and $\mathsf{View}_{V^*}\langle P(x_n, w_n), V^*(x_n)\rangle$ are statistically close.*

4 Extractable Commitments

4.1 Definitions

Our notion of extractable commitments tailors the definition in [27] to the setting of statistically hiding commitments. We begin by (re-)defining the notion of a commitment scheme. As before, we use κ to denote the security parameter, and we let $p = \mathsf{poly}(\kappa)$ 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 protocols.

Definition 8 *[Statistically Hiding Commitment Scheme]. A commitment $\langle \mathcal{C}, \mathcal{R}\rangle$ is a two-phase protocol between a committer $\mathcal{C}$ and receiver $\mathcal{R}$, consisting of a tuple of algorithms*

$$\mathsf{Commit}, \mathsf{Decommit}, \mathsf{Verify}.$$

At the beginning of the protocol, $\mathcal{C}$ obtains as input a message $M \in \{0,1\}^p$. Next, $\mathcal{C}$ and $\mathcal{R}$ execute the commit phase, and obtain a commitment transcript, denoted by τ, together with a private state for $\mathcal{C}$, denoted by $\mathsf{state}_{\mathcal{C},\tau}$. We use the notation

$$(\tau, \mathsf{state}_{\mathcal{C},\tau}) \leftarrow \mathsf{Commit}\langle \mathcal{C}(M), \mathcal{R}\rangle.$$

Later, $\mathcal{C}$ and $\mathcal{R}$ possibly engage in a decommit phase, where the committer $\mathcal{C}$ computes and sends message $y = \mathsf{Decommit}(\tau, \mathsf{state}_{\mathcal{C},\tau})$ to $\mathcal{R}$. At the end, $\mathcal{R}$ computes $\mathsf{Verify}(\tau, y)$ to output $\perp$ or a message $\widetilde{M} \in \{0,1\}^p$.[10]

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

- **(Perfect) Completeness.** *If $\mathcal{C}, \mathcal{R}$ honestly follow the protocol, then for every $M \in \{0,1\}^p$:*

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

  where the probability is over $(\tau, \mathsf{state}_{\mathcal{C},\tau}) \leftarrow \mathsf{Commit}\langle \mathcal{C}(M), \mathcal{R}\rangle$.
- **Statistical Hiding.** *For every two messages $M_1, M_2 \in \{0,1\}^{2p}$, every unbounded malicious receiver $\mathcal{R}^*$ and honest committer $\mathcal{C}$, a commitment is $\delta(\kappa)$-statistically hiding if the statistical distance between the distributions $\mathsf{View}_{\mathcal{R}^*}(\mathsf{Commit}\langle \mathcal{C}(M_1), \mathcal{R}^*\rangle)$ and $\mathsf{View}_{\mathcal{R}^*}(\mathsf{Commit}\langle \mathcal{C}(M_2), \mathcal{R}^*\rangle)$ is at most $\delta(\kappa)$. The scheme is statistically hiding if $\delta(\kappa) \leq \frac{1}{\mathsf{poly}(\kappa)}$ for every polynomial $\mathsf{poly}(\cdot)$.*
- **Computational Binding.** *Consider any non-uniform PPT committer $\mathcal{C}^*$ that produces $\tau \leftarrow \mathsf{Commit}\langle \mathcal{C}^*, \mathcal{R}\rangle$, and then outputs y_1, y_2. Let $\widetilde{M}_1 = \mathsf{Verify}(\tau, y_1)$ and $\widetilde{M}_2 = \mathsf{Verify}(\tau, y_2)$. Then, we require that*

$$\Pr\left[(\widetilde{M}_1 \neq \perp) \ \wedge \ (\widetilde{M}_2 \neq \perp) \ \wedge \ (\widetilde{M}_1 \neq \widetilde{M}_2)\right] = \mathsf{negl}(\kappa),$$

  over the randomness of sampling $\tau \leftarrow \mathsf{Commit}\langle \mathcal{C}^*, \mathcal{R}\rangle$.

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

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

Definition 9 *[$\mathcal{T}$-Extractable Commitment Scheme]. We say that a statistically hiding commitment scheme is $\mathcal{T}$-extractable if there exists a $\mathcal{T} \cdot \mathsf{poly}(\kappa)$-time*

[10] We note that in our definition, $\mathcal{R}$ does not need to keep a state from the commitment phase in order to execute the decommitment phase.

uniform oracle machine $\mathcal{E}$ such that the following holds. Let $\mathcal{C}^*$ 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* $\mathsf{Samp}^{\mathcal{C}^*}$ *that samples* $(\tau_{\mathcal{C}^*}, \mathsf{aux}) \leftarrow \mathsf{Commit}\langle \mathcal{C}^*, \mathcal{R}\rangle$. *Let* $\mathsf{Exp}_{\mathsf{Samp}^{\mathcal{C}^*}} = (\tau_{\mathcal{C}^*}, \mathsf{aux})$ *be the output of* $\mathsf{Samp}^{\mathcal{C}^*}$.
- $\mathcal{E}^{\mathcal{C}^*}$ *outputs* $(\tau_{\mathcal{C}^*}, \mathsf{aux}, \widetilde{M})$, *while only making oracle calls to* $\mathcal{C}^*$ *during the commit phase (without ever running the decommit phase). We denote by* $\mathsf{Exp}_{\mathcal{E}^{\mathcal{C}^*}} = (\tau_{\mathcal{C}^*}, \mathsf{aux})$.

We require that:

- **Indistinguishability.** *The distributions* $(\mathsf{Exp}_{\mathsf{Samp}^{\mathcal{C}^*}}, z)$ *and* $(\mathsf{Exp}_{\mathcal{E}^{\mathcal{C}^*}}, z)$ *are computationally indistinguishable.*
- **Correctness of Extraction.** *Consider any non-uniform PPT $\mathcal{C}^*$ and let $(\tau, \mathsf{aux}, \widetilde{M})$ denote the output of $\mathcal{E}^{\mathcal{C}^*}$. Then for any string y_1, denoting $\widetilde{M}_1 = \mathsf{Verify}(\tau, y_1)$,*

$$\Pr\left[(\widetilde{M} \neq \bot) \ \wedge \ (\widetilde{M}_1 \neq \bot) \ \wedge \ (\widetilde{M} \neq \widetilde{M}_1)\right] = \mathsf{negl}(\kappa),$$

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

4.2 Protocol

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

Let $\mathsf{OT} = (\mathsf{OT}_1, \mathsf{OT}_2)$ denote a two-message string oblivious transfer protocol according to Definition 2. Let $\mathsf{OT}_1(b; r_1)$ denote the first message of the OT protocol with receiver input b and randomness r_1, and let $\mathsf{OT}_2(M_0, M_1; r_2)$ denote the second message of the OT protocol with sender input strings M_0, M_1 and randomness r_2.[11]

In the full version of this paper, we 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, $(1 - \delta_{\mathsf{OT}})$ secure against malicious receivers according to Definition 2, the scheme in Fig. 5 is a $(1 - 2^m - \delta_{\mathsf{OT}})$ statistically hiding, T-extractable commitment scheme according to Definition 9.*

We prove this theorem by showing statistical hiding, computational binding, and extractability. The proof of statistical hiding follows by $(1 - \delta)$-statistical sender security of the OT. To prove computational binding, we build a reduction to the receiver security of OT according to Definition 2. The proof of extractability follows by building.

Extraction parameter: m.[a]
Committer Input: Message $M \in \{0,1\}^p$.
Commit Stage:
Receiver Message.

- Pick challenge string $\mathsf{ch} \xleftarrow{\$} \{0,1\}^m$.
- Sample uniform randomness $\{r_{1,i}\}_{i\in[m]}$.
- Compute and send $\{\mathsf{OT}_1(\mathsf{ch}_i, r_{1,i})\}_{i\in[m]}$ using m instances of two-message OT.

Committer Message.

- Sample a random string $r \xleftarrow{\$} \{0,1\}^m$.
  For every $i \in [m]$ and every $b \in \{0,1\}$, sample $M_i^b \xleftarrow{\$} \{0,1\}^p$ subject to $\bigoplus_{i\in[m]} M_i^{r_i} = M$.
- For every $i \in [m]$ compute $o_{2,i} = \mathsf{OT}_2(M_i^0, M_i^1; r_{2,i})$ 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 $i \in [m]$, $o_{2,i} = \mathsf{OT}_2(M_i^0, M_i^1; r_{2,i})$,
2. $\bigoplus_{i\in[m]} M_i^{r_i} = M$.

[a] 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 $1 - 2^{-m} - \delta_{\mathsf{OT}}$, when the underlying OT has statistical sender security $1 - \delta_{\mathsf{OT}}$.

Fig. 5. Extractable commitments

$\mathcal{E}^{\mathcal{C}^*}$ 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 $\perp$. We will call each iteration a *trial*.

1. Choose $\mathsf{ch} \xleftarrow{\$} \{0,1\}^m$. Compute $\tau_1 = \mathsf{OT}_1(\mathsf{ch}_i, R_i)$ using uniform randomness $R = \{R_i\}_{i\in[m]}$.
2. Query the oracle $\mathcal{C}^*$ in the Commit phase with τ_1, and obtain response (τ_2, aux), where τ_2 also contains r. If $\mathcal{C}^*$ aborts or sends an invalid message, do the following.
   - If this is the first iteration, output $(\tau_1, \tau_2, \mathsf{aux}, \perp)$ and stop.
   - If this is not the first iteration, go to Step 1 and start a new trial.
3. Else, $\mathcal{C}^*$ did not abort. If $r \neq \mathsf{ch}$, go to Step 1 and start a new trial.
4. Else, $\mathcal{C}^*$ did not abort and $r = \mathsf{ch}$ (this iteration is considered a success). Then use R to obtain $\{M_i^{\mathsf{ch}_i}\}_{i\in[m]}$. Next, compute $\widetilde{M} = \bigoplus_{i\in[m]} \{\widetilde{M}_i^{\mathsf{ch}_i}\}_{i\in[m]}$. Output $(R, \tau_1, \tau_2, \mathsf{aux}, \widetilde{M})$.

Fig. 6. Description of the extractor $\mathcal{E}^{\mathcal{C}^*}$

We build the following extractor $\mathcal{E}$ for Definition 9, in Fig. 6. In the figure, we denote the first message of transcript τ by τ_1 and the second message by τ_2. $\mathcal{E}$ will obtain oracle access to $\mathcal{C}^*$, and the running time of $\mathcal{E}^{\mathcal{C}^*}$ will be $T = 2^m \cdot \kappa^{\log \kappa}$.

The analysis of the extractor builds on the analysis of [27], and can be found in the full version of the paper.

5 Two-Message Arguments with Statistical Privacy

5.1 Modified Blum Protocol

We begin by describing a very simple modification to the Blum Σ-protocol for Graph Hamiltonicity. The protocol we describe will have soundness error $\frac{1}{2} - \mathsf{negl}(\kappa)$ 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 Fig. 7.

We give an overview of the protocol here. Note that the only modification to the original protocol of Blum [9] 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 [9], replacing statistically binding commitments with statistically hiding commitments.

In the full version of the paper, we prove that the protocol in Fig. 7 satisfies soundness against PPT provers that may choose x adaptively in the second round of the protocol. We also prove that assuming that extcom is statistically hiding, the protocol in Fig. 7 satisfies statistical zero-knowledge.

5.2 Compressing Four Message Argument to a Two Message Argument

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

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

Let $\Sigma = (q, a, e, z)$ denote the four messages of the modified Blum protocol from Fig. 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.

[11] Note that OT_2 also depends on OT_1. We omit this dependence in our notation for brevity.

Modified Blum Argument

1. **Verifier Message:** The verifier does the following:
   - Send the first message $\mathsf{extcom}_{1,i,j}$ for independent instances of the extractable commitment, where $i, j \in [p(\kappa)] \times [p(\kappa)]$.
   - Send an additional first message $\mathsf{extcom}_{1,P}$ 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 π on $p(\kappa)$ nodes, and compute $c_P = \mathsf{extcom}_{2,P}(\pi)$ as a commitment to π using extcom.
   - Compute $\pi(G)$, which is the adjacency matrix corresponding to the graph G when its nodes are permuted according to π. Compute $c_{i,j} = \mathsf{extcom}_{2,i,j}(\pi(G)[i][j])$ 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 $c \xleftarrow{\$} \{0,1\}$ to the prover.
4. **Prover Message:** The prover does the following:
   - If $c = 0$, send π and the decommitments of $\mathsf{extcom}_P, \mathsf{extcom}_{i,j}$ for $(i,j) \in [p(\kappa)] \times [p(\kappa)]$.
   - If $c = 1$, send the decommitment of $\mathsf{extcom}_{i,j}$ 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 extcom openings were accepted and $\pi(G)$ was computed correctly by applying π on G.
   - If $c = 1$, accept if and only if all extcom openings were accepted and all the opened commitments form a Hamiltonian cycle.

Fig. 7. Modified blum SZK argument

We state our main lemma here, which we prove in the full version of the paper.

Lemma 1. *Assuming that* extcom *is a* $2^m \cdot \kappa^{\log \kappa}$*-extractable commitment scheme according to Definition 9 and that* OT *is* $2^{\kappa m} \cdot \kappa^{\log \kappa}$*-secure, the protocol in Fig. 8 satisfies soundness against PPT malicious provers.*

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

Remark 2. Our proof also generalizes to executing only $\Omega(\log \kappa)$ parallel executions of the Blum protocol, while still yeilding negligible soundness error. Furthermore, we will see that statistical privacy guarantees will hold even when $m = \Omega(\log \kappa)$. Therefore, the protocol in Fig. 8 can be realized only relying on quasi-polynomially secure oblivious transfer according to Definition 2.

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 $\{o_{1,i} = \mathsf{OT}_{1,i}(e_i)\}_{i\in[\kappa]}$.
  - 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 $\{z_i^0\}_{i\in[\kappa]}$ according to the strategy in Figure 7, using (q_i, a_i, e_i') corresponding to verifier challenge bit $e_i' = 0$.
  - Compute $\{z_i^1\}_{i\in[\kappa]}$ according to the strategy in Figure 7, using (q_i, a_i, e_i') and corresponding to verifier challenge bit $e_i' = 1$.
  - Compute $o_{2,i} = \mathsf{OT}_{2,i}(z_i^0, z_i^1)$ and send $\{a_i, o_{2,i}\}_{i\in[\kappa]}$.
- **Verifier Output:** The verifier V recovers z_i as the output of $\mathsf{OT}_{1,i}, \mathsf{OT}_{2,i}$ 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.

Fig. 8. Two message argument system for NP

Similar to the extractability of commitments, we also define an additional property of two-message arguments, that we call extractability. Roughly, this property requires the existence of a super-polynomial time uniform oracle machine $\mathcal{E}$ that extracts the witness used by any prover generating accepting proofs. It is somewhat more subtle to define, and we refer the reader to the full version for a formal definition. This property is useful in our applications to obtaining stronger forms of OT, and we believe will also be useful for other future applications. We show that the scheme in Fig. 8 is also extractable, where the extractor for the argument can extract a transcript with a witness, from any prover, by relying the extractor of the commitment scheme extcom.

5.3 Proofs of Privacy

Lemma 2. *The protocol in Fig. 8 satisfies statistical zero-knowledge with super-polynomial 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 OT_1 via brute-force to extract $\{e_i\}_{i\in[\kappa]}$. Given $\{e_i\}_{i\in[\kappa]}$, it runs the semi malicious verifier ZK simulator for modified Blum on input $\{a_i, e_i\}_{i\in[\kappa]}$. It obtains $\{a_i, z_{i,e_i}\}_{i\in[\kappa]}$ from the semi malicious verifier ZK simulator. Finally, it sends for $i \in [\kappa]$, a_i together with $\mathsf{OT}_{2,i}(z_{i,e_i}, z_{i,e_i})$.

Statistical zero-knowledge then follows because of statistical zero knowledge of the underlying four-message protocol, and from the statistical security of OT against unbounded verifiers.

This also yields the following lemma.

Lemma 3. *The protocol in Fig. 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 2 (without using any witness). By Lemma 2, this intermediate hybrids is statistically close to any hybrid where a specific witness is used. This proves witness indistinguishability of the protocol. Refer to [3] for a more detailed proof.

Lemma 4. *The protocol in Fig. 8 satisfies distributional statistical delayed-input ϵ-weak zero-knowledge according to Definition 6.*

Following [23], we develop an inductive analysis and a simulation strategy that learns the receiver's challenge bit-by-bit. The proof follows the strategy in [23], and can be found in the full version of the paper.

Therefore, we have the following main theorem.

Theorem 2. *Assuming quasi-polynomially secure oblivious transfer according to Definition 2, there exists a two-message argument system that satisfies statistical witness indistinguishability (Definition 5), statististical zero-knowledge with super-polynomial simulation (Definition 6), and statistical weak distributional ϵ-zero-knowledge for delayed-input statements (Definition 7).*

We also observe that all our two-message arguments can be made resettable statistical witness indistinguishable by applying [5].

6 Oblivious Transfer: Stronger Security and Reversal

In this section, we build OT protocols, in the two-message and three-message setting, that satisfy stronger security properties than previously known. Because of space restrictions, we only describe the protocols and defer proofs to the full version of the paper.

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 $\langle S(m_0, m_1), R(b)\rangle$ denote an execution of the OT protocol with sender input (m_0, m_1) and receiver input bit b. We consider OT that satisfies the following properties (which are both defined using simulation-based security definitions):*

- **Computational Receiver Security.** *There exists a* T_{Sim}*-time oracle-aided simulator* Sim^{S^*} *that interacts with any non-uniform malicious PPT sender* S^* *and outputs* $\mathsf{View}(\mathsf{Sim}^{S^*})$*. It also extracts and sends* S^**'s inputs* m_0, m_1 *to an ideal functionality* $\mathcal{F}_{\mathsf{ot}}$*, which obtains choice bit* b *from the honest receiver* R *and outputs* $\mathsf{Output}_{\mathsf{Ideal}} = m_b$ *to* R*. Then, we require that for every non-uniform PPT* S^**, the joint distributions* $(\mathsf{View}(\mathsf{Sim}^{S^*}), \mathsf{Output}_{\mathsf{Ideal}})$ *and* $(\mathsf{View}_{S^*}\langle S^*, R(b)\rangle, \mathsf{Output}_R\langle S^*, R(b)\rangle)$ *are computationally indistinguishable.*
- **Statistical Sender Security.** *There exists a (possibly unbounded) oracle-aided simulator* Sim^{R^*} *that interacts with any unbounded* adversarial *receiver* R^**, and with an ideal functionality* $\mathcal{F}_{\mathsf{ot}}$ *on behalf of* R^**. Here* $\mathcal{F}_{\mathsf{ot}}$ *is an oracle that obtains the inputs* (m_0, m_1) *from* S *and* b *from* Sim^{R^*} *(simulating the malicious receiver), and outputs* m_b *to* Sim^{R^*}*. Then we require that for all* m_0, m_1*,* Sim^{R^*} *outputs a receiver view that is statistically indistinguishable from the real view of the malicious receiver* $\mathsf{View}_{R^*}\langle S(m_0, m_1, z), R^*\rangle$*.*

Our construction of two-message OT satisfying Definition 10 is described in Fig. 9. It uses a two-message OT scheme according to Definition 2, whose messages are denoted by OT_1 and OT_2. It also uses a statistical SPS zero-knowledge $\mathsf{stat\text{-}sps\text{-}zk}$ according to Definition 7, whose first and second messages are denoted by $\mathsf{stat\text{-}sps\text{-}zk}_1$ and $\mathsf{stat\text{-}sps\text{-}zk}_2$.

Sender Input: Message bits x_0, x_1. **Receiver Input:** Choice bit b.

- **Receiver Message.**
  - Sample $r_R \xleftarrow{\$} \{0,1\}^*$ and send $m_R = \mathsf{OT}_1(b; r_R)$.
  - Sample and send $\mathsf{stat\text{-}sps\text{-}zk}_1$.
- **Sender Message.**
  - Send $m_S = \mathsf{OT}_2(m_R, x_0, x_1; r_S)$.
  - Send $\mathsf{stat\text{-}sps\text{-}zk}_2$ proving that $\exists(x_0, x_1, r_S)$ such that $m_S = \mathsf{OT}_2(m_R, x_0, x_1; r_S)$.
- **Receiver Output.**
  - If $\mathsf{stat\text{-}sps\text{-}zk}$ does not verify, output $\perp$ and abort.
  - Else obtain output a of the two-message OT using (m_S, r_R). Output a.

Fig. 9. Simulation secure oblivious transfer

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* $\langle S(m_0, m_1), R(b)\rangle$ *denote an*

execution of the OT protocol with sender input (m_0, m_1) *and receiver input bit* b*. We consider OT that satisfies the following properties (which are both defined using real-ideal security definitions):*

- **Computational Sender Security.** *There exists an oracle-aided simulator* Sim^{R^*} *that interacts with any non-uniform malicious PPT receiver* R^* *and interacts with the ideal functionality* $\mathcal{F}_{\mathsf{ot}}$ *on behalf of* R^**. Here* $\mathcal{F}_{\mathsf{ot}}$ *is an oracle that obtains the inputs* (m_0, m_1) *from* S *and* b *from* Sim^{R^*} *(simulating the malicious receiver), and outputs* m_b *to* Sim^{R^*}*. Then we require that for all* m_0, m_1*,* Sim^{R^*} *outputs a receiver view that is computationally indistinguishable from the real view of the malicious receiver* $\mathsf{View}_{R^*}(\langle S(m_0, m_1, z), R^* \rangle)$*.*
- **Statistical Receiver Security.** *There exists a (possibly unbounded) oracle-aided simulator* Sim^{S^*} *that interacts with any unbounded* adversarial *sender* S^**, and with an ideal functionality* $\mathcal{F}_{\mathsf{ot}}$ *on behalf of* S^**. Here* $\mathcal{F}_{\mathsf{ot}}$ *is an oracle that obtains the inputs* (m_0, m_1) *from* Sim^{S^*} *and* b *from* R *and outputs* $\mathsf{Output}_{\mathsf{Ideal}} = m_b$ *to* R*. Then, we require that for every unbounded* S^**, the two joint distributions* $(\mathsf{View}(\mathsf{Sim}^{S^*}), \mathsf{Output}_{\mathsf{Ideal}})$ *and* $(\mathsf{View}_{S^*}\langle S^*, R(b)\rangle, \mathsf{Output}_{S^*}\langle S^*, R(b)\rangle)$ *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 Fig. 9. Let $\mathsf{OT}_R(b; r_R)$ denote the receiver message of such an oblivious transfer protocol computed as a function of input bit b and randomness r_R, and let $\mathsf{OT}_S(m_R, x_0, x_1; r_S)$ 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 Fig. 10.

Sender Input: Message bits x_0, x_1. **Receiver Input:** Choice bit b.

- **Sender Message.** Sample $x'_0, x'_1 \xleftarrow{\$} \{0,1\}^2$ and r_S uniformly at random. Set $c = x'_0 \oplus x'_1$, and send $m_S = \mathsf{OT}_R(c; r_S)$.
- **Receiver Message.**
  - Sample input (single-bit) messages m_0, m_1 uniformly at random such that $m_0 \oplus m_1 = b$.
  - Send $m_R = \mathsf{OT}_S(m_0, m_1; r_R)$.
- **Sender Message.**
  - Obtain output a of the two-message OT using (m_R, r_S).
  - Send $z = a \oplus x'_0, z_0 = x'_0 \oplus x_0, z_1 = x'_1 \oplus x_1$.
- **Receiver Output:** The receiver outputs $y = (z \oplus z_b \oplus m_0)$.

Fig. 10. Oblivious transfer reversal

Acknowledgements. Research of D. Khurana and A. Sahai supported in part from a UCLA Dissertation Year Fellowship, a DARPA/ARL SAFEWARE award, NSF Frontier Award 1413955, and NSF grant 1619348, a Xerox Faculty Research Award, a Google Faculty Research Award, an equipment grant from Intel, and an Okawa Foundation Research Grant. This material is based upon work supported by the Defense Advanced Research Projects Agency through the ARL under Contract W911NF-15-C-0205. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense, the National Science Foundation, or the U.S. Government.

References

1. Aiello, W., Bhatt, S., Ostrovsky, R., Rajagopalan, S.R.: Fast verification of any remote procedure call: short witness-indistinguishable one-round proofs for NP. In: Montanari, U., Rolim, J.D.P., Welzl, E. (eds.) ICALP 2000. LNCS, vol. 1853, pp. 463–474. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-45022-X_39
2. Aiello, B., Ishai, Y., Reingold, O.: Priced oblivious transfer: how to sell digital goods. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 119–135. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44987-6_8
3. Badrinarayanan, S., Garg, S., Ishai, Y., Sahai, A., Wadia, A.: Two-message witness indistinguishability and secure computation in the plain model from new assumptions. IACR Cryptology ePrint Archive 2017, 433 (2017). http://eprint.iacr.org/2017/433
4. Badrinarayanan, S., Goyal, V., Jain, A., Khurana, D., Sahai, A.: Round optimal concurrent MPC via strong simulation. In: Kalai, Y., Reyzin, L. (eds.) TCC 2017, Part I. LNCS, vol. 10677, pp. 743–775. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70500-2_25
5. Barak, B., Goldreich, O., Goldwasser, S., Lindell, Y.: Resettably-sound zero-knowledge and its applications. In: 42nd Annual Symposium on Foundations of Computer Science, FOCS 2001, Las Vegas, Nevada, USA, 14–17 October 2001, pp. 116–125 (2001). https://doi.org/10.1109/SFCS.2001.959886
6. Barak, B., Ong, S.J., Vadhan, S.: Derandomization in cryptography. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 299–315. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-45146-4_18
7. Biehl, I., Meyer, B., Wetzel, S.: Ensuring the integrity of agent-based computations by short proofs. In: Rothermel, K., Hohl, F. (eds.) MA 1998. LNCS, vol. 1477, pp. 183–194. Springer, Heidelberg (1998). https://doi.org/10.1007/BFb0057658
8. Bitansky, N., Paneth, O.: ZAPs and non-interactive witness indistinguishability from indistinguishability obfuscation. In: Dodis, Y., Nielsen, J.B. (eds.) TCC 2015, Part II. LNCS, vol. 9015, pp. 401–427. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46497-7_16
9. Blum, M.: How to prove a theorem so no one else can claim it. In: Proceedings of the International Congress of Mathematicians, Berkeley, CA, pp. 1444–1451 (1986)
10. Brassard, G., Chaum, D., Crépeau, C.: Minimum disclosure proofs of knowledge. J. Comput. Syst. Sci. **37**(2), 156–189 (1988)
11. Canetti, R., Chen, Y., Reyzin, L., Rothblum, R.D.: Fiat-Shamir and correlation intractability from strong KDM-secure encryption. Cryptology ePrint Archive, Report 2018/131 (2018). https://eprint.iacr.org/2018/131
12. Chung, K.M., Lui, E., Mahmoody, M., Pass, R.: Unprovable security of two-message zero knowledge. IACR Cryptology ePrint Archive 2012, 711 (2012)

13. Chung, K.-M., Lui, E., Pass, R.: From weak to strong zero-knowledge and applications. In: Dodis, Y., Nielsen, J.B. (eds.) TCC 2015, Part I. LNCS, vol. 9014, pp. 66–92. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46494-6_4
14. Dwork, C., Naor, M.: Zaps and their applications. In: 41st Annual Symposium on Foundations of Computer Science, FOCS 2000, Redondo Beach, California, USA, 12–14 November 2000, pp. 283–293 (2000)
15. Dwork, C., Naor, M., Reingold, O., Stockmeyer, L.J.: Magic functions. In: 40th Annual Symposium on Foundations of Computer Science, FOCS 1999, New York, NY, USA, 17–18 October 1999, pp. 523–534 (1999)
16. Feige, U., Shamir, A.: Witness indistinguishable and witness hiding protocols. In: Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, Baltimore, Maryland, USA, 13–17 May 1990, pp. 416–426 (1990)
17. Fiat, A., Shamir, A.: How to prove yourself: practical solutions to identification and signature problems. In: Odlyzko, A.M. (ed.) CRYPTO 1986. LNCS, vol. 263, pp. 186–194. Springer, Heidelberg (1987). https://doi.org/10.1007/3-540-47721-7_12
18. Garg, S., Ostrovsky, R., Visconti, I., Wadia, A.: Resettable statistical zero knowledge. In: Cramer, R. (ed.) TCC 2012. LNCS, vol. 7194, pp. 494–511. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-28914-9_28
19. Goldreich, O.: A uniform-complexity treatment of encryption and zero-knowledge. J. Cryptology **6**(1), 21–53 (1993)
20. Goldreich, O., Oren, Y.: Definitions and properties of zero-knowledge proof systems. J. Cryptology **7**(1), 1–32 (1994)
21. Groth, J., Ostrovsky, R., Sahai, A.: Non-interactive zaps and new techniques for NIZK. In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 97–111. Springer, Heidelberg (2006). https://doi.org/10.1007/11818175_6
22. Halevi, S., Kalai, Y.T.: Smooth projective hashing and two-message oblivious transfer. J. Cryptology **25**(1), 158–193 (2012). https://doi.org/10.1007/s00145-010-9092-8
23. Jain, A., Kalai, Y.T., Khurana, D., Rothblum, R.: Distinguisher-dependent simulation in two rounds and its applications. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017, Part II. LNCS, vol. 10402, pp. 158–189. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-63715-0_6
24. Kalai, Y.T.: Smooth projective hashing and two-message oblivious transfer. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 78–95. Springer, Heidelberg (2005). https://doi.org/10.1007/11426639_5
25. Kalai, Y.T., Raz, R.: Probabilistically checkable arguments. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 143–159. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-03356-8_9
26. Kalai, Y.T., Rothblum, G.N., Rothblum, R.D.: From obfuscation to the security of Fiat-Shamir for proofs. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017, Part II. LNCS, vol. 10402, pp. 224–251. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-63715-0_8
27. Khurana, D., Sahai, A.: Two-message non-malleable commitments from standard sub-exponential assumptions. IACR Cryptology ePrint Archive 2017, 291 (2017). http://eprint.iacr.org/2017/291
28. Naor, M., Pinkas, B.: Efficient oblivious transfer protocols. In: Proceedings of the Twelfth Annual Symposium on Discrete Algorithms, Washington, DC, USA, 7–9 January 2001, pp. 448–457 (2001)
29. Pass, R.: Simulation in quasi-polynomial time, and its application to protocol composition. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 160–176. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-39200-9_10

30. Sahai, A., Vadhan, S.P.: A complete problem for statistical zero knowledge. J. ACM **50**(2), 196–249 (2003). http://doi.acm.org/10.1145/636865.636868
31. Sahai, A., Waters, B.: How to use indistinguishability obfuscation: deniable encryption, and more. In: Shmoys, D.B. (ed.) Symposium on Theory of Computing, STOC 2014, New York, NY, USA, 31 May–03 June 2014, pp. 475–484. ACM (2014). http://doi.acm.org/10.1145/2591796.2591825
32. Wolf, S., Wullschleger, J.: Oblivious transfer is symmetric. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 222–232. Springer, Heidelberg (2006). https://doi.org/10.1007/11761679_14