Point Obfuscation and 3-Round Zero-Knowledge*

Nir Bitansky and Omer Paneth

Tel Aviv University, Boston University

Abstract. We construct 3-round proofs and arguments with negligible soundness error satisfying two relaxed notions of zero-knowledge (ZK): *weak ZK* and *witness hiding* (WH). At the heart of our constructions lie new techniques based on *point obfuscation with auxiliary input* (AIPO).

It is known that such protocols cannot be proven secure using black-box reductions (or simulation). Our constructions circumvent these lower bounds, utilizing AIPO (and extensions) as the "non-black-box component" in the security reduction.

1 Introduction

Interactive proofs and arguments [GMR85, BCC88] are fundamental notions in the theory of computation. In cryptography, these are typically used to prove NP-statements and the proof is required to maintain the prover's privacy. Different notions of privacy were considered, the most comprehensive one being zero-knowledge (ZK). ZK protocols allow proving an assertion without revealing anything but its validity. That is, the information learned by the verifier from the interaction can be simulated only from the (valid) statement itself.

Since ZK was introduced [GMR85], questions regarding the round complexity of ZK protocols were studied extensively. While it is known that 2-round ZK protocols (with auxiliary input) do not exist for languages outside BPP [GO94], a classical open question is whether there exist 3-round ZK protocols for NP with negligible soundness error. The difficulty of this problem is expressed by the lower bound of [GK96]: there do not exist 3-round black-box ZK (BBZK) protocols with negligible soundness for languages outside BPP. Namely, to prove that a 3-round protocol is ZK, one must demonstrate a simulator that uses the verifier in a non-black-box way.

The work of [Bar01] shows that using non-black-box simulation it is possible to go beyond existing black-box bounds. However, so far we do not know how to use similar techniques to obtain 3-round ZK protocols. Nevertheless, 3-round ZK protocols have been constructed based on non-standard "knowledge assumptions". [HT98, BP04] show a 3-round ZK argument based on the *knowledge of exponent assumption* (KEA) and variants of it. A different "knowledge assumption" was used to show the existence of 3-round ZK proofs for NP [LM01]. (See further discussion in Section 1.2.)

* This research was funded by the Check Point Institute for Information Security, by Marie Curie grant PIRG03-GA-2008-230640, and ISF grant 0603805843.

R. Cramer (Ed.): TCC 2012, LNCS 7194, pp. 189–207, 2012.

In light of the difficulties in achieving 3-round ZK, it is natural to examine relaxations of ZK that might enable the construction of such protocols. We discuss several previously studied relaxations.

Witness indistinguishability (WI). A protocol is WI [FS90] if any two proofs for the same statement that use two different witnesses are indistinguishable. [FS90] show that, while the parallel repetition of basic (3-round) ZK protocols is not BBZK, it is WI. Furthermore, the soundness error decreases exponentially in the number of repetitions. However, WI protocols do not always guarantee witness secrecy; in particular, for statements with a unique NP-witness, WI is meaningless. Nevertheless, [FS90] show how to use WI to achieve other notions of secrecy such as ZK and *witness-hiding*.

Witness hiding (WH). Roughly speaking, a protocol is WH [FS90] with respect to a distribution $\mathcal{D}$ on an NP-language $\mathcal{L}$ if no verifier can extract a witness from its interaction with the honest prover on a common instance $x \leftarrow \mathcal{D}$. For WH to be meaningful, it should be restricted to *hard distributions*; namely, distributions $\mathcal{D}$ for which poly-size circuits cannot find a witness $w \in \mathcal{R}_{\mathcal{L}}(x)$ for instances $x \leftarrow \mathcal{D}$. WH is in a sense a "minimal" notion of privacy; indeed, leaking the entire witness does not leave much room for imagination.

[FS90] present a 3-round protocol with negligible soundness error that is only WH with respect to a specific type of (hard) distributions on languages, where every instance has two witnesses. In contrast, extending the lower bounds of [GK96], the work of [HRS09] show that, for distributions with unique witnesses, 3-round WH cannot be "black-box reduced" to any "standard cryptographic assumption" (e.g., existence of OWFs), given natural limitations on the reduction.

In this work, we are interested in protocols that are WH with respect to all hard distributions (including the unique witness case). We remark that constructing WH protocols for restricted classes of distributions, where a lower bound on their hardness is apriori known, is a relatively easy task (and is not ruled out by [HRS09]). Indeed, using super-polynomial black-box reductions, it is possible to obtain 3-round WH protocols with respect to super-polynomial hard distributions. (For example, $f(n) = \omega(\log n)$ parallel repetitions of any 3-round ZK protocol with constant soundness error is WH with respect to distributions that are hard for $2^{f(n)}$-size adversaries.) Typical cryptographic scenarios, however, do call for secrecy with respect to general languages/distributions where no apriori super-poly hardness bound is known at the protocol's design time. Here, efficient reductions requiring non-black-box techniques are needed.

Weak zero-knowledge (WZK). The standard notion of ZK requires that for any (potentially adversarial) verifier there exist a simulator that simulates its view in an interaction with the honest prover. The simulated view should be indistinguishable from the real one by any (efficient) distinguisher. The notion of WZK [DNRS99] relaxes ZK by changing the order of quantifiers. Specifically, it allows the ZK simulator to depend on the particular distinguisher in question.

While ZK is often used as a sub-protocol in larger systems, WZK is not always suitable for this purpose due to its weaker simulation guarantee. In particular, WZK is not known to be closed under sequential repetition. Nevertheless, WZK is useful in settings where the verifier tries to learn a specific type of information and we can present a distinguisher that can test whether the verifier succeeded in learning it. Examples include verifiers that try to lean a specific predicate of the witness, or any function of the witness that is efficiently verifiable. In particular, WZK implies WH (by considering a distinguisher that tests if the verifier's view contains a valid witness). We note that, for black-box simulation, WZK and (standard) ZK coincide; hence, by [GK96], a 3-round protocol with negligible soundness error cannot even be shown to be WZK with black-box simulation.

To sum up the above discussion, 3-round arguments with negligible soundness error that are ZK, WH or WZK cannot be constructed using black-box techniques. (From this point on, we only consider proofs/arguments with negligible soundness error). In light of the existing non-black-box constructions, it is interesting to investigate which techniques and assumptions could suffice for constructing such protocols. Another interesting related question is understanding whether the relaxed notions of WH and WZK require simpler techniques than for full-fledged ZK; indeed, all existing WH constructions are based on the stronger notion of ZK as a building block. The question of finding "more direct" constructions of WH was already raised by [FS90]. This work sheds new light on both questions, introducing techniques based on *point obfuscation* (PO). We next briefly review the concept of PO.

Point obfuscation and extensions. Informally, an obfuscator is a randomized algorithm $\mathcal{O}$ that gets as input a program C (given by a circuit) and outputs a new program $\mathcal{O}(C)$ that has the same functionality as the original one, but does not leak any additional information on C [BGI$^+$01]. A stronger variant is *obfuscation with auxiliary input*, in which $\mathcal{O}(C)$ does not leak any information even given a related auxiliary input z_C [GK05].

In this work, we consider obfuscation of *point circuits* and their extensions. A point circuit I_s outputs 1 on s and $\perp$ on all other inputs. A *multibit point circuit* $I_{s\to t}$ outputs t on s and $\perp$ otherwise. We also consider a new extension of point circuits which we call *circular point circuits* - these are circuits $I_{s\leftrightarrows t}$ which output t on input s, s on input t, and $\perp$ otherwise. Obfuscators for multibit point circuits are called *Digital Lockers* (DL). We introduce the new notion of *circular digital lockers* (CDL) that are obfuscators for circular point circuits. Point circuits and their extensions are among the very few functionalities for which obfuscators have been shown (albeit, typically, under rather strong hardness assumptions.) So far, however, POs have found only a handful of applications in cryptographic theory, mostly to strong forms of encryption [Can97, Wee05, CD08, CKVW10, BC10].

1.1 Our Contribution

We construct 3-round WH and WZK protocols based on two different variants of point obfuscation:

- 3-round negligible soundness WH proofs for NP, given auxiliary input point obfuscators that satisfy a relatively mild distributive security requirement. The protocol is WH with respect to general hard distributions (including the unique witness case).
- 3-round WZK arguments for NP, given auxiliary input digital lockers that satisfy a worst-case simulation security requirement.

We next give an overview of our constructions, followed by a discussion on the nature of our obfuscation assumptions and how they relate to previous assumptions used for 3-round ZK protocols.

3-round witness-hiding. The high level idea behind our WH protocol is as follows. Given an NP statement $x \in \mathcal{L}$, have the verifier $\mathcal{V}$ construct a modified NP verification circuit $\mathsf{Ver}^{y}_{\mathcal{L},x}$ that on a valid witness $w \in \mathcal{R}_{\mathcal{L}}(x)$ outputs a secret random point y and outputs $\perp$ otherwise. $\mathcal{V}$ then "garbles" this circuit using Yao's technique and both parties execute a 2-message oblivious-transfer protocol, at the end of which the prover $\mathcal{P}$ possesses the garbled circuit and the corresponding labels for the witness w. Next, $\mathcal{P}$ evaluates the circuit (on w) and obtains the point y. (This is essentially a *conditional disclosure of secrets* protocol, as termed by [GIKM00, AIR01], where $\mathcal{P}$ learns the output y only if it inputs a valid witness.) In the third message, $\mathcal{P}$ sends back to $\mathcal{V}$ a point obfuscation of y. $\mathcal{V}$ accepts only after verifying it got a valid obfuscation of y.

Informally, soundness follows from the secrecy of the garbled circuit that prevents a dishonest prover from obtaining the random y in case there is no valid witness. In fact, we show that our protocol is a *proof of knowledge*.

The witness-hiding property is based on the security of the underlying obfuscator. To exemplify, consider a version of the protocol where $\mathcal{P}$ sends back y in the clear. Following is an attack on this simple version of the protocol. Consider a cheating verifier $\mathcal{V}^*$ that, instead of garbling $\mathsf{Ver}^{y}_{\mathcal{L},x}$, garbles the identity circuit. $\mathcal{P}$ now evaluates the garbled circuit on w and obtains the point $y = w$. If $\mathcal{P}$ was to simply send back y in the clear, $\mathcal{V}^*$ would have learned w and the protocol would be completely insecure. Instead, $\mathcal{P}$ sends back an obfuscation $\mathcal{O}(y)$. The security of the obfuscator $\mathcal{O}$ should then assure that $\mathcal{V}^*$ cannot obtain w, unless "it was already known" to $\mathcal{V}^*$ in advance.

The security reduction and required obfuscation assumptions. As we have seen, the WH guarantee of our protocol depends on the security of the underlying point obfuscator $\mathcal{O}$. We now discuss the properties of the obfuscation used to show WH. Concretely, our underlying obfuscator should satisfy a distributional indistinguishability requirement with respect to points and related auxiliary information that are jointly sampled from an *unpredictable distribution*. We say that a distribution ensemble $\mathcal{D} = \{(Z_n, Y_n)\}_{n\in\mathbb{N}}$ on pairs of strings is unpredictable (UPD) if poly-size circuits cannot predict (with noticeable chance) the point Y_n, given the potentially related auxiliary input Z_n. We say that $\mathcal{O}$ is a distributional auxiliary input point obfuscator (AIPO) if, for any UPD $\mathcal{D} = \{(Z_n, Y_n)\}$, no poly-size circuit family can distinguish, given Z_n, an obfuscation of $\mathcal{O}(Y_n)$ from an obfuscation of a random point $\mathcal{O}(U_n)$.

In our setting, Z_n represents the common input x and the prover's first message (during the OT protocol). Y_n is the obfuscated point (returned by the honest prover). That is, Z_n is explicitly known to the verifier, while Y_n is obfuscated. A malicious $\mathcal{V}^*$ might choose its (garbled) circuit to output illegitimate information on the witness (i.e., information it could not predict on its own only from Z_n); the obfuscation, however, should prevent it from doing so.

3-round weak zero-knowledge. The WH protocol described above is not ZK - it enables a cheating verifier $\mathcal{V}^*$ to learn arbitrary predicates of the witness. For example, to learn w_1, the first bit of w, $\mathcal{V}^*$ can maliciously choose its garbled circuit to map w to one of two arbitrary points y_0, y_1 according to w_1. In this case, the honest prover sends an obfuscation $\mathcal{O}(y_{w_1})$, and $\mathcal{V}^*$ learns w_1 by simply running the obfuscation on each of the two points y_0, y_1. (This can be generalized to any function $f(w)$ where $|f(w)| = O(\log n)$, using a poly-size set $\{y_i\}$).

Towards making the protocol ZK, we try to cope with the above attack by requiring that the verifier "proves" it "fully knows" the secret point y (rather than just a poly-size set containing y). To achieve this without adding rounds, we ask that the verifier itself includes an obfuscation of y in its message. The prover then checks the obfuscation's consistency with the point extracted from the circuit evaluation. In case of inconsistency, the prover aborts. This modification, however, still does not prevent the above attack. The verifier $\mathcal{V}^*$ can learn w_1 by sending an obfuscation of the string y_0 and observing whether the prover aborts. Moreover, the protocol may no longer be sound since a cheating prover might use the verifier's obfuscation to create an obfuscation of the same point y without "knowing" y.

We resolve these issues as follows: (a) to regain soundness, we use an obfuscation scheme with non-malleability properties, based on an obfuscated circular point circuit. (b) to achieve WZK, we require that, instead of a plain point obfuscation, the verifier sends an obfuscated multibit point circuit that on the secret input y outputs the coins used by the verifier to garble the circuit. Now, the prover can verify that the garbled circuit is indeed $\mathsf{Ver}^y_{\mathcal{L},x}$ (for some y).

In order to show that the protocol is WZK, we use stronger notions of obfuscation. Since WZK requires worst-case simulation (i.e., simulation for any x), we require that our obfuscators also satisfy a worst-case simulation guarantee (rather than the weaker distributive definition used for WH). To simulate any verifier $\mathcal{V}^*$, our simulator must make use of the obfuscation simulator for $\mathcal{V}^*$. However, an obfuscation simulator for general adversaries with long output could not exist (see [BGI$^+$01]); in fact, known constructions of PO only address simulation of adversaries with a single output bit. To overcome this, we use the fact that the WZK simulator is given a specific distinguisher $\mathcal{D}$, and the simulated verifier view should only needs to fool this specific $\mathcal{D}$. We show how to use an obfuscation simulator for the binary adversary $\mathcal{D}(\mathcal{V}^*)$, which is the composition of the distinguisher and the verifier, in order to construct a WZK simulator. Indeed, this limitation on simulating adversaries with long output is the reason we do not achieve full-fledged ZK.

1.2 Reflections on the Use of Point Obfuscation

The results of [GK96, HRS09] imply that our 3-round protocols cannot be shown secure using reductions that only make black-box use of the adversary. This is not surprising: indeed, neither auxiliary input nor standard point obfuscators can be shown to be secure using black-box reductions [Wee05]. Hence, our use of obfuscation inherently implies that the verifier is not used as a black-box.

To demonstrate the non-black-box nature of POs, we briefly review the techniques used in existing constructions [Can97, Wee05]. We can view POs as a special case of AIPOs, where the auxiliary input Z_n is empty. In this case, Y_n is unpredictable if it is *well-spread* (i.e., has super-logarithmic min-entropy) and the security requirement is that $\mathcal{O}(Y_n) \approx_c \mathcal{O}(U_n)$ for any well-spread Y_n.

The hardness assumptions made in [Can97, Wee05] are shown to imply that the strategy of any distinguisher essentially consists of a poly-size set of "distinguishing elements". That is, only obfuscations of points within this set are distinguishable from an obfuscation of a random point. However, these elements cannot be extracted using black-box access to the adversary. Hence, they are given to the reduction (or simulator) as non-uniform advice.

These techniques allow achieving the stronger worst-case simulation definition, thus showing that the distributive and worst-case definitions are in fact equivalent in the case of no auxiliary input. When considering auxiliary input, we can no longer apply these techniques. Indeed, the set of distinguishing elements can now depend on the auxiliary input in an arbitrary way. That is, no short advice suffices for the reduction to go through. In general, we do not know whether the distributive AIPO definition implies the worst-case simulation definition in the auxiliary input case (the converse still holds).

Concrete constructions. There exist very few constructions that were shown to be secure with respect to auxiliary input. [GK05] show that any point obfuscator is also secure with respect to auxiliary input that is chosen independently of the obfuscated point. [DKL09] suggest a construction that, under a variant of the LWE assumption, satisfies a restricted definition where the distribution $\mathcal{D}$ is "highly unpredictable". Both results are insufficient for our needs.

In this work, we consider two concrete constructions of AIPOs based on two different assumptions. The first AIPO, known as the (r, r^x) obfuscator, was suggested by Canetti [Can97] based on a strong variant of DDH. Informally, the assumption states that there exists an ensemble of prime order groups $\mathcal{G} = \{\mathbb{G}_n : |\mathbb{G}_n| = p_n\}$ such that for any unpredictable distribution $\mathcal{D} = (Z_n, Y_n)$ with support $\{0,1\}^{\mathrm{poly}(n)} \times \mathbb{Z}_{p_n}$: $(z, r, r^y) \approx_c (z, r, r^u)$, where $(z, y) \leftarrow (Z_n, Y_n), u \xleftarrow{U} \mathbb{Z}_{p_n}$ and r is a random generator of $\mathbb{G}_n$ [1].

For the second construction, we suggest a new assumption that is stated in terms of uninvertibility rather than indistinguishability. The assumption strengthens the assumption made by Wee [Wee05] to account for auxiliary inputs. Roughly, to construct (non auxiliary input) POs, Wee assumes a strong one-way

[1] Both [Can97, DKL09], make use of a slightly different formulation for the distributional AIPO requirement. Their formulation is essentially equivalent to ours.

permutation f that is "uninvertible" with respect to all well-spread distributions. A natural extension of the latter to the auxiliary input setting is to assume that the permutation is hard to invert, even given side information Z on the pre-image Y, from which Y cannot be predicted. An additional fact used by Wee is that permutations inherently preserve (information-theoretic) entropy; in particular, if Y is well-spread, so is $f(Y)$. In the (computational) auxiliary input setting, this might not be true; namely, it might be that Y is unpredictable from Z, while $f(Y)$ is predictable from Z. One possible way to deal with this issue is to assume a trapdoor permutation family (with the above strong uninvertibility). Further details can be found in the full version of this paper [BP11].

We remark that both the assumptions we consider (or any assumption that states that a specific obfuscation candidate is an AIPO satisfying either a the worst-case or the distributive security definition) are considered to be non-standard. In particular, any such assumption is non-falsifiable in the terms of Naor [Nao03].

Comparison with previous work on 3-round ZK. As already mentioned, it is known how to construct 3-round ZK arguments and proofs using non-falsifiable "knowledge assumptions," such as the knowledge of exponent assumption (KEA) [HT98, BP04], the POK assumption [LM01], or the existence of "extractable perfect one-way functions" (EPOWF)[CD09].

The KEA assumption [Dam91], essentially asserts that any algorithm that produces a DDH tuple, must "know" the corresponding exponents. Upon the formulation of KEA, [Dam91] raised a more general question regarding the existence of "sparse range one-way functions", such that any algorithm that can sample an element within the function's image, must also "know" a primage (KEA indeed yields such a OWF). The EPOWF primitive of [CD09] formalizes this generalization. All in all, all the above assumptions essentially fall under the abstract notion of EPOWF. (Indeed, [CD09] show that either one of the KEA or the POK assumptions imply the EPOWF primitive, when combined with a hardness assumption such as DDH.)

In this work, we show how to circumvent the black-box impossibility results for 3-round WZK and WH based on a *different* set of primitives; namely, (variants of) point obfuscation with auxiliary input. Currently, we do not know of any formal relation between the AIPO and EPOWF primitives, beyond the relation established in this work (through 3-round ZK). Formalizing such a relation is an interesting question on its own (going beyond the scope of 3-round ZK).

On the efficiency of the construction. Basing our constructions on (Yao-based) secure function evaluation results in efficient protocols with a practical implementation (similarly to [IKOS07]). By working directly with the verification circuit $\mathsf{Ver}_{\mathcal{L}}$, we avoid the overhead of *Karp reducions*, existing in most ZK protocols. Specifically, we can achieve communication complexity $O(ns)$, where n is the security parameter and s is the size of $\mathsf{Ver}_{\mathcal{L}}$. This is not optimal as there

exist ZK arguments with polylog communication complexity [Kil92]. However, these require using PCPs, making them less practical.

Finally, we consider the techniques in use. Unlike previous works, our work demonstrates a direct WH construction that is not based on a ZK protocol. We then strengthen it to a limited form of ZK. Our WH to WZK transformation is specifically tailored for our construction. An interesting open question is whether a general transformation of this type exists.

Organization. In Section 2 we present the main definitions and tools used in this work. In Section 3 and Section 4 we introduce our WH and WZK protocols. For lack of space many of the details and proof are omitted and can be found in the full version of this paper [BP11].

2 Definitions and Tools

2.1 Weak Zero-Knowledge and Witness Hiding

In this work, we discuss two relaxations of ZK which are formalized next.

Weak zero-knowledge. In ZK, we require that the view of any verifier $\mathcal{V}^*$, in an interaction with the honest prover $\mathcal{P}$, can be simulated by an efficient simulator $\mathcal{S}$. The simulated view should be indistinguishable from the view of $\mathcal{V}^*$ for any poly-size distinguisher. In weak ZK (WZK), the simulator is only required to output a view that is indistinguishable from that of $\mathcal{V}^*$ for a specific distinguisher. This is modeled by supplying the simulator with the distinguisher circuit as additional auxiliary input.

Definition 2.1 (Weak zero-knowledge). *An argument system $(\mathcal{P}, \mathcal{V})$ is WZK if for every PPT verifier $\mathcal{V}^*$ there exist a PPT simulator $\mathcal{S}$ such that for every poly-size circuit family of distinguishers $\mathcal{D} = \{D_n\}_{n\in\mathbb{N}}$ and any $x \in \mathcal{L} \cap \{0,1\}^n$, $w \in \mathcal{R}_{\mathcal{L}}(x)$, $z \in \{0,1\}^{\mathrm{poly}(n)}$ it holds that:*

$$|\Pr[D_n((\mathcal{P}(w), \mathcal{V}^*(z))(x)) = 1] - \Pr[D_n(\mathcal{S}(D_n, x, z)) = 1]| \leq \mathrm{negl}(n) \ .$$

Witness-hiding. A protocol is WH if the verifier cannot fully learn a witness from its interaction with $\mathcal{P}$. This requirement is restricted to instances and witnesses (x, w) sampled from "hard distributions".

Definition 2.2 (Hard distribution). *Let $\mathcal{D} = \{D_n\}_{n\in\mathbb{N}}$ be an efficiently samplable distribution ensemble on $\mathcal{R}_{\mathcal{L}}$, i.e., the support of D_n is $\mathsf{Supp}(D_n) = \{(x, w) : x \in \mathcal{L} \cap \{0,1\}^n, w \in \mathcal{R}_{\mathcal{L}}(x)\}$. We say that $\mathcal{D}$ is hard if for any poly-size circuit family $\{C_n\}$ and sufficiently large n it holds that:*

$$\Pr_{(x,w)\overset{D_n}{\leftarrow}\mathcal{R}_{\mathcal{L}}}[C_n(x) \in \mathcal{R}_{\mathcal{L}}(x)] \leq \mathrm{negl}(n) \ .$$

Definition 2.3 ($\mathcal{D}$-witness-hiding). *An argument system $(\mathcal{P}, \mathcal{V})$ for an NP language $\mathcal{L}$ is WH with respect to a hard distribution $\mathcal{D} = \{D_n\}_{n\in\mathbb{N}}$, if for any poly-size verifier $\mathcal{V}^*$ and all large enough $n \in \mathbb{N}$:*

$$\Pr_{(x,w)\leftarrow D_n}[(\mathcal{P}(w), \mathcal{V}^*)(x) \in \mathcal{R}_{\mathcal{L}}(x)] \leq \text{negl}(n) \ .$$

We say that $(\mathcal{P}, \mathcal{V})$ is WH if it is $\mathcal{D}$-WH for every hard distribution $\mathcal{D}$.

As discussed in the introduction, in this work we will be interested in WH protocols (with respect to a every hard distribution), and not with protocols that are WH with respect to a specific hard distribution.

2.2 2-Message Delegation

A central tool used in our constructions is a 2-message delegation protocol in which the prover and verifier jointly evaluate the NP verification circuit of the language on the common instance and the prover's witness. We use this primitive (following the formulation in [IP07]) to abstract the use of the Yao's garbled circuit construction.

A 2-message delegation protocol is executed by parties (A, B) where A has an input x, and B has as input a function f (given by a boolean circuit). The protocol should allow A to obtain $f(x)$ using two messages: $A \to B \to A$, without compromising the input secrecy of either party. We additionally require that, given B's message and secret randomness, one can reconstruct f. The protocol is defined by a tuple of algorithms $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Eval}, \mathsf{Dec}, \mathsf{Open})$ and proceeds as follows:

A: Obtains a key $sk \leftarrow \mathsf{Gen}(1^n)$, computes an encryption of its input $c \leftarrow \mathsf{Enc}(sk, x)$, and sends c.
B: Computes an encrypted output $\hat{c} \leftarrow \mathsf{Eval}(c, f)$ using randomness r, and sends back $\hat{c}$.
A: Outputs $y = \mathsf{Dec}(sk, \hat{c})$.

We briefly describe the security properties required from 2-message delegation schemes in this work:

- **Correctness:** When both parties are honest A outputs $f(x)$.
- **Input Hiding:** An adversarial B cannot learn A's input x (in the semantic security sense).
- **Function Hiding:** An adversarial A learns nothing about B's input f, other than the value of $f(x)$ (security in this case is simulation based).
- **Function Binding:** In a later stage, B can reveal its input function f by exhibiting its random coins. We require that for any message sent by B, it can reveal at most one function. While function-binding is not required in common formulations of delegation protocols, we show that a Yao-based construction (when instantiated with natural forms encryption) has this property.

In the full version of this paper [BP11], we provide a formal definition of secure 2-message delegation and describe a concrete instantiation based on *Yao's garbled circuit* technique and 2-message OT. We also define an information-theoretic version of this primitive, which we use in order achieve a WH protocol with unconditional soundness (i.e., a proof).

2.3 Point Obfuscation with Auxiliary Input

We start by recalling the standard definition for circuit obfuscation with auxiliary input. The definition is a worst-case definition, in the sense that simulation must hold for any circuit in the family and any related auxiliary input.

Definition 2.4 (Worst-case obfuscator with auxiliary input [BGI$^+$01, GK05]). *A PPT $\mathcal{O}$ is an obfuscator with auxiliary input for an ensemble $\mathcal{C} = \{\mathcal{C}_n\}_{n\in\mathbb{N}}$ of families of poly-size circuits if it satisfies:*

- **Functionality.** *For any $n \in \mathbb{N}$, $C \in \mathcal{C}_n$, $\mathcal{O}(C)$ is a circuit that computes the same function as C.*
- **Polynomial slowdown.** *For any $n \in \mathbb{N}$, $C \in \mathcal{C}_n$, $|\mathcal{O}(C)| \leq \mathrm{poly}(|C|)$.*
- **Virtual black box.** *For any PPT adversary $\mathcal{A}$ there is a PPT simulator $\mathcal{S}$ such that for all sufficiently large $n \in \mathbb{N}, C \in \mathcal{C}_n$ and $z \in \{0,1\}^{\mathrm{poly}(n)}$:*

$$\left|\Pr[\mathcal{A}(z, \mathcal{O}(C)) = 1] - \Pr[\mathcal{S}^C(z, 1^{|C|}) = 1]\right| \leq \mathrm{negl}(n) \ ,$$

where the probability is taken over the coins of $\mathcal{A}, \mathcal{S}$ and $\mathcal{O}$.

An obfuscator $\mathcal{O}$ is recognizable if given a program C and an alleged obfuscation of C, $\tilde{C}$, it is easy to verify that C and $\tilde{C}$ compute the same function.

- **Recognizability.** *There exist a polynomial time recognition algorithm $\mathbb{V}$ such that for any $C \in \mathcal{C}_n$:*
  - $\Pr_{\mathcal{O}}[\mathbb{V}(C, \mathcal{O}(C)) = 1] = 1$
  - *For any $\tilde{C} \in \{0,1\}^{\mathrm{poly}(n)}$ if $\mathbb{V}(C, \tilde{C}) = 1$ then $\tilde{C}$ and C compute the same function.*

Point obfuscation. We consider obfuscation of *point circuits* and their extensions. A point circuit I_s outputs 1 on string s and $\perp$ on all other inputs.

Definition 2.5 (Worst-Case auxiliary-input point obfuscation (AIPO)). *A PPT algorithm $\mathcal{O}$ is a worst-case AIPO if it is a recognizable obfuscator (according to Definition 2.4) for the circuit ensemble: $\mathcal{C} = \{\mathcal{C}_n = \{I_s | s \in \{0,1\}^n\}\}_{n\in\mathbb{N}}$.*

Remark 2.1. The notion of recognizable obfuscation was not explicitly defined in previous works. We only consider this property in the context of point obfuscation. While, in general, point obfuscators are not required to be recognizable, previously constructed obfuscators [Can97, Wee05] are trivially recognizable. This is due to the fact that they use public randomness, i.e., the randomness used by the obfuscator appears in the clear as part of the obfuscated circuit. The recognition algorithm, given a program and its obfuscation, can simply rerun the obfuscation algorithm with the public randomness and compare the result to the obfuscation in hand.

We next present a weaker distributional definition for point obfuscation with auxiliary input that previously appeared in [Can97] (in a slightly different formulation). We first give a preliminary definition of unpredictable distributions (generalizing Definition 2.2) and then present the obfuscation definition.

Definition 2.6 (Unpredictable distribution). *A distribution ensemble* $\mathcal{D} = \{D_n = (Z_n, Y_n)\}_{n\in\mathbb{N}}$*, on pairs of strings is unpredictable if no poly-size circuit family can predict* Y_n *from* Z_n*. That is, for every poly-size circuit family* $\{C_n\}_{n\in\mathbb{N}}$ *and for all large enough* n*:*

$$\Pr_{(z,y)\leftarrow D_n}[C_n(z) = y] \leq \text{negl}(n) \ .$$

Definition 2.7 (Auxiliary input point obfuscation for unpredictable distributions (AIPO)). *A PPT algorithm* $\mathcal{O}$ *is a point obfuscator for unpredictable distributions if it satisfies the functionality and polynomial slowdown requirements as in Definition 2.4, and the following secrecy property. For any unpredictable distribution* $\mathcal{D} = \{D_n = (Z_n, Y_n)\}$ *over* $\{0,1\}^{\text{poly}(n)} \times \{0,1\}^n$ *it holds that:*

$$\{z, \mathcal{O}(y) : (z, y) \leftarrow D_n\}_{n\in\mathbb{N}} \approx_c \left\{z, \mathcal{O}(u) : z \leftarrow Z_n, u \overset{U}{\leftarrow} \{0,1\}^n\right\}_{n\in\mathbb{N}} \ .$$

Remark 2.2. Using this definition in our WH construction, we can settle for a slightly relaxed definition with *bounded auxiliary input*; namely $|Y_n| = \omega(|Z_n|)$. We do not know if such a bounded form of auxiliary-input indeed weakens the requirement. However, it does seem to withstand certain "diagonalization attacks" that can be performed for the non-restrictive (under certain obfuscation assumptions).

2.4 Digital Lockers and Circular Digital Lockers

We also consider obfuscation of several extensions of point circuits. Specifically, *multibit point circuits* and *circular point circuits*. A multibit point circuit $I_{s\to t}$ outputs t on s and $\perp$ otherwise. A circular Point circuit $I_{s\leftrightarrows t}$ outputs t on input s, s on input t, and $\perp$ otherwise. Obfuscators satisfying the worst-case AIPO definition (Definition 2.5) for multibit point circuits and circular point circuits are called *digital lockers* (DLs) and *circular digital lockers* (CDLs).

Definition 2.8 (Digital locker (DL)). *A PPT algorithm is a DL if it is a recognizable obfuscator (according to Definition 2.4) for the circuit ensemble:* $\mathcal{C} = \{\mathcal{C}_n = \{I_{s\to t} | s, t \in \{0,1\}^n\}\}_{n\in\mathbb{N}}$.

Definition 2.9 (Circular digital locker (CDL)). *A PPT algorithm is a CDL if it a recognizable obfuscator (according to Definition 2.4) for the circuit ensemble:* $\mathcal{C} = \{\mathcal{C}_n = \{I_{s\leftrightarrows t} | s, t \in \{0,1\}^n\}\}_{n\in\mathbb{N}}$.

Remark 2.3. We note that the "security under circularity" feature is inherently provided by the strong obfuscation guarantees, was already considered in previous work for constructing strong encryption schemes which withstand *key dependent messages* and *related keys attacks* [CKVW10, BC10].

While AIPOs are sufficient for our WH protocol, our WZK protocol requires DLs and CDLs. In the full version of this paper [BP11], we describe how DLs and CDLs can be constructed based on a worst-case AIPOs that satisfy an additional property of *composability*.

3 3-Round WH

Overview of the protocol. As a warmup, consider first the following **unsound** protocol: to prove an NP statement $x \in \mathcal{L}$, the prover $\mathcal{P}$ and verifier $\mathcal{V}$ first engage in a 2-message delegation protocol where $\mathcal{P}$'s (secret) input is the witness w and $\mathcal{V}$'s input function is the NP verification circuit $\mathsf{Ver}_{\mathcal{L},x}$. $\mathcal{P}$ obtains the result $\mathsf{Ver}_{\mathcal{L},x}(w)$ and sends it to $\mathcal{V}$. This is unsound since a cheating prover can always send "1" as its last message.

To make the protocol sound, we augment it as follows. Let $\mathsf{Ver}^{y}_{\mathcal{L},x}$ be a circuit that outputs y on valid witnesses and $\perp$ otherwise. Now, $\mathcal{V}$ will choose a secret string $y \in_R \{0,1\}^n$ and use the circuit $\mathsf{Ver}^{y}_{\mathcal{L},x}$ as its secret input in the delegation protocol. In order to convince $\mathcal{V}$ of the statement, $\mathcal{P}$ should send back y. Indeed, in case $x \notin \mathcal{L}$ we have $\mathsf{Ver}^{y}_{\mathcal{L},x} \equiv \perp$, and hence, the "function hiding" property of the delegation protocol assures that $\mathcal{P}$ does not learn the random y.

However, this protocol is not witness hiding. Indeed, a cheating verifier can try to obtain w by maliciously choosing its input function. For instance, choosing the function to be the identity results in the prover sending back w.

A natural approach towards fixing the latter problem would be to have the verifier "prove" it behaved honestly, without revealing its secret. In other words, it should give a round-efficient witness-hiding proof, which is what we set out to do to begin with. Thus, we take a different approach. We note that an honest verifier that "knows" y should only be able to verify that the prover "knows" it as well; hence, it suffices to have the prover send a *point obfuscation* of y, instead of sending y in the clear. The security of the obfuscation would then guarantee that any information that the verifier learns on w could also be learned (with noticeable probability) without the obfuscation.

The protocol. Let $\mathsf{DEL} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Eval}, \mathsf{Dec}, \mathsf{Open})$ be a secure 2-message delegation protocol and let $\mathcal{O}$ be a point obfuscator for unpredictable distributions (AIPO) with recognition algorithm $\mathbb{V}$. The protocol is given by Figure 1.

Theorem 3.1. *Let* DEL *be a secure 2-message delegation protocol, and let* $\mathcal{O}$ *be an AIPO. Protocol 1 is a WH interactive argument.*

We briefly overview the proof of Theorem 3.1. The full proof as well as an extension from an argument to a proof can be found in the full version of this paper [BP11].

Soundness. The soundness of Protocol 1 follows from the function hiding of the underlying delegation scheme DEL and the recognizability of the point obfuscator. Indeed, in case there is no valid witness the verifier's message reveals

Common Input: $x \in \mathcal{L}$. **Auxiliary Input to** $\mathcal{P}$**:** $w \in \mathcal{R}_{\mathcal{L}}(x)$.

1. $\mathcal{P}$**:** Obtains $sk \leftarrow \mathsf{Gen}(1^n)$ and sends $c = \mathsf{Enc}(sk, w)$.
2. $\mathcal{V}$**:** Samples $y \overset{U}{\leftarrow} \{0,1\}^n$, obtains $\hat{c} \leftarrow \mathsf{Eval}(c, \mathsf{Ver}^y_{\mathcal{L},x})$ and sends $\hat{c}$.
3. $\mathcal{P}$**:** Decrypts $\tilde{y} = \mathsf{Dec}(sk, \hat{c})$, computes a point obfuscation $\mathcal{O}(\tilde{y})$ and sends it.
4. $\mathcal{V}$**:** Accepts iff $\mathbb{V}(I_y, \mathcal{O}(\tilde{y})) = 1$, i.e., $\mathcal{O}(\tilde{y})$ computes the same function as I_y.

Fig. 1. Protocol 1, 3-round Witness Hiding

no information regarding the verifier's secret random point y. Specifically, the prover's view can be simulated independently of y. Since the obfuscation is recognizable, in order to fool the verifier, the prover must send a point circuit computing I_y and can only succeed with negligible probability.

Proof of Knowledge. In fact, we can show that our WH protocol satisfies a stronger soundness property, namely it is a *proof of knowledge*. For this purpose, we use a similar idea to the one in the "knowledge attack" that shows why the protocol is not ZK (described in the introduction). In order to extract a witness, we essentially apply this attack repeatedly "against" the prover, revealing the witness' bits one by one. Our extractor only makes black-box use of the prover and extracts the witness using rewinding.

Witness hiding. The WH property is based on the input hiding of the delegation scheme DEL and the indistinguishability with respect to unpredictable distributions guarantee of the AIPO $\mathcal{O}$. Concretely, we show how any $\mathcal{V}^*$ that manages to extract a witness w from its interaction with $\mathcal{P}$ can be used to break the input hiding property of DEL. The reduction samples (x, w) from the hard distribution and submits $c_0 = w, c_1 = 1^{|w|}$ to the challenger. Upon receiving a challenge $c = \mathsf{Enc}(sk, c_b)$ it simulates $\mathcal{V}^*(x)$ with c as the first message. $\mathcal{V}^*$ then generates its own message $\hat{c}$, and it is left to simulate the last obfuscation message. To do so, we treat two cases, corresponding to whether the secret point y (induced by $\mathcal{V}^*$'s choice of input circuit to DEL) is (a) unpredictable from (x, c) or (b) is predictable by some poly size predictor Π. Intuitively, the first corresponds to a verifier that chooses its input circuit maliciously to gain information on w. The second, corresponds to a verifier that chooses its circuit honestly. To simulate the obfuscation in the second case, we apply the prediction circuit to compute $y \leftarrow \Pi(x, c)$ and feed $\mathcal{V}^*$ with $\mathcal{O}(y)$. In the case that y is unpredictable, we feed $\mathcal{V}^*$ an obfuscation $\mathcal{O}(u)$ of a random point u. Finally, when $\mathcal{V}^*$ outputs $\tilde{w}$, we check whether it is a valid witness, and if so answer the challenger with $b = 0$. Otherwise, we guess b at random. Indeed, by the indistinguishability guarantee of the AIPO, in case $b = 0$ (i.e., the simulation is done with an encryption of w) the simulated $\mathcal{V}^*$ will manage to extract a witness with noticeable probability

(related to the the prediction probability of Π and the success probability of $\mathcal{V}^*$ in a true interaction). In case that $b = 1$, the reduction is unlikely to produce a valid witness since its view is completely independent of w and the underlying distribution is hard. We stress that the reduction is, indeed, not black box in $\mathcal{V}^*$; in particular, it applies the predictor Π implied by the AIPO guarantee, which is not black-box in $\mathcal{V}^*$.

On restricted auxiliary input. In our WH protocol we require the AIPO distributional guarantee to hold with respect to any unpredictable distribution. However, we can in fact settle for less. Specifically, the auxiliary input distribution in Protocol 1 is essentially restricted to a very "benign" form; namely, the first delegation message (ciphertext) and the hard instance x; in particular, the auxiliary input is of fixed polynomial size and can be made much shorter than the obfuscated random point.

Why isn't Protocol 1 ZK? Protocol 1 is not ZK and in fact enables a cheating verifier $\mathcal{V}^*$ to learn arbitrary predicates on the witness. Specifically, $\mathcal{V}^*$ can deviate from the protocol by maliciously selecting its input circuit C for the delegation protocol as follows. Let $B : \{0,1\}^* \to \{0,1\}^t$ be a polynomial time computable function with $t = O(\log(n))$ output bits. To learn $B(w)$, $\mathcal{V}^*$ fixes an arbitrary set of strings $Y = \{y_j\}_{j\in\{0,1\}^t}$ and sets its input circuit $C = C_B$ to map the witness w to $y_{B(w)}$. Indeed, given an obfuscation of $C_B(w)$, $\mathcal{V}^*$ can simply run the obfuscation on all points in $\{y_j\}$ and learn $B(w)$. In the following section we explain how to transform Protocol 1 to a WZK protocol.

4 3-Round WZK

Overview of the protocol. To make Protocol 1 WZK, we try to cope with verifiers executing the "malicious circuit choice attack" described in the previous section. As explained in the introduction, this involves two main modifications:

1. We require that the verifier's message also includes a *digital locker* $\mathsf{DL}(I_{y\to r_\mathcal{V}})$, which on the secret input y "unlocks" the secret coins $r_\mathcal{V}$ used by the verifier in the delegation protocol. Upon receiving this message, the honest prover $\mathcal{P}$ applies Dec as in the previous protocol, obtains y, and then retrieves the coins $r_\mathcal{V}$. Now $\mathcal{P}$ can apply the Open algorithm of the delegation to verify that the input circuit of $\mathcal{V}^*$ was honestly chosen (to be $\mathsf{Ver}^y_{\mathcal{L},x}$). In case it was not, $\mathcal{P}$ returns a *circular digital locker* (CDL) (Definition 2.9) of a randomly selected circular point circuit.
2. The prover is required to send back an obfuscation of y (as in the previous protocol). However, to maintain soundness we should prevent a malicious prover from using (or mauling) the verifier's message $\mathsf{DL}(I_{y\to r_\mathcal{V}})$ to get the required obfuscation. For this purpose, we apply a "non-malleable obfuscation scheme", implemented as follows.[2] In its first message, the prover

[2] We only consider a very restricted form of non-malleability where the adversary tries to copy an obfuscation of the same point. A more general notion of non-mailable obfuscation can be found in [CV08].

commits to a random $r \in \{0,1\}^n$ (by sending the image of r under some injective OWF f). Then in the last message, it sends a *circular digital locker* $\mathsf{CDL}(I_{y \leftrightarrows r})$ that "binds" r and the secret point y. The honest verifier then runs the CDL on y, retrieves r and uses the CDL recognition algorithm to validate the CDL.

We now fully describe the protocol and then turn to analyze it.

The protocol. Let $\mathsf{DEL} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Eval}, \mathsf{Dec}, \mathsf{Open})$ be a secure 2-message delegation scheme. Let DL, CDL be a digital locker and a circular digital locker. Let $\mathbb{V}$ be the recognition algorithm for the CDL. Let f be an *injective one way function*. The protocol is presented in Figure 2.

Common Input: $x \in \mathcal{L}$. **Auxiliary Input to $\mathcal{P}$:** $w \in \mathcal{R}_{\mathcal{L}}(x)$.

1. $\mathcal{P}$**:** Obtains $sk \leftarrow \mathsf{Gen}(1^n)$ and $c \leftarrow \mathsf{Enc}(sk, w)$, samples $r \xleftarrow{U} \{0,1\}^n$, sends c and $f(r)$.
2. $\mathcal{V}$**:** Samples $y \xleftarrow{U} \{0,1\}^n$, obtains $\hat{c} \leftarrow \mathsf{Eval}(c, \mathsf{Ver}^{y}_{\mathcal{L},x})$ using random coins $r_{\mathcal{V}}$, sends $\hat{c}$ and $\mathsf{DL}_{\mathcal{V}} = \mathsf{DL}(I_{y \to r_{\mathcal{V}}})$.
3. $\mathcal{P}$**:** Decrypts $\tilde{y} = \mathsf{Dec}(sk, \hat{c})$, obtains $\tilde{r}_{\mathcal{V}} = \mathsf{DL}_{\mathcal{V}}(\tilde{y})$, verifies that $\mathbb{V}(I_{\tilde{y} \to r_{\mathcal{V}}}, \mathsf{DL}_{\mathcal{V}}) = 1$ and $\mathsf{Open}(\hat{c}, \tilde{r}_{\mathcal{V}}) = \mathsf{Ver}^{\tilde{y}}_{\mathcal{L},x}$.

   If so, sends back $\mathsf{CDL}_{\mathcal{P}} = \mathsf{CDL}(I_{\tilde{y} \leftrightarrows r})$. Otherwise, samples $u \xleftarrow{U} \{0,1\}^n$ and sends back $\mathsf{CDL}_{\mathcal{P}} = \mathsf{CDL}(I_{u \leftrightarrows u})$.
4. $\mathcal{V}$**:** Obtains $\tilde{r} = \mathsf{CDL}_{\mathcal{P}}(y)$, accepts iff $f(\tilde{r}) = f(r)$ and $\mathbb{V}(I_{y \leftrightarrows \tilde{r}}, \mathsf{CDL}_{\mathcal{P}}) = 1$.

Fig. 2. Protocol 2, 3-round WZK

Theorem 4.1. *Let* DEL *be a 2-message delegation protocol, let* DL *be a digital locker and* CDL *a circular digital locker, and let* f *be an injective one way function, then Protocol 2 is a WZK argument.*

We briefly overview the proof of Theorem 3.1. The full proof can be found in the full version of this paper [BP11].

Soundness. Soundness is shown in two stages. First, we argue that given $\mathcal{V}$'s message $(\hat{c}, \mathsf{DL}_{\mathcal{V}})$, it is hard to recover the underlying secret point y. I.e, no poly-size circuit family can recover y, except with negligible chance. Indeed, the auxiliary input obfuscation guarantee implies that if y can be recovered from $\mathsf{DL}_{\mathcal{V}}$ and the related auxiliary information $\hat{c}$, it can also be recovered solely from $\hat{c}$. However, since $x \notin \mathcal{L}$ and DEL is function hiding, y cannot be recovered from $\hat{c}$ (similarly to the WH protocol).

Second, we show that any cheating prover $\mathcal{P}^*$ can be used to recover y from $\mathcal{V}$'s message. Assume WLOG that $\mathcal{P}^*$ is deterministic, and note that, in its

first message, $\mathcal{P}^*$ sends some (fixed) $f(r)$. Since f is injective, $\mathcal{P}^*$ is in fact "committed" to the corresponding fixed r. We can then feed $\mathcal{P}^*$ with $\mathcal{V}$'s message and get back $\mathsf{CDL}_{\mathcal{P}}$. Noting that whenever $\mathcal{P}^*$ convinces $\mathcal{V}$, $\mathsf{CDL}_{\mathcal{P}}(r) = y$, we can run $\mathsf{CDL}_{\mathcal{P}}$ on r (given as non-uniform advice) and obtain y with noticeable probability.

Weak zero-knowledge. We present a WZK simulator that, given an adversary $\mathcal{V}^*$ and a distinguisher $\mathcal{D}$, simulates the view of V^* with respect to $\mathcal{D}$. Let $\mathcal{V}^*_{\mathcal{D}}$ be the composition of $\mathcal{D}$ with $\mathcal{V}^*$. $\mathcal{V}^*_{\mathcal{D}}$ outputs a bit after receiving $\mathsf{CDL}_{\mathcal{P}} = \mathsf{CDL}(I_{y \leftrightarrows r})$ as the last message. In particular, there exist a PPT $\mathcal{S}_{\mathsf{CDL}}$ that simulates $\mathcal{V}^*_{\mathcal{D}}$'s output given oracle access to $I_{y \leftrightarrows r}$ and auxiliary input $\mathsf{ai} = (z, x, c, f(r))$, representing the rest of $\mathcal{V}^*_{\mathcal{D}}$'s view.

The WZK simulator $\mathcal{S}$ will simulate ai on its own, and utilize $\mathcal{S}_{\mathsf{CDL}}$ to simulate $\mathsf{CDL}_{\mathcal{P}}$ as the last message. To simulate ai, $\mathcal{S}$ samples r and computes $f(r)$. c is simulated by generation a random key $sk \leftarrow \mathsf{Gen}(1^n)$ and computing $c = \mathsf{Enc}(sk, 0^{|w|})$ (instead of w as in a true interaction). The input hiding of DEL implies that the simulated ai is indistinguishable from the true ai. We explain how $\mathcal{S}_{\mathsf{CDL}}$ is used to simulate the last obfuscation message. $\mathcal{S}$ first obtains the verifier's message $(\mathsf{DL}_{\mathcal{V}^*}, \hat{c})$. It then runs $\mathcal{S}_{\mathsf{CDL}}$ with the simulated ai, monitoring all its oracle queries. We treat two separate cases: (a) $\mathcal{S}_{\mathsf{CDL}}$ makes a query y which unlocks $\mathsf{DL}_{\mathcal{V}^*}$; (b) $\mathcal{S}_{\mathsf{CDL}}$ never makes such a query, in which case we always answer its queries with $\perp$.

The first case corresponds to a verifier that "knows" the secret point y. In this case, our simulator can perfectly simulate the behavior of $\mathcal{P}$. That is, it can "open" $\hat{c}$ to check its validity and consistency with $\mathsf{DL}_{\mathcal{V}^*}$, and send back the corresponding CDL.

The second case corresponds to a cheating $\mathcal{V}^*$ that either produces an invalid message or somehow produces a valid message but without actually "knowing" the secret y. In this case, the simulator will always return a "dummy obfuscation". This simulates the behavior of the honest prover $\mathcal{P}$. Indeed, if $\mathcal{V}^*$'s message is invalid, the prover also produces a "dummy obfuscation". If $\mathcal{V}^*$ does not "know" y, it can not distinguish $\mathcal{P}$'s message from a "dummy obfuscation".

The simulator. Let $\mathcal{V}^*$ be any verifier, and let $\mathcal{D}$ be the distinguisher circuit. Denote by $\mathcal{V}^*_1(z, x, c, f(r))$ the algorithm that runs $\mathcal{V}^*(z, x)$, feeds it with $(c, f(r))$ as the first message, and outputs $\mathcal{V}^*$'s message. Denote by $\mathcal{V}^*_2(x, z, c, f(r), \mathsf{CDL}_{\mathcal{P}})$ the algorithm that runs $\mathcal{V}^*(x, z)$, feeds it with $(c, f(r))$ as a first message, with $\mathsf{CDL}_{\mathcal{P}}$ as a second message, and returns $\mathcal{V}^*$'s output. Also, denote by $\mathcal{V}^*_{\mathcal{D}}(x, z, c, f(r), \mathsf{CDL}_{\mathcal{P}})$ the algorithm that runs $\mathcal{V}^*_2(x, z, c, f(r), \mathsf{CDL}_{\mathcal{P}})$, applies the circuit $\mathcal{D}$ on the output of $\mathcal{V}^*_2$ and returns the output bit of $\mathcal{D}$.

Let $\mathcal{S}_{\mathcal{V}^*,\mathcal{D}}(x, z, c, f(r))$ be the PPT obfuscation simulator of $\mathcal{V}^*_{\mathcal{D}}$ as specified by Definition 2.4. Also let $\ell(n)$ be the length of a witness for instances of length n. The description of the simulator is given by Algorithm 1.

Algorithm 1. Simulator $\mathcal{S}$

Input: $x \in \mathcal{L}, z \in \{0,1\}^*$

Set $\tilde{y} = \perp$.
Sample $r, u \xleftarrow{U} \{0,1\}^n$.
Obtain $sk \leftarrow \mathsf{Gen}(1^n)$.
Compute $c \leftarrow \mathsf{Enc}(sk, 1^{\ell(|x|)})$
Compute $(\hat{c}, \mathsf{DL}_{\mathcal{V}}) = \mathcal{V}_1^*(x, z, c, f(r))$.
Emulate $\mathcal{S}_{\mathcal{V}^*,\mathcal{D}}(x, z, c, f(r))$.
for each oracle query Q made by $\mathcal{S}_{\mathcal{V}^*,\mathcal{D}}$ **do**
  if $\mathsf{DL}_{\mathcal{V}}(Q) = \perp$ **then**
    Answer $\mathcal{S}$'s query with $\perp$ and continue the emulation.
  else
    Set $\tilde{r}_{\mathcal{V}} = \mathsf{DL}_{\mathcal{V}}(Q)$
    if $\mathbb{V}(I_{Q \to r_{\mathcal{V}}}, \mathsf{DL}_{\mathcal{V}}) = 1$ **then**
      Set $\tilde{y} = Q$
    end if
    End the emulation of $\mathcal{S}_{\mathcal{V}^*,D}$.
  end if
end for
if $\tilde{y} = \perp$ **or** $\mathsf{Open}(\hat{c}, \tilde{r}_{\mathcal{V}}) \neq \mathsf{Ver}^{\tilde{y}}_{\mathcal{L},x}$ **then**
  return $\mathcal{V}_2^*(x, z, c, f(r), \mathsf{CDL}(I_{u \leftrightarrows u}))$.
else
  return $\mathcal{V}_2^*(x, z, c, f(r), \mathsf{CDL}(I_{\tilde{y} \leftrightarrows r}))$.
end if

Acknowledgements. We thank Amit Sahai for introducing us to the problem of 3-round witness hiding. We thank Ran Canetti and Yuval Ishai for valuable discussions. In particular, we thank Yuval for the idea of how to transform the witness-hiding protocol from an argument to a proof and for bringing to our attention previous applications of conditional disclosure of secrets.

References

[AIR01] Aiello, W., 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)

[Bar01] Barak, B.: How to go beyond the black-box simulation barrier. In: FOCS, pp. 106–115 (2001)

[BC10] Bitansky, N., Canetti, R.: On Strong Simulation and Composable Point Obfuscation. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 520–537. Springer, Heidelberg (2010)

[BCC88] Brassard, G., Chaum, D., Crépeau, C.: Minimum disclosure proofs of knowledge. J. Comput. Syst. Sci. 37(2), 156–189 (1988)

[BGI+01] Barak, B., Goldreich, O., Impagliazzo, R., Rudich, S., Sahai, A., Vadhan, S.P., Yang, K.: On the (Im)possibility of Obfuscating Programs. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 1–18. Springer, Heidelberg (2001)

[BP04] Bellare, M., Palacio, A.: The Knowledge-of-Exponent Assumptions and 3-Round Zero-Knowledge Protocols. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 273–289. Springer, Heidelberg (2004)

[BP11] Bitansky, N., Paneth, O.: Point obfuscation and 3-round zero-knowledge, Cryptology ePrint Archive, Report 2011/493 (2011), http://eprint.iacr.org/

[Can97] Canetti, R.: Towards Realizing Random Oracles: Hash Functions that Hide All Partial Information. In: Kaliski Jr., B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 455–469. Springer, Heidelberg (1997)

[CD08] Canetti, R., Dakdouk, R.R.: Obfuscating Point Functions with Multibit Output. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 489–508. Springer, Heidelberg (2008)

[CD09] Canetti, R., Dakdouk, R.R.: Towards a Theory of Extractable Functions. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 595–613. Springer, Heidelberg (2009)

[CKVW10] Canetti, R., Tauman Kalai, Y., Varia, M., Wichs, D.: On Symmetric Encryption and Point Obfuscation. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 52–71. Springer, Heidelberg (2010)

[CV08] Canetti, R., Varia, M.: Non-malleable obfuscation, Cryptology ePrint Archive, Report 2008/495 (2008), http://eprint.iacr.org/

[Dam91] Damgård, I.: Towards Practical Public Key Systems Secure against Chosen Ciphertext Attacks. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 445–456. Springer, Heidelberg (1992)

[DKL09] Dodis, Y., Kalai, Y.T., Lovett, S.: On cryptography with auxiliary input. In: STOC, pp. 621–630 (2009)

[DNRS99] Dwork, C., Naor, M., Reingold, O., Stockmeyer, L.J.: Magic functions. In: FOCS, pp. 523–534 (1999)

[FS90] Feige, U., Shamir, A.: Witness indistinguishable and witness hiding protocols. In: STOC, pp. 416–426 (1990)

[GIKM00] Gertner, Y., Ishai, Y., Kushilevitz, E., Malkin, T.: Protecting data privacy in private information retrieval schemes. In: JCSS, pp. 151–160. ACM Press (2000)

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

[GK05] Goldwasser, S., Kalai, Y.T.: On the impossibility of obfuscation with auxiliary input. In: FOCS, pp. 553–562 (2005)

[GMR85] Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof-systems (extended abstract). In: STOC, pp. 291–304 (1985)

[GO94] Goldreich, O., Oren, Y.: Definitions and properties of zero-knowledge proof systems. J. Cryptology 7(1), 1–32 (1994)

[HRS09] Haitner, I., Rosen, A., Shaltiel, R.: On the (Im)Possibility of Arthur-Merlin Witness Hiding Protocols. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 220–237. Springer, Heidelberg (2009)

[HT98] Hada, S., Tanaka, T.: On the Existence of 3-Round Zero-Knowledge Protocols. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 408–423. Springer, Heidelberg (1998)

[IKOS07] Ishai, Y., Kushilevitz, E., Ostrovsky, R., Sahai, A.: Zero-knowledge from secure multiparty computation. In: STOC, pp. 21–30 (2007)

[IP07] Ishai, Y., Paskin, A.: Evaluating Branching Programs on Encrypted Data. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 575–594. Springer, Heidelberg (2007)

[Kil92] Kilian, J.: A note on efficient zero-knowledge proofs and arguments (extended abstract). In: STOC, pp. 723–732 (1992)

[LM01] Lepinski, M., Micali, S.: On the existence of 3-round zero-knowledge proof systems. Tech. report, MIT LCS (2001)

[Nao03] Naor, M.: On Cryptographic Assumptions and Challenges. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 96–109. Springer, Heidelberg (2003)

[Wee05] Wee, H.: On obfuscating point functions. In: STOC, pp. 523–532 (2005)