Point Obfuscation and 3-round Zero-Knowledge *

Nir Bitansky and Omer Paneth

Tel Aviv University and Boston University

September 21, 2011

Abstract

We construct 3-round proofs and arguments with negligible soundness error satisfying two relaxed notions of *zero-knowledge*: *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. We also investigate the relation between AIPO and the assumptions previously used to achieve 3-round ZK.

*This research was funded by the Check Point Institute for Information Security.

Contents

1 Introduction

Interactive proofs (IP's) and arguments (IA's) [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 on a valid statement learned by the verifier from the interaction can be simulated only from the 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) for languages outside BPP do not exist [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 exemplified by the lower bound of [GK96]: There do not exist 3-round black-box ZK (BBZK) protocols with negligible soundness for languages outside BPP (in BBZK the simulator only has black-box access to the verifier). Namely, to prove that a 3-round protocol is ZK, one must demonstrate a non-black-box simulator.

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.)

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* (WH).

Witness hiding. Roughly speaking, a protocol is WH [FS90] w.r.t 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 w.r.t a specific type of (hard) distributions on languages where every instance has two witnesses. In contrast, extending the lower bounds of [GK96], [HRS09] show that for distributions with unique witnesses, 3-round WH can not be "black-box reduced" to any "standard cryptographic assumption" (e.g. existence of OWFs), under some natural limitations on the reduction.

In this work, we are interested in protocols that are WH w.r.t 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 w.r.t to super-polynomial hard distributions. (For example, $f(n) = \omega(\log n)$ parallel repetitions of a basic 3-round ZK protocol with constant soundness error, such as Blum, is WH w.r.t distributions that are hard for $2^{f(n)}$-size adversaries.) Typical cryptographic scenarios, however, do call for secrecy w.r.t 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. 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] weakens 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 which 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 can not be shown to even be WZK with a black-box simulator.

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*.

Point obfuscation (PO) and extensions. We briefly review the concept of PO. Informally, an obfuscator is a randomized algorithm $\mathcal{O}$ which 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; in particular there are several constructions that realize PO (and variants), under a number of strong hardness assumptions. So far, however, PO's 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 IP for NP given auxiliary input point obfuscators that satisfy a relatively mild distributive security requirement. The protocol is WH w.r.t general hard distributions (including the unique witness case).
- 3-round WZK IA 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 $\mathcal{V}^*$ can not 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 w.r.t 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 and in fact 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 any witness 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 attack can be generalized to any function of w with output length $O(\log n)$ (using a poly-size set of strings $\{y_i\}$).

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 might 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 (CDL). (b) To achieve WZK, we require that instead of a plain point obfuscation, the verifier sends an obfuscated multibit point circuit (DL) 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, in 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 and it only needs to simulate the output of this distinguisher on $\mathcal{V}^*$. Since $\mathcal{V}^*$ and the distinguisher together can be viewed as an adversary for the obfuscation that outputs a single bit, there exists an obfuscation simulator for this adversary. We show how to use this simulator to construct a WZK simulator. Indeed, this limitation on simulating adversaries with long output is the reason we do not achieve full-fledged ZK.

We note that while we do not know whether our WZK protocol remains secure under sequential composition, we show that if the DL and CDL used are "composable obfuscators" the protocol remains WZK under parallel composition.

1.2 Reflections on the Use of Point Obfuscation

The results of [GK96, HRS09] imply that our 3-round protocols can not 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, the distribution 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 can not 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 w.r.t auxiliary input. [GK05] show that any point obfuscator is also secure w.r.t 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 $(z, y) \leftarrow (Z_n, Y_n), u \stackrel{U}{\leftarrow} \mathbb{Z}_{p_n}$ 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 permutation f that is "uninvertible" w.r.t 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). In Section 2.3, we show a more general (or weaker) assumption and a corresponding construction of AIPOs.

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]. For example, to falsify the the distributive AIPO definition, one has to come up not only with a distinguisher but also with an unpredictable distribution and a proof of its unpredictability.

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 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 range, 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. At this point, 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). We find that formalizing such a relation is an interesting question on its own (going beyond the scope of 3-round ZK).

Coming up with different assumptions (even non falsifiable ones) that can be used to overcome known black-box bounds opens up new directions for overcoming these bounds. In this case, we show that the research of AIPO can also be instrumental for the attempts to overcome black-box impossibility results for 3-round WZK and WH.

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.

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

On the efficiency of the construction. We note that basing our constructions on 2-party secure function evaluation (using Yao's garbled circuit technique) 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* most existing 3-round ZK IA. Specifically, using existing constructions for the relevant primitives, 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 argument with polylog communication complexity [Kil92]. However, these require using *PCP* techniques, making them impractical.

2 Definitions and Tools

2.1 Weak Zero-Knowledge and Witness Hiding

We consider interactive argument systems for NP languages $\mathcal{L}$ with a corresponding witness relation $\mathcal{R}_{\mathcal{L}}$. Each system consists of a pair of PPT prover and verifier algorithms $(\mathcal{P}, \mathcal{V})$. We require that all our protocols satisfy:

- **Perfect completeness.** For any $(x, w) \in \mathcal{R}_{\mathcal{L}}$:
$$\Pr[(\mathcal{P}(w), \mathcal{V})(x) = 1] = 1 \, .$$
- **Negligible soundness error.** For any poly-size prover strategy $\mathcal{P}^*$, any large enough n, and any $x \in \{0,1\}^n \setminus \mathcal{L}$:
$$\Pr[(\mathcal{P}^*, \mathcal{V})(x) = 1] \leq \mathrm{negl}(n) \, .$$
We say that the system is a *proof* (rather than an argument) if it is also sound against provers of unbounded size.

We also consider the following notion of *proof of knowledge*: an interactive proof $(\mathcal{P}, \mathcal{V})$ is a proof of knowledge (POK) if there exist an oracle machine E s.t. for every prover strategy $\mathcal{P}^*$, for every long enough $x \in \mathcal{L}$ and every polynomial p, if $\Pr[(\mathcal{P}^*, \mathcal{V})(x) = 1] \geq p(|x|)$ then $E^{\mathcal{P}}(x) \in \mathcal{R}_{\mathcal{L}}(x)$ and the expected running time of $E^{\mathcal{P}}(x)$ is polynomial in $1/p(|x|)$.

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)**.** *The 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. $\mathsf{Supp}(D_n) = \{(x, w) : x \in \mathcal{L} \cap \{0,1\}^n, w \in \mathcal{R}_{\mathcal{L}}(x)\}$. 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 w.r.t to a hard distribution $\mathcal{D} = \{D_n\}_{n\in\mathbb{N}}$, if for any poly-size verifier $\mathcal{V}^*$ and all large enough $n \in \mathbb{N}$:*

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

We say that $(\mathcal{P}, \mathcal{V})$ is WH if it is WH w.r.t to a every hard distribution.

As discussed in the introduction, in this work we will be interested in WH protocols (w.r.t to a every hard distribution), and not with protocols that are WH w.r.t 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 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: $A \to B \to A$, and 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 $y = \mathsf{Dec}(sk, \hat{c})$.

Definition 2.4 (Secure 2-message delegation)**.** *a protocol $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Eval}, \mathsf{Dec}, \mathsf{Open})$ is a secure 2-message delegation protocol if for every ensemble $\mathcal{C} = \{\mathcal{C}_n\}_{n\in\mathbb{N}}$ of poly-size circuits, where each $C \in \mathcal{C}_n$ has input length n, the following requirements hold:*

- **Correctness:** *For all $n \in \mathbb{N}, x \in \{0,1\}^n$ and $C \in \mathcal{C}_n$, the following procedure outputs $C(x)$ with probability 1:*
  - *Obtain $sk \leftarrow \mathsf{Gen}(1^n)$.*
  - *Compute $c \leftarrow \mathsf{Enc}(sk, x)$.*
  - *Compute $\hat{c} \leftarrow \mathsf{Eval}(c, C)$.*
  - *Output $\mathsf{Dec}(sk, \hat{c})$.*
- **Input Hiding:** *For any poly-size $\mathcal{D}$, the probability that $\mathcal{D}$ wins the following game is at most $1/2 + \mathrm{negl}(n)$:*
  - *On 1^n, $\mathcal{D}$ submits a pair of strings $x_0, x_1 \in \{0,1\}^n$.*
  - *Sample $sk \leftarrow \mathsf{Gen}(1^n)$.*
  - *For a random bit $b \in_R \{0,1\}$, compute $c \leftarrow \mathsf{Enc}(sk, x_b)$ and give c to $\mathcal{D}$.*
  - *$\mathcal{D}$ outputs a guess b' and wins if $b = b'$.*

- **Function Hiding:** *A randomized evaluation should not leak information on the input circuit C. This should hold even when A is malicious and sends an arbitrary first message. Formally, let $\mathcal{E}(x) = \mathsf{Supp}(\mathsf{Enc}(\cdot, x))$ be the set of all legal encryptions of x, and let $\mathcal{E}_n = \cup_{x\in\{0,1\}^n}\mathcal{E}(x)$ be the set legal encryptions for strings of length n. Then there exist a PPT simulator $\mathcal{S}$ such that:*

$$\{C, \mathsf{Eval}(c, C)\}_{\substack{n\in\mathbb{N}, C\in\mathcal{C}_n \\ x\in\{0,1\}^n, c\in\mathcal{E}(x)}} \approx_c \{C, \mathcal{S}(c, C(x))\}_{\substack{n\in\mathbb{N}, C\in\mathcal{C}_n \\ x\in\{0,1\}^n, c\in\mathcal{E}(x)}} \quad (1)$$

$$\{C, \mathsf{Eval}(c, C)\}_{\substack{n\in\mathbb{N} \\ C\in\mathcal{C}_n, c\notin\mathcal{E}_n}} \approx_c \{C, \mathcal{S}(c, \perp)\}_{\substack{n\in\mathbb{N} \\ C\in\mathcal{C}_n, c\notin\mathcal{E}_n}} . \quad (2)$$

- **Function Binding:** *We require that, given the result $\hat{c}$ of evaluating C on an encrypted input and the randomness r used by* Eval*, one can efficiently reconstruct C. The bind should also hold against a malicious evaluator B:*

  1. *For all $n \in \mathbb{N}, x \in \{0,1\}^n$, $C \in \mathcal{C}_n$, the following procedure outputs C with probability 1:*
     - *Obtain $sk \leftarrow \mathsf{Gen}(1^n)$.*
     - *Compute $c \leftarrow \mathsf{Enc}(sk, x)$.*
     - *Compute $\hat{c} \leftarrow \mathsf{Eval}(c, C)$ using randomness r to* Eval.
     - *Output* $\mathsf{Open}(\hat{c}, r)$.
  2. *For any $\hat{c} \in \{0,1\}^*$ there is at most one value of r s.t. $\mathsf{Open}(\hat{c}, r) \neq \perp$.*

2.2.1 Remarks on Definition 2.4.

1. The function-binding property is required in our construction of WZK IA. 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.

2. To get WH proofs (rather than arguments), we use a slightly stronger variant where the function-hiding is information-theoretic (and not computational). In this case, we no longer require the delegation protocol to be function-binding (as in standard commitments, the two properties cannot coexist).

3. The security definition presented is weaker than the standard definition of *secure function evaluation*. Specifically, if the evaluating party B is malicious we can not fully simulate.

4. The definition of function-hiding does not require that the simulator knows whether the encryption $\hat{c}$ is valid or not. In our case, this will be sufficient; specifically, we shall utilize the simulator for circuits the output $\perp$ on all inputs and $\mathcal{S}(c, \perp)$ will correctly simulate the garbled circuit whether c is a valid encryption or not.

Instantiating the 2-message delegation scheme. We describe how to implement a 2-message delegation scheme using *Yao's garbled circuit* technique and 2-message OT. We require a 2-message OT scheme, with the following security guarantee:

1. Computational security for the receiver: for every two inputs, the receiver's messages are computationally indistinguishable.

2. Information-theoretic security for the sender: the view of every receiver can be simulated in an information-theoretic way by a (possibly unbounded) simulator interacting with the ideal OT functionality.

We will use the OT scheme of [NP01] that satisfy this security requirement under the DDH assumption. A description of Yao's garbled circuit construction can be found in [LP09].

Gen**:** Simply outputs as sk random coins for the OT receiver.

Enc**:** Use the randomness string sk to generate and output the receiver's message of the OT, where the choice bits correspond to the bits of the input value x.

Eval**:** Generate a garbled *universal circuit* $\hat{U}$ taking as input a description of the circuit C and another input x for C and outputs $C(x)$. Output the garbled circuit $\hat{U}$, the labels of the input wires describing the input circuit C, and the OT sender's message encoding the labels of the input wires received from A.

Dec**.** Use sk to obtain the labels for the input wires corresponding to x (from the OT sender-message); evaluate the garbled circuit and obtain the result.

Open**.** Given the randomness used to garble the circuit $\hat{U}$, open all the gates of the garble circuit; reveal and output the values of the inputs wires encoding C. For this we require that the underlying encryption scheme for Yao's protocol is committing; namely, any cipher should information-theoretically determine the plaintext (even without the secret key).

The details of the security proof for the function hiding property of the scheme in the semi-honest case are similar to [CCKM00]. By using an OT scheme that is secure against malicious senders, we can show that the function hiding property of the construction holds also in the malicious case. Informally, if a malicious A sends a malformed first message, the security of the OT guaranties that it will learn nothing about the values of the input wires to the garbled circuit. Together with the security of the garbled circuit it follows that the message sent by B can be simulated independently of A's first message. The function binding property follows directly from the fact that the underlying encryption scheme for the garbled circuit is committing.

Instantiating the 2-message delegation protocol with perfect function hiding. We showed how to construct a 2-message delegation protocol using Yao's garbled circuit. Note that the OT scheme used ([NP01]) is also secure against unbounded receivers. We can use an information-theoretic variant of Yao's garbled circuit in a similar way to get a 2-message delegation protocol with perfect function-hiding for NC^1 circuits. A description of an information-theoretic variant can be found in [IK02]. As mentioned above, this variant will be used in order to construct IP rather than IA.

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 definitions in the sense that simulation must hold for any circuit in the family and any related auxiliary input.

Definition 2.5 (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 which 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 2.6 (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 2.5) for the following 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 2.7 (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:*

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

Definition 2.8 (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 2.5, and the following secrecy property. For any unpredictable distribution $\mathcal{D} = \{D_n = (Z_n, Y_n)\}$ over $\{0,1\}^{\mathrm{poly}(n)} \times \{0,1\}^n$ 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.

AIPO constructions. Following are two constructions of AIPOs based on two different assumptions. We recall the construction of [Can97]. Then, we describe a modification of the construction in [**?**] that yields AIPOs given a new assumption that strengthens Wee's original assumption.

Construction 2.1 (The r, r^x point obfuscator [Can97])**.** *Let $\mathcal{G} = \{\mathbb{G}_n\}_{n\in\mathbb{N}}$ be a group ensemble, where each $\mathbb{G}_n$ is a group of prime order $p_n \in (2^{n-1}, 2^n)$. We define an obfuscator, $\mathcal{O}$, for points in the domain $\mathbb{Z}_{p_n}$ as follows: $I_x \overset{\mathcal{O}}{\longmapsto} \mathcal{C}(r, r^x)$, where $r \overset{U}{\leftarrow} \mathbb{G}_n^*$ is a random generator of $\mathbb{G}_n$, and $\mathcal{C}(r, r^x)$ is a circuit which on input i, checks whether $r^x = r^i$.*

[Can97] considers a strong variant of the DDH assumption implying that the (r, r^x) obfuscator is an AIPO satisfying Definition 2.8. The assumption essentially states there exist an ensemble of prime order groups $\mathcal{G} = \{\mathbb{G}_n : |\mathbb{G}_n| = p_n\}$ such that for any unpredictable distribution $\mathcal{D} = (Z_n, X_n)$ with support $\{0,1\}^{\text{poly}(n)} \times \mathbb{Z}_{p_n}$, it holds that that $(z, r, r^x) \approx_c (z, r, r^u)$ where $(z, x) \leftarrow (Z_n, X_n), u \overset{U}{\leftarrow} \mathbb{Z}_{p_n}$ and r is a random generator of $\mathbb{G}_n$. Candidate group ensembles include any ensemble where standard DDH is assumed to hold, e.g. quadratic residues modulo a prime, or elliptic curves groups. We note that Construction 2.1 also satisfies the recognizability requirement given in Definition 2.5. Indeed, the recognition algorithm $\mathbb{V}(I_x, \tilde{C})$, simply checks whether $\tilde{C}$ is of the form $\mathcal{C}(g, h)$, and that $g^x = h$ (here g is the public randomness used in the obfuscation).

The second construction is based on a new assumption that strengthens the assumption of Wee [Wee05] to account for auxiliary inputs. As explained in the introduction the natural extension of Wee's assumption to auxiliary input is insufficient as is. Instead, we make the following assumption and augment Wee's original construction.

Assumption 2.1. *There exists an ensemble of permutation families $\mathcal{F} = \{\mathcal{F}_n = \{f\}\}$ such that for any unpredictable distribution ensemble $\mathcal{D} = \{\mathcal{D}_n = (Z_n, Y_n)\}$, the following two distribution ensembles are also unpredictable:*

- $((Z_n, f(Y_n), f); Y_n)$
- $((Z_n, f); f(Y_n))$,

where in both $f \overset{U}{\leftarrow} \mathcal{F}_n$ (independently of $\mathcal{D}_n$).

We remark that the first property naturally generalizes Wee's assumption regarding strong uninvertibility; in particular, when Z is empty and Y is simply well-spread the assumption coincides with Wee's. The second property essentially guarantees that $f(Y)$ is unpredictable from Z just as Y is; In Wee's assumption (where Z is empty and Y is well-spread) this is inherently guaranteed by the fact that permutations preserve information-theoretic entropy.

We note that the second part of the assumption has a flavor of weak extractability; still, it appears to be significantly weaker than the sparse-range extractability discussed in the introduction. We also note that any trapdoor permutation family would inherently imply the second part of the assumption; hence, it suffices to have trapdoor permutations that satisfy the strong uninvertibility (first part of the assumption).

Construction 2.2. *Let $\mathcal{F}$ be a family of permutations given by Assumption 2.1. The obfuscator $\mathcal{O}$ works as follows: given a point $y \in \{0,1\}^n$, $\mathcal{O}$ samples $3n$ permutations $\{f_i\}_{i\in[3n]}$ from $\mathcal{F}_n$, and $3n$ strings $\{r_i\}_{i\in[3n]}$ from $\{0,1\}^n$. For every $i \in [3n]$, let $f^i = f_i \circ f_{i-1} \circ \cdots \circ f_1$ (where $\circ$ denotes composition). $\mathcal{O}$ outputs a circuit $\mathcal{C}_y$ that has hardcoded into it the randomness of $\mathcal{O}$, $\{f_i, r_i\}_{i\in[3n]}$ and the bits $\left\{b_i = \langle r_i, f^i(y)\rangle\right\}_{i\in[3n]}$, where $\langle ., .\rangle$ denotes the inner product in $\mathbb{F}_2$. $\mathcal{C}_y$ outputs 1 on a point x if $\forall i \in [3n] : b_i = \langle r_i, f^i(x)\rangle$; otherwise, $\mathcal{C}_y$ outputs 0.*

The proof of security follows similar ideas to the proof in Wee, we defer the details to a later extended version of this work.

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 2.6) for multibit point circuits and circular point circuits are called *digital lockers* (DLs) and *circular digital lockers* (CDLs).

Definition 2.9 (Digital locker (DL))**.** *A PPT algorithm is a DL if it is a recognizable obfuscator (according to Definition 2.5) for the following circuit ensemble:* $\mathcal{C} = \{\mathcal{C}_n = \{I_{s\to t} | s,t \in \{0,1\}^n\}\}_{n\in\mathbb{N}}$

Definition 2.10 (Circular digital locker (CDL))**.** *A PPT algorithm is a CDL if it a recognizable obfuscator (according to Definition 2.5) for the following 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. We now explain how these can be constructed based on a worst-case $AIPO$ that satisfy the additional property of *composability*.

Definition 2.11 (Composable obfuscation [LPS04])**.** *A PPT* $\mathcal{O}$ *is a* t-composable obfuscator *for a circuit ensemble* $\mathcal{C} = \{\mathcal{C}_n\}$ *if for any PPT adversary* $\mathcal{A}$*, there is a PPT simulator* $\mathcal{S}$*, such that for any sufficiently large* n*, any sequence of circuits* $C^1, \ldots, C^t \in \mathcal{C}_n$ *(where* $t = \mathrm{poly}(n)$*) and auxiliary input* $z \in \{0,1\}^{\mathrm{poly}(n)}$*:*

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

where $C^1, \ldots, C^t$ *gets as input* (x, i) *and returns* $C^i(x)$*.*

Composable point obfuscators yield a natural construction of DLs[CD08].

Construction 2.3 (Digital lockers)**.** *Let* $\mathcal{O}$ *be a point obfuscator. Define a PPT* DL *for point circuits with* n*-bit output as follows. For a point* $x \in \{0,1\}^n$ *and output* $y = y_1 y_2 \ldots y_n \in \{0,1\}^n$*, choose a random* $u \in \{0,1\}^n - \{x\}$ *and define* $\bar{a} = (a_0, a_1, \ldots, a_n)$ *as follows.* $a_0 = x$*, and for any* $i \in [n]$ $a_i = x$ *if* $y_i = 1$ *and* $a_i = u$ *otherwise. The output of the obfuscator is:*

$$\mathsf{DL}(I_{x\to y}) = \mathcal{C}(\mathcal{O}(C_{a_0}), \ldots, \mathcal{O}(C_{a_n}))\,,$$

where $\mathcal{C}$ *is a circuit which performs as follows. On input* z*, it first checks whether* $z = a_0 = x$ *(using the first point circuit). If it does not, it returns* $\perp$*. Otherwise, it finds all other coordinates such that* $a_i = z = x$ *and outputs* $y_1 \ldots y_n$*, where* $y_i = 1$ *if* $a_i = z = x$ *and* 0 *otherwise.*

Proposition 2.1 ([CD08])**.** *If* $\mathcal{O}$ *is an* $(n+1)$*-composable worst-case AIPO then* DL *(given by Construction 2.3) is a digital locker.*

Proof. The proof of the functionality, polynomial slow down, and virtual black box properties of DL, appears in [CD08]. It is left to prove that DL is recognizable. Let $\mathbb{V}_{\mathcal{O}}$ be the recognition algorithm for (the single-bit output) $\mathcal{O}$. We construct a recognition algorithm $\mathbb{V}_{\mathsf{DL}}$ for DL. $\mathbb{V}_{\mathsf{DL}}$ is given a program $I_{s\to t}$ and an obfuscated circuit C. First, $\mathbb{V}_{\mathsf{DL}}$ verifies that the format of C is correct; i.e., that C contains $n+1$ circuits $\tilde{\mathcal{O}}_0, \ldots, \tilde{\mathcal{O}}_n$ and performs according to Construction 2.3 (we assume that a legal obfuscation always have the same canonical form). If this is the case, it checks whether $\mathbb{V}_{\mathcal{O}}(I_s, \tilde{\mathcal{O}}_0) = 1$, if so it checks that $C(s) = t$. In case any of the above checks fails, it outputs $\perp$; otherwise, it outputs 1. Note that while $\mathbb{V}_{\mathsf{DL}}$ might output 1 on circuits that are not a proper obfuscation (for example the circuits $\tilde{\mathcal{O}}_i$ corresponding to "0" might encode different points, rather than the same u, or even not be point circuits). However, it is guaranteed that C has the same functionality as $I_{s\to t}$. □

Construction 2.4 (Circular digital lockers)**.** *Given* DL *specified by Construction 2.3, define a PPT* CDL *that on input points* $s, t \in \{0,1\}_n$ *outputs the following circuit:*

$$\mathsf{CDL}(I_{s\leftrightarrows t}) = \mathcal{C}(\mathsf{DL}(I_{s\to t}), \mathsf{DL}(I_{t\to s}))\,,$$

where $\mathcal{C}$ *is a circuit that returns* $\perp$ *if both* DL*s output* $\perp$ *and otherwise it the output of the* DL *that does not output* $\perp$*.*

Proposition 2.2. *If $\mathcal{O}$ is an $(2n+2)$-composable worst-case AIPO then* CDL *(given by Construction 2.4) is a circular digital locker.*

Proof. It follows directly from the construction that CDL satisfies the functionality and polynomial slow down properties. We show that CDL satisfies the virtual black box property. Let $\mathcal{A}$ be an adversary that is given a circuit CDL as input. This CDL contains a pair of DLs constructed from $\mathcal{O}$ according to Construction 2.3. In [CD08] it is shown that if $\mathcal{O}$ is $2(n+1)$-composable, the pair of DLs are 2-composable and therefore there exist a simulator $\mathcal{S}$ such that for all sufficiently large $n \in \mathbb{N}$, every $s, t \in \{0,1\}^n$ and $z \in \{0,1\}^{\mathrm{poly}(n)}$:

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

Now we transform $\mathcal{S}$ to be the obfuscation simulator for $\mathcal{A}$ by answering all queries $\mathcal{S}$ makes to the pair of oracles $I_{s\to t}, I_{t\to s}$ using the single oracle $I_{s \leftrightarrows t}$.

It is left to prove that CDL is recognizable. Let $\mathbb{V}_{\mathsf{DL}}$ be the recognition algorithm for the underlying DL. Given a program $I_{s \leftrightarrows t}$ and an obfuscated circuit C, the recognition algorithm for the CDL will simply verify that C is correctly composed of two DLs and use $\mathbb{V}_{\mathsf{DL}}$ to verify that these DLs have the same functionality as $I_{s\to t}, I_{t\to s}$. □

3 3-round WH

3.1 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 it's last message.

To make the protocol sound, we augment it as follows. Let $\mathsf{Ver}^y_{\mathcal{L},x}$ be a circuit which 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 IA.*

Protocol 1

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. $\mathcal{O}(\tilde{y})$ is a valid point obfuscation of y.

Figure 1: Protocol 1, 3-round Witness Hiding

3.2 Soundness

3.2.1 Overview of the proof.

The protocol presented is an IA. Later we show how to modify the protocol to get an IP. 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 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 valid point obfuscation of y and can only succeed with negligible probability.

3.2.2 Proof of Theorem 3.1 - soundness.

Proof. Let $\mathcal{P}^*$ be any poly-size prover strategy. Let c be the first message of $\mathcal{P}^*$, let y be the random point sampled by $\mathcal{V}$, and let $\hat{c} = \mathsf{Eval}(c, \mathsf{Ver}^y_{\mathcal{L},x})$ be the corresponding message sent by $\mathcal{V}$. Assume towards contradiction that for infinitely many $x \notin \mathcal{L} \cap \{0,1\}^n$:

$$\Pr_{\mathcal{V}}\left[(\mathcal{P}^*, \mathcal{V})(x) = 1\right] \geq \epsilon(n)$$

For some non-negligible function ϵ. That is, $\mathcal{P}^*$ manages to send $\mathcal{V}$ a circuit $\tilde{C}$ such that $\mathbb{V}(I_y, \tilde{C}) = 1$, namely a circuit with the same functionality as I_y.

Since $x \notin \mathcal{L}$, it holds that $\mathsf{Ver}^y_{\mathcal{L},x}(w) = \perp$ for all $w \in \{0,1\}^{\mathrm{poly}(n)}$. Hence, by the function hiding of the underlying delegation scheme, there exist a PPT simulator $\mathcal{S}$ such that $(I_y, \mathcal{S}(c, \perp))$ is indistinguishable from $(I_y, \hat{c})$. We now consider a simulated verifier $\mathcal{V}'$, which given the first message c returns a simulated evaluation $\mathcal{S}(c, \perp)$. By the simulation guarantee it follows that:

$$\Pr_{\mathcal{V}}\left[(\mathcal{P}^*, \mathcal{V}')(x) = 1\right] \geq \epsilon(n) - \mathrm{negl}(n)$$

Otherwise, it is possible to distinguish between the distributions $(I_y, \hat{c})$ and $(I_y, \mathcal{S}(c, \perp))$ as follows. Given a sample $(I_y, \hat{c})$, the distinguisher runs $\mathcal{P}^*$, gives it $\hat{c}$ as the second message and obtains the circuit $\tilde{C}$ returned as the third message. The distinguisher outputs $\mathbb{V}(I_y, \tilde{C})$.

To complete the proof, note that when interacting with $\mathcal{V}'$, $\mathcal{P}^*$'s view is completely independent of the random point y. Hence, $\mathcal{P}^*$ can not produce a circuit $\tilde{C}$ with the same functionality as I_y w.p. greater than $2^{-|y|}$. □

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" described in Section 3 to show why the protocol is not ZK. 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 bit by bit using rewinding.

Proof. Let $\mathcal{P}^*$ be a prover strategy s.t. $\Pr[(\mathcal{P}^*, \mathcal{V})(x) = 1] \geq 1/p(n)$ for some polynomial p and $x \in \mathcal{L} \cap \{0,1\}^n$ for large enough n. We construct an oracle machine E s.t. $E^{\mathcal{P}^*}(x) \in \mathcal{R}_{\mathcal{L}}(x)$. Let c be a random variable representing the first message sent by $\mathcal{P}^*(x)$. Let $G(c)$ be the event that there exist $w \in \mathcal{R}_{\mathcal{L}}(x)$ and sk in the range of Gen s.t. $c = \mathsf{Enc}(sk, w)$. It follows from the soundness proof that $\Pr[(\mathcal{P}^*, \mathcal{V})(x) = 1 | \neg G(c)] < \mathrm{negl}(n)$. But since $\mathcal{P}^*$ convince $\mathcal{V}$ with noticeable probability, $\Pr[(\mathcal{P}^*, \mathcal{V})(x) = 1 \wedge G(c)] \geq 1/q(n)$ for some polynomial q. We show how to reconstruct w bit by bit. For every $i \in [n]$ and every two points $y_0, y_1 \in \{0,1\}^n$ let $B_{i,y_0,y_1}(w)$ be a circuit that outputs y_b where b is the i'th bit of w. If $G(c)$ holds, Consider the messages $\hat{c} = \mathsf{Eval}(c, \mathsf{Ver}^{y_b}_{\mathcal{L},x})$ and $\hat{b} = \mathsf{Eval}(c, B_{i,y_0,y_1})$ for randomly selected points y_0, y_1. Since $G(c)$ holds, it follows from the function hiding property of the delegation scheme that $(y_b, \hat{c})$ and $(y_b, \hat{b})$ are both indistinguishable from $(y_b, \mathcal{S}(c, y_b))$. We denote by $\mathcal{P}^*_m$ the obfuscation sent by $\mathcal{P}^*$ as the third message when given m as the verifier's message. Since y_b is a random point, $\mathcal{P}^*_{\hat{c}}$ is distributed the same as the obfuscation sent by $\mathcal{P}^*$ in a real interaction with $\mathcal{V}$. On one hand, since $\Pr[(\mathcal{P}^*, \mathcal{V})(x) = 1 \wedge G(c)] \geq 1/q(n)$ then also $\Pr[\mathbb{V}(I_{y_b}, \mathcal{P}^*_{\hat{c}}) = 1 \wedge G(c)] \geq 1/q(n)$ and therefore $\Pr[\mathbb{V}(I_{y_b}, \mathcal{P}^*_{\hat{b}}) = 1 \wedge G(c)] \geq 1/q(n)$. On the other hand, $\Pr[\mathbb{V}(I_{y_{1-b}}, \mathcal{P}^*_{\hat{c}}) = 1] \leq \mathrm{negl}(n)$ since the view of $\mathcal{P}^*$ is independent of y_{1-b} and therefore also $\Pr[\mathbb{V}(I_{y_{1-b}}, \mathcal{P}^*_{\hat{b}}) = 1] \leq \mathrm{negl}(n)$. Give that $G(c)$ holds, $E^{\mathcal{P}^*}$ can sample y_0, y_1 and $\mathcal{P}^*_{\hat{b}}$ enough times and learn b for every $i \in [n]$. Since $\Pr[G(c)] \geq 1/q(n)$, the expected running time of E is polynomial in p. □

3.3 Witness Hiding

3.3.1 Overview of the proof.

The WH property is based on the input hiding of the delegation scheme, DEL and the indistinguishability w.r.t unpredictable distributions guarantee of the AIPO, $\mathcal{O}$. Concretely, we show how any $\mathcal{V}^*$ which 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 which chooses its input circuit maliciously to gain information on w. The second, corresponds to a verifier which chooses its circuit honestly. To simulate the obfuscation in the second case, we apply the a prediction circuit $y \leftarrow \Pi(x, c)$ and feed $\mathcal{V}^*$ with $\mathcal{O}(y)$. In the case that y is unpredictable, we obfuscate a random point $\mathcal{O}(u)$. 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 the case, $b = 1$, the reduction is unlike to produce a valid witness, as 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 w.r.t 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.

3.3.2 Proof of Theorem 3.1 - witness hiding.

Proof. Assume towards contradiction there exist a poly-size adversary $\mathcal{V}^*$ and a hard distribution $\mathcal{D}$ on $\mathcal{L}$, such that for a non-negligible ϵ and infinitely many $n \in \mathbb{N}$:

$$\Pr_{(x,w)\overset{D_n}{\leftarrow}\mathcal{R}_\mathcal{L}}[(\mathcal{P}(w),\mathcal{V}^*)(x) \in \mathcal{R}_\mathcal{L}(x)] \geq \epsilon(n) \tag{1}$$

We construct a poly-size adversary that breaks the input hiding property of the delegation scheme. Denote by $\mathcal{V}_1^*$ the circuit which on input $z = (x, c)$, outputs $\mathcal{V}^*$'s message after it is given x as input and c as the prover's first message. Denote by $\mathcal{V}_2^*$ the circuit which on input $(z, \mathcal{O}(y))$, outputs $\mathcal{V}^*$'s output, after it is given x as input, c as the first prover message and $\mathcal{O}(y)$ as the second prover message. We define the following distribution ensemble.

$$\left\{ S_n = (Z_n, Y_n) : \begin{array}{c} (x,w) \overset{D_n}{\leftarrow} \mathcal{R}_\mathcal{L}, sk \leftarrow \mathsf{Gen}(1^n) c \leftarrow \mathsf{Enc}(sk, x), \\ \hat{c} = \mathcal{V}_1^*(x,c), \tilde{s} = \mathsf{Dec}(sk, \hat{c}) \\ Z_n = (x,c), Y_n = \tilde{s} \end{array} \right\}_{n\in\mathbb{N}}$$

Intuitively, any instance of S_n corresponds to an execution of $(\mathcal{P}, \mathcal{V}^*)$ on input x sampled from $\mathcal{D}$, where $Z = (x, c)$ are the input and first message, and Y is the point obfuscated in the last message. Let $\mathbb{I} \subseteq \mathbb{N}$ be the infinite set of indices $n \in \mathbb{N}$ for which (1) holds. By the definition of S_n, for all $n \in \mathbb{I}$:

$$\Pr_{Z_n,Y_n,\mathcal{O}}[\mathcal{V}_2^*(Z_n, \mathcal{O}(Y_n)) \in \mathcal{R}_\mathcal{L}(x)] \geq \epsilon(n) \tag{1}$$

Let $G(z, y)$ be the event that $\Pr_\mathcal{O}[\mathcal{V}_2^*(z, \mathcal{O}(y)) \in \mathcal{R}_\mathcal{L}(x)] \geq \frac{\epsilon(n)}{2}$. By (1) follows that:

Claim 3.1. *For all* $n \in \mathbb{I}$, $\Pr_{Z_n,Y_n}[G(Z_n, Y_n)] \geq \frac{\epsilon(n)}{2}$.

Consider the distribution ensemble $S^G = \left\{S_n^G = (Z_n^G, Y_n^G)\right\}_{n\in\mathbb{I}}$ where S_n^G is the distribution S_n conditioned on the occurrence of G. We distinguish between the case that the distribution ensemble S^G is unpredictable, and the case that it is not.

Case 1 - S^G is predictable. In this case there exist an efficient predictor Π, a non-negligible function δ and an infinite set $\mathbb{I}^G \subseteq \mathbb{I}$ such that for all $n \in \mathbb{I}^G$:

$$\Pr_{(z,y)\leftarrow S_n^G}[\Pi(z) = y] \geq \delta(n) \tag{1}$$

We describe an adversary $\mathcal{A}_1$ that breaks the input hiding property of DEL. $\mathcal{A}_1(1^n)$ will sample $(x, w) \leftarrow D_n$, output the two messages $m_0 = 0^{|w|}, m_1 = w$, and will receive back a challenge c. It will then invoke

$\mathcal{V}_2^*((x,c),\mathcal{O}(\Pi(x,c)))$. In case the output of $\mathcal{V}_2^*$ is in $\mathcal{R}_{\mathcal{L}}(x)$, $\mathcal{A}_1$ will guess $b=1$ otherwise it will guess b at random. Indeed, for all $n \in \mathbb{I}^G$, in case that $b=1$:

$$\Pr_{\substack{(x,w)\leftarrow D_n,\mathcal{O}\\ \mathsf{Gen},\mathsf{Enc}}}[sk=\mathsf{Gen}(1^n), z=(x,\mathsf{Enc}(sk,w)), \mathcal{V}_2^*(z,\mathcal{O}(\Pi(z)))\in\mathcal{R}_{\mathcal{L}}(x)] = \tag{1}$$

$$\Pr_{Z_n,\mathcal{O}}[\mathcal{V}_2^*(Z_n,\mathcal{O}(\Pi(Z_n)))\in\mathcal{R}_{\mathcal{L}}(x)] \geq \tag{2}$$

$$\frac{\epsilon(n)}{2}\cdot\Pr_{Z_n^G,\mathcal{O}}\left[\mathcal{V}_2^*(Z_n^G,\mathcal{O}(\Pi(Z_n^G)))\in\mathcal{R}_{\mathcal{L}}(x)\right] \geq \tag{3}$$

$$\frac{\epsilon(n)\delta(n)}{2}\cdot\Pr_{Z_n^G,Y_n^G,\mathcal{O}}[\mathcal{V}_2^*(Z_n,\mathcal{O}(Y_n))\in\mathcal{R}_{\mathcal{L}}(x)] \geq \frac{\epsilon^2(n)\delta(n)}{4} \tag{4}$$

Where (1) follows from the definition of Z_n, (2) follows from the definition of Z_n^G and Claim 3.1, (3) follows from (1), and (4) is due to the way we defined the event G. It follows that whenever $b=1$, A_1 guesses b with non-negligible advantage.

On the other hand, we show that when $b=0$, $\mathcal{A}_1$ guesses b only with a negligible advantage. Indeed, since $\mathcal{D}$ is hard for $\mathcal{L}$, for all large enough n:

$$\Pr_{\substack{(x,w)\leftarrow D_n,\mathcal{O}\\ \mathsf{Gen},\mathsf{Enc}}}\left[sk=\mathsf{Gen}(1^n), z=(x,\mathsf{Enc}(sk,0^{|w|})), \mathcal{V}_2^*(z,\mathcal{O}(\Pi(z)))\in\mathcal{R}_{\mathcal{L}}(x)\right] \leq \mathrm{negl}(n)$$

Overall, A_1 breaks the input hiding property of DEL with a non-negligible advantage.

Case 2 - S^G is unpredictable. By the definition of S^G it holds that for all $n \in \mathbb{I}^G$.

$$\Pr_{Z_n^G,Y_n^G,\mathcal{O}}\left[\mathcal{V}_2^*(Z_n^G,\mathcal{O}(Y_n^G))\in\mathcal{R}_{\mathcal{L}}(x)\right] \geq \frac{\epsilon(n)}{2}$$

Since $\mathcal{O}$ is a secure point obfuscator for unpredictable distributions, it holds that for all large enough $n \in \mathbb{I}^G$:

$$\Pr_{\substack{Z_n^G,\mathcal{O}\\ y\in_R\{0,1\}^n}}\left[\mathcal{V}_2^*(Z_n^G,\mathcal{O}(y))\in\mathcal{R}_{\mathcal{L}}(x)\right] \geq \frac{\epsilon(n)}{2}-\mathrm{negl}(n) \tag{1}$$

Otherwise, $\mathcal{V}_2^*$ can be used to break the security of $\mathcal{O}$. Similarly to case 1, we describe an adversary $\mathcal{A}_2$ that breaks the input hiding property of DEL. $\mathcal{A}_2(1^n)$ samples $(x,w)\leftarrow D_n$, outputs the two messages $m_0=0^{|w|}, m_1=w$, and receives back a challenge c. It then samples $y \xleftarrow{U} \{0,1\}^n$ and invokes $\mathcal{V}_2^*((x,c),\mathcal{O}(y))$. In case the output of $\mathcal{V}_2^*$ is in $\mathcal{R}_{\mathcal{L}}(x)$, $\mathcal{A}_2$ guesses $b=1$. Otherwise, it guesses b at random. In case $b=1$, it holds for $n \in \mathbb{I}$:

$$\Pr_{\substack{(x,w)\leftarrow D_n,\mathcal{O}\\ \mathsf{Gen},\mathsf{Enc}\\ y\in_R\{0,1\}^n}}[sk=\mathsf{Gen}(1^n), z=(x,\mathsf{Enc}(sk,w)), \mathcal{V}_2^*(z,\mathcal{O}(y))\in\mathcal{R}_{\mathcal{L}}(x)] = \tag{1}$$

$$\Pr_{Z_n,\mathcal{O},y\in_R\{0,1\}^n}[\mathcal{V}_2^*(Z_n,\mathcal{O}(y))\in\mathcal{R}_{\mathcal{L}}(x)] \geq \tag{2}$$

$$\frac{\epsilon(n)}{2}\cdot\Pr_{Z_n^G,\mathcal{O},y\in_R\{0,1\}^n}\left[\mathcal{V}_2^*(Z_n^G,\mathcal{O}(y))\in\mathcal{R}_{\mathcal{L}}(x)\right] \geq \frac{\epsilon^2(n)}{4}-\mathrm{negl}(n) \tag{3}$$

Where (1) follows from the definition of Z_n, (2) follows from the definition of Z_n^G and Claim 3.1, and (3) follows from (1). It follows that whenever $b=1$, A_2 guesses b with non-negligible advantage.

On the other hand, we show that when $b = 0$ $\mathcal{A}_2$ guesses b only with negligible advantage. Indeed, since $\mathcal{D}$ is hard for $\mathcal{L}$, then for all large enough n:

$$\Pr_{\substack{(x,w)\leftarrow D_n,\mathcal{O}\\ y\in_R\{0,1\}^n},\mathsf{Gen},\mathsf{Enc}}\left[sk = \mathsf{Gen}(1^n), \mathcal{V}_2^*((x, \mathsf{Enc}(sk, 0^{|w|}), \mathcal{O}(y)) \in \mathcal{R}_{\mathcal{L}}(x)\right] \leq \mathrm{negl}(n)$$

Overall, A_2 breaks the input hiding property of DEL with non-negligible advantage. □

3.4 From an Argument to a Proof

We modify Protocol 1 and obtain a WH IP. Note that the computational soundness proof relies only an the function hiding property of DEL. To obtain IP, we use a 2-massage delegation protocol with information-theoretic function-hiding (see Section 2.2.1). We would like to use the construction of a 2-message delegation protocol with information-theoretic function-hiding that is described in Section 2.2. However, this construction only allows to evaluate circuits of logarithmic depth while the circuit $\mathsf{Ver}^y_{\mathcal{L},x}$ might not be such. To solve this we use a techniques similar to [AIR01]. Note that except for the point y selected by the verifier, all of the circuit $\mathsf{Ver}^y_{\mathcal{L},x}$ is public. to construct a new circuit $\tilde{\mathsf{Ver}}^y_{\mathcal{L},x}$ of logarithmic depth that has the same functionality as $\mathsf{Ver}^y_{\mathcal{L},x}$. The input wires of $\tilde{\mathsf{Ver}}^y_{\mathcal{L},x}$ correspond to all the wires including internal wires) of the circuit $\mathsf{Ver}_{\mathcal{L},x}$. For every gate G of $\mathsf{Ver}_{\mathcal{L},x}$ that takes two input wires w_1, w_2 and outputs the wire w_3 we check in $\tilde{\mathsf{Ver}}^y_{\mathcal{L},x}$ the condition $w'_3 = G(w'_1, w'_2)$ where w'_1, w'_2, w'_3 are the corresponding inputs of $\tilde{\mathsf{Ver}}^y_{\mathcal{L},x}$. Let w'_O be input wire of $\tilde{\mathsf{Ver}}^y_{\mathcal{L},x}$ corresponding to the output wire of $\mathsf{Ver}_{\mathcal{L},x}$. The output of $\tilde{\mathsf{Ver}}^y_{\mathcal{L},x}$ will be y if $w'_O = 1$ and all of conditions hold. Otherwise, $\tilde{\mathsf{Ver}}^y_{\mathcal{L},x}$ outputs $\perp$. We modify Protocol 1 as follows. Instead of the executing a 2-message delegation protocol where the prover inputs a witness w and the verifier inputs the circuit $\mathsf{Ver}^y_{\mathcal{L},x}$, execute a 2-message delegation protocol with information-theoretic function-hiding where the prover inputs the values of all wires of the circuit $\mathsf{Ver}_{\mathcal{L},x}$ evaluated on w and the verifier inputs the circuit $\tilde{\mathsf{Ver}}^y_{\mathcal{L},x}$.

Theorem 3.2. *Let* DEL *be a secure 2-message delegation protocol with information-theoretic function hiding, and let $\mathcal{O}$ be an AIPO. The modified Protocol is a WH IP.*

Proof. In the soundness proof of Theorem 3.1 we only rely on the function hiding property of the the 2-message delegation protocol and on the recognizability property of the obfuscation. If the 2-massage delegation protocol has information-theoretic function hiding, even an unbounded prover will not be able to distinguish a real verifier message from a simulated one. Since the recognizability property holds for every obfuscated circuit, we have that the same arguments used in the soundness proof of Theorem 3.1 hold also in this case. Similarly, The proof of the WH property remains unchanged since the circuits $\tilde{\mathsf{Ver}}^y_{\mathcal{L},x}$ and $\mathsf{Ver}^y_{\mathcal{L},x}$ compute the same function. □

4 3-round WZK

4.1 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 2.10 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" [2], implemented as follows. In its first message, the prover 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 further explain the role of the above modifications.

The protocol. Let $\mathsf{DEL} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Eval}, \mathsf{Dec}, \mathsf{Open})$ be a secure 2-message delegation protocol. 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.

Protocol 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 $r \overset{U}{\leftarrow} \{0,1\}^n$,
   sends c and $f(r)$.
2. $\mathcal{V}$**:** Samples $y \overset{U}{\leftarrow} \{0,1\}^n$,
   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 $u \overset{U}{\leftarrow} \{0,1\}^n$ 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 $\mathbb{V}(I_{y\leftrightarrows\tilde{r}}, \mathsf{CDL}_\mathcal{P}) = 1$.

Figure 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 IA.*

4.2 Soundness

4.2.1 Overview of the proof.

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 as $\hat{c}$, it can also be recovered solely from $\hat{c}$. However, since $x \notin \mathcal{L}$ and DEL is function hiding, y can not be recovered from $\hat{c}$ (similarly to the WH protocol).

[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].

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.

4.2.2 Proof of Theorem 4.1 - soundness.

Proof. Let $\mathcal{P}^*$ be a cheating prover, and assume WLOG that $\mathcal{P}^*$ is deterministic Assume that for infinitely many $x \notin \mathcal{L}$, $\mathcal{P}^*$ manages to fool $\mathcal{V}$ with non-negligible probability δ. Let $(c, f(r))$ be the (fixed) first message sent by $\mathcal{P}^*$. We first show that it is hard to recover $\mathcal{V}$'s secret point y from $\mathcal{V}$'s message $(\hat{c}, \mathsf{DL}_{\mathcal{V}})$.

Claim 4.1. *For any poly-size $\mathcal{A}$:*

$$\Pr_{\mathcal{V}}[\mathcal{A}(\hat{c}, \mathsf{DL}_{\mathcal{V}}) = y] \leq \mathrm{negl}(n)$$

Where y is the secret point selected by $\mathcal{V}$, $\hat{c} \leftarrow \mathsf{Eval}(c, \mathsf{Ver}^y_{\mathcal{L},x})$ is the result of Eval *applied by $\mathcal{V}$ using random coins $r_{\mathcal{V}}$, and $\mathsf{DL}_{\mathcal{V}} = \mathsf{DL}(I_{y \to r_{\mathcal{V}}})$ is the* DL *sent by $\mathcal{V}$.*

Proof. Assume towards contradiction there exist a poly-size $\mathcal{A}$ which recovers y with non-negligible probability ϵ. We first consider an adversary $\mathcal{A}'$ which predicts the first bit of y, y_1,. $\mathcal{A}'(\hat{c}, \mathsf{DL}_{\mathcal{V}})$ runs $\mathcal{A}(\hat{c}, \mathsf{DL}_{\mathcal{V}})$ and obtains its output $\tilde{y}$. If $\tilde{y}$ "unlocks" $\mathsf{DL}_{\mathcal{V}}$ (i.e. $\tilde{y} = y$), $\mathcal{A}'$ outputs $\tilde{y}_1$, otherwise it outputs a random bit. By our assumption on $\mathcal{A}$:

$$\Pr_{\mathcal{V}}[\mathcal{A}'(\hat{c}, \mathsf{DL}_{\mathcal{V}}) = y_1] \geq \frac{1}{2} + \epsilon(n)$$

Now, let $\mathcal{S}_{\mathcal{A}'}$ be the obfuscation simulator for $\mathcal{A}'$. By the obfuscation guarantee it holds that:

$$\Pr_{\mathcal{S}_{\mathcal{A}'}, \mathcal{V}}[\mathcal{S}^{\mathsf{DL}_{\mathcal{V}}}_{\mathcal{A}'}(\hat{c}) = y_1] \geq \frac{1}{2} + \epsilon - \mathrm{negl}(n)) \tag{1}$$

Since $x \notin \mathcal{L}$, $\mathsf{Ver}^y_{\mathcal{L},x}(w) \equiv \perp$ and hence by the function hiding property of Eval there exist a PPT simulator $\mathcal{S}$ such that:

$$\hat{c} \doteq \left\{\mathsf{Eval}(c, \mathsf{Ver}^y_{\mathcal{L},x})\right\}, (\hat{c}, \mathsf{DL}_{\mathcal{V}}) \approx_c \{(\mathcal{S}(c, \perp), \mathsf{DL}_{\mathcal{V}})\} \tag{2}$$

We now claim that $\mathcal{S}_{\mathcal{A}'}$ does not query its oracle on the point y except with negligible chance. Otherwise, $\mathcal{S}_{\mathcal{A}'}$ can be used to predict y from $\hat{c}$ with noticeable probability. (2) implies that $\mathcal{S}_{\mathcal{A}'}$ can also predict y from $\mathcal{S}(c, \perp)$ which is in turn independent of y, resulting in a contradiction.

Putting this together with (1) implies:

$$\Pr_{\mathcal{S}_{\mathcal{A}'}, \mathcal{V}}[\mathcal{S}^{\perp}_{\mathcal{A}'}(\hat{c}) = y_1] \geq \frac{1}{2} + \epsilon - \mathrm{negl}(n)$$

where $\perp$ is the oracle that answers $\perp$ on all queries. It follows that $\mathcal{S}^{\perp}_{\mathcal{A}'}$ can be used to predict y_1 from $\hat{c}$ with noticeable advantage. In addition, by (2), $\mathcal{S}^{\perp}_{\mathcal{A}'}$ will also be able to recover y by applying $\mathcal{S}(c, \perp)$ which is independent of y, leading once again to a contradiction. Completing the proof of Claim 4.1. □

To complete the proof, we use the cheating prover $\mathcal{P}^*$ to construct a poly-size $\mathcal{A}$ which recovers y from $(\hat{c}, \mathsf{DL}_{\mathcal{V}})$. $\mathcal{A}$ will have the point r hardwired in to it. It will run $\mathcal{P}^*$ and feed it with $(\hat{c}, \mathsf{DL}_{\mathcal{V}})$. $\mathcal{P}^*$ then outputs $\mathsf{CDL}_{\mathcal{P}^*}$, and $\mathcal{A}$ outputs $\mathsf{CDL}_{\mathcal{P}^*}(r)$. By our assumption on $\mathcal{P}^*$, with probability at least δ, $\mathcal{V}$ accepts $\mathsf{CDL}_{\mathcal{P}^*}$, implying that $\mathsf{CDL}_{\mathcal{P}^*}$ is an obfuscation of $I_{y \hookrightarrow \tilde{r}}$ such that $f(\tilde{r}) = f(r)$ or equivalently, $\tilde{r} = r$ (as f is injective). Hence, $\mathcal{A}$ also manages to recover y w.p. at least δ contradicting Claim 4.1. □

4.3 Weak Zero-Knowledge

4.3.1 Overview of the proof.

We present a WZK simulator that given an adversary $\mathcal{V}^*$ and a distinguisher D, simulates the view of V^* w.r.t D. Let $\mathcal{V}_D^*$ be the composition of D with $\mathcal{V}^*$. $\mathcal{V}_D^*$ 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 $\mathcal{S}_{\mathsf{CDL}}$ which simulates $\mathcal{V}_D^*$'s output given oracle access to $I_{y \leftrightarrows r}$ and auxiliary input $\mathsf{ai} = (z, x, c, f(r))$, representing the rest of $\mathcal{V}_D^*$'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 which "knows" the secret point y. In this case, our simulator can perfectly simulate the behavior of $\mathcal{P}$. That is, "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 $\mathcal{V}^*$, which 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 full description of the simulator as well as the proof of its validity are provided in Section **??**.

4.3.2 Proof of Theorem 4.1 - weak zero-knowledge.

The simulator. Let $\mathcal{V}^*$ be any verifier, and let 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. Denote by $\mathcal{V}_D^*(x, z, c, f(r), \mathsf{CDL}_{\mathcal{P}})$ the algorithm that runs $\mathcal{V}_2^*(x, z, c, f(r), \mathsf{CDL}_{\mathcal{P}})$, applies the circuit D on the output of $\mathcal{V}_2^*$ and returns the output bit of D. Let $\mathcal{S}_{\mathcal{V}^*,D}(x, z, c, f(r))$ be the PPT obfuscation simulator of $\mathcal{V}_D^*$ as specified by Definition 2.5. Also let $\ell(n)$ be the length of a witness for instances of length n. The description of the simulator is given by Algorithm (4.3.2)

Proof of validity. Let $\mathsf{View}_{\mathcal{S}}$ be output distribution of $\mathcal{S}(z, x)$ and let View_{V^*} be the output distribution of $(\mathcal{P}(w), \mathcal{V}^*(z))(x)$. We show that $\{\mathcal{D}(\mathsf{View}_{\mathcal{S}})\} \approx_c \{\mathcal{D}(\mathsf{View}_{\mathcal{V}^*})\}$. We first consider an alternative hybrid simulation process $\mathcal{S}'(z, x, w)$ behaving exactly like $\mathcal{S}(z, x)$ except that to simulate the first message it generates $sk \overset{\mathsf{Gen}}{\leftarrow} (1^n)$ and computes $c \leftarrow \mathsf{Enc}(sk, w)$ instead of $c \leftarrow \mathsf{Enc}(sk, 1^{\ell(|x|)})$. Let $\mathsf{View}_{\mathcal{S}'}$ be output distribution of $\mathcal{S}'(x, z, w)$. By the input hiding property of the delegation protocol, it follows that $\mathsf{View}_{\mathcal{S}} \approx_c \mathsf{View}_{\mathcal{S}'}$. Hence, it suffices to show that $\{\mathcal{D}(\mathsf{View}_{\mathcal{S}'})\} \approx_c \{\mathcal{D}(\mathsf{View}_{\mathcal{V}^*})\}$.

Let $(c, f(r))$ be the first message used by $\mathcal{S}'$. Notice that $(c, f(r))$ is distributed exactly like the first message in the real interaction. We define the following events:

- E_r indicates that the obfuscation simulator $\mathcal{S}_{\mathcal{V}^*,D}(z, x, c, f(r))$ queries its oracle on r.
- E_y indicates that the obfuscation simulator $\mathcal{S}_{\mathcal{V}^*,D}(x, z, c, f(r))$ performs a query Q such that $\mathsf{DL}_{\mathcal{V}}(Q) \neq \perp$

Algorithm 4.1 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 $\mathcal{S}_{\mathcal{V}^*,D}(x, z, c, f(r))$.
for each oracle query Q made by $\mathcal{S}_{\mathcal{V}^*,D}$ **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** $\mathsf{Open}(\hat{c}, \tilde{r}_{\mathcal{V}}) \neq \mathsf{Ver}_{\mathcal{L},x}^{\tilde{y}}$ **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

- E_V indicates that the second message $(\hat{c}, \mathsf{DL}_{\mathcal{V}}) = \mathcal{V}_1^*(x, z, c, f(r))$ is valid, i.e. that $\exists \tilde{y}, \tilde{r}_{\mathcal{V}}$ such that $\mathbb{V}(I_{\tilde{y} \to \tilde{r}_{\mathcal{V}}}, \mathsf{DL}_{\mathcal{V}}) = 1$ and $\mathsf{Open}(\hat{c}, \tilde{r}_{\mathcal{V}}) = \mathsf{Ver}^{\tilde{y}}_{\mathcal{L},x}$.

First, we claim that: $\Pr_{\mathcal{S}'}[E_r] \leq \mathrm{negl}(n)$. Otherwise, we can utilize $\mathcal{S}'$ in order to invert the OWF f. We thus assume henceforth that E_r does not occur. We no treat several cases:

Case 1 - E_V, E_y **both occur.** In this case $\mathcal{S}_{\mathcal{V}^*,D}$ makes an oracle query that opens $\mathsf{DL}_{\mathcal{V}}$, and since $\mathcal{V}^*$'s message is valid, this query is identical to the secret point, allowing $\mathcal{S}'$ to perform perfect simulation. Indeed, in this case both $\mathsf{View}_{\mathcal{V}^*}$ and $\mathsf{View}_{\mathcal{S}'}$ are distributed as $\mathcal{V}_2^*(z, x, c, f(r), \mathsf{CDL}(I_{\tilde{y} \leftrightarrows r}))$, where $\tilde{y}$ is the secret point defined by the event E_V.

Case 2 - E_V **does not occur and** E_y **does.** In this case $\mathcal{S}_{\mathcal{V}^*,D}$ makes an oracle query that opens $\mathsf{DL}_{\mathcal{V}}$, and both the simulator and the real prover detect that $\mathcal{V}^*$'s message is invalid, and reply with a "dummy" CDL. Hence, $\mathsf{View}_{\mathcal{V}^*}$ and $\mathsf{View}_{\mathcal{S}'}$ are distributed as $\mathcal{V}_2^*(z, x, c, f(r), \mathsf{CDL}(I_{u \leftrightarrows u}))$ where u is random, yielding perfect simulation.

Case 3 - both E_y, E_V **do not occur.** In this case, the prover detects that $\mathcal{V}^*$'s message is invalid and sends back a "dummy obfuscation". Since non of $\mathcal{S}_{\mathcal{V}^*,D}$'s queries unlocked $\mathsf{DL}_{\mathcal{V}}$, $\mathcal{S}'$ also produces a "dummy" obfuscation. Hence, both $\mathsf{View}_{\mathcal{V}^*}$ and $\mathsf{View}_{\mathcal{S}'}$ are distributed as $\mathcal{V}_2^*(z, x, c, f(r), \mathsf{CDL}(I_{u \leftrightarrows u}))$ where u is randomly selected, yielding perfect simulation.

Case 4 - E_y **does not occur and** E_V **does.** In this case, the real prover extracts the secret point $\tilde{y}$ defined by the event E_V. $\mathcal{S}'$ on the other hand does not, since non of $\mathcal{S}_{\mathcal{V}^*,D}$'s queries unlocked $\mathsf{DL}_{\mathcal{V}}$, $\mathcal{S}'$ sends a dummy CDL. However, the fact that all of $\mathcal{S}_{\mathcal{V}^*,D}$ queries result in $\perp$ implies that $\mathcal{V}^*$ does not distinguish the CDL used by the simulator from the dummy one used by the prover. More accurately,

$$\{\mathcal{D}(\mathsf{View}_{\mathcal{S}'})\} \approx_c \{\mathcal{D}(\mathcal{V}_2^*(x, z, c, f(r), \mathsf{CDL}(I_{u \leftrightarrows u})))\} \quad (1)$$

$$\approx_c \mathcal{S}^{I_{u \leftrightarrows u}}_{\mathcal{V}^*,D}(x, z, c, f(r)) \quad (2)$$

$$\approx_c \mathcal{S}^{I_{\tilde{y} \leftrightarrows r}}_{\mathcal{V}^*,D}(x, z, c, f(r)) \quad (3)$$

$$\approx_c \{\mathcal{D}(\mathcal{V}_2^*(x, z, c, f(r), \mathsf{CDL}(I_{\tilde{y} \leftrightarrows r})))\} \quad (4)$$

$$\approx_c \{\mathcal{D}(\mathsf{View}_{\mathcal{V}^*})\} \quad (5)$$

Where (2), (4) follows by the obfuscation guarantee (Definition 2.5), and (3) holds since $\mathcal{S}_{\mathcal{V}^*,D}$ does not query $\tilde{y}$ nor r, and queries the random u only with negligible probability.

Putting together all the cases, yields $\{\mathcal{D}(\mathsf{View}_{\mathcal{S}'})\} \approx_c \{\mathcal{D}(\mathsf{View}_{\mathcal{V}^*})\}$ as required. □

Composition. We do not know whether Protocol 2 remains secure under sequential composition. On the other hand, given that $\mathsf{DL}, \mathsf{CDL}$ are composable obfuscators (see Definition 2.11), Protocol 2 remains secure under parallel composition.

Proposition 4.1 (Informal)**.** *Given that* $\mathsf{DL}, \mathsf{CDL}$ *are* t*-composable obfuscators, Protocol 2 remains secure when* t *parallel copies of the protocol.*

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 uses of conditional disclosure of secrets.

References

[AIR01] William Aiello, Yuval Ishai, and Omer Reingold, *Priced oblivious transfer: How to sell digital goods*, Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques: Advances in Cryptology (London, UK), EUROCRYPT '01, Springer-Verlag, 2001, pp. 119–135.

[Bar01] Boaz Barak, *How to go beyond the black-box simulation barrier*, FOCS, 2001, pp. 106–115.

[BC10] Nir Bitansky and Ran Canetti, *On strong simulation and composable point obfuscation*, CRYPTO, 2010, pp. 520–537.

[BCC88] Gilles Brassard, David Chaum, and Claude Crépeau, *Minimum disclosure proofs of knowledge*, J. Comput. Syst. Sci. **37** (1988), no. 2, 156–189.

[BGI+01] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang, *On the (im)possibility of obfuscating programs*, CRYPTO, 2001, pp. 1–18.

[BP04] Mihir Bellare and Adriana Palacio, *The knowledge-of-exponent assumptions and 3-round zero-knowledge protocols*, CRYPTO, 2004, pp. 273–289.

[Can97] Ran Canetti, *Towards realizing random oracles: Hash functions that hide all partial information*, CRYPTO, 1997, pp. 455–469.

[CCKM00] Christian Cachin, Jan Camenisch, Joe Kilian, and Joy Müller, *One-round secure computation and secure autonomous mobile agents*, ICALP, 2000, pp. 512–523.

[CD08] Ran Canetti and Ronny Ramzi Dakdouk, *Obfuscating point functions with multibit output*, EUROCRYPT, 2008, pp. 489–508.

[CD09] ———, *Towards a theory of extractable functions*, TCC, 2009, pp. 595–613.

[CKVW10] Ran Canetti, Yael Kalai, Mayank Varia, and Daniel Wichs, *On symmetric encryption and point obfuscation*, TCC, 2010, pp. 52–71.

[CV08] Ran Canetti and Mayank Varia, *Non-malleable obfuscation*, Cryptology ePrint Archive, Report 2008/495, 2008, `http://eprint.iacr.org/`.

[Dam91] Ivan Damgård, *Towards practical public key systems secure against chosen ciphertext attacks*, CRYPTO, 1991, pp. 445–456.

[DKL09] Yevgeniy Dodis, Yael Tauman Kalai, and Shachar Lovett, *On cryptography with auxiliary input*, STOC, 2009, pp. 621–630.

[DNRS99] Cynthia Dwork, Moni Naor, Omer Reingold, and Larry J. Stockmeyer, *Magic functions*, FOCS, 1999, pp. 523–534.

[FS90] Uriel Feige and Adi Shamir, *Witness indistinguishable and witness hiding protocols*, STOC, 1990, pp. 416–426.

[GIKM00] Yael Gertner, Yuval Ishai, Eyal Kushilevitz, and Tal Malkin, *Protecting data privacy in private information retrieval schemes*, JCSS, ACM Press, 2000, pp. 151–160.

[GK96] Oded Goldreich and Hugo Krawczyk, *On the composition of zero-knowledge proof systems*, SIAM J. Comput. **25** (1996), no. 1, 169–192.

[GK05] Shafi Goldwasser and Yael Tauman Kalai, *On the impossibility of obfuscation with auxiliary input*, FOCS, 2005, pp. 553–562.

[GMR85] Shafi Goldwasser, Silvio Micali, and Charles Rackoff, *The knowledge complexity of interactive proof-systems (extended abstract)*, STOC, 1985, pp. 291–304.

[GO94] Oded Goldreich and Yair Oren, *Definitions and properties of zero-knowledge proof systems*, J. Cryptology **7** (1994), no. 1, 1–32.

[HRS09] Iftach Haitner, Alon Rosen, and Ronen Shaltiel, *On the (im)possibility of arthur-merlin witness hiding protocols*, TCC, 2009, pp. 220–237.

[HT98] Satoshi Hada and Toshiaki Tanaka, *On the existence of 3-round zero-knowledge protocols*, CRYPTO, 1998, pp. 408–423.

[IK02] Yuval Ishai and Eyal Kushilevitz, *Perfect constant-round secure computation via perfect randomizing polynomials*, In Proc. 29th ICALP, 2002, pp. 244–256.

[IKOS07] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai, *Zero-knowledge from secure multiparty computation*, STOC, 2007, pp. 21–30.

[IP07] Yuval Ishai and Anat Paskin, *Evaluating branching programs on encrypted data*, Proceedings of the 4th conference on Theory of cryptography (Berlin, Heidelberg), TCC'07, Springer-Verlag, 2007, pp. 575–594.

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

[LM01] Matthew Lepinski and Silvio Micali, *On the existence of 3-round zero-knowledge proof systems*, Tech. report, MIT LCS, 2001.

[LP09] Yehuda Lindell and Benny Pinkas, *A proof of security of yao's protocol for two-party computation*, J. Cryptology **22** (2009), no. 2, 161–188.

[LPS04] Ben Lynn, Manoj Prabhakaran, and Amit Sahai, *Positive results and techniques for obfuscation*, EUROCRYPT, 2004, pp. 20–39.

[Nao03] Moni Naor, *On cryptographic assumptions and challenges*, CRYPTO, 2003, pp. 96–109.

[NP01] Moni Naor and Benny Pinkas, *Efficient oblivious transfer protocols*, SODA, 2001, pp. 448–457.

[Wee05] Hoeteck Wee, *On obfuscating point functions*, STOC, 2005, pp. 523–532.