Instance-Hiding Interactive Proofs

Changrui Mu* Prashant Nalini Vasudevan†

Abstract

In an Instance-Hiding Interactive Proof (IHIP) [BFS90], an efficient verifier with a *private* input x interacts with an unbounded prover to determine whether x is contained in a language $\mathcal{L}$. In addition to completeness and soundness, the *instance-hiding* property requires that the prover should not learn anything about x in the course of the interaction. Such proof systems capture natural privacy properties, and may be seen as a generalization of the influential concept of Randomized Encodings [IK00, AIK04, AIKPC15], and as a counterpart to Zero-Knowledge proofs [GMR85].

We investigate the properties and power of such instance-hiding proofs, and show the following:

1. Any language with an IHIP is contained in $\mathsf{NP/poly} \cap \mathsf{coNP/poly}$.
2. If an average-case hard language has a constant-round IHIP, then infinitely-often One-Way Functions exist.
3. There is an oracle with respect to which there is a language that has an IHIP but not an SZK proof.
4. IHIP's are closed under composition with any efficiently computable function.

We further study a stronger version of IHIP (that we call Simulatable IHIP) where the view of the honest prover can be efficiently simulated. For these, we obtain stronger versions of some of the above:

5. Any language with a Simulatable IHIP is contained in $\mathsf{AM} \cap \mathsf{coAM}$.
6. If a *worst-case* hard language has a Simulatable IHIP, then One-Way Functions exist.

*Department of Computer Science, National University of Singapore. Email: changrui.mu@u.nus.edu
†Department of Computer Science, National University of Singapore. Email: prashant@comp.nus.edu.sg

Contents

1 Introduction

An Interactive Proof system (IP) [BM88, GMR85] for a language $\mathcal{L}$ is an interactive protocol between a polynomial-time verifier V and a computationally unbounded prover P where both are given as input a string x, and the prover tries to prove to the verifier that $x \in \mathcal{L}$. Such proofs are required to be complete – if $x \in \mathcal{L}$, the verifier will accept at the end of the interaction – and sound – if $x \notin \mathcal{L}$, no prover strategy P^* can make the verifier accept with large probability. Such proofs are very powerful and have been shown to exist for all languages computable with polynomial space (PSPACE) [LFKN92, Sha92].

Often in applications of interactive proof systems, one or both of the parties may hold some secret that they do not want the other to learn in the course of the interaction. For instance, the prover P may hold a secret key and wish to prove to the verifier something about a ciphertext encrypted using that key, without revealing the key itself. A powerful general formalization of a property that enables this is *Zero-Knowledge (ZK)* [GMR85]. This requires that there be a computationally efficient simulator that, for any $x \in \mathcal{L}$, can simulate the entire view of the verifier V interacting with P on input x. This ensures that the verifier learns nothing from the interaction other than the membership of x in $\mathcal{L}$.

This simulation may be perfect (PZK), statistically close to (SZK), or computationally indistinguishable from (CZK) the actual view. In contrast to general interactive proofs, it is known that languages that have PZK or SZK proofs are contained in $\mathsf{AM} \cap \mathsf{coAM}$ [For87, AH91]. Using computational assumptions, however, one can again construct CZK proofs for all languages in PSPACE [GMW91, BGG+88]. ZK proofs have been and continue to be the subject of extensive research, have found numerous applications in practice, and we understand them quite well (see, e.g., references in [Vad99, Tha22]).

Instance-Hiding Interactive Proofs. Whereas Zero-Knowledge provides security for the prover, *Instance-Hiding Interactive Proofs (IHIP)* [BFS90] provide a similar security guarantee for the verifier. In an IHIP, the input x is given only to the verifier V. Apart from completeness and soundness as in an IP, it is required that the protocol be instance-hiding – for any *prover* strategy P^*, there should exist a simulator (computationally unbounded) that, given just the length of the input, can simulate the view of P^* when interacting with V on any input x. This ensures that the prover cannot learn anything about the input except at most its length.

In other words, the prover proves to the verifier that $x \in \mathcal{L}$ without knowing anything at all about x. Seemingly paradoxical, such proof systems can, in fact, be constructed for several strutured languages, such as those that have certain random self-reduction properties [AFK89, FO91]. Nevertheless, a theorem of Abadi et al. [AFK89] implies that any language that has an IHIP in which the simulation of the prover's view is *perfect* is contained in $\mathsf{NP/poly} \cap \mathsf{coNP/poly}$. In particular, this implies that NP-hard languages do not have perfect IHIP protocols unless the polynomial hierarchy collapses [Yap83].

Despite the fact that they capture this fundamental cryptographic property of protocols, not much else is known about the complexity of IHIP's today, decades after they were first defined. Further, even the aforementioned results do not hold if even a negligible amount of statistical error is allowed in the hiding property – that is, when the prover's view corresponding to any instance can have non-zero but negligibly small statistical distance from the simulator's output. In this paper, we undertake a systematic study of the complexity of such general imperfect IHIP's, with the objective of understanding what properties they have, how powerful they are, what kinds of structure they create, and how they compare to other cryptographic protocols like ZK proofs.

1.1 Our Results

In the rest of the paper, we simply use IHIP to denote instance-hiding IP's that have a negligible statistical hiding error as described above. We also study a strengthening of these proofs where the simulator that simulates the honest prover's view is required to be efficient; we refer to these as *Simulatable IHIP's*. A number of natural constructions of IHIP's do, in fact, have this stronger property (see, e.g., Appendix A). We show a collection of results about various aspects of these proof systems, some of which follow from techniques common in the study of Secure Multi-Party Computation (MPC) and SZK proofs, while others

require the development of new methods. We define these proof systems in Section 2, and in Appendix A we present examples of non-trivial languages that have such proof systems, including one that seems to require multiple rounds of interaction.

Power of Instance-Hiding Proofs. We start by asking which languages can possibly have IHIP's. The results of Abadi et al. [AFK89] imply that languages that have *perfect* IHIP's are contained in $\mathsf{NP/poly} \cap \mathsf{coNP/poly}$. However, their techniques stop working if there is even a small amount of error in the hiding property. Using some carefully designed interactive proofs, we show that any language that has an IHIP with small enough hiding error is still contained in $\mathsf{NP/poly} \cap \mathsf{coNP/poly}$. Further, if the IHIP has an efficient simulator for the honest prover and the simulator and verifier are uniform, we can get rid of the non-uniformity and show that the language lies in $\mathsf{AM} \cap \mathsf{coAM}$.

Informal Theorem 1.1 (Theorems 4.1 and 4.3)**.** *If a language $\mathcal{L}$ has an IHIP, then both $\mathcal{L}$ and its complement $\bar{\mathcal{L}}$ have constant-round public-coin interactive proofs with non-uniform verifiers. Further, if $\mathcal{L}$ has a simulatable IHIP, this conclusion holds with uniform verifiers.*

This too implies that NP-hard problems do not have such proofs unless the polynomial hierarchy collapses [Yap83, FF91]. This upper bound on the power of such proofs is complemented by the existence a number of interesting non-trivial languages do have instance-hiding interactive proofs – see the examples in Appendix A and the connections to Randomized Encodings described below.

Implications for One-Way Functions. Investigating further the implications of non-trivial languages having instance-hiding proofs, we show that the existence of hard languages with some of such proofs implies the existence of One-Way Functions (OWF's).

Informal Theorem 1.2 (Theorems 5.3 and 5.9)**.** *We show the following:*

- *Infinitely-often one-way function from an average-case hard language that has a constant-round IHIP*
- *One-way function from a worst-case hard language that has a simulatable IHIP*

The proof of the first statement in the theorem above is non-constructive – we prove that an OWF exists, but given an explicit average-case hard language with an IHIP, our proof does not construct an explicit function that is one-way.

Relationship with SZK. Both the above properties – membership in $\mathsf{AM} \cap \mathsf{coAM}$ [For87, AH91] and the implication of OWFs from average-case hardness [Ost91] – are shared by the class SZK of languages that have Statistical Zero Knowledge proofs. Intuitively, SZK proofs and IHIP's seem to rely on different properties of the underlying language, and it seems unlikely that one is contained in the other. We provide some evidence for this in the form of an oracle separation between SZK and Simulatable-IHIP, the class of languages that have simulatable IHIP's.

Informal Theorem 1.3 (Theorem 6.4)**.** *There exists an oracle $\mathcal{O}$ such that* $\mathsf{Simulatable\text{-}IHIP}^{\mathcal{O}} \not\subseteq \mathsf{SZK}^{\mathcal{O}}$.

We essentially prove this statement in the Generic Group Model [Sho97] – we show a group problem that cannot be decided by SZK protocols that treat the group in a certain "generic" manner, but can be decided by a similarly generic IHIP protocol. Showing an oracle separation in the other direction is an interesting open problem here.

Closure Properties. Finally, we show that IHIP's have rather strong closure properties – that they are closed under composition with any efficiently computable function. For any language $\mathcal{L}$, and functions $f : \{0,1\}^* \to \{0,1\}$ and $k : \mathbb{N} \to \mathbb{N}$, consider the "composed" language $f \circ \mathcal{L}^{\otimes k}$, in which an instance consists of $k = k(n)$ strings x_i of length n, which is in the language if and only if $f(\mathcal{L}(x_1), \ldots, \mathcal{L}(x_k)) = 1$. That is, given these k strings x_i, first check whether each of these is in $\mathcal{L}$, and then compute the function f on the result of these checks.

Informal Theorem 1.4 (Theorem 2.7)**.** *Consider any language $\mathcal{L}$ that has an IHIP protocol, and any efficiently computable function $f : \{0,1\}^* \to \{0,1\}$. For any polynomial $k : \mathbb{N} \to \mathbb{N}$, the composed language $f \circ \mathcal{L}^{\otimes k}$ also has an IHIP protocol.*

Similar properties for SZK are only known to hold for composition with polynomial-sized *formulas* (or NC^1) [SV97], whereas the above corresponds to polynomial-sized *circuits* (or P). This is another indication that this class is likely to be different from SZK.

The above theorem implies, in particular, that IHIP is closed under complement. While not directly implied by the theorem, our proof also shows that the OR or AND of two languages that have IHIP's also has an IHIP. This extends to the statement that for any constant k and languages $\mathcal{L}_1, \ldots, \mathcal{L}_k$ that all have IHIP's, any language expressible as a function of membership in these languages also has an IHIP.

Our proof of this theorem goes through Instance-Hiding Delegation Schemes [FO91], which are an alternative formulation of instance-hiding protocols that are interesting in their own right. See Section 7 for details.

Useful Tools. In the course of our constructions and proofs, we also show two lemmas that are meaningful outside this context – a protocol for proving lower bounds on a weighted sum of the sizes of sets (Lemma 3.7), and an equivalence between randomized and deterministic advice for AM proofs (Corollary 3.5).

1.2 Technical Overview

We now present an overview of the proofs of Theorems 1.1 to 1.3. The proof of Theorem 1.4 follows from a straightforward extension of existing work together with some standard transformations, so we leave its details to the relevant section.

1.2.1 Instance-Hiding to AM Proofs

Suppose a language $\mathcal{L}$ has an instance-hiding IP with prover P and verifier V – denoted $\langle \mathsf{P}, \mathsf{V} \rangle$. We will use this to construct a constant-round interactive proof $\langle \mathsf{M}, \mathsf{A} \rangle$ with non-uniform verifier A for $\mathcal{L}$. We will then show how to remove the non-uniformity if there is an efficient simulator for the honest prover.

Denote by $r \in \mathcal{R}_{\mathsf{V}}$ the random string used by the verifier. For any input x, denote by S_x the distribution over the transcript $s = (u_1, y_1, \ldots, u_q, y_q)$ generated by the execution of the protocol $\langle \mathsf{P}, \mathsf{V}(x; r) \rangle$, where u_i and y_i denote the verifier's and prover's message in i^{th} round respectively. For simplicity, assume that P is deterministic[1]. Given input x, our approach is to have M prove to A that the probability that $\langle \mathsf{P}, \mathsf{V}(x; r) \rangle$ accepts is large. Towards discussing this, we define the following two sets for any x and transcript s:

$$\beta_s^x = \{r \mid s \text{ is the transcript of } \langle \mathsf{P}, \mathsf{V}(x; r) \rangle\}$$
$$\alpha_s^x = \{r \mid r \in \beta_s^x \text{ and } \langle \mathsf{P}, \mathsf{V}(x; r) \rangle \text{ accepts}\}$$

We start with the observation that the probability of acceptance may be expressed as follows:

$$\Pr_r \left[\langle \mathsf{P}, \mathsf{V}(x; r) \rangle \text{ accepts}\right] = \frac{\sum_s |\alpha_s^x|}{|\mathcal{R}_{\mathsf{V}}|} = \sum_s \frac{|\beta_s^x|}{|\mathcal{R}_{\mathsf{V}}|} \cdot \frac{|\alpha_s^x|}{|\beta_s^x|} = \mathop{\mathbb{E}}_{s \leftarrow S_x} \left[\frac{|\alpha_s^x|}{|\beta_s^x|}\right] \tag{1}$$

So it is sufficient to construct a sound protocol where, for the given input x, M proves to A that the above expectation is large. Notice that for any s, the membership of a string r in the set α_s^x can be efficiently verified. This means that we can use the Goldwasser-Sipser set lower bound protocol [GS86] to prove lower bounds on the size of α_s^x. Now suppose the following three conditions were satisfied:

[1] In general interactive proofs, this assumption is without loss of generality. It is not clear that this is the case with instance-hiding interactive proofs without some worsening of parameters. In our actual proof, we do not need this assumption and the randomness of the prover is easy to deal with.

1. A has the ability to sample transcripts s from the distribution S_x
2. For s sampled as above, A can find out the value of $|\beta_s^x|$, and,
3. With high probability over s sampled from S_x, the value of $\frac{|\alpha_s^x|}{|\beta_s^x|}$ is close to its expectation

Then, we can construct the required protocol as follows:

- A samples an $s \leftarrow S_x$, and computes $|\beta_s^x|$
- M and A run the GS protocol where M proves that $|\alpha_s^x| \gg |\beta_s^x|/2$

If $x \in \mathcal{L}$, then with high probability over s, the condition involved in the GS protocol above is true, and A will accept with high probability. Similarly, if $x \notin \mathcal{L}$, this condition is false and A will reject with high probability.

If the protocol $\langle \mathsf{P}, \mathsf{V} \rangle$ were *perfectly* instance-hiding, then the distribution S_x is the same for any x of a given length, as is the size of the set β_s^x for any s – call this common distribution S and the common set size b_s. Then, we can handle the first two conditions above by providing to A a sample s from S and the corresponding value b_s as non-uniform advice (these are now independent of x). One catch here is that the completeness and soundness would then only hold for such randomly sampled advice. We show that an AM protocol with this property can be derandomized to a standard AM protocol with deterministic non-uniform advice (Corollary 3.5).

However, our protocol is only *statistically* instance-hiding, and so there may be no such common distribution. So we instead pick some canonical instance x_0, and use a sample s from the distribution S_{x_0} and the quantity $|\beta_s^{x_0}|$ as advice instead. Then we show that the instance-hiding property implies that the quantities $\mathbb{E}_{s \leftarrow S_x}\left[\frac{|\alpha_s^x|}{|\beta_s^x|}\right]$ and $\mathbb{E}_{s \leftarrow S_{x_0}}\left[\frac{|\alpha_s^x|}{|\beta_s^{x_0}|}\right]$ are close for any x. We use this to then show that using x_0 instead of x in the sampling of the advice will not affect the protocol by much.

All that is left is to ensure that the third condition above holds – that with high probablity $|\alpha_s^x| / |\beta_s^x|$ is close to its expectation. But it very well might not be. We deal with this by using instead the sum of many independent copies of this random variable: $\sum_{i \in [g]}(\left|\alpha_{s_i}^x\right| / \left|\beta_{s_i}^x\right|)$, where each s_i is sampled independently. By the Hoeffding bound, this sum is indeed close to its expectation with high probability. Now, instead of proving a lower bound on the size of a single set, M needs to prove that the weighted sum of the sizes of a number of sets is large. We design an AM protocol for this by extending the Goldwasser-Sipser protocol (Lemma 3.7).

Overall, the final constant-round protocol with randomized advice is as follows on input x:

- A receives as advice several samples $s_1, \ldots, s_g \leftarrow S_{x_0}$, and the quantities $\left|\beta_{s_1}^{x_0}\right|, \ldots, \left|\beta_{s_g}^{x_0}\right|$
- M and A run our weighted set-lower-bound protocol where M proves that: $\sum_i \frac{|\alpha_{s_i}^x|}{|\beta_{s_i}^{x_0}|} \gg \frac{g}{2}$

An AM protocol can be obtained from this constant-round IP following standard transformations [GS86].

Uniform Verifier from Simulatable-IHIP. Midway through the argument above, we observed that it is sufficient for the verifier to be able to obtain a number of samples s from S_x, together with the values $|\beta_s^x|$. Above, we resolved this by providing approximations of these as non-uniform advice. If we have an efficient simulator Sim (that only takes the size of the instance as input) for the view of the honest prover, however, we can compute these in the protocol itself, without needing such advice. The first part is clear – if we sample s from $\mathsf{Sim}(n)$, its distribution is guaranteed to be close to S_x for any x, by the instance-hiding property.

What remains is to arrange for the verifier to learn the (approximate) value of $|\beta_s^x|$ for such s. We do this by taking advantage of the fact that for any x and s, by the instance-hiding property, the probability

that $\mathsf{Sim}(n)$ outputs s is a reasonably good approximation of the probability that the protocol with verifier input x results in the transcript s. To be more specific, define the following quantity for any s:

$$\zeta_s = \{r_{\mathsf{Sim}} \mid s \text{ is the transcript of } \mathsf{Sim}(n; r_{\mathsf{Sim}})\}$$

Let the randomness space of the simulator be $\mathcal{R}_{\mathsf{Sim}}$. Then, we show that the simulator's output being negligibly close to the actual transcript implies the following for any x, with high probability over s sampled from either S_x or $\mathsf{Sim}(n)$:

$$\frac{1}{2} \cdot \frac{|\zeta_s|}{|\mathcal{R}_{\mathsf{Sim}}|} \leq \frac{|\beta_s^x|}{|\mathcal{R}_{\mathsf{V}}|} \leq 2 \cdot \frac{|\zeta_s|}{|\mathcal{R}_{\mathsf{Sim}}|}$$

Thus, if M can prove good bounds on the size of ζ_s to A, these would also imply good bounds on the size of β_s^x, and we can the proceed with the protocol as before. Notice that membership in ζ_s is again efficiently testable. So a lower-bound on $|\zeta_s|$ can again be proven using the GS protocol.

To prove an upper bound, we use the set upper-bound protocol of Fortnow [For87]. This protocol requires, in addition to membership in the set being testable, that the verifier privately obtain a uniformly random element from the set. This is easy for us to arrange – the verifier A simply samples a random $r \in \mathcal{R}_{\mathsf{Sim}}$, runs $\mathsf{Sim}(n; r)$ to obtain s, sends s to the prover M and keeps r private. Here, according to M, r is indeed a uniformly random element in ζ_s.

The entire protocol is now roughly as follows on input x:

- A samples $\{s_i \leftarrow \mathsf{Sim}_{\mathsf{P}}(n; r_i)\}_{i \in [g]}$, and sends the s_i's to M
- M and A run upper and lower bound protocols for A to obtain an approximation of each $|\zeta_{s_i}|$
- A assumes that $\left|\beta_{s_i}^x\right| = |\zeta_{s_i}| \cdot (|\mathcal{R}_{\mathsf{V}}| \,/\, |\mathcal{R}_{\mathsf{Sim}}|)$, using the values obtained above for the right-hand side
- M and A run our weighted set-lower-bound protocol where M proves that: $\sum_i \frac{\left|\alpha_{s_i}^x\right|}{\left|\beta_{s_i}^x\right|} \gg \frac{g}{2}$

The actual protocol is slightly different because the set upper-bound protocol's guarantees are a bit weaker than ideal, but in essence it is as above. We refer the reader to Section 4.1 for details.

1.2.2 One-Way Functions from IHIP for Hard Problems

Recall that Theorem 1.2 shows that if a *worst-case hard* (resp. *average-case hard*) problem has a Simulatable-IHIP (resp. IHIP), then there is explicit constructions of one-way functions (resp. non-explicit construction). The proofs assume, for the sake of contradiction, the non-existence of one-way function, and then use the efficient inverter algorithm for efficient functions guaranteed by the assumption to decide the problem.

Specifically, we assume the non-existence of distributional one-way function, which is implied by the non-existence of one-way functions [IL89]. If an efficient function f is not distributionally one-way, then there exists an efficient inverter A, which takes a random image of f as input, and samples preimages almost-uniformly. For the overview, we assume that the inverters work perfectly and sample uniformly random preimages[2], and also will focus on perfect-hiding protocols, and later describe how to make everything work with errors.

Non-Explicit OWFs from Average-Case Hard Constant-round IHIP. Consider any language $\mathcal{L}$ that has a q-round IHIP with prover P and verifier V, with a computationally unbounded simulator Sim for the honest prover's view. Further, there is an efficiently sampleable distribution X over which it is hard. We will work with $q = 2$ for this overview, which is sufficient to demonstrate the ideas behind the proof. Denote the algorithm of the verifier that computes the next message at any point in the protocol using random string

[2]This assumption clearly loses generality. If it were true, then one would obtain OWF's from worst-case hard problems in SZK and IHIP rather than needing average-case hardness.

r by $\mathsf{V}(x, u_1, \ldots, y_i; r)$. As before, for any (possibly partial) transcript s, denote by β_s^x the set of random strings r consisted with s, and by α_s^x the set of r that lead to V accepting with transcript s.

We observe that if $\langle \mathsf{P}, \mathsf{V} \rangle$ is a one-round (two-message $s = (u_1, y_1)$) protocol, then it is easy to show that $F_1(x, r_1) = (x, \mathsf{V}(x; r_1))$ must be an distributional one-way function[3]. Suppose not, there must exist an efficient inverter A_1, that on a random image (x, u_1) as input, samples r_1^* uniformly over $\beta_{u_1}^x$. Now consider a non-uniform adversary B^s that has as advice a random transcript $s = (u_1, y_1)$ sampled according to the simulation $\mathsf{Sim}(n)$, and works as follows on in put x:

- $\mathsf{B}^s(x)$ runs $\mathsf{A}_1(x, u_1)$ to obtain a random $r_1^* \in \beta_{u_1}^x$ and accepts iff $\mathsf{V}(x, u_1, y_1; r_1^*)$ accepts.

This last event above happens with probability $|\alpha_s^x| \,/\, |\beta_s^x|$. So in expectation over s, by (1), the probability that B^s accepts will be large if x is in $\mathcal{L}$ and small otherwise. The fact that the advice s is random in this algorithm is not a concern – by taking multiple s and repeating B^s several times, one can then show that there is a single set of transcripts s that works well as advice for all possible instances of size n. This gives us a non-uniform algorithm for $\mathcal{L}$, which is a contradiction, so F_1 must be a distributional OWF.

In general, if we can set up an efficiently computable function such that inverting it lets us sample a random element of β_s^x given x and s, we can repeat the above argument. When the protocol $\langle \mathsf{P}, \mathsf{V} \rangle$ is more than one round (even just two rounds), however, this approach is not straightforward. This is because the messages of the verifier starting from the second round are dependent on prover's messages and are in general not guaranteed to be efficiently sampleable in the way the first message is. For instance, a natural candidate for such a function might be defined as $F_2^{(u_1, y_1)}(x, r) = \mathsf{V}(x, u_1, y_1; r)$. However, we are only interested in inverses of this function where the r is consistent with the first message u_1, and this might not be efficiently sampleable.

Instead, we use an recursive argument that either finds a distributional one-way function or obtain a "useful" efficient sampler for define F_2. The argument proceeds as follows:

First, if F_1 is already a distributional OWF, we are done with our proof. If not, consider a (perfect) inverter A_1 for F_1. Let $\mathsf{C}_1(x, u_1)$ be the algorithm that computes $(x, r_1) \leftarrow \mathsf{A}(x, u_1)$, and just outputs r_1. We define the function $F_2^{(u_1, y_1)}$ for any (u_1, y_1) as follows:

$$F_2^{(u_1, y_1)}(x, r_2) = (x, \mathsf{V}(x, u_1, y_1; \mathsf{C}_1(x, u_1; r_2)))$$

That is, $F_2^{(u_1, y_1)}(x, r_2)$ first runs C_1 with randomness r_2 to sample an r_1 that is contained in $\beta_{u_1}^x$, and then runs V on the partial transcript (u_1, y_1) with that r_1 to produce the next verifier message u_2 in the protocol. If C_1 is perfect, this achieves what the earlier attempt at defining F_2 did not – any output of $\mathsf{C}_1(x, u_1)$ is a random element of $\beta_{u_1}^x$, so we are never in a situation where we have in hand a verifier random string r that is not consistent with the partial transcript so far.

Let us look at the property of random inverses of F_2 more closely. First, given a random r_2, $\mathsf{C}_1(x, u_1; r_2)$ is a random element of $\beta_{u_1}^x$, and so also of $\beta_{(u_1, y_1)}^x$. So a random inverse of $F_2^{(u_1, y_1)}$ on output (x, u_2) is a random (x, r_2) such that $\mathsf{C}_1(x, u_1; r_2)$ is also further contained in $\beta_{(u_1, y_1, u_2)}^x$. So given a random inverse (x, r_2) of $F_2^{(u_1, y_1)}$ on output (x, u_2), the output of $\mathsf{C}_1(x, u_1; r_2)$ is distributed uniformly over $\beta_{(u_1, y_1, u_2)}^x$, and thus also β_s^x for $s = (u_1, y_1, u_2, y_2)$ for any y_2. So given a random such $s = (u_1, y_1, u_2, y_2)$ from the simulator $\mathsf{Sim}(n)$ along with a perfect distributional inverter for $F_2^{(u_1, y_1)}$, we can efficiently sample from β_s^x, which is exactly what we needed! So unless $F_2^{(u_1, y_1)}$ is distributionally one-way for such an s, we can decide $\mathcal{L}$ on infinitely often n.

There are two remaining issues here – one is that we do not actually have a perfect distributional inverter, only a very good one; the other is the question of where the distributional inverters come from for the eventual algorithm we construct for $\mathcal{L}$. The solution to the latter is non-uniform advice. As before, we can argue that there is a set of transcripts s that work for all instances, and then if the F_2's defined with those transcripts are not distributionally one-way, their inverters can be provided as non-uniform advice. The former issue

[3]In case X is not uniform distribution, F should instead take the randomness used by the sampler of X rather than x as input.

can again be dealt with using standard techniques to carefully account for the inversion (and also hiding) errors and show that it remains small enough to not matter.

This process can then be inductively carried out for any constant number of protocol rounds. However, it cannot be directly applied to protocols with superconstant rounds. Proving such a claim would require iterating the argument unboundedly (e.g. polynomially in n), and we do not know how to prevent the description size of the constructed one-way functions from blowing up excessively. Finally note that the reason this does not extend to being able to use the *worst-case* hardness of $\mathcal{L}$ is that the functions F_1, etc., that we construct take x as an input. So inverting them on random outputs cannot give guarantees for every possible x.

Explicit OWFs from Worst-Case Hard Simulatable-IHIP. Consider any language $\mathcal{L}$ that has a q-round Simulatable-IHIP with prover P, verifier V and efficient honest-prover simulator Sim. Our approach is to use the possibility of inverting any efficiently computable function to efficiently implement the "simulation-based prover" [AH91] for this interactive proof for any instance. This proof is almost the same as the proof that average-case hardness of SZK implies OWF's [Ost91], though see below for further discussion.

The simulation-based prover $\mathsf{P}_{\mathsf{Sim}}$ is defined to behave as follows on interaction with verifier V: at any point in the protocol, if the current transcript is $(u_1^*, \ldots, y_{i-1}^*, u_i^*)$, it samples an $s = (u_1, \ldots, u_i, y_i)$ from $\mathsf{Sim}(n)$ conditioned on $u_j = u_j^*$ for $j \leq i$ and $y_j = y_j^*$ for $j < i$, and then responds to V with the message y_i. By the instance-hiding property, the view of the verifier generated by V interacting with $\mathsf{P}_{\mathsf{Sim}}$ is statistically close to that when interacting with the honest verifier P. Thus, due to the completeness and soundness of the protocol, if $\mathsf{P}_{\mathsf{Sim}}$ can be implemented efficiently, the language can also be decided efficiently.

For each $i \in [q]$, define the efficiently computable function $\mathsf{Sim}_i(n, r)$ that runs $\mathsf{Sim}(n; r)$ (with randomness r), and outputs the first $(2i-1)$ messages $(u_1, y_1, \ldots, u_i)$. If each Sim_i had a perfect distributional inverter, then given $(u_1^*, \ldots, u_i^*)$, the inverter can be used to sample a uniformly random r that when used by $\mathsf{Sim}(n; r)$ as randomness produces this partial transcript. Then computing $\mathsf{Sim}(n; r)$ can be used to sample the y_i that is exactly as required by the simulation-based prover. So if every efficiently computable function can be perfectly distributionally inverted, then the simulation-based prover can be implemented efficiently, giving us the contradiction we want. If the distributional inverter available is not perfect, there are some errors that come up throughout this process, but they can be handled using standard techniques.

This proof is very similar to the proof that average-case hardness of SZK implies OWF's [Ost91]. In the SZK case, the simulator also takes as input the instance x, and therefore being able to invert the simulator does not necessarily imply that the language can be decided for all instances x, which is why average-case hardness is needed. In this case, however, the simulator works for all instances x, and so inverting it gives an algorithm for all instances x, and worst-case hardness is sufficient.

1.2.3 Oracle Separation from SZK

To demonstrate an oracle separation between IHIP and SZK, we construct an oracle language for which these two protocols have different query complexities. This separation in query complexity can then be translated into an oracle separation using standard diagonalization techniques.

Our language is defined using the Discrete Log problem with a generic group oracle [Sho97]. For any $n \in \mathbb{N}$, and a prime number $N \approx 2^n$, given any bijection $\sigma : \mathbb{Z}_N \to [N]$, the generic group oracle $\mathcal{G}_\sigma$ encodes the group $\mathbb{Z}_N$ using otherwise meaningless labels from $[N]$. Given inputs $g, h \in [N]$, $\mathcal{G}_\sigma(g, h)$ is equal to $\sigma(\sigma^{-1}(g) + \sigma^{-1}(h))$. Consider in addition to this another oracle $\mathcal{I} : \mathbb{Z}_N \to \{0, 1\}$. We define our language (technically, promise problem) as:

$$\mathcal{L}^{\mathcal{I},\sigma} = \{(\sigma(1), \sigma(x)) \mid \mathcal{I}(x) = 1\}$$

This language has an IHIP for any σ and $\mathcal{I}$ when the parties are given access to $\mathcal{I}$ and $\mathcal{G}_\sigma$ as oracles. This is as follows: given input $(\sigma(1), \sigma(x))$, compute $y = \sigma(x + r)$ for a random $r \in \mathbb{Z}_N$, send y to the prover, who is supposed to return $r' \leftarrow \sigma^{-1}(y)$. Check that $\sigma(r') = y$, and if so output $\mathcal{I}(r' - r)$. The efficiency of the verifier here relies on the fact that $\sigma(r)$ for any r can be computed using $\mathrm{poly}(n)$ calls to $\mathcal{G}_\sigma$ using repeated

doubling. Completeness and soundness follow from the fact that (r, r') is an NP witness for the instance, and strong instance-hiding follows from the fact that the prover only sees a uniformly random element of $[N]$.

On the other hand, it is known from generic lower-bounds for the Discrete Log problem [Sho97, CGK18] that no algorithm can compute x given $(\sigma(1), \sigma(x))$ and oracle access to $\mathcal{G}_\sigma$ for a random σ with substantially fewer than $\sqrt{N}$ queries. This implies that any candidate efficient SZK simulator would, with very high probability, not query oracle $\mathcal{I}$ on x given input $(\sigma(1), \sigma(x))$ and oracle access to $\mathcal{G}_\sigma$. The simulation property then implies that if $\mathcal{I}(x) = 1$, then the verifier for the corresponding protocol, with high probability, would not query $\mathcal{I}$ on x either. If this happens, then the outcome of the protocol would have been the same irrespective of whether $\mathcal{I}(x)$ was 0 or 1. This shows that at least one of zero-knowledge, completeness, or soundness breaks at such an input x. This shows the required query complexity lower bound.

1.3 Related Work

The concept of instance-hiding proof systems was first introduced, albeit in the multi-prover setting, by Beaver et al. [BFS90]. Their definition was based on that of instance-hiding *schemes* as defined by Abadi et al. [AFK89], which may be seen as honest-prover instance-hiding proofs without the soundness property. The latter also showed that any language that has a perfect instance-hiding scheme (which is implied by a proof) is contained in $\mathsf{NP/poly} \cap \mathsf{coNP/poly}$. The former showed that a language has a multi-prover instance-hiding proof iff it is contained in $\mathsf{NEXP} \cap \mathsf{coNEXP}$, and further that such a proof could be made zero-knowledge.

Feigenbaum and Ostrovsky [FO91] and Beaver et al. [BFOS93] showed further connections between (single-prover) instance-hiding schemes and proofs assuming the existence of one-way permutations. To be more accurate, most of these papers consider instance-hiding proofs for certifying the evaluations of functions and discuss the feasibility of such proofs based on the complexity of these functions. Multi-prover instance-hiding schemes were also studied by Beaver and Feigenbaum [BF90], who showed that they exist for all functions.

Randomized Encodings. Randomized Encodings are closely related to instance-hiding proofs. The properties we show for IHIP – membership in $\mathsf{AM} \cap \mathsf{coAM}$ and the implication of OWFs from *worst-case* hardness [AR16] – are also shared by the class of languages that have Statistical Randomized Encodings (SRE) [IK00, AIK04, AIKPC15]. An SRE for a language $\mathcal{L}$ is a randomized function whose output on an input x reveals whether $x \in \mathcal{L}$ and nothing else about x, in a statistical sense. Randomized encodings with very low complexity have been used extensively in constructing MPC protocols [Yao86, Kil88, IK00, FKN03, AIK04, ...]. Agrawal et al. [AIKPC15] showed examples of languages that have SREs that, under reasonable computational assumptions, are not efficiently computable.

It is known that languages that have an SRE also have SZK proofs [App14a] (and the above oracle separation also carries over). Techniques from the literature also show that any language that has an SRE also has an IHIP (see e.g. [AIK10], and references in Section 1.1 therein). For completeness, we include a self-contained proof of this statement in Appendix C.1. These techniques can further be extended to show that an interactive version of Randomized Encodings (as defined by Applebaum et al. [AIK10]) is equivalent to IHIP.

1.4 Discussion and Open Problems

There are number of fundamental questions about the properties and power of instance-hiding proofs that are yet to be answered. We list a few of these below.

1. Are there natural complete problems for the class of languages that have instance-hiding proofs?
2. What is the relationship between this class and SZK? Both of them are contained in $\mathsf{NP/poly} \cap \mathsf{coNP/poly}$, but is one contained in the other? In this work, we provide an oracle separation $\mathsf{IHIP}^O \not\subseteq \mathsf{SZK}^O$; can we show one in the other direction?

3. Is Simulatable-IHIP closed under complement?

4. Are there other cryptographic consequences of the existence of hard problems in this class, beyond one-way functions?

5. Can worst-case hard IHIP imply one-way function? This question is also open for SZK.

6. Can the instance-hiding error be amplified? Note that this question is also open for SRE.

7. Is there a separation between perfect and imperfect instance-hiding proofs?

8. What is the power of *computational* instance-hiding proofs? What assumptions are needed for these to be constructed for all of NP?

9. Similarly, what is the power of instance-hiding *argument systems*, which have efficient provers (given, say, an NP witness) and only computational soundness? How do we even define these, given that the witness might already reveal information about the witness?

There is also a looming non-technical question here that it would be useful to know the answer to. Instance-hiding proofs and zero-knowledge proofs were defined at around the same time. The initial results regarding these – multi-prover constructions for large classes of languages, limitations of perfect single-prover constructions, etc. – seem to have been similar. In strong contrast to zero-knowledge proofs, however, research on instance-hiding proofs (at least explicitly) has been very sparse after a brief period following their definition. Why is this the case? Could it be because we succeeded in constructing computational ZK proofs for all of NP shortly thereafter, whereas similar results for instance-hiding proofs were lacking? Given the more advanced cryptographic toolkit available to us today, can we construct computational instance-hiding proofs for large classes of languages? Would they be as useful as ZK proofs?

2 Instance-Hiding Interactive Proofs

In this section, we define Instance-Hiding Interactive Proofs, and prove closure and amplification properties for such proof systems. We demonstrate some examples of such proofs for non-trivial problems in Appendix A. We start by setting up notation that we will use for the rest of the paper.

Notation. Throughout this paper, for an interactive proof involving a prover P and a verifier V, we use $\textsc{View}_{\mathsf{P}}(\mathsf{P},\mathsf{V})$ to represent the prover's view during an execution (this includes the prover's random coins, if any, and the set of messages in the execution). Similarly, $\textsc{View}_{\mathsf{P}}(\mathsf{P},\mathsf{V})$ and $\textsc{View}_{pub}(\mathsf{P},\mathsf{V})$ denote the verifier's view and public view (that is, the transcript of messages) during execution, respectively. The private inputs of provers and verifier, which can include elements such as private coins and instances, are associated with the respective parties. For instance, $\textsc{View}_{\mathsf{P}}\big(\mathsf{P}(|x|;r_P),\mathsf{V}(x;r_V)\big)$ denotes the prover's view when the verifier, receiving private instance x, uses randomness r_V, and the prover, given only length of instance $|x|$, uses randomness r_P. For conciseness, some private and public inputs may be omitted when they are evident from context. Moreover, we naturally associate the verifier accepting with an output of 1, and rejection with 0.

Our various results involve both uniform and non-uniform algorithms. Unless otherwise specified, algorithms in our discussion are uniform. Further, unless otherwise specified, the transformations (between protocols or algorithms) in statements of our results preserve uniformity. For any probabilistic algorithm A that samples its randomness uniformly, we denote its randomness space by $\mathcal{R}_{\mathsf{A}}$.

We consider interactive proofs for promise problems, rather than languages. A promise problem is a pair $\Pi = (\textsc{Yes},\textsc{No})$ of disjoint sets. We say that x satisfies the promise for Π, or is an instance of Π, if $x \in \textsc{Yes} \cup \textsc{No}$. We denote by $\textsc{Yes}(\Pi) = \textsc{Yes}$ and refer to $x \in \textsc{Yes}(\Pi)$ as "Yes" instances of Π and

$x \in \text{NO}(\Pi) = \text{NO}$ as "No" instances of Π. We employ Π_n (similarly YES_n, NO_n) as shorthand for the set of instances of Π contained in $\{0,1\}^n$. For any instance x of Π, we overload the notation $\Pi(x)$ as:

$$\Pi(x) = \begin{cases} 1, \text{ if } x \in \text{YES}(\Pi). \\ 0, \text{ if } x \in \text{NO}(\Pi). \end{cases}$$

For any set S, U_S denotes a uniformly random sample from S, and U_ℓ denotes a uniformly random sample from the set of ℓ-bit strings. We use $\text{negl}(n)$ to denote a negligible function in n and ϕ to denote empty string.

Definition 2.1 (Statistical Distance)**.** The *statistical distance* between two distributions X and Y over a finite set $\mathcal{X}$ is defined as:

$$\Delta(X,Y) = \max_{S \subseteq \mathcal{X}} \left| \Pr[X \in S] - \Pr[Y \in S] \right| = \frac{1}{2} \sum_{u \in \mathcal{X}} \left| \Pr[X = u] - \Pr[Y = u] \right|,$$

Definition 2.2 (Instance-Hiding Interactive Proof (IHIP) [BFS90])**.** Consider a promise problem $\Pi = (\text{YES}, \text{NO})$, and functions $\delta, \epsilon : \mathbb{N} \to [0,1]$. A (δ,ϵ)-*Instance-Hiding Interactive Proof* (IHIP) for Π is a protocol $\langle \mathsf{P}, \mathsf{V} \rangle$ in which a probabilistic polynomial-time verifier V interacts with a computationally unbounded prover P. For some $n \in \mathbb{N}$, the verifier gets a *private* input $x \in \text{YES}_n \cup \text{NO}_n$, while the prover only gets the input length n. At the end of the interaction, V outputs either 1 (Accept) or 0 (Reject). The protocol is required to satisfy the following properties for all large enough $n \in \mathbb{N}$:

- **Completeness:** For any input $x \in \text{YES}_n$:

$$\Pr[\langle \mathsf{P}(n), \mathsf{V}(x) \rangle = 1] \geq 1 - \delta(n).$$

- **Soundness:** For any input $x \in \text{NO}_n$, and any prover P^*:

$$\Pr[\langle \mathsf{P}^*(n), \mathsf{V}(x) \rangle = 1] \leq \delta(n).$$

- **Hiding:** For any prover P^*, there exists a computationally unbounded randomized algorithm $\mathsf{Sim}_{\mathsf{P}^*}$, called a simulator, such that for any input $x \in \{0,1\}^n$,

$$\Delta\big(\mathsf{Sim}_{\mathsf{P}^*}(n), \text{VIEW}_{\mathsf{P}^*}(\mathsf{P}^*(n), \mathsf{V}(x))\big) \leq \epsilon(n).$$

If the simulator corresponding to the *honest* prover runs in polynomial time in n, we say the protocol is *Simulatable-Instance-Hiding* (Simulatable-IHIP). The protocol is *perfectly-hiding* IHIP if $\epsilon(n) = 0$ for all n. If a simulator is only guaranteed to exist only for the honest prover P, the protocol is *honest-prover* IHIP[4].

Definition 2.3 (Class IHIP, $\mathsf{IHIP/poly}$)**.** The class IHIP consists of all promise problems that have a (δ,ϵ)-IHIP with uniform verifier protocol for some negligible $\delta(n)$ and $\epsilon(n)$. For concrete functions (δ,ϵ), we denote by (δ,ϵ)-IHIP the class of problems possessing (δ,ϵ)-IHIP. Similarly, $\mathsf{IHIP/poly}$ denotes the class of promise problem that have a (δ,ϵ)-IHIP with non-uniform verifier protocol for some negligible $\delta(n)$ and $\epsilon(n)$

Remark 2.4. Prior work in this area, such as Beaver et al. [BFS90], defined instance-hiding proof systems for function delegation rather than promise problem decision. For a function f, at the end of the protocol, completeness required that the verifier learn $f(x)$ when interacting with the honest prover; and soundness required that no prover could convince the verifier of an incorrect value of $f(x)$. Definition 2.2 is weaker than just the restriction of this to Boolean functions, in that we only require completeness guarantees to hold for YES instances ($\Pi(x) = 1$), and soundness guarantees for NO instances ($\Pi(x) = 0$). This relaxed definition is still meaningful, and lets us compare IHIPs directly to IPs, ZK proofs, etc., that are also similarly defined. The relaxation also makes it harder to prove our containment results and closure properties. Some of our results also extend to the definition involving function delegation – see Section 7 for this definition and further details.

[4]In fact, there exists a transformation that enables the construction of an IHIP against all provers from an honest-prover IHIP, as shown in Theorem 7.5.

Remark 2.5. Earlier definitions of instance-hiding proof systems only considered *perfect* instance-hiding. This made showing containment of problems with such proofs in NP and coNP considerably simpler than our proofs showing containment of problems with imperfect instance-hiding proofs in AM and coAM.

Remark 2.6. As we show in Appendix C, instance-hiding IPs are closely related to the notion of Randomized Encodings (RE) of promise problems [AIK04] (see also [AIK05]). In fact, using the techniques in that section, instance-hiding IPs (for non-negligible values of the hiding error) can be shown to be equivalent to an interactive version of RE as defined by Applebaum et al. [AIK10].

2.1 Closure Properties and Amplification

Since their inception, the composition properties of zero-knowledge proofs have been extensively studied. It is established that Statistical Zero-Knowledge (SZK) is preserved under sequential repetition, and that the existence of an SZK proof is preserved under composition with arbitrary polynomial-sized formulas [GK96, Oka00, DSDCPY08, CD96, SV97]. However, similar properties for instance-hiding have not received as much attention yet. In this subsection, we present positive results regarding these properties of IHIP. We show that the existence of such proofs is preserved under composition with any efficiently computable function, and not just polynomial-sized formulas.

Again, for any promise problem $\Pi = (\text{YES}, \text{NO})$, we also denote by $\Pi : \{0,1\}^* \to \{0,1,\bot\}$ its characteristic function, which outputs 1 on any input $x \in \text{YES}$, 0 on any $x \in \text{NO}$, and $\bot$ on all other inputs. Similarly, given any function $f : \{0,1\}^* \to \{0,1,\bot\}$, we define the corresponding promise problem Π_f whose characteristic function it is. Consider any function $f : \{0,1,\bot\}^* \to \{0,1,\bot\}$ satisfying the property that its output is $\bot$ whenever any of its inputs is $\bot$. For any function $k : \mathbb{N} \to \mathbb{N}$, we define the composed promise problem $f \circ \Pi^{\otimes k}$ as follows:

$$\text{YES}_n(f \circ \Pi^{\otimes k}) = \left\{(x_1, \ldots, x_{k(n)}) \mid \forall i : |x_i| = n \wedge f(\Pi(x_1), \ldots, \Pi(x_{k(n)})) = 1\right\}.$$
$$\text{NO}_n(f \circ \Pi^{\otimes k}) = \left\{(x_1, \ldots, x_{k(n)}) \mid \forall i : |x_i| = n \wedge f(\Pi(x_1), \ldots, \Pi(x_{k(n)})) = 0\right\}.$$

Theorem 2.7 (Closure under Composition with Efficient Functions)**.** *Consider any promise problems Π that has an IHIP protocol, and any efficiently computable function $f : \{0,1,\bot\}^* \to \{0,1,\bot\}$ whose output is $\bot$ whenever any of its inputs is $\bot$. For any polynomial $k : \mathbb{N} \to \mathbb{N}$, the composed promise problem $f \circ \Pi^{\otimes k}$ also has an IHIP protocol.*

The full proof of this theorem needs tools that we develop in Section 7, and so we defer the proof to Section 7.2. An important special case of this theorem, which we use in its proof, is the closure of IHIP under complementation. This is stated in the following lemma, which is also proven in Section 7.2. The transformation we use to prove this lemma does not preserve the efficiency of the simulator, and so it does not carry over to simulatable IHIP. This happens to be the missing piece in extending Theorem 2.7 to simulatable IHIP as well.

Lemma 2.8 (Closure under Complementation)**.** *Suppose, for some negligible functions δ, ϵ, that a problem Π has a (δ, ϵ)-IHIP (possibly with a non-uniform verifier). Then the complement of Π has a (δ', ϵ')-IHIP (resp. with a non-uniform verifier if starting with a non-uniform verifier), where δ', ϵ' are also negligible.*

Corollary 2.9. $\mathsf{IHIP/poly} = \mathsf{coIHIP/poly}$ *and* $\mathsf{IHIP} = \mathsf{coIHIP}$.

Another component of the proof of Theorem 2.7 is the following lemma regarding the behavior of instance-hiding proofs under repetition in parallel that is significant on its own. In contrast to zero-knowledge proofs, instance-hiding proofs show robustness under parallel repetition.

Lemma 2.10 (Preservation of Instance-Hiding Under Parallel Repetition)**.** *For any functions $k : \mathbb{N} \to \mathbb{N}$ and $\epsilon : \mathbb{N} \to [0,1]$, consider the protocol $\left\langle \vec{\mathsf{P}}, \vec{\mathsf{V}} \right\rangle$ where $\vec{\mathsf{V}}$ takes as input $k(n)$ instances/inputs $x_1, \ldots, x_{k(n)}$, each of size n, and executes $\langle \mathsf{P}_1(n), \mathsf{V}_1(x_1) \rangle, \ldots, \langle \mathsf{P}_k(n), \mathsf{V}_k(x_k) \rangle$ independently in parallel, where each $\langle \mathsf{P}_i, \mathsf{V}_i \rangle$ is ϵ-instance-hiding. Then $\left\langle \vec{\mathsf{P}}, \vec{\mathsf{V}} \right\rangle$ is $(k \cdot \epsilon)$-instance-hiding.*

We prove this lemma later in this section. This gives us a round-efficient way to strongly amplify the completeness and soundness in an instance-hiding proof at a small cost in the instance-hiding error.

Lemma 2.11 (Amplifying IHIP by Parallel Repetition)**.** *For functions δ, ϵ such that $\delta(n) < 1/2 - \Omega(1)$ for all n, consider a (δ, ϵ)-IHIP $\langle \mathsf{P}, \mathsf{V} \rangle$ for a promise problem Π. For any polynomial k, let $\langle \mathsf{P}^{\otimes k}, \mathsf{V}^{\otimes k} \rangle$ be the protocol where $\mathsf{V}^{\otimes k}$, on input x, runs $k(n)$ executions of $\langle \mathsf{P}(n), \mathsf{V}(x) \rangle$ independently with $\mathsf{P}^{\otimes k}$, and outputs the majority of the results. Then, $\langle \mathsf{P}^{\otimes k}, \mathsf{V}^{\otimes k} \rangle$ is a $(2^{-\Omega(k)}, k \cdot \epsilon)$-IHIP for Π.*

Proof of Lemma 2.11. The $(k \cdot \epsilon)$-hiding of the repeated protocol follows directly from Lemma 2.10. Amplification of completeness and soundness follows from Lemma B.1, which asserts the amplification of general interactive proofs by parallel repetition. For completeness, we include a self-contained proof of this lemma in Appendix B.

□

Proof of Lemma 2.10. Assume for contradiction that there exists a prover $\vec{\mathsf{P}}^*$ and infinitely many n for which there exists $\vec{x}, \vec{y} \in (\{0,1\}^n)^{k(n)}$ such that:

$$\Delta\Big(\textsc{View}_{\vec{\mathsf{P}}^*}\big(\vec{\mathsf{P}}^*, \vec{\mathsf{V}}(\vec{x})\big), \textsc{View}_{\vec{\mathsf{P}}^*}\big(\vec{\mathsf{P}}^*, \vec{\mathsf{V}}(\vec{y})\big)\Big) > k(n) \cdot \epsilon(n).$$

We derive a contradiction by constructing a sequence of provers $\mathsf{P}_0^*, \mathsf{P}_1^* \ldots, \mathsf{P}_k^*$, with P_i^* interact with V_i, and proceed to argue that there exists P_j^*, x, y such that:

$$\Delta\Big(\textsc{View}_{\mathsf{P}_j^*}\big(\mathsf{P}_j^*, \mathsf{V}_j(x)\big), \textsc{View}_{\mathsf{P}_j^*}\big(\mathsf{P}_j^*, \mathsf{V}_j(y)\big)\Big) > \epsilon(n).$$

For each $i \in [k]$, we define machine P_i^* that simulates the $\vec{\mathsf{P}}^*$ and $\mathsf{V}_1(x_1), \ldots, \mathsf{V}_{i-1}(x_{i-1}), \mathsf{V}_{i+1}(y_{i+1}), \ldots, \mathsf{V}_k(y_k)$ and interacts with V_i on input x in parallel. The prover P_i^* writes the simulated $k-1$ interactions on the communication tape in position so that the distribution on communication tape after the interaction is the same to that during the interaction between $\vec{\mathsf{P}}^*$ and $\vec{\mathsf{V}}$ on input $(x_1, \ldots, x_{i-1}, x, y_{i+1} \ldots, y_k)$. We define $D(\vec{z})$ as the shorthand of the distribution of prover's view $\textsc{View}_{\vec{P}^*}\big(\vec{\mathsf{P}}^*, \vec{\mathsf{V}}(\vec{z})\big)$ on instances $\vec{z}$ and construct $k+1$ hybrid distributions:

- $H_0 = D(\vec{y}) = \textsc{View}_{\vec{\mathsf{P}}^*}\big(\vec{\mathsf{P}}^*, \vec{\mathsf{V}}(\vec{y})\big)$.
- $H_i = D(x_1, \ldots, x_i, y_{i+1}, \ldots, y_k)$, for $i \in [k-1]$.
- $H_k = D(\vec{x}) = \textsc{View}_{\vec{\mathsf{P}}^*}\big(\vec{\mathsf{P}}^*, \vec{\mathsf{V}}(\vec{x})\big)$.

We notice that, for $i \in [k]$, the distribution of P_i^*'s view on input x_i $\Big($denoted as $\textsc{View}_{\mathsf{P}_i^*}\big(\langle \mathsf{P}_i^*, \mathsf{V}(x_i)\rangle\big)\Big)$ is identical to H_i, and identical to $\textsc{View}_{\mathsf{P}_{i+1}^*}\big(\langle \mathsf{P}_{i+1}^*, \mathsf{V}(y_{i+1})\rangle\big)$. By triangle inequality, we have that:

$$\begin{aligned}
\sum_{i \in [k]} \Delta\Big(\textsc{View}_{\mathsf{P}_i^*}\big(\langle \mathsf{P}_i^*, \mathsf{V}(y_i)\rangle\big), \textsc{View}_{\mathsf{P}_i^*}\big(\langle \mathsf{P}_i^*, \mathsf{V}(x_i)\rangle\big)\Big) &= \sum_{i \in [k]} \Delta\big(H_{i-1}, H_i\big) \\
&\geq \Delta\big(H_0, H_k\big), \\
&= \Delta\Big(\textsc{View}_{P_k^*}\big(\vec{\mathsf{P}}^*, \vec{\mathsf{V}}(\vec{x})\big), \textsc{View}_{P_k^*}\big(\vec{\mathsf{P}}^*, \vec{\mathsf{V}}(\vec{y})\big)\Big) \\
&> k(n) \cdot \epsilon(n).
\end{aligned}$$

By pigeonhole principle, there must exists some $j \in [k]$ such that

$$\Delta\Big(\textsc{View}_{\mathsf{P}_j^*}\big(\langle \mathsf{P}_j^*, \mathsf{V}_j(x_j)\rangle\big), \textsc{View}_{\mathsf{P}_j^*}\big(\langle \mathsf{P}_j^*, \mathsf{V}_j(y_j)\rangle\big)\Big) > \epsilon(n).$$

This contradicts the assumption and the lemma follows. □

3 Tools

In this section, we define terms and prove lemmas that will be useful in other parts of the paper. This section may be read on its own, and the results here might find use in other contexts as well.

3.1 AM Proofs and Advice Oracles

We leave out standard definitions of interactive proofs and their properties. An AM proof is a constant-round public-coin interactive proof. In our study, we will concern certain scenarios where the verifier in such proofs is non-uniform. In such cases, the verifier receives polynomial-sized advice that is a function of the input length. We define the following related complexity classes and state some relevant facts about them.

Definition 3.1 ($\mathsf{AM/poly}$)**.** The class $\mathsf{AM/poly}$ consists of all promise problems that have a constant-round public-coin interactive proof with a non-uniform polynomial-time verifier and unbounded prover. The class $\mathsf{coAM/poly}$ consists of promise problems whose complements have such proofs.

Lemma 3.2 ([FF91, Lemma 4])**.** *If* $\mathsf{coNP} \subseteq \mathsf{AM/poly}$*, then the polynomial hierarchy collapses to the third level.*

In the constructions of AM proof systems, it is advantageous to consider the verifier's non-uniform advice as a sample from a advice distribution outputed by an advice oracle $\mathsf{O}(n)$ instead of merely a string determined by the input length. This shift allows for a nuanced analysis on the completeness and soundness of the protocol, considering the inherent randomness of this advice oracle.

We define this type of proof system below and demonstrate that it is equivalent to those with standard non-uniform verifiers. We denote by $\langle \mathsf{P}, \mathsf{V} \rangle^a$ the execution of the protocol with V given a as advice and apply $\langle \mathsf{P}, \mathsf{V} \rangle^{O(n)}$ as shorthand of that samples $a \leftarrow O(n)$ and apply a as the non-uniform advice $\langle \mathsf{P}, \mathsf{V} \rangle^a$.

Definition 3.3. An *interactive proof with randomized advice oracle* for a promise problem Π is described by a pair of Turing machines (P, V) and an *advice oracle* O. An execution of this protocol on input x is denoted by $\langle \mathsf{P}, \mathsf{V} \rangle^{\mathsf{O}}(x)$, and consists of sampling a string $a \leftarrow \mathsf{O}(n)$, and running the protocol on input x with a as the advice string for V. It is required to have the following properties:

- **Completeness:** For any input $x \in \textsc{Yes}(\Pi_n)$:

$$\Pr_{\mathsf{O},P,V}\left[\langle \mathsf{P}, \mathsf{V} \rangle^{\mathsf{O}}(x) = 1\right] \geq \frac{2}{3}.$$

- **Soundness:** For any input $x \in \textsc{No}(\Pi_n)$, and any prover P^*:

$$\Pr_{\mathsf{O},P^*,V}\left[\langle \mathsf{P}^*, \mathsf{V} \rangle^{\mathsf{O}}(x) = 1\right] \leq \frac{1}{3}.$$

Lemma 3.4. *Suppose a promise problem* Π *has a* $q(n)$*-rounds interactive proof* $\langle \mathsf{P}, \mathsf{V} \rangle$ *with a randomized advice oracle, then* Π *has a* $q(n)$*-rounds interactive proof* $\langle \mathsf{P}', \mathsf{V}' \rangle$ *with a fixed non-uniform advice string of polynomial size to input size. Furthermore, if the* $\langle \mathsf{P}, \mathsf{V} \rangle$ *is a public-coin, then* $\langle \mathsf{P}', \mathsf{V}' \rangle$ *remains to be public-coin.*

Proof of Corollary 3.5. Let $\langle \mathsf{P}, \mathsf{V} \rangle$ be a interactive proof system with advice oracle O. We define $\langle \mathsf{P}', \mathsf{V}' \rangle$ as the interactive proof protocol that runs $\langle \mathsf{P}, \mathsf{V} \rangle$ for some polynomial $t(n)$ repetitions in parallel[5] with the advice strings drawn independently from the advice oracle O, and takes the majority result as the output. We have the following using standard arguments about the parallel composition of interactive proofs:

[5]We remark that the lemma can be extended to interactive proof with special properties such as hiding, provided that these properties are preserved under parallel repetition. It's important to note, however, that certain properties like zero-knowledge, might not be maintained through this process.

- If $x \in \textsc{Yes}(\Pi_n)$:
$$\Pr_{\mathsf{P}',\mathsf{V}',a\leftarrow \mathsf{O}^{\otimes t}}[\langle \mathsf{P}',\mathsf{V}'\rangle^a(x)=1] \geq 1-e^{-t(n)}.$$
- If $x \in \textsc{No}(\Pi_n)$, for any P^* :
$$\Pr_{\mathsf{P}^*,\mathsf{V}'\ a\leftarrow \mathsf{O}^{\otimes t}}[\langle \mathsf{P}^*,\mathsf{V}'\rangle^a(x)=1] < e^{-t(n)}.$$

We call advice $\bar{a}=(a_1,\ldots,a_t)$ "bad" if there exists an instance $x \in \textsc{Yes}(\Pi_n)\cup\textsc{No}(\Pi_n)$ such that $\langle \mathsf{M}',\mathsf{A}'\rangle^a$ incurs a soundness and completeness error greater than 2^{-n} on x. Set $t(n)$ such that $e^{-t(n)} \leq 2^{-3n}$, ensuring that for any instance x, there are at most a 2^{-2n} fraction of possible advice $\bar{a}$, sampled by $\mathsf{O}^{\otimes t}$, will lead to a scenario where $\langle \mathsf{P}',\mathsf{V}'\rangle^a$ has a soundness and completeness error greater than 2^{-n}. Apply an union bound over all n-bit strings we have
$$\Pr_{\bar{a}\leftarrow \mathsf{O}^{\otimes s}}[\bar{a} \text{ is "bad"}] \leq 2^n \cdot 2^{-2n} \leq 2^{-n}.$$
Consequently, there must exist at least one "good" advice $\bar{a}$, generated by $\mathsf{O}^{\otimes t}$, with which $\langle \mathsf{P}',\mathsf{V}'\rangle^a$ exhibits negligible completeness/soundness error for all instances, leading to the conclusion of the theorem. □

[FF91] show that $\mathsf{AM/poly}$ is, in fact, equivalent to $\mathsf{NP/poly}$. Combining this result with Lemma 3.4, we obtain:

Corollary 3.5. *Suppose a promise problem Π has a constant-round public-coin interactive proof with a randomized advice oracle. Then $\Pi \in \mathsf{NP/poly}$.*

3.2 Weighted Set Lower Bound Protocol

In this section, we construct a protocol that, given some sets $S_1,\ldots,S_g$, and corresponding positive numbers $c_1,\ldots,c_g$, can prove that the sum $\sum_i c_i\cdot|S_i|$ is larger than some specified threshold. This is an extension of a similar protocol of Goldwasser and Siper [GS86] for a single set.

The input to [GS86] protocol includes a set $S \subseteq \{0,1\}^m$ defined using a "membership oracle" $\mathsf{O}_S : \{0,1\}^m \to \{0,1\}$. That is $\mathsf{O}_S(x)=1$ if $x \in S$, and $\mathsf{O}_S(x)=0$ otherwise. Both the verifier and the prover in the protocol have access to this oracle.

Lemma 3.6 (Goldwasser-Sipser Lower Bound Lemma [GS86])**.** *For any $m,t \in \mathbb{N}$, there is a 2-message public-coin interactive protocol that, given access to the membership oracle for any $S \subseteq \{0,1\}^m$ and an input $K \in \mathbb{N}$, has the following properties:*

- *If $|S| \geq K$, the verifier accepts with probability at least $1-2^{-t}$ when interacting with the honest prover.*
- *If $|S| \leq K/2$, the verifier accepts with probability at most 2^{-t} when interacting with any prover.*

Further, the number of verifier's oracle calls is $O(t)$, its running time is $(t\cdot\mathrm{poly}(m))$, and the protocol has $(t\cdot\mathrm{poly}(m))$ bits of communication.

Lemma 3.7 (Weighted Set-Lower-Bound Lemma)**.** *For any $m,g,t \in \mathbb{N}$, there is a 3-message public-coin interactive protocol such that, given access to the membership oracles for any $S_1,\ldots,S_g \subseteq \{0,1\}^m$, and inputs $c_1,\ldots,c_g \in (0,1]$, and $K>0$:*

- *If $\sum_{i\in[g]} c_i\cdot|S_i| \geq K$, the verifier accepts with probability at least $1-g\cdot 2^{-t}$ when interacting with the honest prover.*
- *If $\sum_{i\in[g]} c_i\cdot|S_i| \leq K/4$, the verifier accepts with probability at most 2^{-t} when interacting with any prover.*

Further, the verifier makes $O(t)$ calls to each oracle, has running time $O(g \cdot t \cdot \text{poly}(m))$, and the protocol has $(g \cdot t \cdot \text{poly}(m))$ bits of communication.[6]

Proof of Lemma 3.7. Let LBP be the set lowerbound protocol from Lemma 3.6. We construct a public-coin interactive protocol that satisfies Lemma 3.7 as shown in Figure 1.

Weighted Set-Lower-Bound Protocol (WSLBP): $\langle \mathsf{M}, \mathsf{A} \rangle^{\mathbf{O}_{\vec{S}}}(\vec{c}, K)$

Parameters: m (set size), g (number of sets), t (threshold parameter).

Inputs:

- Coefficients Vector: $\vec{c} = (c_1, \ldots, c_g)$ with each $c_i \in (0, 1]$.
- Threshold: A positive number $K \in \mathbb{R}$.
- Sets: Oracle access to $\mathbf{O}_{\vec{S}} = (\mathbf{O}_{S_1}, \ldots, \mathbf{O}_{S_g})$, where $\mathbf{O}_{S_1}, \ldots, \mathbf{O}_{S_g}$ are membership oracles for $S_1, \ldots, S_g \subseteq \{0,1\}^m$ respectively.

Outputs: A binary decision $b \in \{0, 1\}$, indicating the outcome of the lower-bound assessment.

Ingredients:

- $\mathsf{LBP}^{\mathsf{O}_S}(K)$ is [GS86] protocol with parameters m and t given oracle access O_S for some set S, as promised by Lemma 3.6.

Protocol:

1. The prover M:
   - **If** $\sum_{i \in [g]} c_i \cdot |S_i| \geq K$:
     - Send to the verifier A the values $y_1, \ldots, y_g$, where $y_i = |S_i|$.
   - **Else**: Abort.
2. **For** $i \leftarrow 1$ to g:
   - Both A and M run the protocol LBP with membership oracle O_{S_i} and input y_i, obtaining the output b_i.
   - if $b_i = 0$, the verifier A outputs 0 and aborts .
3. The verifier A computes $z = \sum_{i \in [g]} c_i \cdot y_i$.
4. The verifier A outputs 1 if $z \geq K$; otherwise, outputs 0.

Figure 1: Weighted Sets Lower-Bound Protocol (WSLBP)

Completeness: The completeness of the protocol follows from the completeness of LBP. For a scenario where $\sum_{i \in [g]} c_i \cdot |S_i| \geq K$, the prover M sets assigns $y_i = |S_i|$. We define an event E_i to represent the occurence of "$b_i = 1$ in ith iteration of WSLBP". Leveraging Lemma 3.6, we have:

$$\begin{aligned} Pr[A \text{ accepts}] &\geq Pr[\bigwedge_{i \in [g]} E_i] \\ &\geq \prod_{i \in [g]} (1 - 2^{-t}) \\ &= (1 - 2^{-t})^g, \\ &\geq (1 - g \cdot 2^{-t}). \end{aligned}$$

[6] Here we assume that K and the coefficients $c_1, \ldots, c_g$ are encoded using $\text{poly}(m)$ bits.

Soundness: Let y_i^* represent the values received by A in the first message. For any set of y_i^*'s sent by a prover M^* in the first message. We observe that for each $i \in [g]$:

- If $|S_i| \geq y_i^*$:
$$\Pr[b_i = 1] \geq 1 - 2^{-t}.$$
- If $|S_i| \leq \frac{y_i^*}{2}$:
$$\Pr[b_i = 1] \leq 2^{-t}.$$
- If $\frac{y_i^*}{2} < |S_i| < y_i^*$:
$$\Pr[b_i = 1] \leq 1.$$

Thus, if for any $i \in [g]$, $y_i^* \geq 2 \cdot |S_i|$, the verifier's probability of acceptance is bounded by 2^{-t}. Conversely, if $y_i^* \leq 2 \cdot |S_i|$ for all i, and $\sum_{i\in[g]} c_i \cdot y_i^* \leq \sum_{i\in[g]} 2c_i \cdot |S_i| \leq \frac{K}{2}$ and verifier will always reject in the last step. Hence the soundness error of the protocol in Figure 1 is bounded by 2^{-t}.

Efficiency: Both the prover M and verifier A execute LBP for g times. Each execution entails a computational cost of $O(t \cdot poly(m))$ for verifier. Furthermore, the operation to compare $\sum_{i\in[g]} c_i \cdot y_i$ against z is bounded by $poly(m)$. Consequently, the total running time for verifier is $O(g \cdot t \cdot poly(m))$.

The communication cost in the protocol is determined by the vector $y_1, \ldots, y_g$ transmitted by M, along with the g rounds of LBP. Since $y_i \leq 2^m$ for each $i \in [g]$, each y_i can be encoded in m bits. Consequently the overall communication cost is $g \cdot t \cdot poly(m)$ bits. □

4 Upper Bounds

While the existence of IHIP is noteworthy, it is also important to explore the limitations of such constructions. Abadi et al. [AFK89] established an upperbound for *perfect* instance-hiding proofs, showing that problems that have such proofs are contained in $\mathsf{NP/Poly} \cap \mathsf{coNP/Poly}$.

Our work in this section extends these results by showing that every promise problem that has a (δ, ϵ)-IHIP, even with δ, ϵ as large as some small constant, is still contained in $\mathsf{NP/poly} \cap \mathsf{coNP/poly}$.

Theorem 4.1. *Consider functions $\delta, \epsilon : \mathbb{N} \to [0, 1]$ such that for all sufficient large n, we have $\delta(n), \epsilon(n) < \frac{1}{32}$. If a promise problem Π possesses an honest-prover (δ, ϵ)-IHIP, where the verifier can be non-uniform, then $\Pi \in \mathsf{NP/poly} \cap \mathsf{coNP/poly}$.*

Corollary 4.2. $\mathsf{IHIP/poly} \subseteq \mathsf{NP/poly} \cap \mathsf{coNP/poly}$.

Further, we show that if the proof is simulatablely instance-hiding (that is, with an efficient simulator for the honest prover), then the problem is contained in $\mathsf{AM} \cap \mathsf{coAM}$. Here we need the errors to be negligible, though.

Theorem 4.3. *For any negligible functions δ, ϵ, if a promise problem Π has a simulatable honest-prover (δ, ϵ)-IHIP, then both Π and its complement $\bar{\Pi}$ have constant-round public-coin interactive proofs with* uniform *verifiers (that is, an AM proof system).*

Corollary 4.4. $\mathsf{Simulatable\text{-}IHIP} \subseteq \mathsf{AM} \cap \mathsf{coAM}$.

Remark 4.5. Theorems 4.1 and 4.3 can in fact be extended to similar but stronger statements with the hypothesis only requiring protocols that have ϵ-hiding, completeness, and soundness against the honest prover, because our proofs don't depend on the protocols' behavior with malicious provers.

Showing an upper bound for the simulatable-IHIP, where the simulator is efficient, is relatively more straightforward. This efficiency allows the simulator to be used directly in verifying or recognizing the promising problem. Simulators are often used in this manner in efficient reductions when studying concepts like Zero-Knowledge Proof and Randomized Encoding, where a simulator is defined to capture the desired privacy (see, e.g. [BMO90, Vad99, App07]). An upperbound for simulatable-IHIP can leverage this intuition to construct a constant-round public-coin verifier to recognize the language. In the case of general IHIP, however, the simulator is inefficient, necessitating new techniques (and non-uniformity) to establish the bound.

Setup. Before proceeding to the proof of Theorem 4.1 and Theorem 4.3, we set up some notation for IHIP and Simulatable-IHIP proofs and prove some useful preliminary statements. Suppose $\langle \mathsf{P}, \mathsf{V} \rangle$ is a *q-round* (δ, ϵ)-IHIP (or (δ, ϵ)-Simulatable-IHIP) for a promise problem Π, and for any prover P^* denote by $\mathsf{Sim}_{\mathsf{P}^*}$ the corresponding simulator of the prover's view. The only difference between IHIP and Simulatable-IHIP is the efficiency of $\mathsf{Sim}_{\mathsf{P}}$ on simulating the view of honest prover. The $\langle \mathsf{P}, \mathsf{V} \rangle$ can be viewed as two deterministic algorithms on the input, random seed, and public view (transcript) of the protocol:

- $\mathsf{V} : \mathcal{X} \times \mathcal{R}_{\mathsf{V}} \times \Sigma^* \to \mathcal{U} \cup \{0, 1\}$,
- $\mathsf{P} : \mathcal{R}_{\mathsf{P}} \times \Sigma^* \to \mathcal{Y}$,

where $\mathcal{X}$, $\mathcal{R}_{\mathsf{V}}$ (resp., $\mathcal{R}_{\mathsf{P}}$) are the space of possible input instances and verifier's (resp., prover's) random seed respectively, and Σ^* is the space of the public view of the protocol. $\mathcal{U}$ and $\mathcal{Y}$ are verifier's and prover's message space respectively. The outputs $\{0, 1\}$ represent the verifier accepting or rejecting at the end of the interaction. Let $u_i \in \mathcal{U}, y_i \in \mathcal{Y}$ be the messages of verifier and prover at round i respectively, and denote by $\vec{s}_i = (u_1, y_1, \ldots, u_i, y_i)$ the public view up to the end of i^{th} round ($\vec{s}_0 = \phi$). We denote $\mathsf{V}\Big(x, r_{\mathsf{V}}, (u_1, y_1, \ldots, u_i, y_i)\Big) = u_{i+1}$ that V on input instance x, with random seed r_{V}, and current public view $\vec{s}_i$ produces the next message $u_{i+1} \in \mathcal{U}$. Let $r_{\mathsf{V}} \in \mathcal{R}_{\mathsf{V}}, r_{\mathsf{P}} \in \mathcal{R}_{\mathsf{P}}$ be the randomness of V/P respectively. For $i \in [q]$, we have:

- $\mathsf{V}(x, r_{\mathsf{V}}, \vec{s}_i) = u_{i+1}$.
- $\mathsf{P}\left(r_{\mathsf{P}}, (\vec{s}_i, u_{i+1})\right) = y_{i+1}$.

For conciseness, we denote by $\vec{S}_x(r_{\mathsf{V}}, r_{\mathsf{P}})$ the public view in the protocol $\langle \mathsf{P}, \mathsf{V} \rangle$ when the instance is x and random seeds are $r_{\mathsf{V}}, r_{\mathsf{P}}$ respectively (i.e. $\textsc{View}_{\mathsf{P}}\big(\langle \mathsf{P}(n; r_{\mathsf{P}}), \mathsf{V}(x; r_{\mathsf{V}}) \rangle\big) = \Big(r_{\mathsf{P}}, \vec{S}_x(r_{\mathsf{V}}, r_{\mathsf{P}})\Big)$). Abusing notation, $\vec{S}_x$ also represents the output distribution of $\vec{S}_x(U_{\mathcal{R}_{\mathsf{V}}}, U_{\mathcal{R}_{\mathsf{P}}})$. Let $\mathcal{S}$ be the union of supports of $\vec{S}_x$ for all x, and for any x and $\vec{s} \in \mathcal{S}$, define:

- $\beta^x_{\vec{s}} = \{r_{\mathsf{V}} \in \mathcal{R}_{\mathsf{V}} \mid \forall i \in [q] : \mathsf{V}(r_{\mathsf{V}}, x, \vec{s}_{i-1}) = u_i\}$.
- $\alpha^x_{\vec{s}} = \{r_{\mathsf{V}} \in \beta^x_{\vec{s}} \mid \mathsf{V}(x, r_{\mathsf{V}}, \vec{s}) = 1\}$.
- $\gamma_{\vec{s}} = \{r_{\mathsf{P}} \in \mathcal{R}_{\mathsf{P}} \mid \forall i \in [q] : \mathsf{P}\left(r_{\mathsf{P}}, (\vec{s}_{i-1}, u_i)\right) = y_i\}$.

Intuitively, $\beta^x_{\vec{s}}$ (resp. $\gamma_{\vec{s}}$) is the set of V's (resp. P's) randomnesses that, when instance is x, makes V (resp. P) behave as in view $\vec{s}$. $\alpha^x_{\vec{s}}$ is a subset of $\beta^x_{\vec{s}}$ with which V accepts in the end seeing the instance x, randomness r_{V} and public view $\vec{s}$. The following claims apply to any instance-hiding protocol $\langle \mathsf{P}, \mathsf{V} \rangle$.

Claim 4.5.1. *For any $\vec{s} \in \mathcal{S}$ and any x,*

$$\Pr_{(r_{\mathsf{V}}, r_{\mathsf{P}}) \leftarrow \mathcal{R}_{\mathsf{V}} \times \mathcal{R}_{\mathsf{P}}} \left[\vec{S}_x(r_{\mathsf{V}}, r_{\mathsf{P}}) = \vec{s}\right] = \frac{|\beta^x_{\vec{s}}|}{|\mathcal{R}_{\mathsf{V}}|} \cdot \frac{|\gamma_{\vec{s}}|}{|\mathcal{R}_{\mathsf{P}}|}.$$

Proof of Claim 4.5.1. An important observation here is that the randomnesses of P and V are independent given the public view $\vec{s}$.

$$\begin{aligned}\Pr_{(r_\mathsf{V},r_\mathsf{P})\leftarrow\mathcal{R}_\mathsf{V}\times\mathcal{R}_\mathsf{P}}\left[\vec{S}_x(r_\mathsf{V},r_\mathsf{P})=\vec{s}\right] &= \Pr_{(r_\mathsf{V},r_\mathsf{P})\leftarrow\mathcal{R}_\mathsf{V}\times\mathcal{R}_\mathsf{P}}\left[r_\mathsf{V}\in\beta^x_{\vec{s}}\wedge r_\mathsf{P}\in\gamma_{\vec{s}}\right]\\ &= \Pr_{r_\mathsf{V}\leftarrow\mathcal{R}_\mathsf{V}}\left[r_\mathsf{V}\in\beta^x_{\vec{s}}\right]\cdot\Pr_{r_\mathsf{P}\leftarrow\mathcal{R}_\mathsf{P}}\left[r_\mathsf{P}\in\gamma_{\vec{s}}\right]\\ &= \frac{|\beta^x_{\vec{s}}|}{|\mathcal{R}_\mathsf{V}|}\cdot\frac{|\gamma_{\vec{s}}|}{|\mathcal{R}_\mathsf{P}|}.\end{aligned}$$

□

Claim 4.5.2. *For any x,*

$$\Pr_{(r_\mathsf{V},r_\mathsf{P})\leftarrow\mathcal{R}_\mathsf{V}\times\mathcal{R}_\mathsf{P}}\left[\langle\mathsf{P},\mathsf{V}(x)\rangle=1\right]=\mathop{\mathbb{E}}_{\vec{s}\leftarrow\vec{S}_x}\left[\frac{|\alpha^x_{\vec{s}}|}{|\beta^x_{\vec{s}}|}\right].$$

Proof of Claim 4.5.2. Given a public view $\vec{s}\in\mathcal{S}$, the verifier's randomness r_V and prover's randomness r_P are independent. By the definition of $\beta^x_{\vec{s}}$, $\alpha^x_{\vec{s}}$ and $\gamma_{\vec{s}}$, we have:

$$\begin{aligned}\Pr_{r_\mathsf{V},r_\mathsf{P}}\left[\langle\mathsf{P},\mathsf{V}(x)\rangle=1\right] &= \Pr_{r_\mathsf{V},r_\mathsf{P}}\left[\mathsf{V}\big(x,r_\mathsf{V},\vec{S}_x(r_\mathsf{V},r_\mathsf{P})\big)=1\right] &(1)\\ &= \sum_{\vec{s}\in\mathcal{S}}\Pr_{(r_\mathsf{V},r_\mathsf{P})\leftarrow\mathcal{R}_\mathsf{V}\times\mathcal{R}_\mathsf{P}}\left[\vec{S}_x(r_\mathsf{V},r_\mathsf{P})=\vec{s}\ \wedge\ \mathsf{V}\big(x,r_\mathsf{V},\vec{s}\big)=1\right] &(2)\\ &= \sum_{\vec{s}\in\mathcal{S}}\Pr\left[\vec{S}_x=\vec{s}\right]\cdot\Pr_{(r_\mathsf{V},r_\mathsf{P})\leftarrow\beta^x_{\vec{s}}\times\gamma_{\vec{s}}}\left[\mathsf{V}(x,r_\mathsf{V},\vec{s})=1\right] &(3)\\ &= \sum_{\vec{s}\in\mathcal{S}}\Pr\left[\vec{S}_x=\vec{s}\right]\cdot\Pr_{(r_\mathsf{V},r_\mathsf{P})\leftarrow\beta^x_{\vec{s}}\times\gamma_{\vec{s}}}\left[r_\mathsf{V}\in\alpha^x_{\vec{s}}\right] &(4)\\ &= \sum_{\vec{s}\in\mathcal{S}}\Pr\left[\vec{S}_x=\vec{s}\right]\cdot\Pr_{r_\mathsf{V}\leftarrow\beta^x_{\vec{s}}}\left[r_\mathsf{V}\in\alpha^x_{\vec{s}}\right] &(5)\\ &= \sum_{\vec{s}\in\mathcal{S}}\Pr\left[\vec{S}_x=\vec{s}\right]\cdot\frac{|\alpha^x_{\vec{s}}|}{|\beta^x_{\vec{s}}|} &(6)\\ &= \mathop{\mathbb{E}}_{\vec{s}\leftarrow\vec{S}}\left[\frac{|\alpha^x_{\vec{s}}|}{|\beta^x_{\vec{s}}|}\right]. &(7)\end{aligned}$$

□

Claim 4.5.3.

$$\left|\mathop{\mathbb{E}}_{\vec{s}\leftarrow\vec{S}}\left[\frac{|\alpha^x_{\vec{s}}|}{|\beta^x_{\vec{s}}|}\right]-\mathop{\mathbb{E}}_{(r_\mathsf{P},\vec{s})\leftarrow\mathsf{Sim}_\mathsf{P}(n)}\left[\frac{|\alpha^x_{\vec{s}}|}{|\beta^x_{\vec{s}}|}\right]\right|\le 2\cdot\epsilon.$$

Proof of Claim 4.5.3.

$$\begin{aligned}\left|\mathop{\mathbb{E}}_{\vec{s}\leftarrow\vec{S}}\left[\frac{|\alpha^x_{\vec{s}}|}{|\beta^x_{\vec{s}}|}\right]-\mathop{\mathbb{E}}_{(r_\mathsf{P},\vec{s})\leftarrow\mathsf{Sim}_\mathsf{P}(n)}\left[\frac{|\alpha^x_{\vec{s}}|}{|\beta^x_{\vec{s}}|}\right]\right| &\le \sum_{\vec{s}\in\mathcal{S}}\frac{|\alpha^x_{\vec{s}}|}{|\beta^x_{\vec{s}}|}\cdot\left|\Pr\left[\vec{S}=\vec{s}\right]-\Pr\left[\mathsf{Sim}_\mathsf{P}=(*,\vec{s})\right]\right|\\ &\le \sum_{\vec{s}\in\mathcal{S}}\left|\Pr\left[\vec{S}=\vec{s}\right]-\Pr\left[\mathsf{Sim}_\mathsf{P}=(*,\vec{s})\right]\right|\\ &\le 2\epsilon.\end{aligned}$$

where the first inequality follows triangle inequality and the second inequality follows the fact that $|\frac{\alpha^x_{\vec{s}}|}{|\beta^x_{\vec{s}}|}\le 1$ for all $x,\vec{s}$. □

4.1 Constant-Round Proofs from Simulatable Instance-Hiding

Theorem 4.3. *For any negligible functions δ, ϵ, if a promise problem Π has a simulatable honest-prover (δ, ϵ)-IHIP, then both Π and its complement $\bar{\Pi}$ have constant-round public-coin interactive proofs with* uniform *verifiers (that is, an* AM *proof system).*

Proof of Theorem 4.3. Given that $\langle \mathsf{P}, \mathsf{V} \rangle$ is a honest-prover (δ, ϵ)-Simulatable-IHIP for a promise problem Π, Claim 4.5.2 suggests that $\Pr_{r_\mathsf{V}, r_\mathsf{P}}[\langle \mathsf{P}, \mathsf{V}(x) \rangle = 1] = \mathbb{E}_{\vec{s} \leftarrow \vec{S}_x}\left[\frac{|\alpha^x_{\vec{s}}|}{|\beta^x_{\vec{s}}|}\right]$. By the correctness of the honest prover, for any instance $x \in \textsc{Yes}(\Pi_n) \cup \textsc{No}(\Pi_n)$:

- If $x \in \textsc{Yes}(\Pi_n)$:
$$\mathbb{E}_{\vec{s} \leftarrow \vec{S}_x}\left[\frac{|\alpha^x_{\vec{s}}|}{|\beta^x_{\vec{s}}|}\right] \geq 1 - \delta(n).$$
- If $x \in \textsc{No}(\Pi_n)$:
$$\mathbb{E}_{\vec{s} \leftarrow \vec{S}_x}\left[\frac{|\alpha^x_{\vec{s}}|}{|\beta^x_{\vec{s}}|}\right] \leq \delta(n).$$

We will prove the theorem by constructing a constant-round interactive protocol to prove that $\mathbb{E}_{\vec{s} \leftarrow \vec{S}_x}\left[\frac{|\alpha^x_{\vec{s}}|}{|\beta^x_{\vec{s}}|}\right]$ is large. Such a protocol can be made if (1) The distribution $\vec{S}_x$ can be sampled by verifier; (2) There is a constant-round interactive protocol protocol, for a given $\vec{s}$, to prove that $\frac{|\alpha^x_{\vec{s}}|}{|\beta^x_{\vec{s}}|}$ is large. Although the distribution $\vec{S}_x$ may not be efficiently samplable, the Sim_P in Simulatable-IHIP is efficient, and thus verifier can call $\mathsf{Sim}_\mathsf{P}(n)$ to sample a distribution that is ϵ-close to $\vec{S}_x$. For condition (2), we note that both $\alpha^x_{\vec{s}}$ and $\beta^x_{\vec{s}}$ are efficiently verifiable, and thus we can use [GS86] to prove a lowerbound for $|\alpha^x_{\vec{s}}|$. To complete the proof, the protocol should also prove an upperbound for $|\beta^x_{\vec{s}}|$. [For87] proposed a set upperbound protocol, which require the verifier being able to sample a private uniform random element from the set. Here, however, we don't know how to sample a random private element from $\beta^x_{\vec{s}}$ efficiently. In fact suppose we can, the promise problem will collapse to BPP because the verifier's view can be efficiently sampled, and thus this is unlikely.

We do, however, know how to upperbound the probability $\Pr[\mathsf{Sim}_\mathsf{P}(n) = (r_\mathsf{P}, \vec{s})]$ for a random $(r_\mathsf{P}, \vec{s}) \leftarrow \mathsf{Sim}_\mathsf{P}(n)$. This is achieved by running [For87]'s upperbound protocol, as described in Lemma 4.6, on the set of simulator's random seeds that produce $(r_\mathsf{P}, \vec{s})$. Specifically, we denote by $\mathcal{R}_{\mathsf{Sim}_\mathsf{P}}$ the simulator's random space and let $\zeta_{r_\mathsf{P}, \vec{s}} = \{r_{\mathsf{Sim}} \mid \mathsf{Sim}_\mathsf{P}(n; r_{\mathsf{Sim}}) = (r_\mathsf{P}, \vec{s})\}$. On a random drawn $r_{\mathsf{Sim}} \leftarrow \mathcal{R}_{\mathsf{Sim}}$, let $(r_\mathsf{P}, \vec{s}) \leftarrow \mathsf{Sim}_\mathsf{P}(n; r_{\mathsf{Sim}})$ be the simulated view. The used random seed r_{Sim} for sampling $(r_\mathsf{P}, \vec{s})$ can be kept in private as the random element of $\zeta_{r_\mathsf{P}, \vec{s}}$ in the upper-bound protocol. The probability that the simulator simulates the view $(r_\mathsf{P}, \vec{s})$ is $\Pr[\mathsf{Sim}_\mathsf{P}(n) = (r_\mathsf{P}, \vec{s})] = \frac{|\zeta_{r_\mathsf{P}, \vec{s}}|}{|\mathcal{R}_{\mathsf{Sim}}|}$, where the value $|\mathcal{R}_{\mathsf{Sim}_\mathsf{P}}|$ is efficiently computable because Sim_P is a uniform efficient algorithm. Furthermore, for any two randomness $r, r' \in \zeta_{r_\mathsf{P}, \vec{s}}$:

$$\Pr_{r_{\mathsf{Sim}}}[r_{\mathsf{Sim}} = r | \mathsf{Sim}_\mathsf{P}(n) = (r_\mathsf{P}, \vec{s})] = \Pr_{r_{\mathsf{Sim}}}[r_{\mathsf{Sim}} = r' | \mathsf{Sim}_\mathsf{P}(n) = (r_\mathsf{P}, \vec{s})] = \frac{1}{|\zeta_{r_\mathsf{P}, \vec{s}}|}.$$

Thus, on a random simulated view $\vec{s}$ sampled with randomness r_{Sim}, given only $\vec{s}$, the private r_{Sim} is a random element from set $\zeta_{r_\mathsf{P}, \vec{s}}$. Similarly, define $\zeta_{r_\mathsf{P}} = \{r_{\mathsf{Sim}} \mid \mathsf{Sim}_\mathsf{P}(n; r_{\mathsf{Sim}}) = (r_\mathsf{P}, *)\}$, that is the set of simulator's randomness with which the simulated prover's randomness is r_P. We estimate $\frac{|\beta^x_{\vec{s}}|}{|\mathcal{R}_\mathsf{V}|}$ by establishing an upper bound for $\frac{|\zeta_{r_\mathsf{P}, \vec{s}}|}{|\zeta_{r_\mathsf{P}}|}$. This involves executing a [GS86] lower-bound protocol on $|\zeta_{r_\mathsf{P}}|$ and a [For87] upper-bound protocol on $|\zeta_{r_\mathsf{P}, \vec{s}}|$.

Lemma 4.6 (Set Upper-Bound Protocol [For87])**.** *For any $m \in \mathbb{N}$ and set $S \subseteq \{0,1\}^m$, there is a 2-message public-coin interactive protocol where the prover and verifier are given access to the membership oracle for*

S and an public input $K \in \mathbb{N}$, and the verifier is additionally given a random element of S as private input, which has the following properties:

- *When interacting with the honest prover:* $\Pr[\textit{verifier accepts}] \geq 1 - \frac{|S|-1}{K}$.
- *When interacting with any prover:* $\Pr[\textit{verifier accepts}] \leq \frac{6 \cdot K}{|S|-1}$.

Given the necessary tools, we construct our constant-round IP protocol as shown in Figure 2.

We would like to get an estimated upperbound $k_{\beta^x_{\vec{s}(i)}}$ for $|\beta^x_{\vec{s}(i)}|$ by bounding $\frac{|\zeta_{r_\mathsf{P},\vec{s}}|}{|\zeta_{r_\mathsf{P}}|}$ for random r_P. Ideally, we require the estimated upperbound $k_{\beta^x_{\vec{s}(i)}}$ be ceiled by $p(|\beta^x_{\vec{s}(i)}|)$ for some polynomial $p(\cdot)$, which is enough for us to differentiate the super-polynomial gap of $\mathbb{E}[\frac{|\alpha^x_{\vec{s}}|}{|\beta^x_{\vec{s}}|}]$ between YES and NO cases. Recall that the efficient simulator Sim_P outputs a distribution composed of simulated prover randomness and public view of the protocol. We notice that, with high probability over a simulated prover's randomness r_P, the simulated public view conditioned on the fixed randomness r_P is statistically close to the protocol public view conditioned on the fixed r_P.

To formalize the intuition, for any $r_\mathsf{P} \in \mathcal{R}_\mathsf{P}$, denote the set of possible public views produced by the simulator and protocol conditioned on r_P by $\mathcal{S}_{\mathsf{Sim}_\mathsf{P},r_\mathsf{P}} = \left\{\vec{s} \in \mathcal{S} |\ \Pr\left[\mathsf{Sim}_\mathsf{P}(n) = (r_\mathsf{P}, \vec{s})\right] > 0\right\}$ and $\mathcal{S}_{\langle \mathsf{P},\mathsf{V}\rangle,r_\mathsf{P}} = \left\{\vec{s} \in \mathcal{S} |\ \Pr\left[\textsc{View}_\mathsf{P}\big(\mathsf{P}(n; r_\mathsf{P}), \mathsf{V}(x)\big) = (r_\mathsf{P}, \vec{s})\right] > 0\right\}$ respectively. It's worth noticing that Sim_P can possibly output $(r_\mathsf{P}, \vec{s})$ that never appears in the real protocol due to its hiding error, in which case $\mathcal{S}_{\mathsf{Sim}_\mathsf{P},r_\mathsf{P}} \backslash \mathcal{S}_{\langle \mathsf{P},\mathsf{V}\rangle,r_\mathsf{P}}$ is non-empty. Define $R_\mathsf{P}\text{-}BAD = \left\{ r_\mathsf{P} \middle|\ \sum_{\vec{s} \in \mathcal{S}_{\langle \mathsf{P},\mathsf{V}\rangle,r_\mathsf{P}}} \left| \frac{|\zeta_{r_\mathsf{P},\vec{s}}|}{|\zeta_{r_\mathsf{P}}|} - \frac{|\beta^x_{\vec{s}}|}{|\mathcal{R}_\mathsf{V}|} \right| > \sqrt{\epsilon} \right\}$, where ϵ is the hiding error.

Claim 4.6.1.

$$\Pr_{(r_\mathsf{P},*) \leftarrow \mathsf{Sim}_\mathsf{P}(n)} \Big[r_\mathsf{P} \in R_\mathsf{P}\text{-}BAD \Big] \leq 4\sqrt{\epsilon},$$

where $(r_\mathsf{P}, *)$ denotes views with prover's randomness being r_P.

Proof of Claim 4.6.1.

Fact 4.7 (See e.g., [KRV21, Fact 2.6])**.** *For any jointly distributed X_0, X_1, and Y it holds that*

$$\Delta\big((X_0, Y), (X_1, Y)\big) = \mathop{\mathbb{E}}_{y \leftarrow Y} \Big[\Delta\big(X_0 | Y = y, X_1 | Y = y\big) \Big].$$

Fact 4.8 (See, e.g. [KRV21, Fact 2.3])**.** *Let (X_0, Y_0) and (X_1, Y_1) be two pairs of jointly distributed random variables s.t. $\mathrm{supp}(X_0) = \mathrm{supp}(X_1)$ and for every $x \in \mathrm{supp}(X_0)$, the distributions $Y_0|X_0 = x, Y_1|X_1 = x$ are identically distributed. Then,*

$$\Delta\big((X_0, Y_0), (X_1, Y_1)\big) = \Delta(X_0, X_1).$$

Denote by R_0 the uniform distribution over $\mathcal{R}_\mathsf{P}$, and let R_1 be the distribution that samples $(r_\mathsf{P}, *) \leftarrow \mathsf{Sim}_\mathsf{P}(n)$, outputting r_P. For any choice of r_P, let $\vec{S}_1(r_\mathsf{P})$ denote the simulated public view conditioned on simulated prover randomness being r_P. Thus $(R_1, \vec{S}_1(R_1)$ represent the output distribution of $\mathsf{Sim}_\mathsf{P}(n)$. Consider the following random process on input r_P:

$$\vec{S}_0(r_\mathsf{P}) : (r_\mathsf{P}, \vec{s}) \leftarrow \textsc{View}_\mathsf{P}\big(\langle \mathsf{P}(n; r_\mathsf{P}), \mathsf{V}(x)\rangle\big), \text{ outputs } \vec{s}.$$

Because of ϵ-hiding, the distance between R_0 and R_1 is bounded:

$$\Delta\big(R_0, R_1\big) \leq \Delta\Big(\big(R_0, \vec{S}_0(R_0)\big), \big(R_1, \vec{S}_1(R_1)\big)\Big) = \Delta\big(\textsc{View}_\mathsf{P}\big(\langle \mathsf{P}, \mathsf{V}(x)\rangle\big), \mathsf{Sim}_\mathsf{P}(n)\big) \leq \epsilon.$$

Follows Fact 4.8, we have:

$$\Delta\Big(\big(R_0, \vec{S}_0(R_0)\big), \big(R_1, \vec{S}_0(R_1)\big)\Big) = \Delta(R_0, R_1) \leq \epsilon.$$

Intuitively, this means the the distribution of P's view will not change much even if we sample prover's randomness using $\mathsf{Sim}_{\mathsf{P}}$ (instead of uniformly). Therefore, from triangle inequality, the distance $\Delta\big((R_1, \vec{S}_0(R_1)), (R_1, \vec{S}_1(R_1))\big) < 2\epsilon$ and by Fact 4.7 we have:

$$\begin{aligned}
\mathop{\mathbb{E}}_{(r_{\mathsf{P}},*)\leftarrow\mathsf{Sim}_{\mathsf{P}}(n)}\left[\sum_{\vec{s}\in\mathcal{S}_{\langle\mathsf{P},\mathsf{V}\rangle,r_{\mathsf{P}}}}\left|\frac{|\zeta_{r_{\mathsf{P}},\vec{s}}|}{|\zeta_{r_{\mathsf{P}}}|}-\frac{|\beta^x_{\vec{s}}|}{|\mathcal{R}_{\mathsf{V}}|}\right|\right] &= 2\cdot\mathop{\mathbb{E}}_{r_{\mathsf{P}}\leftarrow R_1}\left[\Delta(\vec{S}_0(r_{\mathsf{P}}),\vec{S}_1(r_{\mathsf{P}}))\right]-\mathop{\mathbb{E}}_{r_{\mathsf{P}}\leftarrow R_1}\left[\sum_{\vec{s}\in\mathcal{S}_{\mathsf{Sim}_{\mathsf{P}},r_{\mathsf{P}}}\setminus\mathcal{S}_{\langle\mathsf{P},\mathsf{V}\rangle,r_{\mathsf{P}}}}\frac{|\zeta_{r_{\mathsf{P}},\vec{s}}|}{|\zeta_{r_{\mathsf{P}}}|}\right]\\
&\le 2\cdot\mathop{\mathbb{E}}_{r_{\mathsf{P}}\leftarrow R_1}\left[\Delta(\vec{S}_0(r_{\mathsf{P}}),\vec{S}_1(r_{\mathsf{P}}))\right]\\
&= 2\cdot\Delta\big((R_1,\vec{S}_0(R_1)),(R_1,\vec{S}_1(R_1))\big)\\
&< 4\epsilon.
\end{aligned}$$

Apply Markov bound and the claim follows. □

This observation is important because we need to fix some prover's "good" randomness r_{P} in order to estimate $\frac{|\beta^x_{\vec{s}}|}{|\mathcal{R}_{\mathsf{V}}|}$ using $\frac{\zeta_{r_{\mathsf{P}},\vec{s}}}{\zeta_{r_{\mathsf{P}}}}$. Formally, a random string r_{P} is R_{P}-*good* if $r_{\mathsf{P}} \notin R_{\mathsf{P}}$-$BAD$. For any r_{P}, denote by $\zeta_{r_{\mathsf{P}}}$-BAD the set of $\vec{s}\in\mathcal{S}_{\langle\mathsf{P},\mathsf{V}\rangle,r_{\mathsf{P}}}$ such that $\left|\frac{|\zeta_{r_{\mathsf{P}},\vec{s}}|}{|\zeta_{r_{\mathsf{P}}}|}-\frac{|\beta^x_{\vec{s}}|}{|\mathcal{R}_{\mathsf{V}}|}\right| > \frac{|\zeta_{r_{\mathsf{P}},\vec{s}}|}{|\zeta_{r_{\mathsf{P}}}|}$.

Claim 4.8.1. *For any r_{P} that is R_{P}-good,*

$$\Pr_{(r^*_{\mathsf{P}},\vec{s})\leftarrow\mathsf{Sim}_{\mathsf{P}}(n)}\left[\vec{s}\in\zeta_{r_{\mathsf{P}}}\text{-}BAD\middle|r^*_{\mathsf{P}}=r_{\mathsf{P}}\right]\le\sqrt{\epsilon}.$$

Proof.

$$\begin{aligned}
\Pr_{(r^*_{\mathsf{P}},\vec{s})\leftarrow\mathsf{Sim}_{\mathsf{P}}(n)}\left[\vec{s}\in\zeta_{r_{\mathsf{P}}}\text{-}BAD\middle|r^*_{\mathsf{P}}=r_{\mathsf{P}}\right] &\le \sum_{\vec{s}\in\zeta_{r_{\mathsf{P}}}\text{-}BAD}\frac{|\zeta_{r_{\mathsf{P}},\vec{s}}|}{|\zeta_{r_{\mathsf{P}}}|}\\
&\le \sum_{\vec{s}\in\zeta_{r_{\mathsf{P}}}\text{-}BAD}\left|\frac{|\zeta_{r_{\mathsf{P}},\vec{s}}|}{|\zeta_{r_{\mathsf{P}}}|}-\frac{|\beta^x_{\vec{s}}|}{|\mathcal{R}_{\mathsf{V}}|}\right|\\
&\le \sqrt{\epsilon}.
\end{aligned}$$

where the first inequality follows definition of conditional probability, the second inequality from the definition of $\zeta_{r_{\mathsf{P}}}$-BAD, and the last inequality follows the fact that r_{P} is r_{P}-good and $\zeta_{r_{\mathsf{P}}}$-$BAD \subseteq \mathcal{S}_{\langle\mathsf{P},\mathsf{V}\rangle,r_{\mathsf{P}}}$. □

Similarly, let $\beta_{r_{\mathsf{P}}}$-BAD denote the set of $\vec{s}\in\mathcal{S}_{\langle\mathsf{P},\mathsf{V}\rangle,r_{\mathsf{P}}}$ such that $\left|\frac{|\zeta_{r_{\mathsf{P}},\vec{s}}|}{|\zeta_{r_{\mathsf{P}}}|}-\frac{|\beta^x_{\vec{s}}|}{|\mathcal{R}_{\mathsf{V}}|}\right| > \frac{|\beta^x_{\vec{s}}|}{|\mathcal{R}_{\mathsf{V}}|}$.

Claim 4.8.2. *For any r_{P} that is R_{P}-good,*

$$\Pr_{(r^*_{\mathsf{P}},\vec{s})\leftarrow\mathsf{Sim}_{\mathsf{P}}(n)}\left[\vec{s}\in\beta_{r_{\mathsf{P}}}\text{-}BAD\middle|r^*_{\mathsf{P}}=r_{\mathsf{P}}\right]\le 2\cdot\sqrt{\epsilon}.$$

Proof.

$$\begin{aligned}
\Pr_{(r^*_{\mathsf{P}},\vec{s})\leftarrow\mathrm{VIEW}_{\mathsf{P}}\left(\langle\mathsf{P},\mathsf{V}(x)\rangle\right)}\left[\vec{s}\in\beta_{r_{\mathsf{P}}}\text{-}BAD\middle|r^*_{\mathsf{P}}=r_{\mathsf{P}}\right] &= \sum_{\vec{s}\in\beta_{r_{\mathsf{P}}}\text{-}BAD}\frac{|\beta^x_{\vec{s}}|}{|\mathcal{R}_{\mathsf{V}}|}\\
&\le \sum_{\vec{s}\in\beta_{r_{\mathsf{P}}}\text{-}BAD}\left|\frac{|\zeta_{r_{\mathsf{P}},\vec{s}}|}{|\zeta_{r_{\mathsf{P}}}|}-\frac{|\beta^x_{\vec{s}}|}{|\mathcal{R}_{\mathsf{V}}|}\right|\\
&\le \sqrt{\epsilon}.
\end{aligned}$$

Define $\vec{S}_0(\cdot), \vec{S}_1(\cdot)$ as in the proof of Claim 4.6.1, given that r_{P} is R_{P}-good, the distance between $\vec{S}_0(r_{\mathsf{P}})$ and $\vec{S}_1(r_{\mathsf{P}})$ is bounded:

$$\begin{aligned}\Delta\left(\vec{S}_0(r_{\mathsf{P}}), \vec{S}_1(r_{\mathsf{P}})\right) &= \frac{1}{2}\cdot\left(\sum_{\vec{s}\in\mathcal{S}_{\langle \mathsf{P},\mathsf{V}\rangle, r_{\mathsf{P}}}}\left|\frac{|\zeta_{r_{\mathsf{P}},\vec{s}}|}{|\zeta_{r_{\mathsf{P}}}|} - \frac{|\beta^x_{\vec{s}}|}{|\mathcal{R}_{\mathsf{V}}|}\right|\right) + \frac{1}{2}\cdot\left(\sum_{\vec{s}\in\mathcal{S}_{\mathsf{Sim}_{\mathsf{P}}, r_{\mathsf{P}}}\setminus\mathcal{S}_{\langle \mathsf{P},\mathsf{V}\rangle, r_{\mathsf{P}}}}\frac{|\zeta_{r_{\mathsf{P}},\vec{s}}|}{|\zeta_{r_{\mathsf{P}}}|}\right)\\ &\le \frac{1}{2}\cdot\sqrt{\epsilon} + \frac{1}{2}\epsilon\\ &\le \sqrt{\epsilon}.\end{aligned}$$

Following triangle inequality and data processing inequality,

$$\begin{aligned}\Pr_{(r^*_{\mathsf{P}},\vec{s})\leftarrow\mathsf{Sim}_{\mathsf{P}}(n)}\left[\vec{s}\in\beta_{r_{\mathsf{P}}}\text{-}BAD\middle| r^*_{\mathsf{P}} = r_{\mathsf{P}}\right] &\le \Pr_{(r^*_{\mathsf{P}},\vec{s})\leftarrow\textsc{View}_{\mathsf{P}}\left(\langle \mathsf{P},\mathsf{V}(x)\rangle\right)}\left[\vec{s}\in\beta_{r_{\mathsf{P}}}\text{-}BAD\middle| r^*_{\mathsf{P}} = r_{\mathsf{P}}\right] + \Delta\left(\vec{S}_0(r_{\mathsf{P}}), \vec{S}_1(r_{\mathsf{P}})\right)\\ &\le 2\cdot\sqrt{\epsilon}.\end{aligned}$$

Thus the proof is completed. □

We say $(r_{\mathsf{P}}, \vec{s})$ is $\textsc{View}_{\mathsf{P}}$-good if $r_{\mathsf{P}} \notin R_{\mathsf{P}}\text{-}BAD$, conditioned on which $\vec{s}\notin\beta_{r_{\mathsf{P}}}\text{-}BAD\cup\zeta_{r_{\mathsf{P}}}\text{-}BAD$.

$$\Pr_{(r_{\mathsf{P}},\vec{s})\leftarrow\mathsf{Sim}_{\mathsf{P}}(n)}[(r_{\mathsf{P}},\vec{s}) \text{ is } \textsc{View}_{\mathsf{P}}\text{-good}] = \Pr[r_{\mathsf{P}}\notin R_{\mathsf{P}}\text{-}BAD]\cdot\Pr[\vec{s}\notin\beta_{r_{\mathsf{P}}}\text{-}BAD\cup\zeta_{r_{\mathsf{P}}}\text{-}BAD | r_{\mathsf{P}}\notin R_{\mathsf{P}}\text{-}BAD] \tag{8}$$

$$\ge (1-4\sqrt{\epsilon})(1-2\sqrt{\epsilon}-\sqrt{\epsilon}) \tag{9}$$

$$\ge 1-7\sqrt{\epsilon}. \tag{10}$$

Claim 4.8.3. *For any $(r_{\mathsf{P}}, \vec{s})$ that is $\textsc{View}_{\mathsf{P}}$-good:*

$$\frac{1}{2}\frac{|\beta^x_{\vec{s}}|}{|\mathcal{R}_{\mathsf{V}}|} \le \frac{|\zeta_{r_{\mathsf{P}},\vec{s}}|}{|\zeta_{r_{\mathsf{P}}}|} \le 2\frac{|\beta^x_{\vec{s}}|}{|\mathcal{R}_{\mathsf{V}}|}.$$

Claim 4.8.4.

- *If $x\in\textsc{Yes}(\Pi_n)$:*

$$\mathop{\mathbb{E}}_{(r_{\mathsf{P}},\vec{s})\leftarrow\mathsf{Sim}_{\mathsf{P}}(n)}\left[\frac{|\alpha^x_{\vec{s}}|\cdot|\zeta_{r_{\mathsf{P}}}|}{|\zeta_{r_{\mathsf{P}},\vec{s}}|}\middle|(r_{\mathsf{P}},\vec{s}) \textit{ is } \textsc{View}_{\mathsf{P}}\textit{-good}\right]\cdot\frac{1}{|\mathcal{R}_{\mathsf{V}}|} > \frac{1}{3}.$$

- *If $x\in\textsc{No}(\Pi_n)$:*

$$\mathop{\mathbb{E}}_{(r_{\mathsf{P}},\vec{s})\leftarrow\mathsf{Sim}_{\mathsf{P}}(n)}\left[\frac{|\alpha^x_{\vec{s}}|\cdot|\zeta_{r_{\mathsf{P}}}|}{|\zeta_{r_{\mathsf{P}},\vec{s}}|}\middle|(r_{\mathsf{P}},\vec{s}) \textit{ is } \textsc{View}_{\mathsf{P}}\textit{-good}\right]\cdot\frac{1}{|\mathcal{R}_{\mathsf{V}}|} \le 4\cdot(\delta+2\cdot\epsilon).$$

Proof of Claim 4.8.4. Denote by G the event that $(r_{\mathsf{P}},\vec{s})$ is $\textsc{View}_{\mathsf{P}}$-good. On YES instance $x\in\textsc{Yes}(\Pi_n)$,

$$\begin{aligned}\mathop{\mathbb{E}}_{(r_{\mathsf{P}},\vec{s})\leftarrow\mathsf{Sim}_{\mathsf{P}}}\left[\frac{|\alpha^x_{\vec{s}}|\cdot|\zeta_{r_{\mathsf{P}}}|}{|\zeta_{r_{\mathsf{P}},\vec{s}}|}\middle|(r_{\mathsf{P}},\vec{s}) \text{ is } \textsc{View}_{\mathsf{P}}\text{-good}\right]\cdot\frac{1}{|\mathcal{R}_{\mathsf{V}}|} &\ge \frac{1}{2}\cdot\mathop{\mathbb{E}}_{(r_{\mathsf{P}},\vec{s})\leftarrow\mathsf{Sim}_{\mathsf{P}}}\left[\frac{|\alpha^x_{\vec{s}}|}{|\beta^x_{\vec{s}}|}\middle|G\right]\\ &= \frac{1}{2}\cdot\left(\mathop{\mathbb{E}}_{(r_{\mathsf{P}},\vec{s})\leftarrow\mathsf{Sim}_{\mathsf{P}}}\left[\frac{|\alpha^x_{\vec{s}}|}{|\beta^x_{\vec{s}}|}\right] - \mathop{\mathbb{E}}_{(r_{\mathsf{P}},\vec{s})}\left[\frac{|\alpha^x_{\vec{s}}|}{|\beta^x_{\vec{s}}|}\middle|\bar{G}\right]\cdot\Pr_{(r_{\mathsf{P}},\vec{s})}[\bar{G}]\right)\cdot\left(\frac{1}{\Pr_{(r_{\mathsf{P}},\vec{s})}[G]}\right)\\ &\ge \frac{1}{2}\cdot\left(\mathop{\mathbb{E}}_{(r_{\mathsf{P}},\vec{s})\leftarrow\mathsf{Sim}_{\mathsf{P}}}\left[\frac{|\alpha^x_{\vec{s}}|}{|\beta^x_{\vec{s}}|}\right] - \Pr_{(r_{\mathsf{P}},\vec{s})\leftarrow\mathsf{Sim}_{\mathsf{P}}}[\bar{G}]\right)\cdot\left(\frac{1}{\Pr_{(r_{\mathsf{P}},\vec{s})\leftarrow\mathsf{Sim}_{\mathsf{P}}}[G]}\right)\\ &\ge \frac{1}{2}\cdot(1-\delta-2\epsilon-7\sqrt{\epsilon})\cdot\frac{1}{1-7\sqrt{\epsilon}}\\ &> \frac{1}{3}.\end{aligned}$$

where the line (1) follows Claim 4.8.3, line (2) follows conditional expectation, line (3) follows from Claim 4.5.3, and the line (4) follows Eq. (10) and Claim 4.8.3.

Similarly, for NO instance $x \in \text{NO}(\Pi_n)$:

$$\begin{aligned}
\mathop{\mathbb{E}}_{(r_\mathsf{P},\vec{s})\leftarrow\mathsf{Sim}_\mathsf{P}}\left[\frac{|\alpha^x_{\vec{s}}|\cdot|\zeta_{r_\mathsf{P}}|}{|\zeta_{r_\mathsf{P},\vec{s}}|}\middle|(r_\mathsf{P},\vec{s}) \text{ is } \textsc{View}_\mathsf{P}\text{-good}\right]\cdot\frac{1}{|\mathcal{R}_\mathsf{V}|} &\le 2\cdot\mathop{\mathbb{E}}_{(r_\mathsf{P},\vec{s})\leftarrow\mathsf{Sim}_\mathsf{P}}\left[\frac{|\alpha^x_{\vec{s}}|}{|\beta^x_{\vec{s}}|}\middle|G\right]\\
&\le 2\cdot\mathop{\mathbb{E}}_{(r_\mathsf{P},\vec{s})\leftarrow\mathsf{Sim}_\mathsf{P}}\left[\frac{|\alpha^x_{\vec{s}}|}{|\beta^x_{\vec{s}}|}\right]\cdot\left(\frac{1}{\Pr\limits_{(r_\mathsf{P},\vec{s})\leftarrow\mathsf{Sim}_\mathsf{P}}[G]}\right)\\
&\le 2\cdot(\delta+2\cdot\epsilon)\cdot\frac{1}{1-7\sqrt{\epsilon}}\\
&\le 4\cdot(\delta+2\cdot\epsilon),
\end{aligned}$$

where line (1) follows Claim 4.8.3, line (2) follows conditional expectation and line (3) follows Claim 4.5.3 and triangle inequality. □

Overall, we get a large gap on $\mathop{\mathbb{E}}_{(r_\mathsf{P},\vec{s})\leftarrow\mathsf{Sim}_\mathsf{P}}\left[\frac{|\alpha^x_{\vec{s}}|\cdot|\zeta_{r_\mathsf{P}}|}{|\zeta_{r_\mathsf{P},\vec{s}}|}\middle|(r_\mathsf{P},\vec{s}) \text{ is } \textsc{View}_\mathsf{P}\text{-good}\right]\cdot\frac{1}{|\mathcal{R}_\mathsf{V}|}$, between YES and NO cases. Looking into the execution of the protocol in Figure 2, on the sampled views $(r_\mathsf{P}^{(1)},\vec{s}^{(1)}),\ldots,(r_\mathsf{P}^{(g)},\vec{s}^{(g)})$:

Claim 4.8.5.

- *If* $x \in \text{YES}(\Pi_n)$*:*

$$\Pr\left[\frac{1}{g}\frac{1}{|\mathcal{R}_\mathsf{V}|}\sum_{i\in[g]}\frac{|\alpha^x_{\vec{s}^{(i)}}|\cdot|\zeta_{r_\mathsf{P}^{(i)}}|}{|\zeta_{r_\mathsf{P}^{(i)},\vec{s}^{(i)}}|}<\frac{1}{6}\right]\le\text{negl}(n).$$

- *If* $x \in \text{NO}(\Pi_n)$,

$$\Pr_{(r_\mathsf{P}^{(1)},\vec{s}^{(1)}),\ldots,(r_\mathsf{P}^{(g)},\vec{s}^{(g)})\leftarrow\mathsf{Sim}_\mathsf{P}}\left[\exists i\in[g]:\ \frac{|\alpha^x_{\vec{s}^{(i)}}|\cdot|\zeta_{r_\mathsf{P}^{(i)}}|}{|\zeta_{r_\mathsf{P}^{(i)},\vec{s}^{(i)}}|\cdot|\mathcal{R}_\mathsf{V}|}>\sqrt{\delta+\epsilon}\right]\le\text{negl}(n).$$

Proof of Claim 4.8.5. Consider the $(r_\mathsf{P}^{(1)},\vec{s}^{(1)}),\ldots,(r_\mathsf{P}^{(g)},\vec{s}^{(g)})$ sampled in Figure 2, define event E as the occurrence that $(r_\mathsf{P}^{(1)},\vec{s}^{(1)}),\ldots,(r_\mathsf{P}^{(g)},\vec{s}^{(g)})$ are all $\textsc{View}_\mathsf{P}$-good.

- If $x \in \text{YES}(\Pi_n)$, conditioned on the occurrence of E, following Claim 4.8.3, each $\frac{|\alpha^x_{\vec{s}^{(i)}}|\cdot|\zeta_{r_\mathsf{P}^{(i)}}|}{|\zeta_{r_\mathsf{P}^{(i)},\vec{s}^{(i)}}|\cdot|\mathcal{R}_\mathsf{V}|}$ is a independent variable in range $[0:2]$ with expectation as assured by Claim 4.8.4. Therefore, following Hoeffding bound,

$$\begin{aligned}
\Pr\left[\frac{1}{g}\frac{1}{|\mathcal{R}_\mathsf{V}|}\sum_{i\in[g]}\frac{|\alpha^x_{\vec{s}^{(i)}}|\cdot|\zeta_{r_\mathsf{P}^{(i)}}|}{|\zeta_{r_\mathsf{P}^{(i)},\vec{s}^{(i)}}|}<\frac{1}{6}\right] &\le \Pr_{(r_\mathsf{P}^{(1)},\vec{s}^{(1)}),\ldots,(r_\mathsf{P}^{(g)},\vec{s}^{(g)})\leftarrow\mathsf{Sim}_\mathsf{P}}\left[\frac{1}{g}\frac{1}{|\mathcal{R}_\mathsf{V}|}\sum_{i\in[g]}\frac{|\alpha^x_{\vec{s}^{(i)}}|\cdot|\zeta_{r_\mathsf{P}^{(i)}}|}{|\zeta_{r_\mathsf{P}^{(i)},\vec{s}^{(i)}}|}<\frac{1}{6}\middle|E\right]+\Pr[\bar{E}]\\
&\le\frac{1}{2^{-\Omega(g)}}+7\cdot g\cdot\sqrt{\epsilon}\\
&\le\text{negl}(n).
\end{aligned}$$

- If $x \in \text{NO}(\Pi_n)$, for any $i \in [g]$, applying Markov inequality using the expectation upperbound ensured by Claim 4.8.4, we have

$$\Pr_{(r_\mathsf{P}^{(i)},\vec{s}^{(i)})\leftarrow\mathsf{Sim}_\mathsf{P}}\left[\frac{|\alpha^x_{\vec{s}^{(i)}}|\cdot|\zeta_{r_\mathsf{P}^{(i)}}|}{|\zeta_{r_\mathsf{P}^{(i)},\vec{s}^{(i)}}|\cdot|\mathcal{R}_\mathsf{V}|}>\sqrt{\delta+\epsilon}\middle|E\right]\le 8\sqrt{\delta+\epsilon}\le\text{negl}(n).$$

Apply an union bound over g samples, the probability that there exists $i \in [g]$ such that $\frac{|\alpha^x_{\vec{s}^{(i)}}| \cdot |\zeta_{r_{\mathsf{P}}^{(i)}}|}{|\zeta_{r_{\mathsf{P}}^{(i)},\vec{s}^{(i)}}| \cdot |\mathcal{R}_{\mathsf{V}}|} > \sqrt{\delta+\epsilon}$ is bounded by:

$$\Pr_{(r_{\mathsf{P}}^{(1)},\vec{s}^{(1)}),\ldots,(r_{\mathsf{P}}^{(g)},\vec{s}^{(g)}) \leftarrow \mathsf{Sim}_{\mathsf{P}}}\left[\exists i \in [g] : \frac{|\alpha^x_{\vec{s}^{(i)}}| \cdot |\zeta_{r_{\mathsf{P}}^{(i)}}|}{|\zeta_{r_{\mathsf{P}}^{(i)},\vec{s}^{(i)}}| \cdot |\mathcal{R}_{\mathsf{V}}|} > \sqrt{\delta+\epsilon}\right] \leq g \cdot \mathrm{negl}(n) + \Pr\left[\bar{E}\right] \leq \mathrm{negl}(n).$$

□

Consider the bounds k_{α_i} and $k_{r_{\mathsf{P}}^{(i)},\vec{s}^{(i)}}$ claimed by prover M in the g iterations.

Completeness. Given $x \in \textsc{Yes}(\Pi_n)$, the honest prover M set $k_{\alpha_i} = |\alpha^x_{\vec{s}^{(i)}}|$, $k_{r_{\mathsf{P}}^{(i)}} = |\zeta_{r_{\mathsf{P}}^{(i)}}|$, and $k_{r_{\mathsf{P}}^{(i)},\vec{s}^{(i)}} = 4 \cdot g \cdot |\zeta_{r_{\mathsf{P}},\vec{s}}|$. Taking an union bound over g samples, the probabiltity that verifier doesn't reject after g iterations is

$$\begin{aligned}
\Pr[\mathsf{A} \text{ doesn't reject after } g \text{ iterations}] &\geq \left(\Pr\left[\mathsf{LBP}^{\mathsf{O}_{\alpha^x_{\vec{s}^{(i)}}}}(k_{\alpha_i}), \mathsf{LBP}^{\mathsf{O}_{\zeta_{r_{\mathsf{P}}^{(i)}}}}(k_{r_{\mathsf{P}}^{(i)}}), \mathsf{UBP}^{\mathsf{O}_{\zeta_{r_{\mathsf{P}}(i),\vec{s}^{(i)}}}}(k_{r_{\mathsf{P}}^{(i)},\vec{s}^{(i)}}, r_i) \text{ all accept}\right]\right)^g \\
&\geq \left(1 - \mathrm{negl}(n)(1 - \mathrm{negl}(n)(1 - \frac{1}{4g})\right)^g \\
&\geq \frac{3}{4} - \mathrm{negl}(n).
\end{aligned}$$

Follows Claim 4.8.5 the probability that $\bar{k} < \frac{1}{10 \cdot g}$ is bounded by $\mathrm{negl}(n)$. And thus the probability that A accepts in the end is at least:

$$\begin{aligned}
\Pr\left[\langle \mathsf{M}, \mathsf{A}\rangle(x) = 1\right] &\geq 1 - \Pr\left[\text{verifier reject in } g \text{ iterations}\right] - \Pr\left[\bar{k} < \frac{1}{10 \cdot g}\right] \\
&\geq 1 - \frac{1}{4} - \mathrm{negl}(n) - \mathrm{negl}(n) \\
&> \frac{2}{3}.
\end{aligned}$$

Soundness. Consider any NO instance $x \in \textsc{No}(\Pi_n)$, A will accept only if at the end of g iterations, $\sum_{i \in [g]} \frac{k_{\alpha_i} \cdot k_{r_{\mathsf{P}}^{(i)}}}{k_{r_{\mathsf{P}}^{(i)},\vec{s}^{(i)}} \cdot |\mathcal{R}_{\mathsf{V}}|} \geq \frac{1}{10 \cdot g}$.

By the pigeonhole principle, it follows that

Claim 4.8.6. *If* $\sum_{i \in [g]} \frac{k_{\alpha_i} \cdot k_{r_{\mathsf{P}}^{(i)}}}{k_{r_{\mathsf{P}}^{(i)},\vec{s}^{(i)}} \cdot |\mathcal{R}_{\mathsf{V}}|} \geq \frac{1}{10 \cdot g}$, *there exists a* $j \in [g]$ *such that* $\frac{k_{\alpha_j} \cdot k_{r_{\mathsf{P}}^{(j)}}}{k_{r_{\mathsf{P}}^{(j)},\vec{s}^{(j)}} \cdot |\mathcal{R}_{\mathsf{V}}|} \geq \frac{1}{10 \cdot g^2}$.

Let T be the event that "$\forall i \in [g] : \frac{|\alpha^x_{\vec{s}^{(i)}}| \cdot |\zeta_{r_{\mathsf{P}}^{(i)}}|}{|\zeta_{r_{\mathsf{P}}^{(i)},\vec{s}^{(i)}}| \cdot |\mathcal{R}_{\mathsf{V}}|} \leq \sqrt{\delta+\epsilon}$". Define the event W_i being that A does not reject after i^{th} iterations of bound protocol, and let W be that A doesn't not reject after all g repetitions. Define H to be the event that $\sum_{i \in [g]} \frac{k_{\alpha_i} \cdot k_{r_{\mathsf{P}}^{(i)}}}{k_{r_{\mathsf{P}}^{(i)},\vec{s}^{(i)}} \cdot |\mathcal{R}_{\mathsf{V}}|} \geq \frac{1}{10 \cdot g}$. Denote by L_j the event that $k_{\alpha_j} \leq 2 \cdot |\alpha^x_{\vec{s}^{(i)}}|$ and $k_{r_{\mathsf{P}}^{(j)}} \leq 2 \cdot |\zeta_{r_{\mathsf{P}}^{(j)}}|$. We have:

$$\begin{aligned}
\Pr[\mathsf{A} \text{ accept}] &= \Pr[H \wedge W] &(1)\\
&\leq \Pr[H \wedge W | T] + \Pr\left[\bar{T}\right] &(2)\\
&\leq \Pr[W | H, T] + \Pr\left[\bar{T}\right]. &(3)
\end{aligned}$$

On the event H: $\sum_{i\in[g]} \frac{k_{\alpha_i}\cdot k_{r_{\mathsf{P}}^{(i)}}}{k_{r_{\mathsf{P}}^{(i)},\vec{s}^{(i)}}\cdot|\mathcal{R}_{\mathsf{V}}|} \geq \frac{1}{10\cdot g}$, let index j be such that $H_j : \frac{k_{\alpha_j}\cdot k_{r_{\mathsf{P}}^{(j)}}}{k_{r_{\mathsf{P}}^{(j)},\vec{s}_j}\cdot|\mathcal{R}_{\mathsf{V}}|} \geq \frac{1}{10\cdot g^2}$ by Claim 4.8.6.

$$\begin{aligned}
\Pr[W|H,T] &\leq \Pr[W_j|H,H_j,T] && (1)\\
&= \Pr\left[W_j \wedge L_j \middle| H,H_j,T\right] + \Pr\left[W_j \wedge \bar{L}_j \middle| H,H_j,T\right] && (2)\\
&\leq \Pr\left[W_j \wedge L_j \middle| H,H_j,T\right] + \Pr\left[W_j \middle| H,H_j,T,\bar{L}_j\right] && (3)\\
&\leq \Pr\left[W_j \middle| H,H_j,T,L_j\right] + \mathrm{negl}(n). && (4)
\end{aligned}$$

Line 1 follows Claim 4.8.6, Line 4 follows from conditional probability and Lemma 3.6.

Conditioned on H_j $\left(\frac{k_{\alpha_j}\cdot k_{r_{\mathsf{P}}^{(j)}}}{k_{r_{\mathsf{P}}^{(j)},\vec{s}_j}\cdot|\mathcal{R}_{\mathsf{V}}|} \geq \frac{1}{10\cdot g^2}\right)$, T $\left(\text{In particular, } \frac{|\alpha^x_{\vec{s}_j}|\cdot|\zeta_{r_{\mathsf{P}}^{(j)}}|}{|\zeta_{r_{\mathsf{P}}^{(j)},\vec{s}_j}|\cdot|\mathcal{R}_{\mathsf{V}}|} \leq \sqrt{\delta+\epsilon}\right)$, and L_j, following Lemma 4.6:

$$\begin{aligned}
\Pr\left[W_j \middle| H,H_j,T,L_j\right] &\leq \frac{6\cdot k_{r_{\mathsf{P}}^{(j)},\vec{s}_j}}{\left|\zeta_{r_{\mathsf{P}}^{(i)},\vec{s}^{(i)}}\right| - 1}\\
&\leq 80\cdot g^2\cdot\sqrt{\delta+\epsilon}\\
&\leq \mathrm{negl}(n).
\end{aligned}$$

To conclude, $\Pr\left[\langle \mathsf{M},\mathsf{V}\rangle(x)=1\right] \leq \mathrm{negl}(n)$, and thus Figure 2 is a constant-round interactive proof for Π.

Lemma 4.9 ([GS86])**.** *Given that $\langle \mathsf{P},\mathsf{V}\rangle$ is a $q(n)$-rounds uniform-verifier (resp. non-uniform V) interactive proof for a problem Π, then there exists a $(q(n)+2)$-rounds* public-coin *uniform-verifier (resp. non-uniform V') interactive proof $\langle \mathsf{P}',\mathsf{V}'\rangle$ for Π.*

Following the transformation assured by Lemma 4.9, we can get an AM protocol for Π.

Remark that the transformation in Lemma 2.8 does not maintain the efficiency of the simulator, and thus we cannot use that directly to build AM protocol for the complemented problem $\bar{\Pi}$. Consider any instance x of $\bar{\Pi}$ and let $\langle \mathsf{P},\mathsf{V}\rangle$ be the (δ,ϵ)-instance-hiding interactive proof for Π. We define $\bar{\alpha}^x_{\vec{s}} = \{r \in \beta^x_{\vec{s}} \mid \mathsf{V}(x,r,\vec{s})=0\}$, which is the subset of $\beta^x_{\vec{s}}$ with which V rejects in the end on instance x. It's clear that $\bar{\alpha}^x_{\vec{s}}, \alpha^x_{\vec{s}}$ are partitions of $\beta^x_{\vec{s}}$, and hence $(\frac{|\alpha^x_{\vec{s}}|}{|\beta^x_{\vec{s}}|} + \frac{|\bar{\alpha}^x_{\vec{s}}|}{|\beta^x_{\vec{s}}|}) = 1$ for any possible view $\vec{s}$. By linearity of expectation we have:

- If $x \in \textsc{Yes}(\bar{\Pi}_n) = \textsc{No}(\Pi_n)$: $\mathop{\mathbb{E}}_{\vec{s}\leftarrow\vec{S}}\left[\frac{|\bar{\alpha}^x_{\vec{s}}|}{|\beta^x_{\vec{s}}|}\right] = 1 - \mathop{\mathbb{E}}_{\vec{s}\leftarrow\vec{S}}\left[\frac{|\alpha^x_{\vec{s}}|}{|\beta^x_{\vec{s}}|}\right] \geq 1-\delta(n)$.
- If $x \in \textsc{No}(\bar{\Pi}_n) = \textsc{Yes}(\Pi_n)$: $\mathop{\mathbb{E}}_{\vec{s}\leftarrow\vec{S}}\left[\frac{|\bar{\alpha}^x_{\vec{s}}|}{|\beta^x_{\vec{s}}|}\right] = 1 - \mathop{\mathbb{E}}_{\vec{s}\leftarrow\vec{S}}\left[\frac{|\alpha^x_{\vec{s}}|}{|\beta^x_{\vec{s}}|}\right] \leq \delta(n)$.

Furthermore there exists an efficient membership oracle $\mathsf{O}_{\bar{\alpha}^x_{\vec{s}}}$ for $\bar{\alpha}^x_{\vec{s}}$, which essentially checks the consistency of $(r,\vec{s},x)$ like $\mathsf{O}_{\alpha^x_{\vec{s}}}$ but outputs 1 only if $\mathsf{V}(r,x,\vec{s})=0$ in the end. A constant-round interactive proof can be constructed as in Figure 2, but using membership oracle for set $\bar{\alpha}^x_{\vec{s}}$, the rest of the argument mirrors that of problem Π, thus completing the proof of the theorem.

Constant-round IP for Π: $\langle \mathsf{M}, \mathsf{A} \rangle(x)$

Parameters: Input length n, amplification parameter g.

Inputs: A promise instance x of Π_n.

Outputs: Verifier's decision $d \in \{0,1\}$

Ingredients:

- $\langle \mathsf{P}, \mathsf{V} \rangle$: An (δ, ϵ)-simulatable instance-hiding interactive proof for Π.
- For any transcript $\vec{s} \in \mathcal{S}$, we define the efficient membership oracle for set $\alpha^x_{\vec{s}}$ as follows:

> **Algorithm** $\mathsf{O}_{\alpha^x_{\vec{s}}}(r)$:
>
> 1. Interpret $\vec{s}$ as $(u_1, y_1, \ldots, u_q, y_q)$
> 2. Denote by $\vec{s}_i$ the prefix $(u_1, y_1, \ldots, u_i, y_i)$, with $\vec{s}_0$ being the empty string
> 3. **For** $j \leftarrow 1$ to q
>
> Check $\mathsf{V}(x, r, \vec{s}_{j-1}) = u_j$, if not output 0.
> 4. **If** $\mathsf{V}(x, r, \vec{s}) = 1$ output 1; **Else** output 0.

- For any $\vec{s} \in \mathcal{S}$ and $r_\mathsf{P} \in \mathcal{R}_\mathsf{P}$, define the efficient membership oracles for sets $\zeta_{r_\mathsf{P}, \vec{s}}$ and ζ_{r_P} as:

> **Algorithm** $\mathsf{O}_{\zeta_{r_\mathsf{P}, \vec{s}}}(r)$:
>
> 1. **If** $\mathsf{Sim}_\mathsf{P}(n; r) = (r_\mathsf{P}, \vec{s})$ output 1; **Else** output 0.
>
> **Algorithm** $\mathsf{O}_{\zeta_{r_\mathsf{P}}}(r)$:
>
> 1. $(r'_\mathsf{P}, \vec{s}) \leftarrow \mathsf{Sim}_\mathsf{P}(n; r)$.
> 2. **If** $r'_\mathsf{P} = r_\mathsf{P}$ output 1; **Else** output 0.

- $\mathsf{LBP}^{\mathsf{O}_S}(K)$: Set Lower-Bound protocol from Lemma 3.6 for the set S, with input K, and errors negligible in n.
- $\mathsf{UBP}^{\mathsf{O}_S}(K, r)$: Set Upper-Bound protocol from Lemma 4.6 for the set S, with public input K, and verifier's private input r.

Protocol:

For $i \leftarrow 1$ to g, in parallel:

1. A samples $r_i \leftarrow \mathcal{R}_{\mathsf{Sim}}$, computes $(r^{(i)}_\mathsf{P}, \vec{s}^{(i)}) \leftarrow \mathsf{Sim}_\mathsf{P}(n; r_i)$ and sends $(r^{(i)}_\mathsf{P}, \vec{s}^{(i)})$ to prover M.
2. Honest M sets proposed lower bound $k_{\alpha_i} = |\alpha^x_{\vec{s}^{(i)}}|$ (for size of $\alpha^x_{\vec{s}^{(i)}}$), and $k_{r^{(i)}_\mathsf{P}} = |\zeta_{r^{(i)}_\mathsf{P}}|$, and upper bound $k_{r^{(i)}_\mathsf{P}, \vec{s}^{(i)}} = 4 \cdot g \cdot |\zeta_{r_\mathsf{P}, \vec{s}}|$ (for size of $\zeta_{r^{(i)}_\mathsf{P}, \vec{s}^{(i)}}$), and sends k_{α_i}, $k_{r^{(i)}_\mathsf{P}}$, $k_{r^{(i)}_\mathsf{P}, \vec{s}^{(i)}}$ to verifier A.
3. A and M run lowerbound protocol on $\alpha^x_{\vec{s}^{(i)}}$: $d_{\alpha_i} \leftarrow \mathsf{LBP}^{\mathsf{O}_{\alpha^x_{\vec{s}^{(i)}}}}(k_{\alpha_i})$.
4. A and M run lowerbound protocol on ζ_{r_P}: $d_{r_\mathsf{P}, i} \leftarrow \mathsf{LBP}^{\mathsf{O}_{\zeta_{r^{(i)}_\mathsf{P}}}}(k_{r^{(i)}_\mathsf{P}})$.
5. A and M run upperbound protocol on $\zeta_{r^{(i)}_\mathsf{P}, \vec{s}^{(i)}}$ with private input r_i: $d_{\zeta_i} \leftarrow \mathsf{UBP}^{\mathsf{O}_{\zeta_{r_\mathsf{P}(i), \vec{s}^{(i)}}}}(k_{r^{(i)}_\mathsf{P}, \vec{s}^{(i)}}, r_i)$.
6. If $d_{\alpha_i} = 0$, or $d_{\zeta_i} = 0$, or $d_{r_\mathsf{P}, i} = 0$: A outputs 0 (Reject).

Let $t_\mathsf{V} = |\mathcal{R}_\mathsf{V}|$ be the size of verifier's random space. A computes $\bar{k} \leftarrow \frac{1}{g} \cdot \sum_{i \in [g]} \frac{k_{r^{(i)}_\mathsf{P}} \cdot k_{\alpha_i}}{t_\mathsf{V} \cdot k_{r^{(i)}_\mathsf{P}, \vec{s}^{(i)}}}$.

If $\bar{k} \geq \frac{1}{10 \cdot g}$, A outputs 1 (Accept); Otherwise Reject.

Figure 2: Constant-Round IP from simulatable (δ, ϵ)-IHIP

□

4.2 Constant-Round Non-Uniform Proofs from Instance-Hiding

Proof of Theorem 4.1. As observed earlier, the expectation $\mathop{\mathbb{E}}_{\vec{s}\leftarrow\vec{S}_x}\left[\frac{|\alpha^x_{\vec{s}}|}{|\beta^x_{\vec{s}}|}\right]$ distinguishes the promise problem. A natural approach is thus to reduce the task to proving the size of sets such as $\alpha^x_{\vec{s}}$. A major challenge, however, is the inefficiency of simulator coupled with the lack of an efficient method to sample $\vec{s}$. Fortunately, we observe that the simulator only requires the instance length as input, which allows us to relegate the inefficient simulation to the sampling of non-uniform advice. This methodology will be useful in subsequent sections.

Another challenge we will encounter is absence of protocol to prove the upperbound of $|\beta^x_{\vec{s}}|$. In response, we will provide the necessary information via non-uniform advice. We observe that for any two distinct instances, x and w in Π_n, the value of $|\beta^w_{\vec{s}}|$ will approximate $|\beta^x_{\vec{s}}|$ on average. This observation is formalized in the following claims:

Claim 4.9.1. *for any $x, w \in \textsc{Yes}(\Pi_n) \cup \textsc{No}(\Pi_n)$:*

$$\Delta(\vec{S}_x, \vec{S}_w) \leq \Delta\big(\vec{S}_x, \mathsf{Sim}_{\mathsf{P}}(n)\big) + \Delta\big(\vec{S}_w, \mathsf{Sim}_{\mathsf{P}}(n)\big) < 2 \cdot \epsilon(n),$$

where $\mathsf{Sim}_{\mathsf{P}}(n)$ is the distribution simulated by simulator $\mathsf{Sim}_{\mathsf{P}}$.

Proof of Claim 4.9.1. From the hiding property, we have a simulator $\mathsf{Sim}_{\mathsf{P}}$ such that for any instance $x \in \textsc{Yes}(\Pi_n) \cup \textsc{No}(\Pi_n) : \Delta\big(\vec{S}_x, \mathsf{Sim}_{\mathsf{P}}(n)\big) \leq \epsilon(n)$. Therefore by triangle inequality we have:

$$\begin{aligned}\Delta(\vec{S}_x, \vec{S}_w) &\leq \Delta\big(\vec{S}_x, \mathsf{Sim}_{\mathsf{P}}(n)\big) + \Delta\big(\vec{S}_w, \mathsf{Sim}_{\mathsf{P}}(n)\big) \\ &\leq 2 \cdot \epsilon(n).\end{aligned}$$

□

Claim 4.9.2. *For any two instances $x, w \in \textsc{Yes}(\Pi_n) \cup \textsc{No}(\Pi_n)$:*

$$\left|\mathop{\mathbb{E}}_{\vec{s}\leftarrow\vec{S}_x}\left[\frac{|\alpha^x_{\vec{s}}|}{|\beta^x_{\vec{s}}|}\right] - \mathop{\mathbb{E}}_{\vec{s}\leftarrow\vec{S}_w}\left[\frac{|\alpha^x_{\vec{s}}|}{|\beta^w_{\vec{s}}|}\right]\right| \leq 2 \cdot \epsilon(n).$$

Claim 4.9.3. *For any two instances $x, w \in \textsc{Yes}(\Pi_n) \cup \textsc{No}(\Pi_n)$:*

- *If $x \in \textsc{Yes}(\Pi_n)$:*
$$\mathop{\mathbb{E}}_{\vec{s}\leftarrow\vec{S}_w}\left[\frac{|\alpha^x_{\vec{s}}|}{|\beta^w_{\vec{s}}|}\right] \geq 1 - \delta(n) - 2 \cdot \epsilon(n).$$
- *If $x \in \textsc{No}(\Pi_n)$:*
$$\mathop{\mathbb{E}}_{\vec{s}\leftarrow\vec{S}_w}\left[\frac{|\alpha^x_{\vec{s}}|}{|\beta^w_{\vec{s}}|}\right] \leq \delta(n) + 2 \cdot \epsilon(n).$$

We remark that here $\vec{s}$ is sampled from $\vec{S}_w$. The denominator is the size of verifier's randomness consistent with w and $\vec{s}$, while the numerator is the size of $\alpha^x_{\vec{s}}$. Looking ahead, this would allow us to decide an instance x using view $\vec{s}$ sampled from the protocol public view of another instance $\vec{w}$.

Proof of Claim 4.9.3. From the completeness and soundness of $\langle \mathsf{P}, \mathsf{V}\rangle$ and Claim 4.9.1 we have:

- **Completeness**: If $x \in \textsc{Yes}(\Pi_n)$:
$$\begin{aligned}\mathop{\mathbb{E}}_{\vec{s}\leftarrow\vec{S}_x}\left[\frac{|\alpha^x_{\vec{s}}|}{|\beta^x_{\vec{s}}|}\right] &= \Pr\left[\langle \mathsf{P}(n), \mathsf{V}(x)\rangle = 1\right] &(1)\\ &\geq 1 - \delta(n). &(2)\end{aligned}$$

- **Soundness**: $x \in \text{NO}(\Pi_n)$:

$$\mathop{\mathbb{E}}_{\vec{s} \leftarrow \vec{S}_x} \left[\frac{|\alpha^x_{\vec{s}}|}{|\beta^x_{\vec{s}}|} \right] = \Pr\left[\langle \mathsf{P}(n), \mathsf{V}(x) \rangle = 1\right] \tag{1}$$

$$\leq \delta(n). \tag{2}$$

Combine with Claim 4.9.2, the claim follows. □

AM protocol for Π: $\langle \mathsf{M}, \mathsf{A} \rangle^{\mathsf{Q}}(x)$

Parameters: input length n, g

Inputs:
- An instance $x \in \text{YES}(\Pi_n) \cup \text{NO}(\Pi_n)$

Outputs: $d \in \{0, 1\}$

Ingredients:
- $\langle \mathsf{P}, \mathsf{V} \rangle$ is an (δ, ϵ)-Instance-Hiding Interactive Proof
- Advice: Takes advice $\vec{s}^{(1)}, \ldots, \vec{s}^{(g)}, b_1, \ldots, b_g$ from advide oracle Q
- $\mathsf{WSLBP}^{\mathsf{O}_{\vec{s}}}(\vec{c}, K)$: Weighted Sets Lowerbound Protocol from Lemma 3.7 for the sets $S_1, \ldots, S_g$, with input $\vec{c}$, and errors negligible in n.
- For $i \in [g]$ and transcript $\vec{s}^{(i)}$, we define the efficient membership oracle for set $\alpha^x_{\vec{s}^{(i)}}$ as follows:

  Algorithm $\mathsf{O}_{\alpha^x_{\vec{s}^{(i)}}}(r)$:
  1. $\vec{s}_{(i)} = (u_1, y_1, \ldots, u_q, y_q)$
  2. $s_0 \leftarrow \phi$ *(the empty string)*
  3. **For** $j \leftarrow 1$ to q
     Check $\mathsf{V}\Big(x, r, (u_1, y_1, \ldots, u_{j-1}, y_{j-1})\Big) = u_j$, if not output 0.
  4. **If** $\mathsf{V}(x, r, \vec{s}) = 1$ output 1; **Else** output 0.

Protocol:

1. M and A run the protocol $\mathsf{WSLBP}^{\mathsf{O}_{\alpha^x_{\vec{s}^{(1)}}}, \ldots, \mathsf{O}_{\alpha^x_{\vec{s}^{(g)}}}}$ with membership oracles access to $\mathsf{O}_{\alpha^x_{\vec{s}^{(1)}}}, \ldots, \mathsf{O}_{\alpha^x_{\vec{s}^{(g)}}}$ and inputs $(\frac{1}{b_1}, \ldots, \frac{1}{b_g})$, $(\frac{1}{2} \cdot g)$:

$$d \leftarrow \mathsf{WSLBP}^{\mathsf{O}_{\alpha^x_{\vec{s}^{(1)}}}, \ldots, \mathsf{O}_{\alpha^x_{\vec{s}^{(g)}}}} \left(\left(\frac{1}{b_1}, \ldots, \frac{1}{b_g} \right), \frac{1}{2} \cdot g \right).$$

2. A outputs d.

Figure 3: AM Protocol from (δ, ϵ)-IHIP

$\Pi \in \mathsf{NP/poly}$. Given the necessary tools above, we construct our AM protocol with non-uniform verifier as shown in Figure 3 where the advice oracle Q is defined in Figure 4.

Claim 4.9.4. *Consider the protocol in Figure 3 with the advice oracle in Figure 4. On input any instance x, it satisfies the following properties:*

- *If $x \in \textsc{Yes}(\Pi_n)$, with probability at least $1 - 2^{\Omega(g)}$,*

$$\sum_{i\in[g]} \frac{|\alpha^x_{\vec{s}^{(i)}}|}{b_i} \geq \frac{g}{2}.$$

- *If $x \in \textsc{No}(\Pi_n)$, with probability at least $1 - 2^{\Omega(g)}$,*

$$\sum_{i\in[g]} \frac{|\alpha^x_{\vec{s}^{(i)}}|}{b_i} \leq \frac{g}{8}.$$

Proof of Claim 4.9.4. Note that $b_i = |\beta^w_{\vec{s}^{(i)}}|$ and w may not be equal to x. Define $\mu = \mathbb{E}\left[\sum_{i\in[g]} \frac{|\alpha^x_{\vec{s}^{(i)}}|}{|\beta^w_{\vec{s}^{(i)}}|}\right] = g \cdot \mathbb{E}_{\vec{s}\leftarrow\vec{S}_w}\left[\frac{|\alpha^x_{\vec{s}}|}{|\beta^w_{\vec{s}}|}\right]$, and following Claim 4.9.3 we have:

- If $x \in \textsc{Yes}(\Pi_n)$:

$$\mu \geq g \cdot \big(1 - \delta(n) - 2\cdot\epsilon(n)\big).$$

- If $x \in \textsc{No}(\Pi_n)$:

$$\mu \leq g \cdot \big(\delta(n) + 2\cdot\epsilon(n)\big).$$

Given that $\delta + 2\cdot\epsilon \leq \frac{1}{16}$, and follows the Hoeffding bound :

- If $x \in \textsc{Yes}(\Pi_n)$: the probability that $\sum_{i\in[g]} \frac{|\alpha^x_{\vec{s}^{(i)}}|}{b_i} < \frac{g}{2}$ is:

$$\begin{aligned}
\Pr\left[\sum_{i\in[g]} \frac{|\alpha^x_{\vec{s}^{(i)}}|}{|\beta^w_{\vec{s}^{(i)}}|} < \frac{g}{2}\right] &\leq \Pr\left[\sum_{i\in[g]} \frac{|\alpha^x_{\vec{s}^{(i)}}|}{|\beta^w_{\vec{s}^{(i)}}|} < \big(1-\delta-2\epsilon\big)\cdot g - \frac{g}{4}\right] \\
&\leq \Pr\left[\sum_{i\in[g]} \frac{|\alpha^x_{\vec{s}^{(i)}}|}{|\beta^w_{\vec{s}^{(i)}}|} < \mu - \frac{g}{4}\right] \\
&\leq 2\cdot e^{\frac{-2\cdot(g/4)^2}{g\cdot(1-0)^2}} \\
&\leq 2\cdot e^{\frac{-g}{8}} \\
&\leq 2^{-\Omega(g)}.
\end{aligned}$$

- If $x \in \textsc{No}(\Pi_n)$: the probability that $\sum_{i\in[g]} \frac{|\alpha^x_{\vec{s}^{(i)}}|}{b_i} > \frac{g}{8}$ is:

$$\begin{aligned}
\Pr\left[\sum_{i\in[g]} \frac{|\alpha^x_{\vec{s}^{(i)}}|}{|\beta^w_{\vec{s}^{(i)}}|} > \frac{g}{8}\right] &\leq \Pr\left[\sum_{i\in[g]} \frac{|\alpha^x_{\vec{s}^{(i)}}|}{|\beta^w_{\vec{s}^{(i)}}|} > (\delta+2\epsilon)\cdot g + \frac{g}{16}\right] \\
&\leq \Pr\left[\sum_{i\in[g]} \frac{|\alpha^x_{\vec{s}^{(i)}}|}{|\beta^w_{\vec{s}^{(i)}}|} > \mu + \frac{g}{16}\right] \\
&\leq 2\cdot e^{\frac{-2\cdot(g/16)^2}{g\cdot(1-0)^2}} \\
&\leq 2^{-\Omega(g)}.
\end{aligned}$$

□

We define the following events:

- E_{Yes} : $\sum_{i\in[g]} \frac{|\alpha^x_{\vec{s}(i)}|}{b_i} \geq \frac{g}{2}$.
- E_{No} : $\sum_{i\in[g]} \frac{|\alpha^x_{\vec{s}(i)}|}{b_i} \leq \frac{g}{8}$.
- W_1 : The execution of WSLBP accepts.
- W_0 : The execution of WSLBP rejects.

Following Claim 4.9.4 and Lemma 3.7, we have the completeness and soundness:

- **Completeness:** If $x \in \textsc{Yes}(\Pi_n)$:

$$\begin{aligned}\Pr\left[\langle \mathsf{A}, \mathsf{M}\rangle(x) = 1\right] &\geq \Pr\left[E_{Yes} \wedge W_1\right] \\ &= \Pr\left[E_{Yes}\right] \cdot \Pr\left[W_1 | E_{Yes}\right] \\ &\geq (1 - 2^{-\Omega(g)}) \cdot (1 - g \cdot \mathrm{negl}(n)).\end{aligned}$$

- **Soundness:** If $x \in \textsc{No}(\Pi_n)$, for any prover M^*:

$$\begin{aligned}\Pr\left[\langle \mathsf{A}, \mathsf{M}^*\rangle(x) = 1\right] &\leq \Pr\left[E_{No} \wedge W_1\right] + \Pr\left[E_{Yes}\right] \\ &\leq \Pr\left[E_{No}\right] \cdot \Pr\left[W_1 | E_{No}\right] + \Pr\left[E_{Yes}\right] \\ &\leq (1 - 2^{-\Omega(g)}) \cdot \mathrm{negl}(n) + 2^{-\Omega(g)}.\end{aligned}$$

By setting g to some polynomial of n and following Corollary 3.5, the protocol shown in Figure 3 is a AM protocol for Π. Thus by Corollary 3.5, we conclude that $\Pi \in \mathsf{NP/poly}$.

For any Π that has (δ, ϵ)-IHIP with negligibly small δ, ϵ, it directly follows Lemma 2.8 that $\Pi \in \mathsf{coNP/poly}$. The transformation described in Lemma 2.8, however, introduces a polynomial loss in hiding due to the execution of protocol outlined in Figure 7. Here, we are proving a stronger assertion where $\delta, \epsilon < \frac{1}{32}$. Similarly to that in Theorem 4.3, we apply linearity of expectation to prove that the same advice-oracle AM protocol is complete and sound for the complement problem $\bar{\Pi}$.

$\Pi \in \mathsf{coNP/poly}$. Consider any instance $x \in \textsc{Yes}(\bar{\Pi}) \cup \textsc{No}(\bar{\Pi})$ and let $\langle \mathsf{P}, \mathsf{V}\rangle$ be the (δ, ϵ)-instance-hiding interactive proof for Π. We define $\bar{\alpha}^x_{\vec{s}} = \{r \in \beta^x_{\vec{s}} \mid \mathsf{V}(x, r, \vec{s}) = 0\}$ as in the proof of Theorem 4.3. Because both $\alpha^x_{\vec{s}}$ and $\bar{\alpha}^x_{\vec{s}}$ are subset of $\beta^x_{\vec{s}}$, by switching the names of the two sets and following Claim 4.9.2 we have that for any two instances $x, w \in \textsc{Yes}(\Pi_n) \cup \textsc{No}(\Pi_n)$:

$$\left| \mathop{\mathbb{E}}_{\vec{s}\leftarrow\vec{S}_x} \left[\frac{|\bar{\alpha}^x_{\vec{s}}|}{|\beta^x_{\vec{s}}|}\right] - \mathop{\mathbb{E}}_{\vec{s}\leftarrow\vec{S}_w} \left[\frac{|\bar{\alpha}^x_{\vec{s}}|}{|\beta^w_{\vec{s}}|}\right] \right| \leq 2\epsilon(n).$$

By triangle inequality, we get that:

- If $x \in \textsc{Yes}(\bar{\Pi}_n)$: $\mathop{\mathbb{E}}_{\vec{s}\leftarrow\vec{S}} \left[\frac{|\bar{\alpha}^x_{\vec{s}}|}{|\beta^w_{\vec{s}}|}\right] \geq 1 - \delta(n) - 2\epsilon(n)$.
- If $x \in \textsc{No}(\bar{\Pi}_n)$: $\mathop{\mathbb{E}}_{\vec{s}\leftarrow\vec{S}} \left[\frac{|\bar{\alpha}^x_{\vec{s}}|}{|\beta^w_{\vec{s}}|}\right] \leq \delta(n) + 2\epsilon(n)$.

Consequently, a constant-round public-coin interactive proof for $\bar{\Pi}$ can be constructed using the same technique of applying weighted lower bound protocol with identical advice oracle.

□

Advice oracle $\mathsf{Q}_w(\mathsf{P},\mathsf{V},n)$

Inputs:

- Input length n (Omitted when clear from context).
- $\langle \mathsf{P},\mathsf{V}\rangle$ is an (δ,ϵ)-IHIP protocol for Π.

Ingredients:

- w is some arbitrary instance of Π.

Procedure:

1. **For** $i \leftarrow 1$ to g

   Samples random seed for verifier: $r_v^i \leftarrow \mathcal{R}_{\mathsf{V}}$

   Samples random seed for prover: $r_p^i \leftarrow \mathcal{R}_{\mathsf{P}}$

   Simulate the view with the prover's and verifier's algorithm and randomness:

   $$\vec{s}^{(i)} \leftarrow \langle \mathsf{P}(r_p^i), \mathsf{V}(w, r_v^i)\rangle.$$

   Computes number of verifier's randomness consistent to the view $\vec{s}^{(i)}$:

   $$b_i \leftarrow |\beta^w_{\vec{s}^{(i)}}|.$$

2. Output $\vec{s}^{(1)},\ldots,\vec{s}^{(g)}, b_1,\ldots,b_g$.

Figure 4: Advice Oracle $\mathsf{Q}_w(\mathsf{P},\mathsf{V},n)$ for AM Protocol From IHIP

Proof of Claim 4.9.2. For any two instances $x, w \in \textsc{Yes}(\Pi_n) \cup \textsc{No}(\Pi_n)$, denote by $\mathcal{S}_x$ and $\mathcal{S}_w$ the support of $\vec{S}_x$ and $\vec{S}_w$ respectively and following Claim 4.9.1 we have:

$$\begin{aligned}
\mathop{\mathbb{E}}_{\vec{s}\leftarrow\vec{S}_w}\left[\frac{|\alpha^x_{\vec{s}}|}{|\beta^w_{\vec{s}}|}\right] &= \sum_{\vec{s}\in\mathcal{S}_w} \Pr\left[\vec{S}_w=\vec{s}\right]\cdot\frac{|\alpha^x_{\vec{s}}|}{|\beta^w_{\vec{s}}|} && (1)\\
&= \sum_{\vec{s}\in\mathcal{S}_w\cap\mathcal{S}_x} \Pr\left[\vec{S}_w=\vec{s}\right]\cdot\frac{|\alpha^x_{\vec{s}}|}{|\beta^w_{\vec{s}}|} + \sum_{\vec{s}\in\mathcal{S}_w\setminus\mathcal{S}_x} \Pr\left[\vec{S}_w=\vec{s}\right]\cdot\frac{|\alpha^x_{\vec{s}}|}{|\beta^w_{\vec{s}}|} && (2)\\
&= \sum_{\vec{s}\in\mathcal{S}_w\cap\mathcal{S}_x} \frac{|\beta^w_{\vec{s}}|}{|\mathcal{R}_{\mathsf{V}}|}\cdot\frac{|\gamma_{\vec{s}}|}{|\mathcal{R}_{\mathsf{P}}|}\cdot\frac{|\alpha^x_{\vec{s}}|}{|\beta^w_{\vec{s}}|} + \sum_{\vec{s}\in\mathcal{S}_w\setminus\mathcal{S}_x} \Pr\left[\vec{S}_w=\vec{s}\right]\cdot\frac{0}{|\beta^w_{\vec{s}}|} && (3)\\
&= \sum_{\vec{s}\in\mathcal{S}_w\cap\mathcal{S}_x} \frac{|\gamma_{\vec{s}}|\cdot|\alpha^x_{\vec{s}}|}{|\mathcal{R}_{\mathsf{P}}|\cdot|\mathcal{R}_{\mathsf{V}}|}, && (4)
\end{aligned}$$

where Line 3 is because $|\alpha^x_{\vec{s}}| \le |\beta^x_{\vec{s}}| = 0$ for $\vec{s}\in\mathcal{S}_w\setminus\mathcal{S}_x$.

$$\begin{aligned}
\mathop{\mathbb{E}}_{\vec{s}\leftarrow\vec{S}_x}\left[\frac{|\alpha^x_{\vec{s}}|}{|\beta^x_{\vec{s}}|}\right] &= \sum_{\vec{s}\in\mathcal{S}_x\cap\mathcal{S}_w} \Pr\left[\vec{S}_x=\vec{s}\right]\cdot\frac{|\alpha^x_{\vec{s}}|}{|\beta^x_{\vec{s}}|} + \sum_{\vec{s}\in\mathcal{S}_x\setminus\mathcal{S}_w} \Pr\left[\vec{S}_x=\vec{s}\right]\cdot\frac{|\alpha^x_{\vec{s}}|}{|\beta^x_{\vec{s}}|} && (1)\\
&= \sum_{\vec{s}\in\mathcal{S}_w\cap\mathcal{S}_x} \frac{|\beta^w_{\vec{s}}|}{|\mathcal{R}_{\mathsf{V}}|}\cdot\frac{|\gamma_{\vec{s}}|}{|\mathcal{R}_{\mathsf{P}}|}\cdot\frac{|\alpha^x_{\vec{s}}|}{|\beta^w_{\vec{s}}|} + \sum_{\vec{s}\in\mathcal{S}_x\setminus\mathcal{S}_w} \Pr\left[\vec{S}_x=\vec{s}\right]\cdot\frac{|\alpha^x_{\vec{s}}|}{|\beta^x_{\vec{s}}|} && (2)\\
&= \sum_{\vec{s}\in\mathcal{S}_w\cap\mathcal{S}_x} \frac{|\gamma_{\vec{s}}|\cdot|\alpha^x_{\vec{s}}|}{|\mathcal{R}_{\mathsf{P}}|\cdot|\mathcal{R}_{\mathsf{V}}|} + \sum_{\vec{s}\in\mathcal{S}_x\setminus\mathcal{S}_w} \Pr\left[\vec{S}_x=\vec{s}\right]\cdot\frac{|\alpha^x_{\vec{s}}|}{|\beta^x_{\vec{s}}|}. && (3)\\
& && (4)
\end{aligned}$$

Therefore we have:

$$\left| \mathop{\mathbb{E}}_{\vec{s}\leftarrow\vec{S}_x}\left[\frac{|\alpha^x_{\vec{s}}|}{|\beta^x_{\vec{s}}|}\right] - \mathop{\mathbb{E}}_{\vec{s}\leftarrow\vec{S}_w}\left[\frac{|\alpha^x_{\vec{s}}|}{|\beta^w_{\vec{s}}|}\right] \right| = \sum_{\vec{s}\in\mathcal{S}_x\setminus\mathcal{S}_w} \Pr\left[\vec{S}_x = \vec{s}\right] \cdot \frac{|\alpha^x_{\vec{s}}|}{|\beta^x_{\vec{s}}|} \tag{1}$$

$$\leq \sum_{\vec{s}\in\mathcal{S}_x\setminus\mathcal{S}_w} \Pr\left[\vec{S}_x = \vec{s}\right] \tag{2}$$

$$\leq \Delta(\vec{S}_x, \vec{S}_w) \tag{3}$$

$$\leq 2\cdot\epsilon(n). \tag{4}$$

□

Combining with Lemma 3.2, the corollary follows:

Corollary 4.10. *If* NP $\subseteq$ IHIP/poly*, the polynomial hierarchy collapses to the third level.*

5 Implications for One-Way Functions

As suggested in [Imp95], it is unclear whether hardness of problems in NP implies the existence of one-way function. This is important in the sense that if our world is in "Pessiland" or "Heuristica", where NP problems are hard either on average or on worst case but one-way functions do not exist, a huge set of cryptography primitives including pseudorandom generator [Yao82, BM82] and digital signatures [GMR88] would be impossible in a strong sense.

In this section, we provide positive implications by assuming the hard problem also possessing an instance-hiding interactive proof. We show two separate proofs – one for problems that only have an *constant-round* IHIP with *average-case hardness*, and another for those that have a simulatableIHIP with only *worst-case hardness*. The former proof is non-constructive – we can only prove that a one-way function exists given an average-hard problem has constant-round IHIP. Moreover, this construction of one-way functions uses potential adversaries in a non-blackbox manner. Similar techniques have found use in the context of collision-resistant hash functions [KY18, RV22].

The latter proof is constructive – we show a construction of one-way function from any worst-case hard problem that has a Simulatable-IHIP, where the simulator is efficient. Implications similar to this are known for other classes such as SRE [AR16] (and SZK [Ost91], though that needs average-case hardness), which similarly rely on the efficiency of the simulator (or the encoding function in case of SRE).

5.1 OWFs from Average-Case Hard Constant-Round IHIP

Definition 5.1 (Average-Case Hard Problems)**.** Consider a promise problem $\Pi = (\textsc{Yes}, \textsc{No})$, and an ensemble of efficiently sampleable distributions $X = \{X_n\}_{n\in\mathbb{N}}$, where X_n is supported on $(\textsc{Yes}_n \cup \textsc{No}_n)$. Π is said to be *hard on average* over X if for any (non-uniform, if specified) polynomial-time algorithm A, there is a negligible function negl such that for all large enough n,

$$\Pr_{x\leftarrow X_n}[\mathsf{A}(x) = \Pi(x)] \leq \frac{1}{2} + \mathrm{negl}(n).$$

Remark 5.2. In the above definition, we require that X_n be supported fully on $(\textsc{Yes}_n \cup \textsc{No}_n)$, and also be efficiently sampleable. Often in natural hard problems, these may not be simultaneously perfectly satisfied – a natural efficiently sampleable hard distribution might have a small probability of not satisfying the promise. While we do not explicitly state this in our theorems, our proofs are robust to this and continue to hold as long as the probability of not satisfying the promise is small enough, e.g. negligible.

Theorem 5.3. *If any problem that is hard on average against non-uniform polynomial-time algorithms possesses a constant-round (δ,ϵ)-instance-hiding interactive proof for some negligible functions δ and ϵ, then non-uniform infinitely-often one-way functions exist.*

For simplicity, let us first consider the case of 1-round IHIP.

Lemma 5.4. *If any problem that is hard on average against non-uniform polynomial-time algorithms possesses a* one-round (δ, ϵ)*-instance-hiding interactive proof for some negligible functions* δ *and* ϵ*, then there exists an explicit construction of a non-uniform infinitely often one-way function.*[7]

One of the building blocks for our proof is the notion of distributional one-way function defined in [IL89], which also proved that the existence of distribitionally one-way functions imply existence of one-way functions.

Definition 5.5 ([IL89])**.** Consider a family of efficiently computable functions $F = \left\{f_n : \{0,1\}^n \to \{0,1\}^{m(n)}\right\}_{n\in\mathbb{N}}$. For any $n \in \mathbb{N}$ and algorithm A, define the following two distributions: $D_{0,n}$ $\left(x \leftarrow \{0,1\}^n, \text{outputs } \left(x, f(x)\right)\right)$ and $D^A_{1,n}$ $\Big(x \leftarrow \{0,1\}^n, \text{outputs } \Big(A\big(f(x)\big), f(x)\Big)\Big)$. $F = \{f_n\}$ is said to be distributionally one-way if, there is a constant $c > 0$ such that, for any efficient algorithm A, for all large enough n,

$$\Delta(D_{0,n}, D^A_{1,n}) > n^{-c}.$$

Remark 5.6. While the above definition refers to the distribution of the input x being uniform over some $\{0,1\}^n$, we will treat it as coming from some sampleable distribution. That is, we will construct functions f_n where the above lower bound on statistical distance holds when x is sampled from some efficiently sampleable distribution X_n rather than $\{0,1\}^n$. This is sufficient because this function f_n composed with the efficient sampling algorithm for X_n then satisfies Definition 5.5.

Lemma 5.7 ([IL89])**.** *If there is a distributional one-way function then there is a one-way function. Further, there is an explicit transformation from any distributional one-way function to a one-way function. This transformation also works for non-uniform (distributional) one-way functions, and preserves uniformity.*

Note that the above implication is explicit – given a distributional one-way function family, one can explicitly construct a one-way function family. In what follows, we will borrow the notations of elements in Section 4.

Proof of Lemma 5.4. Consider any efficiently sampleable ensemble $X = \{X_n\}_{n\in\mathbb{N}}$ over Π. Let $\Pi = (\textsc{Yes}, \textsc{No})$ be a hard on-average problem with respect to efficiently sampleable distributions X_n over $\textsc{Yes}_n \cup \textsc{No}_n$. Suppose Π has a single-round (two-message) (δ, ϵ)-IHIP for some negligible δ, ϵ. Define function $F_1(x, r) = \Big(x, \mathsf{V}(x, r, \phi)\Big)$[8] and note that F_1 is efficiently computable [9]. Suppose, for contradiction, that F_1 is not distributionally one-way function when x is sampled from X_n and r is uniform over $\mathcal{R}_\mathsf{V}$. There exists a negligible function negl and an efficient algorithm A_1 that, given a randomly sampled image $F_1(x, r)$, outputs a distribution $\mathsf{A}_1\big(F_1(x, r)\big)$, which is negl$(n)$-close to uniform distribution over $F_1^{-1}\big(F_1(x, r)\big)$; that is,

$$\Delta\left(((x, r), F_1(x, r)), (\mathsf{A}_1(F_1(x, r)), F_1(x, r))\right) \le \text{negl}(n).$$

Without loss of generality, assume that A_1, on input (x, u_1), will never output an (x', r) such that $x' \neq x$. We will proceed to argue that a non-uniform efficient algorithm $\mathsf{B}_1^{\vec{s}}$ can be made using A_1 to decide Π with all but negligible probability. Specifically, $\mathsf{B}_1^{\vec{s}}$ is hard-coded with $\vec{s} = (u_1, y_1)$, and upon receiving input x, it runs $\mathsf{A}_1(x, u_1)$. If $\mathsf{A}_1(x, u_1)$ outputs $r \in \mathcal{R}_\mathsf{V}$, it returns the output of $\mathsf{V}(x, r, \vec{s})$; otherwise it outputs $\perp$.

Define advice oracle $\mathcal{O}$ based on inefficient simulator on honest prover Sim_P as in Section 3.1. This oracle $\mathcal{O}(n)$ runs $\mathsf{Sim}_\mathsf{P}(n)$ and provides the public view $\vec{s}$ that it outputs as non-uniform advice for B_1. The ϵ-hiding ensures that for any $x \in \textsc{Yes}_n \cup \textsc{No}_n$:

$$\Delta\Big(\mathcal{O}(n), \textsc{View}_{pub}\big(\mathsf{P}, \mathsf{V}(x)\big)\Big) \le \Delta\Big(\mathsf{Sim}_\mathsf{P}(n), \textsc{View}_\mathsf{P}\big(\mathsf{P}, \mathsf{V}(x)\big)\Big) \le \epsilon(n) \le \text{negl}(n).$$

[7] In fact, this one-way function is uniform if the verifier in the IHIP is uniform, while remaining secure against non-uniform adversaries.

[8] F_1 can be well-defined for all input lengths by truncating the input to the nearest $(n, \log|\mathcal{R}_\mathsf{V}|)$, where V is defined.

[9] In case X is not uniform distribution, F should instead take the randomness used by the sampler of X rather than x as input.

Let $\mathcal{O}'$ be an ideal advice oracle such that for any $x \in \text{YES}_n \cup \text{NO}_n$, it samples the exact public view distribution of the IHIP protocol for instance x: $\Delta\Big(\mathcal{O}'(x), \text{VIEW}_{pub}\big(\mathsf{P}, \mathsf{V}(x)\big)\Big) = 0$. We define the following four hybrid distributions, where $\vec{s} = (u_1, y_1)$:

- D_0 : Samples $x \leftarrow X_n, r \leftarrow \mathcal{R}_{\mathsf{V}}$, runs $\vec{s} \leftarrow \text{VIEW}_{pub}\big(\mathsf{P}, \mathsf{V}(x, r)\big)$, and outputs $(x, r, \vec{s})$.
- D_1 : Samples $x \leftarrow X_n, \vec{s} \leftarrow \mathcal{O}'(x)$ and samples $(x, r) \leftarrow F_1^{-1}(x, u_1)$, and outputs $(x, r, \vec{s})$.
- D_2 : Samples $x \leftarrow X_n, \vec{s} \leftarrow \mathcal{O}'(x)$ and runs $(x, r) \leftarrow \mathsf{A}_1(x, u_1)$, and outputs $(x, r, \vec{s})$.
- D_3 : Samples $x \leftarrow X_n, \vec{s} \leftarrow \mathcal{O}(n)$ and runs $(x, r) \leftarrow \mathsf{A}_1(x, u_1)$, and outputs $(x, r, \vec{s})$.

Claim 5.7.1. $\Delta(D_0, D_3) \leq \text{negl}(n)$

Proof of Claim 5.7.1. It's clear that $\Delta(D_0, D_1) = 0$, and $\Delta(D_1, D_2) \leq \text{negl}(n)$ follows from the property of A_1. Furthermore, we have:

$$\Delta(D_2, D_3) \leq \max_x \Big(\Delta\big(\mathcal{O}'(x), \mathcal{O}(n)\big)\Big) = \max_x \Big(\Delta\big(\text{VIEW}_{\mathsf{P}}\big(\mathsf{P}, \mathsf{V}(x)\big), \mathsf{Sim}_{\mathsf{P}}(n)\big)\Big) \leq \text{negl}(n),$$

where the first inequality follows data processing inequality and triangle inequality, and the second inequality follows from the instance-hiding property. □

Thus by the Markov inequality, there exists negligible function negl such that:

$$\Pr_{x \leftarrow X_n}\left[\Delta(D_0|x, D_3|x) \leq \text{negl}(n)\right] \geq 1 - \text{negl}(n),$$

where $D_0|x$ denotes the distribution of D_0 given instance x is fixed (Similarly for $D_3|x$). Fix the inverter A_1, define the set $\mathsf{A}_1\text{-}Good = \{x \in \text{YES}_n \cup \text{NO}_n |\ \Delta(D_0|x, D_3|x) \leq \text{negl}(n)\}$. For any $x \in \mathsf{A}_1\text{-}Good$, following the correctness of honest prover of the protocol, we have that:

$$\Pr_{\vec{s} \leftarrow \mathcal{O}, \mathsf{B}_1^{\vec{s}}}[\mathsf{B}_1^{\vec{s}}(x) = \Pi(x)] \geq \Pr[\langle \mathsf{P}, \mathsf{V}(x)\rangle = \Pi(x)] - \Delta(D_0|x, D_3|x) \geq 1 - \delta(n) - \text{negl}(n) = 1 - \text{negl}(n),$$

where the first inequality is from triangle inequality. Thus for a random instance x, conditioned on that $x \in \mathsf{A}_1\text{-}Good$:

$$\Pr_{\vec{s} \leftarrow \mathcal{O}, \mathsf{B}_1^{\vec{s}}, x}[\mathsf{B}_1^{\vec{s}}(x) = \Pi(x) | x \in \mathsf{A}_1\text{-}Good] \geq 1 - \text{negl}(n).$$

In conclusion:

$$\begin{aligned}
\Pr_{\vec{s} \leftarrow \mathcal{O}, \mathsf{B}_1^{\vec{s}}, x \leftarrow X_n}[\mathsf{B}_1^{\vec{s}}(x) = \Pi(x)] &\geq \Pr_{\vec{s} \leftarrow \mathcal{O}, \mathsf{B}_1^{\vec{s}}, x}[\mathsf{B}_1^{\vec{s}}(x) = \Pi(x) | x \in \mathsf{A}_1\text{-}Good] \cdot \Pr_{x \leftarrow X_n}[x \in \mathsf{A}_1\text{-}Good] \\
&\geq \big(1 - \text{negl}(n)\big) \cdot \big(1 - \text{negl}(n)\big) \\
&= 1 - \text{negl}(n).
\end{aligned}$$

Therefore, there must exists some view $\vec{s}$ such that $\Pr_{\mathsf{B}_1^{\vec{s}}, x \leftarrow X_n}[\mathsf{B}_1^{\vec{s}}(x) = \Pi(x)] \geq 1 - \text{negl}(n)$, which is a contradiction and so F_1 has to be a distributional one-way function, and the lemma follows. □

Proof of Theorem 5.3. While the underlying ideas bear resemblance to those in the proof of Lemma 5.4, the multi-round scenario, even with just two rounds, turns out to need other non-trivial ideas. We firstly prove the case of two rounds before applying underlying idea to prove any constant rounds recursively. Consider the existence of a hard on-average promise problem that allows for a two-round (four-message) (δ, ϵ)-instance-hiding interactive proof. Now, suppose (for the sake of contradiction) that distributional one-way functions do not exist. In attempting to apply the same strategy utilized in Lemma 5.4, our goal would be to devise an efficient algorithm C_2, that takes a random instance x and a random protocol view $\vec{s} \leftarrow \mathsf{Sim}_{\mathsf{P}}(n)$, and with

high probability over its input domain, outputs r'_{V} such that $(x, r'_{\mathsf{V}}, \vec{s})$ is statistically close to the real view of verifier V (when interacting with honest prover): $\textsc{View}_{\mathsf{V}}\big(\mathsf{P}, \mathsf{V}(x)\big)$. If we can construct such C_2, we can take a "good" public view as non-uniform advice, run C_2 to obtain consistent randomness of V, and decide the problem on most instances.

The proof differs from the that of Lemma 5.4 because the verifier's messages (u_1, u_2) may not be efficiently computable given x and r_{V}, even if verifier's algorithm is efficient. This is because the second message of the verifier, u_2, depends on y_1, which, despite solely dependent on u_1 and some randomness of prover, may not be efficiently computable. So we do not have a direct analogue of the function F_1 we used in the earlier proof.

To address the challenge, we note that the first message of verifier u_1 is in fact efficiently samplable.Recall that $\vec{s} = (u_1, y_1, u_2, y_2)$, and define:

- $\mathcal{R}_{x,u_1} = \{r \in \mathcal{R}_{\mathsf{V}} |\ \mathsf{V}(x, r, \phi) = u_1\}$.
- $\mathcal{R}_{x,\vec{s}} = \{r \in \mathcal{R}_{\mathsf{V}} |\ \mathsf{V}(x, r, \phi) = u_1 \wedge \mathsf{V}\big(x, r, (u_1, y_1)\big) = u_2\}$.

Note that $\mathcal{R}_{x,\vec{s}} \subseteq \mathcal{R}_{x,u_1} \subseteq \mathcal{R}_{\mathsf{V}}$. Consider the function $F_1(x, r) = \Big(x, \mathsf{V}(x, r, \phi)\Big)$. We will show that if F_1 is not a distributional one-way function, then we can either use this to construct another distributional one-way function F_2 or construct an efficient algorithm B_2 that decides Π on-average.

Specifically, suppose F_1 is not a distributional one-way function, then there exists an efficient inverter A_1 of F_1. Let C_1 be the algorithm that takes x, u_1 as input, runs $\mathsf{A}_1(x, u_1)$ to get (x, r), and outputs r. Given (x, u_1), C_1 serves to sample r from a distribution that is statistically close to uniform distribution over $\mathcal{R}_{x,u_1}$. A key observation is that the distribution of u_2 conditioned on (u_1, y_1) is efficiently (approximately) samplable given C_1 and u_1, y_1 because one can simply run the verifier on (x, r, u_1, y_1). Thus, we will apply C_1 to construct an efficient function F_2 that samples u_2 efficiently given (u_1, y_1).[10]

Particularly, let r_2 denote the randomness of C_1, and for each (u_1, y_1), define the following function

$$F_2^{u_1,y_1}(x, r_2) = \Big(x, \mathsf{V}\big(x, \mathsf{C}_1(x, u_1; r_2), (u_1, y_1)\big)\Big),$$

where C_1 takes r_2 as randomness [11] and u_1, y_1 are hard-coded in the function $F_2^{u_1,y_1}$.[12] For any (u_1, y_1), it is clear that $F_2^{u_1,y_1}$ is efficient. If any of these functions are distributionally one-way, we are done. If not, then for every (u_1, y_1), there exists an efficient algorithm $\mathsf{A}_2^{u_1,y_1}$ that samples close-uniform pre-image of $F_2^{u_1,y_1}$. This means following two distributions are statistically negligibly close for any (u_1, y_1):

- $H_0(u_1, y_1)$: Samples $x \leftarrow X_n, r_2 \leftarrow \mathcal{R}_{\mathsf{C}_1}, r_1 \leftarrow \mathsf{C}_1(x, u_1; r_2), u_2 \leftarrow \mathsf{V}\big(x, r_1, (u_1, y_1)\big)$, and outputs (r_2, x, u_1, y_1, u_2).
- $H_1(u_1, y_1)$: Samples $x \leftarrow X_n, r_2 \leftarrow \mathcal{R}_{\mathsf{C}_1}, r_1 \leftarrow \mathsf{C}_1(x, u_1; r_2), u_2 \leftarrow \mathsf{V}\big(x, r_1, (u_1, y_1)\big)$, computes $(x, r_2^*) \leftarrow \mathsf{A}_2^{u_1,y_1}\big(x, u_2\big)$, and outputs $(r_2^*, x, u_1, y_1, u_2)$.

By the data processing inequality, the following two distributions are also statistically close:

- $H'_0(u_1, y_1)$: Samples $(r_2, x, u_1, y_1, u_2) \leftarrow H_0(u_1, y_1)$, $r_1 \leftarrow \mathsf{C}_1(x, u_1; r_2)$, and outputs (r_1, x, u_1, y_1, u_2).
- $H'_1(u_1, y_1)$: Samples $(r_2^*, x, u_1, y_1, u_2) \leftarrow H_1(u_1, y_1)$, $r_1^* \leftarrow \mathsf{C}_1(x, u_1; r_2^*)$, and outputs $(r_1^*, x, u_1, y_1, u_2)$.

Let $\mathcal{O}'$ be an ideal advice oracle such that for any $x \in \textsc{Yes}_n \cup \textsc{No}_n$, it samples the exact public view distribution of the IHIP protocol for instance x: $\Delta\Big(\mathcal{O}'(x), \textsc{View}_{pub}\big(\mathsf{P}, \mathsf{V}(x)\big)\Big) = 0$. Define the following distributions: [13]

[10] In this proof, we usually denote by A_i the efficient inverter for efficient function F_i; and denote by C_i an efficient algorithm that samples an almost uniform distribution over $\mathcal{R}_{x,(u_1,y_1,\ldots,u_i)}$.

[11] The length of r_2 is non-explicit inherently from the non-explicitness of A_1.

[12] This is where we apply the adversary in a non-black-box way.

[13] $\vec{s} = (u_1, y_1, u_2, y_2)$ is a tuple of four elements here, when say $(u_1, y_1) \leftarrow \vec{s}$, we mean taking (u_1, y_1) from $\vec{s}$.

- D_0: Samples $x \leftarrow X_n$, $r_1 \leftarrow \mathcal{R}_{\mathsf{V}}$, $u_1 \leftarrow \mathsf{V}(x, r_1, \phi)$, $y_1 \leftarrow \mathsf{P}(u_1)$, $u_2 \leftarrow \mathsf{V}\big(x, r_1, (u_1, y_1)\big), y_2 \leftarrow \mathsf{P}(u_1, y_1, u_2)$, and outputs $(x, r_1, u_1, y_1, u_2, y_2)$.
- D_1: Samples $x \leftarrow X_n$, $\vec{s} \leftarrow \mathcal{O}'(x)$, $(u_1, y_1) \leftarrow \vec{s}$, $(x, r_1) \leftarrow F_1^{-1}(x, u_1)$, $u_2^* \leftarrow \mathsf{V}\big(x, r_1, (u_1, y_1)\big)$, $y_2 \leftarrow \mathsf{P}(u_1, y_1, u_2^*)$, and outputs $(x, r_1, u_1, y_1, u_2^*, y_2)$.
- D_2: Samples $x \leftarrow X_n$, $\vec{s} \leftarrow \mathcal{O}'(x)$, $(u_1, y_1) \leftarrow \vec{s}$, $r_2 \leftarrow \mathcal{R}_{\mathsf{C}_1}$, $r_1 \leftarrow \mathsf{C}_1(x, u_1, r_2)$, $u_2^* \leftarrow \mathsf{V}\big(x, r_1, (u_1, y_1)\big)$, $y_2 \leftarrow \mathsf{P}(u_1, y_1, u_2^*)$, and outputs $(x, r_1, u_1, y_1, u_2^*, y_2)$.
- D_3: Samples $x \leftarrow X_n$, $\vec{s} \leftarrow \mathcal{O}'(x)$, $(u_1, y_1) \leftarrow \vec{s}$, $r_2 \leftarrow \mathcal{R}_{\mathsf{C}_1}$, $r_1 \leftarrow \mathsf{C}_1(x, u_1, r_2)$, $(x, u_2^*) \leftarrow F_2^{u_1, y_1}(x, r_2)$, $y_2 \leftarrow \mathsf{P}(u_1, y_1, u_2^*)$, and outputs $(x, r_1, u_1, y_1, u_2^*, y_2)$.
- D_4: Samples $x \leftarrow X_n$, $\vec{s} \leftarrow \mathcal{O}'(x)$, $r_2 \leftarrow \mathcal{R}_{\mathsf{C}_1}$, $(x, u_2^*) \leftarrow F_2^{u_1, y_1}(x, r_2)$, $y_2 \leftarrow \mathsf{P}(u_1, y_1, u_2^*)$, $(x, r_2^*) \leftarrow \left(F_2^{u_1, y_1}\right)^{-1}(x, u_2^*)$, $r_1^* \leftarrow \mathsf{C}_1(x, u_1, r_2^*)$, outputs $(x, r_1^*, u_1, y_1, u_2^*, y_2)$

Main differences between the hybrid distributions are highlighted using different colors. D_1 alters D_0 by replacing u_2 in the original $\vec{s}$ with u_2^*, achieved by sampling a uniform preimage (x, r_1) from $F_1^{-1}(x, u_1)$ and then running V with r_1. D_2 modifies the sampling of (x, r_1) by employing the efficient algorithm C_1 instead of using uniform sampling. D_3 is a reconfiguration of D_2, driven by the function $F_2^{u_1, y_1}$. D_4 generates r_1^* by sampling a random preimage over $\left(F_2^{u_1, y_1}\right)^{-1}(x, u_2)$ to get r_2^*, and feeding $\mathcal{C}_1$ with (x, r_2^*), instead of outputs r_1 directly.

Claim 5.7.2. $\Delta(D_0, D_4) < \mathrm{negl}(n)$.

Proof of Claim 5.7.2. It is clear that $\Delta(D_0, D_1) = 0$ by definition of F_1 and $\Delta(D_1, D_2) < \mathrm{negl}(n)$ according to the definition of C_1. Furthermore, $\Delta(D_2, D_3) = 0$ holds because $F_2^{u_1, y_1} = \Big(x, \mathsf{V}\big(x, \mathsf{C}_1(x, u_1, r_2), (u_1, y_1)\big)\Big)$, and D_3 is essentially the same distribution as D_2, albeit with a modified description. By definition of $\left(F_2^{u_1, y_1}\right)^{-1}$, the r_2^* in D_4 follows the same distribution as r_2 in D_3 conditioned on (x, u_1, y_1, u_2^*). Following data processing inequality, $\Delta(D_3, D_4) = 0$.

□

Suppose $F_2^{u_1, y_1}$ is not a distributional one-way on length n, there exists an efficient inverter $\mathsf{A}_2^{u_1, y_1}$ for $F_2^{u_1, y_1}$ that, given a random image (x, u_2) of $F_2^{u_1, y_1}$, outputs an almost uniform distribution over preimage set $(F_2^{u_1, y_1})^{-1}(x, u_2)$. We will use $\mathsf{A}_2^{u_1, y_1}$ to construct an efficient algorithm that decides Π on-average with respect to X_n, which contradicts our assumptions. Specifically, let r_3 denote the randomness used by A_2, we define algorithm $\mathsf{C}_2^{u_1, y_1}(x, u_2, r_3)$ as follows:

$\mathsf{C}_2^{u_1, y_1}(x, u_2, r_3)$:

1. $(x, r_2) \leftarrow \mathsf{A}_2^{u_1, y_1}(x, u_2, r_3)$
2. $r_1 \leftarrow \mathsf{C}_1(x, u_1, r_2)$
3. outputs r_1.

We write $\mathsf{C}_2^{u_1, y_1}(x, u_2)$ as shorthand when the randomness is sampled uniformly. Define hybrid distributions:

- D_5 : Samples $x \leftarrow X_n$, $\vec{s} \leftarrow \mathcal{O}'(x)$, $r_2 \leftarrow \mathcal{R}_{\mathsf{C}_1}$, $r_1 \leftarrow \mathsf{C}_1(x, u_1, r_2)$, $(x, u_2^*) \leftarrow F_2^{u_1, y_1}(x, r_2)$, $y_2 \leftarrow \mathsf{P}(u_1, y_1, u_2^*)$, $(x, r_2^*) \leftarrow \mathsf{A}_2^{u_1, y_1}(x, u_2^*)$, $r_1^* \leftarrow \mathsf{C}_1(x, u_1, r_2^*)$, outputs $(x, r_1^*, u_1, y_1, u_2^*, y_2)$.
- D_6: Samples $x \leftarrow X_n$, $\vec{s} \leftarrow \mathcal{O}'(x)$, $(x, r_2) \leftarrow \mathsf{A}_2^{u_1, y_1}(x, u_2)$, $r_1^* \leftarrow \mathsf{C}_1(x, u_1, r_2)$, outputs $(x, r_1, \vec{s})$.
- D_7: Samples $x \leftarrow X_n$, $\vec{s} \leftarrow \mathcal{O}'(x)$, $r_1^* \leftarrow \mathsf{C}_2^{u_1, y_1}(x, u_2)$, outputs $(x, r_1^*, \vec{s})$.

- D_8: Samples $x \leftarrow X_n$, $\vec{s} \leftarrow \mathcal{O}(n)$, $r_1^* \leftarrow \mathsf{C}_2^{u_1,y_1}(x,u_2)$, outputs $(x, r_1^*, \vec{s})$.

Similarly, D_5 samples r_2^* using algorithm $\mathsf{A}_2^{u_1,y_1}(x,u_2^*)$ instead of sampling uniformly from $(F_2^{u_1,y_1})^{-1}(x,u_2^*)$. D_6 feeds $\mathsf{A}_2^{u_1,y_1}(x,u_2)$ with u_2 in $\vec{s}$ instead of the output of $F_2^{u_1,y_1}$ u_2^*. D_7 is a rearrangement of D_6, defined by the function $\mathsf{C}_2^{u_1,y_1}$. D_8 samples $\vec{s}$ from the advice oracle $\mathcal{O}(n)$, unlike $\mathcal{O}'(x) = \vec{S}(x)$, does not take the instance x as input.

Claim 5.7.3. $\Delta(D_0, D_8) < \text{negl}(n)$

Proof of Claim 5.7.3. $\Delta(D_4, D_5) \leq \text{negl}(n)$ follows the property of $\mathsf{A}_2^{u_1,y_1}$. Follows triangle inequality, we have that $\Delta(D_0, D_5) \leq \text{negl}$, which also indicates that the distribution of (u_1, y_1, u_2^*, y_2) in D_5 is negl-close to the distribution $\vec{s}$ in D_6. Thus $\Delta(D_5, D_6) \leq \text{negl}(n)$ follows data processing inequality. $\Delta(D_6, D_7) = 0$ according to definition of $\mathsf{A}_2^{u_1,y_1}$ and $\mathsf{C}_2^{u_1,y_1}$ respectively. Following data processing inequality, $\Delta(D_7, D_8) \leq \max_x \Big(\mathcal{O}'(x), \mathcal{O}(n)\Big) \leq \text{negl}(n)$. By triangle inequality, we have that $\Delta(D_0, D_8) \leq \text{negl}(n)$ in general. □

Similarly to Lemma 5.4, applying $\mathsf{C}_2^{u_1,y_1}$ as a sub-routine, construct $B_2^{\vec{s}}$:

$B_2^{\vec{s}}(x)$:

1. $(u_1, y_1), u_2 \leftarrow \vec{s}$.
2. $r \leftarrow \mathsf{C}_2^{u_1,y_1}(x, u_2)$.
3. outputs $\mathsf{V}(x, r, \vec{s})$.

The probability of B_2 decides Π is:

$$\begin{aligned}
\Pr_{\vec{s}\leftarrow\mathcal{O}(n), x\leftarrow X_n, \mathsf{B}_2^{\vec{s}}}\left[\mathsf{B}_2^{\vec{s}}(x) = \Pi(x)\right] &= \Pr_{(x,r,\vec{s})\leftarrow D_7}[\mathsf{V}(x,r,\vec{s}) = \Pi(x)], \\
&\geq \Pr_{(x,r,\vec{s})\leftarrow D_0}[\mathsf{V}(x,r,\vec{s}) = \Pi(x)] - \Delta(D_0, D_7), \\
&= \Pr[\langle \mathsf{P}, \mathsf{V}(x)\rangle = \Pi(x)] - \Delta(D_0, D_7), \\
&\geq 1 - \text{negl}(n),
\end{aligned}$$

where the first inequality follows triangle inequality and data processing inequality. Thus there exists a "good" advice $\vec{s}$ such that $\Pr_{x\leftarrow X_n, \mathsf{B}_2^{\vec{s}}}\left[\mathsf{B}_2^{\vec{s}}(x) = \Pi(x)\right] \geq 1 - \text{negl}(n)$. Fix $\vec{s}$ and non-uniform C^{u_1,y_1}, $\mathsf{B}_2^{\vec{s}}$ is a non-uniform efficient machine, which contradicts the assumption that Π is hard on-average with respect to X.

Now we extend the proof of implication to any constant number of rounds of IHIP. Consider a problem Π that is hard on average against non-uniform adversaries with respect to ensemble $X = \{X_n\}_{n\in\mathbb{N}}$ for large enough n and it has a q rounds (δ, ϵ)-IHIP $\langle \mathsf{P}, \mathsf{V}\rangle$. For any possible public view of the protocol $\vec{s} = (u_1, y_1, \ldots, u_q, y_q)$ and instance x, for $i \in [q(n)]$, define:

$$\mathcal{R}_{x,(u_1,y_1,\ldots,u_i)} = \{r \in \mathcal{R}_{\mathsf{V}} |\ \forall j \in [i] : \mathsf{V}\big(x, r, (u_1, y_1, \ldots, u_{j-1}, y_{j-1})\big) = u_j\}.$$

The conditional probability mass ensures that:

$$\mathcal{R}_{x,\vec{s}} \subseteq \mathcal{R}_{x,(u_1,y_1,\ldots,u_{q-1})} \subseteq \cdots \subseteq \mathcal{R}_{x,(u_1,y_1,u_2)} \subseteq \mathcal{R}_{x,u_1} \subseteq \mathcal{R}_{\mathsf{V}}.$$

The theorem follows a recursive argument. Let F_1, C_1 and A_1 be as defined before. From $i = 2$ to q, given that $\mathsf{C}_{i-1}^{u_1,y_1,\ldots,u_{i-2},y_{i-2}}(x, u_{i-1})$ takes randomness of length $|r_i(n)|$ and efficiently samples an almost uniform

distribution over $\mathcal{R}_{x,u_1,y_1,\ldots,u_{i-1}}$ for all large enough input length n_{i-1}. For $n_i = n + |r_i(n)|$, we define:[14]

$$F_i^{u_1,y_1,\ldots,u_{i-1},y_{i-1}}(x,r_i) = \Big(x, V\big(x, C_{i-1}^{u_1,y_1,\ldots,u_{i-2},y_{i-2}}(x,u_{i-1},r_i),(u_1,y_1,\ldots,u_{i-1},y_{i-1})\big)\Big).$$

We note that $F_i^{u_1,y_1,\ldots,u_{i-1},y_{i-1}}$ is efficient for all large enough n_i, which is infinite often. Assume $F_i^{u_1,y_1,\ldots,u_{i-1},y_{i-1}}$ is not an infinitely often distributional one-way function, there exists an efficient inverter $A_i^{u_1,y_1,\ldots,u_{i-1},y_{i-1}}$ that, for all large enough $n = |x|$, on input (x,u_{i-1}), samples (x,r_i) statistically close to uniform distribution over $\big(F_i^{u_1,y_1,\ldots,u_{i-1},y_{i-1}}\big)^{-1}(x,u_i)$. With $A_i^{u_1,y_1,\ldots,u_{i-1},y_{i-1}}$, we define $\mathsf{C}_i^{u_1,y_1,\ldots,u_{i-1},y_{i-1}}$ for the next round of the argument:

$\mathsf{C}_i^{u_1,y_1,\ldots,u_{i-1},y_{i-1}}(x,u_i,r_{i+1})$:

1. $(x,r_i) \leftarrow A_i^{u_1,y_1,\ldots,u_{i-1},y_{i-1}}(x,u_i,r_{i+1})$.
2. $r_1 \leftarrow C_{i-1}^{u_1,y_1,\ldots,u_{i-2},y_{i-2}}(x,u_{i-1},r_i)$.
3. outputs r_1.

Suppose none of F_i is distributional one-way function and the arguments continue until q^{th} round, where we have $\mathsf{C}_q^{\vec{s}}$ that, for all large enough $n = |x|$, given $\vec{s}$ and on input x, samples a distribution statistically close to the uniform distribution over $\mathcal{R}_{x,\vec{s}}$. Then we construct the decision algorithm B_q on all large enough $n = |x|$:

$B_q^{\vec{s}}(x)$:

1. $(u_1,y_1,u_{q-1},y_{q-1}), u_q \leftarrow \vec{s}$.
2. $r \leftarrow \mathsf{C}_q^{u_1,y_1,\ldots,y_{q-1}}(x,u_q)$.
3. outputs $\mathsf{V}(x,r,\vec{s})$.

With each iteration of the recursive transformation, the size of functions increase at most polynomially. Consequently, if q is a constant, the circuit size of $B_q^{\vec{s}}$ remains polynomial in the original input length. The rest of the arguments mirror those in the two-round case and the theorem follows. □

5.2 Explicit OWFs from Worst-Case Hard Simulatable IHIP

In addition to the positive result regarding OWF, this section also providing insights for comparing IHIP/Simulatable-IHIP with classes SZK and SRE by examining the consequences of the existence of hard problems within these classes. For instance, Ostrovsky [Ost91] proves an implication of *average-case hard* SZK of OWFs, while Applebaum and Raykov [AR16] demonstrate a similar implication of *worst-case hard* SRE. In this subsection, we present an implication of *worst-case* hard Simulatable-IHIP on OWF.

Definition 5.8 (Worst-Case Hard Problems)**.** A promise problem $\Pi = (\textsc{Yes}, \textsc{No})$ is said to be *worst-case hard* if for any (non-uniform, if specified) polynomial-time algorithm A, there is a negligible function negl, such that for all large enough n, there exists an instance $x \in \textsc{Yes}_n \cup \textsc{No}_n$,

$$\Pr_{\mathsf{A}}[\mathsf{A}(x) = \Pi(x)] \leq \frac{1}{2} + \mathrm{negl}(n).$$

[14]The function F_i for all input length $n_i \in \mathbb{N}$ is defined by truncating the input to the nearest $(n, |r_i(n)|)$, on which C_{i-1} is well defined.

Theorem 5.9. *If any worst-case hard problem has a honest-prover* simulatable (δ, ϵ)*-instance-hiding interactive proof for some negligible functions δ and ϵ, then there is an explicit construction of one-way functions. This one-way function is uniform if the simulator of the IHIP is uniform, and it is secure against non-uniform adversaries if the problem is worst-case hard against non-uniform algorithms.*

Proof of Theorem 5.9. Consider any promise problem Π that has $q(n)$-round honest-prover simulatable (δ, ϵ)-instance-hiding interactive proof $\langle \mathsf{P}, \mathsf{V} \rangle$ for some negligible δ, ϵ and polynomial $q(n)$ with efficient simulator Sim_P. Different from Theorem 5.3, here the transcript distribution can be efficiently sampled. Specifically, there exists an efficient simulator Sim_P that on the input n and randomness $r_S \in \mathcal{R}_S$ outputs a possible public view of the protocol: $\vec{s} = (u_1, y_1, \ldots, u_q, y_q)$. It also simulates the randomness of the prover, but we do not need this here, and throughout simply ignore that part of the simulator's output. We define Sim_i as the algorithm that outputs the public transcripts of Sim_P up to virtual verifier's i^{th} message, that is $(u_1, y_1, \ldots, u_i)$. Define function $f(\mathbf{r}) = f(r_1, \ldots, r_q) = (\mathsf{Sim}_1(r_1), \ldots, \mathsf{Sim}_q(r_q))$[15]. We will proceed to prove that if Π is hard in worst case, then there f is distributionally one-way.

Suppose, for the sake of contradiction, f is not distributionally one-way, then none of Sim_is are distributional one-way . This is because an efficient algorithm that inverts the concatenated function $\mathsf{Sim}_1 || \ldots || \mathsf{Sim}_{q(n)}$ with neglibibly-bias from uniformity can be used to invert any individual Sim_i uniformly with negligible bias[16]. Then, for each Sim_i, there exists a probabilitic polynomial-time algorithm A_i which takes output of Sim_i, represented as $(u_1, y_1, \ldots, u_i)$, samples a randomness $r_{\mathsf{Sim},i}$ negl-close to uniform distribution over $\mathsf{Sim}_i^{-1}(u_1, y_2, \ldots, u_i)$. We construct a efficient algorithm B that utilizes $\mathsf{A}_1, \ldots, \mathsf{A}_q, \mathsf{Sim}_\mathsf{P}$ as sub-rountines to get $y_1, \ldots, y_q$ and computes Π most of the time. Formally, we define B as follows:

$\mathsf{B}(x)$:

1. $r_\mathsf{V} \leftarrow \mathcal{R}_\mathsf{V}$.
2. $u_1 \leftarrow \mathsf{V}(x, r_\mathsf{V}, \phi)$.

   For $i \leftarrow 1$ to q

   (a) $(n, r_{\mathsf{Sim},i}) \leftarrow \mathsf{A}_i(u_1, y_1, \ldots, u_i)$.

   (b) $(\ldots, u_i, y_i, \ldots) \leftarrow \mathsf{Sim}_\mathsf{P}(n; r_{\mathsf{Sim},i})$.

   (c) $u_{i+1} \leftarrow \mathsf{V}(x, r_\mathsf{V}, (u_1, \ldots, y_i))$.

Outputs $\mathsf{V}\big(x, r_\mathsf{V}, (u_1, y_1, \ldots, u_q, y_q)\big)$.

Figure 5: Algorithm from Simulatable-IHIP to Decide Π

Intuitively, in the proof of Theorem 5.3, we use simulator to generates the entire public view, or $\text{VIEW}_{pub}\big(\mathsf{P}_S, \mathsf{V}_S\big)$, and utilize the inverting adversaries to obtain the "simulated verifier randomness", which is proven to have correctness. Since the simulator Sim_P is efficient here, we have better control over the simulated protocol, which can be executed during the actual execution of algorithm.

Specifically, the algorithm B above uses Sim_P to simulate prover and runs the real verifier to interacts with the simulated prover, ultimately outputting the result. Under the assumption that $\mathsf{Sim}_1, \ldots, \mathsf{Sim}_q$ are not distributional one-way funtion, B employs the inverters to "simulate" the "simulated prover" efficiently. We will proceed to prove that this prover ensures correctness. Formally, we define the provers as follows:

- Real prover $\mathsf{P}(\vec{s}_{i-1}, u_i)$: On input $(\vec{s}_{i-1}, u_i)$, it generates response, $y_i \leftarrow \mathsf{P}((\vec{s}_{i-1}, u_i))$, and outputs y_i.

[15]While $\mathbf{r}$ is defined only for infinitely many input lengths, the function f is well-defined for all input lengths by truncating the input to nearest length where $\mathbf{r}$ is defined.

[16]This is achieved by feeding the algorithm the target image for Sim_i, padded with independently simulated outputs from the other $q(n) - 1$ distributions.

- Simulated prover $\mathsf{P}_S\left(\vec{s}_{i-1}, u_i\right)$: On input $(\vec{s}_{i-1}, u_i)$, sample the next message over the simulator's conditional probability mass $(n, r_{\mathsf{Sim},i}) \leftarrow \mathsf{Sim}_i^{-1}(\vec{s}_{i-1}, u_i)$; $(\dots, u_i, y_i, \dots) \leftarrow \mathsf{Sim}_{\mathsf{P}}(n; r_{\mathsf{Sim},i})$, and outputs y_i.
- Efficiently simulated prover $\mathsf{P}_{SS}\left(\vec{s}_{i-1}, u_i\right)$: On input $(\vec{s}_{i-1}, u_i)$, use inverter A_i to sample the next message over simulator's conditional probability mass efficiently $(n, r_{\mathsf{Sim},i}) \leftarrow \mathsf{A}_i(\vec{s}_{i-1}, u_i)$; $(\dots, u_i, y_i, \dots) \leftarrow \mathsf{Sim}_{\mathsf{P}}(n; r_{\mathsf{Sim},i})$, and outputs y_i.

The B in Figure 5 is essentially an execution of $\langle \mathsf{P}_{SS}, \mathsf{V} \rangle$. We will proceed to prove that

$$\textsc{View}_{\mathsf{V}}\big(\mathsf{P}_{SS}, \mathsf{V}(x)\big) \approx \textsc{View}_{\mathsf{V}}\big(\mathsf{P}_S, \mathsf{V}(x)\big) \approx \textsc{View}_{\mathsf{V}}\big(\mathsf{P}, \mathsf{V}(x)\big),$$

and the correctness of B will follow.

To begin, we will firstly prove that $\textsc{View}_{\mathsf{V}}\big(\mathsf{P}_S, \mathsf{V}(x)\big) \approx \textsc{View}_{\mathsf{V}}\big(\mathsf{P}, \mathsf{V}(x)\big)$. Define a sequence of hybrid algorithm W_j for $j \in [0, q(n)]$ as follows:

$\mathsf{W}_j(x)$:

1. $r_{\mathsf{V}} \leftarrow \mathcal{R}_{\mathsf{V}}$.
2. $r_{\mathsf{P}} \leftarrow \mathcal{R}_{\mathsf{P}}$.
3. $u_1 \leftarrow \mathsf{V}(x, r_{\mathsf{V}}, \phi)$.

   For $i \leftarrow 1$ to j

   (a) $(\dots, u_i, y_i, \dots) \leftarrow \mathsf{P}(u_1, y_1, \dots, u_i)$.
   (b) $u_{i+1} \leftarrow \mathsf{V}\left(x, r_{\mathsf{V}}, (u_1, \dots, y_i)\right)$.

   For $i \leftarrow j+1$ to q

   (a) $(n, r_{\mathsf{Sim},i}) \leftarrow \mathsf{Sim}_i^{-1}(u_1, y_2, \dots, u_i)$.
   (b) $(\dots, u_i, y_i, \dots) \leftarrow \mathsf{Sim}_{\mathsf{P}}(n; r_{\mathsf{Sim},i})$.
   (c) $u_{i+1} \leftarrow \mathsf{V}\left(x, r_{\mathsf{V}}, (u_1, \dots, y_i)\right)$.

Outputs $(x, r_{\mathsf{V}}, u_1, y_1, \dots, u_q, y_q)$.

Claim 5.9.1. *For any $x \in \textsc{Yes}_n \cup \textsc{No}_n$, $j \in [q(n)]$:*

$$\Delta\left(\mathsf{W}_j(x), \mathsf{W}_{j-1}(x)\right) \leq \epsilon.$$

Proof of Claim 5.9.1. In the rest, let $\mathsf{W}_j(x)[2i]$ denote the state of W_j's first public $2i$ messages, that is $(u_1, y_1, \dots, u_i, y_i)$ (Similarly for $\textsc{View}_{pub}\big(\mathsf{P}, \mathsf{V}(x)\big)[2i]$). We noted that because we are using real verifier V in all cases, we would like to ignore the verifier's private randomness when analyzing the distances between hybrid algorithms, this is because the distribution of $(x, r_{\mathsf{V}}, u_1, y_1, \dots, u_q, y_q)$ identical to one obtained by firstly sampling $(x, u_1, y_1, \dots, u_q, y_q)$ and then sampling r_{V} uniformly among the randomness consistent with $(x, u_1, y_1, \dots, u_q, y_q)$, or $\beta^x_{u_1,\dots,y_1}$ as in Section 4. Thus it's sufficed to focus on $(x, u_1, y_1, \dots, y_q)$ by data processing inequality. According to ϵ-hiding of $\langle \mathsf{P}, \mathsf{V} \rangle$, we have that for any $j \in [q(n)]$

$$\Delta\Big(\mathsf{W}_j(x)[2j], \mathsf{Sim}_j(n)\Big) \leq \Delta\left(\textsc{View}_{pub}\big(\mathsf{P}, \mathsf{V}(x)\big), \textsc{View}_{pub}\big(\mathsf{P}_S, \mathsf{V}(x)\big)\right) \leq \epsilon.$$

For $j \in [q(n)]$, and any possible $(\vec{s}_{j-1}, u_j)$,

$$\Delta\left(\mathsf{P}\left(\vec{s}_{j-1},u_j\right),\mathsf{P}_S\left(\vec{s}_{j-1},u_j\right)\right)=\Delta\left(\mathsf{W}_j(x)[2j]\middle|\mathsf{W}_j(x)[2j-1]=(\vec{s}_{j-1},u_j),\mathsf{Sim}_j(n)\middle|\mathsf{Sim}_j(n)=(\vec{s}_{j-1},u_j)\right) \tag{5}$$

$$\leq \Delta\left(\mathsf{W}_j(x)[2j],\mathsf{Sim}_j(n)\right) \tag{6}$$

$$\leq \epsilon, \tag{7}$$

where the first line of equation is by definition of P and P_S, the second line follows the definition of total variance distance.

Fact 5.10. *Let (X_1,Y_1) and (X_2,Y_2) be two joint distributions over space $\mathcal{X}\times\mathcal{Y}$. Suppose it holds that $\Delta(X_1,X_2)\leq\epsilon_1$ and for any $x\in\mathcal{X}$, $\Delta(Y_1|X_1=x,Y_2|X_2=x)\leq\epsilon_2$, then $\Delta\big((X_1,Y_1),(X_2,Y_2)\big)\leq\epsilon_1+\epsilon_2$.*

Proof.

$$\begin{aligned}
\Delta\left((X_1,Y_1),(X_2,Y_2)\right) &= \frac{1}{2}\sum_{(x,y)\in\mathcal{X}\times\mathcal{Y}}\Big|\Pr\left[X_1=x,Y_1=y\right]-\Pr\left[X_2=x,Y_2=y\right]\Big| \\
&= \frac{1}{2}\sum_{(x,y)\in\mathcal{X}\times\mathcal{Y}}\Big|\Pr\left[X_1=x\right]\cdot\Pr\left[Y_1=y|X_1=x\right]-\Pr\left[X_2=x\right]\cdot\Pr\left[Y_2=y|X_2=x\right]\Big| \\
&\leq \frac{1}{2}\sum_{(x,y)\in\mathcal{X}\times\mathcal{Y}}\Big|\Pr\left[X_1=x\right]\cdot\Pr\left[Y_1=y|X_1=x\right]-\Pr\left[X_2=x\right]\cdot\Pr\left[Y_1=y|X_1=x\right]\Big| \\
&\quad+\Big|\Pr\left[X_2=x\right]\cdot\Pr\left[Y_1=y|X_1=x\right]-\Pr\left[X_2=x\right]\cdot\Pr\left[Y_2=y|X_2=x\right]\Big| \\
&= \frac{1}{2}\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}}\Pr\left[Y_1=y|X_1=x\right]\cdot\Big|\Pr\left[X_1=x\right]-\Pr\left[X_2=x\right]\Big| \\
&\quad+\sum_{x\in\mathcal{X}}\Pr\left[X_2=x\right]\cdot\left(\frac{1}{2}\cdot\sum_{y\in\mathcal{Y}}\Big|\Pr\left[Y_1=y|X_1=x\right]-\Pr\left[Y_2=y|X_2=x\right]\Big|\right) \\
&= \frac{1}{2}\sum_{x\in\mathcal{X}}\Big|\Pr\left[X_1=x\right]-\Pr\left[X_2=x\right]\Big|+\sum_{x\in\mathcal{X}}\Pr\left[X_2=x\right]\cdot\epsilon_2 \\
&\leq \epsilon_1+\epsilon_2.
\end{aligned}$$

where the first inequality follows triangle inequality. □

Denote by $D_{j-1}=\textsc{View}_{pub}\big(\mathsf{P},\mathsf{V}(x)\big)[2j-1]$ for brevity, we have:

$$\begin{aligned}
\Delta\left(\mathsf{W}_j(x),\mathsf{W}_{j-1}(x)\right) &= \Delta\left(\left(D_{j-1},\mathsf{P}\left(D_{j-1}\right)\right),\left(D_{j-1},\mathsf{P}_S\left(D_{j-1}\right)\right)\right) \\
&\leq \epsilon,
\end{aligned}$$

where the last inequality follows Fact 5.10 and Eq. 7. Thus, the claim follows.

□

Apply Fact 4.8 and triangle inequality, we have the following claim:

Claim 5.10.1. *For any $x\in\textsc{Yes}_n\cup\textsc{No}_n$:*

$$\begin{aligned}
\Delta(\textsc{View}_{\mathsf{V}}\big(\mathsf{P},\mathsf{V}(x)\big),\textsc{View}_{\mathsf{V}}\big(\mathsf{P}_S,\mathsf{V}(x)\big)) &= \Delta\left(\mathsf{W}_q(x),\mathsf{W}_0(x)\right) \\
&\leq q\cdot\epsilon.
\end{aligned}$$

We now proceed to check the distance $\Delta(\text{VIEW}_{\mathsf{V}}(\mathsf{P}_S, \mathsf{V}(x)), \text{VIEW}_{\mathsf{V}}(\mathsf{P}_{SS}, \mathsf{V}(x)))$. Define a sequence of hybrid algorithms W'_j:

$\mathsf{W}'_j(x)$:

1. $r_{\mathsf{V}} \leftarrow \mathcal{R}_{\mathsf{V}}$.
2. $r_{\mathsf{P}} \leftarrow \mathcal{R}_{\mathsf{P}}$.
3. $u_1 \leftarrow \mathsf{V}(x, r_{\mathsf{V}}, \phi)$.

   For $i \leftarrow 1$ to j

   (a) $(n, r_{\mathsf{Sim},i}) \leftarrow \mathsf{Sim}_i^{-1}(u_1, y_1, \ldots, u_i)$.
   (b) $(\ldots, u_i, y_i, \ldots) \leftarrow \mathsf{Sim}_{\mathsf{P}}(n; r_{\mathsf{Sim},i})$.
   (c) $u_{i+1} \leftarrow \mathsf{V}(x, r_{\mathsf{V}}, u_1, \ldots, y_i)$.

   For $i \leftarrow j+1$ to q

   (a) $(n, r_{\mathsf{Sim},i}) \leftarrow \mathsf{A}_i(u_1, y_1, \ldots, u_i)$.
   (b) $(\ldots, u_i, y_i, \ldots) \leftarrow \mathsf{Sim}_{\mathsf{P}}(n; r_{\mathsf{Sim},i})$.
   (c) $u_{i+1} \leftarrow \mathsf{V}(x, r_{\mathsf{V}}, (u_1, \ldots, y_i))$.

Outputs $x, r_{\mathsf{V}}, u_1, y_1, \ldots, u_q, y_q$.

It's clear that $\mathsf{B}(x) = \mathsf{V}(\mathsf{W}'_0(x)) = \langle \mathsf{P}_{SS}, \mathsf{V}(x) \rangle$ and $\mathsf{W}'_q(x) = \mathsf{W}_0(x) = \text{VIEW}_{pub}(\mathsf{P}_S, \mathsf{V})$.

Claim 5.10.2. *For any $x \in \text{YES}_n \cup \text{NO}_n$, $j \in [q(n)]$:*

$$\Delta\left(\mathsf{W}'_j(x), \mathsf{W}'_{j-1}(x)\right) \leq 3(q+1) \cdot \epsilon.$$

Proof of Claim 5.10.2. For $j \in [q(n)], i \in [2q(n)]$, similarly define $\mathsf{W}'_j(x)[i]$ as in the proof of Claim 5.9.1. Following Claim 5.10.1 and ϵ-hiding,

$$\begin{aligned}
\Delta\left(\mathsf{W}'_{j-1}(x)[2\cdot(j-1)], \mathsf{Sim}_{j-1}(n)\right) &\leq \Delta\left(\text{VIEW}_{pub}(\mathsf{P}_S, \mathsf{V}(n)), \text{VIEW}_{pub}(\mathsf{P}_S, \mathsf{V}_S(x))\right) \\
&\leq \Delta\left(\text{VIEW}_{pub}(\mathsf{P}_S, \mathsf{V}_S(n)), \text{VIEW}_{pub}(\mathsf{P}, \mathsf{V}(x))\right) \\
&\qquad + \Delta\left(\text{VIEW}_{pub}(\mathsf{P}, \mathsf{V}(n)), \text{VIEW}_{pub}(\mathsf{P}_S, \mathsf{V}(x))\right) \\
&\leq \epsilon + q \cdot \epsilon \\
&\leq (q+1) \cdot \epsilon,
\end{aligned}$$

where the first inequality is because the first $(j-1)$-round of W'_{j-1} is $\text{VIEW}_{pub}(\mathsf{P}_S, \mathsf{V}(n))[2\cdot(j-1)]$, the second inquality follows triangle inequality, and the third inequality follows hiding and Claim 5.10.1.

By definition of A_{j-1}, we have $\Delta\left(\mathsf{A}_{j-1}\left(\mathsf{Sim}_{j-1}(n; R_{\mathsf{Sim},j-1})\right), \mathsf{Sim}_{j-1}^{-1}(\mathsf{Sim}_{j-1}(n; R_{\mathsf{Sim},j-1}))\right) < \text{negl}(n)$, and thus:

$$\begin{aligned}
\Delta\left(\mathsf{W}'_j(x), \mathsf{W}'_{j-1}(x)\right) &\leq \Delta\left(\mathsf{A}_j\left(\mathsf{W}'_{j-1}(x)[j-1]\right), \mathsf{Sim}_{j-1}^{-1}(\mathsf{W}'_{j-1}(x)[j-1])\right) \\
&\leq \Delta\left(\mathsf{A}_j\left(\mathsf{Sim}_{j-1}(n; R_{\mathsf{Sim},j-1})\right), \mathsf{Sim}_{j-1}^{-1}(\mathsf{Sim}_{j-1}(n; R_{\mathsf{Sim},j-1}))\right) + 2 \cdot \Delta\left(\mathsf{W}'_{j-1}(x)[j-1], \mathsf{Sim}_{j-1}(n)\right) \\
&\leq \text{negl}(n) + 2(q+1) \cdot \epsilon.
\end{aligned}$$

where the first inequality follows data processing inequality, and the second inequality follows triangle inequality, data processing inequality and the fact that $\mathsf{W}'_{j-1}(x)[j-1] = \mathsf{W}'_j(x)[j-1]$. □

Thus by triangle ineqaulity, we have:

Claim 5.10.3. *For any $x \in \text{YES}_n \cup \text{NO}_n$:*

$$\begin{aligned}\Delta(\text{VIEW}_{\mathsf{V}}(\mathsf{P}_{SS}, \mathsf{V}(x)), \text{VIEW}_{\mathsf{V}}(\mathsf{P}_S, \mathsf{V}(x))) &= \Delta\left(\mathsf{W}'_q(x), \mathsf{W}'_0(x)\right) \\ &\leq 3q \cdot (q+1)\epsilon.\end{aligned}$$

Given Claim 5.9.1 and Claim 5.10.3, and by triangle inequality and data processing inequality, the following claim follows:

Claim 5.10.4. *For any $x \in \text{YES}_n \cup \text{NO}_n$:*

$$\begin{aligned}\Delta\left(\mathsf{B}(x), \langle \mathsf{P}, \mathsf{V}(x)\rangle\right) &\leq \Delta(\text{VIEW}_{\mathsf{V}}(\mathsf{P}_{SS}, \mathsf{V}(x)), \text{VIEW}_{\mathsf{V}}(\mathsf{P}, \mathsf{V}(x))) \\ &\leq \Delta(\text{VIEW}_{pub}(\mathsf{P}_{SS}, \mathsf{V}(x)), \text{VIEW}_{pub}(\mathsf{P}_S, \mathsf{V}(x))) \\ &\qquad + \Delta(\text{VIEW}_{pub}(\mathsf{P}_S, \mathsf{V}(x)), \text{VIEW}_{pub}(\mathsf{P}, \mathsf{V}(x))) \\ &\leq \text{negl}(n) + 3q \cdot (q+1) \cdot \epsilon + q \cdot \epsilon.\end{aligned}$$

Therefore, by triangle inequality

$$\Pr\left[B(x) = \Pi(x)\right] \geq \Pr\left[\langle \mathsf{P}, \mathsf{V}(x)\rangle = \Pi(x)\right] - \Delta\left(\mathsf{B}(x), \langle \mathsf{P}, \mathsf{V}(x)\rangle\right) \geq 1 - 4q^2\epsilon.$$

This shows that B solves Π in worst case with bounded error, which contradicts our assumption. Thus there must exist one Sim_i among $(\mathsf{Sim}_1, \ldots, \mathsf{Sim}_q)$ that is one-way function. By concatenating $(\mathsf{Sim}_1, \ldots, \mathsf{Sim}_q)$, we get an explicit construction for one-way function. [17]

□

Remark 5.11. Unlike the OWF in Theorem 5.3, which incorporates the verifier algorithm V as part of the construction, the explicit construction of OWF from Simulatable-IHIP uses solely the efficient simulator $\mathsf{Sim}_{\mathsf{P}}$, which does not take the instance x as input. This is what allows the worst-case hardness of the problem to be useful in proving one-wayness of the latter function. Notably, the construction of OWF in [Ost91], based on SZK simulator, also takes the instance as part of the input, thus requiring hardness also over some distribution of instances. It is of interest to find whether a *worst-case hard* with just IHIP, which is not guaranteed an efficient similator for honest prover, implies OWF.

6 Oracle Separation from SZK

Given that SZK and IHIP are both contained in $\mathsf{NP/poly} \cap \mathsf{coNP/poly}$, it is natural to ask whether one is contained in the other. While we don't know how to construct IHIP protocols for SZK problems such as Statistical Difference and Graph Non-Isomorphism, it is also unclear whether IHIP is contained in SZK. Towards understanding their relationship, we exhibit an oracle relative to which $\mathsf{IHIP} \not\subset \mathsf{SZK}$. Before advancing, it is essential to define the associated complexity classes.

Definition 6.1 (Class SZK)**.** A promise problem $\Pi = (\text{YES}, \text{NO})$ is in SZK if there exists a protocol $\langle \mathsf{P}, \mathsf{V}\rangle$, where the verifier runs in polynomial time, satisfying the following:

- $\langle \mathsf{P}, \mathsf{V}\rangle$ is an interactive proof for Π (where both P and V get the input instance) with negligible completeness and soundness errors.

[17] Similarly to the arguments in Lemma 5.4, the functions can be well defined on all input lengths by truncating the input to the nearest $(|r_{\mathsf{Sim},1}|, \ldots, |r_{\mathsf{Sim},q}|)$, thereby establishing an implication for one-way function.

- There exists an efficient algorithm Sim such that for any efficient V^* and any $x \in \textsc{Yes}_n$,

$$\Delta\Big(\mathsf{Sim}_{\mathsf{V}^*}(x), \textsc{View}_{\mathsf{V}^*}\big(\mathsf{P}(x), \mathsf{V}^*(x)\big) \leq \mathrm{negl}(n).$$

Definition 6.2. An oracle protocol is a protocol $\langle \mathsf{P}, \mathsf{V}\rangle$ in which both P and V are allowed to make calls to an oracle. For any oracle $\mathcal{O} : \{0,1\}^* \to \{0,1\}^*$, such a protocol with oracle $\mathcal{O}$ is denoted by $\langle \mathsf{P}, \mathsf{V}\rangle^{\mathcal{O}}$. The view of each party in such a protocol also includes the set of oracle queries it makes and the corresponding responses. The conditions for $\langle \mathsf{P}, \mathsf{V}\rangle^{\mathcal{O}}$ being a (Simulatable) instance-hiding proof (resp. SZK) system for a promise problem are the same as those in Definition 2.2 (resp. Definition 6.1), except that the simulator is also allowed access to the same oracle.

Definition 6.3. For any oracle $\mathcal{O} : \{0,1\}^* \to \{0,1\}^*$, $\mathsf{Simulatable\text{-}IHIP}^{\mathcal{O}}$ (resp. $\mathsf{SZK}^{\mathcal{O}}$) is the class of promise problems that have a simulatable instance-hiding proof (resp. SZK proof) system with oracle $\mathcal{O}$. This includes promise problems whose definitions involve properties of the oracle.

Theorem 6.4. *There exists an oracle $\mathcal{O}$ such that $\mathsf{Simulatable\text{-}IHIP}^{\mathcal{O}} \not\subseteq \mathsf{SZK}^{\mathcal{O}}$.*

Looking ahead, our oracle separation will use oracles based on the generic group model [Sho97]. Rather than Shoup's original formulation of this model, we will use the following formulation as in Corrigan-Gibbs and Kogan [CGK18], which will be more convenient to use.

Definition 6.5 (Generic Group Oracles)**.** For any $N \in \mathbb{N}$, and bijective function $\sigma : \mathbb{Z}_N \to [N]$, the oracle $\mathcal{G}_\sigma : [N] \times [N] \to [N]$ is defined as: $\mathcal{G}_\sigma(g,h) = \sigma(\sigma^{-1}(g) + \sigma^{-1}(h))$. We refer to elements of $[N]$ in this context as the group elements, and the corresponding inverses of σ as their discrete logarithms.

Let $\mathcal{N} : \mathbb{N} \to \mathbb{N}$ be the function that, on input any $n \in \mathbb{N}$, outputs the smallest prime number larger than 2^{n-1}. For any family of oracles $\mathcal{I} = \{\mathcal{I}_n : \mathbb{Z}_{\mathcal{N}(n)} \to \{0,1\}\}_{n\in\mathbb{N}}$, and family of bijective functions $\sigma = \{\sigma_n : \mathbb{Z}_{\mathcal{N}(n)} \to [\mathcal{N}(n)]\}_{n\in\mathbb{N}}$, we define the promise problem $\Pi^{\mathcal{I},\sigma} = (\textsc{Yes}, \textsc{No})$ as follows:

$$\textsc{Yes}_n = \{(n, \mathcal{N}(n), \sigma_n(1), \sigma_n(x)) \mid \mathcal{I}_n(x) = 1\}.$$
$$\textsc{No}_n = \{(n, \mathcal{N}(n), \sigma_n(1), \sigma_n(x)) \mid \mathcal{I}_n(x) = 0\}.$$

We extend the notation $\mathcal{G}_\sigma$ in the natural manner to denote the family of oracles $\mathcal{G}_{\sigma_n}$ for all $n \in \mathbb{N}$. Similarly, note that to define $\textsc{Yes}_n(\Pi^{\mathcal{I},\sigma})$ and $\textsc{No}_n(\Pi^{\mathcal{I},\sigma})$, only $\mathcal{I}_n$ and σ_n need to be specified; we denote the corresponding promise problem restricted to instances of size n by $\Pi^{\mathcal{I}_n,\sigma_n}$.

We show that there is an instance-hiding oracle protocol that, given oracle access to $\mathcal{I}$ and $\mathcal{G}_\sigma$, is a valid instance-hiding proof for $\Pi^{\mathcal{I},\sigma}$, whereas every oracle protocol fails to be an SZK proof for this language for most such oracles. This already gives a separation between "generic" instance-hiding and SZK protocols that only use group elements in a generic manner. To show the oracle separation, a careful diagonalization argument is needed. We first state and prove the following two lemmas that show the above statements, and then set up and perform the required diagonalization, which proves Theorem 6.4.

Lemma 6.6. *There is an oracle protocol $\langle \mathsf{P}, \mathsf{V}\rangle$ such that, for any $\mathcal{I}$ and σ as above, $\langle \mathsf{P}, \mathsf{V}\rangle^{\mathcal{I},\mathcal{G}_\sigma}$ is a simulatable instance-hiding proof system for $\Pi^{\mathcal{I},\sigma}$.*

Proof of Lemma 6.6. The required IHIP protocol is presented in Figure 6. The completeness and soundness of the protocol follows from the fact that $y = \sigma_n(x + r)$, σ_n is a bijection, and so for V to accept we have $x^* = x + r$, and so $\mathcal{I}_n(x^* - r) = \mathcal{I}(x)$. It is perfectly instance-hiding because the only message sent by V, for any instance of size n, is a uniformly random element of $[\mathcal{N}(n)]$. It is simulatable instance-hiding because the view of the honest prover P can be efficiently simulated (perfectly) given n by sampling a random element $x^* \leftarrow \mathbb{Z}_{\mathcal{N}(n)}$, and outputting $(\sigma_n(x^*), x^*)$.[18] This completes the proof. □

[18] For this simulator to be uniform, $\mathcal{N}(n)$ has to be efficiently computable, which it may be not. We ignore this detail as this is resolved by simply adding another oracle that, on input n, outputs $\mathcal{N}(n)$.

Simulatable-IHIP protocol for $\Pi^{\mathcal{I},\sigma}$: $\langle \mathsf{P}, \mathsf{V} \rangle^{\mathcal{I},\mathcal{G}_\sigma}$

Inputs:

- P is given the input size n.
- V has input $(n, \mathcal{N}(n), \sigma_n(1), \sigma_n(x))$ for some $x \in \mathbb{Z}_{\mathcal{N}(n)}$.

Protocol:

1. V : randomly sample $r \leftarrow \mathbb{Z}_{\mathcal{N}(n)}$ and compute $\sigma_n(r)$.
   - This can be done with $O(\log \mathcal{N}(n))$ calls to $\mathcal{G}_{\sigma_n}$ using repeated doubling of $\sigma_n(1)$.
2. V : compute and send $y \leftarrow \mathcal{G}_{\sigma_n}(\sigma_n(x), \sigma_n(r))$ to P.
3. P : compute $x^* \in \mathbb{Z}_{\mathcal{N}(n)}$ such that $y = \sigma_n(x^*)$ and send x^* to V
4. V : accept if and only if $y = \sigma_n(x^*)$ and $\mathcal{I}_n(x^* - r) = 1$.

Figure 6: Simulatable-IHIP protocol for $\Pi^{\mathcal{I},\sigma}$

Lemma 6.7. *For any oracle protocol $\langle \mathsf{P}, \mathsf{V} \rangle$ and polynomial-time oracle algorithm Sim, there is an $n_0 \in \mathbb{N}$ such that for all $n \geq n_0$, there exists some $\mathcal{I}_n$ and σ_n as above such there is either an input in $\textsc{No}_n(\Pi^{\mathcal{I}_n,\sigma_n})$ on which $\langle \mathsf{P}, \mathsf{V} \rangle^{\mathcal{I}_n,\mathcal{G}_{\sigma_n}}$ has soundness error $\Omega(1/n)$, or an input in $\textsc{Yes}_n(\Pi^{\mathcal{I}_n,\sigma_n})$ on which it has either completeness error $\Omega(1/n)$, or honest-verifier statistical zero-knowledge error $\Omega(1/n)$ with $\mathsf{Sim}^{\mathcal{I}_n,\mathsf{G}_{\sigma_n}}$ as the simulator.*

Our proof of Lemma 6.7 relies on the hardness of the discrete logarithm problem for generic algorithms, which require at least $N^{1/2}$ time on groups of order N, as shown by Shoup [Sho97]. Our modelling of such algorithms is more general than Shoup's, so we instead use the following corollary of a theorem of Corrigan-Gibbs and Kogan [CGK18, Theorem 2].

Lemma 6.8. *For any prime $N \in \mathbb{N}$ and oracle algorithm A that receives at most $\sqrt{N}$-bit non-uniform advice and runs in time T,*

$$\Pr_{\sigma,x}[\mathsf{A}^{\mathcal{G}_\sigma}(\sigma(1), \sigma(x)) = x] \leq \frac{T^{2.01}}{\sqrt{N}}.$$

where σ is a uniformly random bijective function from $\mathbb{Z}_N$ to $[N]$, and x is uniformly random over $\mathbb{Z}_N$.

Proof of Lemma 6.7. Consider any oracle protocol $\langle \mathsf{P}, \mathsf{V} \rangle$ and polynomial-time oracle algorithm Sim, and some $n \in \mathbb{N}$. Suppose, towards a contradiction, that for every $\mathcal{I}_n$ and σ_n as above, the protocol $\langle \mathsf{P}, \mathsf{V} \rangle^{\mathcal{I}_n,\mathcal{G}_{\sigma_n}}$ is complete and sound as an interactive proof for $\Pi^{\mathcal{I}_n,\sigma_n}$, and is also honest-verifier SZK, with $\mathsf{Sim}^{\mathcal{I}_n,\mathcal{G}_{\sigma_n}}$ as the simulator, all with errors $o(1/n)$. We will simply denote $\mathcal{I}_n$ and σ_n as $\mathcal{I}$ and σ in the rest of the proof for brevity.

By the zero-knowledge property, for any $\mathcal{I}$ and σ, we have the following on any YES instance $z = (n, \mathcal{N}(n), \sigma(1), \sigma(x))$:

$$\Delta\Big(\mathsf{Sim}^{\mathcal{I},\mathcal{G}_\sigma}(z), \textsc{View}_{\mathsf{V}}\big(\mathsf{P}^{\mathcal{I},\mathcal{G}_\sigma}(z), \mathsf{V}^{\mathcal{I},\mathcal{G}_\sigma}(z)\big)\Big) \leq o\left(\frac{1}{n}\right). \tag{8}$$

Denote by $E(\mathcal{I}, \sigma, x)$ the event that, given oracle access to $\mathcal{I}$ and $\mathcal{G}_\sigma$, on input $z = (n, \mathcal{N}(n), \sigma(1), \sigma(x))$ as input, the verifier V queries the oracle $\mathcal{I}$ at x during the course of its interaction with P. Note that the view of the verifier consists of its randomness, the transcript of the protocol and oracle queries/responses.

Together, given the oracle, these also fully determine the set of oracle queries made by the verifier during its execution. Denote by $E_{\mathsf{Sim}}(\mathcal{I}, \sigma, x)$ the event that x is among the set of oracle queries so determined by the view output by the simulator $\mathsf{Sim}^{\mathcal{I},\mathcal{G}_\sigma}(z)$. By Lemma 6.8 and the poly(n)-time efficiency of Sim, since $\mathcal{N}(n) > 2^{n-1}$, we have:

$$\Pr_{\mathcal{I},\sigma,x\leftarrow\mathbb{Z}_{\mathcal{N}(n)}}\left[E_{\mathsf{Sim}}(\mathcal{I},\sigma,x)\right] \leq o\left(\frac{1}{n}\right). \tag{9}$$

This is because if not, x can be extracted from the view output of Sim by an algorithm that simulates the oracle $\mathcal{I}$ to it. As $\mathcal{I}$ above is a uniformly random Boolean function, the probability that more than $1/3$ of the $x \in \mathbb{Z}_{\mathcal{N}(n)}$ are in $\mathcal{I}^{-1}(1)$ is $(1 - \mathrm{negl}(n))$; denote by I the set of $\mathcal{I}$ in which this happens. Thus, by some elementary probability arguments, we have the following for all n:

$$\begin{aligned}
\Pr_{\mathcal{I},\sigma,x\leftarrow\mathbb{Z}_{\mathcal{N}(n)}\cap\mathcal{I}^{-1}(1),\mathsf{Sim}}\left[E_{\mathsf{Sim}}(\mathcal{I},\sigma,x)\right] &\leq \Pr\left[\mathcal{I}\in I\right] \mathop{\mathbb{E}}_{\mathcal{I}\leftarrow I}\left[\Pr_{\sigma,x\leftarrow\mathbb{Z}_{\mathcal{N}(n)}\cap\mathcal{I}^{-1}(1)}\left[E_{\mathsf{Sim}}(\mathcal{I},\sigma,x)\right]\right] + \Pr\left[\mathcal{I}\notin I\right] \\
&\leq \Pr\left[\mathcal{I}\in I\right] \mathop{\mathbb{E}}_{\mathcal{I}\leftarrow I}\left[\frac{\Pr_{\sigma,x\leftarrow\mathbb{Z}_{\mathcal{N}(n)}}\left[E_{\mathsf{Sim}}(\mathcal{I},\sigma,x)\right]}{\Pr_{x\leftarrow\mathbb{Z}_{\mathcal{N}(n)}}\left[x\in\mathcal{I}^{-1}(1)\right]}\right] + \Pr\left[\mathcal{I}\notin I\right] \\
&\leq \Pr\left[\mathcal{I}\in I\right] \mathop{\mathbb{E}}_{\mathcal{I}\leftarrow I}\left[\frac{\Pr_{\sigma,x\leftarrow\mathbb{Z}_{\mathcal{N}(n)}}\left[E_{\mathsf{Sim}}(\mathcal{I},\sigma,x)\right]}{1/3}\right] + \Pr\left[\mathcal{I}\notin I\right] \\
&\leq 3\cdot \Pr_{\mathcal{I},\sigma,x\leftarrow\mathbb{Z}_{\mathcal{N}(n)}}\left[E_{\mathsf{Sim}}(\mathcal{I},\sigma,x)\right] + \Pr\left[\mathcal{I}\notin I\right] \\
&\leq o\left(\frac{1}{n}\right).
\end{aligned} \tag{10}$$

where the last inequality holds for all large enough n. Such $x \in \mathcal{I}^{-1}(1)$ correspond directly to YES instances of $\Pi^{\mathcal{I},\mathcal{G}_\sigma}$, in which case the simulator is supposed to work. So by Eqs. (8) and (10), we have:

$$\Pr_{\mathcal{I},\sigma,x\leftarrow\mathbb{Z}_{\mathcal{N}(n)}\cap\mathcal{I}^{-1}(1),\mathsf{P},\mathsf{V}}\left[E(\mathcal{I},\sigma,x)\right] \leq o\left(\frac{1}{n}\right) \tag{11}$$

This implies that, for all large enough n, there exists an $\mathcal{I}$, σ, and $x \in \mathbb{Z}_{\mathcal{N}(n)} \cap \mathcal{I}^{-1}(1)$ such that, when interacting with the honest prover and given input $(n, \mathcal{N}(n), \sigma(1), \sigma(x))$, the verifier queries $\mathcal{I}$ at x with $o(1)$ probability. Consider the oracle $\mathcal{I}^{-x}$, which is the same as $\mathcal{I}$, except that $\mathcal{I}^{-x}(x) = 0$. The verifier can be made to accept the input $(n, \mathcal{N}(n), \sigma(1), \sigma(x))$ with probability $(1 - o(1))$ given oracle access to $\mathcal{I}^{-x}$ and $\mathcal{G}_\sigma$ by a malicious prover that simply emulates the honest prover's strategy with oracle $\mathcal{I}$. This breaks the soundness of the protocol for the problem $\Pi^{\mathcal{I}^{-x},\sigma}$. This is a contradiction to our assumption at the start of the proof, and this proves the lemma.

□

Now, we show how to use Lemma 6.7 to construct functions $\mathcal{I}$ and σ such that with respect the oracles $(\mathcal{I}, \mathcal{G}_\sigma)$, the problem $\Pi^{\mathcal{I},\sigma}$ does not have an SZK protocol. The standard approach to doing so is diagonalization: to enumerate all possible oracle protocols and simulators, and for each pick an input size n and corresponding $\mathcal{I}_n$ and σ_n on which it fails (as promised by Lemma 6.7), and include that in $\mathcal{I}$ and σ. However, we cannot do this directly, as the set of all protocols is not countable. Instead, we will do this for a countable set of protocols and show that for any potential SZK protocol, there is a protocol in this set that computes the same problem. This countable set will be the set of protocols with simulation-based provers [For87].

Definition 6.9 (Simulation-based Protocols)**.** Given (oracle) algorithms V and Sim, the *simulation-based protocol* defined by these is the protocol $\langle \mathsf{P}_{\mathsf{Sim}}, \mathsf{V}\rangle$, where $\mathsf{P}_{\mathsf{Sim}}$ is the *simulation-based prover* that behaves as follows: to compute the prover's message at any point in the protocol, sample from the distribution of this message in the output of Sim, conditioned on this output matching the protocol transcript so far (if this conditional distribution is not defined, set the message to be $\perp$).

The following lemma follows immediately from the statistical zero-knowledge property. (Roughly this statement is also proven as part of the proof of Theorem 5.9.)

Lemma 6.10. *Suppose $\langle \mathsf{P}, \mathsf{V} \rangle$ is an honest-verifier SZK proof system for a promise problem Π with simulator Sim, all with respect to some oracle $\mathcal{O}$. Then the simulation-based protocol $\langle \mathsf{P}_{\mathsf{Sim}}, \mathsf{V} \rangle$ is also an honest-verifier SZK proof for Π with simulator Sim, with respect to oracle $\mathcal{O}$.*

Given this lemma, we only need to enumerate over pairs of polynomial-time algorithms $(\mathsf{V}, \mathsf{Sim})$ in our diagonalization argument, and this is indeed a countable set. Together, Lemmas 6.6, 6.10 and 6.11 prove Theorem 6.4.

Lemma 6.11. *There exist $\mathcal{I} = \{\mathcal{I}_n\}$ and $\sigma = \{\sigma_n\}$ such that no simulation-based protocol is a valid SZK proof system for $\Pi^{\mathcal{I},\sigma}$ with respect to oracles $(\mathcal{I}, \mathcal{G}_\sigma)$.*

Proof. We prove this by enumerating over all polynomial-time oracle algorithms $(\mathsf{V}, \mathsf{Sim})$, and constructing $\mathcal{I}$ and σ such that each of them fails to be an SZK proof on some input size. One subtlety here is that Lemma 6.7 is stated only for algorithms that have access to $\mathcal{I}_n$ and σ_n for some specific value of n, whereas in reality they will be able to query $\mathcal{I}_{n'}$ or $\sigma_{n'}$ for values of n' different from the input size. We will need to make sure these queries are not useful to the algorithm.

Consider an enumeration of all pairs of polynomial-time oracle algorithms $((\mathsf{V}_1, \mathsf{Sim}_1), (\mathsf{V}_2, \mathsf{Sim}_2), \dots)$. We will process them in order to construct $\mathcal{I}$ and σ as follows. Below, the "default" values for any $\mathcal{I}_n$ are the all-zero function, and for σ_n it is the identity function. When we say a protocol "fails to be an SZK proof" at some input size n, we mean that one of its errors on some input of that size is at least $1/n$. Note that we only need to rule out protocols that on every input size n has errors smaller than $1/n$.

First, we claim that there has to exist an n_1 such that if for all $n \neq n_1$ we set $\mathcal{I}_n$ and σ_n to the default values, there exist $\mathcal{I}^*_{n_1}$ and $\sigma^*_{n_1}$ such that the simulation-based protocol defined by $(\mathsf{V}_1, \mathsf{Sim}_1)$ fails to be an SZK proof for $\Pi^{\mathcal{I},\sigma}$ at input length n_1. If not, then we can contradict Lemma 6.7 by running this protocol with the oracles artifically set to the default values at n that is different from the input size. Further, as V_1 and Sim_1 are polynomial-time algorithms, there is a number n'_1 such that when run on inputs of size n_1, neither algorithm ever makes a query to an oracle $\mathcal{I}_n$, etc., for $n > n'_1$. We will start by setting $\mathcal{I}_{n_1} = \mathcal{I}^*_{n_1}$, $\sigma_{n_1} = \sigma^*_{n_1}$, and $\mathcal{I}_n$ and σ_n to be the default values for all $n \leq n'_1$.

Next, we will essentially repeat this argument for $(\mathsf{V}_2, \mathsf{Sim}_2)$ – there has to exist an n_2 larger than n'_1 such that if we set the oracles as above for $n \leq n'_1$, to the default values for $n > n'_1$ except for $n = n_2$, there exist $\mathcal{I}^*_{n_2}$ and $\sigma^*_{n_2}$ for which $(\mathsf{V}_2, \mathsf{Sim}_2)$ fails to be an SZK proof. Otherwise we can contradict Lemma 6.7 by running this protocol with oracles artificially set to the ones determined above for $n \leq n'_1$[19], and to default values otherwise. We then set $\mathcal{I}_{n_2}$ and σ_{n_2} to be these $\mathcal{I}^*_{n_2}$ and $\sigma^*_{n_2}$, and set default values for n up to the similarly defined n'_2. Repeating this argument throughout the enumeration then gives us the $\mathcal{I}$ and σ that prove the lemma.

□

7 Instance-Hiding Delegation Schemes

In this section, we extend the study to a setting in which a machine V, given an private input x, delegates the computational task of computing a function $f(x)$ to a computationally stronger machine P. We seek solutions in which the prover P does this without learning x, but without asking for any guarantees in case P does not follow the protocol. We further generalize the definition to allow the prover to learn leakage $\ell(x)$ of instance x defined by some PSPACE function ℓ. We note that because some results (e.g. Theorem 4.1, Theorem 4.3, Theorem 5.3, Theorem 5.9) about IHIP in previous sections rely solely on its correctness of honest prover and hiding properties (and do not need the soundness property), these can be generalized to

[19] Given that $n_1 \ll n_2$, these oracles can be encoded as non-uniform advice with random access and a length no longer than $\sqrt{\mathcal{N}(n_2)}$.

this setting. We further show strong connections between the existence of such delegation schemes and of IHIP's, and then use these connections together with closure properties of these schemes to show the closure properties of IHIP's stated in Section 2.1. Below, as before, for any set S, S_n denotes its intersection with $\{0,1\}^n$. For any promise problem $\Pi = (\textsc{Yes}, \textsc{No})$, we define its characteristic function to be the partial function that maps inputs in Yes to 1, inputs in No to 0, and is undefined on other inputs.

Definition 7.1 (Instance-Hiding Delegation Scheme (IHD) [FO91])**.** Consider a function $f : \mathcal{X} \to \mathcal{Y}$, and functions $\delta, \epsilon : \mathbb{N} \to [0,1]$ and $\ell : \mathcal{X} \to \{0,1\}^*$. A (δ, ϵ, ℓ)-*Instance-Hiding Delegation Scheme* (IHD) for f is a protocol $\langle \mathsf{P}, \mathsf{V} \rangle$ in which a probabilistic polynomial-time verifier V interacts with a computationally unbounded prover P. For some $n \in \mathbb{N}$, V gets a *private* input $x \in \mathcal{X}_n$, while P gets the input n. At the end of the interaction, V outputs $y \in \mathcal{Y} \cup \{\perp\}$. The protocol is required to satisfy the following properties for all large enough $n \in \mathbb{N}$:

- **Correctness:** For any input $x \in \mathcal{X}_n$:

$$\Pr\left[\langle \mathsf{P}(n), \mathsf{V}(x) \rangle = f(x)\right] \geq 1 - \delta(n).$$

- **Hiding Against Honest Prover (with leakage):** There exists a computationally unbounded randomized algorithm Sim_P such that for any input $x \in \mathcal{X}_n$:

$$\Delta\Big(\mathsf{Sim}_\mathsf{P}\big(n, \ell(x)\big), \textsc{View}_P(\mathsf{P}(n), \mathsf{V}(x))\Big) \leq \epsilon(n).$$

If the simulator Sim_P is efficient, we call the protocol *Simulatable-Instance-Hiding Delegation* (Simulatable-IHD). The protocol is *perfectly-hiding* if $\epsilon(n) = 0$. If left unspecified, we assume ℓ is the constant function that always outputs $\perp$, corresponding to the absence of leakage.

The following proposition directly follows the completeness and soundness for the honest prover in IHIP, and ϵ-hiding.

Proposition 7.2. *Consider a promise problem Π with characteristic function $f = \{f_n : \{0,1\}^n \to \{0,1\} \cup \{\perp\}\}_{n \in \mathbb{N}}$, and let $\langle \mathsf{P}, \mathsf{V} \rangle$ be a (δ, ϵ)-IHIP (resp. simulatable IHIP) for Π, then $\langle \mathsf{P}, \mathsf{V} \rangle$ is a (ϵ, δ)-IHD (resp. simulatable IHD) for f.*

Next, we define a version of instance-hiding delegation schemes that has an additional verifiability property that protects against provers that may deviate from the protocol.

Definition 7.3 (Verifiable Instance-Hiding Delegation Scheme (VIHD) [FO91])**.** Consider any function $f : \mathcal{X} \to \mathcal{Y}$, and functions $\delta, \epsilon : \mathbb{N} \to [0,1]$, and $\ell : \mathcal{X} \to \mathcal{Z}$. A (δ, ϵ, ℓ)-*Verifiable Instance-Hiding Delegation Scheme* (VIHD) for f is a IHD protocol that additionally achieves verifiability and hiding against malicious provers:

- **Correctness:** For any input $x \in \mathcal{X}_n$:

$$\Pr\left[\langle \mathsf{P}(n), \mathsf{V}(x) \rangle = f(x)\right] \geq 1 - \delta(n).$$

- **Verifiability:** For any prover P^*, for any input $x \in \mathcal{X}_n$:

$$\Pr\left[\langle \mathsf{P}^*(n), \mathsf{V}(x) \rangle \in \{f(x), \perp\}\right] \geq 1 - \delta(n).$$

- **Hiding against malicious prover (with leakage):** For any prover P^*, there exists a computationally unbounded randomized algorithm $\mathsf{Sim}_{\mathsf{P}^*}$ such that for any input $x \in \mathcal{X}_n$,

$$\Delta\big(\mathsf{Sim}_{\mathsf{P}^*}\big(n, \ell(x)\big), \textsc{View}_{\mathsf{P}^*}(\mathsf{P}^*, \mathsf{V}(x))\big) \leq \epsilon(n).$$

The VIHD is simulatable if the simulator for honest prover $\mathsf{Sim}_{\mathsf{P}}$ is efficient. If left unspecified, we assume ℓ is the constant function that always outputs $\perp$, corresponding to the absence of leakage.

We then have the following proposition, which states that VIHD protocols are essentially stronger than IHIP protocols.

Proposition 7.4. *Let f be the characteristic function for a promise problem Π. If $\langle \mathsf{P}, \mathsf{V}\rangle$ is a (δ, ϵ)-verifiable instance-hiding delegation (resp. simulatable (δ, ϵ)-verifiable instance-hiding delegation) for f, then there exists a (δ, ϵ)-instance-hiding proof (resp. simulatable (δ, ϵ)-instance-hiding interactive proof) $\langle \mathsf{P}', \mathsf{V}'\rangle$ for Π.*

Proof Sketch. We define $\mathsf{P}' = \mathsf{P}$ and let V' run V as a black box on the same input, and output 1 (Accept) if and only if V outputs 1. The δ-completeness and ϵ-hiding follow the correctness and ϵ-hiding of $\langle \mathsf{P}, \mathsf{V}\rangle$ respectively. The soundness of $\langle \mathsf{P}', \mathsf{V}'\rangle$ follows from the correctness and verifiability of $\langle \mathsf{P}, \mathsf{V}\rangle$. □

7.1 Verifiable IHD from IHD

It is immediate that any VIHD is also a IHD. If restricting ℓ to be constant function, [FO91] demonstrates that if any function $f \in \mathsf{PSPACE}$ has perfect-hiding $(\delta, 0, \ell)$-IHD, then f also has a perfect-hiding $(\delta', 0, \ell)$-VIHD. We extend their theorem to ϵ-hiding schemes with a richer class of leakage function ℓ. The protocol and proof essentially closely follow that of [FO91, Lemma 3.1], with the only difference being in the hiding statements that the prover proves to the verifier in each round. In our case, the prover proves ϵ-hiding with respect to any PSPACE leakage function, whereas in [FO91] the focus is on perfect hiding and only the constant function as leakage.

Theorem 7.5. *Suppose that a function f is computable in polynomial space and has a (δ, ϵ, ℓ)-IHD $\langle \mathsf{P}, \mathsf{V}\rangle$ for some negligible δ, ϵ and $\ell \in \mathsf{PSPACE}$, then there exists negligible δ', ϵ' and $\langle \mathsf{P}', \mathsf{V}'\rangle$ such that $\langle \mathsf{P}', \mathsf{V}'\rangle$ is a $(\delta', \epsilon', \ell)$-VIHD for f.*

Corollary 7.6. *Suppose that a function f is computable in polynomial space and has a (δ, ϵ)-IHD for some negligible δ, ϵ. If f is the characteristic function for a promise problem Π, then Π has a (δ', ϵ')-IHIP for some negligible δ', ϵ'.*

The above corollary follows from Theorem 7.5 and Proposition 7.4, and carries over our results from other sections about the implications of IHIP protocols to IHD schemes. Because the protocol format of an IHD is related to IHIP, just with a possibly different output space at the end, we adopt the notations of elements in Section 4.

Proof. Inspired by the proof for perfect-hiding in [FO91], we rely on the fact that both the correctness and the hiding aspects of an execution are statements in PSPACE, and thus can be proven with an appropriate interactive proof protocol following the celebrated $\mathsf{IP} = \mathsf{PSPACE}$ theorem [LFKN92][Sha92]. Intuitively, consider a q-round IHD $\langle \mathsf{P}, \mathsf{V}\rangle$ for a function f, we construct a new protocol $\langle \mathsf{P}', \mathsf{V}'\rangle$ that ensures verifiability and hiding against a malicious prover. $\langle \mathsf{P}', \mathsf{V}'\rangle$ runs $\langle \mathsf{P}, \mathsf{V}\rangle$ in a round-by-round manner. Before V sends a message in each round, P' proves to V' that for any input x, the distribution of V's next message in this round will not reveal much additional information about x. This ensures that hiding against malicious prover. After the execution of $\langle \mathsf{P}, \mathsf{V}\rangle$, V' has the view of V: $(x, r_{\mathsf{V}}, \vec{s})$, which is supposed to achieve correctness if V interacts with honest prover. To enforce verifiability, P' proves to the verifier that for any input $x \in \mathcal{X}_n$, the public view $\vec{s}$ achieves correctness with high probability. The protocol is presented in Figure 7

Verifiable Instance Hiding Delegation: $\langle \mathsf{P}', \mathsf{V}' \rangle$

Parameters:

- Function $f : \mathcal{X} \to \mathcal{Y}$.
- Input length n.
- Leakage function ℓ.

Inputs: An instance $x \in \mathcal{X}_n$.

Outputs: $y \in \mathcal{Y} \cup \{\perp\}$.

Ingredients:

- $\langle \mathsf{P}, \mathsf{V} \rangle$ is a q-round (δ, ϵ)-IHD for R as described.
- Consider the public view of protocol $\langle \mathsf{P}, \mathsf{V} \rangle$ up to ith round $(u_1, y_1, \ldots, u_i, y_i)$, and any input $x \in \mathcal{X}_n$. Define $\beta^x_{(u_1,y_1,\ldots,u_i,y_i)}$ as the set of verifier's randomness consistent with input x and $(u_1, y_1, \ldots, u_i, y_i)$. Denote by $\mathcal{U}_{(u_1,y_1,\ldots,u_i,y_i)}$ the set of the possible next messages of V:

$$\mathcal{U}_{(u_1,y_1,\ldots,u_i,y_i)} = \bigcup_{x \in \mathcal{X}} \{u_{i+1} = \mathsf{V}\left(x, r_{\mathsf{V}}, (u_1, y_1, \ldots, u_i, y_i)\right) \big| r_{\mathsf{V}} \in \beta^x_{(u_1,y_1,\ldots,u_i,y_i)}\}.$$

Protocol:

1. V' samples a randomness $r_{\mathsf{V}} \leftarrow \mathcal{R}_{\mathsf{V}}$.
2. V' generates its first message $u_1^* \leftarrow \mathsf{V}(x, r_{\mathsf{V}}, \phi)$, and sends it to P', and P' responds with the y_1 that is P's response to this message.
3. **For** $i \in [2, q]$:
   (a) P' and V' execute an interactive proof protocol where P' proves the following hiding statement:

$$\text{"For any two inputs } x_1, x_2 \in \mathcal{X}_n \text{ such that } \ell(x_1) = \ell(x_2),$$
$$\sum_{u_i \in \mathcal{U}_{(u_1^*, y_1^*, \ldots, y_i^*)}} \left| \frac{\left|\beta^{x_1}_{(u_1^*, y_1^*, \ldots, u_{i-1}^*, y_{i-1}^*, u_i)}\right|}{\left|\beta^{x_1}_{(u_1^*, y_1^*, \ldots, u_{i-1}^*, y_{i-1}^*)}\right|} - \frac{\left|\beta^{x_2}_{(u_1^*, y_1^*, \ldots, u_{i-1}^*, y_{i-1}^*, u_i)}\right|}{\left|\beta^{x_2}_{(u_1^*, y_1^*, \ldots, u_{i-1}^*, y_{i-1}^*)}\right|} \right| \leq 4\epsilon\text{".}$$

   The above statement is in PSPACE if $\ell, f \in \mathsf{PSPACE}$ and the IP follows from $\mathsf{IP} = \mathsf{PSPACE}$.
   (b) **If** the verifier in the above IP rejects, V' rejects immediately.
   Else
      i. V' computes $u_i^* \leftarrow \mathsf{V}\left(x, r_{\mathsf{V}}, (u_1^*, y_1^*, \ldots, u_{i-1}^*, y_{i-1}^*)\right)$ and sends u_i^* to P'.
      ii. P' computes the response y_i^* according to P and sends it to V'.
4. P' and V' execute an interactive proof protocol where P' proves that:

$$\text{"For any } x \in \mathcal{X}_n\text{: } \Pr_{r_{\mathsf{V}} \leftarrow \beta^x_{\vec{s}}}\left[\mathsf{V}\left(x, r_{\mathsf{V}}, (u_1^*, \ldots, y_q^*)\right) \neq f(x)\right] < \tfrac{1}{2^{n/2}}\text{".}$$

5. **If** the verifier in the above protocol rejects, then V' rejects.
   Else V' outputs $\mathsf{V}\left(x, r_{\mathsf{V}}, (u_1^*, \ldots, y_q^*)\right)$.

Figure 7: Transformation from IHD to VIHD

Round-by-Round Hiding The ϵ-hiding for the first round is ensured by ϵ-hiding of $\langle \mathsf{P}, \mathsf{V} \rangle$ as it only involves V's single message. From the second round, however, malicious prover P'^* may deviate from the behavior of the honest prover P' leading to possible leakage of x if V' sticks to V. When considering only the hiding property, the protocol described in Figure 7 can be viewed as a q-fold sequential composition of one-round ϵ-hiding protocols on the same input x, where only the first ϵ-hiding protocol is determined before the execution. For each $i \in [2, q]$, the ith protocol is defined by a prefix, $(u_1^*, y_1^*, \ldots, y_{i-1}^*)$. The prover P'^* is required to prove to V' that the protocol defined by $(u_1^*, y_1^*, \ldots, y_{i-1}^*)$ remains to be ϵ-hiding before the protocol's execution.

Assume for simplicity first that the IP for proving hiding statements is perfectly complete and sound. That is, if V' does not reject until end of q rounds (in this case all of the hiding statements are true), then conditioned on any prover's view up to $i-1$ round, the next message of V u_i^* will be 2ϵ-hiding. We proceed to use Fact 5.10 to inductively prove that the overall prover's view of the protocol will be $2 \cdot q \cdot \epsilon$-hiding. For the analysis of hiding, we ignore the parts of views between P'^* and V' on the proof of hiding statement because they are independent of the input x conditioned on view of previous rounds. Denote by $U_i^*(x)$ V's message in i^{th} round when the input is x. Similarly, $(R_{\mathsf{P}'^*}, U_1^*, Y_1^*, \ldots, , U_i^*, Y_i^*)(x)$ denotes P's view up to i^{th} round. For any two inputs $x, x' \in \mathcal{X}_n$,

- Base step: It's clear that $\Delta\left((R_{\mathsf{P}'^*}, U_1^*, Y_1^*)(x), (R_{\mathsf{P}'^*}, U_1^*, Y_1^*)(x')\right) \leq 2\epsilon$ by ϵ-hiding of $\langle \mathsf{P}, \mathsf{V} \rangle$.
- Inductive step: For each round $i \in [q]$, suppose $\Delta\left((R_{\mathsf{P}'^*}, U_1^*, \ldots, Y_i^*)(x), (R_{\mathsf{P}'^*}, U_1^*, \ldots, Y_i^*)(x')\right) = \epsilon_i$. For any for any possible $(u_1^*, y_1^*, \ldots, u_i^*, y_i^*)$, conditioned on that V' not rejecting up to $i+1$ rounds, and on the prover's view of previous i^{th} rounds equals to $(r_{\mathsf{P}'^*}, u_1^*, y_1^*, \ldots, u_i^*, y_i^*)$ (i.e. $(R_{\mathsf{P}'}, U_1^*, \ldots, Y_i^*)(x) = (R_{\mathsf{P}'}, U_1^*, \ldots, Y_i^*)(x') = (r_{\mathsf{P}'^*}, u_1^*, \ldots, y_q^*)$), the distance between verifier's $(i+1)^{th}$ message on the two inputs is bounded:

  $$\Delta\left(U_{i+1}^*(x) | (R_{\mathsf{P}'}, U_1^*, \ldots, Y_i^*)(x) = (u_1^*, \ldots, y_q^*), U_{i+1}^*(x') | (U_1^*, \ldots, Y_i^*)(x') = (u_1^*, \ldots, y_q^*)\right) \leq 2\epsilon.$$

  This follows the hiding argument. Thus by Fact 5.10 and data processing inequality,

  $$\Delta\left((R_{\mathsf{P}'}, U_1^*, \ldots, Y_{i+1}^*)(x), (R_{\mathsf{P}'}, U_1^*, \ldots, Y_{i+1}^*)(x')\right) \leq \epsilon_i + 2\epsilon.$$

This concludes that the protocol view is $2 \cdot q(n) \cdot \epsilon(n)$ if the IP for proving hiding statements are perfect complete and sound. In the case where the IP for hiding statement has negl-completeness/soundness error, the base case remain unchanged, and we argue that this error won't affect the inductive argument. Formally, define A_i as the indicator that P' makes V' accept on i^{th}-round hiding statements. For any two inputs $x, x' \in \mathcal{X}_n$, and each round $i \in [q]$, conditioned on any possible view in the first i rounds $(R_{\mathsf{P}'}, U_1^*, \ldots, Y_i^*)(x) = (R_{\mathsf{P}'}, U_1^*, \ldots, Y_i^*)(x') = (r_{\mathsf{P}'^*}, u_1^*, \ldots, y_i^*)$, and let $A_j = 1$ for $j \in [i]$, by Fact 4.7:

- In the case that i^{th}-round hiding statement is true:

  $$\begin{aligned}
  \Delta\left(\left(U_{i+1}^*(x), A_{i+1}\right), \left(U_{j+1}^*(x'), A_{i+1}\right)\right) &= \mathop{\mathbb{E}}_{b \leftarrow A_{i+1}}\left[\Delta\left(\left(U_{i+1}^*(x) | A_{i+1} = b\right), \left(U_{i+1}^*(x') | A_{i+1} = b\right)\right)\right] \\
  &\leq \Delta\left(\left(U_{i+1}^*(x) | A_{i+1} = 1\right), \left(U_{i+1}^*(x') | A_{i+1} = 1\right)\right) + \Pr\left[A_{i+1} = 0\right] \\
  &\leq 2\epsilon(n) + \mathrm{negl}(n) \\
  &\leq \mathrm{negl}(n).
  \end{aligned}$$

- In the case that i^{th}-round hiding statement is false:

  $$\begin{aligned}
  \Delta\left(\left(U_{i+1}^*(x), A_{i+1}\right), \left(U_{j+1}^*(x'), A_{i+1}\right)\right) &= \mathop{\mathbb{E}}_{b \leftarrow A_{i+1}}\left[\Delta\left(\left(U_{i+1}^*(x) | A_{i+1} = b\right), \left(U_{i+1}^*(x') | A_{i+1} = b\right)\right)\right] \\
  &\leq \Delta\left(\left(U_{i+1}^*(x) | A_{i+1} = 0\right), \left(U_{i+1}^*(x') | A_{i+1} = 0\right)\right) + \Pr\left[A_{i+1} = 1\right] \\
  &\leq 0 + \mathrm{negl}(n) \\
  &\leq \mathrm{negl}(n),
  \end{aligned}$$

  where the second last inequality is because V' will reject and set $U_{i+1}^* = \perp$ in both cases if $A_{i+1} = 0$.

Therefore, following the same induction as in the case of perfect completeness and soundness, there exists some negligible ϵ' such that $\langle \mathsf{P}', \mathsf{V}' \rangle$ is ϵ'-hiding against any prover.

Correctness. We say a public view $\vec{s}$ of $\langle \mathsf{P}(n), \mathsf{V}(x) \rangle$ is $\vec{S}$-good if for any $x \in \mathcal{X}_n$: $\Pr_{r_\mathsf{V} \leftarrow \beta^x_{\vec{s}}} [\mathsf{V}(x, r_\mathsf{V}, \vec{s}) \neq f(x)] < \frac{1}{2^{n/2}}$, and by definition for any $x \in \mathcal{X}_n$:

$$\Pr_{(r_\mathsf{V}, \vec{s}) \leftarrow \text{VIEW}_\mathsf{V}\left(\mathsf{P}, \mathsf{V}(x)\right)} \left[\mathsf{V}(x, r_\mathsf{V}, \vec{s}) = f(x) | \vec{s} \in \vec{S}\text{-good} \right] \geq 1 - \frac{1}{2^{n/2}}.$$

We will proceed to show that $\vec{s}$ is $\vec{S}$-good with high probability. Similarly to that in [FO91], we assume that $\delta < \frac{1}{2^{2n}}$, which is made through standard amplication procedure with parallel repetition; the proof that parallel repetition works for amplification follows from a Chernoff bound for correctness, and Lemma 2.10 for hiding. Define $\bar{\alpha}^x_{\vec{s}} = \{r_\mathsf{V} \in \beta^x_{\vec{s}} \mid \mathsf{V}(x, r_\mathsf{V}, \vec{s}) \neq f(x)\}$. By δ-correctness and following the same argument as in Claim 4.5.2, for any input instance x:

$$\mathop{\mathbb{E}}_{\vec{s} \leftarrow \text{VIEW}_{pub}\left(\mathsf{P}(n), \mathsf{V}(x)\right)} \left[\frac{|\bar{\alpha}^x_{\vec{s}}|}{|\beta^x_{\vec{s}}|} \right] \leq \delta \leq \frac{1}{2^{2n}}.$$

Follows Markov bound,

$$\Pr_{\vec{s} \leftarrow \text{VIEW}_{pub}\left(\mathsf{P}(n), \mathsf{V}(x)\right)} \left[\Pr_{r_\mathsf{V} \leftarrow \beta^x_{\vec{s}}} [\mathsf{V}(x, r_\mathsf{V}, \vec{s}) = f(x)] > \frac{1}{2^{n/2}} \right] = \Pr_{\vec{s} \leftarrow \text{VIEW}_{pub}\left(\mathsf{P}(n), \mathsf{V}(x)\right)} \left[\frac{|\bar{\alpha}^x_{\vec{s}}|}{|\beta^x_{\vec{s}}|} > \frac{1}{2^{n/2}} \right] < \frac{1}{2^{3n/2}}.$$

Apply a union bound over all possible $x \in \{0,1\}^n$, and we get that:

$$\Pr_{\vec{s} \leftarrow \text{VIEW}_{pub}\left(\mathsf{P}(n), \mathsf{V}(x)\right)} \left[\exists x \in \mathcal{X}_n : \Pr_{r_\mathsf{V} \leftarrow \beta^x_{\vec{s}}} [\mathsf{V}(x, r_\mathsf{V}, \vec{s}) \neq f(x)] > \frac{1}{2^{n/2}} \right] \leq \frac{1}{2^{n/2}}.$$

Thus we have $\Pr_{\vec{s} \leftarrow \text{VIEW}_{pub}\left(\mathsf{P}(n), \mathsf{V}(x)\right)} \left[\vec{s} \text{ is } \vec{S}\text{-good} \right] \geq 1 - \frac{1}{2^{n/2}}$. Moreover, because the statement "$\vec{s}$ is $\vec{S}$-good" is a PSPACE statement if f can be computed in PSAPACE, it can be proved by an interactive proof protocol with completeness and soundness error $1 - \text{negl}(n)$[LFKN92][Sha92]. Therefore, the probability that $\langle \mathsf{P}(n)', \mathsf{V}'(x) \rangle$ outputs correctly is:

$$\begin{aligned}
\Pr\left[\langle \mathsf{P}(n)', \mathsf{V}'(x) \rangle = f(x)\right] &\geq \Pr_{r_\mathsf{V}, r_\mathsf{P}} \left[\langle \mathsf{P}(n)', \mathsf{V}'(x) \rangle = f(x) | \vec{s} \text{ is } \vec{S}\text{-good} \right] \cdot \Pr_{\vec{s} \leftarrow \text{VIEW}_{pub}\left(\mathsf{P}(n), \mathsf{V}(x)\right)} \left[\vec{s} \text{ is } \vec{S}\text{-good} \right] \\
&\geq \left(1 - \frac{1}{2^{n/2}}\right) \cdot \left(1 - \frac{1}{2^{n/2}}\right) \\
&\geq 1 - \text{negl}(n).
\end{aligned}$$

Verifiability. The verifiability relies on the soundness of the interactive proof for the statement "$\vec{s}$ is $\vec{S}$-good". Specifically, if $\vec{s}$ is not $\vec{S}$-good, V' will output $\perp$ with $1 - \text{negl}(n)$ probability. On the other hand, if $\vec{s}$ is $\vec{S}$-good, V' will output $f(x)$ with $1 - \text{negl}(n)$ probability. □

7.2 Closure Properties

In this section, we use some obvious closure properties of instance-hiding delegation schemes, together with the connections to IHIP proven so far, to show similar closure properties for IHIP's. Consider any functions $f : \mathcal{X} \to \mathcal{Y}$ and $p : \mathcal{Y}^* \to \mathcal{Z}$. For any function $k : \mathbb{N} \to \mathbb{N}$, we define the composed function $p \circ f^{\otimes k} : \mathcal{X}^* \to \mathcal{Z}$ as the following partial function for each $n \in \mathbb{N}$:

$$(p \circ f^{\otimes k})(x_1, \ldots, x_{k(n)}) = p\left(f(x_1), \ldots, f(x_{k(n)})\right).$$

where each x_i is of size n.

Proposition 7.7 (Closure of VIHD under Composition with Efficient Functions)**.** *Consider any function f that has a (δ, ϵ)-VIHD for some negligible functions δ and ϵ, and any efficiently computable function p. For any polynomial $k : \mathbb{N} \to \mathbb{N}$, the composed function $p \circ f^{\otimes k}$ also has a (δ', ϵ')-VIHD for some negligible functions δ' and ϵ'.*

Proof Sketch. Let $\langle \mathsf{P}, \mathsf{V} \rangle$ be the VIHD protocol for f. Define $\langle \mathsf{P}', \mathsf{V}' \rangle$ as the protocol that runs $\langle \mathsf{P}, \mathsf{V} \rangle$ on each x_i, gets result y_i, and finally outputs $p(y_1, \ldots, y_k)$ if no $\langle \mathsf{P}_i, \mathsf{V}_i \rangle$ outputs $\perp$ (otherwise outputs $\perp$).

Given that $\langle \mathsf{P}, \mathsf{V} \rangle$ is ϵ-hiding, and following Lemma 2.10, $\langle \mathsf{P}', \mathsf{V}' \rangle$ is a $(k \cdot \epsilon)$-hiding protocol. Furthermore, because the k protocols are independent, and each constitutes δ-correctness, the protocol $\langle \mathsf{P}', \mathsf{V}' \rangle$ thus also possesses $k \cdot \delta$-correctness through a union bound. Verifiability of $\langle \mathsf{P}, \mathsf{V} \rangle$ implies that except with probability $k \cdot \delta$, all the values of y_i obtained, if not $\perp$, are correct. This is thus the protocol we want. □

Combining Propositions 7.2, 7.4 and 7.7 and Theorem 7.5, we get the following as corollaries.

Theorem 2.7 (Closure under Composition with Efficient Functions)**.** *Consider any promise problems Π that has an IHIP protocol, and any efficiently computable function $f : \{0, 1, \perp\}^* \to \{0, 1, \perp\}$ whose output is $\perp$ whenever any of its inputs is $\perp$. For any polynomial $k : \mathbb{N} \to \mathbb{N}$, the composed promise problem $f \circ \Pi^{\otimes k}$ also has an IHIP protocol.*

Lemma 2.8 (Closure under Complementation)**.** *Suppose, for some negligible functions δ, ϵ, that a problem Π has a (δ, ϵ)-IHIP (possibly with a non-uniform verifier). Then the complement of Π has a (δ', ϵ')-IHIP (resp. with a non-uniform verifier if starting with a non-uniform verifier), where δ', ϵ' are also negligible.*

Finally, we prove the following propositions regarding closure properties of VIHD schemes in the presence of leakage that will be useful in Appendix C.

Proposition 7.8. *Consider two functions $\ell, \ell' \in$ PSPACE defined over domain $\mathcal{X}$ such that for any two inputs $x, x' \in \mathcal{X}$, $\ell(x) = \ell(x')$ if and only if $\ell'(x) = \ell'(x')$. For functions δ, ϵ and any function $f : \mathcal{X} \to \mathcal{Y}$, if $\langle \mathsf{P}, \mathsf{V} \rangle$ is a (δ, ϵ, ℓ)-IHD (resp. (δ, ϵ, ℓ)-VIHD) for f, then $\langle \mathsf{P}, \mathsf{V} \rangle$ is a $(\delta, \epsilon, \ell')$-IHD (resp. $(\delta, \epsilon, \ell')$-VIHD) for f. Furthermore, if there is an efficient bijective map between ℓ and ℓ', the transformation holds for simulatable IHD (resp. simulatable VIHD).*

Proof Sketch. ℓ and ℓ' are renaming of each other and thus a simulator with respect to ℓ' can be made given a simulator with respect to ℓ. □

Acknowledgements

We thank anonymous reviewers of this paper, and Yunqi Li for their useful comments and references. Both authors are supported by the National Research Foundation, Singapore, under its NRF Fellowship programme, award no. NRF-NRFF14-2022-0010.

References

[AFK89] Martín Abadi, Joan Feigenbaum, and Joe Kilian. On hiding information from an oracle. *Journal of Computer and System Sciences*, 39(1):21–50, 1989.

[AH91] William Aiello and Johan Håstad. Statistical zero-knowledge languages can be recognized in two rounds. *J. Comput. Syst. Sci.*, 42(3):327–345, 1991.

[AIK04] B. Applebaum, Y. Ishai, and E. Kushilevitz. Cryptography in nc/sup 0/. In *45th Annual IEEE Symposium on Foundations of Computer Science*, pages 166–175, 2004.

[AIK05] B. Applebaum, Y. Ishai, and E. Kushilevitz. Computationally private randomizing polynomials and their applications (extended abstract). In *20th Annual IEEE Conference on Computational Complexity (CCC'05)*, pages 260–274, 2005.

[AIK10] Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. From secrecy to soundness: Efficient verification via secure computation. In *International Colloquium on Automata, Languages and Programming*, 2010.

[AIKPC15] Shweta Agrawal, Yuval Ishai, Dakshita Khurana, and Anat Paskin-Cherniavsky. Statistical randomized encodings: A complexity theoretic view. In Magnús M. Halldórsson, Kazuo Iwama, Naoki Kobayashi, and Bettina Speckmann, editors, *Automata, Languages, and Programming*, pages 1–13, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg.

[App07] Benny Applebaum. *Cryptography in constant parallel time*. PhD thesis, Technion - Israel Institute of Technology, ISR, 2007.

[App14a] Benny Applebaum. *Cryptography in Constant Parallel Time*. Information Security and Cryptography. Springer, 2014.

[App14b] Benny Applebaum. *Randomized Encoding of Functions*, pages 19–31. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.

[AR16] Benny Applebaum and Pavel Raykov. On the relationship between statistical zero-knowledge and statistical randomized encodings. In *Advances in Cryptology – EUROCRYPT 2016*, page 449–477, Berlin, Heidelberg, 2016. Springer-Verlag.

[BF90] Donald Beaver and Joan Feigenbaum. Hiding instances in multioracle queries. In Christian Choffrut and Thomas Lengauer, editors, *STACS 90*, pages 37–48, Berlin, Heidelberg, 1990. Springer Berlin Heidelberg.

[BFOS93] Donald Beaver, Joan Feigenbaum, Rafail Ostrovsky, and Victor Shoup. Instance-hiding proof systems. *Work done at Harvard University, supported in part by NSF grant CCR-870-4513*, 1993.

[BFS90] Donald Beaver, Joan Feigenbaum, and Victor Shoup. Hiding instances in zero-knowledge proof systems. In Alfred J. Menezes and Scott A. Vanstone, editors, *Advances in Cryptology – CRYPTO 1990, Proceedings*, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), pages 326–338. Springer Verlag, 1990.

[BGG+88] Michael Ben-Or, Oded Goldreich, Shafi Goldwasser, Johan Håstad, Joe Kilian, Silvio Micali, and Phillip Rogaway. Everything provable is provable in zero-knowledge. In Shafi Goldwasser, editor, *Advances in Cryptology - CRYPTO '88, 8th Annual International Cryptology Conference, Santa Barbara, California, USA, August 21-25, 1988, Proceedings*, volume 403 of *Lecture Notes in Computer Science*, pages 37–56. Springer, 1988.

[BM82] Manuel Blum and Silvio Micali. How to generate cryptographically strong sequences of pseudo random bits. In *23rd Annual Symposium on Foundations of Computer Science (sfcs 1982)*, pages 112–117, 1982.

[BM88] László Babai and Shlomo Moran. Arthur-merlin games: A randomized proof system, and a hierarchy of complexity classes. *Journal of Computer and System Sciences*, 36(2):254–276, 1988.

[BMO90] M. Bellare, S. Micali, and R. Ostrovsky. The (true) complexity of statistical zero knowledge. In *Proceedings of the Twenty-Second Annual ACM Symposium on Theory of Computing*, STOC '90, page 494–502, New York, NY, USA, 1990. Association for Computing Machinery.

[CD96] Ronald Cramer and Ivan Damgård. On monotone function closure of statistical zero-knowledge. *IACR Cryptol. ePrint Arch.*, page 3, 1996.

[CGK18] Henry Corrigan-Gibbs and Dmitry Kogan. The discrete-logarithm problem with preprocessing. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2018*, pages 415–447, Cham, 2018. Springer International Publishing.

[DSDCPY08] Alfredo De Santis, Giovanni Di Crescenzo, Giuseppe Persiano, and Moti Yung. On monotone formula composition of perfect zero-knowledge languages. *SIAM J. Comput.*, 38:1300–1329, 01 2008.

[Elg85] T. Elgamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31(4):469–472, 1985.

[FF91] J. Feigenbaum and L. Fortnow. On the random-self-reducibility of complete sets. In *[1991] Proceedings of the Sixth Annual Structure in Complexity Theory Conference*, pages 124–132, 1991.

[FKN03] Uri Feige, Joe Kilian, and Moni Naor. A minimal model for secure computation. *Conference Proceedings of the Annual ACM Symposium on Theory of Computing*, 01 2003.

[FO91] Joan Feigenbaum and Rafail Ostrovsky. A note on one-prover, instance-hiding zero-knowledge proof systems. In *Advances in Cryptology - ASIACRYPT '91, International Conference on the Theory and Applications of Cryptology, Fujiyoshida, Japan, November 11-14, 1991, Proceedings*, volume 739 of *Lecture Notes in Computer Science*, pages 352–359. Springer, 1991.

[For87] L. Fortnow. The complexity of perfect zero-knowledge. In *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing*, STOC '87, page 204–209, New York, NY, USA, 1987. Association for Computing Machinery.

[GGH+07] Shafi Goldwasser, Dan Gutfreund, Alexander Healy, Tali Kaufman, and Guy N. Rothblum. Verifying and decoding in constant depth. In *Proceedings of the Thirty-Ninth Annual ACM Symposium on Theory of Computing*, STOC '07, page 440–449, New York, NY, USA, 2007. Association for Computing Machinery.

[GK96] Oded Goldreich and Hugo Krawczyk. On the composition of zero-knowledge proof systems. *SIAM Journal on Computing*, 25(1):169–192, 1996.

[GMR85] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof-systems. In *Symposium on the Theory of Computing*, 1985.

[GMR88] Shafi Goldwasser, Silvio Micali, and Ronald L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on Computing*, 17(2):281–308, 1988.

[GMW91] Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity for all languages in NP have zero-knowledge proof systems. *J. ACM*, 38(3):691–729, 1991.

[Gol98] Oded Goldreich. *Modern Cryptography, Probabilistic Proofs and Pseudorandomness*, volume 17 of *Algorithms and Combinatorics*. Springer-Verlag, 1998.

[GS86] S Goldwasser and M Sipser. Private coins versus public coins in interactive proof systems. In *Proceedings of the Eighteenth Annual ACM Symposium on Theory of Computing*, STOC '86, page 59–68, New York, NY, USA, 1986. Association for Computing Machinery.

[IK00] Y. Ishai and E. Kushilevitz. Randomizing polynomials: A new representation with applications to round-efficient secure computation. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*, pages 294–304, 2000.

[IL89] R. Impagliazzo and M. Luby. One-way functions are essential for complexity based cryptography. In *30th Annual Symposium on Foundations of Computer Science*, pages 230–235, 1989.

[Imp95] R. Impagliazzo. A personal view of average-case complexity. In *Proceedings of Structure in Complexity Theory. Tenth Annual IEEE Conference*, pages 134–147, 1995.

[Kil88] Joe Kilian. Founding crytpography on oblivious transfer. In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, STOC '88, page 20–31, New York, NY, USA, 1988. Association for Computing Machinery.

[KRV21] Inbar Kaslasi, Ron D. Rothblum, and Prashant Nalini Vasudevan. Public-coin statistical zero-knowledge batch verification against malicious verifiers. Cryptology ePrint Archive, Paper 2021/233, 2021. https://eprint.iacr.org/2021/233.

[KY18] Ilan Komargodski and Eylon Yogev. On distributional collision resistant hashing. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part II*, volume 10992 of *Lecture Notes in Computer Science*, pages 303–327. Springer, 2018.

[LFKN92] Carsten Lund, Lance Fortnow, Howard Karloff, and Noam Nisan. Algebraic methods for interactive proof systems. *J. ACM*, 39(4):859–868, oct 1992.

[Luk69] Yudell L Luke. *Special Functions and Their Approximations: v. 2*. Academic press, 1969.

[Oka00] Tatsuaki Okamoto. On relationships between statistical zero-knowledge proofs. *Journal of Computer and System Sciences*, 60(1):47–108, 2000.

[Ost91] R. Ostrovsky. One-way functions, hard on average problems, and statistical zero-knowledge proofs. In *[1991] Proceedings of the Sixth Annual Structure in Complexity Theory Conference*, pages 133–138, 1991.

[RV22] Ron D. Rothblum and Prashant Nalini Vasudevan. Collision-resistance from multi-collision-resistance. Cryptology ePrint Archive, Paper 2022/173, 2022. https://eprint.iacr.org/2022/173.

[Sch91] C. P. Schnorr. Efficient signature generation by smart cards. *J. Cryptol.*, 4(3):161–174, jan 1991.

[Sha92] Adi Shamir. IP = PSPACE. *J. ACM*, 39(4):869–877, 1992.

[Sho97] Victor Shoup. Lower bounds for discrete logarithms and related problems. In *Advances in Cryptology — EUROCRYPT '97*, pages 256–266, Berlin, Heidelberg, 1997. Springer Berlin Heidelberg.

[SS01] Douglas R Stinson and Reto Strobl. Provably secure distributed schnorr signatures and a (t, n) threshold scheme for implicit certificates. In *Australasian Conference on Information Security and Privacy*, 2001.

[SV97] A. Sahai and S.P. Vadhan. A complete promise problem for statistical zero-knowledge. In *Proceedings 38th Annual Symposium on Foundations of Computer Science*, pages 448–457, 1997.

[Tha22] Justin Thaler. Proofs, arguments, and zero-knowledge. *Found. Trends Priv. Secur.*, 4(2-4):117–660, 2022.

[TW87] Martin Tompa and Heather Woll. Random self-reducibility and zero knowledge interactive proofs of possession of information. In *28th Annual Symposium on Foundations of Computer Science (sfcs 1987)*, pages 472–482, 1987.

[Vad99] Salil Pravin Vadhan. *A Study of Statistical Zero-Knowledge Proofs.* PhD thesis, Harvard University, USA, 1999. AAI0801528.

[Yao82] Andrew C. Yao. Theory and application of trapdoor functions. In *23rd Annual Symposium on Foundations of Computer Science (sfcs 1982)*, pages 80–91, 1982.

[Yao86] Andrew Chi-Chih Yao. How to generate and exchange secrets. In *27th Annual Symposium on Foundations of Computer Science (sfcs 1986)*, pages 162–167, 1986.

[Yap83] Chee K. Yap. Some consequences of non-uniform conditions on uniform classes. *Theoretical Computer Science*, 26(3):287–300, 1983.

A Examples of Instance-Hiding Proofs

In an instance-hiding interactive proof for a promise problem Π, the prover proves to the verifier that $x \in \textsc{Yes}(\Pi)$ without knowing anything about x. One approach to constructing such proofs is to use random-self-reducibility properties. One class of problems with such properties, based on discrete logarithm problem, is described below.

Definition A.1 (Discrete Log Problem (DLP))**.** Consider any cyclic group $\mathbb{G}$. Given a generator $g \in \mathbb{G}$ and an element $h \in \mathbb{G}$, the Discrete Log Problem is to find an $a \in \mathbb{Z}_{|\mathbb{G}|}$ such that $g^a = h$.

The Discrete Log problem over $\mathbb{Z}_p^*$ is believed to be hard on average for most primes p (see, e.g. [Elg85]).

Example 1: DL on $\mathbb{Z}_p^*$

Definition A.2. For a prime p, a generator $g \in \mathbb{Z}_p^*$, and an arbitrary predicate $f : \mathbb{Z}_{p-1} \to \{0,1\} \cup \{\perp\}$, we define the promise problem $\mathcal{DL}_{p,g}^f$ as follows:

- $\textsc{Yes}(\mathcal{DL}_{p,g}^f) = \{y \mid \exists x \in \mathbb{Z}_{p-1} : g^x = y \wedge f(x) = 1\}$.
- $\textsc{No}(\mathcal{DL}_{p,g}^f) = \{y \mid \exists x \in \mathbb{Z}_{p-1} : g^x = y \wedge f(x) = 0\}$.

We demonstrate that this problem possesses a simple, strong instance-hiding interactive proof, utilizing the random self-reducibility property of the discrete logarithm.

Proposition A.3. *Consider any prime p, generator g of $\mathbb{Z}_p^*$, and function $f : \mathbb{Z}_{p-1} \to \{0,1\} \cup \{\perp\}$ that is computable in* $\text{polylog}(p)$ *time. The promise problem $\mathcal{DL}_{p,g}^f$ has a perfect Simulatable-IHIP where the verifier runs in time* $\text{polylog}(p)$.

There are no known universal constructions of Statistical Randomized Encodings (SRE), or even Statistical Zero Knoweldge (SZK) protocols, for this problem for such arbitrary efficient functions f. In fact, the oracle separation we show in Section 6 suggests that this problem is unlikely to have generic SRE/SZK when defined with arbitrary cyclic groups and unstructured functions f, even given oracle access to f.

Proof. Because DLP possesses random self-reduction, the verifier can efficiently reduce the instance to a random instance and ask the prover to compute the discrete log for it.[20] In the end, verifier verifies the discrete log and reduces the instance back to original, and computes f to decide the promise problem. Specifically, the protocol is as shown in Figure 8.

[20] We remark that the promise problem $\mathcal{DL}_{p,g}^f$ itself may not possess random self-reducibility. Indeed, [TW87] demonstrated that random self-reducible problems have perfect zero-knowledge proof.

IHIP for $\mathcal{DL}^f_{p,g}$: $\langle \mathsf{P}, \mathsf{V}(y)\rangle$

Parameters: Prime p, generator g of $\mathbb{Z}^*_p$.

Ingredient: Function $f : \mathbb{Z}_{p-1} \to \{0,1\} \cup \{\perp\}$.

Protocol:

1. V does:
   (a) Samples a random $r \leftarrow \mathbb{Z}_{p-1}$.
   (b) Randomize the instance using r: $y' \leftarrow y \cdot g^r \mod p$.
2. V sends y' to P.
3. P compute $x' \in \mathbb{Z}_{p-1}$ such that $g^{x'} = y'$.
4. P sends x' to V.
5. V verifies that $g^{x'} = y'$, if not outputs 0.
6. V computes $d \leftarrow x' - r$ and outputs 1 (Accept) if $f(d) = 1$; otherwise outputs 0 (Reject).

Figure 8: Ins-HIP for $\mathcal{DL}^f_{p,g}$

Completeness and Soundness: Given that d serves as an NP witness for $\mathcal{DL}^f_{p,g}$, the completeness and soundness follow immediately.

Hiding:

- **Simulating for the Honest Prover**: We define a simulator Sim_P that samples r uniformly at random from $\mathbb{Z}_{p-1}$ and outputs (g^r, r). This Sim_P is clearly efficient and the simulated distribution is identical to the honest prover's view in the protocol, thus ensuring perfect hiding.
- **Simulating for a Malicious Prover**: In this one-round instance-hiding interactive proof, the prover's P^* is a random process dependent on verifier's message. Because the simulator is unbounded and verifier's message follows the uniform distribution over $\mathbb{Z}^*_p$, $\mathsf{Sim}_{\mathsf{P}^*}$ will sample a uniform $\alpha \leftarrow \mathbb{Z}^*_p$, and output $(\alpha, \mathsf{P}^*(\alpha))$.

□

Example 2: DL2 on Schnorr group The Schnorr group is commonly used in the construction of public-key encryption and signatures schemes (see, e.g.[SS01]). Here we provide an example of problem based on the Schnorr group, for which we show a two-round instance-hiding interactive proof. However, it is not clear whether it has a one-round instance-hiding IP.

Definition A.4 (Schnorr group [Sch91])**.** A Schnorr group is a large prime-order subgroup of $\mathbb{Z}^*_p$, defined by a tuple (p, q, g) where:

- p and q are primes,
- q divides $p - 1$,
- g is a generator of the subgroup $\mathbb{G}$ of $\mathbb{Z}^*_p$ of order q

Definition A.5. Given a Schnorr group $\mathbb{G}$ defined by (p, q, g_1) and any generator g_2 of $\mathbb{Z}^*_q$, we define the promise problem $\mathcal{DL}2^f_{p,q,g_1,g_2}$ with respect to an arbitrary predicate $f : \mathbb{Z}_{q-1} \to \{0,1\} \cup \{\perp\}$ as:

- $\textsc{Yes}(\mathcal{DL}2^f_{p,q,g_1,g_2}) = \{y \in \mathbb{G}\backslash\{1\} \mid \exists x \in \mathbb{Z}_{q-1} : f(x) = 1 \wedge g_1^{g_2^x} = y\}$.

- $\text{No}(\mathcal{DL}2^f_{p,q,g_1,g_2}) = \{y \in \mathbb{G}\backslash\{1\} \mid \exists x \in \mathbb{Z}_{q-1} : f(x) = 0 \wedge g_1^{g_2^x} = y\}$.

Proposition A.6. *Consider any Schnorr group $\mathbb{G}$ defined by (p, q, g_1), generator g_2 of $\mathbb{Z}_q^*$, and function $f : \mathbb{Z}_{q-1} \to \{0, 1\} \cup \{\perp\}$ that is computable in* polylog(p) *time. The promise problem $\mathcal{DL}2^f_{p,q,g_1,g_2}$ has a perfect Simulatable-IHIP where the verifier and the simulator for the honest prover run in time* polylog(p).

Proof of Proposition A.6. The IHIP protocol (as shown in Figure 9) for $\mathcal{DL}2^f_{p,q,g_1,g_2}$, similar to that for $\mathcal{DL}^f_{p,g}$, leverages the random self-reducibility of discrete logarithm in $\mathbb{Z}_p^*$ to compute the value $z \in \mathbb{Z}_q^*$ in an oblivious manner such that $y = g_1^z$. This procedure is then repeated to get $x \in \mathbb{Z}_{q-1}$ such that $z = g_2^x$.

IHIP for $\mathcal{DL}2^f_{p,q,g_1,g_2}$: $\langle \mathsf{P}, \mathsf{V}(y) \rangle$

Parameters: A Schnorr group $\mathbb{G}$ defined by (p, q, g_1), and generator g_2 of $\mathbb{Z}_q^*$.

Ingredient:

- Function $f : \mathbb{Z}_{q-1} \to \{0, 1\} \cup \{\perp\}$.
- A bit $t \in \{0, 1\}$ initialized to be 1 that verifier keeps for delay rejection.

Protocol:

1. V does:
   (a) Samples a random $r_1 \leftarrow \mathbb{Z}_q$.
   (b) Randomize the y using r_1: $y' \leftarrow y \cdot g_1^{r_1} \mod p$.
2. V sends y' to P.
3. P compute $z'^* \in \mathbb{Z}_q$ such that $g^{z'^*} = y'$, and sends z'^* to V.
4. V
   (a) Verifies that $g^{z'^*} = y'$, if not, set $t = 0$. [a]
   (b) Computes $z^* \leftarrow z'^* - r_1$. By promise, $z^* \in \mathbb{Z}_q^*$.
   (c) Samples $r_2 \leftarrow \mathbb{Z}_{q-1}$, and computes $z'' \leftarrow z^* \cdot g_2^{r_2} \mod q$, and sends z'' to P.
5. P finds $x'^* \in \mathbb{Z}_{q-1}$ such that $g_2^{x'^*} = z''$ and sends x'^* to V.
6. V
   (a) Checks $g_2^{x'^*} = z''$, if not outputs 0 (Reject).
   (b) Computes $x^* \leftarrow x'^* - r_2 \mod q - 1$.
   (c) If $f(x') = 1$ and $t = 1$, outputs 1 (Accept), otherwise outputs 0 (Reject).

[a] Verifier will reject at the final stage even if the prover fails at an earlier stage to ensure hiding.

Figure 9: Ins-HIP for $\mathcal{DL}2^f_{p,q,g_1,g_2}$

Completeness and Soundness: Given that x^* is an NP witness for $\mathcal{DL}2^f_{p,q,g_1,g_2}$, the completeness and soundness follow immediately.

Hiding:

- **Simulating for the Honest Prover**: We define the efficient simulator Sim_P for honest prover as follows:

$\mathsf{Sim}_{\mathsf{P}}(n)$:

1. $a \leftarrow \mathbb{Z}_q$.
2. $b \leftarrow g_1^a \mod p$.
3. Samples $c \leftarrow \mathbb{Z}_{q-1}$.
4. Computes $b' \leftarrow g_2^c \mod q$.
5. Outputs (b, a, b', c).

$\mathsf{Sim}_{\mathsf{P}}$ perfectly simulates the public view because:

- $z'^* = (g_2^x + r_1 \mod q)$ in Figure 9 is uniformly distributed over $\mathbb{Z}_q$.
- $x'^* = (x^* + r_2 \mod q - 1)$ in Figure 9 is uniformly distributed over $\mathbb{Z}_{q-1}$.

Hence the output distribution of $\mathsf{Sim}_{\mathsf{P}}$ (a, b, b', c) is identical to the public view of the protocol Figure 9 (y', z'^*, z'', x'^*).

- **Simulating for a Malicious Prover**: Similarly, the two messages from V are independently uniformly distributed over $\mathbb{G}$ and $\mathbb{Z}_q^*$ respectively. For any malivious prover P^*, $\mathsf{Sim}_{\mathsf{P}^*}$ will simulate identical public view distribution as follows:

$\mathsf{Sim}_{\mathsf{P}^*}(n)$:

1. $a \leftarrow \mathbb{Z}_q$.
2. $b \leftarrow g_1^a \mod p$.
3. Samples P^* randomness: $r_{\mathsf{P}^*} \leftarrow \mathcal{R}_{\mathsf{P}^*}$
4. Runs $a^* \leftarrow \mathsf{P}^*(b; r_{\mathsf{P}^*})$.
5. Samples $c \leftarrow \mathbb{Z}_{q-1}$.
6. Computes $b' \leftarrow g_2^c \mod q$.
7. Runs $c^* \leftarrow \mathsf{P}^*(b, a^*, b')$
8. Outputs $(r_{\mathsf{P}^*}, b, a^*, b', c^*)$.

□

Example 3: IHD from SRE. As noted in [AIK05, AIK10], an ϵ-private, δ-correct interactive RE/Input-hiding for the function $g(x, r) = f(x) \oplus r$ is also an IHD scheme for f with the same parameters and essentially the same complexity.

B Parallel Repetition of Interactive Proofs

Lemma B.1 (See also [Gol98, Appendix C.]). *Consider an interactive proof $\langle \mathsf{P}, \mathsf{V} \rangle$ for a promise problem Π with completeness and soundness errors at most $\delta \leq (1/2 - \Omega(1))$, where the verifier V's input might possibly be private. Let $\langle \mathsf{P}^{\otimes k}, \mathsf{V}^{\otimes k} \rangle$ be the interactive proof obtained by running k independent copies of $\langle \mathsf{P}, \mathsf{V} \rangle$ on the same input in parallel and $\mathsf{V}^{\otimes k}$ outputting the majority of the outputs. Then $\langle \mathsf{P}^{\otimes k}, \mathsf{V}^{\otimes k} \rangle$ is an interactive proof for Π with completeness and soundness errors at most $2^{-\Omega(k)}$.*

Proof. Denote the original IP for promise problem $\Pi = (\textsc{Yes}, \textsc{No})$ by $\langle \mathsf{P}, \mathsf{V} \rangle$, and the IP after parallel repetition by $(\mathsf{P}^{\otimes k}, \mathsf{V}^{\otimes k})$. Suppose $\langle \mathsf{P}, \mathsf{V} \rangle$ has t rounds of interactions (that is, $2t$ messages), without loss of generality with the verifier speaking first. We will borrow the notations of elements on interative proof from Section 4. The completeness of $(\mathsf{P}^{\otimes k}, \mathsf{V}^{\otimes k})$ follows immediately from the completeness of $\langle \mathsf{P}, \mathsf{V} \rangle$ using a Chernoff bound. So we concentrate on proving its soundness.

Fix any input $x \notin \textsc{No}_n$. We define the following function that keeps track of the greatest probability of the verifier accepting starting from any point in the protocol. For any $i \in [0, t]$, any $u_1, \ldots, u_i \in \mathcal{U}$ and $y_1, \ldots, y_i \in \mathcal{Y}$:

$$p(u_1, y_1, \ldots, u_i, y_i) = \max_{\mathsf{P}^*} \Pr\left[\langle \mathsf{P}^*, \mathsf{V} \rangle \text{ accepts} \mid \text{the first } 2i \text{ messages were } (u_1, y_1, \ldots, u_i, y_i)\right]$$

Note that it is sufficient to take the max above over deterministic prover strategies P^*, as given any randomized prover strategy, we can simply fix the random string that maximises the acceptance probability for the given input.[21] Thus, the above probability is only over the remaining random choices $(u_{i+1}, \ldots, u_t)$ of the verifier. Using the notation $\bar{u}_i = (u_i^1, \ldots, u_i^k)$, we similarly define the following function for the repeated protocol:

$$p'(\bar{u}_1, \bar{y}_1, \ldots, \bar{u}_i, \bar{y}_i) = \max_{\mathsf{P}^*} \Pr\left[\left\langle \mathsf{P}^{\otimes k*}, \mathsf{V}^{\otimes k} \right\rangle \text{ accepts} \mid \text{the first } 2i \text{ messages were } (\bar{u}_1, \bar{y}_1, \ldots, \bar{u}_i, \bar{y}_i)\right]$$

The soundness guarantee of $\langle \mathsf{P}, \mathsf{V} \rangle$ may be interpreted as the statement $p(\phi) \leq \delta$ (where ϕ represents the empty string). Our objective will be to similarly prove that $p'(\phi) \leq 2^{-\Omega(k)}$. Define the following function $M : \{0,1\}^k \to \{0,1\}$ that computes the probability that the sum of k independent Bernoulli random variables, with parameters $q_1, \ldots, q_k$, is at least $k/2$:

$$M(q_1, \ldots, q_k) = \sum_{\bar{b} \in \{0,1\}^k : |\bar{b}| \geq k/2} \prod_{j \in [k]} \left(b_j \cdot q_j + (1 - b_j) \cdot (1 - q_j)\right).$$

We will use the following property of M that follows from the fact that it is a multilinear polynomial in the q_j's. For any set of distributions $U_1, \ldots, U_k$ over $\mathcal{U}$ and any set of functions $g_1, \ldots, g_k$,

$$\begin{aligned}
\mathop{\mathbb{E}}_{(u_1,\ldots,u_k) \leftarrow U_1 \times \cdots \times U_k} \left[M(g_1(u_1), \ldots, g_t(u_k))\right] &= \sum_{|\bar{b}| \geq k/2} \mathop{\mathbb{E}}_{(u_1,\ldots,u_k)} \left[\prod_{j \in [k]} \left(b_j \cdot g(u_j) + (1 - b_j) \cdot (1 - g(u_j))\right)\right] \\
&= \sum_{|\bar{b}| \geq k/2} \prod_{j \in [k]} \mathop{\mathbb{E}}_{u_j} \left[\left(b_j \cdot g(u_j) + (1 - b_j) \cdot (1 - g(u_j))\right)\right] \\
&= M\left(\mathop{\mathbb{E}}_{u_1}\left[g(u_1)\right], \ldots, \mathop{\mathbb{E}}_{u_k}\left[g(u_k)\right]\right). \qquad (12)
\end{aligned}$$

where the first and third equalities are from linearity of expectation, and the second follows from the fact that the u_j's are being sampled independently of each other. We have similar behavior with the max operator as well because $M(q_1, \ldots, q_k)$ is non-decreasing with each of the q_j's:

$$\max_{(y_1,\ldots,y_k) \in \mathcal{Y}^k} M(g_1(y_1), \ldots, g_k(y_k)) = M\left(\max_{y_1} g(y_1), \ldots, \max_{y_k} g(y_k)\right). \qquad (13)$$

Recall that our objective is be to prove that $p'(\phi) \leq 2^{-\Omega(k)}$. We will prove the following claim, which implies this statement (through a Chernoff bound) when the i in it is set to 0 as long as the original protocol has soundness error δ less than $(1/2 - \Omega(1))$.

[21] In the case where the prover does not take instance as input (e.g. IHIP), a universal deterministic optimal prover that is optimal for all intances may not exist. But for one particular instance x, an deterministic optimal strategy always exists (see, also [GS86, Section 4.2]).

Claim B.1.1. *For each $i \in [0,t]$, and any $\bar{u}_1, \ldots, \bar{u}_i \in \mathcal{U}^k$ and $\bar{y}_1, \ldots, \bar{y}_i \in \mathcal{Y}^k$,*

$$p'(\bar{u}_1, \ldots, \bar{y}_i) = M\left(p(u_1^1, \ldots, y_i^1), \ldots, p(u_1^k, \ldots, y_i^k)\right).$$

We prove this claim by induction. First, note that for $i = t$, $p'(\bar{u}_1, \ldots, \bar{y}_i)$ is either 1 or 0 depending on whether a majority of the complete proofs $(u_1^j, \ldots, y_t^j)$ are accepted by V. In other words, it is 1 if a majority of $p(u_1^j, \ldots, y_t^j)$ are 1, and 0 otherwise. This proves the base case $i = t$.

Denote by $U_{(u_1,\ldots,y_i)}$ (similarly, $\bar{U}_{(\bar{u}_1,\ldots,\bar{y}_i)}$) the distribution of verifier's message conditioned on the first $2i$ messages in the protocol being $(u_1, y_1, \ldots, u_i, y_i)$. To perform our induction, we start by noting the following properties of p that follow by definition.

Claim B.1.2. *For $i \in [0, t-1]$, and any $u_1, \ldots, u_i \in \mathcal{U}$ and $y_1, \ldots, y_i \in \mathcal{Y}$,*

$$p(u_1, \ldots, y_i) = \mathop{\mathbb{E}}_{u_{i+1} \leftarrow U_{(u_1,\ldots,y_i)}} \left[\max_{y_{i+1} \in \mathcal{Y}} p(u_1, \ldots, y_i, u_{i+1}, y_{i+1})\right].$$

Proof of Claim B.1.2. The claim follows immediately by the manner in which an IP operates. Given that $u_1, \ldots, y_i$ are the first $2i$ messages, the next message u_{i+1} is sampled from $\mathcal{U}$ by the verifier according to the distribution $U_{(u_1,\ldots,y_i)}$. Thus, $p(u_1, \ldots, y_i)$ is the expectation over such u_{i+1} of the maximum probability that the verifier can be made to accept given u_{i+1} was its next message. This is precisely the expression in the right-hand side. □

By identical arguments, we also have the following analogous relation for p' for any $i \in [0, t-1]$ and any $\bar{u}_1, \ldots, \bar{u}_i \in \mathcal{U}^k$ and $\bar{y}_1, \ldots, \bar{y}_i \in \mathcal{Y}^k$:

$$p'(\bar{u}_1, \ldots, \bar{y}_i) = \mathop{\mathbb{E}}_{\bar{u}_{i+1} \leftarrow \bar{U}_{(\bar{u}_1,\ldots,\bar{y}_i)}} \left[\max_{\bar{y}_{i+1} \in \mathcal{Y}^k} p'(\bar{u}_1, \ldots, \bar{y}_i, \bar{u}_{i+1}, \bar{y}_{i+1})\right]. \tag{14}$$

Now, suppose Claim B.1.1 was true for some $i^* \in [0,t]$. Then, we show how to prove it for $i^* - 1$, which would complete the inductive argument. We do this by writing (14) as follows:

$$\begin{aligned}
p'(\bar{u}_1, \ldots, \bar{y}_{i*-1}) &= \mathop{\mathbb{E}}_{\bar{u}_{i^*} \leftarrow \bar{U}_{(\bar{u}_1,\ldots,\bar{y}_{i^*-1})}} \left[\max_{\bar{y}_{i^*} \in \mathcal{Y}^k} p'(\bar{u}_1, \ldots, \bar{y}_{i^*-1}, \bar{u}_{i^*}, \bar{y}_{i^*})\right] \\
&= \mathop{\mathbb{E}}_{\bar{u}_{i^*} \leftarrow \bar{U}_{(\bar{u}_1,\ldots,\bar{y}_{i^*-1})}} \left[\max_{\bar{y}_{i^*} \in \mathcal{Y}^k} M\left(\ldots, p(u_1^j, \ldots, y_{i^*-1}^j, u_{i^*}^j, y_{i^*}^j), \ldots\right)\right] \\
&= \mathop{\mathbb{E}}_{u_{i^*}^1 \leftarrow U_{(u_1^1,\ldots,y_{i^*-1}^1)}, \ldots, u_{i^*}^k \leftarrow U_{(u_1^k,\ldots,y_{i^*-1}^k)}} \left[M\left(\ldots, \max_{y_{i^*}^j \in \mathcal{Y}} p(u_1^j, \ldots, y_{i^*-1}^j, u_{i^*}^j, y_{i^*}^j), \ldots\right)\right] \\
&= M\left(\ldots, \mathop{\mathbb{E}}_{u_{i^*}^j \leftarrow U_{(u_1^j,\ldots,y_{i^*-1}^j)}} \left[\max_{y_{i^*}^j \in \mathcal{Y}} p(u_1^j, \ldots, y_{i^*-1}^j, u_{i^*}^j, y_{i^*}^j)\right], \ldots\right) \\
&= M\left(\ldots, p(u_1^j, \ldots, y_{i^*-1}^j), \ldots\right).
\end{aligned}$$

where the second equality follows from the induction hypothesis, the third from (13), and the fourth from (12). The final inequality follows from Claim B.1.2. This proves Claim B.1.1 by induction, and thus the requisite bound on the soundness error of $\langle \mathsf{P}^{\otimes k}, \mathsf{V}^{\otimes k} \rangle$.

□

C Randomized Encodings and Input-Hiding IP

In this section, we study proof systems with hiding properties slightly weaker than those of instance-hiding interactive proofs. In these *input-hiding* interactive proofs, the prover is allowed to learn the whether the

instance is a YES or NO instance, but nothing else about it. We study the relationship between such proofs, IHIPs, and Statistical Randomized Encodings.

Definition C.1 (Input-Hiding Interactive Proof (Inp-HIP))**.** Consider a promise problem $\Pi = (\text{YES}, \text{NO})$, and functions $\delta, \epsilon : \mathbb{N} \to [0,1]$. A (δ, ϵ)-*Input-Hiding Interactive Proof* (Inp-HIP) for Π is a protocol $\langle \mathsf{P}, \mathsf{V} \rangle$ in which a probabilistic polynomial-time verifier V interacts with a computationally unbounded prover P. For some $n \in \mathbb{N}$, the verifier gets a *private* input $x \in \text{YES}_n \cup \text{NO}_n$, while the prover only gets the input length n. At the end of the interaction, V outputs either 1 (Accept) or 0 (Reject). The protocol is required to satisfy the following properties for all large enough $n \in \mathbb{N}$:

- **Completeness:** For any input $x \in \text{YES}_n$:

$$\Pr\left[\langle \mathsf{P}(n), \mathsf{V}(x) \rangle = 1\right] \geq 1 - \delta(n).$$

- **Soundness:** For any input $x \in \text{NO}_n$, and any prover P^*:

$$\Pr\left[\langle \mathsf{P}^*(n), \mathsf{V}(x) \rangle = 1\right] \leq \delta(n).$$

- **Input-Hiding**: For any prover P^*, there exists a computationally unbounded randomized algorithm $\mathsf{Sim}_{\mathsf{P}^*}$, called a simulator, such that for any input $x \in \text{YES}_n \cup \text{NO}_n$,

$$\Delta\Big(\mathsf{Sim}_{\mathsf{P}^*}\left(n, \Pi(x)\right), \text{VIEW}_{\mathsf{P}^*}\left(\mathsf{P}^*(n), \mathsf{V}(x)\right)\Big) < \epsilon(n).$$

If the simulator corresponding to the *honest* prover runs in polynomial time in n, we say the protocol is *Strong-Input-Hiding* (Strong-Inp-HIP). The protocol is *perfectly-hiding* Inp-HIP if $\epsilon(n) = 0$ for all n. If a simulator is only guaranteed to exist only for the honest prover P, the protocol is *honest-prover* Inp-HIP.

Definition C.2 (Class Inp-HIP, Inp-HIP/Poly)**.** The class Inp-HIP consists of all promise problems that have a (δ, ϵ)-Inp-HIP with uniform verifier protocol for some negligible $\delta(n)$ and $\epsilon(n)$. For concrete functions (δ, ϵ), we denote by (δ, ϵ)-Inp-HIP the class of problems possessing (δ, ϵ)-Inp-HIP. Similarly, Inp-HIP/Poly denotes the class of promise problem that have a (δ, ϵ)-Inp-HIP with non-uniform verifier protocol for some negligible $\delta(n)$ and $\epsilon(n)$.

Lemma C.3 (Parallel Amplification for Inp-HIP)**.** *Consider a promise problem $\Pi \in$ Inp-HIP, and suppose $\langle \mathsf{P}, \mathsf{V} \rangle$ is a (δ, ϵ)-Inp-HIP for Π. Let $k(\cdot)$ be such that $k < 1/\epsilon$, and define $\langle \mathsf{P}_k, \mathsf{V}_k \rangle$ as the protocol that, given input x, execute $\langle \mathsf{P}, \mathsf{V} \rangle$ in parallel $k(n)$ times on input x and outputs the majority of the results. Then $\langle \mathsf{P}_k, \mathsf{V}_k \rangle$ is a $(min(\delta, 2^{-\Omega(k)}), k \cdot \epsilon)$-Inp-HIP for Π.*

Proof Sketch. The proof closely mirrors that of Lemma 2.11 except that the canonical instances $x_1, \ldots, x_k$ are of the same output of Π (i.e. $\Pi(x_i) = \Pi(x)$). □

Proposition C.4 (Inp-HIP/Poly=co-Inp-HIP/Poly)**.** *If a problem Π has a (δ, ϵ)-Inp-HIP protocol (possibly with a non-uniform verifier), where δ and ϵ are negligible functions, then the complement of Π also possesses a (δ', ϵ')-Inp-HIP protocol (resp. with a non-uniform verifier if starting with a non-uniform verifier), for some negligible functions δ' and ϵ'.*

Proof Sketch of Proposition C.4.

Claim C.4.1. *Consider a promise problem Π with characteristic function $f : \text{YES} \cup \text{NO} \to \{0,1\}$, and let $\langle \mathsf{P}, \mathsf{V} \rangle$ be a (δ, ϵ)-Inp-HIP for Π and $\ell = f$, then $\langle \mathsf{P}, \mathsf{V} \rangle$ is a (ϵ, δ, ℓ)-IHD for f.*

Claim C.4.2. *Let f be the characteristic function for the promise problem Π and $\ell = f$. If $\langle \mathsf{P}, \mathsf{V} \rangle$ is a (δ, ϵ, ℓ)-verifiable instance-hiding delegation (resp.* strong *(δ, ϵ, ℓ)-verifiable instance-hiding delegation) for f, then there exists a (δ, ϵ)-input-hiding proof (resp. (δ, ϵ, ℓ)-strong input-hiding interactive proof) $\langle \mathsf{P}', \mathsf{V}' \rangle$ for Π.*

Proof Sketch. The proof of Claim C.4.1 and Claim C.4.2 mirror the proof of Proposition 7.2 and Proposition 7.4 respectively. □

Combining Claim C.4.1, Claim C.4.2, Proposition 7.7, and Proposition 7.8, the proposition follows. □

C.1 IHIP/Inp-HIP from SRE

Input-hiding IPs are closely related to the notion of Randomized Encodings (RE) of promise problems [AIK04], which has been implicitly studied in the context of secure multiparty computation [Yao86, Kil88, FKN03], and has subsequently been explicitly explored as a cryptographic primitive [IK00, AIK04, AIKPC15]. In fact, Inp-HIP can be shown to be equivalent to an interactive version of RE for promise problems as defined by Applebaum et al. [AIK10]. We will show how to use a Statistical Randomized Encoding (SRE) for a problem to construct Inp-HIP for it. An oracle seperation of IHIP from SRE follows Theorem 6.4 because $\mathsf{SRE} \subseteq \mathsf{SZK}$[App14b].

Given a function f and an input x, the Randomized Encoding reveals the value $f(x)$ without revealing anything else about x. The proof methodology used to prove Theorem C.7 can be broadened to demonstrate that input-hiding IPs are in fact equivalent to an interactive variant of Randomized Encodings, as defined by Applebaum et al. [AIK10].

Definition C.5 (Statistical Randomized Encodings [IK00, AIK04, AIKPC15])**.** We say that an efficient randomized algorithm Enc is an *ϵ-private and δ-correct Statistical Randomized Encoding* of a promise problem $\Pi = (\textsc{Yes}, \textsc{No})$ (abbreviated (δ, ϵ)-SRE), if the following holds:

- **ϵ-privacy :** There exists an efficient simulator Sim such that:
  - For any yes-instance $x_{\text{yes}} \in \textsc{Yes}_n$:
$$\Delta\big(\mathsf{Sim}(n,1), \mathsf{Enc}(x_{\text{yes}})\big) \leq \epsilon(n).$$
  - For every no-instance $x_{no} \in \textsc{No}_n$:
$$\Delta\big(\mathsf{Sim}(n,0), \mathsf{Enc}(x_{no})\big) \leq \epsilon(n).$$
- **δ-correctness:** There exists a computationally unbounded decoder Dec, such that for every instance $x \in \textsc{Yes}_n \cup \textsc{No}_n$,
$$\Pr\Big[\mathsf{Dec}\big(\mathsf{Enc}(x)\big) \neq \Pi(x)\Big] \leq \delta(n).$$

If left unspecified, $\epsilon(n)$ and $\delta(n)$ are required to be negligible functions by default.

Definition C.6 (Class SRE [AIKPC15])**.** The class SRE is defined as the set of all promise problems that admit a (δ, ϵ)-SRE for negligible functions $\epsilon(n), \delta(n)$.

Theorem C.7. *Suppose, for some functions δ and ϵ, promise problem Π has a (δ, ϵ)-SRE. Then for any polynomial g, Π also has a (δ', ϵ')-Inp-HIP and* strong *(δ'', ϵ'')-Inp-HIP, where $\delta'' = \delta' = \max\big(\epsilon + O(\frac{1}{g}), g \cdot (\epsilon + \delta)\big)$, $\epsilon' = \epsilon$ and $\epsilon'' = \epsilon + g \cdot \delta$.*

Corollary C.8. $\mathsf{SRE} \subseteq \mathsf{Inp\text{-}HIP}$

Proof of Theorem C.7. For a promise problem Π with SRE $(\mathsf{Enc}, \mathsf{Dec}, \mathsf{Sim}_{SRE})$, given input instance $x \in \textsc{Yes}_n \cup \textsc{No}_n$, the high level idea is to hide $\mathsf{Enc}(x)$ among a sequence of samples from $\mathsf{Sim}_{SRE}(n,1)$ and $\mathsf{Sim}_{SRE}(n,0)$, and ask the prover for to run Dec on them and return the results. The verifier can verify that the correctness of responses to the simulated samples, and accept the prover's assertion regarding $\mathsf{Enc}(x)$ if they are all correct.[22] The Inp-HIP protocol for Π is presented in Figure 10. We now compute its completeness, soundness, and instance-hiding errors in terms of the parameters $\delta = \delta(n)$, $\epsilon = \epsilon(n)$, and g.

[22]This formulation of method is similar to the one from section 1.1 of [AIK04] and section 1.2 of [GGH+07]

Inp-HIP from SRE: $\langle \mathsf{P}(\Pi(x)), \mathsf{V}(x) \rangle$

Parameters: Input length n, g.

Input:

- Prover's Input: A bit $b = \Pi_n(x) = \begin{cases} 1, \text{ if } x \in \textsc{Yes}(\Pi_n). \\ 0, , \text{ if } x \in \textsc{No}(\Pi_n). \end{cases}$.
- Verifier's Input: Private instance x.

Ingredients:

- $(\mathsf{Enc}, \mathsf{Dec}, \mathsf{Sim}_{SRE})$ is a (δ, ϵ)-SRE for $\mathcal{L}$.

Protocol:

1. V does:
   (a) Samples a random index $j \leftarrow [g]$ (to place the instance x).
   (b) Encodes the instance as: $q_j \leftarrow \mathsf{Enc}(x)$.
   (c) **For** $i \in [g] \backslash \{j\}$:
   Sample a random bit $b_i \leftarrow \{0,1\}$,
   If $b_i = 1$
   Simulates a "yes" instance: $q_i \leftarrow \mathsf{Sim}_{SRE}(n, 1)$.
   Else
   Simulates a "no" instance: $q_i \leftarrow \mathsf{Sim}_{SRE}(n, 0)$.
2. V sends the $(q_1, \ldots, q_g)$ to the prover P.
3. P decodes each q_i: $b_i^* \leftarrow \mathsf{Dec}(q_i)$
4. P sends the bits $(b_1^*, \ldots, b_g^*)$ back to the verifier V.
5. V accepts if $b_i^* = b_i$ for all $i \neq j$ and $b_j^* = 1$; otherwise, it rejects.

Figure 10: Simulator for Inp-HIP from SRE

Completeness: Given the definitions of ϵ-privacy and δ-correctness from SRE, it is implied that the probability of the prover finding the correct value $b_j^* = b_j$ for each $j \neq i$ is at least $(1-\epsilon-\delta)$. Consequently, the probability of P determining the correct values of b_i for all $i \in [g]$ is bounded below by $\big(1-\epsilon-\delta\big)^{g-1} \cdot \big(1-\delta\big) \geq 1 - g \cdot \big(\epsilon + \delta\big)$. Thus, the completeness error is at most $g \cdot \big(\epsilon + \delta\big)$.

Soundness: We use Y_0 and Y_1 as the shorthand for distribution $\mathsf{Sim}_{SRE}(n, 0)$ and $\mathsf{Sim}_{SRE}(n, 1)$ respectively. Let *swap* be a randomized function that takes as input a tuple of g elements $\bar{y}$ from the domain of Y_0 (and Y_1). It samples a $j \leftarrow [g]$, swaps the first co-ordinate y_1 of its input with the j^{th} element y_j, and outputs the resulting tuple $(y_j, y_2, \ldots, y_1, y_{j+1}, \ldots, y_g)$. We denote by $\vec{b} = (b_1, \ldots, b_{g-1})$ a set of I.I.D uniform bits. Consider any No instance $x \in \textsc{No}(\Pi_n)$, we can see that the message from verifier in Figure 10 as $D_0 : \mathit{swap}\big(\mathsf{Enc}(x), Y_{b_1}, \ldots, Y_{b_{g-1}}\big)$. Define $m_1 = \sum_{i \in [g-1]} b_i$ and $m_0 = g - 1 - m_1$, as the number of samples from Y_1 and Y_0, respectively, in the particular execution of the sampling process. Then the probability that P^* can cheat and make V accept in the end is bounded by the probability of guessing correctly the j given a sample from D_0. Consider a hybrid distribution $D_1 = \mathit{swap}\big(Y_0, Y_{b_1}, \ldots, Y_{b_{g-1}}\big)$. Given a sample from D_1, the probability of guessing the j chosen during its sampling process correctly is at most $\frac{1}{m_0}$. By the ϵ-privacy

of SRE and data processing inequality we have:

$$\Delta(D_0, D_1) \leq \Delta(\mathsf{Enc}(x), Y_0) \leq \epsilon.$$

Thus, the probability that P^* can make V accept is at most $\left(\frac{1}{m_0} + \epsilon\right)$. Because m_0 is essentially computed as the sum of $(g-1)$ random bits, follows Chernoff bound:

$$\Pr\left[m_0 < \frac{g}{3}\right] \leq e^{-\Omega(g)}.$$

Apply a union bound, we have:

$$\Pr\left[\langle \mathsf{P}^*, \mathsf{V}(x)\rangle = 1\right] \leq e^{-\Omega(g)} + O\left(\frac{1}{g}\right) + \epsilon = \epsilon + O\left(\frac{1}{g}\right),$$

which is the soundness error in the resulting protocol.

ϵ'-Input-hiding: For any prover P^*, the corresponding simulator $\mathsf{Sim}_{\mathsf{P}^*}$ for input-hiding is defined as in Figure 11.

Simulator for Inp-HIP from SRE: $\mathsf{Sim}_{\mathsf{P}^*}\ (n, b = \Pi(x))$

Parameters: input length n, g

Ingredients:

- Prover algorithm P^*
- $(\mathsf{Enc}, \mathsf{Dec}, \mathsf{Sim}_{SRE})$ represents an (δ, ϵ)-SRE for $\mathcal{L}$.

Inputs:

- Input length n
- A bit b, indicating Yes/No outcomes.

Procedure:

1. $j \leftarrow [g]$
2. **If** $b = 0$, $q_j \leftarrow \mathsf{Sim}_{SRE}(n, 0)$, **Else**, $q_j \leftarrow \mathsf{Sim}_{SRE}(n, 1)$
3. **For** $i \in [g] \backslash \{j\}$
   $b_i \leftarrow \{0, 1\}$
   If $b_i = 1$, then $q_i \leftarrow \mathsf{Sim}_{SRE}(n, 1)$.
   Else, $q_i \leftarrow \mathsf{Sim}_{SRE}(n, 0)$.
4. Sample prover's randomness $r_{\mathsf{P}^*} \leftarrow \{0,1\}^{|r_{\mathsf{P}^*}|}$.
5. Compute $(b_1^* \ldots, b_g^*) \leftarrow \mathsf{P}^*(q_1, \ldots, q_g; r_{\mathsf{P}^*})$
6. Output $(r_{\mathsf{P}^*}, q_1, \ldots, q_g, b_1^*, \ldots, b_g^*)$.

Figure 11: Simulator for Inp-HIP from SRE

In this protocol, which involves only two messages, the prover's response is determined solely by the verifier's initial message. Therefore, we can disregard the prover's randomness and focus solely on verifier's message. According to the data processing inequality, this approach is sufficient for proving the quality of the simulation.

Observe that the simulator's behavior mirrors the actual protocol, with one key difference: instead of using an encoding of the input x as the verifier does, the simulator uses a sample from Sim_{SRE}. By the

privacy of the SRE, the distance between $\mathsf{Enc}(x)$ and $\mathsf{Sim}_{SRE}(n, \Pi(x))$ is at most ϵ. Consequently, by the data processing inequality, this also sets a upperbound on the statistical distance between the simulated transcript and the actual protocol.

Strong (ϵ'')-Input-hiding: An efficient honest-prover simulator $\mathsf{Sim}_{\mathsf{P}}$ can be constructed similar to that in Figure 11, except that $\mathsf{Sim}_{\mathsf{P}}$ outputs $(q_1, \ldots, q_g, b_1, \ldots, b_g)$ directly instead of running inefficient P. Due to the correctness error of P, conditioned on any $(q_1, \ldots, q_g)$, the distance between $(b_1, \ldots, b_g)$ and $(b_1^*, \ldots, b_g^*)$ is bounded by $g \cdot \delta$. Therefore, the hiding error of $\mathsf{Sim}_{\mathsf{P}}$ is $\epsilon'' = \epsilon + g \cdot \delta$. The lemma thus follows.

□

Setting Parameters: Following the above lemma, the errors in the protocol are as follow:

- Completeness: $g \cdot (\epsilon + \delta)$.
- Soundness: $\epsilon + O\left(\frac{1}{g}\right)$.
- Input-hiding: ϵ.
- Strong Input-hiding: $\epsilon + g \cdot \delta$

For any inverse polynomial (δ, ϵ), by setting $g = \frac{1}{\sqrt{\epsilon+\delta}}$, we get IHIP with $\sqrt{\epsilon + \delta}$ completeness/soundness error. For negligible (δ, ϵ), by setting g to be large enough polynomial, we get a input-hiding with slightly weaker soundness, which can be amplified through parallel repetition ensured by Lemma C.3.

Theorem C.9. *Suppose, for some negligible functions δ and ϵ, the promise problem Π has a (δ, ϵ)-SRE. Then, for any polynomial $p(\cdot)$, there exists a negligible function δ' such that Π has a* strong *$(\delta', 1/p)$-IHIP.*

Proof of Theorem C.9. Given that a promise problem Π has SRE $(\mathsf{Enc}, \mathsf{Dec}, \mathsf{Sim}_{SRE})$, the IHIP protocol that we construct is the same as the protocol in Figure 10. The completeness and soundness errors are as in the proof of Theorem C.7, and the only difference is the construction of the simulator, which is given in Figure 12. In the setting of IHIP, different from Inp-HIP, the result $\Pi(x)$, is not given to the simulator as an input.

As seen in the proof of Theorem C.7, for an inefficient simulator, we can focus solely on the verifier's first message due to data processing inequality. The analysis of efficient simualtor $\mathsf{Sim}_{\mathsf{P}}$ for honest prover, as described in the footnote of Figure 12, will be provided at the end of the proof. Consider any prover P^*, simulator $\mathsf{Sim}_{\mathsf{P}^*}$ as shown in Figure 11, and $\mathsf{Sim}'_{\mathsf{P}^*}$ in Figure 12. We remark that the output distribution of $\mathsf{Sim}'_{\mathsf{P}^*}(n)$ is an equal convex combination of $\mathsf{Sim}_{\mathsf{P}^*}(n, 0)$ and $\mathsf{Sim}_{\mathsf{P}^*}(n, 1)$. Consider a Yes instance $x_{yes} \in \textsc{Yes}(\Pi_n)$ (arguments for No instances are symmetric). Following the observation about the convex combination, the distance between the simulated transcript and the actual protocol is bounded as:

$$\begin{aligned}
&\Delta\left(\mathsf{Sim}'_{\mathsf{P}^*}(n), \textsc{View}_P(\langle \mathsf{P}^*, \mathsf{V}(x_{yes})\rangle)\right) \\
&\quad \leq \frac{1}{2} \cdot \Delta\left(\mathsf{Sim}_{\mathsf{P}^*}(n, 0), \textsc{View}_P(\langle \mathsf{P}^*, \mathsf{V}(x_{yes})\rangle)\right) + \frac{1}{2} \cdot \Delta\left(\mathsf{Sim}_{\mathsf{P}^*}(n, 1), \textsc{View}_P(\langle \mathsf{P}^*, \mathsf{V}(x_{yes})\rangle)\right). \qquad (15)
\end{aligned}$$

The second term above is bounded in the proof of Theorem C.7 by ϵ, demonstrating input-hiding. We now bound the first term.

Simulator for Ins-HIP from SRE: $\mathsf{Sim}'_{\mathsf{P}^*}(n)$

Parameters: input length n, g

Ingredients:

- Prover algorithm P^*
- $(\mathsf{Enc}, \mathsf{Dec}, \mathsf{Sim}_{SRE})$ represents a (δ, ϵ)-SRE for $\mathcal{L}$.

Inputs:

- Input length n

Procedure:

1. **For** $i \in [g]$

   $b_i \leftarrow \{0,1\}$

   If $b_i = 1$, then $q_i \leftarrow \mathsf{Sim}_{SRE}(n, 1)$.

   Else, $q_i \leftarrow \mathsf{Sim}_{SRE}(n, 0)$.
2. Compute $(b_1^* \ldots, b_g^*) \leftarrow \mathsf{P}^*(q_1, \ldots, q_g)$[a]
3. Output $(q_1, \ldots, q_g, b_1^*, \ldots, b_g^*)$.

[a] Similarly, an efficient $\mathsf{Sim}_{\mathsf{P}}$ will not run P but set $b_i^* = b_i$ directly.

Figure 12: Simulator for Ins-HIP from SRE

We define the symbols m_0, m_1, Y_0, Y_1 as in the proof of Theorem C.7, and recall that m_0 and m_1 follow the binomial distribution $Binomial(\frac{1}{2}, g-1)$ as noted there. We define the following sampling process that captures the protocol and the simulator for different choices of Y:

$D(Y_0, Y_1, Y)$:

1. Sample $m_0 \leftarrow Binomial(\frac{1}{2}, g-1)$
2. $m_1 \leftarrow g - 1 - m_0$
3. $y_0^1, \ldots, y_0^{m_0} \leftarrow Y_0$
4. $y_1^1, \ldots, y_1^{m_1} \leftarrow Y_1$
5. $y \leftarrow Y$
6. Output a random permutation of $(y, \underbrace{y_0^1, \ldots, y_0^{m_0}}_{m_0}, \underbrace{y_1^1, \ldots, y_1^{m_1}}_{m_1})$

For any instance $x_{yes} \in \textsc{Yes}(\Pi)$, define the following hybrid distributions:

- $D_0 : D(Y_0, Y_1, \mathsf{Enc}(x_{yes}))$.
- $D_1 : D(Y_0, Y_1, Y_1)$.
- $D_2 : D(Y_0, Y_1, Y_0)$.

The first term in Eq. (15) is the distance $\Delta(D_0, D_2)$, which we seek to bound. The arguments about input-hiding in the proof of Theorem C.7 give us the following claim.

Claim C.9.1. $\Delta(D_0, D_1) \leq \epsilon$.

We bound the distance between D_1 and D_2 as follows.

Claim C.9.2. $\Delta\big(D_1, D_2\big) \leq \frac{2}{\sqrt{g}}$.

Proof of Claim C.9.2. Define two distributions H_0, H_1:

H_1 :

1. Sample $m_0 \leftarrow \mathit{Binomial}\left(\frac{1}{2}, g-1\right)$
2. Outcome m_0

H_0 :

1. Sample $m_0 \leftarrow \mathit{Binomial}\left(\frac{1}{2}, g-1\right)$
2. Outcome $m_0 + 1$

Furthermore, consider a process W:

$W(m, Y_1, Y_0)$:

1. $m_0 \leftarrow m$
2. $m_1 \leftarrow g - m_0$
3. **For** $i \leftarrow 1$ to m_0 : $y_i^0 \leftarrow Y_0$
4. **For** $i \leftarrow 1$ to m_1 : $y_i^1 \leftarrow Y_1$
5. Output a random permutation of $(\underbrace{y_1^0, \ldots, y_{m_0}^0}_{m_0}, \underbrace{y_1^1, \ldots, y_{m_0}^1}_{m_1})$

We observe that the sampling processes for D_1 and D_2 can be represented as $W(H_1, Y_1, Y_0)$ and $W(H_0, Y_1, Y_0)$, respectively. Let $\mathcal{Y}$ denote the union the support of Y_1 and Y_0. Then, we have:

$$
\begin{aligned}
\Delta(D_1, D_2) &= \Delta\big(W(H_1, Y_1, Y_0), W(H_0, Y_1, Y_0)\big) && (1)\\
&\leq \Delta\big((H_1, W(H_1, Y_1, Y_0)), (H_0, W(H_0, Y_1, Y_0))\big) && (2)\\
&= \Delta(H_1, H_0) && (3)\\
&= \frac{1}{2} \sum_{m \in \{0,1,\ldots,g\}} \left|\Pr\left[H_1 = m\right] - \Pr\left[H_0 = m\right]\right| && (4)\\
&= \frac{1}{2} \cdot \left(\Pr\left[H_1 = 0\right] + \Pr\left[H_0 = g\right] + \sum_{m \in \{1,\ldots,g-1\}} \left|\frac{\binom{g-1}{m}}{2^{g-1}} - \frac{\binom{g-1}{m-1}}{2^{g-1}}\right|\right) && (5)\\
&= \frac{1}{2} \cdot \Big(2 \cdot \frac{1}{2^{g-1}} + 2 \cdot \Big(\sum_{m \in \{1,\ldots,\lfloor (g-1)/2 \rfloor\}} \frac{\binom{g-1}{m} - \binom{g-1}{m-1}}{2^{g-1}}\Big)\Big) && (6)\\
&= \frac{1}{2^{g-1}} \cdot \Big(1 + \Big(\binom{g-1}{\lfloor (g-1)/2 \rfloor} - 1\Big)\Big) && (7)\\
&= \frac{\binom{g-1}{\lfloor (g-1)/2 \rfloor}}{2^{g-1}} && (8)\\
&\leq O\left(\frac{1}{\sqrt{g}}\right). && (9)
\end{aligned}
$$

Where Line (2) follows from the data processing inequality, Line (3) is due to the Fact 4.8, and Line (9) from the value of the central binomial coefficient [See, e.g., Section 2.11 in [Luk69]]. □

Following the triangle inequality, we deduce:

$$\Delta(D_0, D_2) \leq \Delta(D_0, D_1) + \Delta(D_1, D_2) \leq \epsilon + O\left(\frac{1}{\sqrt{g}}\right).$$

This allows us to bound the first term in Eq. (15), showing that the instance-hiding error is at most $\epsilon(n) + O(1/\sqrt{g})$. Setting g such that $\epsilon(n) + O(1/\sqrt{g}) < 1/p$ and the theorem follows.

The efficient simulator Sim_P, as described in the footnote of Figure 12, is similar to that in the proof of Theorem C.7. It will outputs $(q_1, \ldots, q_g, b_1, \ldots, b_g)$ directly instead of running decoders on q_i to get b_i^*. We consider an inefficient simulator Sim'_P for honest prover P as defined in Figure 12. Following the same argument in the the strong-hiding of Theorem C.7, the distance $\Delta\left(\mathsf{Sim}_\mathsf{P}(n), \mathsf{Sim}'_\mathsf{P}(n)\right) \leq g \cdot \delta$. By triangle inequality, $\Delta\left(\mathsf{Sim}_\mathsf{P}(n), \textsc{View}_\mathsf{P}\left(\mathsf{P}, \mathsf{V}(x)\right)\right) \leq \epsilon(n) + O(1/\sqrt{g}) + g \cdot \delta$, which is smaller than $1/p$ for an appropriate polynomial $g \ll p^2$. □

C.2 IHIP from Inp-HIP

An instance-hiding IP is also clearly an input-hiding IP, and thus $\mathsf{IHIP} \subseteq \mathsf{Inp\text{-}HIP}$. We will show that an input-hiding IP can be used to construct an instance-hiding IP for the same problem, with slightly worse hiding. In particular, for any polynomial $p(\cdot)$, it can be proved that that $\mathsf{Inp\text{-}HIP} \Rightarrow (1/p, 1/p)\text{-}\mathsf{IHIP} \Rightarrow (1/p, 1/p)\text{-}\mathsf{Inp\text{-}HIP}$. However, it's important to note that input-hiding proofs do not strongly equate to instance-hiding proofs. Specifically, we currently lack a method to transform an Inp-HIP with negligible hiding error into an IHIP while only incurring an polynomial-times increase in error.

Theorem C.10. *Consider two negligible functions δ, ϵ, and a promise problem Π that possesses a (δ, ϵ)-Inp-HIP, then for any polynomial $p(\cdot)$, there exists a negligible δ' such that Π also has $(\delta', 1/p)$-IHIP. Similarly, if Π is a* strong *(δ, ϵ)-Inp-HIP, then it has an* strong *$(\delta', 1/p)$-IHIP, for the same p and δ'.*

Together with Theorem 4.1, this implies the following.

Corollary C.11. $\mathsf{Inp\text{-}HIP} \subseteq \mathsf{NP/poly} \cap \mathsf{coNP/poly}$.

Combining this with Lemma 3.2, we have the following.

Corollary C.12. *If $\mathsf{NP} \subseteq \mathsf{Inp\text{-}HIP/Poly}$, the polynomial hierarchy collapses to the third level.*

Proof Sketch of Theorem C.10. This proof almost mirrors that of Theorem C.9, with the random variables defined differently but analogously. So we describe the high level approach here and direct the reader to Theorem C.9 to complete the proof.

Starting with a (δ, ϵ)-Inp-HIP denoted as $\langle \mathsf{P}, \mathsf{V} \rangle$, we design a verifier V' which has hardcoded into it a Yes instance $x_1 \in \textsc{Yes}(\Pi)$ and a No instance $x_0 \in \textsc{No}(\Pi)$[23]. At the start of its execution, the verifier generates a random sequence of bits $b_1, \ldots b_{g-1} \leftarrow \{0, 1\}$, and a corresponding sequence $x_{b_1}, \ldots x_{b_{g-1}}$ composed of the canonical instances. Given an input x, it is inserted into a random location in this sequence.

Subsequently, P' and V' execute $\langle \mathsf{P}, \mathsf{V} \rangle$ on each instance in the sequence $x_{b_1}, \ldots, x, \ldots, x_{b_{g-1}}$. V' accepts only if the outcome of the executions are correct on all canonical instances x_{b_i}, and V accepts on x. The completeness and soundness of $\langle \mathsf{P}', \mathsf{V}' \rangle$ follows from those of $\langle \mathsf{P}, \mathsf{V} \rangle$ and standard arguments about composition of interactive proofs.

In order to show instance-hiding, we need to construct a simulator for any prover P^*. Our simulator simply runs the protocol $\langle \mathsf{P}^*, \mathsf{V}' \rangle$, except instead of planting a given instance in the sampled sequence as V' does, it again uses a random canonical instance (x_0 or x_1) in its place. Consider the protocol transcript for a Yes instance x_{yes}. The input-hiding property guarantees that if the canonical instance x_1 was used by the simulator, then the simulated transcript is close to the protocol. We then need to show that even if the simulator uses x_0, the simulated transcript is still close. We do this using the fact that the distributions of the

[23] The canonical instances can be from either non-uniform advice or a solved instance generator (SIG), as discussed in [AIK10], which efficiently samples random instances along their solutions.

number of Yes instances in the sequence $x_{b_1} \ldots, x, \ldots, x_{b_g}$ in the case of x being a Yes or a No instance are close (with distance $\approx 1/\sqrt{g}$)). In essence, the instance x_{yes} is effectively hidden by all the other randomly chosen instances. In case that $\langle \mathsf{P}, \mathsf{V} \rangle$ is a strong (ϵ, δ)-Inp-HIP, a strong (ϵ', δ')-IHIP can be made similarly except that the efficiently simulator $\mathsf{Sim}_{\mathsf{P}}$ will output $(b_1, \ldots, b_g)$ directly as in Theorem C.7. The details are identical to those in the proof of Theorem C.9. The proof there shows that any problem with a Statistical Randomized Encoding also has an Ins-HIP. The only change needed in the proof is to format the output of the simulator appropriately, and use the view of P^* instead of the encoding $\mathsf{Enc}(x)$ used there. □