Languages with Efficient Zero-Knowledge PCPs are in **SZK**

Mohammad Mahmoody* David Xiao†

April 24, 2012

Abstract

A *Zero-Knowledge PCP* (ZK-PCP) is a randomized PCP such that the view of any (perhaps cheating) efficient verifier can be efficiently simulated up to small statistical distance. Kilian, Petrank, and Tardos (STOC '97) constructed ZK-PCPs for all languages in **NEXP**. Ishai, Mahmoody, and Sahai (TCC '12), motivated by cryptographic applications, revisited the possibility of *efficient* ZK-PCPs for all $L \in \mathbf{NP}$ where the PCP is encoded as a polynomial-size circuit that given a query i returns the i^{th} symbol of the PCP. Ishai *et al.* showed that there is no efficient ZK-PCP for **NP** with a *non-adaptive* verifier, who prepares all of its PCP queries before seeing any answers, unless $\mathbf{NP} \subseteq \mathbf{coAM}$ and polynomial-time hierarchy collapses. The question of whether *adaptive* verification can lead to efficient ZK-PCPs for **NP** remained open.

In this work, we resolve this question and show that any language or promise problem with efficient ZK-PCPs must be in **SZK** (the class of promise problems with a statistical zero-knowledge *single prover* proof system). Therefore, no **NP**-complete problem can have an efficient ZK-PCP unless $\mathbf{NP} \subseteq \mathbf{SZK}$ (which also implies $\mathbf{NP} \subseteq \mathbf{coAM}$ and the polynomial-time hierarchy collapses).

We prove our result by reducing any promise problem with an efficient ZK-PCP to two instances of the Conditional Entropy Approximation problem defined and studied by Vadhan (FOCS'04) which is known to be complete for the class **SZK**.

Keywords: Statistical Zero-Knowledge, Probabilistically Checkable Proofs, Conditional Entropy.

*Cornell, mohammad@cs.cornell.edu. Supported in part by NSF Award CCF-0746990, AFOSR Award FA9550-10-1-0093, and DARPA and AFRL under contract FA8750-11-2-0211. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the US government.

†LIAFA, CNRS, Université Paris 7. dxiao@liafa.univ-paris-diderot.fr.

1 Introduction

Since their inception, interactive proofs [GMR89, BM88] have had a transformative effect on theoretical computer science in general and the foundations of cryptography in particular. In an interactive proof for a language L, a computationally bounded randomized verifier V and an all-powerful prover P are given a common input x, and P tries to convince V that $x \in L$. The proof must be *complete*: P successfully convinces V that $x \in L$; as well as *sound*: no cheating prover $\widehat{\mathsf{P}}$ should be able to convince V that $x \in L$ when $x \notin L$. [GMR89] showed that by allowing interaction and probabilistic verification, nontrivial languages outside of **BPP** can be proved while the verifier statistically "learns nothing" beyond the fact that $x \in L$. Thus in eyes of the verifier, the interaction remains "zero-knowledge". Shortly after, [GMW91] extend this fundamental result to *all* of **NP** based on computational assumptions and a computational variant of the notion of zero-knowledge.

The notion of zero-knowledge is formalized using the *simulation paradigm*: for each (possibly cheating) efficient verifier, there is an efficient simulator that generates a verifier view that is indistinguishable from the view the verifier would obtain by honestly interacting with the prover, and therefore anything the verifier could do using a transcript of his interaction with the prover, he could do by using the simulator (without talking to the prover). Throughout this paper by default we mean *statistical* indistinguishability and *statistical* zero knowledge, namely they must hold against any (possibly computationally inefficient) distinguisher. Any discussion about computational indistinguishability will be made explicit.

Motivated by the goal of *unconditional* security, Ben-Or *et al.* [BGKW88] showed that if a verifier V interacts with *multiple* interactive provers (MIPs) $\mathsf{P}_1, \mathsf{P}_2, \ldots$ who may coordinate on a strategy beforehand, but are unable to communicate once the interaction with V starts, then all languages in **NP** can be proved in a (statistical) zero-knowledge way *without* any computational assumption. Fortnow, Rompel, and Sipser [FRS94] showed that, the MIP model is essentially equivalent to having a (perhaps exponentially long) *proof*, whose answers to all possible queries are *fixed* before interaction begins (in contrast to the usual notion of a prover, who may choose to alter his answers based on the queries he has seen so far). Such proof systems are now known as *probabilistically checkable proofs* (PCPs for short) and have found applications throughout theoretical computer science, notably in the areas of hardness of approximation through the celebrated PCP theorem [BFL90, AS98, ALM+98] and communication-efficient interactive proofs [Kil92].

The existence of of ZK proofs for **NP** in the MIP model [BGKW88] and the "equivalence" of MIP and PCP models (as a proof system) raised the following basic question:

Does **NP** *have PCPs that remain zero-knowledge against malicious verifiers?*

The work of [BGKW88] does not resolve this question, because their protocol, when implemented in the PCP model, remains ZK only if cheating verifiers follows the protocol honestly. This highlights an important point: since we have no control over the cheating verifier (except that we assume it is efficient), if the proof is polynomial size then a cheating verifier may read the entire proof and this is not zero knowledge. Therefore, the proof π should be super-polynomially long, and we assume that an efficient (perhaps cheating) verifier $\widehat{\mathsf{V}}$ is only allowed black-box access to the proof. Since $\widehat{\mathsf{V}}$ is polynomially bounded, having black-box access to such a proof π means that $\widehat{\mathsf{V}}$ will be able to query only polynomially many symbols in the proof at will. Thus, by definition, ZK-PCPs are incomparable to standard (statistical) zero knowledge proofs in the single or multi-prover proof systems: **(1)** the zero knowledge property is harder to achieve in the PCP model because the proof

is fixed and there is no control on which queries the verifier chooses to make, **(2)** but the soundness property may be easier to achieve in the PCP model because the soundness is required only against *fixed* cheating proofs (rather than cheating provers who may adaptively manipulate their answers).

Kilian, Petrank, and Tardos [KPT97] were the first to explicitly study the question above and (relying on the previous work of [DFK+92] which in turn relied on the PCP theorem) showed that in fact every language in **NEXP** has a ZK-PCP. Their ZK-PCPs, however, were not *efficient* even when constructed for languages in **NP**, where by an efficient PCP for $L \in \mathbf{NP}$, we mean any PCP π whose answer $\pi(q)$ to any query q can be computed using a polynomial size circuit (which may depend on the common input $x \in L$, a witness w that $x \in L$, and an auxiliary random string r_π). This limitation is inherent in the approach of [KPT97], since in order to be ZK, their PCP requires more entropy than the number of queries made by any cheating verifier.

Motivated by the lack of progress for over 10 years towards giving ZK-PCPs for **NP** that are ZK with respect to all efficient cheating verifiers, Ishai, Mahmoody, and Sahai [IMS12] asked whether this may be inherently impossible. Namely, they asked the following question, which is also the main question studied in this work.

Main Question: *Are there* efficient *ZK-PCPs for* **NP***?*

Ishai *et al.* proved that any language or promise problem L with an efficient ZK-PCP *where the honest verifier's queries are non-adaptive* must satisfy $L \in \mathbf{coAM}$. Therefore, **NP** does not have such efficient ZK-PCPs unless the polynomial-time hierarchy collapses [BHZ87]. Thus, the main question above remained open whether there exist efficient ZK-PCPs for **NP** if we allow the verifier to be adaptive. In this paper we resolve this question in the negative; namely we prove:

Theorem 1.1 (Main Result)**.** *Any promise problem L with an efficient ZK-PCP is in* **SZK***.*

This strengthens the negative result of [IMS12] in two ways: **(1)** we lift the restriction that the verifier be non-adaptive, and **(2)** we can conclude that $L \in \mathbf{SZK}$ which is stronger than $L \in \mathbf{AM} \cap \mathbf{coAM}$, since it is known that $\mathbf{SZK} \subseteq \mathbf{AM} \cap \mathbf{coAM}$ [For89, AH91]. We emphasize that Theorem 1.1 does *not* assume that the simulation is black-box.

Relation to Resettable Zero-Knowledge. The notion of *resettable* zero-knowledge single prover proof systems introduced by Canetti *et al.* [CGGM00] is comparably stronger than the notion of ZK-PCPs. Essentially, a resettable-ZK proof is a ZK-PCP where soundness is required to hold even against adaptive cheating provers who may manipulate their answers based on the queries they see (rather than just fixed cheating proofs). Canetti *et al.* [CGGM00] showed how to obtain *efficient* PCPs that are *computational* zero-knowledge based on computational hardness assumptions. But recall that in this work, the notion of ZK is statistical, and so their result does not resolve our main question.

Recently, Garg *et al.* [GOVW12] showed that *efficient* resettable *statistical* ZK proof systems exist for non-trivial languages (*e.g.* Quadratic Residuosity) based on computational assumptions. Therefore under the same assumptions, these languages also possess efficient ZK-PCPs. Garg *et al.* also showed that assuming the existence of exponentially hard one-way functions, statistical zero-knowledge proof systems can be made resettable. Unfortunately this transformation does not preserve the efficiency of the prover. Therefore, even though by the works of Micciancio, Ong, and Vadhan [MV03, OV08] we know that $\mathbf{SZK} \cap \mathbf{NP}$ has statistical zero-knowledge proofs with an efficient prover, the result of [GOVW12] does not necessarily preserve this efficiency.

Finally note that if one can transform any statistical ZK proof into a resettable statistical ZK proof without losing the efficiency of the prover, then together with our main result of Theorem 1.1 this would imply that the problems with efficient ZK-PCPs are exactly those in $\mathbf{SZK} \cap \mathbf{NP}$.

Relation to Basing Cryptography on Tamper-Proof Hardware. A main motivation of [IMS12] to study the possibility of efficient ZK-PCPs for **NP** comes from a recent line of work on basing cryptography on tamper-proof hardware (*e.g.* [Kat07, MS08, CGS08, GKR08, GIS+10, Kol10, GIMS10]). In this model, the parties can exchange classical bits as well as *hardware tokens* that hide a stateful or stateless *efficient* algorithm. The receiver of a hardware token is only able to use it as a black-box and call it polynomially many inputs. Using *stateless* hardware tokens makes the protocol secure against "resetting" attacks where the receiver of a token is able to reset the state of the token (say, by cutting its power). The work of Goyal *et al.* [GIMS10] focused on the power and limits of stateless tamper-proof hardware tokens in achieving *statistical* security and proved that statistical zero-knowledge for all of **NP** is possible using a single stateless token sent from the prover to the verifier followed by $O(1)$ rounds of classical interaction. A natural question remaining open after the work of [GIMS10] was whether the classical interaction can be eliminated and achieve statistical ZK for **NP** using only a single stateless token. It is easy to see that this question is in fact equivalent to our main question above, and thus our Theorem 1.1 proves that a single (efficient) stateless token is not sufficient for achieving statistical ZK proofs for **NP**.

2 Our Techniques

In this section we describe the ideas and techniques behind the proof of Theorem 1.1 and compare our approach to that of [IMS12]. If L has a ZK-PCP (for now, let us assume for notational simplicity that L is a language; the idea is identical for general promise problems), one naive approach to decide L using its simulator is to run the simulator to obtain a view $\nu = (r, (q_1, a_1), \ldots, (q_m, a_m))$, where r is the random seed of the verifier and the (q_i, a_i) are queries/answers to the ZK-PCP, and accept iff ν is an accepting view. This approach would obtain accepting views if $x \in L$ due to the zero-knowledge property, but there is no guarantee about the case $x \notin L$.

A more promising approach is to "extract" a PCP $\widehat{\pi}$ from the simulator $\textsc{Sim}$ and run V against $\widehat{\pi}$. Using this approach, due to the soundness of the PCP we would obtain a rejecting view if $x \notin L$, but the issue shifts to the case $x \in L$ and ensuring that the extracted PCP is a valid proof on YES instances. Therefore, a goal could be trying to construct $\widehat{\pi}$ in a way that it is "close" to an accepting PCP $\pi \leftarrow \boldsymbol{\pi}_{x,w}$ whenever $x \in L$.

To see at a high level why this may be possible for *efficient* ZK-PCP's, let $\boldsymbol{\pi}_{x,w}$ denote the true distribution of proofs on an instance $x \in L$. Since the ZK-PCP is efficient, each proof $\boldsymbol{\pi}_{x,y}$ is computable by some circuit of polynomial size; let $\eta(n) = \mathrm{poly}(n)$ be the number of bits it takes to describe this circuit. If we look at the whole description of $\boldsymbol{\pi}_{x,w}$ as a random variable, its entropy $\mathrm{H}(\boldsymbol{\pi}_{x,y})$ can be at most η. Now consider $\mathsf{V}^{[\ell]}$, which is the cheating verifier that executes ℓ independent copies of V, all of them accessing the same proof $\pi \leftarrow \boldsymbol{\pi}_{x,w}$. Let $(\nu^1, \ldots, \nu^\ell)$ be the views generated. Since $\mathrm{H}(\boldsymbol{\pi}_{x,w}) \leq \eta$, if we pick $i \leftarrow [\ell]$ then the average entropy in the answers returned to the i^{th} verification ν^i *conditioned* on the views of the first $i-1$ verifications $\nu^1, \ldots, \nu^{i-1}$ is at most η/ℓ, which can be made less than any arbitrarily small polynomial by increasing ℓ. Therefore, we will use the simulator for $\mathsf{V}^{[\ell]}$ to generate views $(\nu^1, \ldots, \nu^\ell)$, pick $i \leftarrow [\ell]$ and look at ν^i conditioned on $(\nu^1, \ldots, \nu^{i-1})$. The extracted proof $\widehat{\pi}$ is defined based on how the queries are

answered in ν^i. On YES instances $\widehat{\pi}$ should have low entropy and therefore behave like a fixed accepting proof (because of the statistical indistinguishability of the simulation). On NO instances, $\widehat{\pi}$ either behaves like a fixed proof and therefore is rejecting (because of soundness), or behaves very different from a fixed proof (which we will be able to detect).

This was the approach used in [IMS12]: they give an **AM** (*i.e.* constant-round public-coin) protocol that allows an efficient verifier to extract $\widehat{\pi}$ using the help of an unbounded prover. We also extract a PCP $\widehat{\pi}$ from the simulator, but our extracted PCP is defined differently from the one in [IMS12] and this difference allows us to also use it differently: we do not use the prover to help us construct the extracted PCP in the **SZK** protocol we give for L, but rather we use $\widehat{\pi}$ *only in the analysis* to show that the **SZK** protocol we give is correct.

2.1 The Approach of [IMS12]

Let $\textsc{Sim}$ be the simulator for $\mathsf{V}^{[\ell]}$. Roughly speaking, [IMS12] defines the PCP $\widehat{\pi}$ based on the simulator as follows.

$$\widehat{\pi}(q) \leftarrow (\mathbf{a}_1^i \mid \mathbf{a}_1^i \text{ answer to query } q_1^i = q \text{ in } \textsc{Sim}(x) \text{ conditioned on } \nu^1, \ldots, \nu^{i-1} \text{ being first } i-1 \text{ views})$$

In other words, we first sample $i \leftarrow [\ell]$ and generate views $\nu^1, \ldots, \nu^{i-1}$ according to the simulator. Then to answer any query q, we run $\textsc{Sim}(x)$ conditioned on getting q as the first query of the i^{th} execution, then we output the answer $\textsc{Sim}(x)$ gives to q. It may not be possible to sample $\widehat{\pi}(q)$ efficiently, but [IMS12] show how to sample $\widehat{\pi}$ through an **AM** protocol, using the following ideas.

Simulating $\widehat{\pi}$ with Help of a Prover. Using old and new constant-round sampling and lower-bound protocols [GS89, For89, AH91, GVW01, HMX10] Ishai *et al.* show an **AM** protocol using an unbounded (but also *untrusted*) prover so that if the prover is honest we get a simulation of the oracle $\widehat{\pi}$, and if he cheats then the verifier catches him. Essentially, in the **AM** protocol, Arthur uses Merlin to help repeatedly *rewind* the simulator back to the first query. This way, we obtain that L and its *complement* are both in **AM**. This approach of [IMS12] is inspired by works of [FF93, BT06, AGGM06] in the context of studing worst-case to average-case reductions in **NP** where an unbounded prover (Merlin) is forced to simulate a (hard to compute) *oracle*.

Relying on Nonadaptivity of V. Note that if the distribution of the first and second queries are statistically far, then by asking some second query q_2 from the oracle $\widehat{\pi}$ we might simply get no answer because it is possible that $\textsc{Sim}(x)$ never generates q_2 as the first query. But if the honest verifier V is nonadaptive, *w.l.o.g.* we can assume that it randomly permutes its queries before asking them and therefore the marginal distribution of all queries will be identical (though perhaps correlated). [IMS12] show that if the PCP $\widehat{\boldsymbol{\pi}}$ (as a random variable) has very low entropy, then this implies that the view of $\mathsf{V}^{\widehat{\boldsymbol{\pi}}}(x)$ is close to a simulator-generated view, and so on YES instances by statistical closeness of the simulator and an honest interaction, $\mathsf{V}^{\widehat{\boldsymbol{\pi}}(x)}$ is accepting. On the other hand, since the proof $\widehat{\boldsymbol{\pi}}$ is generated independently of the final verfication's queries, it holds on NO instances that $\mathsf{V}^{\widehat{\boldsymbol{\pi}}}(x)$ is rejecting because of the soundness of the ZK-PCP.

2.2 Our Approach

We use the same cheating verifier $\mathsf{V}^{[\ell]}$ and its corresponding simulator $\textsc{Sim}$, but our extracted PCP $\widehat{\pi}$ is defined *without* rewinding the simulator back to the first query. Roughly speaking, our oracle

is defined as:

$$\widehat{\pi}(q) \leftarrow \left(\mathbf{a}^i_{\mathbf{j}} \quad | \quad \begin{array}{c} \mathbf{a}^i_{\mathbf{j}} \text{ is the answer to the } \mathbf{j}^{\text{th}} \text{ query } q = \mathbf{q}^i_{\mathbf{j}} \text{ for a random } \mathbf{j} \\ \text{in } \textsc{Sim}(x) \text{ conditioned on } \nu^1, \ldots, \nu^{i-1} \text{ first } i-1 \text{ views} \end{array} \right)$$

Notice that $\widehat{\pi}$ is defined without rewinding back to the first query, and so we do not require the queries to have the same distribution and thus we do not need to assume the verifier to be nonadaptive. Furthermore, the way we use $\widehat{\pi}$ differs from [IMS12] because we do not construct $\widehat{\pi}$ in our **SZK** protocol, but only use its definition in the analysis of our reduction to **SZK**.

To obtain an **SZK** protocol for L, we give a Karp (many-to-one) reduction from L to a problem in **SZK**. More formally, using the simulator, we map each x to three circuits (C_1, C_2, C_3) such that we will be able to verify certain statistical properties about them in **SZK** (the reduction is given in Reduction 4.2). Essentially, for $j \in \{1, 2, 3\}$, C_j runs the simulator for the cheating verifier $\mathsf{V}^{[\ell]}$ on input x to obtain views $(\nu^1, \ldots, \nu^\ell)$. Given these executions, C_j picks a random execution i and verifies some statistical properties about i^{th} execution conditioned on the first $(i-1)$ executions.

Here, we just describe the properties that each circuit checks at a high level, and we defer the formal discussions to Section 4. All of the following are conditioned on the first $i-1$ views.

1. C_1 checks that the simulated randomness of V in the i^{th} execution is close to uniform.
2. C_2 checks that, sampling a random set of queries and answers for the i^{th} execution and picking one query/answer pair at random, that answer has low entropy given the that query.
3. C_3 checks that the i^{th} execution is accepting.

First we argue that the reduction maps YES instances of L to (C_1, C_2, C_3) satisfying all three properties, and NO instances to circuits not satisfying all three properties.

- $(x \in L)$. Since the simulator's output is statistically close to the honest distribution, so the simulated verifier's random coins are also close to uniform, and the first property is satisfied. Also, since the PCP is efficient and thus has entropy at most $\eta = \text{poly}(n)$, it means the average conditioned entropy of the answers to queries in the i^{th} verification is at most η/ℓ which we set to be small, so the second property is also satisfied. (Actually, assuming that in the i^{th} execution the entire set of answers has low entropy given the entire set of queries, it is non-trivial to show that a *random* answer has low entropy given its corresponding query, because the queries may be adaptive and a random query might indirectly reveal information about other queries and answers. Despite that, in Lemma 4.4 we prove this claim even for adaptive verifiers.) Finally, for a YES instance the simulator produces accepting views with high probability, so the third property is also satisfied. This is proved in Section 4.1.
- $(x \notin L)$. It suffices to show that if (C_1, C_2, C_3) satisfy the first two properties, then they do not satisfy the third. If the verifier coins in the simulator's output are close to uniform and there is low entropy in the query-answer pairs, then we can show that the view output by the simulator in the i^{th} execution is statistically close to V^i executing against the oracle $\widehat{\boldsymbol{\pi}}$ defined above. Therefore, the verifier must reject in V^i because of the soundness property of the ZK-PCP. This is proved in Section 4.2.

Finally, we note that the desired properties of the circuits C_1, C_2 can be verified in **SZK** by two reductions to the problem of Conditional Entropy Approximation (see Definition 3.8) which

is known to be **SZK**-complete [Vad06], while C_3 can be verified in $\mathbf{BPP} \subseteq \mathbf{SZK}$. Since **SZK** is closed under conjunction, disjunction, and complement (Lemma 3.6, see also [Vad99]), all three properties can simultaneously be verified in **SZK**.

3 Preliminaries

Basic Terminology and Notation. We use bold letters to denote random variables (*e.g.* $\mathbf{X}$ or $\mathbf{x}$). By $x \leftarrow \mathbf{x}$ we mean that x is sampled according to the distribution of the random variable $\mathbf{x}$. We write $\mathbb{E}_x[\cdot]$ to denote $\mathbb{E}_{x \leftarrow \mathbf{x}}[\cdot]$, where any x appearing inside the expression in the expectation is fixed. For any finite set $\mathcal{S}$, $x \leftarrow \mathcal{S}$ denotes x sampled uniformly from $\mathcal{S}$. $\mathbf{U}_n$ denotes the uniform distribution over $\{0,1\}^n$, and $[n]$ denotes the set $\{1, 2, \ldots, n\}$. For jointly distributed random variables $(\mathbf{x}, \mathbf{y})$, and for a specific value $y \leftarrow \mathbf{y}$, by $(\mathbf{x} \mid y)$ we mean the random variable $\mathbf{x}$ conditioned on $\mathbf{y} = y$. When we say an event occurs with *negligible* probability denoted by $\mathrm{negl}(n)$, we mean it occurs with probability $n^{-\omega(1)}$. We call two random variables $\mathbf{x}, \mathbf{y}$ (or their corresponding distributions) over the support set $\mathcal{S}$ *ϵ-close* if their *statistical distance* $\Delta(\mathbf{x}, \mathbf{y}) = \frac{1}{2} \cdot \sum_{s \in S} |\Pr[\mathbf{x} = s] - \Pr[\mathbf{y} = s]|$ is at most ϵ. By an *ensemble* (of random variables) $\{\mathbf{y}_x\}_{x \in \mathcal{I}}$ we denote a set of random variables indexed by a set $\mathcal{I}$. We call two ensembles $\{\mathbf{y}_x\}_{x \in \mathcal{I}}$ and $\{\mathbf{z}_x\}_{x \in \mathcal{I}}$ with the same index set *statistically close* if $\Delta(\mathbf{y}_x, \mathbf{z}_x) = \mathrm{negl}(|x|)$. We use the terms *efficient* and PPT to refer to any probabilistic polynomial time (perhaps oracle-aided) algorithm. For an oracle π and an (oracle-aided) algorithm V by V^π we refer to an execution of V given access to π and by $\mathrm{View}\langle \mathsf{V}^\pi \rangle$ we refer to the *view* of V in its execution given π which consists of its randomness r and the sequence of its oracle query-answer pairs $[(q_1, a_1), (q_2, a_2), \ldots]$ (having only the oracle answers and r is sufficient to know $\mathrm{View}\langle \mathsf{V}^\pi \rangle$). All logarithms are base 2. By $\mathrm{H}(\mathbf{X})$ we denote the Shannon entropy of $\mathbf{X}$ defined as $\mathrm{H}(\mathbf{X}) = \mathbb{E}_X \lg(1/\Pr[\mathbf{X} = x])$. By $\mathrm{H}(\mathbf{X} \mid \mathbf{Y})$, we denote the conditional entropy as $\mathbb{E}_Y[\mathrm{H}(\mathbf{X} \mid Y)]$, and we note the conditional mutual information as $\mathrm{I}(\mathbf{X}; \mathbf{Y} \mid \mathbf{Z}) = \mathrm{H}(\mathbf{X} \mid \mathbf{Z}) - (\mathbf{X} \mid \mathbf{YZ})$.

3.1 Promise Problems

A language L is simply a *partition* of $\{0,1\}^*$ into L^{Y} and L^{N} (*i.e.* $L^{\mathrm{Y}} \cup L^{\mathrm{N}} = \{0,1\}^*$ and $L^{\mathrm{Y}} \cap L^{\mathrm{N}} = \varnothing$).

A *promise* language (or problem) $L = (L^{\mathrm{Y}}, L^{\mathrm{N}})$ generalizes the notion of a language by only requiring that $L^{\mathrm{Y}} \cap L^{\mathrm{N}} = \varnothing$ (but there could be some $x \in \{0,1\}^* \setminus (L^{\mathrm{Y}} \cup L^{\mathrm{N}})$). For promise problems, we will sometimes use $x \in L$ to denote $x \in L^Y$.

Definition 3.1 (Operations on Promise Languages)**.** We define the following three operations over promise languages.

- The *complement* $\overline{L} = (\overline{L}^{\mathrm{Y}}, \overline{L}^{\mathrm{N}})$ of a promise language $L = (L^{\mathrm{Y}}, L^{\mathrm{N}})$ is another promise language such that $\overline{L}^{\mathrm{Y}} = L^{\mathrm{N}}$ and $\overline{L}^{\mathrm{N}} = L^{\mathrm{Y}}$.
- For two promise languages L_1 and L_2 we define their *conjunction* $L = L_1 \wedge L_2$ as:
  - $x = (x_1, x_2) \in L^{\mathrm{Y}}$ iff $x_1 \in L_1^{\mathrm{Y}}$ and $x_2 \in L_2^{\mathrm{Y}}$,
  - $x = (x_1, x_2) \in L^{\mathrm{N}}$ iff $x_1 \in L_1^{\mathrm{N}}$ or $x_2 \in L_2^{\mathrm{N}}$.
- For two promise languages L_1 and L_2 we define their *disjunction* $L = L_1 \vee L_2$ as:

– $x = (x_1, x_2) \in L^{\mathrm{Y}}$ iff $x_1 \in L_1^{\mathrm{Y}}$ or $x_2 \in L_2^{\mathrm{Y}}$,
– $x = (x_1, x_2) \in L^{\mathrm{N}}$ iff $x_1 \in L_1^{\mathrm{N}}$ and $x_2 \in L_2^{\mathrm{N}}$.

It is easy to see that $L_1 \vee L_2 = \overline{\overline{L_1} \wedge \overline{L_1}}$.

Definition 3.2 (Karp Reduction)**.** A Karp reduction R from a promise problem L_1 to another promise problem L_2 is a deterministic efficient algorithm such that $R(x) \in L_2^{\mathrm{Y}}$ for every $x \in L_1^{\mathrm{Y}}$ and $R(x) \in L_2^{\mathrm{N}}$ for every $x \in L_1^{\mathrm{N}}$.

3.2 Interactive Proof Systems

Definition 3.3 (PCPs)**.** A (randomized) *probabilistically checkable proof* (PCP for short) $\Pi = (\{\boldsymbol{\pi}_{x \in L}\}, \mathsf{V})$ for a promise problem L consists of an ensemble of random variables $\{\boldsymbol{\pi}_x\}$ for $x \in L$ whose values are *oracles* (also called *proofs*) and also a verifier V which is an oracle-aided PPT with randomness r. We require the following properties to hold.

- **Completeness:** For every $x \in L^Y$ and every $\pi \in \mathrm{Supp}(\boldsymbol{\pi}_x)$ it holds that $\Pr_r[\mathsf{V}_r^{\pi}(x) = 1] \geq 2/3$.
- **Soundness:** If $x \in L^N$, then for *every* oracle $\widehat{\pi}$ it holds that $\Pr_r[\mathsf{V}_r^{\widehat{\pi}}(x) = 0] \geq 2/3$.

If the PCP also receives an *auxiliary input* w, the distribution of the oracles might depend on x and w both, denoted as: $\{\boldsymbol{\pi}_{x,w}\}$. We call a PCP for problem $L \in \mathbf{NP}$ *efficient*, if for all $x \in L$ and witnesses w for $x \in L$, and all $x \in \mathrm{Supp}(\boldsymbol{\pi}_{x,w})$, there exists a $\mathrm{poly}(n)$-sized circuit C_π such that for all queries q, $C_\pi(q) = \pi(q)$. Namely, C_π encodes π.

Notice that this definition of efficiency is non-uniform: the distribution of proofs C_π may depend non-uniformly on x, w. This makes our results stronger than if we required C_π to depend uniformly on x, w, since we are proving a negative result.

Definition 3.4. Let $\Pi = (\{\boldsymbol{\pi}_{x \in L, w}\}, \mathsf{V})$ be a PCP for the problem L with some auxiliary input given to the oracle. Π is called *zero-knowledge* (ZK) if for every malicious $\mathrm{poly}(n)$-time verifier $\widehat{\mathsf{V}}$, there exists a *simulator* Sim which runs in (expected) $\mathrm{poly}(n)$-time and the following ensembles are statistically close:

$$\{\mathrm{Sim}(x)\}_{x \in L} \quad , \quad \{\mathrm{View}\langle \widehat{\mathsf{V}}^{\boldsymbol{\pi}_{x,w}}(x) \rangle\}_{x \in L}.$$

Note that $\widehat{\mathsf{V}}$ only has oracle access to $\pi_{x,w}$, the auxiliary input is not given to the simulator and the statistical indistinguishability should hold for large enough x (regardless of the witness w). We call Π *perfect* ZK if the simulator distribution conditioned on not aborting is identically distributed to the honest interaction.

Since we do not need the exact definition of the class **SZK**, here we only describe it at a high level. The definition of **SZK** is indeed very similar to Definition 3.4 with the difference that the soundness holds against *provers* (which can be thought of as *stateful* oracles who could answer new queries depending on the previous queries asked.)

Definition 3.5 (Complexity Class **SZK**)**.** The class **SZK** consists of promise problems which have an interactive proof system with soundness error $\leq 1/3$ and the view of any malicious verifier can be simulated up to $\mathrm{negl}(n)$ statistical error.

Lemma 3.6. *For a constant k, let $L_1, \ldots, L_k$ be a set of promise languages all in* **SZK**, *and let F be a constant-size k-input formula with operations: complement, conjunction, and disjunction as in Definition 3.1. Then $F(L_1, \ldots, L_K) \in$* **SZK**.

Here we give a sketch of the proof for completeness. (See Section 4.5 and Corollary 6.5.1 of [Vad99] for a more general and improved statement than that of Lemma 3.6.)

Proof Sketch. We will use the following:

Theorem 3.7 ([Oka96])**.** *The class* **SZK** *is closed under complement.*

Since k is constant, we just need to prove the claim for formulas which have only a single operation, and then the lemma follows by an induction. Moreover since $L_1 \vee L_2 = \overline{\overline{L_1} \wedge \overline{L_1}}$, we just need to prove the claim for complement and conjunction operations. Theorem 3.7 proves this for the complement. To obtain $L_1 \wedge L_2 \in$ **SZK**, given the input (x_1, x_2), the prover provides an (interactive) **SZK** proof that $x_1 \in L_1^{\mathrm{Y}}$ and then (if the first interaction is accepted), he also provides a **SZK** proof that $x_2 \in L_2^{\mathrm{Y}}$. (More formally, the prover and the verifier start with an *amplified* version of the original protocols with soundness error $< 1/6$, the soundness error of the sequential composition in this case remains $< 1/3$). On the other hand, if either of $x_1 \in L_1^{\mathrm{N}}, x_2 \in L_2^{\mathrm{N}}$ holds, the corresponding interaction rejects with probability at leat $2/3$. □

3.3 Shannon Entropy and Related Computational Problems

Definition 3.8 (Conditional Entropy Approximation)**.** The promise problem CEA_ϵ is defined as follows. Suppose C is a $\mathrm{poly}(n)$-size circuit sampling a joint distribution $(\mathbf{X}, \mathbf{Y})$. Then given (C, r) we have:

- $(\mathbf{X}, \mathbf{Y}, r) \in \mathrm{CEA}_\epsilon^{\mathrm{Y}}$ if $\mathrm{H}(\mathbf{X} \mid \mathbf{Y}) \geq r$.
- $(\mathbf{X}, \mathbf{Y}, r) \in \mathrm{CEA}_\epsilon^{\mathrm{N}}$ if $\mathrm{H}(\mathbf{X} \mid \mathbf{Y}) \leq r - \epsilon$.

Lemma 3.9. *For any $\epsilon > 1/\mathrm{poly}(n)$, $\mathrm{CEA}_\epsilon \in$* **SZK**.

Proof. We give a reduction from CEA_ϵ to CEA, which is known to be **SZK**-complete [Vad06]. The reduction maps

$$(\mathbf{X}, \mathbf{Y}, r) \mapsto ((\mathbf{X}^1, \ldots, \mathbf{X}^{1/\epsilon}), (\mathbf{Y}^1, \ldots, \mathbf{Y}^{1/\epsilon}), r/\epsilon)$$

where for every $i \in [1/\epsilon]$, $(\mathbf{X}_1^i, \mathbf{X}_2^i)$ is sampled identically to $(\mathbf{X}, \mathbf{Y})$ and independently of all other components (*i.e.* by an independent copy of the circuit C). It is easy to see that

$$\mathrm{H}((\mathbf{Y}^1, \ldots, \mathbf{Y}^{1/\epsilon}) \mid (\mathbf{X}^1, \ldots, \mathbf{X}^{1/\epsilon})) = \frac{1}{\epsilon} \cdot \mathrm{H}(\mathbf{Y} \mid \mathbf{X}).$$

□

In our main reduction, we will reduce problems to the following problem in **SZK**:

Definition 3.10 (Conditional Entropy Bound)**.** $\mathrm{CEB}_{\alpha,\beta}$ is the following promise problem where inputs are $\mathrm{poly}(n)$-size circuits C sampling a joint distribution $(\mathbf{X}, \mathbf{Y})$:

1. $(\mathbf{X}, \mathbf{Y}) \in \mathrm{CEB}_{\alpha,\beta}^{\mathrm{Y}}$ if $\mathrm{H}(\mathbf{X} \mid \mathbf{Y}) \geq \alpha$.

2. $(\mathbf{X}, \mathbf{Y}) \in \mathrm{CEB}^{\mathrm{N}}_{\alpha,\beta}$ if $\mathrm{H}(\mathbf{X} \mid \mathbf{Y}) \leq \beta$.

The following is immediate from Lemma 3.9:

Lemma 3.11. *For all $\alpha - \beta > 1/\operatorname{poly}(n)$, $\mathrm{CEB}_{\alpha,\beta} \in \mathbf{SZK}$.*

3.4 Useful Facts and Lemmas

Fact 3.12 (Basic Facts about Entropy)**.** *The following hold for any random variables $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$:*

1. $\mathrm{H}(\mathbf{X} \mid \mathbf{Y}) \leq \mathrm{H}(\mathbf{X})$.
2. $\mathrm{I}(\mathbf{X}; \mathbf{Y} \mid \mathbf{Z}) = \mathrm{H}(\mathbf{X} \mid \mathbf{Z}) - \mathrm{H}(\mathbf{X} \mid \mathbf{YZ}) = \mathrm{H}(\mathbf{Y} \mid \mathbf{Z}) - \mathrm{H}(\mathbf{Y} \mid \mathbf{XZ}) \geq 0$
3. *Data processing inequality: for any randomized function $\mathbf{F}$ (whose randomness is independent of $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$), it holds that $\mathrm{I}(\mathbf{F}(\mathbf{X}); \mathbf{Y} \mid \mathbf{Z}) \leq \mathrm{I}(\mathbf{X}; \mathbf{Y} \mid \mathbf{Z})$.*

In the following, define for $\epsilon \in [0, 1]$ the value $\mathrm{H}(\epsilon) = \epsilon \lg(1/\epsilon) + (1 - \epsilon) \lg(1/1-\epsilon)$.

Lemma 3.13 (Bounding Statistical Distance from Conditional Entropy)**.** *If $\mathrm{Supp}(\mathbf{X}) = \{0,1\}^n$ then $\mathbb{E}_{Y \leftarrow \mathbf{Y}} \Delta(\mathbf{X} \mid Y, \mathbf{U}_n) \leq \sqrt{n - \mathrm{H}(\mathbf{X} \mid \mathbf{Y})}$.*

Proof. We use the following definition.

Definition 3.14 (Kullback-Leibler Divergence)**.** For random variables $\mathbf{X}, \mathbf{Y}$ such that $\mathrm{Supp}(\mathbf{X}) \subseteq \mathrm{Supp}(\mathbf{Y})$, the Kullback-Leibler divergence is defined as $\mathrm{KL}(\mathbf{X}, \mathbf{Y}) = \mathbb{E}_X \lg(\frac{\Pr[\mathbf{X}=X]}{\Pr[\mathbf{Y}=X]})$.

It can be verified by straightforward calculation that $\mathrm{H}(\mathbf{X}) = n - \mathrm{KL}(\mathbf{X}, \mathbf{U}_n)$.

Pinsker's inequality states that for any random variables $\mathbf{x}, \mathbf{y}$, it holds that $\Delta(\mathbf{x}, \mathbf{y}) \leq \sqrt{\mathrm{KL}(\mathbf{x}, \mathbf{y})}$. Applying Pinsker's inequality to $(\mathbf{X} \mid Y)$ and $\mathbf{U}_n$ for every fixed value of $Y \leftarrow \mathbf{Y}$ and using Jensen's inequality, we have:

$$\mathop{\mathbb{E}}_{Y \leftarrow \mathbf{Y}} \Delta(\mathbf{X} \mid Y, \mathbf{U}_n) \leq \mathop{\mathbb{E}}_{Y \leftarrow \mathbf{Y}} [\sqrt{n - \mathrm{H}(\mathbf{X} \mid Y)}] \leq \sqrt{\mathop{\mathbb{E}}_{Y \leftarrow \mathbf{Y}} [n - \mathrm{H}(\mathbf{X} \mid Y)]} = \sqrt{n - \mathrm{H}(\mathbf{X} \mid \mathbf{Y})}$$

□

Lemma 3.15 (Bounding Conditional Entropy from Statistical Distance)**.** *Suppose $\Delta((\mathbf{X}, \mathbf{Y}), (\mathbf{X}', \mathbf{Y}')) \leq \epsilon$ and $\mathrm{Supp}(\mathbf{X}) \cup \mathrm{Supp}(\mathbf{X}') \subseteq \{0,1\}^n$. Then it holds that $|\mathrm{H}(\mathbf{X} \mid \mathbf{Y}) - \mathrm{H}(\mathbf{X}' \mid \mathbf{Y}')| \leq 4(\mathrm{H}(\epsilon) + \epsilon \cdot n)$.*

To prove this lemma, we need the following:

Lemma 3.16. *Suppose $\Delta((\mathbf{X}, \mathbf{Y}), (\mathbf{X}', \mathbf{Y}')) \leq \epsilon$ and let $\mathbf{Z} = (\mathbf{X}'', \mathbf{Y})$ be a random variable distributed as follows. The component $Y \leftarrow \mathbf{Y}$ is sampled, and then $\mathbf{X}''$ conditioned on Y is distributed similar to $(\mathbf{X}' | \mathbf{Y}' = Y)$. Then it holds that $\Delta((\mathbf{X}, \mathbf{Y}), \mathbf{Z}) \leq 2\epsilon$.*

Proof. It is sufficient to show that $\Delta((\mathbf{X}', \mathbf{Y}'), \mathbf{Z}) \leq \epsilon$, which is true because the second components have statistical distance at most ϵ and the first components have statistical distance zero *conditioned* on the second components being equal. □

Proof of 3.15. We first prove the lemma for the case that $\mathbf{Y}, \mathbf{Y}'$ do not exist (*i.e.* $\Delta(\mathbf{X}, \mathbf{X}') \leq \epsilon$). It is well known that in this case there is a random variable $\overline{\mathbf{X}}$ which has a measure of $1-\epsilon$ in *both* of $\mathbf{X}$ and $\mathbf{X}'$. Namely, one can think of a Boolean random variable $\mathbf{b}$ jointly distributed with $\mathbf{X}, \mathbf{X}'$ such that $\Pr[\mathbf{b}=0] = \epsilon$ and $(\mathbf{X} \mid \mathbf{b}=1) \equiv \overline{\mathbf{X}} \equiv (\mathbf{X}' \mid \mathbf{b}=1)$. Based on this "decomposition" we get:

$$\mathrm{H}(\mathbf{X}) \geq \mathrm{H}(\mathbf{X} \mid \mathbf{b}) \geq (1-\epsilon)\,\mathrm{H}(\mathbf{X} \mid \mathbf{b}=1) = (1-\epsilon)\,\mathrm{H}(\overline{\mathbf{X}}).$$

On the other hand it holds that

$$\mathrm{H}(\mathbf{X}) \leq \mathrm{H}(\mathbf{b}) + \mathrm{H}(\mathbf{X} \mid \mathbf{b}) = \mathrm{H}(\epsilon) + \epsilon\,\mathrm{H}(\mathbf{X} \mid b=0) + (1-\epsilon)\cdot\mathrm{H}(\mathbf{X} \mid b=1) \leq \mathrm{H}(\epsilon) + \epsilon n + (1-\epsilon)\,\mathrm{H}(\overline{\mathbf{X}}).$$

Namely, both of $\mathrm{H}(\mathbf{X}), \mathrm{H}(\mathbf{X}')$ are lower-bounded by $(1-\epsilon)\,\mathrm{H}(\overline{\mathbf{X}})$ an upper-bounded by $\mathrm{H}(\epsilon) + \epsilon n + (1-\epsilon)\,\mathrm{H}(\overline{\mathbf{X}})$, and therefore $|\,\mathrm{H}(\mathbf{X}) - \mathrm{H}(\mathbf{X}')| \leq \mathrm{H}(\epsilon) + \epsilon n$.

Now, using the result above, we prove the conditional case through a hybrid argument. Given the two pairs of random variables $(\mathbf{X}, \mathbf{Y}), (\mathbf{X}', \mathbf{Y}')$ define the hybrid random variable $\mathbf{Z} = (\mathbf{X}'', \mathbf{Y})$ as defined in the statement of Lemma 3.16. We claim that

1. $|\,\mathrm{H}(\mathbf{X}'' \mid \mathbf{Y}) - \mathrm{H}(\mathbf{X} \mid \mathbf{Y})| \leq 2(\mathrm{H}(\epsilon) + \epsilon n)$, and
2. $|\,\mathrm{H}(\mathbf{X}'' \mid \mathbf{Y}) - \mathrm{H}(\mathbf{X}' \mid \mathbf{Y}')| \leq 2\epsilon n$.

Using these two bounds, Lemma 3.15 follows by a triangle inequality.

We obtain the first bound as follows.

$$\begin{aligned}
|\,\mathrm{H}(\mathbf{X}'' \mid \mathbf{Y}) - \mathrm{H}(\mathbf{X} \mid \mathbf{Y})| &\leq \mathop{\mathbb{E}}_{Y \leftarrow \mathbf{Y}} \left[|\,\mathrm{H}(\mathbf{X}'' \mid Y) - \mathrm{H}(\mathbf{X} \mid Y)| \right] \\
&\leq \mathop{\mathbb{E}}_{Y \leftarrow \mathbf{Y}} \left[\mathrm{H}(\Delta(\mathbf{X}'' \mid Y, \mathbf{X} \mid Y)) + \Delta(\mathbf{X}'' \mid Y, \mathbf{X} \mid Y) \cdot n \right] \\
\text{(by concavity of Entropy)} \quad &\leq \mathrm{H}\left(\mathop{\mathbb{E}}_{Y \leftarrow \mathbf{Y}} \left[\Delta(\mathbf{X}'' \mid Y, \mathbf{X} \mid Y) \right] \right) + \mathop{\mathbb{E}}_{Y \leftarrow \mathbf{Y}} \left[\Delta(\mathbf{X}'' \mid Y, \mathbf{X} \mid Y) \right] \cdot n \\
\text{(by Lemma 3.16)} \quad &\leq \mathrm{H}(2\epsilon) + (2\epsilon) \cdot n \\
\text{(by concavity of Entropy)} \quad &\leq 2(\mathrm{H}(\epsilon) + \epsilon \cdot n).
\end{aligned}$$

To obtain the second bound we do as follows.

$$\begin{aligned}
|\,\mathrm{H}(\mathbf{X}'' \mid \mathbf{Y}) - \mathrm{H}(\mathbf{X}' \mid \mathbf{Y}')| &= \left| \mathop{\mathbb{E}}_{Y \leftarrow \mathbf{Y}} \left[\mathrm{H}(\mathbf{X}' \mid \mathbf{Y}' = Y) \right] - \mathop{\mathbb{E}}_{Y \leftarrow \mathbf{Y}'} \left[\mathrm{H}(\mathbf{X}' \mid \mathbf{Y}' = Y) \right] \right| \\
&= \left| \sum_{Y \in \mathrm{Supp}(\mathbf{Y}) \cup \mathrm{Supp}(\mathbf{Y}')} \left(\Pr[\mathbf{Y} = Y] - \Pr[\mathbf{Y}' = y] \right) \cdot \mathrm{H}(\mathbf{X}' \mid \mathbf{Y}' = Y) \right| \\
&\leq \sum_{Y \in \mathrm{Supp}(\mathbf{Y}) \cup \mathrm{Supp}(\mathbf{Y}')} \left| \Pr[\mathbf{Y} = Y] - \Pr[\mathbf{Y}' = y] \right| \cdot n \\
&\leq 2\Delta(\mathbf{Y}, \mathbf{Y}') \cdot n
\end{aligned}$$

□

4 Proving the Main Result

Theorem 4.1. *Suppose the promise problem $L = (L^Y, L^N)$ has a ZK-PCP $\Pi = (\{\boldsymbol{\pi}_{x\in L,w}\}, \mathsf{V})$ of entropy at most $\mathrm{H}(\boldsymbol{\pi}_{x,w}) \leq \mathrm{poly}(|x|)$. Then $L \in \mathbf{SZK}$.*

(Note that the theorem extends beyond efficient ZK-PCP's and encompasses all ZK-PCP's where proofs have low entropy.) In the rest of this section we prove Theorem 4.1. Fix an efficient ZK-PCP for L. Efficiency means there exists $\eta = \mathrm{poly}(n)$ such that every possible honest proof π can be encoded by a circuit with binary description size at most η. This implies that for all $x \in L$ with witness w, if we let $\boldsymbol{\pi}_{x,w}$ be the distribution of proofs defined by the ZK-PCP, it holds that $\mathrm{H}(\boldsymbol{\pi}_{x,w}) \leq \eta$.

Let $\mathsf{V}^{[\ell]} = (\mathsf{V}^1, \ldots, \mathsf{V}^\ell)$ be a verifier who executes ℓ independent instances of V against the given oracle and let V^i be its i^{th} verification. (We will fix a choice of $\ell = \mathrm{poly}(n)$ later.) Let $\textsc{Sim}$ be the simulator that simulates the view of $\mathsf{V}^{[\ell]}$ statistically well (*i.e.* $\textsc{Sim}(x)$ is $\mathrm{negl}(|x|)$-close to the view of $\mathsf{V}^{[\ell]}(x)$ when accessing $\pi_x \leftarrow \boldsymbol{\pi}_{x,w}$ for $x \in L$). The view of V^i can be represented as $\nu^i = (r^i, q^i_1, a^i_1, \ldots, q^i_m, a^i_m)$ where $r^i \in \{0,1\}^k$ is the randomness used by V^i, q^i_j is its j^{th} oracle query and a^i_j is the answer to q^i_j. We use the notation $\overline{a}^i = (a^i_1, \ldots, a^i_m), \overline{q}^i = (q^i_1, \ldots, q^i_m)$. The view of $\mathsf{V}^{[\ell]}$ consists of $(\nu^1, \ldots, \nu^\ell)$.

In order to prove $L \in \mathbf{SZK}$, we show how to reduce L to a constant size formula over $\mathbf{SZK}$ languages. Roundly speaking, our reduction reduces L to $\mathrm{CEB} \wedge \overline{\mathrm{CEB}} \wedge D$ (see Definition 3.1) where $D \in \mathbf{BPP} \subseteq \mathbf{SZK}$ which makes $\mathrm{CEB} \wedge \overline{\mathrm{CEB}} \wedge D \in \mathbf{SZK}$ (see Lemma 3.6). To describe our reduction formally we first need to define a circuit C^{SIM}_x and a promise problem $D_{\alpha,\beta}$ as follows.

- The circuit C^{SIM}_x takes as input r_{SIM} (for input length $|x|$). The circuit C_x outputs $\textsc{Sim}(x; r_{\text{SIM}}) = (\nu^1, \ldots, \nu^\ell)$ where for each $i \in [\ell]$, $\nu^i = (r^i, q^i_1, a^i_1, \ldots, q^i_m, a^i_m)$.
- For $\alpha > \beta$, $D_{\alpha,\beta}$ is a promise problem whose inputs are Boolean circuits C. Suppose the input length of C is n, then:
  1. $C \in D^{\text{Y}}_{\alpha,\beta}$ iff $\Pr[C(\mathbf{U}_n) = 1] \geq \alpha]$, and
  2. $C \in D^{\text{N}}_{\alpha,\beta}$ iff $\Pr[C(\mathbf{U}_n) = 1] \leq \beta]$.

  It is easy to see that for $\alpha - \beta > 1/\mathrm{poly}(n)$, $D_{\alpha,\beta} \in \mathbf{BPP}$.

Reduction 4.2 (Main Reduction)**.** *Given a parameter ℓ, we map $x \mapsto (C_1, C_2, C_3)$ as follows.*

1. *C_1 is a circuit sampling the joint distribution $(\mathbf{X}_1, \mathbf{Y}_1)$ defined as follows. On input (r_{SIM}, i), C_1 executes the circuit C^{SIM}_x on a random r_{SIM} to get $(\nu^1, \ldots, \nu^\ell) \leftarrow C^{\text{SIM}}_x(r_{\text{SIM}})$ and sets:*

$$X_1 = r^i \qquad \text{and} \qquad Y_1 = (\nu^1, \ldots, \nu^{i-1}).$$

2. *C_2 is a circuit sampling the joint distribution $(\mathbf{X}_2, \mathbf{Y}_2)$ defined as follows. On input (r_{SIM}, i, j), C_2 executes the circuit C^{SIM}_x on a random r_{SIM} to get $(\nu^1, \ldots, \nu^\ell) \leftarrow C^{\text{SIM}}_x(r_{\text{SIM}})$ and sets:*

$$X_2 = a^i_j \qquad \text{and} \qquad Y_2 = (\nu^1, \ldots, \nu^{i-1}, q^i_j).$$

   We emphasize the fact that while a^i_j, q^i_j appear in the output of C_2, the actual index j itself does not *appear in the output.*

3. *C_3 is a circuit computing the following. On input (r_{SIM}, i), run $C_x^{\text{SIM}}(r_{\text{SIM}}) = (\nu^1, \ldots, \nu^\ell)$, and output 1 iff ν^i is an accepting view of V.*

Claim 4.3. *Reduction 4.2 is a Karp reduction from L (specified in Theorem 4.1) to the promise language $Z = \text{CEB}_{k-1/200,k-1/100} \wedge \overline{\text{CEB}}_{2\eta/\ell,1.1\eta/\ell} \wedge D_{0.66,\beta}$ for $\beta = 1/3 + 1/10 + 2m\eta/\ell$.*

Proving Theorem 4.1 using Claim 4.3. By taking $\ell = 40m\eta$, it holds that $2m \cdot \eta/\ell < 1/20$ in Lemma 4.8 and so $\beta < 1/2$, which implies that $D_{\alpha,\beta} \in \mathbf{BPP}$, $Z \in \mathbf{SZK}$, and so $L \in \mathbf{SZK}$.

In the following we prove Claim 4.3 by studying each cases of $x \in L^Y$ and $x \in L^N$ separately. We begin with a lemma that will be useful for the case $x \in L$.

The following lemma bounds the conditional entropy of a single answer to a single randomly chosen verifier query by the conditional entropy of the set of *all* answers to the set of *all* verifier queries. This is non-trivial because the verifier queries may be asked adaptively.

Lemma 4.4. *Let A be any randomized algorithm that (adaptively) queries a PCP π. Let $r \in \{0,1\}^k$ denote the random coins of A. Let $\overline{q} = (q_1, \ldots, q_m)$ be the queries that $A^\pi(r)$ makes and let $a_j = \pi(q_j)$ be the corresponding answers. Let $\boldsymbol{\pi}$ be an arbitrary distribution over proofs, and let $\overline{\mathbf{q}}$ and $\overline{\mathbf{a}}$ be the distribution over (the vectors of) queries and answers obtained by querying $\boldsymbol{\pi}$ using algorithm A on uniform random coins $\mathbf{r}$. Let also $\mathbf{j}$ be an arbitrary distribution over $[m]$.*

Then $\mathrm{H}(\mathbf{a_j} \mid \mathbf{q_j}) \leq \mathrm{H}(\overline{\mathbf{a}} \mid \mathbf{r})$ where in the notation $\mathbf{q_j}$ the value of $\mathbf{j}$ is not explicitly revealed.

Proof. By the definition of conditional entropy and adding $0 = \mathrm{H}(\mathbf{a_j q_j} \mid \boldsymbol{\pi}) - \mathrm{H}(\mathbf{a_j q_j} \mid \boldsymbol{\pi})$, we get

$$\mathrm{H}(\mathbf{a_j} \mid \mathbf{q_j}) = \mathrm{H}(\mathbf{a_j q_j}) - \mathrm{H}(\mathbf{a_j q_j} \mid \boldsymbol{\pi}) - (\mathrm{H}(\mathbf{q_j}) - \mathrm{H}(\mathbf{a_j q_j} \mid \boldsymbol{\pi})).$$

Since a proof π is *stateless* for any fixed π, given any query q asked at some point during the execution of A^π, the answer $a = \pi(q)$ is also fixed. Therefore it holds that $\mathrm{H}(\mathbf{a_j q_j} \mid \boldsymbol{\pi}) = \mathrm{H}(\mathbf{q_j} \mid \boldsymbol{\pi})$, and by the definition of mutual information, we may deduce that

$$\mathrm{H}(\mathbf{a_j} \mid \mathbf{q_j}) = \mathrm{I}(\mathbf{a_j q_j}; \boldsymbol{\pi}) - \mathrm{I}(\mathbf{q_j}; \boldsymbol{\pi}) \leq \mathrm{I}(\mathbf{a_j q_j}; \boldsymbol{\pi}).$$

Since $\mathrm{I}(\mathbf{a_j q_j}; \boldsymbol{\pi}) = \mathrm{H}(\boldsymbol{\pi}) - \mathrm{H}(\boldsymbol{\pi} \mid \mathbf{a_j q_j})$ and since $\boldsymbol{\pi}$ and $\mathbf{r}$ are independent, Item 1 of Fact 3.12 implies that

$$\mathrm{H}(\mathbf{a_j} \mid \mathbf{q_j}) \leq \mathrm{I}(\mathbf{a_j q_j}; \boldsymbol{\pi}) = \mathrm{H}(\boldsymbol{\pi}) - \mathrm{H}(\boldsymbol{\pi} \mid \mathbf{a_j q_j}) \leq \mathrm{H}(\boldsymbol{\pi} \mid \mathbf{r}) - \mathrm{H}(\boldsymbol{\pi} \mid \mathbf{a_j q_j r}) = \mathrm{I}(\mathbf{a_j q_j}; \boldsymbol{\pi} \mid \mathbf{r}).$$

Let $\mathbf{F}$ be the function that takes as input $(\overline{\mathbf{a}}, \overline{\mathbf{q}})$ and outputs $(\mathbf{a_j}, \mathbf{q_j})$ by sampling $\mathbf{j}$. By the data processing inequality (Item 3 of Fact 3.12) it holds that

$$\mathrm{H}(\mathbf{a_j} \mid \mathbf{q_j}) \leq \mathrm{I}(\mathbf{a_j q_j}; \boldsymbol{\pi} \mid \mathbf{r}) = \mathrm{I}(\mathbf{F}(\overline{\mathbf{aq}}); \boldsymbol{\pi} \mid \mathbf{r}) \leq \mathrm{I}(\overline{\mathbf{aq}}; \boldsymbol{\pi} \mid \mathbf{r}) \leq \mathrm{H}(\overline{\mathbf{aq}} \mid \mathbf{r}) = \mathrm{H}(\overline{\mathbf{a}} \mid \mathbf{r}) + \mathrm{H}(\overline{\mathbf{q}} \mid \overline{\mathbf{a}}\mathbf{r}).$$

Finally, since $\mathrm{H}(\overline{\mathbf{q}} \mid \overline{\mathbf{a}}\mathbf{r}) = 0$, this implies the proposition. □

Remark 4.5. We emphasize that if $\boldsymbol{\pi}$ was *stateful* (*i.e.* a "prover", rather than a "proof"), then Lemma 4.4 would be *false*. Even a deterministic prover can correlate his answers to the verifier's queries, and so it may be that $\mathrm{H}(\overline{\mathbf{a}} \mid \overline{\mathbf{q}}) = 0$ but $\mathrm{H}(\mathbf{a_j} \mid \mathbf{q_j}) > 0$. Namely, even given π (say for a stateful prover that π gives the random coins of the prover) and a query q, the answer to q may have entropy because π's answer to q may be different depending on whether q was asked as the first query or second query or third query, etc. In particular, the equality $\mathrm{H}(\mathbf{a_j q_j} \mid \boldsymbol{\pi}) = \mathrm{H}(\mathbf{q_j} \mid \boldsymbol{\pi})$ used in the proof of Lemma 4.4 would not hold anymore. This is one place where we crucially use the fixed nature of a PCP.

4.1 Proof of Claim 4.3: the Case $x \in L^Y$

Here we would like to show that $(C_1 \in \mathrm{CEB}^{\mathrm{Y}}_{k-1/200,k-1/100}) \wedge (C_2 \in \overline{\mathrm{CEB}}^{\mathrm{Y}}_{2\eta/\ell,1.1\eta/\ell}) \wedge (C_3 \in D^{\mathrm{Y}}_{0.66,\beta})$. We study each of the generated instances C_i for $i \in [3]$. In all these cases, we first assume that the simulator's output is identically distributed to the view of $\mathsf{V}^{[\ell]}$ interacting with a prover and then will show how remove this assumption.

The Instance C_1. If the simulator's outputs were identically distributed to the view of $\mathsf{V}^{[\ell]}$ interacting with a prover, then the simulated randomness $\mathbf{X}_1 = \mathbf{r}^i$ will be uniformly distributed over $\{0,1\}^k$ with entropy k *independently* of $\mathbf{Y}_1 = (\boldsymbol{\nu}^1, \ldots, \boldsymbol{\nu}^{i-1})$. Since the simulator generates a view that is statistically close to the honest interaction (and since $k = \mathrm{poly}(|x|)$ and $\mathrm{H}(\mathrm{negl}(n)) = \mathrm{negl}(n)$) we may apply Lemma 3.15 to deduce that $\mathrm{H}(\mathbf{X}_1 \mid \mathbf{Y}_1) \geq k - \mathrm{negl}(n) \geq k - 1/200$. Therefore, $C_1 \in \mathrm{CEB}^{\mathrm{Y}}_{k-1/200,k-1/100}$.

The Instance C_2. Fix an arbitrary witness w of $x \in L$, and we study the view of $\mathsf{V}^{[\ell]}$ while interacting with a proof generated according to the distribution $\boldsymbol{\pi}_{x,w}$ whose entropy is bounded by η. Suppose first that the simulator's outputs were identically distributed to the view of $\mathsf{V}^{[\ell]}$ interacting with $\boldsymbol{\pi}_{x,w}$. In this case, by an argument similar to [IMS12], one can show that

Claim 4.6. $\mathbb{E}_{i \leftarrow [\ell]} \mathrm{H}(\overline{\mathbf{a}}^i \mid \boldsymbol{\nu}^1, \ldots, \boldsymbol{\nu}^{i-1}, \mathbf{r}^i) \leq \eta/\ell$.

Proof.

$$\begin{aligned}
\eta + k\ell &\geq \mathrm{H}(\boldsymbol{\pi}_{x,w}) + \mathrm{H}(\mathbf{r}^1, \ldots, \mathbf{r}^\ell) \\
(\boldsymbol{\pi}_{x,w} \text{ and } \mathbf{r}^1, \ldots, \mathbf{r}^\ell \text{ are independent}) &= \mathrm{H}(\boldsymbol{\pi}_{x,w}, \mathbf{r}^1, \ldots, \mathbf{r}^\ell) \\
(\boldsymbol{\pi}_{x,w} \text{ and } \mathbf{r}^1, \ldots, \mathbf{r}^\ell \text{ determine } \boldsymbol{\nu}^1, \ldots, \boldsymbol{\nu}^\ell) &\geq \mathrm{H}(\boldsymbol{\nu}^1, \ldots, \boldsymbol{\nu}^\ell) \\
&= \sum_{i \in [\ell]} \mathrm{H}(\boldsymbol{\nu}^i \mid \boldsymbol{\nu}^1, \ldots, \boldsymbol{\nu}^{i-1}) \\
(\mathbf{r}^i \text{ and } \overline{\mathbf{a}}^i \text{ determine } \overline{\mathbf{q}}^i) &= \sum_{i \in [\ell]} \mathrm{H}(\mathbf{r}^i \mid \boldsymbol{\nu}^1, \ldots, \boldsymbol{\nu}^{i-1}) + \mathrm{H}(\overline{\mathbf{a}}^i \mid \boldsymbol{\nu}^1, \ldots, \boldsymbol{\nu}^{i-1}, \mathbf{r}^i) \\
&= k\ell + \sum_{i \in [\ell]} \mathrm{H}(\overline{\mathbf{a}}^i \mid \boldsymbol{\nu}^1, \ldots, \boldsymbol{\nu}^{i-1}, \mathbf{r}^i).
\end{aligned}$$

Therefore, by averaging over i we obtain that $\mathbb{E}_{i \leftarrow [\ell]} \mathrm{H}(\overline{\mathbf{a}}^i \mid \boldsymbol{\nu}^1, \ldots, \boldsymbol{\nu}^{i-1}, \mathbf{r}^i) \leq \eta/\ell$. □

The following claim is also based on the assumption that the simulation is perfect, and thus the distribution of $(\boldsymbol{\nu}^1, \ldots, \boldsymbol{\nu}^m)$ generated by the simulator is identical to the view of $\mathsf{V}^{[\ell]}$ run against $\pi \leftarrow \boldsymbol{\pi}_{x \in L,w}$.

Claim 4.7. *For each fixed value of i and $(\nu^1, \ldots, \nu^{i-1})$, it holds that*

$$\mathrm{H}(\mathbf{a}^i_j \mid \mathbf{q}^i_j, \nu^1, \ldots, \nu^{i-1}) \leq \mathrm{H}(\overline{\mathbf{a}}^i \mid \mathbf{r}^i, \nu^1, \ldots, \nu^{i-1}) \tag{1}$$

Namely, the entropy of the answers of the i^{th} verification gives an upper-bound on the entropy of the answer to a randomly chosen query of the verifier without *revealing its index.*

Proof. Let $(\boldsymbol{\pi}_{x,w}, \boldsymbol{\nu}^1, \ldots, \boldsymbol{\nu}^{i-1})$ be the joint distribution of an honest proof $\boldsymbol{\pi}_{x,w}$ and $i-1$ executions of the honest verifier $\mathsf{V}^1, \ldots, \mathsf{V}^{i-1}$ using proof $\boldsymbol{\pi}_{x,w}$. Apply Lemma 4.4 using the distribution over proofs given by $(\boldsymbol{\pi}_{x,w} \mid \nu^1, \ldots, \nu^{i-1})$, and with the honest verifier algorithm V^i as the query algorithm accessing the proof. □

Using Claims 4.6 and 4.7, we conclude that $\mathrm{H}(\mathbf{X}_2 \mid \mathbf{Y}_2) \leq \eta/\ell$, *assuming that* the simulator was perfect. If we only assume that the simulator's output is statistically close to the view of $\mathsf{V}^{[\ell]}$ interacting with $\boldsymbol{\pi}_{x,w}$, then we can apply Lemma 3.15 and deduce that $\mathrm{H}(\mathbf{X}_2 \mid \mathbf{Y}_2) \leq \eta/\ell + \mathrm{negl}(n) < 1.1\eta/\ell$ which implies that $C_2 \in \overline{\mathrm{CEB}}^{\mathrm{Y}}_{2\eta/\ell, 1.1\eta/\ell}$.

The Instance C_3. By the completeness of Π, when $\mathsf{V}^{[\ell]} = (\mathsf{V}^1, \ldots, V^\ell)$ interacts with a proof, for all $i \in [\ell]$, V^i accepts with probability $\geq 2/3$. Since the simulation is statistically close to the real interaction, it holds that $\boldsymbol{\nu}^i$ is accepting with probability $2/3 - \mathrm{negl}(n) \geq 0.66$, and so $C_3 \in D^{\mathrm{Y}}_{0.66,\beta}$.

4.2 Proof of Claim 4.3: the Case $x \in L^N$

Here we would like to show that $(C_1 \in \mathrm{CEB}^{\mathrm{N}}_{k-1/200, k-1/100}) \vee (C_2 \in \overline{\mathrm{CEB}}^{\mathrm{N}}_{2\eta/\ell, 1.1\eta/\ell}) \vee (C_3 \in D^{\mathrm{N}}_{0.66,\beta})$. This follows from the following lemma.

Lemma 4.8. *Suppose $x \in L^N$, $C_1 \notin \mathrm{CEB}^{\mathrm{N}}_{k-1/200, k-1/100}$, and $C_2 \notin \overline{\mathrm{CEB}}^{\mathrm{N}}_{2\eta/\ell, 1.1\eta/\ell}$. Then it holds that $C_3 \in D^{\mathrm{N}}_{0.66,\beta}$ for $\beta = 1/3 + 1/10 + 2m \cdot \eta/\ell$.*

Intuition. Since $C_2 \notin \overline{\mathrm{CEB}}^{\mathrm{N}}_{2\eta/\ell, 1.1\eta/\ell}$, therefore, the oracle answers returned to the verifier in the i^{th} execution (for a random $i \leftarrow [\ell]$) all have very low entropy and thus close to a *fixed* proof. Moreover, due to $C_1 \notin \mathrm{CEB}^{\mathrm{N}}_{k-1/200, k-1/100}$, the randomness of verifier in this execution has almost full entropy, and therefore, the i^{th} execution is close to an honest execution of the verifier against some oracle. Finally, since $x \in L^N$ by the soundness of the PCP, the verifier would accept with probability at most $\approx 1/3$. The formal argument goes through a hybrid argument as follows.

Experiments. The outputs of all experiments described below consist of a view of $\mathsf{V}^{[i]}$ (*i.e.* the first i executions of the verifier). The distribution of $(\nu^1, \ldots, \nu^{i-1})$ in all of these executions is the same and is sampled by $\textsc{Sim}(x)$, and they only differ in the way they sample ν^i.

- **Experiment** Real**.** Choose $i \leftarrow [\ell]$, and take the output $(\nu^1, \ldots, \nu^i)$ by running $\textsc{Sim}(x)$.
- **Experiment** Ideal**.** Choose $i \leftarrow [\ell]$, and take the output $(\nu^1, \ldots, \nu^{i-1})$ by running $\textsc{Sim}(x)$. To sample $\boldsymbol{\nu}^i = (\mathbf{r}^i, \overline{\mathbf{q}}^i, \overline{\mathbf{a}}^i)$ we first sample $r^i \leftarrow \{0,1\}^k$ uniformly at random, and then using r^i we run the verifier against the oracle $\widehat{\pi}$ defined as follows.

  The Oracle $\widehat{\pi}$: Suppose we have fixed $(\nu^i, \ldots, \nu^{i-1})$. Recall the distribution $((\mathbf{q}^i_{\mathbf{j}}, \mathbf{a}^i_{\mathbf{j}}) \mid \nu^i, \ldots, \nu^{i-1})$ defined above when defining the instance C_2 (*i.e.*, $(\mathbf{a}^i_{\mathbf{j}}, \mathbf{q}^i_{\mathbf{j}})$ is a randomly chosen pair of query-answer pairs from the view ν^i without revealing the index j). For every query q, the oracle $\widehat{\pi}$ gets one sample according to $a \leftarrow (\mathbf{a}^i_{\mathbf{j}} \mid \nu^i, \ldots, \nu^{i-1}, \mathbf{q}^i_{\mathbf{j}} = q)$ and sets $\widehat{\pi}(q) = a$ forever. If $\Pr[\mathbf{q}^i_{\mathbf{j}} = q \mid \nu^i, \ldots, \nu^{i-1}] = 0$, we define $\widehat{\pi}(q) = \perp$.

- **Experiment Hyb_j for $j \in [m+1]$.** These experiments are in between Real and Ideal and for larger j they become closer to Real. Here we choose $i \leftarrow [\ell]$, and take the output $(\nu^1, \ldots, \nu^i)$ by running $\textsc{Sim}(x)$. Then we will *re-sample* parts of ν^i as follows. We will keep $(r^i, (q_1^i, a_1^i), \ldots, (q_{j-1}^i, a_{j-1}^i))$ as sampled by $\textsc{Sim}(x)$. For the remaining queries and answers we sample an oracle $\widehat{\pi}$ as described in Ideal, and we let $(q_j^i, a_j^i), \ldots, (q_m^i, a_m^i)$ be the result of continuing the execution of V^i using r^i and the oracle $\widehat{\pi}$. Note that $\mathsf{Hyb}_{m+1} \equiv \mathsf{Real}$.

Claim 4.9. *If $x \in L^N$, then $\Pr_{\mathsf{Ideal}}[\nu^i \text{ accepts }] \leq 1/3$.*

Claim 4.10. *If $C_1 \notin \mathrm{CEB}^{\mathrm{N}}_{k-1/200,k-1/100}$, then $\Delta(\mathsf{Ideal}, \mathsf{Hyb}_1) \leq 1/10$.*

Claim 4.11. *If $C_2 \notin \overline{\mathrm{CEB}}^{\mathrm{N}}_{2\eta/\ell,1.1\eta/\ell}$, then $\mathbb{E}_{j\in[m]} \Delta(\mathsf{Hyb}_j, \mathsf{Hyb}_{j+1}) \leq 2\eta/\ell$.*

Proving Lemma 4.8. Claims 4.9, 4.10, and 4.11 together imply that

$$\Pr_{\mathsf{Real}}[\nu^i \text{ accepts }] \leq \Pr_{\mathsf{Ideal}}[\nu^i \text{ accepts }] + \Delta(\mathsf{Ideal}, \mathsf{Hyb}_1) + \sum_{j\in[m]} \Delta(\mathsf{Hyb}_j, \mathsf{Hyb}_{j+1}) \leq 1/3 + 1/10 + 2m\eta/\ell$$

which proves that $C_3 \in D^{\mathrm{N}}_{2/3,\beta}$. In the following we prove these claims.

Proof of Claim 4.9. Since the oracle $\widehat{\pi}$ is sampled and fixed before choosing r^i and executing V^i, and because $x \in L^N$, by the soundness property of the PCP it holds that $\Pr_{\mathsf{Ideal}}[\nu^i \text{ accepts }] \leq 1/3$. □

Proof of Claim 4.10. If $C_1 \notin \mathrm{CEB}^{\mathrm{N}}_{k-1/200,k-1/100}$, then it means that we have $\mathbb{E}_{i\leftarrow[\ell]}[\mathrm{H}(\mathbf{r}^i \mid \boldsymbol{\nu}^1, \ldots, \boldsymbol{\nu}^{i-1})] \geq k - 1/100$. By Lemma 3.13 it holds that

$$\mathop{\mathbb{E}}_{i\leftarrow[\ell],\nu^1,\ldots,\nu^{i-1}}[\Delta((\mathbf{r}^i \mid \nu^1, \ldots, \nu^{i-1}), \mathbf{U}_k)] \leq \sqrt{1/100} = 1/10.$$

But note that the only difference between Ideal and Hyb_1 is the way we sample r^i conditioned on the previously sampled parts (*i.e.* $\nu^1, \ldots, \nu^{i-1}$). Thus it holds that $\Delta(\mathsf{Ideal}, \mathsf{Hyb}_1) \leq 1/10$. □

Proof of Claim 4.11. The only difference between Hyb_j and Hyb_{j+1} is the way they answer q_j^i. In Hyb_{j+1} the original answer of the simulator is used, while in Hyb_j this answer is provided by the oracle $\widehat{\pi}$. Thus, they are different only when the answer re-sampled by $\widehat{\pi}$ differs from the original answer. Therefore, we have that:

$$\Delta(\mathsf{Hyb}_j, \mathsf{Hyb}_{j+1}) \leq \mathop{\mathbb{E}}_{\nu^1,\ldots,\nu^{i-1},i}\left[\Pr_{\overline{\mathbf{a}}^i,\overline{\mathbf{q}}^i,\widehat{\boldsymbol{\pi}}}[\mathbf{a}_j^i \neq \widehat{\boldsymbol{\pi}}(\mathbf{q}_j^i) \mid i, \nu^1, \ldots, \nu^{i-1}]\right]$$

Taking an expectation over all $j \leftarrow [\ell]$ we conclude Claim 4.11 as follows.

$$\begin{aligned}\mathop{\mathbb{E}}_{j}[\Delta(\mathsf{Hyb}_j, \mathsf{Hyb}_{j+1})] &= \mathop{\mathbb{E}}_{j,i,\nu^1,\ldots,\nu^{i-1}}\left[\Pr_{\overline{\mathbf{a}}^i,\overline{\mathbf{q}}^i,\widehat{\boldsymbol{\pi}}}[\mathbf{a}_j^i \neq \widehat{\boldsymbol{\pi}}(\mathbf{q}_j^i) \mid i, \nu^1, \ldots, \nu^{i-1}]\right] \\ &= \mathop{\mathbb{E}}_{i,\nu^1,\ldots,\nu^{i-1}}\left[\Pr_{\mathbf{j},\overline{\mathbf{a}}^i,\overline{\mathbf{q}}^i,\widehat{\boldsymbol{\pi}}}[\mathbf{a}_{\mathbf{j}}^i \neq \widehat{\boldsymbol{\pi}}(\mathbf{q}_{\mathbf{j}}^i) \mid i, \nu^1, \ldots, \nu^{i-1}]\right]\end{aligned}$$

By combining the sampling of $\mathbf{a}^i_\mathbf{j}, \mathbf{q}^i_\mathbf{j}$ directly, we have that

$$\begin{aligned}
\mathop{\mathbb{E}}_{j}[\Delta(\mathsf{Hyb}_j, \mathsf{Hyb}_{j+1})] &= \mathop{\mathbb{E}}_{i,\nu^1,\ldots,\nu^{i-1}}\left[\Pr_{\mathbf{a}^i_\mathbf{j},\mathbf{q}^i_\mathbf{j},\widehat{\boldsymbol{\pi}}}[\mathbf{a}^i_\mathbf{j} \neq \widehat{\boldsymbol{\pi}}(\mathbf{q}^i_\mathbf{j}) \mid i,\nu^1,\ldots,\nu^{i-1}]\right] \\
&= \mathop{\mathbb{E}}_{i,\nu^1,\ldots,\nu^{i-1}}\left[1 - \Pr_{\mathbf{a}^i_\mathbf{j},\mathbf{q}^i_\mathbf{j},\widehat{\boldsymbol{\pi}}}[\mathbf{a}^i_\mathbf{j} = \widehat{\boldsymbol{\pi}}(\mathbf{q}^i_\mathbf{j}) \mid i,\nu^1,\ldots,\nu^{i-1}]\right] \\
&= \mathop{\mathbb{E}}_{i,\nu^1,\ldots,\nu^{i-1},q^i_j,a^i_j}\left[1 - \Pr_{\widehat{\boldsymbol{\pi}}}[a^i_j = \widehat{\boldsymbol{\pi}}(q^i_j) \mid i,\nu^1,\ldots,\nu^{i-1}]\right] \\
(\text{since } 1-\alpha \leq \lg(1/\alpha) \text{ for } \alpha \in [0,1]) \quad &\leq \mathop{\mathbb{E}}_{i,\nu^1,\ldots,\nu^{i-1},q^i_j,a^i_j}\left[\lg \frac{1}{\Pr_{\widehat{\boldsymbol{\pi}}}[a^i_j = \widehat{\boldsymbol{\pi}}(q^i_j) \mid i,\nu^1,\ldots,\nu^{i-1}]}\right] \\
(\text{by the definition of oracle } \widehat{\boldsymbol{\pi}}) \quad &= \mathop{\mathbb{E}}_{i}\left[\mathrm{H}(\mathbf{a}^\mathbf{i}_\mathbf{j} \mid \boldsymbol{\nu}^1,\ldots,\boldsymbol{\nu}^{\mathbf{i}-1},\mathbf{q}^\mathbf{i}_\mathbf{j})\right] \\
(\text{since } C_2 \notin \overline{\mathrm{CEB}}^{\mathrm{N}}_{2\eta/\ell,1.1\eta/\ell}) \quad &\leq 2\eta/\ell.
\end{aligned}$$

□

References

[AGGM06] Adi Akavia, Oded Goldreich, Shafi Goldwasser, and Dana Moshkovitz. On basing one-way functions on np-hardness. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing (STOC)*, pages 701–710, 2006. 4

[AH91] William Aiello and Johan Håstad. Statistical zero-knowledge languages can be recognized in two rounds. *Journal of Computer and System Sciences*, 42(3):327–345, 1991. Preliminary version in *FOCS'87*. 2, 4

[ALM+98] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and the hardness of approximation problems. *Journal of the ACM*, 45(3):501–555, 1998. Preliminary version in *FOCS'92*. 1

[AS98] Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: a new characterization of NP. *Journal of the ACM*, 45(1):70–122, 1998. Preliminary version in *FOCS'92*. 1

[BFL90] László Babai, Lance Fortnow, and Carsten Lund. Non-deterministic exponential time has two-prover interactive protocols. In *FOCS*, pages 16–25, 1990. 1

[BGKW88] Michael Ben-Or, Shafi Goldwasser, Joe Kilian, and Avi Wigderson. Multi-prover interactive proofs: How to remove intractability assumptions. In *STOC*, pages 113–131, 1988. 1

[BHZ87] Ravi B. Boppana, Johan Håstad, and Stathis Zachos. Does co-NP have short interactive proofs? *Information Processing Letters*, 25:127–132, 1987. 2

[BM88] László Babai and Shlomo Moran. Arthur-merlin games: A randomized proof system, and a hierarchy of complexity classes. *J. Comput. Syst. Sci.*, 36(2):254–276, 1988. 1

[BT06] Andrej Bogdanov and Luca Trevisan. Average-case complexity. *CoRR*, 2006. 4

[CGGM00] Ran Canetti, Oded Goldreich, Shafi Goldwasser, and Silvio Micali. Resettable zero-knowledge (extended abstract). In *STOC*, pages 235–244, 2000. 2

[CGS08] Nishanth Chandran, Vipul Goyal, and Amit Sahai. New constructions for UC secure computation using tamper-proof hardware. In *EUROCRYPT*, pages 545–562, 2008. 3

[DFK+92] Cynthia Dwork, Uriel Feige, Joe Kilian, Moni Naor, and Shmuel Safra. Low communication 2-prover zero-knowledge proofs for np. In *CRYPTO*, pages 215–227, 1992. 2

[FF93] Joan Feigenbaum and Lance Fortnow. Random-self-reducibility of complete sets. *SIAM Journal on Computing*, 22(5):994–1005, 1993. 4

[For89] Lance Fortnow. The complexity of perfect zero-knowledge. *Advances in Computing Research: Randomness and Computation*, 5:327–343, 1989. 2, 4

[FRS94] Lance Fortnow, John Rompel, and Michael Sipser. On the power of multi-prover interactive protocols. *Theoretical Computer Science*, 134(2):545–557, 1994. 1

[GIMS10] Vipul Goyal, Yuval Ishai, Mohammad Mahmoody, and Amit Sahai. Interactive locking, zero-knowledge PCPs, and unconditional cryptography. In Tal Rabin, editor, *CRYPTO*, volume 6223 of *Lecture Notes in Computer Science*, pages 173–190. Springer, 2010. 3

[GIS+10] Vipul Goyal, Yuval Ishai, Amit Sahai, Ramarathnam Venkatesan, and Akshay Wadia. Founding cryptography on tamper-proof hardware tokens. In Daniele Micciancio, editor, *TCC*, volume 5978 of *Lecture Notes in Computer Science*, pages 308–326. Springer, 2010. 3

[GKR08] Shafi Goldwasser, Yael Tauman Kalai, and Guy Rothblum. One-time programs. In *CRYPTO*, Lecture Notes in Computer Science, pages 39–56. Springer, 2008. 3

[GMR89] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*, 18(1):186–208, 1989. Preliminary version in *STOC'85*. 1

[GMW91] Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. *Journal of the ACM*, 38(1):691–729, 1991. Preliminary version in *FOCS'86*. 1

[GOVW12] Sanjam Garg, Rafail Ostrovsky, Ivan Visconti, and Akshay Wadia. Resettable statistical zero knowledge. In Ronald Cramer, editor, *TCC*, volume 7194 of *Lecture Notes in Computer Science*, pages 494–511. Springer, 2012. 2

[GS89] Shafi Goldwasser and Michael Sipser. Private coins versus public coins in interactive proof systems. *Advances in Computing Research: Randomness and Computation*, 5:73–90, 1989. 4

[GVW01] Oded Goldreich, Salil Vadhan, and Avi Wigderson. On interactive proofs with a laconic prover. In *Proc. 28th ICALP*, pages 334–345, 2001. 4

[HMX10] Iftach Haitner, Mohammad Mahmoody, and David Xiao. A new sampling protocol and applications to basing cryptographic primitives on the hardness of NP. In *IEEE Conference on Computational Complexity*, pages 76–87. IEEE Computer Society, 2010. 4

[IMS12] Yuval Ishai, Mohammad Mahmoody, and Amit Sahai. On efficient zero-knowledge PCPs. In Ronald Cramer, editor, *TCC*, volume 7194 of *Lecture Notes in Computer Science*, pages 151–168. Springer, 2012. 2, 3, 4, 5, 13

[Kat07] Jonathan Katz. Universally composable multi-party computation using tamper-proof hardware. In *EUROCRYPT*, Lecture Notes in Computer Science, pages 115–128. Springer, 2007. 3

[Kil92] Joe Kilian. A note on efficient zero-knowledge proofs and arguments (extended abstract). In *Proceedings of the 24th Annual ACM Symposium on Theory of Computing (STOC)*, pages 723–732, 1992. 1

[Kol10] Vladimir Kolesnikov. Truly efficient string oblivious transfer using resettable tamper-proof tokens. In *TCC*, pages 327–342, 2010. 3

[KPT97] Joe Kilian, Erez Petrank, and Gábor Tardos. Probabilistically checkable proofs with zero knowledge. In *STOC: ACM Symposium on Theory of Computing (STOC)*, 1997. 2

[MS08] Tal Moran and Gil Segev. David and Goliath commitments: UC computation for asymmetric parties using tamper-proof hardware. In *EUROCRYPT*, pages 527–544, 2008. 3

[MV03] Daniele Micciancio and Salil Vadhan. Statistical zero-knowledge proofs with efficient provers: lattice problems and more. In *Advances in Cryptology – CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 282–298. Springer, 2003. 2

[Oka96] Tatsuaki Okamoto. On relationships between statistical zero-knowledge proofs. In *Proceedings of the 28th Annual ACM Symposium on Theory of Computing (STOC)*, pages 649–658. ACM Press, 1996. 8

[OV08] Shien Jin Ong and Salil P. Vadhan. An equivalence between zero knowledge and commitments. In Ran Canetti, editor, *TCC*, volume 4948 of *Lecture Notes in Computer Science*, pages 482–500. Springer, 2008. 2

[Vad99] Salil P. Vadhan. *A Study of Statistical Zero-Knowledge Proofs*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 1999. 6, 8

[Vad06] Salil P. Vadhan. An unconditional study of computational zero knowledge. *SIAM Journal on Computing*, 36(4):1160–1214, 2006. Preliminary version in *FOCS'04*. 6, 8