Languages with Efficient Zero-Knowledge PCPs are in SZK

Mohammad Mahmoody[1,⋆] and David Xiao[2,⋆⋆]

[1] Cornell University
mohammad@cs.cornell.edu
[2] LIAFA
CNRS, Université Paris Diderot - Paris 7
dxiao@liafa.univ-paris-diderot.fr

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 of **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, that prepares all of its PCP queries before seeing any answers, unless $\mathbf{NP} \subseteq \mathbf{coAM}$ and the 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: Probabilistically Checkable Proofs, Statistical Zero-Knowledge.

1 Introduction

Since their inception, interactive proofs [GMR89, BM88] have had a transformative effect on theoretical computer science in general and the foundations of

⋆ Supported in part by NSF Awards CNS-1217821 and 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.

⋆⋆ Supported in part by the French ANR Blanc program under contract ANR-12-BS02-005 (RDAM) and the French ANR JCJC program under contract (ROMAnTIC).

A. Sahai (Ed.): TCC 2013, LNCS 7785, pp. 297–314, 2013.

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 follow 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 (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]. On the other hand, [IMS12] only requires zero-knowledge to hold for non-adaptive malicious verifiers, while we assume that the zero-knowledge property holds for general (adaptive) malicious verifiers. (This is natural, since if the honest verifier is adaptive then even honest-verifier zero-knowledge would require zero-knowledge against an adaptive verifier, namely the honest one.) Finally, we emphasize that Theorem 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. Thus, in the case of computational zero knowledge, the question is resolved in the positive direction (under believable computational assumptions). Similarly, it would be possible to get statistical zero-knowledge probabilistically checkable *arguments* (with soundness against computationally bounded stateful provers) if one can construct resettable statistical zero-knowledge arguments. The work of [GMOS07] shows the existence of a closely related object, namely *concurrent* statistical zero-knowledge arguments for all of **NP**. But recall that in this work, both the zero-knowledge and the soundness are statistical, and so these mentioned results do 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 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 on 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 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. We then compare it to the approach of [IMS12], which is restricted to non-adaptive verifiers, and highlight why our technique bypasses this barrier. In the following let us assume for notational simplicity that L is a language; the idea is identical for general promise problems.

2.1 Our Approach

If L has a ZK-PCP, 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 $x \notin L$.

Our proof will show that if in addition to making sure that the view is accepting we do some extra tests on the distribution of the simulated view, then this will allow us to decide L. Suppose for a moment that the ZK-PCP is deterministic, *i.e.* on an input x the prover deterministically generates a proof π. (Of course it is known that ZK with deterministic provers cannot exist for non-trivial languages [GO94]. We make this simplification here only to make our proof sketch easier to describe, and we will argue below how one can do away with this simplification.)

We will show that when the ZK-PCP is deterministic, it suffices to run the simulator and check that the generated view is accepting and to check some entropy-related properties of the view which in our case happen to be a computational task in **SZK**. Let $(\mathbf{r}, (\mathbf{q}_1, \mathbf{a}_1), \ldots, (\mathbf{q}_m, \mathbf{a}_m))$ be the distribution of views generated when running the simulator for the honest verifier. By the ZK property, this is statistically close to the view of an honest verifier interacting with the honest prover on YES (*i.e.* $x \in L$) instances. Let $\mathbf{j}$ be uniform in $[m]$ and consider the distribution $(\mathbf{q_j}, \mathbf{a_j})$.

We argue that to decide the language it suffices to check first that the simulated transcript is accepting, and second that $\mathrm{H}(\mathbf{a_j} \mid \mathbf{q_j})$ is small. On YES instances, the simulated transcript is almost surely accepting because of the ZK property, and furthermore $\mathrm{H}(\mathbf{a_j}|\mathbf{q_j}) = 0$ because the simulated proof is deterministic. On the other hand, on NO (*i.e.* $x \notin L$) instances, we will show that if $\mathrm{H}(\mathbf{a_j} \mid \mathbf{q_j})$ is sufficiently small, then the simulated transcript is statistically close to an interaction between an honest verifier and a proof sampled as follows: for each q, the corresponding answer bit of the proof is sampled according

to $\mathbf{a_j} \mid \mathbf{q_j} = q$. By the soundness condition of the ZK-PCP, it follows that the transcript must be rejecting with high probability. It is clear that checking that the simulated transcript is accepting is in **BPP**, while checking that $\mathrm{H}(\mathbf{a_j} \mid \mathbf{q_j})$ is small is a conditional entropy approximation problem, which is in **SZK** [Vad06].

To generalize the above argument to the case of randomized ZK-PCPs, we use the following argument (which is a stronger version of an argument that first appeared in [IMS12]): Any efficiently computable PCP (as a random variable describing its truth table) has polynomial entropy. Therefore if we repeat the honest verifier ℓ times where ℓ is a polynomial sufficiently larger than the size of the circuit computing the PCP, we can essentially "exhaust" the entropy of the proof observed by the next independent verification over the same PCP. This allows us to prove that $\mathrm{H}(\mathbf{a_j}|\mathbf{q_j})$ is small *conditioned on* the PCP answers observed in the first ℓ verifications. Interestingly, this argument when applied to *a random* query index $\mathbf{j}$ (which is the index distribution we use—see Lemma 6) is rather delicate and heavily relies on the fact that PCPs are fixed; the statement is not true for interactive proofs, where the answers may depend on, say, the order of the queries.

Finally we note that even after making sure that the simulator is choosing its PCP answers close to some fixed oracle, it still might be the case that for NO instances it does not run an honest execution of the verifier against this PCP and somehow manages to generate accepting views for NO instances as well. To complete the proof, one final technicality that we check is that the random coins $\mathbf{r}$ generated by the simulator are indeed close to uniform conditioned on the ℓ previously sampled views. (They are guaranteed to be so on YES instances by ZK, but may not be on NO instances.) This latter task is also reducible to the conditional entropy approximation problem.

Approach of [IMS12]. At a high level, in our work we show that deciding the language using the simulator can be done in **SZK** by a direct reduction to a problem in **SZK**. In contrast, [IMS12] tried to "extract" a PCP from the simulator and then run the honest verifier against this extracted PCP. Since the extraction process requires sampling from inefficiently samplable distributions, this task is accomplished with the aid of an all-powerful yet untrusted prover (this is how they obtain the conclusion that the language is in $\mathbf{AM} \cap \mathbf{coAM}$). This makes our approach conceptually different from the approach of [IMS12].

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 $\Delta(\mathbf{y}_x, \mathbf{z}_x) = \mathrm{negl}(|x|)$ for all $x \in \mathcal{I}$. 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 $\mathrm{I}(\mathbf{X}; \mathbf{Y} \mid \mathbf{Z}) = \mathrm{H}(\mathbf{X} \mid \mathbf{Z}) - (\mathbf{X} \mid \mathbf{YZ})$.

A language L is a *partition* of $\{0,1\}^*$ into $L^{\mathrm{Y}} = L$ and $L^{\mathrm{N}} = \{0,1\}^* \setminus L$.

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 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* $\overline{L}^{\mathrm{N}} = L^{\mathrm{Y}}$.
- Conjunction $L = L_1 \wedge L_2$ *of promise languages* L_1 *and* L_2*:*
  - $x = (x_1, x_2) \in L^{\mathrm{Y}}$ *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* $x_2 \in L_2^{\mathrm{N}}$.
- Disjunction $L = L_1 \vee L_2$ *of promise languages* L_1 *and* L_2*:*
  - $x = (x_1, x_2) \in L^{\mathrm{Y}}$ *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 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 $x \in L_1^{\mathrm{N}}$.*

Definition 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 $\{\boldsymbol{\pi}_x\}$ indexed by $x \in L$ 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 $\widehat{\pi}$: $\Pr_r[\mathsf{V}_r^{\widehat{\pi}}(x) = 0] \geq 2/3$.*

We call a PCP for problem $L \in \mathbf{NP}$ efficient*, if for all* $x \in L$ *and all* $\pi \in \mathrm{Supp}(\boldsymbol{\pi}_x)$*, 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. This only makes our negative results stronger than if we required C_π to depend uniformly on x. We also note that our negative result holds even using a weaker notion of completeness for PCPs in which $\Pr_r[\mathsf{V}^{\boldsymbol{\pi}}_r(x) = 1] \geq 2/3$ holds over the randomness of the verifier and the randomness of sampling the oracle $\boldsymbol{\pi}$. We use the above definition since the positive constructions of randomized PCPs do satisfy this stronger condition, and it is more convenient for amplifying the gap between the completeness and soundness errors.

Definition 4. *Let* $\Pi = (\{\boldsymbol{\pi}_{x\in L}\}, \mathsf{V})$ *be a PCP for the problem* L*.* 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:*

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

Note that $\widehat{\mathsf{V}}$ *only has oracle access to* $\pi \leftarrow \boldsymbol{\pi}_x$*, and the statistical indistinguishability should hold for large enough* $|x|$*. We call* Π perfect *ZK if the simulator's output distribution, conditioned on not aborting, is identically distributed to the view of the verifier* $\widehat{\mathsf{V}}$ *accessing* $\pi \leftarrow \boldsymbol{\pi}_{x\in L}$*.*

Non-uniformity vs. auxiliary input. By combining Definitions 3 and 4 one can obtain the definition of an *efficient ZK-PCP*. Note that, zero-knowledge with an "efficient prover" is typically defined using some auxiliary input given to the "prover", however, since here we prove a negative result using non-uniform efficiency (as in Definition 3) only makes our results stronger. In particular, if there exists an ensemble $\boldsymbol{\pi}_{x,w}$ of efficiently computable proofs that is zero-knowledge and depends on both $x \in L$ and some witness w for $x \in L$, one can always obtain a non-uniformly computable efficient ZK-PCP (according to our Definitions 3 and 4) by hardwiring, for every $x \in L$, the lexicographically first witness w into the efficient algorithm computing $\boldsymbol{\pi}_x$.

The definition of the complexity class **SZK** is indeed very similar to Definition 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.) Since we do not need the exact definition of the class **SZK**, here we only describe it at a high level.

Definition 5 (Complexity Class SZK). *The class* **SZK** *consists of promise problems which have an interactive (single prover) 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 1. *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 1. Then* $F(L_1, \ldots, L_K) \in \mathbf{SZK}$*.*

See Section 4.5 and Corollary 6.5.1 of [Vad99] for a proof.

3.1 Shannon Entropy and Related Computational Problems

Fact 2 (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* $\mathrm{I}(\mathbf{F}(\mathbf{X}); \mathbf{Y} \mid \mathbf{Z}) \leq \mathrm{I}(\mathbf{X}; \mathbf{Y} \mid \mathbf{Z})$.

Definition 6 (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* $(\mathbf{X}, \mathbf{Y})$*,* i.e. *this is the distribution of the output of* C *run over fresh randomness. Then given* (C, r) *we have:*

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

Lemma 2. *For any* $\epsilon > 1/\mathrm{poly}(n)$*,* $\mathrm{CEA}_\epsilon \in \mathbf{SZK}$.

Proof. We give a reduction from CEA_ϵ to CEA_1, 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 $i \in [1/\epsilon]$, $(\mathbf{X}^i, \mathbf{Y}^i)$ 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

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

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

Definition 7 (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. $(\mathbf{X}, \mathbf{Y}) \in \mathrm{CEB}_{\alpha,\beta}^{\mathrm{Y}}$ *iff* $\mathrm{H}(\mathbf{X} \mid \mathbf{Y}) \geq \alpha$.
2. $(\mathbf{X}, \mathbf{Y}) \in \mathrm{CEB}_{\alpha,\beta}^{\mathrm{N}}$ *iff* $\mathrm{H}(\mathbf{X} \mid \mathbf{Y}) \leq \beta$.

The following is immediate from Lemma 2:

Lemma 3. *For all functions* $\alpha(n), \beta(n)$ *uniformly computable in time* $\mathrm{poly}(n)$ *and satisfying* $\alpha(n) - \beta(n) > 1/\mathrm{poly}(n)$*,* $\mathrm{CEB}_{\alpha,\beta} \in \mathbf{SZK}$.

3.2 Statistical Distance vs Conditional Entropy

To prove our main result, we need to bound statistical distance using *conditional* (Shannon) entropy and vice versa. See the full version of the paper for proofs of the following two lemmas.

Lemma 4 (Conditional Entropy to Statistical Distance). *Suppose* $\mathrm{Supp}(\mathbf{X}) = \{0,1\}^n$. *Then it holds that* $\mathbb{E}_{Y \leftarrow \mathbf{Y}} \Delta((\mathbf{X} \mid Y), \mathbf{U}_n) \leq \sqrt{n - \mathrm{H}(\mathbf{X} \mid \mathbf{Y})}$.

Lemma 5 (Statistical Distance to Conditional Entropy). *For* $\epsilon \in [0,1]$ *let* $\mathrm{H}(\epsilon) = \epsilon \lg(1/\epsilon) + (1-\epsilon)\lg(1/1-\epsilon)$. *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* $|\mathrm{H}(\mathbf{X} \mid \mathbf{Y}) - \mathrm{H}(\mathbf{X}' \mid \mathbf{Y}')| \leq 4(\mathrm{H}(\epsilon) + \epsilon \cdot n)$.

4 Proving the Main Result

Theorem 3. *Suppose the promise problem* $L = (L^Y, L^N)$ *has a ZK-PCP* $\Pi = (\{\boldsymbol{\pi}_{x \in L}\}, \mathsf{V})$ *of entropy at most* $\mathrm{H}(\boldsymbol{\pi}_x) \leq \mathrm{poly}(|x|)$. *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 3. Fix such a ZK-PCP for L and let $\eta = H(\boldsymbol{\pi}_x) \leq \mathrm{poly}(n)$.

The first step of our proof is to define a verifier who can "exhaust" all of the entropy of the ZK-PCP so that the proof behaves essentially as if it were deterministic. We use the following verifier: 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 $\ell = \mathrm{poly}(n) \gg \eta$ 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 $\pi \leftarrow \boldsymbol{\pi}_x$ 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. As we mentioned in the introduction, we need to check three conditions: the simulator generates an accepting view, the entropy of a random answer in the view has low entropy given the query, and the distribution of the random coins in the view is uniform.

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 of input length n and size $|C| = \mathrm{poly}(n)$; 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 $\Pr[C(\mathbf{U}_n) = 1] \leq \beta$.

The parameters α and β could be functions of n, and it is easy to see that for efficiently computable α, β (given n) it holds that $D_{\alpha,\beta} \in \mathbf{BPP}$ if $\alpha - \beta > 1/\operatorname{poly}(n)$.

Reduction 4 (Main Reduction). *Given a parameter ℓ, we map $x \mapsto (C_1, C_2, C_3)$ as follows.*

1. *C_1 checks the uniformity of the random coins in the view. 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_x^{SIM} on r_{SIM} to get $(\nu^1, \ldots, \nu^\ell) = C_x^{\text{SIM}}(r_{\text{SIM}})$ and sets: $X_1 = r^i$ and $Y_1 = (\nu^1, \ldots, \nu^{i-1})$.*
2. *C_2 checks that the conditional entropy of a randomly chosen answer is low conditioned on the corresponding query. 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_x^{SIM} on r_{SIM} to get $(\nu^1, \ldots, \nu^\ell) = C_x^{\text{SIM}}(r_{\text{SIM}})$ and sets: $X_2 = a_j^i$ and $Y_2 = (\nu^1, \ldots, \nu^{i-1}, q_j^i)$. We emphasize the fact that while a_j^i, q_j^i appear in the output of C_2, the actual index j itself does* not *appear in the output.*
3. *C_3 checks that the view is accepting. C_3 operates as follows: on input (r_{SIM}, i), C_3 executes the circuit C_x^{SIM} on r_{SIM} to obtain $(\nu^1, \ldots, \nu^\ell) = C_x^{\text{SIM}}(r_{\text{SIM}})$, and output 1 iff ν^i is an accepting view of V.*

Claim. Reduction 4 is a Karp reduction from L (specified in Theorem 3) 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 3 Using Claim 4. By taking $\ell = 40m\eta$, it holds that $2m \cdot \eta/\ell < 1/20$ in Lemma 7 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 by studying each case of $x \in L^Y$ and $x \in L^N$ separately. We begin with a lemma that will be useful for the case $x \in L^Y$.

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 6. *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 that $0 = \mathrm{H}(\mathbf{a_j q_j} \mid \boldsymbol{\pi}) - \mathrm{H}(\mathbf{a_j q_j} \mid \boldsymbol{\pi})$, 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 have $\mathrm{H}(\mathbf{a_j} \mid \mathbf{q_j}) = \mathrm{I}(\mathbf{a_j q_j}; \boldsymbol{\pi}) - \mathrm{I}(\mathbf{q_j}; \boldsymbol{\pi}) \leq \mathrm{I}(\mathbf{a_j q_j}; \boldsymbol{\pi})$. 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 2 implies that

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

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 2) it holds that

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

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

Remark 1. We emphasize that if $\boldsymbol{\pi}$ was *stateful* (*i.e.* a "prover", rather than a "proof"), then Lemma 6 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 6 would not hold anymore. This is one place where we crucially use the fixed nature of a PCP.

Proof of Claim 4: The Case $x \in L^Y$. Here we would like to show that $(C_1 \in \mathrm{CEB}^{\mathrm{Y}}_{k-1/200,k-1/100})$ and $(C_2 \in \overline{\mathrm{CEB}}^{\mathrm{Y}}_{2\eta/\ell,1.1\eta/\ell})$ and $(C_3 \in D^{\mathrm{Y}}_{0.66,\beta})$. 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 to 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 5 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. Here we study the view of $\mathsf{V}^{[\ell]}$ while interacting with a proof generated according to the distribution $\boldsymbol{\pi}_x$ 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 $\boldsymbol{\pi}_x$. In this case, by an argument similar to [IMS12], one can show that

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

Proof.

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

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

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. 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 $(\boldsymbol{\pi}_x, \boldsymbol{\nu}^1, \ldots, \boldsymbol{\nu}^{i-1})$ be the joint distribution of an honest proof $\boldsymbol{\pi}_x$ and $i-1$ executions of the honest verifier $\mathsf{V}^1, \ldots, \mathsf{V}^{i-1}$ using proof $\boldsymbol{\pi}_x$. Apply Lemma 6 using the distribution over proofs given by $(\boldsymbol{\pi}_x \mid \nu^1, \ldots, \nu^{i-1})$, and with the honest verifier algorithm V^i as the query algorithm accessing the proof.

Using Claims 4 and 4, 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 $\boldsymbol{\pi}_x$, then we can apply Lemma 5 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 $C_3 \in D^{\mathrm{Y}}_{0.66,\beta}$.

Proof of Claim 4: The Case $x \in L^N$. Here we would like to show that $C_1 \in \mathrm{CEB}^{\mathrm{N}}_{k-1/200, k-1/100}$ or $C_2 \in \overline{\mathrm{CEB}}^{\mathrm{N}}_{2\eta/\ell, 1.1\eta/\ell}$ or $C_3 \in D^{\mathrm{N}}_{0.66,\beta}$. This follows from the following lemma.

Lemma 7. *Suppose* $x \in L^N$, $C_1 \notin \mathrm{CEB}^{\mathrm{N}}_{k-1/200,k-1/100}$, *and also that* $C_2 \notin \overline{\mathrm{CEB}}^{\mathrm{N}}_{2\eta/\ell,1.1\eta/\ell}$. *Then* $C_3 \in D^{\mathrm{N}}_{0.66,\beta}$ *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 the 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 $(r^i, (q^i_1, a^i_1), \ldots, (q^i_{j-1}, a^i_{j-1}))$ 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 $(q^i_j, a^i_j), \ldots, (q^i_m, a^i_m)$ 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. If $x \in L^N$, then $\Pr_{\mathsf{Ideal}}[\nu^i \text{ accepts }] \leq 1/3$.

Claim. 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. 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 7. Claims 4, 4, and 4 together imply that

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

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

Proof (Proof of Claim 4). 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 (Proof of Claim 4). If $C_1 \notin \mathrm{CEB}^{\mathrm{N}}_{k-1/200,k-1/100}$, then 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 4 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 (Proof of Claim 4). The only difference between Hyb_j and Hyb_{j+1} is the way they answer q^i_j. 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:

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

Taking an expectation over all $j \leftarrow [\ell]$ we conclude Claim 4 as follows.

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

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

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

References

[AH91] Aiello, W., Håstad, J.: 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 1987

[ALM+98] Arora, S., Lund, C., Motwani, R., Sudan, M., Szegedy, M.: Proof verification and the hardness of approximation problems. Journal of the ACM 45(3), 501–555 (1998); Preliminary version in FOCS 1992

[AS98] Arora, S., Safra, S.: Probabilistic checking of proofs: a new characterization of NP. Journal of the ACM 45(1), 70–122 (1998); Preliminary version in FOCS 1992

[BFL90] Babai, L., Fortnow, L., Lund, C.: Non-deterministic exponential time has two-prover interactive protocols. In: FOCS, pp. 16–25 (1990)

[BGKW88] Ben-Or, M., Goldwasser, S., Kilian, J., Wigderson, A.: Multi-prover interactive proofs: How to remove intractability assumptions. In: STOC, pp. 113–131 (1988)

[BHZ87] Boppana, R.B., Håstad, J., Zachos, S.: Does co-NP have short interactive proofs? Information Processing Letters 25, 127–132 (1987)

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

[CGGM00] Canetti, R., Goldreich, O., Goldwasser, S., Micali, S.: Resettable zero-knowledge (extended abstract). In: STOC, pp. 235–244 (2000)

[CGS08] Chandran, N., Goyal, V., Sahai, A.: New Constructions for UC Secure Computation Using Tamper-Proof Hardware. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 545–562. Springer, Heidelberg (2008)

[DFK+92] Dwork, C., Feige, U., Kilian, J., Naor, M., Safra, M.: Low Communication 2-Prover Zero-Knowledge Proofs for NP. In: Brickell, E.F. (ed.) CRYPTO 1992. LNCS, vol. 740, pp. 215–227. Springer, Heidelberg (1993)

[For89] Fortnow, L.: The complexity of perfect zero-knowledge. Advances in Computing Research: Randomness and Computation 5, 327–343 (1989)

[FRS94] Fortnow, L., Rompel, J., Sipser, M.: On the power of multi-prover interactive protocols. Theoretical Computer Science 134(2), 545–557 (1994)

[GIMS10] Goyal, V., Ishai, Y., Mahmoody, M., Sahai, A.: Interactive Locking, Zero-Knowledge PCPs, and Unconditional Cryptography. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 173–190. Springer, Heidelberg (2010)

[GIS+10] Goyal, V., Ishai, Y., Sahai, A., Venkatesan, R., Wadia, A.: Founding Cryptography on Tamper-Proof Hardware Tokens. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 308–326. Springer, Heidelberg (2010)

[GKR08] Goldwasser, S., Kalai, Y.T., Rothblum, G.N.: One-Time Programs. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 39–56. Springer, Heidelberg (2008)

[GMOS07] Goyal, V., Moriarty, R., Ostrovsky, R., Sahai, A.: Concurrent Statistical Zero-Knowledge Arguments for NP from One Way Functions. In: Kurosawa, K. (ed.) ASIACRYPT 2007. LNCS, vol. 4833, pp. 444–459. Springer, Heidelberg (2007)

[GMR89] Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof systems. SIAM Journal on Computing 18(1), 186–208 (1989); Preliminary version in STOC 1985

[GMW91] Goldreich, O., Micali, S., Wigderson, A.: 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 1986

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

[GOVW12] Garg, S., Ostrovsky, R., Visconti, I., Wadia, A.: Resettable Statistical Zero Knowledge. In: Cramer, R. (ed.) TCC 2012. LNCS, vol. 7194, pp. 494–511. Springer, Heidelberg (2012)

[IMS12] Ishai, Y., Mahmoody, M., Sahai, A.: On Efficient Zero-Knowledge PCPs. In: Cramer, R. (ed.) TCC 2012. LNCS, vol. 7194, pp. 151–168. Springer, Heidelberg (2012)

[Kat07] Katz, J.: Universally Composable Multi-party Computation Using Tamper-Proof Hardware. In: Naor, M. (ed.) EUROCRYPT 2007. LNCS, vol. 4515, pp. 115–128. Springer, Heidelberg (2007)

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

[Kol10] Kolesnikov, V.: Truly Efficient String Oblivious Transfer Using Resettable Tamper-Proof Tokens. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 327–342. Springer, Heidelberg (2010)

[KPT97] Kilian, J., Petrank, E., Tardos, G.: Probabilistically checkable proofs with zero knowledge. In: STOC: ACM Symposium on Theory of Computing (STOC) (1997)

[MS08] Moran, T., Segev, G.: David and Goliath Commitments: UC Computation for Asymmetric Parties Using Tamper-Proof Hardware. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 527–544. Springer, Heidelberg (2008)

[MV03] Micciancio, D., Vadhan, S.P.: Statistical Zero-Knowledge Proofs with Efficient Provers: Lattice Problems and More. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 282–298. Springer, Heidelberg (2003)

[OV08] Ong, S.J., Vadhan, S.P.: An Equivalence Between Zero Knowledge and Commitments. In: Canetti, R. (ed.) TCC 2008. LNCS, vol. 4948, pp. 482–500. Springer, Heidelberg (2008)

[Vad99] Vadhan, S.P.: A Study of Statistical Zero-Knowledge Proofs. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, USA (1999)

[Vad06] Vadhan, S.P.: An unconditional study of computational zero knowledge. SIAM Journal on Computing 36(4), 1160–1214 (2006); Preliminary version in FOCS 2004