Comparing Entropies in Statistical Zero Knowledge with Applications to the Structure of SZK

Oded Goldreich* Salil Vadhan†

Abstract

We consider the following (promise) problem, denoted ED *(for* Entropy Difference*): The input is a pair of circuits, and* YES *instances (resp.,* NO *instances) are such pairs in which the first (resp., second) circuit generates a distribution with noticeably higher entropy.*

On one hand we show that any language having a (honest-verifier) statistical zero-knowledge proof is Karp-reducible to ED*. On the other hand, we present a* public-coin *(honest-verifier) statistical zero-knowledge proof for* ED*. Thus, we obtain an alternative proof of Okamoto's result by which $\mathcal{HVSZK}$ (i.e., honest-verifier statistical zero knowledge) equals* public-coin *$\mathcal{HVSZK}$. The new proof is much simpler than the original one. The above also yields a trivial proof that $\mathcal{HVSZK}$ is closed under complementation (since* ED *easily reduces to its complement). Among the new results obtained is an equivalence of a weak notion of statistical zero knowledge to the standard one.*

1 Introduction

Zero-Knowledge proofs, introduced by Goldwasser, Micali and Rackoff [16], are fascinating and extremely useful constructs. Their fascinating nature is due to their seemingly contradictory nature; they are both convincing and yet yield nothing beyond the validity of the assertion being proven. Their applicability in the domain of cryptography is vast; they are typically used to force malicious parties to behave according to a predetermined protocol (which requires parties to provide proofs of the correctness of their secret-based actions without revealing these secrets).

*Department of Computer Science, Weizmann Institute of Science, Rehovot, ISRAEL. E-mail: oded@wisdom.weizmann.ac.il. Work done while visiting LCS, MIT. Supported by DARPA grant DABT63-96-C-0018.

†Laboratory for Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139. E-mail: salil@theory.lcs.mit.edu. Supported by a DOD/NDSEG Graduate Fellowship and in part by DARPA Grant DABT63-96-C-0018.

Zero-knowledge proofs come in many flavors. One central parameter is to the strength of the zero-knowledge (or simulability) condition: The requirement that the *verifier learns nothing* from the proof is formulated by saying that the transcript of its interaction with the prover can be *simulated* by the verifier itself. That is, there exists an efficient procedure that, when given a valid assertion as input, produces a distribution which is "similar" to the distribution of transcripts of the executions of the proof system on that assertion. The key parameter is the interpretation of "similarity". Three notions have been commonly considered in the literature (cf., [16, 10]). Perfect zero knowledge ($\mathcal{PZK}$) requires that the two distributions be identical. *Statistical zero knowledge* ($\mathcal{SZK}$) requires that these distributions be statistically close (i.e., the variation distance between them is negligible). Finally, computational zero knowledge ($\mathcal{CZK}$) refers to the case that these distributions are computationally indistinguishable (cf., [15, 26]).

This paper focuses on statistical zero knowledge ($\mathcal{SZK}$). This class has quite an intriguing status in complexity theory. On one hand, $\mathcal{SZK}$ contains several problems which are commonly believed to be hard (i.e., not in $\mathcal{BPP}$) such as Quadratic Residuosity [16], Graph Isomorphism [13], and a problem equivalent to the Discrete Logarithm Problem [12]. On the other hand, $\mathcal{SZK}$ lies quite low in the Polynomial-Time Hierarchy; specifically, it lies in the intersection $\mathcal{AM} \cap \mathrm{co}\mathcal{AM}$ (cf., [10, 1]). (Recall that $\mathcal{AM}$ denotes the class of two-round Arthur-Merlin proofs, which by [17] and [3] is equivalent to constant-round interactive proofs.) Furthermore, $\mathcal{SZK}$ has a (natural) complete problem [24] (and we are going to see another one in this paper).

Additional motivation for studying statistical zero knowledge comes from cryptography. For one, it offers a higher level of security than computational zero knowledge; that is, $\mathcal{SZK}$ provides information-theoretic (or absolute) security whereas $\mathcal{CZK}$ only provides computational security (i.e., security against adversaries of bounded computational resources). Another motivation for the study of $\mathcal{SZK}$ is that it provides a good test ground for developing techniques to study $\mathcal{CZK}$ proofs (cf., [21, 22, 6, 7, 14]). We note that although it is long known that $\mathcal{CZK} = \mathcal{IP}$ (provided one-way functions exist; cf., [13, 19, 4]), the development of methodologies for the construction of (efficient) zero-knowledge proof systems is still of great importance.

The study of $\mathcal{SZK}$ has gained much momentum in recent years. In particular, two results which assert transformations of one type of $\mathcal{SZK}$ proof system into another, have played an important role in recent study. A key notion in these results is the notion of *honest-verifier* $\mathcal{SZK}$, denoted $\mathcal{HVSZK}$. Unlike the general notion of $\mathcal{SZK}$, which requires that no matter what the verifier does, it learns nothing from the interaction with the prover, here one only requires that the "honest" verifier (i.e., one that follows the prescribed protocol) learns nothing from the interaction. The two results referred to above are:

Thm. I: Every promise problem[1] having a honest-verifier $\mathcal{SZK}$ proof system has also a *public-coin* honest-verifier $\mathcal{SZK}$ proof system (cf., Okamoto [20]).

Thm. II: Every promise problem having a *public-coin* honest-verifier $\mathcal{SZK}$ proof system has a (public-coin) general $\mathcal{SZK}$ proof system (cf., Goldreich, Sahai and Vadhan [14]).

Combining these two results one obtains that any promise problem having a honest-verifier $\mathcal{SZK}$ proof system also has a general $\mathcal{SZK}$ proof system (i.e., one in which zero-knowledge holds with respect to any cheating verifier). We stress the key role of Thm. I in providing the adequate starting point for Thm. II. Furthermore, the starting point provided by Thm. I is relied on also in the following intriguing results:

Thm. III: The class $\mathcal{HVSZK}$ is closed under complementation (cf., Okamoto [20]): That is, if a promise problem has a honest-verifier $\mathcal{SZK}$ proof system then so has its complement.

Thm. IV: The class $\mathcal{HVSZK}$ has a natural complete problem (cf., Sahai and Vadhan [24]).

Thus, Thm. I plays a key role in this area. Unfortunately, the proof of Thm. I in [20] is very complicated and was fully understood by very few researchers.

The primary motivation of this work is to provide a simpler proof of Thm. I. Our basic idea is to apply some of Okamoto's techniques [20] to the Aiello–Hastad transformation [1] of $\mathcal{HVSZK}$ into $\mathcal{AM}$, rather than applying them (as done in [20]) to the Goldwasser–Sipser transformation [17] of $\mathcal{IP}$ into $\mathcal{AM}$. To further clarify the proof, we introduce a new promise problem, and show that: (1) any problem in $\mathcal{HVSZK}$ reduces to the new promise problem, and (2) the new promise problem has a *public-coin* $\mathcal{HVSZK}$ proof system. Combining (1) and (2), Thm. I follows.

[1] A promise problem Π is a pair of disjoint sets of strings, corresponding to YES and NO instances, respectively [9].

1.1 Statistical zero-knowledge proof systems

Following [12], we extend the standard definition of interactive proof systems to promise problems $\Pi = (\Pi_{\text{YES}}, \Pi_{\text{NO}})$. That is, we require the completeness condition to hold for YES instances (i.e., $x \in \Pi_{\text{YES}}$), require the soundness condition to hold for NO instances (i.e., $x \in \Pi_{\text{NO}}$), and do not require anything for inputs which violate the promise (i.e., $x \notin \Pi_{\text{YES}} \cup \Pi_{\text{NO}}$).

This paper focuses on such proof systems which are honest-verifier statistical zero-knowledge:

Definition 1.1 (Honest-verifier statistical zero knowledge – $\mathcal{HVSZK}$): *Let (P, V) be an interactive proof system for a promise problem $\Pi = (\Pi_{\text{YES}}, \Pi_{\text{NO}})$.*

- *We denote by $\langle P, V\rangle(x)$ the* **view** *of the verifier V while interacting with P on common input x; this consists of the common input, V's internal coin tosses, and all messages it has received.*
- *(P, V) is said to be* **honest-verifier statistical zero knowledge** *if there exists a probabilistic polynomial-time machine* (called a simulator)*, S, and a negligible*[2] *function $\mu : \mathbb{N} \mapsto [0, 1]$* (called the simulator deviation) *so that for every $x \in \Pi_{\text{YES}}$ the statistical difference between $S(x)$ and $\langle P, V\rangle(x)$ is at most $\mu(|x|)$.*
- *$\mathcal{HVSZK}$ denotes the class of promise problems having honest-verifier statistical zero-knowledge interactive proof systems.*

We comment that general *statistical zero-knowledge* proof systems are such where the zero-knowledge requirement holds for any (polynomial-time computable) verifier stategy, rather than merely for the prescribed/honest verifier V. Actually, even a stronger requirement can be proven to be equivalent to $\mathcal{HVSZK}$ – see [14].

1.2 Public-coin versus general proof systems

Recall that *public-coin* (a.k.a Arthur-Merlin) proof systems [2, 3] are interactive proof systems [16] in which the prescribed verifier's strategy amounts to sending uniformly chosen messages at each round, and deciding whether to accept by evaluating a polynomial-time predicate of the conversation transcript. That is, in each round, the verifier tosses a predetermined number of coins and sends the outcome to the prover, and at the end it decides whether to accept by applying a predicate to the (full) sequence of messages it has sent and received.

Public-coin proof systems are easier to analyze and manipulate than general interactive proofs, and thus the

[2] Recall that a function $f\colon \mathbb{N} \to \mathbb{N}$ is *negligible* if for any polynomial $p(\cdot)$, $f(n) < 1/p(n)$ for sufficiently large n.

result of Goldwasser and Sipser [17] by which the former are as powerful as the latter found many applications (e.g., [11, 19, 4]). As mentioned above, the same and more so is true regarding statistical zero knowledge: That is, Okamoto's result [20] (i.e., Thm. I), by which public-coin $\mathcal{HVSZK}$ equals $\mathcal{HVSZK}$, has played a major role in subsequent results (e.g., Thms. II, III, and IVmentioned above). Thus, providing a clear proof of Thm. I is of major importance to this area.

1.3 A new $\mathcal{HVSZK}$-complete problem: Entropy Difference

The new promise problem referred to earlier is called `Entropy Difference`. The promise problem involves the entropies of distributions which are encoded by circuits which sample from them. That is, if X is a circuit mapping $\{0,1\}^m$ to $\{0,1\}^n$, we identify X with the probability distribution induced on $\{0,1\}^n$ by feeding X the uniform distribution on $\{0,1\}^m$. We write $\mathrm{H}(X)$ for the entropy of distribution X (defined in Section 2.1).

Definition 1.2 (Entropy Difference): *The promise problem* `Entropy Difference`*, denoted* $\mathtt{ED} = (\mathtt{ED}_{\mathrm{YES}}, \mathtt{ED}_{\mathrm{NO}})$*, consists of*

$$\mathtt{ED}_{\mathrm{YES}} \stackrel{\mathrm{def}}{=} \{(X,Y) : \mathrm{H}(X) > \mathrm{H}(Y)+1\}$$
$$\mathtt{ED}_{\mathrm{NO}} \stackrel{\mathrm{def}}{=} \{(X,Y) : \mathrm{H}(Y) > \mathrm{H}(X)+1\}$$

where X and Y are distributions encoded as circuits which sample from them.

As stated above, our main results are

Theorem 1.3 ($\mathcal{HVSZK}$-hardness): *Any promise problem in $\mathcal{HVSZK}$ reduces (via a Karp reduction) to* `ED`.

(Theorem 1.3 combined with a simple constant-round interactive proof for `ED` implies that $\mathcal{HVSZK} \subseteq \mathcal{AM} \cap \mathrm{co}\mathcal{AM}$. We believe that this provides a much simpler argument than the one presented in [10, 1], although it does use all the underlying ideas of these works.)[3]

Theorem 1.4 (ED in public-coin $\mathcal{HVSZK}$): `ED` *has a public-coin honest-verifier statistical zero-knowledge proof system.*

Combining Theorems 1.3 and 1.4,[4] we see that any promise problem in $\mathcal{HVSZK}$ has a public-coin $\mathcal{HVSZK}$ proof system. Thus, we provide an alternative (and much simpler) proof of Thm. I. Furthermore, observing that `ED` easily reduces to its complement, it follows that $\mathcal{HVSZK}$ is closed under complementation (i.e., we provide an alternative proof of Thm. III).

Discussion: Some superficial similarity does exist between the above and what was done in [24]. In the latter work, the authors defined a promise problem, called `Statistical Difference` (denoted `SD`),[5] and showed that it is complete for the class $\mathcal{HVSZK}$. However, their reduction of $\mathcal{HVSZK}$ to `SD` used Thm. I to restrict attention to public-coin $\mathcal{HVSZK}$ only. Thus, the results in [24] (relying on Thm. I) cannot be used to establish Thm. I. Furthermore, the $\mathcal{HVSZK}$ proof system for `SD` presented in [24] is not of the public-coin type.

In retrospect, the term *statistical* zero knowledge (coined by Goldwasser, Micali and Rackoff [16]) sounds prophetic of the key role played by computational problems regarding statistical measures in the study of this class (which is also known by the name "almost-perfect zero knowledge").

1.4 Extensions

Let us stress that by (honest-verifier) statistical zero knowledge we mean a simulation, up to negligible deviation error, by a *strict* (rather than expected) probabilistic polynomial-time machine. This makes Theorem 1.4 seemingly stronger, but potentially weakens Theorem 1.3. However, as we shortly explain, Theorem 1.3 is in fact stronger than stated.

Definition 1.5 (simulator deviation): *Let (P,V) be a proof system for a promise problem $\Pi = (\Pi_{\mathrm{YES}}, \Pi_{\mathrm{NO}})$, and let M be a probabilistic polynomial-time machine. Suppose that for some function $\epsilon : \mathbb{N} \mapsto [0,1]$ and every $x \in \Pi_{\mathrm{YES}}$ the statistical difference between the verifier's view, denoted $\langle P,V\rangle(x)$ and $M(x)$ is at most $\epsilon(|x|)$. Then we say that M* simulates (P,V) *with* **deviation** ϵ.

As defined above, $\mathcal{HVSZK}$ is the class of promise problems having interactive proofs with negligible simulator deviation. A weaker level of security (or zero-knowledge property) is provided by the notion of *weak-$\mathcal{HVSZK}$* (which is analogous to weak-$\mathcal{SZK}$ considered in, e.g., [8]):

[3] We note that much of the simplification is due to [23].

[4] Actually, we also use the fact that the reduction in Theorem 1.3 is not length-decreasing. Alternatively, one may use the fact that `ED` is easily padded to increase the length of instance descriptions.

[5] `Statistical Difference`, denoted $\mathtt{SD} = (\mathtt{SD}_{\mathrm{YES}}, \mathtt{SD}_{\mathrm{NO}})$, consists of

$$\mathtt{SD}_{\mathrm{YES}} \stackrel{\mathrm{def}}{=} \{(X,Y) : \Delta(X\ ,\ Y) < 1/3\}$$
$$\mathtt{SD}_{\mathrm{NO}} \stackrel{\mathrm{def}}{=} \{(X,Y) : \Delta(X\ ,\ Y) > 2/3\}$$

where X and Y are as in Definition 1.2, and $\Delta(X\ ,\ Y)$ denote the statistical difference between them (i.e., $\Delta(X\ ,\ Y) \stackrel{\mathrm{def}}{=} \frac{1}{2} \cdot \sum_\alpha |\Pr[X=\alpha] - \Pr[Y=\alpha]|$).

Definition 1.6 (weak-$\mathcal{HVSZK}$): *A proof system is said to be* **weak** (honest-verifier) statistical zero knowledge *if for every polynomial p there exists a probabilistic polynomial-time machine M_p which simulates the proof system with simulator deviation $1/p(\cdot)$.*

Specifically, the running-time of M_p may depend on p. Note that weak-$\mathcal{HVSZK}$ contains promise problems having $\mathcal{HVSZK}$ proof sytems under a liberal definition allowing *expected* polynomial-time simulators. That is, suppose that Π has an interactive proof system (P, V) and an *expected* polynomial-time simulator M which simulates (P, V) with negligible deviation. Then, for any polynomial p, we can construct a strict polynomial-time simulator M_p which simulates (P, V) with deviation $1/p(\cdot)$ simply by truncating long runs of M; that is, runs which take more than p times the expected number of steps. It follows that (P, V) is a weak-$\mathcal{HVSZK}$ proof system. All these variants of $\mathcal{HVSZK}$ are covered by the following extension of Theorem 1.3:

Theorem 1.7 (Theorem 1.3, extended): *Any promise problem in weak-$\mathcal{HVSZK}$ reduces* (via a Karp reduction) *to* ED.

In fact, the proof only utilizes a simulator with deviation smaller than the reciprocal of the (cube of the) total number of bits sent in the proof system. On the other hand, Theorem 1.4 can be strengthened as follows:

Theorem 1.8 (Theorem 1.4, extended): ED *has a public-coin proof system which can be simulated with exponentially vanishing deviation.*

Combining Theorems 1.7 and 1.8, we get

Corollary 1.9 *Every language in weak-$\mathcal{HVSZK}$ has a public-coin proof system which can be simulated with exponentially vanishing deviation.*

Using the results in [14] we infer that weak-$\mathcal{HVSZK}$ equals $\mathcal{SZK}$, where the latter refers to statistical zero knowledge against any verifier. Specifically,

Corollary 1.10 *Every language in weak-$\mathcal{HVSZK}$ has a* (public-coin) *general statistical zero-knowledge proof system. Furthermore, the latter can be simulated using a universal probabilistic polynomial-time simulator which uses any verifier strategy as a black-box and has only an exponentially vanishing deviation.*

1.5 Techniques

As stated above our main results are Theorems 1.3 and 1.4 which establish, respectively, a Karp reduction of $\mathcal{HVSZK}$ to ED, and a public-coin honest-verifier $\mathcal{SZK}$ proof system for ED.

The proof of the first main result relies on the works of Fortnow, Aiello and Hastad [10, 1]. The key observation underlying these works is that any simulator establishing the (honest-verifier) $\mathcal{SZK}$ property of a proof system must behave very differently on YES and NO-instances. This difference is used in [10, 1] in order to construct certain *constant-round* proof systems. We use this difference to construct a reduction to ED. Specifically, we use the characterization of the simulator's behavior as provided in [1] and further simplified in [23]. This characterization allows us to reduce instances of any problem in $\mathcal{HVSZK}$ to instances of ED.

The proof of the second main result relies on the work of Okamoto [20]. Specifically, we follow his basic idea of "complementary usage of messages" and use two of his subprotocols. We stress that we provide self-contained definitions, implementations and analysis of the latter two subprotocols.

1.6 Open Problems

Our proof of Thm. I (as well as the original proof of Okamoto [20]) actually provides a transformation of proof systems (from private-coin to public-coin while preserving a certain zero-knowledge property, namely $\mathcal{HVSZK}$). Neither our transformation nor Okamoto's preserves the number of rounds in the original proof system, nor the computational complexity of the prover. It would be desirable to present an alternative transformation which does preserve both complexity measures, and it would be of interest even to present a transformation which preserves only one of these measures.

For a wider perspective, we mention the following facts.

1. The transformation of private-coin interactive proofs to public-coin ones (cf., [17]) preserves the number of rounds (up to an additive constant), but does not preserve the computational complexity of the prover.

   (Note that this transformation does not seem to preserve any zero-knowledge property. Furthermore, it is not known how to transform computational zero-knowledge proofs into public-coin ones (without assuming the existence of one-way functions which allows one to construct the latter from scratch).)

2. The transformation of honest-verifier zero-knowledge public-coin proof systems into general zero-knowledge ones (cf., [14]) preserves the computational complexity of the prover and only increases mildly the round complexity.

   (Actually, this transformation preserves both measures, but introduces a noticeable soundness error which can be eliminated by repeating the proof system sequentially any non-constant number of times.)

1.7 Organization

In Section 2, we prove Theorem 1.3; that is, we show that every problem in $\mathcal{HVSZK}$ reduces to ED. In Section 3, we prove Theorem 1.4; that is, we exhibit a public coin statistical zero-knowledge proof system for ED. This proof system uses two subprotocols which are specified in Section 3 and implemented in Section 4.

2 $\mathcal{HVSZK}$ reduces to ED

In this section, we prove Theorems 1.3 and 1.7, which state that every problem in $\mathcal{HVSZK}$ (and weak-$\mathcal{HVSZK}$) reduces to ED. Our reduction is based on the Aiello–Hastad characterization of statistical zero-knowledge [1]. Following Petrank and Tardos [23], we present the Aiello–Hastad characterization using a formulation of entropy, rather than in the formulation of set sizes used in [1].

In Section 2.1, we define the information-theoretic notions used in the Aiello–Hastad characterization. In Section 2.2, we motivate and state the lemmas which comprise the Aiello–Hastad characterization (with proofs in Appendix A). In Section 2.3, we exhibit the reduction from any problem in $\mathcal{HVSZK}$ to ED, prove its correctness using the Aiello–Hastad characterization, and thereby deduce Theorems 1.3 and 1.7.

2.1 Entropy and Relative Entropy

Recall the definition of the *entropy*, denoted $\mathrm{H}(X)$, of a random variable X:

$$\begin{aligned}\mathrm{H}(X) &\stackrel{\text{def}}{=} \sum_\alpha \Pr[X=\alpha]\cdot\log(1/\Pr[X=\alpha]) \\ &= \mathrm{E}_{\alpha\sim X}\left[\log(1/\Pr[X=\alpha])\right],\end{aligned}$$

where all logarithms above and in the sequel are to base 2. The *binary entropy function*, $\mathrm{H}_2(p) \stackrel{\text{def}}{=} p\log(1/p) + (1-p)\log(1/(1-p))$, equals the entropy of a 0-1 random variable with expectation p.

We will make use of two measures of similarity between probability distributions. The first measure is the well-known statistical difference: The *statistical difference* between the random variables X and Y, denoted $\Delta(X\,,\,Y)$, is defined by

$$\begin{aligned}\Delta(X\,,\,Y) &\stackrel{\text{def}}{=} \frac{1}{2}\cdot\sum_\alpha |\Pr[X=\alpha]-\Pr[X=\alpha]| \\ &= \max_S\{\Pr[X\in S]-\Pr[Y\in S]\}\end{aligned}$$

The second measure is the Kullback–Leibler distance:

Definition 2.1 *Let X and Y be two probability distributions on a finite set D. The* **relative entropy** (or Kullback–Leibler distance) *between X and Y is defined as*

$$\mathrm{KL}(X\,|\,Y) = \mathrm{E}_{\alpha\sim X}\left[\log\frac{\Pr[X=\alpha]}{\Pr[Y=\alpha]}\right].$$

We let $\mathrm{KL}_2(p,q) \stackrel{\text{def}}{=} p\log(p/q) + (1-p)\log((1-p)/(1-q))$. Note that if X and Y are 0-1 random variables with expections p and q respectively, then $\mathrm{KL}(X\,|\,Y) = \mathrm{KL}_2(p,q)$. It can be shown that $\mathrm{KL}(X\,|\,Y)$ is always nonnegative and $\mathrm{KL}(X\,|\,Y)=0$ iff X and Y are identically distributed [5, Thm. 2.6.3]. Hence, $\mathrm{KL}(X\,|\,Y)$ can be viewed as some sort of "distance" between X and Y, though it does not satisfy symmetry or the triangle inequality.

2.2 The Aiello–Hastad Characterization

In this section, we motivate and state the lemmas which comprise the Aiello–Hastad characterization of statistical zero-knowledge. Proofs can be found in Appendix A.

Intution. Let Π be any language (or promise problem) in $\mathcal{HVSZK}$ and consider a statistical zero-knowledge proof system for Π and the corresponding simulator. We think of the output of the simulator as describing the moves of a *virtual prover* and a *virtual verifier*. Following Fortnow [10], the Aiello–Hastad characterization describes properties of the output of the simulator which distinguish between YES instances and NO instances. One thing we are guaranteed by the statistical zero-knowledge property is that the simulator outputs accepting conversations with high probability when the input is a YES instance. Thus, if on some input x, the simulator outputs rejecting or invalid conversations with high probability, x is easily identified to be a NO-instance. The difficulty comes from the fact that the simulator might output accepting conversations with high probability even when x is a NO-instance, even though this cannot occur when any real prover interacts with the true verifier due to the soundness of the proof system. Intuitively, this discrepancy comes from the fact that the virtual prover has the ability to cheat and "see" future verifier messages, a power which the real prover does not have. Thus, Aiello and Hastad consider what happens when one takes away the power of the virtual prover to cheat. That is, following [10], they consider a real prover strategy P_S, called the *simulation-based prover*, which determines its messages based on the same distribution as the virtual prover's residual probability space conditioned only on past messages. Now, the interaction between P_S and the real verifier describes exactly what happens when we take away the power of the simulated prover to cheat. Thus, the relative entropy between the output of S and the interaction between P_S and the real verifier is a measure of the amount of cheating that virtual prover performs, and this distinguishes between YES instances and NO instances. The final crucial observation in the Aiello–Hastad characterization is that this relative entropy can be rewritten as a simple expression involving entropies of prefixes of the simulator's output.

Notation. Let Π be any language (or promise problem) in $\mathcal{HVSZK}$ (or weak-$\mathcal{HVSZK}$) and let (P, V) be a statistical zero-knowledge proof system for Π with simulator S. Without loss of generality, we assume that on inputs of length n, the verifier tosses exactly $\ell = \ell(n)$ coins, and the interaction between P and V consists of $2r = 2r(n)$ messages, each of length $\ell = \ell(n)$ so that the prover's messages are those with odd index. Also, we may assume that the last message of the verifier consists of its random coins. We are interested in the random variables, $\langle P, V\rangle(x)$ and $S(x)$, describing the real interaction and the simulation, respectively. We also consider prefixes of these random variables, where $\langle P, V\rangle(x)_i$ and $S(x)_i$ denote the prefix of length $i \cdot \ell$ of the corresponding random variable. At times, we may drop x from these notations. We say that a $2r \cdot \ell$ bit string γ is a *transcript* (w.r.t V) if the verifier messages in γ correspond to what it would have sent given the random coins (as specified in the last bits in γ) and previous messages of the prover (included in γ). We say that a transcript γ is *accepting* if the verifier accepts on it.

The simulation-based prover. Given an execution prefix $\gamma \in \{0, 1\}^{(i-1)\ell}$, the simulation-based prover, denoted P_S, responses as follows:

- If $S(x)$ outputs conversations that begin with γ with probability 0, then P_S replies with a dummy message, say $0^{\ell(|x|)}$.
- Otherwise, P_S replies according with the same conditional probability as the prover in the output of the simulator. That is, it replies $\beta \in \{0, 1\}^{\ell(|x|)}$ with probability

$$p_\beta = \Pr[S(x)_i = \gamma\beta | S(x)_{i-1} = \gamma]$$

Following our previous notation, we denote conversation transcripts coming from the interaction between P_S and V by $\langle P_S, V\rangle(x)$, and its prefixes by $\langle P_S, V\rangle(x)_i$.

Rewriting $\mathrm{KL}\,(S(x)\,|\,\langle P_S, V\rangle(x))$**.** The Aiello–Hastad characterization uses the relative entropy between $S(x)$ and $\langle P_S, V\rangle(x)$ to distinguish between YES and NO instances. This relative entropy $\mathrm{KL}\,(S(x)\,|\,\langle P_S, V\rangle(x))$ can be rewritten as a simple expression referring only to entropies of prefixes of $S(x)$.

Lemma 2.2 (implicit in [1], explicit in [23]):

$$\begin{aligned} &\mathrm{KL}\,(S(x)\,|\,\langle P_S, V\rangle(x)) \\ &= \ell - \sum_{i=1}^{r} [\mathrm{H}(S(x)_{2i}) - \mathrm{H}(S(x)_{2i-1})] \end{aligned}$$

The behaviour of P_S on YES instances: Note that even in case of a YES instance, the behaviour of P_S need not *exactly* fit the behavior of either the prescribed prover P or the simulated prover (i.e., the distribution of prover messages in the output of the simulator) . Yet, in the case of YES instance, prover P_S behaves "almost" as P and the simulated prover. More generally,

Lemma 2.3 (implicit in [1, 23]): *Let* $\epsilon \stackrel{\text{def}}{=} \Delta(S(x)\,,\,\langle P, V\rangle(x))$ *and suppose that* $\epsilon \leq 1/2$. *Then,*

$$\mathrm{KL}\,(S(x)\,|\,\langle P_S, V\rangle(x)) \leq 3r^2 \cdot \ell \cdot \epsilon + 2r \cdot \mathrm{H}_2(\epsilon)$$

The behaviour of P_S on NO instances: In contrary to the above, for NO instances, if $S(x)$ outputs accepting transcripts with high probability then $S(x)$ and $\langle P_S, V\rangle(x)$ must be very different. More generally,

Lemma 2.4 (implicit in [1, 23]): *Let p denote the probability that $S(x)$ outputs an accepting transcript, and q be the maximum, taken over all possible provers P^*, that $\langle P^*, V\rangle(x)$ is accepting. Suppose that $p \geq q$. Then,*

$$\mathrm{KL}\,(S(x)\,|\,\langle P_S, V\rangle(x)) \geq \mathrm{KL}_2(p, q)$$

2.3 The Reduction

Using the above characterization, we easily Karp-reduce any promise problem Π in $\mathcal{HVSZK}$ (or weak-$\mathcal{HVSZK}$) to ED. Let (P, V) and S be a proof system and a simulator as formulated in the previous subsection (namely, the proof system consists of $2r$ messages of length ℓ and the verifier's last message consists of its random coins). Then, an instance x is reduced to a pair of distributions (X_x, Y_x) as follows:

- X_x is the cross product of the distributions $S(x)_2$, $S(x)_4$, ..., $S(x)_{2r}$.
- Y_x is the cross product of the distributions $S(x)_1$, $S(x)_3$, ..., $S(x)_{2r-1}$ and a uniform distribution on $\ell(|x|) - 2$ bits.

Lemma 2.5 (Validity of the reduction): *Suppose that S simulates a proof system (P, V) with soundness error*[6]

[6] Recall that the *soundness error* $s(n) \in [0, 1]$ of an interactive proof system (P, V) is an upper bound on the probability that the verifier accepts after interacting with any potential prover strategy P^* on input a NO instance of length n.

at most 0.1 *for* Π *with simulator deviation smaller than* $1/(2r\ell)^2$*. Further suppose that* S *always outputs an accepting transcript. Then,*

1. *If* $x \in \Pi_{\text{YES}}$ *then* $\mathrm{H}(X_x) > \mathrm{H}(Y_x) + 1$.
2. *If* $x \in \Pi_{\text{NO}}$ *then* $\mathrm{H}(Y_x) > \mathrm{H}(X_x) + 1$.

The extra condition (of always outputing an accepting transcript) can be easily enforced by a minor modification of the simulator (and possibly the proof systems). See details in the proof of Theorems 1.3 and 1.7 below.

Proof: We may assume that $r\ell > 128$, by simply padding messages with extra bits. Suppose first that $x \in \Pi_{\text{YES}}$. Combining Lemmas 2.2 and 2.3, we have

$$\begin{aligned}
&\mathrm{H}(Y_x) - \mathrm{H}(X_x) \\
&= \left(\ell - 2 + \sum_{i=1}^{r} \mathrm{H}(S(x)_{2i-1})\right) - \left(\sum_{i=1}^{r} \mathrm{H}(S(x)_{2i})\right) \\
&= \mathrm{KL}\left(S(x) \,|\, \langle P_S, V\rangle(x)\right) - 2 \\
&\le 3r^2\ell \cdot \epsilon + 2r \cdot \mathrm{H}_2(\epsilon) - 2 \;<\; -1
\end{aligned}$$

where $\epsilon \stackrel{\text{def}}{=} \Delta(S(x)\,,\, \langle P, V\rangle(x)) \le 1/(2r\ell)^2$, and the last inequality also uses $\mathrm{H}_2(\epsilon) \le \sqrt{\epsilon}/4$ (since $\epsilon < 2^{-14}$) and $\sqrt{\epsilon}/4 < 1/8r$. Thus, $\mathrm{H}(X_x) > \mathrm{H}(Y_x) + 1$ and $(X_x, Y_x) \in \mathtt{ED}_{\text{YES}}$ follows.

Suppose now that $x \in \Pi_{\text{NO}}$. Combining Lemmas 2.2 and 2.4, we have

$$\begin{aligned}
\mathrm{H}(Y_x) - \mathrm{H}(X_x) &= \mathrm{KL}\left(S(x) \,|\, \langle P_S, V\rangle(x)\right) - 2 \\
&\ge \mathrm{KL}_2(1, 0.1) - 2 \\
&= \log 10 \ - 2 \ > \ 1
\end{aligned}$$

(In the first inequality, we used $\mathrm{KL}\left(S(x) \,|\, \langle P_S, V\rangle(x)\right) > \mathrm{KL}_2(1, q)$, where q is the the maximum, taken over all possible provers P^*, that $\langle P^*, V\rangle(x)$ is accepting.) Thus, $\mathrm{H}(Y_x) > \mathrm{H}(X_x) + 1$ and $(X_x, Y_x) \in \mathtt{ED}_{\text{NO}}$ follows. ■

Proof of Theorems 1.3 and 1.7: Let Π be any promise problem in weak-$\mathcal{HVSZK}$ and consider any weak-$\mathcal{HVSZK}$ proof system for Π. Informally, by repeating the proof system $\mathrm{poly}(n)$ times (either sequentially or in parallel) and modifying the proof system and simulator slightly, we can easily satisfy the requirements of Lemma 2.5. Namely, we obtain a proof system with soundness error at most 0.1 in which the last message of the verifier consists of its random coins (as was required throughout the Aiello–Hastad characterization), together with a simulator which always outputs accepting transcripts and has simulator deviation at most $1/(2r\ell)^2$. Once these conditions are satisfied, Lemma 2.5 tells us that the map $x \mapsto (X_x, Y_x)$ is a Karp reduction from Π to $\mathtt{ED}$, yielding Theorem 1.7. Theorem 1.3 then follows as a special case. Below, we do the calculations in more detail to show that the original proof system need only have a simulator achieving deviation smaller than the reciprocal of the (cube of the) total number of bits sent in the proof system (plus the number of coins used by the verifier).

Suppose the proof system for Π consists of $2r' - 1$ messages of length m, and let $\ell' = \max(m, q)$, where q is the number of coins used by the verifier. Assume the proof system has completeness and sounded errors both bounded by 1/3 and simulator deviation $(r'\ell')^{-2} \cdot (\log r'\ell')^{-5}$. We now modify the proof system by having the verifier send the prover its coins at the end and modify the simulator accordingly. This does not affect the completeness error, soundness error, or simulator deviation. Now there are $2r'$ messages, each of length at most ℓ'. Repeating the proof system for k times (either sequentially or in parallel) and ruling by majority, we obtain two-sided error of $\exp(-\Omega(k))$. Using $k = \Theta(\log r'\ell')$ we obtain a proof system with total communication $2r\ell = O(r'\ell' \log r'\ell')$, two-sided error $(2r\ell)^{-2}/2$ and simulation error $(2r\ell)^{-2}/2$.

Next, modify the proof system so that $0^{2r\ell}$ becomes an accepting transcript, and modify the simulator so that it always outputs an accepting transcript (by possibly substituting the output with $0^{2r\ell}$). The resulting proof system has soundness error at most $2^{-\ell} + (2r\ell)^{-2}/2$, and the simulation error is at most $(2r\ell)^{-2}$. Assuming, without loss of generality, that $2^{-\ell} + (2r\ell)^{-2}/2 < 0.1$, we are in position to apply Lemma 2.5, and the theorems follow. ■

3 A public-coin $\mathcal{HVSZK}$ proof system for `ED`

In this section, we prove Theorems 1.4 and 1.8. That is, we present a public-coin honest-verifier statistical zero-knowledge proof system for `Entropy Difference` (`ED`). In presenting the proof system, we will use two subprotocols due to Okamoto [20], which we will describe in Section 4.

In Section 3.1, we give an overview of the proof system. In particular, as a motivation we start by treating a special case of `ED` in which all distributions are "flat" (i.e., uniform over some subset of their range). We conclude the overview by discussion of the ideas underlying the extension of this special case to the general one. In Section 3.2, we discuss a standard technique for "flattening" distributions, which is an essential part of the final proof system. In Section 3.3, we state the properties of Okamoto's subprotocols that are used in the proof system for `ED`; the actual description of these subprotocols and their proofs of correctness are deferred to Section 4. Finally, in Section 3.4, we give the proof system for `ED` and prove its correctness.

3.1 Overview

We begin with an exposition of the standard protocol for proving lower bounds on set sizes, which is the starting point for our proof system. We stress that all protocols described in this section (as well as in the rest of the paper) are public-coin protocols.

3.1.1 The standard lower bound protocol

Suppose S is some subset of $\{0,1\}^n$ and a prover M ("Merlin") wants to convince a verifier A ("Arthur") that $|S| \gg 2^m$. Assuming A has oracle access to a procedure which tests membership in S, there is a simple public-coin protocol which can be used to accomplish this task. The protocol was first described in [2, 17] and orginates with a lemma of Sipser [25]. For every pair of integers k and ℓ, let $\mathcal{H}_{k,\ell}$ be a family of 2-universal hash functions mapping $\{0,1\}^k$ to $\{0,1\}^\ell$.

Lower bound protocol (M, A), on input n and m (and membership oracle for $S \subset \{0,1\}^n$)

1. A selects h uniformly from $\mathcal{H}_{n,m}$ and sends h to M.
2. M selects x uniformly from $S \cap h^{-1}(0)$ (if this intersection is nonempty) and sends x to A.[7] If the intersection is empty, M sends `fail` to A.
3. A accepts if both $h(x) = 0$ and $x \in S$ and rejects otherwise.

The best analysis of the above protocol was provided in [1].

Lemma 3.1 Completeness: *If $|S| \geq 2^k \cdot 2^m$, then A accepts with probability at least $1 - 2^{-k}$.*

Soundness: *If $|S| \leq 2^{-k} \cdot 2^m$, then no matter what strategy M uses, A accepts with probability at most 2^{-k}.*

In fact, this protocol also has a sort of statistical zero-knowledge property. The property holds with respect to the inputs n and m, provided that $|S| \gg 2^m$ and that one is given a uniformly selected element of S.

Lemma 3.2 (implicit in [20]) *Let $\mathcal{H}$ be a 2-universal family of hash functions mapping a domain D to a range R. Let S be a subset of D such that $|R| \leq \epsilon \cdot |S|$. Then the following two distributions have statistical difference $\epsilon^{\Omega(1)}$:*

(A) *Choose h uniformly in $\mathcal{H}$, and x uniformly in $h^{-1}(0) \cap S$. Output (h, x).*[8]

(B) *Choose x uniformly in S, and h uniformly in $\{h' \in \mathcal{H} : h'(x) = 0\}$. Output (h, x).*

Think of $D = \{0,1\}^n$, $R = \{0,1\}^m$, and $\epsilon = 2^m/|S|$. Then, Distribution (A) corresponds to A's view of the execution of the protocol and Distribution (B) provides a simulation with deviation (at most) $(2^m/|S|)^{\Omega(1)}$ for it.

[7] Here 0 is a canonically fixed element of $\{0,1\}^m$.

[8] In case $h^{-1}(0) \cap S = \emptyset$ the output is defined to be a special failure symbol.

3.1.2 A simple case of ED

We now sketch how the above lower bound protocol can be used to give a public-coin $\mathcal{HVSZK}$ proof system for a simplified version of ED. We call a distribution X *flat* if all strings in the support of X have the same probability. That is, X is the uniform distribution on some subset of its domain. The simplifying assumptions we make are that we are working with a pair of distributions X and Y (encoded by circuits which sample from them) such that

1. X and Y are both flat.
2. $|\mathrm{H}(X) - \mathrm{H}(Y)| > k$, where k is the "security parameter."

Now, we want to give a statistical zero-knowledge protocol by which M can convince A to accept if $\mathrm{H}(X) > \mathrm{H}(Y) + k$ and M cannot convince A to accept if $\mathrm{H}(Y) < \mathrm{H}(X) + k$. Since X and Y are flat, they are uniform over subsets S_X and S_Y of their domain. By the definition of entropy, $|S_X| = 2^{\mathrm{H}(X)}$ and $|S_Y| = 2^{\mathrm{H}(Y)}$. So proving that $\mathrm{H}(X) \gg \mathrm{H}(Y)$ is equivalent to proving that $|S_X| \gg |S_Y|$. So, one approach would be to use the above lower bound protocol to prove a lower bound on $|S_X|$, and use an upper bound protocol with similar properties (cf., [10]) to prove an upper bound on $|S_Y|$. Note that this by itself would do for placing the simplified version of ED in $\mathcal{AM}$ (and similar ideas can be applied to the general version ED; see §3.1.3).

The problem with the above is that it requires the prover to reveal $\mathrm{H}(X)$ and $\mathrm{H}(Y)$ (or approximations of these quantities). In fact, the zero-knowledge properties asserted above are relative to the given/asserted lower bound, and do not seem to hold when the bound is not given. Indeed, there seems to be no efficient way for the verifier to approximate the size of S, even when given a membership oracle to S. To overcome this difficulty, we adopt a technique of Okamoto [20] (which he calls "complementary usage of messages").

Recall that we are given a circuit (which we also denote Y) which samples from Y, and let m denote the length of the input to this circuit. So, for any point y in the support of Y, we let $\Omega_Y(y) \subseteq \{0,1\}^m$ denote the set of inputs to the circuit which yield output y. Then, $\Pr[Y = y] = 2^{-m} \cdot |\Omega_Y(y)|$. Since Y is flat, we have

$$|\Omega_Y(y)| = 2^m \cdot \Pr[Y = y] = \begin{cases} 2^m \cdot 2^{-\mathrm{H}(Y)} & \text{if } y \in S_Y. \\ 0 & \text{otherwise.} \end{cases}$$

Thus, proving an upper bound on $\mathrm{H}(Y)$ is equivalent to proving a lower bound on $\Omega_Y(y)$ for any y in the support of Y.

The key observation is that for any $y \in S_Y$, $|S_X \times \Omega_Y(y)| = 2^{\mathrm{H}(X)+m-\mathrm{H}(Y)}$. So proving that $\mathrm{H}(X) \gg \mathrm{H}(Y)$ (which was our original goal) is equivalent to proving that $|S_X \times \Omega_Y(y)| \gg 2^m$. Now we've reduced the problem to proving a lower bound for a set size which we know (namely 2^m, which can be computed by just looking at the circuit which computes Y)! This gives rise to the following "zero-knowledge" protocol.

Proof system (M, A) for simple case of ED, on input (X, Y)
Let m denote the input length of Y, and n denote the output length of X.

1. M selects y distributed according to Y and sends y to A.
2. A selects a hash function h uniformly from $\mathcal{H}_{n+m,m}$ and sends h to M.
3. M selects (x, r) uniformly from $(S_X \times \Omega_Y(y)) \cap h^{-1}(0)$ and sends (x, r) to A.
4. A checks that $Y(r) = y$ and that $h(x, r) = 0$. If either does not hold, A rejects immediately and the protocol ends.
5. M selects q uniformly from $\Omega_X(x)$ and sends q to A.
6. A checks that $X(q) = x$ and accepts if this holds and rejects otherwise.

The last two steps in the above protocol are for M to prove that x is in fact in the support of X. Now it follows immediately from our earlier discussion and the completeness and soundness of the lower bound protocol that this protocol is also complete and sound.

1. Completeness: If $\mathrm{H}(X) > \mathrm{H}(Y) + k$ and X and Y are both flat, then A accepts with probability at least $1 - 2^{-k}$.
2. Soundness: If $\mathrm{H}(Y) < \mathrm{H}(X) + k$ and X and Y are both flat, then no matter what strategy M uses, A accepts with probability at most 2^{-k}.

The statistical zero-knowledge property of this proof system also follows readily from that of the lower bound protocol. Consider the following simulator:

Simulator for simplified ED proof system, on input (X, Y)

1. Choose q and r uniformly at random and let $x = X(q)$, $y = Y(r)$.
2. Choose h uniformly from $\{h \in \mathcal{H}_{n+m,m} : h(x, r) = 0\}$.
3. Output $(y, h, (x, r), q)$.

The deviation of this simulator can be analyzed as follows: The string y is clearly distributed identically in both the proof system and the simulator. In the simulator, conditioned on y, the pair (x, r) is selected uniformly from $S_X \times \Omega_Y(y)$, and then h is selected uniformly among those that map (x, r) to 0. In the protocol, conditioned on y, the function h is selected uniformly in $\mathcal{H}_{n+m,m}$ and then (x, r) is selected uniformly from $(S_X \times \Omega_Y(y)) \cap h^{-1}(0)$. Thus, by Lemma 3.2, it follows that if $\mathrm{H}(X) - \mathrm{H}(Y) > k$ (i.e., $|S_X \times \Omega_Y(y)| > 2^{m+k}$), then the distributions on $(y, h, (x, r))$ in the simulator and the proof system have statistical difference $2^{-\Omega(k)}$. Finally, conditioned on $(y, h, (x, r))$, the string q is selected uniformly from $\Omega_X(x)$ in both distributions, and so it does not increase the statistical difference.

3.1.3 Treating general instances of ED

There are several problems in generalizing the proof system of §3.1.2 to arbitrary instances of ED. Clearly, the simplifying assumptions we made will not hold in general. The assumption that $|\mathrm{H}(X) - \mathrm{H}(Y)| > k$ is easy to achieve. If we let X' (resp., Y') consist of k independent copies of X (resp., Y), then $\mathrm{H}(X') = k \cdot \mathrm{H}(X)$ (resp., $\mathrm{H}(Y') = k \cdot \mathrm{H}(Y)$). So, the difference in entropies is multiplied by k.

The assumption that X and Y are both flat presents more serious difficulties. As we will see, taking many independent copies of each distribution yields distributions that are "nearly flat" (in a sense to be made precise later), but the protocol still needs further modification to work with "nearly flat" rather than truly flat distributions. The first problem is that if Y is only nearly flat, then M may select y to be "too heavy" (i.e., y has probability much greater than $2^{-\mathrm{H}(Y)}$), allowing him too many choices for r and leading to violation of the soundness property. Similarly, although there are only about $2^{\mathrm{H}(X)}$ choices for x that have probability near $2^{-\mathrm{H}(X)}$, if X is only nearly flat, there may be many more choices for x (alas these are "too light" – i.e., have probability much smaller than $2^{-\mathrm{H}(X)}$). This too gives M too much freedom (this time in choice of x) and may lead to violation of the soundness property.

In order to solve these problems, we use two subprotocols of Okamoto [20]: The first is a "sample generation" protocol, which is a protocol for M and A to select a sample from a nearly flat distribution Y such that no matter what strategy M uses, the sample will not be too heavy. This will replace Step 1 in the proof system of §3.1.2, and guarantee that M does not have too much freedom in its choice of r (in Step 3). The second protocol is a "sample test" protocol, which is a way for M to prove that a sample x taken from a nearly flat distribution X is not too light. This will replace Steps 5 and 6 in the proof system of §3.1.2, and guarantee that M does not have too much freedom in its choice of x (in Step 3).

We stress that both of these subprotocols will be public-coin and will possess appropriate simulability properties to ensure that the resulting protocol for ED is a public-coin $\mathcal{HVSZK}$ proof system. In the rest of this section, we will specify the properties of these subprotocols, and formulate and analyze the proof system for ED assuming that these subprotocols exist. In Section 4, we present these subprotocols and prove that they have the asserted properties.

3.2 Flattening distributions

As a preliminary step towards treating the general instances of ED, we formulate the process of "flattening" distributions (i.e., making them "nearly flat" by taking many independent copies).

Definition 3.3 (heavy, light and typical elements): *Let X be a distribution, x an element possibly in its support, and Δ a positive real number. We say that x is* **Δ-heavy** (resp., **Δ-light**) *if* $\Pr[X = x] \geq 2^{\Delta} \cdot 2^{-\mathrm{H}(X)}$ (resp., $\Pr[X = x] \leq 2^{-\Delta} \cdot 2^{-\mathrm{H}(X)}$). *Otherwise, we say that x is* **Δ-typical**.

A natural relaxed definition of flatness follows. The definition links the amount of slackness allowed in "typical" elements with the probability mass assigned to non-typical elements.

Definition 3.4 (flat distributions): *A distribution X is called* **Δ-flat** *if for every $t > 0$ the probability that an element chosen from X is $t \cdot \Delta$-typical is at least $1 - 2^{-t^2+1}$.*

By straightforward application of Hoeffding Inequality (cf., Appendix C), we have

Lemma 3.5 (flattening lemma): *Let X be a distribution, k a positive integer, and $\otimes^k X$ denote the distribution composed of k independent copies of X. Suppose that for all x in the support of X it holds that $\Pr[X = x] \geq 2^{-m}$. Then $\otimes^k X$ is $\sqrt{k} \cdot m$-flat.*

The key point is that the entropy of $\otimes^k X$ grows linearly with k, whereas its deviation from flatness grows significantly more slowy (i.e., linear in $\sqrt{k}$) as a function of k.

3.3 Subprotocol specifications

Below (as above), all distributions are given in form of a circuit which generate them. The input to these protocols will consist of a distribution, denoted X. We will denote by m (resp., n) the length of the input to (resp., output of) the circuit generating the distribution X. In all protocols party A is required to run in polynomial-time (in length of the common input), which means in particular that the total number of bits exchanged in the interaction is so bounded.

Definition 3.6 (Sample Generation Protocol): *A public-coin protocol (M, A) is called a* **sample generation protocol** *if on common input a distribution X and parameters Δ, t, such that X is Δ-flat and $t \leq \Delta$,*[9] *the following holds:*

1. ("completeness"): *If both parties are honest then A's output will be $t \cdot \Delta$-typical with probability at least $1 - m \cdot 2^{-\Omega(t^2)}$.*
2. ("soundness"): *If A is honest then, no matter how M plays, A's output is $2\sqrt{t\Delta} \cdot \Delta$-heavy with probability at most $m \cdot 2^{-\Omega(t^2)}$.* (A may abort with no output.[10])
3. (strong "zero-knowledge"): *There exists a polynomial-time simulator S so that for every (X, Δ, t) as above, the following two distributions have statistical difference at most $m \cdot 2^{-\Omega(t^2)}$:*
   (A) *Execute (M, A) on common input (X, Δ, t) and output the view of A, appended by A's output.*
   (B) *Choose $x \sim X$ and output $(S((X, \Delta, t), x), x)$.*

The above zero-knowledge property is referred to as *strong* since the simulator cannot produce a view-output pair by first generating the view and then computing the corresponding output. Instead, the simulator is forced (by the explicit inclusion of x in Distribution (B)) to generate a consistent random view for a given random output (of A). We comment that the trivial protocol in which A uniformly selects an input r to the circuit X and reveals both r and the output $x = X(r)$ cannot be used since the simulator is only given x and it may be difficult to find an r yielding x in general. Still, a Sample Generation protocol is implicit in Okamoto's work [20] (where it is called a "Pre-test").

Theorem 3.7 (implicit in [20]) *There exists a public-coin sample generation protocol. Furthermore, the number of communication rounds in the protocol is linear in q.*

A proof of Theorem 3.7 is presented in Section 4.

Definition 3.8 (Sample Test Protocol): *A public-coin protocol (M, A) is called a* **sample test protocol** *if on common input a distribution X, a string $x \in \{0,1\}^n$ and parameters Δ, t, such that X is Δ-flat and $t \leq \Delta$, the following holds:*

1. ("completeness"): *If both parties are honest and x is $t \cdot \Delta$-typical then A accepts with probability at least $1 - m \cdot 2^{-\Omega(t^2)}$.*
2. ("soundness"): *If x is $6\sqrt{t\Delta} \cdot \Delta$-light and A is honest then, no matter how M plays, A accepts with probability at most $m \cdot 2^{-\Omega(t^2)}$.*
3. (weak "zero-knowledge"): *There exists a polynomial-time simulator S so that for every (X, Δ, t) as above and for every $t \cdot \Delta$-typical x, the following two distributions have statistical difference at most $m \cdot 2^{-\Omega(t^2)}$:*
   (A) *Execute (M, A) on common input (X, x, Δ, t) and output the view of A, prepended by x.*
   (B) *On input (X, x, Δ, t) and an auxiliary input r uniformly distributed in $\Omega_X(x)$, output $(x, S((X, x, \Delta, t), r))$.*

[9] The condition $t \leq \Delta$ is to simplify the error expressions and will always be satisfied in our applications.

[10] It will indeed do so if detecting cheating.

The above zero-knowledge property is referred to as *weak* since the simulator gets a random r giving rise to x (i.e., $x = X(r)$) as an auxiliary input (whereas A is only given x). We comment that a simple public-coin testing protocol exists in case one can approximate the size of $\Omega_X(x)$ and uniformly sample from it. However, this may not be the case in general. Still, a Sample Testing protocol is implicit in Okamoto's work [20] (where it is called a "Post-test").

Theorem 3.9 (implicit in [20]) *There exists a public-coin sample testing protocol. Furthermore, the number of communication rounds in the protocol is linear in q.*

A proof of Theorem 3.9 is presented in Section 4.

3.4 The protocol for ED

We assume, without loss of generality, that the number of input (resp., output) bits of X equals the number for Y (e.g., by augmenting one circuit by dummy input or output bits). Let m and n denote the corresponding quantities. Furthermore, let s denote the total length of the description of both X and Y. The first step in the following protocol is an "amplification step" which yields distributions which are adequately flat. The protocol uses subprotocols for Sample Generation and Sample Testing as guaranteed by Theorems 3.7 and 3.9, respectively.

Proof system (M, A) for ED, on input (X, Y)

1. Both A and M set $V = \otimes^k X$ and $W = \otimes^k Y$, where $k \stackrel{\text{def}}{=} 2^{16} \cdot m^6 \cdot s$.
2. The parties utilize a Sample Generation protocol, with inputs $(W, \sqrt{k} \cdot m, \sqrt{s})$, obtaining an output denoted w.
3. Party A uniformly selects $h \in \mathcal{H}_{kn+km,km}$, and sends it to M.
4. M selects (v, r) from the distribution $V \times \Omega_W(w)$[11] conditioned on $h(v, r) = 0$, and sends (v, r) to A.
5. A checks that $W(r) = w$ and that $h(v, r) = 0$. If either does not hold, A rejects immediately and the protocol ends.
6. The parties utilize a Sample Test protocol, with inputs $(V, v, \sqrt{k} \cdot m, \sqrt{s})$, and A accepts iff the test was concluded satisfactorily.

We first show that the amplification step (i.e., Step 1) is indeed appropriate. That is,

Fact 3.10 *Distributions V and W are $\sqrt{k} \cdot m$-flat.*

[11] Here, and in the rest of the paper, we write use the same notation for a set (e.g., $\Omega_W(w)$) and the uniform distribution on that set.

Fact 3.10 is immediate by Lemma 3.5 and the setting of the parameters. Given Fact 3.10, we turn to the essence of the analysis of the protocol. The completeness property of the protocol will follow from the zero-knowledge one, and so we start by establishing the soundness property.

Lemma 3.11 (soundness): *Suppose that* $\mathrm{H}(Y) > \mathrm{H}(X) + 1$. *Then A accepts with probability at most* $\exp(-\Omega(s))$.

Proof: By the hypothesis we have $\mathrm{H}(W) > \mathrm{H}(V) + k$. By Fact 3.10, both distributions are Δ-flat, with $\Delta = \sqrt{k} \cdot m = 2^8 m^4 \sqrt{s}$. Observe that the Sample Generation and Testing subprotocols are invoked with parameters $t = \sqrt{s}$ and $\Delta = \sqrt{k} \cdot m$. Thus, the soundness condition of the Sample Generation protocol implies that with probability at most $km \cdot \exp(-\Omega(t^2)) = \exp(-\Omega(s))$ the outcome, w, is $2\sqrt{t\Delta} \cdot \Delta$-heavy.

Suppose that w is not $2\sqrt{t\Delta} \cdot \Delta$-heavy. Then we claim that M will be forced to select a v that is $6\sqrt{t\Delta} \cdot \Delta$-light with probability at least $1 - \exp(-\Omega(s))$. By Lemma 3.1, it suffices to show that the number of pairs (v, r) such that $W(r) = w$ and v is not $6\sqrt{t\Delta} \cdot \Delta$-light is at most $2^{-\Omega(s)} \cdot 2^{km}$. Since w is not $2\sqrt{t\Delta} \cdot \Delta$-heavy, there are at most $2^{km - \mathrm{H}(W) + 2\sqrt{t\Delta} \cdot \Delta}$ values of r such that $W(r) = w$. In addition, the number of non-$6\sqrt{t\Delta} \cdot \Delta$-light choices for v is at most $2^{\mathrm{H}(V) + 6\sqrt{t\Delta} \cdot \Delta}$ (as each such v has probability at least $2^{-6\sqrt{t\Delta} \cdot \Delta} \cdot 2^{-\mathrm{H}(V)}$ under V). Thus, the total number of pairs (v, r) such that $W(r) = w$ and v is not $6\sqrt{t\Delta} \cdot \Delta$-light is at most

$$\begin{aligned} & 2^{km - \mathrm{H}(W) + 2\sqrt{t\Delta} \cdot \Delta} \cdot 2^{\mathrm{H}(V) + 6\sqrt{t\Delta} \cdot \Delta} \\ = \; & 2^{8\sqrt{t\Delta} \cdot \Delta + \mathrm{H}(V) - \mathrm{H}(W)} \cdot 2^{km}. \end{aligned}$$

However, by our hypothesis and our setting of parameters

$$\begin{aligned} 8\sqrt{t\Delta} \cdot \Delta + \mathrm{H}(V) - \mathrm{H}(W) &< 8\sqrt{t\Delta} \cdot \Delta - k \\ &= (8 \cdot 2^{12} - 2^{16}) \cdot m^6 s \\ &< -s. \end{aligned}$$

Thus, by Lemma 3.1, the probability that M can return a suitable non-$6\sqrt{t\Delta} \cdot \Delta$-light v in Step 4 is at most $\exp(-\Omega(s))$. On the other hand, if M returns a $6\sqrt{t\Delta} \cdot \Delta$-light v then the probability that it will be accepted by the Sample Test is at most $km \cdot \exp(-\Omega(t^2)) = \exp(-\Omega(s))$. The claim follows. ■

Simulator for the above protocol, on input (X, Y)

1. Set $V = \otimes^k X$ and $W = \otimes^k Y$, where $k \stackrel{\text{def}}{=} 2^{16} \cdot m^6 \cdot s$.
2. Select uniformly $r', r \in \{0, 1\}^{km}$, and let $v = V(r')$ and $w = W(r)$.

3. Simulate an execution of the Sample Generation protocol on input $((W, \sqrt{k} \cdot m, \sqrt{s}), w)$, obtaining a view, denoted α, ending with output w.
4. Party A uniformly selects $h \in \mathcal{H}_{kn+km,km}$ so that $h(v, r) = 0$.[12]
5. Simulate an execution of the Sample Generation protocol on input $(V, v, \sqrt{k} \cdot m, \sqrt{s})$ and auxiliary input r', obtaining a view, denoted β.
6. Output $((\alpha, w), h, (v, r), \beta)$.

The correctness of this simulator will rely on the following variant of the Leftover Hash Lemma [18], proved in Appendix D.

Lemma 3.12 (implicit in [20]) *Let $\mathcal{H}$ be a 2-universal family of hash functions mapping a domain D to a range R and let 0 be any fixed element of R. Let Z be a distribution on D such that with probability $1 - \delta$ over z selected according to Z, $\Pr[Z = z] \le \varepsilon/|R|$. Then the following two distributions have statistical difference at most $3(\delta + \varepsilon^{1/3})$:*

(A) Choose h uniformly in $\mathcal{H}$. Select z according to Z conditioned on $h(z) = 0$. Output (h, z).

(B) Choose z according to Z. Select h uniformly in $\{h' \in \mathcal{H} : h(z) = 0\}$. Output (h, z).

Lemma 3.13 (zero-knowledge and completeness): *Suppose that $\mathrm{H}(X) > \mathrm{H}(Y) + 1$. Then the statistical difference between the view of the verifier on common input (X, Y) and the output of the simulator on input (X, Y) is at most $\exp(-\Omega(s))$. Furthermore, with probability at least $1 - \exp(-\Omega(s))$, the simulator generates an accepting transcript, and so in the real interaction the verifier accepts with probability at least $1 - \exp(-\Omega(s))$.*

Proof: Analogously to the proof of Lemma 3.11, we note that both V and W are Δ-flat, for $\Delta = 2^8 m^4 \sqrt{s}$, and we have $\mathrm{H}(V) > \mathrm{H}(W) + k$.

By the strong zero-knowledge property of the Sample Generation protocol, the pair (α, w) in the output of the simulator has statistical difference at most $km \cdot 2^{-\Omega(s)} = 2^{-\Omega(s)}$ from a real execution of that protocol. Since W is Δ-flat, the string w is $t\Delta$-light with probability at most $2^{-\Omega(s)}$ in the simulator. Thus, we consider the distributions on $(h, (v, r))$ conditioned on any pair (α, w) such that w is not $t\Delta$-light. To analyze this, we apply Lemma 3.12 with $Z = V \times \Omega_W(w)$, $D = \{0,1\}^{kn+km}$, and $R = \{0,1\}^{km}$. Distribution (A) (resp., (B)) in Lemma 3.12 corresponds to the distribution of $(h, (v, r))$ in the proof system (resp., simulator). Since V is Δ-flat, the following holds with probability $\ge 1 - 2^{-s+1}$ over (v, r) selected according to $V \times \Omega_W(w)$:

$$
\begin{aligned}
&\Pr[V \times \Omega_W(w) = (v, r)] \\
&= \Pr[V = v] \cdot \frac{1}{|\Omega_W(w)|} \\
&< 2^{-\mathrm{H}(V) + t\Delta} \cdot \frac{1}{2^{km - \mathrm{H}(W) - t\Delta}} \\
&< \frac{2^{-k + 2t\Delta}}{|R|} \\
&= \frac{2^{-2^{16} m^6 s + 2 \cdot 2^8 m^4 s}}{|R|} \\
&\le \frac{2^{-s}}{|R|}
\end{aligned}
$$

Thus, we can take $\delta = 2^{-s+1}$ and $\varepsilon = 2^{-s}$ in Lemma 3.12, and see that the two distributions on $(h, (v, r))$ have statistical difference $2^{-\Omega(s)}$ (conditioned on history (α, w)). Finally, including β only increases the statistical difference by $2^{-\Omega(s)}$ by the weak zero-knowledge property of the Sample Test protocol (noting that in the simulator, v is $t\Delta$-light with probability at most 2^{-s+1} and r is distributed uniformly in $\Omega_V(v)$). ■

Lemmas 3.11 and 3.13 and the fact that the given proof system is public coin immediately imply Theorem 1.8. Theorem 1.4 then follows as a special case. Actually, we can strengthen Theorem 1.8 somewhat by applying a transformation of [11] which converts public-coin honest-verifier statistical zero-knowledge proofs into ones with perfect completeness (i.e., the verifier accepts with probability 1 on YES instances). Their transformation also preserves an exponentially small soundness error and an exponentially small simulator deviation. Thus, we obtain:

Corollary 3.14 ED *has a public-coin proof system which has perfect completeness and exponentially small soundness error, and can be simulated with exponentially vanishing deviation.*

4 The Sample Generation and Test Protocols

In this section, we present Okamoto's protocols for generating and testing samples from a nearly flat distribution. Recall that these protocols must be public coin and furthermore must satisfy certain "zero-knowledge" properties.

4.1 Overview

Sample Generation. Here the input to the protocol (M, A) is a Δ-flat distribution X (encoded by a circuit) and the output should be a sample x from this distribution. We require that, no matter what strategy M follows, x will not

[12] This step can be efficiently implemented for all popular constructions of 2-universal families (e.g., the linear transformations family). Also note that by the 2-universal property of such families, functions mapping any fixed string to 0 always exist.

be too heavy. If, however, both parties play honestly, then x should be nearly typical with high probability, and should be simulatable for an *externally specified* x. In particular, the protocol should not reveal an input to the circuit X that yields x, as the simulator is only given x and it may be difficult to find an input yielding x in general. If we remove this condition, the problem becomes trivial: A could just sample x according to X and reveal both x and the input used to produce it. Since X is nearly flat, x will be nearly typical with high probability.

Okamoto's solution to this problem has the following general structure: M proposes a sample x (which is supposed to be distributed according to X) and sends it to A. (Of course, if M is dishonest, he can choose x to be too heavy.) Then M and A engage in a short "game" which ends by M proposing another sample x'. Roughly speaking, this game has the following properties:

1. If x is too heavy, then no matter what strategy M follows, he will be forced to select x' which is noticeably lighter than x.
2. If x is not too heavy, then no matter what strategy M follows, he will be forced to choose x' that is also not too heavy.
3. If x is nearly typical and M plays honestly, then x' will also be nearly typical.
4. If M plays honestly, then A's view of the game is simulatable for an externally specified x'.

Clearly, repeating this game many times to obtain a sequence of samples $x_0, \ldots, x_m$ (where x_0 is proposed by M and $x_{i+1} = x'_i$) will have the effect of pushing a heavy proposal for x_0 closer and closer to the nearly typical set. Taking m sufficiently large (but still polynomial in the appropriate parameters), x_m will be guaranteed to be not too heavy, no matter how M plays. On the other hand, if M plays honestly, all the samples will be nearly typical. Finally, the simulability property of the game enables the entire Sample Generation protocol to be simulated "backwards" for an externally specified x_m.

Sample Test. Here the input to the protcol (M, A) is a Δ-flat distribution X (encoded by a circuit) together with a string x from the domain of X. At the end of the protocol, A accepts or rejects. We require that if x is too light, A should reject with high probabability. If, however, x is nearly typical and both parties play honestly, then A should accept with high probability, and, moreover, A's view of the interaction should be simulatable (given additionally a random input for X which yields x).

The general structure of this protocol is very similar to that of the Sample Generation protocol. Given x, M and A engage in a short game which ends by M proposing another sample x'. Roughly speaking, this game has the following properties:

1. If x is too light, then no matter what strategy M follows, he will be forced to select x' which is noticeably lighter than x.
2. If x is nearly typical and M plays honestly, then x' will also be nearly typical.
3. If both parties play honestly, then A's view of the game is simulatable (given a random input to X which yields x).

Clearly, repeating this game many times to obtain a sequence $x_0, \ldots, x_m$ (where $x_0 = x$ and $x_{i+1} = x'_i$) will have the effect of making a light input sample lighter and lighter. Taking m sufficiently large, x_{m-1} will be so light that it has zero probability, so there is no x_m lighter than x_{m-1} and A will reject! Notice that we do not care what happens in the pushing game if x_i is not too light and M plays dishonestly; if the original input is too light (which is the the only time we worry about a dishonest M), all the subsequent x_i's will also be too light with high probability. On the other hand, if the original input x is nearly typical and M plays honestly, all the samples will be nearly typical. Finally, the simulability property of the game enables the entire Sample Generation protocol to be simulated "forwards" given coins for x. Amazingly, the game used for the Sample Test protocol is identical to the game used for the Sample Generation protocol. We describe this "pushing" game in the next section, and subsequently give formal descriptions of the two protocols.

4.2 The pushing game

Throughout the remainder of Section 4, X is a Δ-flat distribution encoded by a circuit and m (resp., n) denotes the length of the input (resp., output) of the circuit generating X. Recall that for positive integers k and ℓ, $\mathcal{H}_{k,\ell}$ denotes a 2-universal family of hash functions mapping $\{0,1\}^k$ to $\{0,1\}^\ell$.

The basic game underlying the Sample Generation and Sample Test protocols is the following 1-round protocol (called "sequentially recursive hashing" in [20]):

Pushing game (M, A), on input (X, x, Δ, t), where $x \in \{0,1\}^n$ and $t \leq \Delta$

1. A chooses h uniformly from $\mathcal{H}_{m+n,m-3t\Delta}$ and sends h to M.
2. M chooses (r, x') from the distribution $\Omega_X(x) \times X$, conditioned on $h(r, x') = 0$, and sends (r, x') to A. (If there is no such pair (r, x'), then M sends `fail` to A.)

3. A checks that $X(r) = x$ and $h(r, x') = 0$. If both conditions hold, A outputs x'. Otherwise A rejects.

Observe that if $|\Omega_X(x)| = \emptyset$, then A rejects with probability 1. In order to describe remaining the properties of the pushing game, we define the *weight* of a string x relative to a circuit X by $\mathrm{wt}_X(x) = \log(\Pr[X = x] \cdot 2^{\mathrm{H}(X)})$. So, x is γ-heavy iff $\mathrm{wt}_X(x) \geq \gamma$ and x is γ-light iff $\mathrm{wt}_X(x) \leq -\gamma$. Also note that for x in the support of X, $|\mathrm{wt}_X(x)| \leq m$. When the distribution X is clear from the context, we will often write $\mathrm{wt}(x)$ instead of $\mathrm{wt}_X(x)$. The following lemma asserts that no matter how M plays, if the input to the game is atypical, then the output is noticeably lighter. (The behavior on typical inputs is analyzed later — in Lemma 4.2.)

Lemma 4.1 *If A follows the prescribed strategy in the pushing game, then no matter what strategy M uses, the following hold:*

1. *("heavy gets lighter") With probability $\geq 1 - 2^{-\Omega(t^2)}$, either $\mathrm{wt}(x') < \max(\mathrm{wt}(x) - 1, 2\sqrt{t\Delta})$ or A rejects.*

2. *("light gets lighter") If $\mathrm{wt}(x) \leq -6\sqrt{t\Delta}\cdot\Delta$, then with probability $\geq 1 - 2^{-\Omega(t^2)}$, either $\mathrm{wt}(x') < \mathrm{wt}(x) - 1$ or A rejects.*

Proof: 1. Let S be the set of x' such that $\mathrm{wt}(x') \geq \max(\mathrm{wt}(x) - 1, 2\sqrt{t\Delta} \cdot \Delta)$. We need to show that with probability at most $2^{-\Omega(t^2)}$ over the choice of h from $\mathcal{H}_{m+n,m-3t\Delta}$, there exists a pair $(r, x') \in \Omega_X(x) \times S$ such that $h(x, r') = 0$. By the soundness of the standard lower-bound protocol (Lemma 3.2), it suffices to prove that

$$|\Omega_X(x) \times S| \leq 2^{-\Omega(t^2)} \cdot 2^{m-3t\Delta}.$$

The intuition is that the number of x' that are heavier than $\max(\mathrm{wt}(x) - 1, 2\sqrt{t\Delta}\cdot\Delta)$ is so small that not even the size of $\Omega_X(x)$ can compensate.

By definition of $\mathrm{wt}(x)$, $|\Omega_X(x)| = 2^{m-\mathrm{H}(X)+\mathrm{wt}(x)}$. We now bound $|S|$. First, since X is Δ-flat, we have

$$\begin{aligned} 2^{-4t\Delta+1} &\geq \Pr_{x'\sim X}\left[\mathrm{wt}(x') \geq 2\sqrt{t\Delta}\cdot\Delta\right] \\ &\geq \Pr[X \in S] \\ &= \sum_{x' \in S} \Pr[X = x'] \end{aligned}$$

On the other hand, every $x' \in S$ is $(\mathrm{wt}(x) - 1)$-heavy, so $\Pr[X = x'] \geq 2^{-\mathrm{H}(X)+\mathrm{wt}(x)-1}$. Thus,

$$2^{-4t\Delta+1} \geq |S| \cdot 2^{-\mathrm{H}(X)+\mathrm{wt}(x)-1}.$$

Putting everything together, we have

$$\begin{aligned} |\Omega_X(x) \times S| &\leq 2^{m-\mathrm{H}(X)+\mathrm{wt}(x)} \cdot \left(\frac{2^{-4t\Delta+1}}{2^{-\mathrm{H}(X)+\mathrm{wt}(x)-1}}\right) \\ &= 2^{m-4t\Delta+2} \\ &\leq 2^{-t^2+2} \cdot 2^{m-3t\Delta}, \end{aligned}$$

as desired. (In the last inequality, we used the fact that $t \leq \Delta$.)

2. Let $S = \{x' : \mathrm{wt}(x') \geq \mathrm{wt}(x) - 1\}$. Again, it suffices to show that $|\Omega_X(x) \times S| \leq 2^{-\Omega(t^2)} \cdot 2^{m-3t\Delta}$. Here the intuition is that $|\Omega_X(x)|$ is so small (since x is so light) that the only way for M to succeed is to choose x' even lighter than x (since there cannot be too many strings of noticeable probability mass). This time we bound $|S|$ by dividing S into two parts. Define

$$\begin{aligned} S_1 &= \{x' : \mathrm{wt}(x) - 1 \leq \mathrm{wt}(x') \leq -2\sqrt{t\Delta}\cdot\Delta\} \\ S_2 &= \{x' : -2\sqrt{t\Delta}\cdot\Delta < \mathrm{wt}(x')\}, \end{aligned}$$

so that $S = S_1 \cup S_2$. Since every $x' \in S_2$ has probability mass greater than $2^{-\mathrm{H}(X)-2\sqrt{t\Delta}\cdot\Delta}$, we must have

$$\begin{aligned} |S_2| &< 2^{\mathrm{H}(X)+2\sqrt{t\Delta}\cdot\Delta} \\ &\leq 2^{\mathrm{H}(X)-\mathrm{wt}(x)-4t\Delta}, \end{aligned}$$

where the last inequality follows from $\mathrm{wt}(x) \leq -6\sqrt{t\Delta}\cdot\Delta$ and $\Delta \geq t$. We now bound $|S_1|$. Since X is Δ-flat, we have

$$\begin{aligned} 2^{-4t\Delta+1} &\geq \Pr[X' \in S_1] \\ &\geq |S_1| \cdot 2^{-\mathrm{H}(X)+\mathrm{wt}(x)-1}. \end{aligned}$$

Thus, $|S_1| \leq 2^{\mathrm{H}(X)-\mathrm{wt}(x)-4t\Delta+2}$, and so

$$|S| = |S_1| + |S_2| < 2^{\mathrm{H}(X)-\mathrm{wt}(x)-4t\Delta+3},$$

and

$$\begin{aligned} |\Omega_X(x) \times S| &\leq 2^{m-\mathrm{H}(X)+\mathrm{wt}(x)} \cdot 2^{\mathrm{H}(X)-\mathrm{wt}(x)-4t\Delta+3} \\ &= 2^{m-4t\Delta+3} \\ &\leq 2^{-t^2+3} \cdot 2^{m-3t\Delta}, \end{aligned}$$

as desired. ■

The pushing game has the following simulability and "completeness" properties when both parties are honest:

Lemma 4.2 *If both parties follow the protocol in the pushing game and x is $t\Delta$-typical, then the following two distributions have statistical difference at most $2^{-\Omega(t^2)}$:*

(A) Execute the pushing game on input (X, x, Δ, t) to obtain (h, r, x'). Output (h, r, x').

(B) Let x' be distributed according to X and let r be selected uniformly from $\Omega_X(x)$. Choose h uniformly in $\mathcal{H}_{m+n,m-3t\Delta}$ subject to $h(r, x') = 0$. Output (h, r, x').

Proof: We apply Lemma 3.12 with $Z = \Omega_X(x) \times X$, $D = \{0,1\}^{m+n}$ and $R = \{0,1\}^{m-3t\Delta}$. Distribution (A) (resp., (B)) in Lemma 3.12 corresponds to Distribution (A) (resp., (B)) above. Since X is Δ-flat, the following holds with probability $\geq 1-2^{-t^2+1}$ over (r,x') selected according to $\Omega_X(x)\times X$:

$$\begin{aligned}\Pr\left[\Omega_X(x)=(r,x')\right] &= \Pr\left[X=x'\right]\cdot\frac{1}{|\Omega_X(x)|}\\ &< 2^{-\mathrm{H}(X)+t\Delta}\cdot\frac{1}{2^{m-\mathrm{H}(X)-t\Delta}}\\ &= \frac{2^{-t\Delta}}{|R|}\end{aligned}$$

Thus, we can take $\delta=2^{-t^2+1}$ and $\varepsilon=2^{-t\Delta}\leq 2^{-t^2}$ in Lemma 3.12, and see that the two distributions have statistical difference $2^{-\Omega(t^2)}$. ■

4.3 The protocols

The sample generation and test protocols simply consist of many repetitions of the basic pushing game:

Sample Generation Protocol (M,A), on input (X,Δ,t), where $t\leq\Delta$

1. M selects $x_0\in\{0,1\}^n$ according to X and sends x_0 to A.
2. Repeat for i from 1 to m: M and A execute the Pushing Game on input (X,x_{i-1},Δ,t) and let x_i be the output.
3. A outputs x_m unless it rejected in one of the Pushing Games, in which case it rejects.

Sample Test Protocol (M,A), on input (X,x,Δ,t), where $x\in\{0,1\}^n$ and $t\leq\Delta$

1. Let $x_0=x$.
2. Repeat for i from 1 to $m+1$: M and A execute the Pushing Game on input (X,x_{i-1},Δ,t) and let x_i be the output.
3. A rejects if it rejected in any of the Pushing Games, else it accepts.

4.4 Correctness of Sample Generation Protocol

Using the properties of the Pushing Game, we now prove that the Sample Generation Protocol satisfies Definition 3.6 and thus Theorem 3.7 holds.

Soundness. By Lemma 4.1 (Part 1) and induction, we see that for every $0\leq i\leq m$, with probability at least $1-i\cdot 2^{-\Omega(t^2)}$, either $\mathrm{wt}(x_i)<\max(\mathrm{wt}(x_0)-i,2\sqrt{t\Delta})$ or A rejects. In particular, since $\mathrm{wt}(x_0)\leq m$, with probability at least $1-m\cdot 2^{-\Omega(t^2)}$, we have

$$\mathrm{wt}(x_m)<\max(\mathrm{wt}(x_0)-m,2\sqrt{t\Delta}\cdot\Delta)=2\sqrt{t\Delta}\cdot\Delta$$

unless A rejects, as desired.

Completeness and Zero-Knowledge. First we observe that the completeness condition follows from the strong zero-knowledge condition: In Distribution (B) of Definition 3.6, x is distributed according to X, and hence is $t\Delta$-typical with probability $\geq 1-2^{-t^2+1}$ by the Δ-flatness of X. Since x corresponds to the output of the Sample Generation protocol in Distribution (A) and Distributions (A) and (B) have statistical difference at most $2^{-\Omega(t^2)}$, the output of the Sample Generation Protocol must be $t\Delta$-typical with probability at least $1-2^{-t^2+1}-2^{-\Omega(t^2)}=1-2^{-\Omega(t^2)}$.

Now we prove the zero-knowledge condition. Consider the following probabilistic polynomial-time simulator:

Simulator for Sample Generation Protocol, on input $((X,\Delta,t),x)$

1. Let $x_m=x$.
2. For i from m down to 1 repeat:
   (a) Choose r_{i-1} uniformly from $\{0,1\}^m$ and let $x_{i-1}=X(r_{i-1})$.
   (b) Choose h_i uniformly from $\mathcal{H}_{m+n,m-3t\Delta}$ subject to $h_i(r_{i-1},x_i)=0$.
3. Output $(x_0,h_1,(r_0,x_1),h_2,(r_1,x_2),\ldots,h_m,(r_{m-1},x_m))$.

We prove by induction on i that the distribution on $t_i=(x_0,h_1,(r_0,x_1),\ldots,h_i,(r_{i-1},x_i))$ in the output of the simulator (when x is chosen according to X) has statistical difference at most $i\cdot 2^{-\Omega(t^2)}$ from the verifier's view of the Sample Generation protocol up to the end of the i'th execution of the Pushing Game. Clearly this is true for $i=0$, as in both cases x_0 is distributed according to X. Now suppose it is true for i; we will prove it for $i+1$. From the following two observations it follows that the statistical difference only increases by $2^{-t^2+1}+2^{-\Omega(t^2)}=2^{-\Omega(t^2)}$ when going from i to $i+1$:

1. In the simulator, x_i is $t\Delta$-typical with probability at least $1-2^{-t^2+1}$.
2. For any history

   $$t_i=(x_0,h_1,(r_0,x_1),\ldots,h_i,(r_{i-1},x_i))$$

   in which x_i is $t\Delta$-typical, the following two distributions have statistical difference $2^{-\Omega(t^2)}$:
   (A) A's view of the $(i+1)$'st Pushing Game conditioned on history t_i.
   (B) The distribution of $(h_{i+1},(r_i,x_{i+1}))$ conditioned on history t_i in the output of the simulator.

Observation 1 is immediate from the fact that x_i is distributed according to X in the simulator and X is Δ-flat. Observation 2 follows from Lemma 4.2, observing that conditioned on history t_i, the triple $(h_{i+1}, (r_i, x_{i+1}))$ in the output of the simulator is selected exactly according to the Distribution (B) in Lemma 4.2. That is, conditioned on history t_i, r_i is selected uniformly from $\Omega_X(x_i)$, x_{i+1} is distributed according to X, and h is selected uniformly in $\mathcal{H}_{m+n,m-3t\Delta}$ subject to $h(r_i, x_{i+1}) = 0$.

4.5 Correctness of Sample Test Protocol

Finally, we prove that the Sample Test Protocol satisfies Definition 3.8 and thus Theorem 3.9 holds.

Soundness. By Lemma 4.1 (Part 2) and induction, we see that if $\mathrm{wt}(x) \leq -6\sqrt{t\Delta} \cdot \Delta$, then with probability at least $1 - i \cdot 2^{-\Omega(t^2)}$, for every $0 \leq i \leq m+1$, $\mathrm{wt}(x_i) < \mathrm{wt}(x_0) - i$ (or A rejects). In particular, since $\mathrm{wt}(x_0) < \mathrm{H}(X)$, with probability at least $1 - m \cdot 2^{-\Omega(t^2)}$, we have $\mathrm{wt}(x_m) < \mathrm{H}(X) - m$ unless A rejects at some iteration. Since $m - \mathrm{H}(X) + \mathrm{wt}(x_m) = \log|\Omega_X(x_m)|$ cannot be negative unless $|\Omega_X(x_m)| = \emptyset$, it follows that with probability at least $1 - m \cdot 2^{-\Omega(t^2)}$, A must reject in one of the iterations.

Completeness and Zero-Knowledge. First we prove the zero-knowledge condition. Consider the following probabilistic polynomial-time simulator:

Simulator for Sample Test Protocol, on input $((X, x, \Delta, t), r)$

1. Let $x_0 = x$ and $r_0 = r$.
2. For i from 1 to m repeat:
   (a) Choose r_i uniformly from $\{0,1\}^m$ and let $x_i = X(r_i)$.
   (b) Choose h_i uniformly from $\mathcal{H}_{m+n,m-3t\Delta}$ subject to $h_i(r_{i-1}, x_i) = 0$.
3. Output

$$(x_0, h_1, (r_0, x_1), h_2, (r_1, x_2), \ldots, h_{m+1}, (r_m, x_{m+1})).$$

We prove by induction on i that the distribution on $t_i = (x_0, h_1, (r_0, x_1), \ldots, h_i, (r_{i-1}, x_i))$ in the output of the simulator (when r is selected uniformly from $\Omega_X(x)$ and x is $t\Delta$-typical) has statistical difference at most $i \cdot 2^{-\Omega(t^2)}$ from the verifier's view of the Sample Test protocol up to the end of the i'th execution of the Pushing Game. Clearly this is true for $i = 0$. The induction step is proved analogously to the argument used for the Sample Generation Protocol, using the same two observations and noting that, although the simulator works in reverse order, the selection of r_i and h_i is as before.

Now we observe that the completeness condition follows from the weak zero-knowledge condition and the particular simulator we have given above. Specifically, the above simulator always outputs transcripts which would make A accept. Since it has statistical difference at most $m \cdot 2^{-\Omega(t^2)}$ from the Sample Test protocol, A must accept in the Sample Test protocol with probability at least $1 - m \cdot 2^{-\Omega(t^2)}$.

Acknowledgments

We thank Amit Sahai for many discussions about [20] and collaboration at an early stage of this research. We also thank an anonymous reviewer for helpful comments on the presentation.

References

[1] W. Aiello and J. Håstad. Statistical zero-knowledge languages can be recognized in two rounds. *Journal of Computer and System Sciences*, 42(3):327–345, June 1991.

[2] L. Babai. Trading group theory for randomness. In *Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing*, pages 421–429, 1985.

[3] L. Babai and S. Moran. Arthur-Merlin games: A randomized proof system and a hierarchy of complexity classes. *Journal of Computer and System Sciences*, 36:254–276, 1988.

[4] M. Ben-Or, O. Goldreich, S. Goldwasser, J. Håstad, J. Kilian, S. Micali, and P. Rogaway. Everything provable is provable in zero-knowledge. In S. Goldwasser, editor, *Advances in Cryptology—CRYPTO '88*, volume 403 of *Lecture Notes in Computer Science*, pages 37–56. Springer-Verlag, 1990, 21–25 Aug. 1988.

[5] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley Series in Telecommunications. John Wiley & Sons, Inc., 2nd edition, 1991.

[6] I. Damgård. Interactive hashing can simplify zero-knowledge protocol design. In *Proceedings of Crypto '95, Lecture Notes in Computer Science*, volume 403, pages 100–109. Springer-Verlag, 1994.

[7] I. Damgård, O. Goldreich, T. Okamoto, and A. Wigderson. Honest verifier vs. dishonest verifier in public coin zero-knowledge proofs. In *Proceedings of Crypto '95, Lecture Notes in Computer Science*, volume 403. Springer-Verlag, 1995.

[8] G. Di Crescenzo, T. Okamoto, and M. Yung. Keeping the SZK-verifier honest unconditionally. In B. S. Kaliski Jr., editor, *Advances in Cryptology—CRYPTO '97*, volume 1294 of *Lecture Notes in Computer Science*, pages 31–45. Springer-Verlag, 17–21 Aug. 1997.

[9] S. Even, A. L. Selman, and Y. Yacobi. The complexity of promise problems with applications to public-key cryptography. *Information and Control*, 61(2):159–173, May 1984.

[10] L. Fortnow. The complexity of perfect zero-knowledge. In S. Micali, editor, *Advances in Computing Research*, volume 5, pages 327–343. JAC Press, Inc., 1989.

[11] M. Fürer, O. Goldreich, Y. Mansour, M. Sipser, and S. Zachos. On completeness and soundness in interactive proof systems. In S. Micali, editor, *Advances in Computing Research*, volume 5, pages 429–442. JAC Press, Inc., 1989.

[12] O. Goldreich and E. Kushilevitz. A perfect zero-knowledge proof system for a problem equivalent to the discrete logarithm. *Journal of Cryptology*, 6:97–116, 1993.

[13] O. Goldreich, S. Micali, and A. Wigderson. Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. *Journal of the Association for Computing Machinery*, 38(1):691–729, 1991.

[14] O. Goldreich, A. Sahai, and S. Vadhan. Honest-verifier statistical zero-knowledge equals general statistical zero-knowledge. In *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing*, pages 399–408, 1998.

[15] S. Goldwasser and S. Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, 1984.

[16] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*, 18(1):186–208, February 1989.

[17] S. Goldwasser and M. Sipser. Private coins versus public coins in interactive proof systems. In S. Micali, editor, *Advances in Computing Research*, volume 5, pages 73–90. JAC Press, Inc., 1989.

[18] R. Impagliazzo, L. A. Levin, and M. Luby. Pseudo-random generation from one-way functions (extended abstracts). In *Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing*, pages 12–24, Seattle, Washington, 15–17 May 1989.

[19] R. Impagliazzo and M. Yung. Direct minimum-knowledge computations (extended abstract). In C. Pomerance, editor, *Advances in Cryptology—CRYPTO '87*, volume 293 of *Lecture Notes in Computer Science*, pages 40–51. Springer-Verlag, 1988, 16–20 Aug. 1987.

[20] T. Okamoto. On relationships between statistical zero-knowledge proofs. In *Proceedings of the Twenty Eighth Annual ACM Symposium on the Theory of Computing*, 1996. See also preprint of full version, Oct. 1997.

[21] R. Ostrovsky. One-way functions, hard on average problems, and statistical zero-knowledge proofs. In *Proceedings of the Thirty Second Annual Symposium on Foundations of Computer Science*, pages 133–138, 1991.

[22] R. Ostrovsky and A. Wigderson. One-way functions are essential for non-trivial zero-knowledge. In *Proceedings of the Second Israel Symposium on Theory of Computing and Systems*, 1993.

[23] E. Petrank and G. Tardos. On the knowledge complexity of NP. In *Proceedings of the Thirty Seventh Annual Symposium on Foundations of Computer Science*, pages 494–502, 1996.

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

[25] M. Sipser. A complexity theoretic approach to randomness. In *Proceedings of the Fifteenth Annual ACM Symposium on Theory of Computing*, pages 330–335, Boston, Massachusetts, 25–27 Apr. 1983.

[26] A. C. Yao. Theory and application of trapdoor functions. In *Proceedings of the Twenty Third Annual Symposium on Foundations of Computer Science*, pages 80–91, 1982.

A The Aiello–Hastad Characterization – Further Details

Proof of Lemma 2.2: For readability, we will omit x in the notation. For $\gamma \in \{0,1\}^{2r\ell}$ and $i = 0, ..., 2r$, we let γ_i denote the $i \cdot \ell$ prefix of γ. Then, by definition,

$$\begin{aligned} &\mathrm{KL}\left(S \mid \langle P_S, V\rangle\right) \\ &\quad = \sum_{\gamma\in\{0,1\}^{2r\ell}} \Pr\left[S=\gamma\right] \cdot \log \frac{\Pr\left[S=\gamma\right]}{\Pr\left[\langle P_S, V\rangle = \gamma\right]}. \end{aligned}$$

We can rewrite the fraction above as follows:

$$\begin{aligned} &\frac{\Pr\left[S=\gamma\right]}{\Pr\left[\langle P_S, V\rangle = \gamma\right]} \\ &\quad = \frac{\prod_{i=1}^{2r} \Pr\left[S_i = \gamma_i | S_{i-1} = \gamma_{i-1}\right]}{\prod_{i=1}^{2r} \Pr\left[\langle P_S, V\rangle_i = \gamma_i | \langle P_S, V\rangle_{i-1} = \gamma_{i-1}\right]} \\ &\quad = \frac{\prod_{j=1}^{r} \Pr\left[S_{2j} = \gamma_{2j} | S_{2j-1} = \gamma_{2j-1}\right]}{\prod_{j=1}^{r} \Pr\left[\langle P_S, V\rangle_{2j} = \gamma_{2j} | \langle P_S, V\rangle_{2j-1} = \gamma_{2j-1}\right]} \end{aligned}$$

A key observation is that the denominator in the above fraction equals the reciprocal of the number of possible outcomes of the verifier coins (i.e., $2^{-\ell}$), since even-indexed messages of $\langle P_S, V\rangle$ are generated by V exactly as in $\langle P, V\rangle$. Multiplying both the numerator and denominator in the above fraction by $\prod_{j=1}^{r} \Pr\left[S_{2j-1} = \gamma_{2j-1}\right]$, we obtain

$$\frac{\Pr\left[S=\gamma\right]}{\Pr\left[\langle P_S, V\rangle = \gamma\right]} = \frac{\prod_{j=1}^{r} \Pr\left[S_{2j} = \gamma_{2j}\right]}{2^{-\ell} \cdot \prod_{j=1}^{r} \Pr\left[S_{2j-1} = \gamma_{2j-1}\right]},$$

and thus

$$\begin{aligned} &\mathrm{KL}\left(S \mid \langle P_S, V\rangle\right) \\ &\quad = \ell + \sum_{j=1}^{r} \sum_{\gamma\in\{0,1\}^{2r\ell}} \Pr\left[S=\gamma\right] \cdot \log \Pr\left[S_{2j} = \gamma_{2j}\right] \\ &\qquad \sum_{j=1}^{r} \sum_{\gamma\in\{0,1\}^{2r\ell}} \Pr\left[S=\gamma\right] \cdot \log \frac{1}{\Pr\left[S_{2j-1} = \gamma_{2j-1}\right]} \\ &\quad = \ell - \sum_{j=1}^{r} \mathrm{H}(S_{2j}) + \sum_{j=1}^{r} \mathrm{H}(S_{2j-1}) \end{aligned}$$

The lemma follows.

Proof of Lemma 2.3: By Lemma 2.2,

$$\begin{aligned}\mathrm{KL}\left(S \,|\, \langle P_S, V\rangle\right) &= \ell + \sum_{i=1}^{2r}(-1)^{i+1}\cdot \mathrm{H}(S_i) \\ &\le \ell + \sum_{i=1}^{2r}(-1)^{i+1}\cdot \mathrm{H}(\langle P, V\rangle_i) \\ &\quad + \sum_{i=1}^{2r}\left|\mathrm{H}(S_i) - \mathrm{H}(\langle P, V\rangle_i)\right|\end{aligned}$$

Consider a perfect simulator (i.e., of zero deviation), denoted $\overline{S}$, for (P, V). Note that the simulator-based-prover with respect to $\overline{S}$ is P itself. Thus, by Lemma 2.2,

$$\begin{aligned}\ell + \sum_{i=1}^{2r}(-1)^{i+1}\cdot \mathrm{H}(\langle P, V\rangle_i) &= \ell + \sum_{i=1}^{2r}(-1)^{i+1}\cdot \mathrm{H}(\overline{S}_i) \\ &= \mathrm{KL}\left(\overline{S} \,|\, \langle P, V\rangle\right) = 0\end{aligned}$$

Finally, we use the fact (cf., Appendix B) that for any two random variables, X and Y, ranging over domain D it holds that

$$|\mathrm{H}(X) - \mathrm{H}(Y)| \le (\log|D|)\cdot\Delta(X\,,\,Y) + \mathrm{H}_2\left(\Delta(X\,,\,Y)\right)$$

Combining all the above, we get

$$\begin{aligned}&\mathrm{KL}\left(S \,|\, \langle P_S, V\rangle\right) \\ &\le \sum_{i=1}^{2r}\left|\mathrm{H}(S_i) - \mathrm{H}(\langle P, V\rangle_i)\right| \\ &\le \sum_{i=1}^{2r}\left[i\ell\cdot\Delta(S_i\,,\,\langle P, V\rangle_i) + \mathrm{H}_2\left(\Delta(S_i\,,\,\langle P, V\rangle_i)\right)\right] \\ &\le (2r^2 + r)\cdot\ell\cdot\Delta(S\,,\,\langle P, V\rangle) \\ &\quad +2r\cdot\mathrm{H}_2\left(\Delta(S\,,\,\langle P, V\rangle)\right)\end{aligned}$$

and the lemma follows.

Proof of Lemma 2.4: For any random variables X and Y and any function f it holds that $\mathrm{KL}\left(X \,|\, Y\right) \ge \mathrm{KL}\left(f(X) \,|\, f(Y)\right)$ (cf., Appendix B). Letting $f(\gamma) = 1$ if γ is accepting and $f(\gamma) = 0$ otherwise, we have

$$\mathrm{KL}\left(S(x) \,|\, \langle P_S, V\rangle(x)\right) \ge \mathrm{KL}_2(p, q')$$

where $q' \le q$ equals the probability that $\langle P_S, V\rangle(x)$ accepts. Using the fact that $\mathrm{KL}_2(p, q') \ge \mathrm{KL}_2(p, q)$, for any $q' \le q \le p$ (cf., Appendix B), we are done.

B Statistical Inequalities

Fact B.1 *For any two random variables, X and Y, ranging over a domain D it holds that*

$$|\mathrm{H}(X) - \mathrm{H}(Y)| \le \log(|D| - 1)\cdot\delta + \mathrm{H}_2(\delta)$$

where $\delta \stackrel{\text{def}}{=} \Delta(X\,,\,Y)$.

This fact can be inferred from Fano's Inequality (cf., [5, Thm. 2.11.1]). A more direct proof follows.

Proof: Assume $\delta > 0$ or else the claim is obvious. Let $p(x) \stackrel{\text{def}}{=} \Pr[X = x]$ and $q(x) \stackrel{\text{def}}{=} \Pr[X = x]$. Define $m(x) \stackrel{\text{def}}{=} \min\{p(x), q(x)\}$. Then $\sum_{x\in D} m(x) = 1 - \delta$. Define random variables Z', X' and Y' so that

$$\begin{aligned}\Pr[Z' = x] &= m'(x) \stackrel{\text{def}}{=} \frac{1}{1-\delta}\cdot m(x) \\ \Pr[X' = x] &= p'(x) \stackrel{\text{def}}{=} \frac{1}{\delta}\cdot(p(x) - m(x)) \\ \Pr[Y' = x] &= q'(x) \stackrel{\text{def}}{=} \frac{1}{\delta}\cdot(q(x) - m(x))\end{aligned}$$

Think of X (resp., Y) as being generated by picking Z' with probability $1 - \delta$ and X' (resp., Y') otherwise. Then,

$$\begin{aligned}\mathrm{H}(X) &\le (1-\delta)\cdot\mathrm{H}(Z') + \delta\cdot\mathrm{H}(X') + \mathrm{H}_2(\delta) \\ \mathrm{H}(Y) &\ge (1-\delta)\cdot\mathrm{H}(Z')\end{aligned}$$

Observing that $\Pr[X' = x] = 0$ on at least one $x \in D$, it follows that $\mathrm{H}(X') \le \log(|D| - 1)$, and the fact follows. ■

Comment: The above bound is tight. Let $e \in D$ and consider X which is identically e, and Y which with probability $1-\delta$ equals e and otherwise is uniform over $D \setminus \{e\}$. Clearly, $\Delta(X\,,\,Y) = \delta$ and $\mathrm{H}(Y) - \mathrm{H}(X) = \delta\log(|D| - 1) + \mathrm{H}_2(\delta) - 0$.

Fact B.2 *For any random variables X and Y and any function f it holds that* $\mathrm{KL}\left(X \,|\, Y\right) \ge \mathrm{KL}\left(f(X) \,|\, f(Y)\right)$.

This fact can be easily inferred from the Log Sum Inequality (cf., [5, Thm. 2.7.1]). A more direct proof follows.

Proof: Expanding the definition of $\mathrm{KL}\left(X \,|\, Y\right)$, we get $\mathrm{KL}\left(X \,|\, Y\right) = \sum_v \Pr[f(X) = v]\cdot A_v$, where

$$A_v = \sum_{x: f(x)=v}\Pr[X = x|f(X) = v]\cdot \log\frac{\Pr[f(X) = v]\cdot\Pr[X = x|f(X) = v]}{\Pr[f(Y) = v]\cdot\Pr[Y = x|f(Y) = v]}.$$

We can rewrite A_v as $B_v + C_v$, where

$$B_v = \sum_{x: f(x)=v}\Pr[X = x|f(X) = v]\cdot\log\frac{\Pr[f(X) = v]}{\Pr[f(Y) = v]},$$

and

$$C_v = \sum_{x: f(x)=v}\Pr[X = x|f(X) = v]\cdot \log\frac{\Pr[X = x|f(X) = v]}{\Pr[Y = x|f(Y) = v]}$$

Now, $\sum_v \Pr[f(X)=v] \cdot B_v$ equals $\mathrm{KL}(f(X)\,|\,f(Y))$, whereas the equals $\Pr[f(X)=v] \cdot \mathrm{KL}(X_v\,|\,Y_v) \geq 0$, where X_v (resp., Y_v) denotes the residual distribution of X conditioned on $f(X) = v$ (resp., Y conditioned on $f(Y) = v$). ■

Comment: The above bound is in fact equivalent to the Log Sum Inequality (i.e., $\sum_i a_i \log(a_i/b_i) \geq (\sum_i a_i)\log(\sum_i a_i / \sum_i b_i)$, for all non-negative a_i's and b_i's). To deduce to Log Sum Inequality from the above bound, one may first prove a special case in which $\sum_i a_i = \sum_i b_i = 1$ (by defining X and Y so that the a_i's and b_i's represent their probability mass, and let f be a constant function). The general case is derived by easy manipulation.

Fact B.3 *For any $0 \leq q' \leq q \leq p \leq 1$, it holds that $\mathrm{KL}_2(p,q') \geq \mathrm{KL}_2(p,q)$.*

Proof: We use the fact (cf., [5, Thm. 2.7.2]) that for every $0 \leq p, q_1, q_2 \leq 1$ and $0 \leq \lambda \leq 1$.

$$\mathrm{KL}_2(p, \lambda q_1 + (1-\lambda) q_2) \leq \lambda \cdot \mathrm{KL}_2(p, q_1) + (1-\lambda) \cdot \mathrm{KL}_2(p, q_2)$$

Picking $q_1 = q'$, $q_2 = p$ and λ such that $\lambda q_1 + (1-\lambda) q_2 = q$, we have $\mathrm{KL}_2(p,q) \leq \lambda \cdot \mathrm{KL}_2(p,q') + (1-\lambda) \cdot 0$, and the fact follows. ■

C Proof of the Flattening Lemma

For every x in the support of X, we let $w(x) = -\log \Pr[X = x]$. Then w maps the support of X, denoted D, to $[0, m]$. Let $X_1, \ldots, X_k$ be identical and independent copies of X. The lemma asserts that for every t,

$$\Pr\left[\left|\sum_{i=1}^{k} w(X_i) - k \cdot \mathrm{H}(X)\right| \geq t \cdot m\sqrt{k}\right] \;\leq\; 2^{-t^2+1}$$

Observe that $\mathrm{E}(w(X_i)) = \sum_x \Pr[X = x]\, w(x) = \mathrm{H}(X)$, for every i. Thus, the lemma follows by a straightforward application of Hoeffding Inequality: Specifically, define random variables $\xi_i = w(X_i)$, let $\mu = \mathrm{E}(\xi_i)$ and $\delta = tm/\sqrt{k}$, and use

$$\begin{aligned}
\Pr\left[\left|\frac{\sum_{i=1}^{k} \xi_i}{k} - \mu\right| \geq \delta\right] &\leq 2 \cdot \exp\left(-\frac{2\delta^2}{m^2} \cdot k\right) \\
&= 2 \cdot \exp\left(-2t^2\right)
\end{aligned}$$

The lemma follows. ■

D Proof of the Hashing Lemma

We denote the two distributions on pairs (h, z) in Lemma 3.12 by $A = (A_{\mathcal{H}}, A_Z)$ and $B = (B_{\mathcal{H}}, B_Z)$. By the definition of statistical difference, it suffices to show that for every set $S \subset \mathcal{H} \times D$, $\Pr[A \in S] - \Pr[B \in S] \leq 3(\delta + \varepsilon^{1/3})$. In order to do this, we first will argue that for "most" pairs (h, z), $\Pr[A = (h,z)]$ is not too much greater than $\Pr[B = (h,z)]$. Observe that both distributions A and B only output pairs (h, z) such that $h(z) = 0$. Now, for any $(h,z) \in \mathcal{H} \times D$ such that $h(z) = 0$, we have

$$\begin{aligned}
\Pr[A = (h,z)] &= \Pr[A_{\mathcal{H}} = h] \cdot \Pr[A_Z = z | A_{\mathcal{H}} = h] \\
&= \frac{1}{|\mathcal{H}|} \cdot \frac{\Pr[Z = z]}{\sum_{w \in h^{-1}(0)} \Pr[Z = w]},
\end{aligned}$$

and

$$\begin{aligned}
\Pr[B = (h,z)] &= \Pr[B_Z = z] \cdot \Pr[B_H = h | B_Z = z] \\
&= \Pr[Z = z] \cdot \frac{1}{|\{h' : h'(z) = 0\}|} \\
&= \Pr[Z = z] \cdot \frac{|R|}{|\mathcal{H}|},
\end{aligned}$$

where the last equality follows from 2-universality.

Thus, showing that $\Pr[A = (h,z)]$ is not too much greater than $\Pr[B = (h,z)]$ for most pairs (h,z) amounts to showing that for most h, $\sum_{w \in h^{-1}(0)} \Pr[Z = w]$ is not too much smaller than $1/|R|$. In order to prove a lower bound on this sum (for most h), we restrict the sum to a slightly smaller set of w's. Let $L = \{w \in D : \Pr[Z = w] \leq \varepsilon/|R|\}$, so by hypothesis, $\Pr[Z \in L] = 1 - \delta$. For $w \in D$ and $h \in \mathcal{H}$, define indicator functions

$$\chi_w(h) = \begin{cases} 1 & \text{if } h(w) = 0 \\ 0 & \text{otherwise} \end{cases}$$

Define $f(h) = \sum_{w \in L} \Pr[Z = w] \cdot \chi_w(h)$. Thus,

$$\sum_{w \in h^{-1}(0)} \Pr[Z = w] = \sum_{w \in D} \Pr[Z = w] \cdot \chi_w(h) \geq f(h)$$

By 2-universality, for h selected uniformly in $\mathcal{H}$, the random variables $\{\chi_w(h)\}_{w \in D}$ each have mean $1/|R|$ and are pairwise independent. Thus,

$$\mathrm{E}_h[f(h)] \;=\; \sum_{w \in L} \frac{\Pr[Z = w]}{|R|} = \frac{1-\delta}{|R|}$$

and

$$\begin{aligned}
\mathrm{Var}_h[f(h)] &\leq \sum_{w \in L} \frac{\Pr[Z = w]^2}{|R|} \\
&\leq \sum_{w \in L} \frac{\Pr[Z = w] \cdot \varepsilon}{|R|^2} \\
&\leq \frac{\varepsilon}{|R|^2}
\end{aligned}$$

By Chebyshev's inequality,

$$\Pr_h\left[f(h) - \frac{1-\delta}{|R|} < \frac{-\varepsilon^{1/3}}{|R|}\right] \le \frac{\mathrm{Var}_h(f(h))}{(\varepsilon^{1/3}/|R|)^2} \le \varepsilon^{1/3}.$$

Let $G = \{h \in \mathcal{H} : f(h) \ge (1-\delta-\varepsilon^{1/3})/|R|\}$ be the set "good" h's for which $f(h)$ is not too much smaller than $1/|R|$. Then for every $z \in D$ and $h \in G$,

$$\begin{aligned}\Pr[A = (h,z)] &\le \frac{\Pr[Z = z]}{|\mathcal{H}|} \cdot \frac{|R|}{1-\delta-\varepsilon^{1/3}} \\ &= \frac{\Pr[B = (h,z)]}{1-\delta-\varepsilon^{1/3}}.\end{aligned}$$

Thus, for any $S \subset \mathcal{H} \times D$,

$$\begin{aligned}\Pr[A \in S] &\le \Pr[A \in S \text{ and } A_{\mathcal{H}} \in G] + \Pr[A_{\mathcal{H}} \notin G] \\ &\le \frac{\Pr[B \in S \text{ and } B_{\mathcal{H}} \in G]}{1-\delta-\varepsilon^{1/3}} + \varepsilon^{1/3} \\ &\le \Pr[B \in S] \\ &\quad + \left(\frac{\delta+\varepsilon^{1/3}}{1-\delta-\varepsilon^{1/3}}\right) \cdot \Pr[B \in S] + \varepsilon^{1/3} \\ &\le \Pr[B \in S] + 3(\delta + \varepsilon^{1/3}),\end{aligned}$$

(as long as $\delta + \varepsilon^{1/3} \le 1/2$, which we may assume as otherwise the lemma is trivially satisfied). This completes the proof.