Unifying computational entropies via Kullback–Leibler divergence

Rohit Agrawal [*]
rohitagr@seas.harvard.edu

Yi-Hsiu Chen [†]
yhchen@seas.harvard.edu

Thibaut Horel [‡]
thorel@seas.harvard.edu

Salil Vadhan [§]
salil_vadhan@harvard.edu

August 19, 2019

Abstract

We introduce *hardness in relative entropy*, a new notion of hardness for search problems which on the one hand is satisfied by all one-way functions and on the other hand implies both *next-block pseudoentropy* and *inaccessible entropy*, two forms of computational entropy used in recent constructions of pseudorandom generators and statistically hiding commitment schemes, respectively. Thus, hardness in relative entropy unifies the latter two notions of computational entropy and sheds light on the apparent "duality" between them. Additionally, it yields a more modular and illuminating proof that one-way functions imply next-block inaccessible entropy, similar in structure to the proof that one-way functions imply next-block pseudoentropy (Vadhan and Zheng, STOC '12).

Keywords: one-way function, pseudorandom generator, pseudoentropy, computational entropy, inaccessible entropy, statistically hiding commitment, next-bit pseudoentropy.

[*]Harvard John A. Paulson School of Engineering and Applied Sciences. Supported by the Department of Defense (DoD) through the National Defense Science & Engineering Graduate Fellowship (NDSEG) Program.

[†]Harvard John A. Paulson School of Engineering and Applied Sciences. Supported by NSF grant CCF-1763299.

[‡]Harvard John A. Paulson School of Engineering and Applied Sciences. Supported in part by the National Science Foundation under grants CAREER IIS-1149662, CNS-1237235 and CCF-1763299, by the Office of Naval Research under grants YIP N00014-14-1-0485 and N00014-17-1-2131, and by a Google Research Award.

[§]Harvard John A. Paulson School of Engineering and Applied Sciences. Supported by NSF grant CCF-1763299.

1 Introduction

1.1 One-way functions and computational entropy

One-way functions [DH76] are on one hand the minimal assumption for complexity-based cryptography [IL89], but on the other hand can be used to construct a remarkable array of cryptographic primitives, including such powerful objects as CCA-secure symmetric encryption, zero-knowledge proofs and statistical zero-knowledge arguments for all of **NP**, and secure multiparty computation with an honest majority [GGM86, GMW91, GMW87, HILL99, Rom90, Nao91, HNO+09]. All of these constructions begin by converting the "raw hardness" of a one-way function (OWF) to one of the following more structured cryptographic primitives: a pseudorandom generator (PRG) [BM82, Yao82], a universal one-way hash function (UOWHF) [NY89], or a statistically hiding commitment scheme (SHC) [BCC88].

The original constructions of these three primitives from arbitrary one-way functions [HILL99, Rom90, HNO+09] were all very complicated and inefficient. Over the past decade, there has been a series of simplifications and efficiency improvements to these constructions [HRVW09, HRV13, HHR+10, VZ12], leading to a situation where the constructions of two of these primitives — PRGs and SHCs — share a very similar structure and seem "dual" to each other. Specifically, these constructions proceed as follows:

1. Show that every OWF $f : \{0,1\}^n \to \{0,1\}^n$ has a gap between its "real entropy" and an appropriate form of "computational entropy". Specifically, for constructing PRGs, it is shown that the function $\mathsf{G}(x) = (f(x), x_1, x_2, \ldots, x_n)$ has "next-block pseudoentropy" at least $n + \omega(\log n)$ while its real entropy is $\mathrm{H}(\mathsf{G}(U_n)) = n$ [VZ12] where $\mathrm{H}(\cdot)$ denotes Shannon entropy. For constructing SHCs, it is shown that the function $\mathsf{G}(x) = (f(x)_1, \ldots, f(x)_n, x)$ has "next-block accessible entropy" at most $n - \omega(\log n)$ while its real entropy is again $\mathrm{H}(\mathsf{G}(U_n)) = n$ [HRVW09]. Note that the differences between the two cases are whether we break x or $f(x)$ into individual bits (which matters because the "next-block" notions of computational entropy depend on the block structure) and whether the form of computational entropy is larger or smaller than the real entropy.

2. An "entropy equalization" step that converts G into a similar generator where the real entropy in each block conditioned on the prefix before it is known. This step is exactly the same in both constructions.

3. A "flattening" step that converts the (real and computational) Shannon entropy guarantees of the generator into ones on (smoothed) min-entropy and max-entropy. This step is again exactly the same in both constructions.

4. A "hashing" step where high (real or computational) min-entropy is converted to uniform (pseudo)randomness and low (real or computational) max-entropy is converted to a small-support or disjointness property. For PRGs, this step only requires randomness extractors [HILL99, NZ96], while for SHCs it requires (information-theoretic) interactive hashing [NOVY98, DHRS04]. (Constructing full-fledged SHCs in this step also utilizes UOWHFs, which can be constructed from one-way functions [Rom90]. Without UOWHFs, we obtain a weaker binding property, which nevertheless suffices for constructing statistical zero-knowledge arguments for all of **NP**.)

This common construction template came about through a back-and-forth exchange of ideas between the two lines of work. Indeed, the uses of computational entropy notions, flattening, and hashing originate with PRGs [HILL99], whereas the ideas of using next-block notions, obtaining them from breaking $(f(x), x)$ into short blocks, and entropy equalization originate with SHCs [HRVW09]. All this leads to a feeling that the two constructions, and their underlying computational entropy notions, are "dual" to each other and should be connected at a formal level.

In this paper, we make progress on this project of unifying the notions of computational entropy, by introducing a new computational entropy notion that yields both next-block pseudoentropy and next-block accessible entropy in a clean and modular fashion. It is inspired by the proof of [VZ12] that $(f(x), x_1, \ldots, x_n)$ has next-block pseudoentropy $n + \omega(\log n)$, which we will describe now.

1.2 Next-block pseudoentropy via relative pseudoentropy

We recall the definition of next-block pseudoentropy, and the result of [VZ12] relating it to one-wayness.

Definition 1.1 (next-block pseudoentropy, informal)**.** *Let n be a security parameter, and $X = (X_1, \ldots, X_m)$ be a random variable distributed on strings of length* $\mathrm{poly}(n)$*. We say that X has* next-block pseudoentropy *at least k if there is a random variable $Z = (Z_1, \ldots, Z_m)$, jointly distributed with X, such that:*

1. *For all $i = 1, \ldots, m$, $(X_1, \ldots, X_{i-1}, X_i)$ is computationally indistinguishable from $(X_1, \ldots, X_{i-1}, Z_i)$.*
2. *$\sum_{i=1}^{m} \mathrm{H}(Z_i | X_1, \ldots, X_{i-1}) \geq k$.*

Equivalently, for I uniformly distributed in $[m]$, X_I has conditional pseudoentropy *at least k/m given $(X_1, \ldots, X_{i-1})$.*

It was conjectured in [HRV10] that next-block pseudoentropy could be obtained from any OWF by breaking its input into bits, and this conjecture was proven in [VZ12]:

Theorem 1.2 ([VZ12], informal)**.** *Let $f : \{0,1\}^n \to \{0,1\}^n$ be a one-way function, let X be uniformly distributed in $\{0,1\}^n$, and let $X = (X_1, \ldots, X_m)$ be a partition of X into blocks of length $O(\log n)$. Then $(f(X), X_1, \ldots, X_m)$ has next-block pseudoentropy at least $n + \omega(\log n)$.*

The intuition behind Theorem 1.2 is that since X is hard to sample given $f(X)$, then it should have some extra computational entropy given $f(X)$. This intuition is formalized using the following notion of "relative pseudoentropy," which is a renaming of [VZ12]'s notion of "KL-hard for sampling," to better unify the terminology with the notions introduced in this work.

Definition 1.3 (relative pseudoentropy)**.** *Let n be a security parameter, and (X, Y) be a pair of random variables, jointly distributed over strings of length* $\mathrm{poly}(n)$*. We say that X has* relative pseudoentropy at least Δ *given Y if for all probabilistic polynomial-time S, we have*

$$\mathrm{KL}(X, Y \,\|\, \mathsf{S}(Y), Y) \geq \Delta,$$

where $\mathrm{KL}(\cdot \,\|\, \cdot)$ *denotes the relative entropy (a.k.a. Kullback–Leibler divergence).*[1]

That is, it is hard for any efficient adversary S to sample the conditional distribution of X given Y, even approximately.

The first step of the proof of Theorem 1.2 is to show that one-wayness implies relative pseudoentropy (which can be done with a one-line calculation):

Lemma 1.4. *Let $f : \{0,1\}^n \to \{0,1\}^n$ be a one-way function and let X be uniformly distributed in $\{0,1\}^n$. Then X has relative pseudoentropy at least $\omega(\log n)$ given $f(X)$.*

Next, we break X into short blocks, and show that the relative pseudoentropy is preserved:

Lemma 1.5. *Let n be a security parameter, let (X, Y) be random variables distributed on strings of length $\mathrm{poly}(n)$, let $X = (X_1, \ldots, X_m)$ be a partition of X into blocks, and let I be uniformly distributed in $[m]$. If X has relative pseudoentropy at least Δ given Y, then X_I has relative pseudoentropy at least Δ/m given $(Y, X_1, \ldots, X_{I-1})$.*

Finally, the main part of the proof is to show that, once we have short blocks, relative pseudoentropy is *equivalent* to a gap between conditional pseudoentropy and real conditional entropy.

Lemma 1.6. *Let n be a security parameter, Y be a random variable distributed on strings of length $\mathrm{poly}(n)$, and X a random variable distributed on strings of length $O(\log n)$. Then X has relative pseudoentropy at least Δ given Y iff X has conditional pseudoentropy at least $\mathrm{H}(X|Y) + \Delta$ given Y.*

Putting these three lemmas together, we see that when f is a one-way function, and we break X into blocks of length $O(\log n)$ to obtain $(f(X), X_1, \ldots, X_m)$, on average, the conditional pseudoentropy of X_I given $(f(X), X_1, \ldots, X_{I-1})$ is larger than its real conditional entropy by $\omega(\log n)/m$. This tells us that the next-block pseudoentropy of $(f(X), X_1, \ldots, X_m)$ is larger than its real entropy by $\omega(\log n)$, as claimed in Theorem 1.2.

We remark that Lemma 1.6 explains why we need to break the input of the one-way function into short blocks: it is false when X is long. Indeed, if f is a one-way function, then we have already seen that X has $\omega(\log n)$ relative pseudoentropy given $f(X)$ (Lemma 1.4), but it does not have conditional pseudoentropy noticeably larger than $\mathrm{H}(X|f(X))$ given $f(X)$ (as correct preimages can be efficiently distinguished from incorrect ones using f).

1.3 Inaccessible entropy

As mentioned above, for constructing SHCs from one-way functions, the notion of next-block pseudoentropy is replaced with next-block accessible entropy:

Definition 1.7 (next-block accessible entropy, informal)**.** *Let n be a security parameter, and $Y = (Y_1, \ldots, Y_m)$ be a random variable distributed on strings of length $\mathrm{poly}(n)$. We say that Y has* next-block accessible entropy *at most k if the following holds.*

Let $\widetilde{\mathsf{G}}$ be any probabilistic $\mathrm{poly}(n)$-time algorithm that takes a sequence of uniformly random strings $\widetilde{R} = (\widetilde{R}_1, \ldots, \widetilde{R}_m)$ and outputs a sequence $\widetilde{Y} = (\widetilde{Y}_1, \ldots, \widetilde{Y}_m)$ in an "online

[1] Recall that for random variables A and B with $\mathrm{Supp}(A) \subseteq \mathrm{Supp}(B)$, the relative entropy is defined by $\mathrm{KL}(A \,\|\, B) = \mathrm{E}_{a \leftarrow A}[\log(\Pr[A = a] / \Pr[B = a])]$.

fashion" by which we mean that $\widetilde{Y}_i = \widetilde{\mathsf{G}}(\widetilde{R}_1, \ldots, \widetilde{R}_i)$ *depends on only the first* i *random strings of* $\widetilde{\mathsf{G}}$ *for* $i = 1, \ldots, m$*. Suppose further that* $\mathrm{Supp}(\widetilde{Y}) \subseteq \mathrm{Supp}(Y)$*.*

Then we require:

$$\sum_{i=1}^{m} \mathrm{H}\left(\widetilde{Y}_i | \widetilde{R}_1, \ldots, \widetilde{R}_{i-1}\right) \leq k.$$

(Next-block) accessible entropy differs from (next-block) pseudoentropy in two ways:

1. Accessible entropy is useful as an *upper* bound on computational entropy, and is interesting when it is *smaller* than the real entropy $\mathrm{H}(Y)$. We refer to the gap $\mathrm{H}(Y) - k$ as the *next-block inaccessible entropy* of Y.

2. The accessible entropy adversary $\widetilde{\mathsf{G}}$ is trying to *generate* the random variables Y_i conditioned on the history rather than recognize them. Note that we take the "history" to not only be the previous blocks $(\widetilde{Y}_1, \ldots, \widetilde{Y}_{i-1})$, but the coin tosses $(\widetilde{R}_1, \ldots, \widetilde{R}_{i-1})$ used to generate those blocks.

Note that one unsatisfactory aspect of the definition is that when the random variable Y is not *flat* (i.e. uniform on its support), then there can be an adversary $\widetilde{\mathsf{G}}$ achieving accessible entropy even *larger* than $\mathrm{H}(Y)$, for example by making $\widetilde{Y}$ uniform on $\mathrm{Supp}(Y)$.

Similarly to (and predating) Theorem 1.2, it is known that one-wayness implies next-block inaccessible entropy.

Theorem 1.8 ([HRVW09])**.** *Let* $f : \{0,1\}^n \to \{0,1\}^n$ *be a one-way function, let* X *be uniformly distributed in* $\{0,1\}^n$*, and let* $(Y_1, \ldots, Y_m)$ *be a partition of* $Y = f(X)$ *into blocks of length* $O(\log n)$*. Then* $(Y_1, \ldots, Y_m, X)$ *has next-block accessible entropy at most* $n - \omega(\log n)$*.*

Unfortunately, however, the existing proof of Theorem 1.8 is not modular like that of Theorem 1.2. In particular, it does not isolate the step of relating one-wayness to entropy-theoretic measures (like Lemma 1.4 does) or the significance of having short blocks (like Lemma 1.6 does).

1.4 Our results

We remedy the above state of affairs by providing a new, more general notion of hardness in relative entropy that allows us to obtain next-block inaccessible entropy in a modular way while also encompassing what is needed for next-block pseudoentropy.

Like in relative pseudoentropy, we will consider a pair of jointly distributed random variables (Y, X). Following the spirit of accessible entropy, the adversary $\widetilde{\mathsf{G}}$ for our new notion will try to *generate* Y together with X, rather than taking Y as input. That is, $\widetilde{\mathsf{G}}$ will take randomness $\widetilde{R}$ and output a pair $(\widetilde{Y}, \widetilde{X}) = \widetilde{\mathsf{G}}(\widetilde{R}) = (\widetilde{\mathsf{G}}_1(\widetilde{R}), \widetilde{\mathsf{G}}_2(\widetilde{R}))$, which we require to be always within the support of (Y, X). Note that $\widetilde{\mathsf{G}}$ need not be an online generator; it can generate both $\widetilde{Y}$ and $\widetilde{X}$ using the same randomness $\widetilde{R}$. Of course, if (Y, X) is efficiently samplable (as it would be in most cryptographic applications), $\widetilde{\mathsf{G}}$ could generate $(\widetilde{Y}, \widetilde{X})$ identically distributed to (Y, X) by just using the "honest" sampler G for (Y, X). So, in addition, we require that the adversary $\widetilde{\mathsf{G}}$ also come with a *simulator* S, that

can simulate its coin tosses given only $\widetilde{Y}$. The goal of the adversary is to minimize the relative entropy

$$\mathrm{KL}\left(\widetilde{R}, \widetilde{Y} \,\middle\|\, \mathsf{S}(Y), Y\right)$$

for a uniformly random $\widetilde{R}$. This divergence measures both how well $\widetilde{\mathsf{G}}_1$ approximates the distribution of Y as well as how well S simulates the corresponding coin tosses of $\widetilde{\mathsf{G}}_1$. Note that when $\widetilde{\mathsf{G}}$ is the honest sampler G, the task of S is exactly to sample from the conditional distribution of $\widetilde{R}$ given $\mathsf{G}_1(\widetilde{R}) = Y$. However, the adversary may reduce the divergence by instead designing the sampler $\widetilde{\mathsf{G}}$ and simulator S to work in concert, potentially trading off how well $\widetilde{\mathsf{G}}(\widetilde{R})$ approximates Y in exchange for easier simulation by S. Explicitly, the definition is as follows.

Definition 1.9 (hardness in relative entropy, informal version of Definition 3.2)**.** *Let n be a security parameter, and (Y, X) be a pair of random variables jointly distributed over strings of length* $\mathrm{poly}(n)$*. We say that (Y, X) has* hardness at least Δ in relative entropy *if the following holds.*

Let $\widetilde{\mathsf{G}} = (\widetilde{\mathsf{G}}_1, \widetilde{\mathsf{G}}_2)$ and S be probabilistic $\mathrm{poly}(n)$*-time algorithms such that* $\mathrm{Supp}(\widetilde{\mathsf{G}}(\widetilde{R})) \subseteq \mathrm{Supp}((Y, X))$*, where $\widetilde{R}$ is uniformly distributed. Then writing $\widetilde{Y} = \widetilde{\mathsf{G}}_1(\widetilde{R})$, we require that*

$$\mathrm{KL}\left(\widetilde{R}, \widetilde{Y} \,\middle\|\, \mathsf{S}(Y), Y\right) \geq \Delta.$$

Similarly to Lemma 1.4, we can show that one-way functions achieve this notion of hardness in relative entropy.

Lemma 1.10. *Let $f : \{0,1\}^n \to \{0,1\}^n$ be a one-way function and let X be uniformly distributed in $\{0,1\}^n$. Then $(f(X), X)$ has hardness $\omega(\log n)$ in relative entropy.*

Note that this lemma implies Lemma 1.4. If we take $\widetilde{\mathsf{G}}$ to be the "honest" sampler $\widetilde{\mathsf{G}}(x) = (f(x), x)$, then we have:

$$\mathrm{KL}\left(X, f(X) \,\|\, \mathsf{S}(Y), Y\right) = \mathrm{KL}\left(\widetilde{R}, \widetilde{Y} \,\middle\|\, \mathsf{S}(Y), Y\right),$$

which is is $\omega(\log n)$ by Lemma 1.10. That is, relative pseudoentropy (as in Definition 1.3 and Lemma 1.4) is obtained by fixing $\widetilde{\mathsf{G}}$ and focusing on the hardness for the simulator S, i.e. the divergence $\mathrm{KL}\left(X, Y \,\|\, S(Y), Y\right)$. Furthermore, the step of breaking into short blocks (Lemma 1.5) is equivalent to requiring the simulator be *online* and showing that relative pseudoentropy implies the following notion of *next-block relative pseudoentropy*:

Definition 1.11 (next-block relative pseudoentropy, informal)**.** *Let n be a security parameter, (X, Y) be jointly distributed random variables over strings of length* $\mathrm{poly}(n)$*, and let $X = (X_1, \ldots, X_m)$ be a partition of X into blocks. We say that X has* next-block relative pseudoentropy at least Δ given Y *if for all probabilistic polynomial-time S, we have*

$$\sum_{i=1}^{m} \mathrm{KL}\left(X_i | X_{<i}, Y \,\|\, \mathsf{S}(X_{<i}, Y) | X_{<i}, Y\right) \geq \Delta,$$

where we use the notation $z_{<i} \stackrel{\text{def}}{=} (z_1, \ldots, z_{i-1})$.

Here, the simulator S is required to be "online" in the sense that it cannot simulate $(X_1, \ldots, X_m)$ at once, but must simulate X_i only as a function of $X_{<i}$ and Y.

In particular, Lemma 1.6 is thus equivalent to the statement that having next-block relative pseudoentropy at least Δ for blocks of length $O(\log n)$ is equivalent to having next-block pseudoentropy at least $\Delta + \sum_{i=1}^{m} \mathrm{H}\left(X_i | X_{<i}, Y\right)$ in the sense of Definition 1.1.

Conversely, we show that inaccessible entropy arises from hardness in relative entropy by first requiring the *generator* G to be online and breaking the relative entropy into blocks to obtain the following next-block hardness property.

Definition 1.12 (next-block hardness in relative entropy, informal)**.** *Let n be a security parameter, and $Y = (Y_1, \ldots, Y_m)$ be a random variable distributed on strings of length* $\mathrm{poly}(n)$*. We say that Y has* next-block hardness at least Δ in relative entropy *if the following holds.*

Let $\widetilde{\mathsf{G}}$ be any probabilistic $\mathrm{poly}(n)$*-time algorithm that takes a sequence of uniformly random strings $\widetilde{R} = (\widetilde{R}_1, \ldots, \widetilde{R}_m)$ and outputs a sequence $\widetilde{Y} = (\widetilde{Y}_1, \ldots, \widetilde{Y}_m)$ in an "online fashion" by which we mean that $\widetilde{Y}_i = \widetilde{\mathsf{G}}(\widetilde{R}_1, \ldots, \widetilde{R}_i)$ depends on only the first i random strings of $\widetilde{\mathsf{G}}$ for $i = 1, \ldots, m$. Suppose further that* $\mathrm{Supp}(\widetilde{Y}) \subseteq \mathrm{Supp}(Y)$*. Additionally, let S be a probabilistic* $\mathrm{poly}(n)$*-time algorithms such for all $i = 1, \ldots, m$, S takes as input $\widehat{R}_1, \ldots, \widehat{R}_{i-1}$ and Y_i and outputs $\widehat{R}_i$, where $\widehat{R}_j$ has the same length as $\widetilde{R}_j$. Then we require that for all such $(\widetilde{\mathsf{G}}, \mathsf{S})$, we have:*

$$\sum_{i=1}^{m} \mathrm{KL}\left(\widetilde{R}_i, \widetilde{Y}_i | \widetilde{R}_{<i}, \widetilde{Y}_{<i} \,\middle\|\, \widehat{R}_i, Y_i | \widehat{R}_{<i}, Y_{<i}\right) \geq \Delta.$$

Observe that hardness in relative entropy can be seen as the specific case of next-block hardness in relative entropy when there is only one block (*i.e.,* setting $m = 1$ in the previous definition).

Next, we fix the *simulator*, analogously to how relative pseudoentropy was obtained by fixing the generator, and obtain *next-block inaccessible relative entropy*:

Definition 1.13 (next-block inaccessible relative entropy, informal)**.** *Let n be a security parameter, and $Y = (Y_1, \ldots, Y_m)$ be a random variable distributed on strings of length* $\mathrm{poly}(n)$*. We say that Y has* next-block inaccessible relative entropy at least Δ *if the following holds.*

Let $\widetilde{\mathsf{G}}$ be any probabilistic $\mathrm{poly}(n)$*-time algorithm that takes a sequence of uniformly random strings $\widetilde{R} = (\widetilde{R}_1, \ldots, \widetilde{R}_m)$ and outputs a sequence $\widetilde{Y} = (\widetilde{Y}_1, \ldots, \widetilde{Y}_m)$ in an online fashion, and such that* $\mathrm{Supp}(\widetilde{Y}) \subseteq \mathrm{Supp}(Y)$*. Then we require that for all such $\widetilde{\mathsf{G}}$, we have:*

$$\sum_{i=1}^{m} \mathrm{KL}\left(\widetilde{Y}_i | \widetilde{R}_{<i}, \widetilde{Y}_{<i} \,\middle\|\, Y_i | R_{<i}, Y_{<i}\right) \geq \Delta,$$

where $R = (R_1, \ldots, R_m)$ is a dummy random variable independent of Y.

That is, the goal of the online generator $\widetilde{\mathsf{G}}$ is to generate $\widetilde{Y}_i$ given the history of coin tosses $\widetilde{R}_{<i}$ with the same conditional distribution as Y_i given $Y_{<i}$. As promised, there is no explicit simulator in the definition of next-block inaccessible relative entropy, as we essentially dropped all $\widehat{R}$ variables from the definition of next-block hardness in relative entropy. Nevertheless we can obtain it from hardness in relative entropy by using sufficiently short blocks:

Lemma 1.14. *Let n be a security parameter, let Y be a random variable distributed on strings of length* $\mathrm{poly}(n)$*, and let $Y = (Y_1, \ldots, Y_m)$ be a partition of Y into blocks of length* $O(\log n)$*.*

If $(Y_1, \ldots, Y_m)$ has next-block hardness at least Δ in relative entropy, then $(Y_1, \ldots, Y_m)$ has next-block inaccessible relative entropy at least $\Delta - \mathrm{negl}(n)$*.*

An intuition for the proof is that since the blocks are of logarithmic length, given Y_i we can simulate the corresponding coin tosses of $\widetilde{R}_i$ of $\widetilde{\mathsf{G}}$ by rejection sampling and succeed with high probability in $\mathrm{poly}(n)$ tries.

A nice feature of the definition of next-block inaccessible relative entropy compared to inaccessible entropy is that it is meaningful even for non-flat random variables, as the Kullback–Leibler divergence is always nonnegative. Moreover, for flat random variables, it equals the inaccessible entropy:

Lemma 1.15. *Suppose $Y = (Y_1, \ldots, Y_m)$ is a flat random variable. Then Y has next-block inaccessible relative entropy at least Δ if and only if Y has accessible entropy at most* $\mathrm{H}(Y) - \Delta$*.*

Intuitively, this lemma comes from the identity that if Y is a flat random variable and $\mathrm{Supp}(\widetilde{Y}) \subseteq \mathrm{Supp}(Y)$, then $\mathrm{H}\left(\widetilde{Y}\right) = \mathrm{H}(Y) - \mathrm{KL}\left(\widetilde{Y} \,\middle\|\, Y\right)$. We stress that we do not require the individual blocks Y_i have flat distributions, only that the random variable Y as a whole is flat. For example, if f is a function and X is uniform, then $(f(X), X)$ is flat even though $f(X)$ itself may be far from flat.

Putting together Lemmas 1.10, 1.14, and 1.15, we obtain a new, more modular (and slightly tighter) proof of Theorem 1.8. The reduction implicit in the combination of these lemmas is the same as the one in [HRVW09], but the analysis is different. (In particular, [HRVW09] makes no use of KL divergence.) Like the existing proof of Theorem 1.2, this proof separates the move from one-wayness to a form of hardness involving relative entropies, the role of short blocks, and the move from hardness in relative entropy to computational entropy, as summarized in Figure 1. Moreover, this further illumination of and toolkit for notions of computational entropy may open the door to other applications in cryptography.

We remark that another interesting direction for future work is to find a construction of universal one-way hash functions (UOWHFs) from one-way functions that follows a similar template to the above constructions of PRGs and SHCs. There is now a construction of UOWHFs based on a variant of inaccessible entropy [HHR+10], but it remains more complex and inefficient than those of PRGs and SHCs.

2 Preliminaries

Notations. For a tuple $x = (x_1, \ldots, x_n)$, we write $x_{\leq i}$ for $(x_1, \ldots, x_i)$, and $x_{<i}$ for $(x_1, \ldots, x_{i-1})$.

poly denotes the set of polynomial functions and negl the set of all negligible functions: $\varepsilon \in \mathrm{negl}$ if for all $p \in \mathrm{poly}$ and large enough $n \in \mathbb{N}$, $\varepsilon(n) \leq 1/p(n)$. We will sometimes abuse notations and write $\mathrm{poly}(n)$ to mean $p(n)$ for some $p \in \mathrm{poly}$ and similarly for $\mathrm{negl}(n)$.

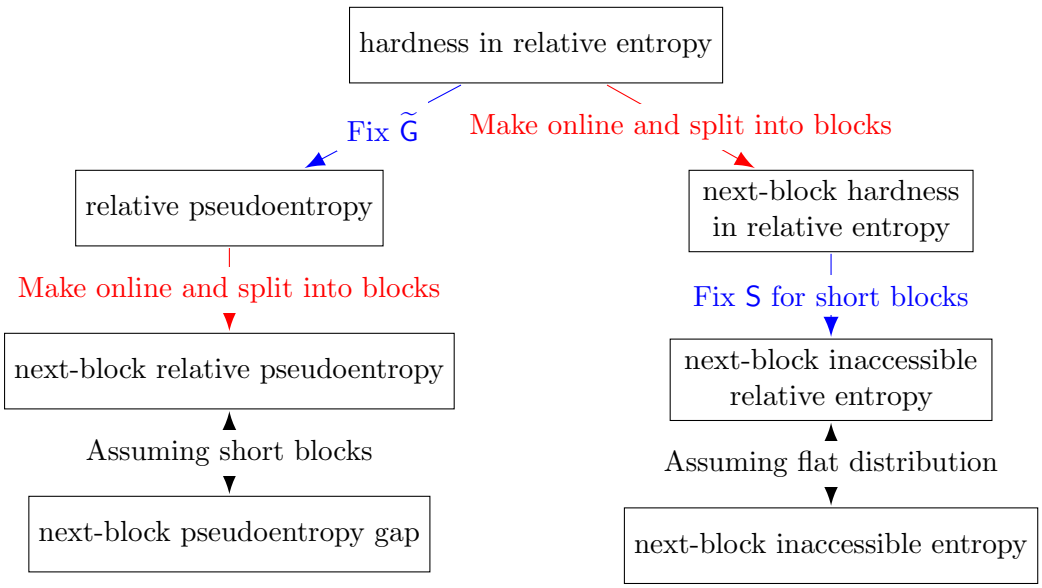

Figure 1: Relationships between hardness notions.

PPT stands for probabilistic polynomial time and can be either in the uniform or non-uniform model of computation. All our results are stated as uniform polynomial time oracle reductions and are thus meaningful in both models.

For a random variable X over $\mathcal{X}$, $\mathrm{Supp}(X) \stackrel{\text{def}}{=} \{x \in \mathcal{X} : \Pr[X = x] > 0\}$ denotes the support of X. A random variable is *flat* if it is uniform over its support. Random variables will be written with uppercase letters and the associated lowercase letter represents a generic element from its support.

Information theory.

Definition 2.1 (Entropy)**.** *For a random variable X and $x \in \mathrm{Supp}(X)$, the* sample entropy *(also called surprise) of x is $\mathrm{H}^*_x(X) \stackrel{\text{def}}{=} \log(1/\Pr[X = x])$. The* entropy $\mathrm{H}(X)$ *of X is the expected sample entropy: $\mathrm{H}(X) \stackrel{\text{def}}{=} \mathrm{E}_{x \leftarrow X}[\mathrm{H}^*_x(X)]$.*

Definition 2.2 (Conditional entropy)**.** *Let (A, X) be a pair of random variables and consider $(a, x) \in \mathrm{Supp}(A, X)$, the* conditional sample entropy *of (a, x) is $\mathrm{H}^*_{a,x}(A|X) \stackrel{\text{def}}{=} \log(1/\Pr[A = a \mid X = x])$ and the* conditional entropy *of A given X is the expected conditional sample entropy:*

$$\mathrm{H}(A|X) \stackrel{\text{def}}{=} \mathop{\mathrm{E}}_{(a,x) \leftarrow (A,X)} \left[\log \frac{1}{\Pr[A = a \mid X = x]} \right] .$$

Proposition 2.3 (Chain rule for entropy)**.** *Let (A, X) be a pair of random variables, then $\mathrm{H}(A, X) = \mathrm{H}(A|X) + \mathrm{H}(X)$ and for $(a, x) \in \mathrm{Supp}(A, X)$, $\mathrm{H}^*_{a,x}(A, X) = \mathrm{H}^*_{a,x}(A|X) + \mathrm{H}^*_x(X)$.*

Definition 2.4 (Relative entropy[2])**.** *For a pair* (A, B) *of random variables and* $(a, b) \in \mathrm{Supp}(A, B)$ *the* sample relative entropy *(*log*-probability ratio) is:*

$$\mathrm{KL}^*_a(A \,\|\, B) \stackrel{\mathrm{def}}{=} \log \frac{\Pr[A = a]}{\Pr[B = a]} ,$$

and the relative entropy *of* A *with respect to* B *is the expected sample relative entropy:*

$$\mathrm{KL}(A \,\|\, B) \stackrel{\mathrm{def}}{=} \mathop{\mathrm{E}}_{a \leftarrow A}\left[\log \frac{\Pr[A = a]}{\Pr[B = a]}\right] .$$

Definition 2.5 (Conditional relative entropy)**.** *For pairs of random variables* (A, X) *and* (B, Y)*, and* $(a, x) \in \mathrm{Supp}(A, X)$*, the* conditional sample relative entropy *is:*

$$\mathrm{KL}^*_{a,x}(A|X \,\|\, B|Y) \stackrel{\mathrm{def}}{=} \log \frac{\Pr[A = a|X = x]}{\Pr[B = a|Y = x]} ,$$

and the conditional relative entropy *is:*

$$\mathrm{KL}(A|X \,\|\, B|Y) \stackrel{\mathrm{def}}{=} \mathop{\mathrm{E}}_{(a,x) \leftarrow (A,X)}\left[\log \frac{\Pr[A = a|X = x]}{\Pr[B = a|Y = x]}\right] .$$

Proposition 2.6 (Chain rule for relative entropy)**.** *For pairs of random variables* (X, A) *and* (Y, B)*:*

$$\mathrm{KL}(A, X \,\|\, B, Y) = \mathrm{KL}(A|X \,\|\, B|Y) + \mathrm{KL}(X \,\|\, Y) ,$$

and for $(a, x) \in \mathrm{Supp}(A, X)$*:*

$$\mathrm{KL}^*_{a,x}(A, X \,\|\, B, Y) = \mathrm{KL}^*_{a,x}(A|X \,\|\, B|Y) + \mathrm{KL}^*_x(X \,\|\, Y) .$$

Proposition 2.7 (Data-processing inequality)**.** *Let* (X, Y) *be a pair of random variables and let* f *be a function defined on* $\mathrm{Supp}(Y)$*, then:*

$$\mathrm{KL}(X \,\|\, Y) \geq \mathrm{KL}(f(X) \,\|\, f(Y)) .$$

Definition 2.8 (min relative entropy)**.** *Let* (X, Y) *be a pair of random variables and* $\delta \in [0, 1]$*. We define* $\mathrm{KL}^\delta_{\min}(X \,\|\, Y)$ *to be the quantile of level* δ *of* $\mathrm{KL}^*_x(X \,\|\, Y)$*, equivalently it is the smallest* $\Delta \in \mathbb{R}$ *satisfying:*

$$\Pr_{x \leftarrow X}[\mathrm{KL}^*_x(X \,\|\, Y) \leq \Delta] \geq \delta ,$$

and it is characterized by the following equivalence:

$$\mathrm{KL}^\delta_{\min}(X \,\|\, Y) > \Delta \iff \Pr_{x \leftarrow X}[\mathrm{KL}^*_x(X \,\|\, Y) \leq \Delta] < \delta .$$

[2] *Relative entropy* is also commonly referred to as *Kullback–Liebler divergence*, which explains the standard KL notation. We prefer to use relative entropy to have more uniformity across the notions discussed in this work.

Block generators

Definition 2.9 (Block generator)**.** *An* m-block generator *is a function* $\mathsf{G} : \{0,1\}^s \to \prod_{i=1}^m \{0,1\}^{\ell_i}$. $\mathsf{G}_i(r)$ *denotes the* i*-th block of* G *on input* r *and* $|\mathsf{G}_i| = \ell_i$ *denotes the bit length of the* i*-th block.*

Definition 2.10 (Online generator)**.** *An* online m-block generator *is a function* $\widetilde{\mathsf{G}} : \prod_{i=1}^m \{0,1\}^{s_i} \to \prod_{i=1}^m \{0,1\}^{\ell_i}$ *such that for all* $i \in [m]$ *and* $r \in \prod_{i=1}^m \{0,1\}^{s_i}$, $\widetilde{\mathsf{G}}_i(r)$ *only depends on* $r_{\le i}$*. We sometimes write* $\widetilde{\mathsf{G}}_i(r_{\le i})$ *when the input blocks* $i+1, \ldots, m$ *are unspecified.*

Definition 2.11 (Support)**.** *The* support *of a generator* G *is the support of the random variable* $\mathrm{Supp}\big(\mathsf{G}(R)\big)$ *for uniform input* R*. If* G *is an* $(m+1)$*-block generator, and* Π *is a binary relation, we say that* G *is* supported on Π *if* $\mathrm{Supp}\big(\mathsf{G}_{\le m}(R), \mathsf{G}_{m+1}(R)\big) \subseteq \Pi$.

When G is an $(m+1)$-block generator supported on a binary relation Π, we will often use the notation $\mathsf{G}_{\mathsf{w}} \stackrel{\text{def}}{=} \mathsf{G}_{m+1}$ to emphasize that the last block corresponds to a witness for the first m blocks.

Cryptography.

Definition 2.12 (One-way Function)**.** *Let* n *be a security parameter,* $t = t(n)$ *and* $\varepsilon = \varepsilon(n)$*. A function* $f : \{0,1\}^n \to \{0,1\}^n$ *is a* (t, ε)-one-way function *if:*

1. *For all time* t *randomized algorithm* A*:* $\Pr_{x \leftarrow U_n}\big[\mathsf{A}\big(f(x)\big) \in f^{-1}\big(f(x)\big)\big] \le \varepsilon$*, where* U_n *is uniform over* $\{0,1\}^n$*.*
2. *There exists a polynomial time algorithm* B *such that* $\mathsf{B}(x) = f(x)$ *for all* $x \in \{0,1\}^n$*.*

If f *is* $(n^c, 1/n^c)$*-one-way for every* $c \in \mathbb{N}$*, we say that* f *is* (strongly) one-way*.*

3 Search Problems and Hardness in Relative Entropy

In this section, we first present the classical notion of hard-on-average search problems and introduce the new notion of hardness in relative entropy. We then relate the two notions by proving that average-case hardness implies hardness in relative entropy.

3.1 Search problems

For a binary relation $\Pi \subseteq \{0,1\}^* \times \{0,1\}^*$, we write $\Pi(y, w)$ for the predicate that is true iff $(y, w) \in \Pi$ and say that w is a *witness* for the *instance* y[3]. To each relation Π, we naturally associate (1) a *search problem*: given y, find w such that $\Pi(y, w)$ or state that no such w exist and (2) the *decision problem* defined by the language $L_\Pi \stackrel{\text{def}}{=} \{y \in \{0,1\}^* : \exists w \in \{0,1\}^*, \Pi(y, w)\}$. **FNP** denotes the set of all relations Π computable by a polynomial time algorithm and such that there exists a polynomial p such that $\Pi(y, w) \Rightarrow |w| \le p(|y|)$. Whenever $\Pi \in \mathbf{FNP}$, the associated decision problem L_Π is in **NP**. We now define average-case hardness.

[3]We used the unconventional notation y for the instance (instead of x) because our relations will often be of the form Π^f for some function f; in this case an instance is some y in the range of f and a witness for y is any preimage $x \in f^{-1}(y)$.

Definition 3.1 (distributional search problem)**.** *A* distributional search problem *is a pair* (Π, Y) *where* $\Pi \subseteq \{0,1\}^* \times \{0,1\}^*$ *is a binary relation and* Y *is a random variable supported on* L_Π*.*

The problem (Π, Y) *is* (t, ε)-hard *if* $\Pr\left[\Pi\big(Y, \mathsf{A}(Y)\big)\right] \leq \varepsilon$ *for all time* t *randomized algorithm* A*, where the probability is over the distribution of* Y *and the randomness of* A*.*

Example. For $f : \{0,1\}^n \to \{0,1\}^n$, the problem of inverting f is the search problem associated with the relation $\Pi^f \stackrel{\text{def}}{=} \{(f(x), x) : x \in \{0,1\}^n\}$. If f is a (t, ε)-one-way function, then the distributional search problem $\big(\Pi^f, f(X)\big)$ of inverting f on a uniform random input $X \in \{0,1\}^n$ is (t, ε)-hard.

Remark. Consider a distributional search problem (Π, Y). Without loss of generality, there exists a (possibly inefficient) two-block generator $\mathsf{G} = (\mathsf{G}_1, \mathsf{G}_\mathsf{w})$ supported on Π such that $\mathsf{G}_1(R) = Y$ for uniform input R. If G_w is polynomial-time computable, it is easy to see that the search problem $\big(\Pi^{\mathsf{G}_1}, \mathsf{G}_1(R)\big)$ is at least as hard as (Π, Y). The advantage of writing the problem in this "functional" form is that the distribution $(\mathsf{G}_1(R), R)$ over (instance, witness) pairs is flat, which is a necessary condition to relate hardness to inaccessible entropy (see Theorem 4.9).

Furthermore, if G_1 is also polynomial-time computable and (Π, Y) is $(\mathrm{poly}(n), \mathrm{negl}(n))$-hard, then $R \mapsto \mathsf{G}_1(R)$ is a one-way function. Combined with the previous example, we see that the existence of one-way functions is equivalent to the existence of $(\mathrm{poly}(n), \mathrm{negl}(n))$-hard search problems for which (instance, witness) pairs can be efficiently sampled.

3.2 Hardness in relative entropy

Instead of considering an adversary directly attempting to solve a search problem (Π, Y), the adversary in the definition of hardness in relative entropy comprises a pair of algorithm $(\widetilde{\mathsf{G}}, \mathsf{S})$ where $\widetilde{\mathsf{G}}$ is a two-block generator outputting valid (instance, witness) pairs for Π and S is a *simulator* for $\widetilde{\mathsf{G}}$: given an instance y, the goal of S is to output randomness r for $\widetilde{\mathsf{G}}$ such that $\widetilde{\mathsf{G}}_1(r) = y$. Formally, the definition is as follows.

Definition 3.2 (hardness in relative entropy)**.** *Let* (Π, Y) *be a distributional search problem. We say that* (Π, Y) *has* hardness (t, Δ) in relative entropy *if:*

$$\mathrm{KL}\left(\widetilde{R}, \widetilde{\mathsf{G}}_1(\widetilde{R}) \,\middle\|\, \mathsf{S}(Y), Y\right) > \Delta \,,$$

for all pairs $(\widetilde{\mathsf{G}}, \mathsf{S})$ *of time* t *algorithms where* $\widetilde{\mathsf{G}}$ *is a two-block generator supported on* Π *and* $\widetilde{R}$ *is uniform randomness for* $\widetilde{\mathsf{G}}_1$*. Similarly, for* $\delta \in [0,1]$*,* (Π, Y) *has* hardness (t, Δ) in δ-min relative entropy *if for all such pairs:*

$$\mathrm{KL}^{\delta}_{\min}\left(\widetilde{R}, \widetilde{\mathsf{G}}_1(\widetilde{R}) \,\middle\|\, \mathsf{S}(Y), Y\right) > \Delta \,.$$

Note that a pair $(\widetilde{\mathsf{G}}, \mathsf{S})$ achieves a relative entropy of zero in Definition 3.2 if $\widetilde{\mathsf{G}}_1(R)$ has the same distribution as Y and if $\widetilde{\mathsf{G}}_1\big(\mathsf{S}(y)\big) = y$ for all $y \in \mathrm{Supp}(Y)$. In this case, writing $\widetilde{\mathsf{G}}_\mathsf{w} \stackrel{\text{def}}{=} \widetilde{\mathsf{G}}_2$, we have that $\widetilde{\mathsf{G}}_\mathsf{w}\big(\mathsf{S}(Y)\big)$ is a valid witness for Y since $\widetilde{\mathsf{G}}$ is supported on Π.

More generally, the composition $\widetilde{\mathsf{G}}_\mathsf{w} \circ \mathsf{S}$ solves the search problem (Π, Y) whenever $\widetilde{\mathsf{G}}_1\big(\mathsf{S}(Y)\big) = Y$. When the relative entropies in Definition 3.2 are upper-bounded, we can lower bound the probability of the search problem being solved (Lemma 3.4). This immediately implies that hard search problems are also hard in relative entropy.

Theorem 3.3. *Let (Π, Y) be a distributional search problem. If (Π, Y) is (t, ε)-hard, then it has hardness (t', Δ') in relative entropy and (t', Δ'') in δ-min relative entropy for every $\delta \in [0,1]$ where $t' = \Omega(t)$,[4] $\Delta' = \log(1/\varepsilon)$ and $\Delta'' = \log(1/\varepsilon) - \log(1/\delta)$.*

Remark. As we see, a "good" simulator S for a generator $\widetilde{\mathsf{G}}$ is one for which $\widetilde{\mathsf{G}}_1\big(\mathsf{S}(Y)\big) = Y$ holds often. It will be useful in Section 4 to consider simulators S which are allowed to fail by outputting a failure string $r \notin \mathrm{Supp}(\widetilde{R})$, (e.g. $r = \perp$) and adopt the convention that $\widetilde{\mathsf{G}}_1(r) = \perp$ whenever $r \notin \mathrm{Supp}(\widetilde{R})$. With this convention, we can without loss of generality add the requirement that $\widetilde{\mathsf{G}}_1\big(\mathsf{S}(Y)\big) = Y$ whenever $\mathsf{S}(Y) \in \mathrm{Supp}(\widetilde{R})$: indeed, S can always check that it is the case and if not output a failure symbol. For such a simulator S, observe that for all $r \in \mathrm{Supp}(\widetilde{R})$, the second variable on both sides of the relative entropy in Definition 3.2 is obtained by applying $\widetilde{\mathsf{G}}_1$ on the first variable and can thus be dropped, leading to a simpler definition of hardness in relative entropy: $\mathrm{KL}\left(\widetilde{R} \middle\| \mathsf{S}(Y)\right) > \Delta$.

Theorem 3.3 is an immediate consequence of the following lemma.

Lemma 3.4. *Let (Π, Y) be a distributional search problem and $(\widetilde{\mathsf{G}}, \mathsf{S})$ be a pair of algorithms with $\widetilde{\mathsf{G}} = (\widetilde{\mathsf{G}}_1, \widetilde{\mathsf{G}}_{\mathsf{w}})$ a two-block generator supported on Π. Define the linear-time oracle algorithm $\mathsf{A}^{\widetilde{\mathsf{G}}_{\mathsf{w}},\mathsf{S}}(y) \stackrel{\text{def}}{=} \widetilde{\mathsf{G}}_{\mathsf{w}}(\mathsf{S}(y))$. For $\Delta \in \mathbb{R}^+$ and $\delta \in [0,1]$:*

1. *If $\mathrm{KL}\left(\widetilde{R}, \widetilde{\mathsf{G}}_1(\widetilde{R}) \middle\| \mathsf{S}(Y), Y\right) \le \Delta$ then $\Pr\left[\Pi(Y, \mathsf{A}^{\widetilde{\mathsf{G}}_{\mathsf{w}},\mathsf{S}}(Y))\right] \ge 1/2^{\Delta}$.*
2. *If $\mathrm{KL}^{\delta}_{\min}\left(\widetilde{R}, \widetilde{\mathsf{G}}_1(\widetilde{R}) \middle\| \mathsf{S}(Y), Y\right) \le \Delta$ then $\Pr\left[\Pi(Y, \mathsf{A}^{\widetilde{\mathsf{G}}_{\mathsf{w}},\mathsf{S}}(Y))\right] \ge \delta/2^{\Delta}$.*

Proof. We have:

$$\begin{aligned}
\Pr\left[\Pi\big(Y, \mathsf{A}^{\widetilde{\mathsf{G}}_{\mathsf{w}},\mathsf{S}}(Y)\big)\right] &= \Pr\left[\Pi(Y, \widetilde{\mathsf{G}}_{\mathsf{w}}(\mathsf{S}(Y)))\right] \\
&\ge \Pr\left[\widetilde{\mathsf{G}}_1(\mathsf{S}(Y)) = Y\right] && (\widetilde{\mathsf{G}} \text{ is supported on } \Pi) \\
&= \sum_{r \in \mathrm{Supp}(\widetilde{R})} \Pr\left[\mathsf{S}(Y) = r \wedge Y = \widetilde{\mathsf{G}}_1(r)\right] \\
&= \mathop{\mathrm{E}}_{r \leftarrow \widetilde{R}}\left[\frac{\Pr\left[\mathsf{S}(Y) = r \wedge Y = \widetilde{\mathsf{G}}_1(r)\right]}{\Pr\left[\widetilde{R} = r\right]}\right] \\
&= \mathop{\mathrm{E}}_{\substack{r \leftarrow \widetilde{R} \\ y \leftarrow \widetilde{\mathsf{G}}_1(r)}}\left[2^{-\mathrm{KL}^*_{r,y}\left(\widetilde{R}, \widetilde{\mathsf{G}}_1(\widetilde{R}) \,\middle\|\, \mathsf{S}(Y), Y\right)}\right].
\end{aligned}$$

Now, the first claim follows by Jensen's inequality (since $x \mapsto 2^{-x}$ is convex) and the second claim follows by Markov' inequality when considering the event that the sample relative entropy is smaller than Δ (which occurs with probability at least δ by assumption). □

[4] For the theorems in this paper that relate two notions of hardness, the notation $t' = \Omega(t)$ means that there exists a constant C depending *only* on the computational model such that $t' \ge C \cdot t$.

Relation to relative pseudoentropy. In [VZ12], the authors introduced the notion of relative pseudoentropy[5]: for jointly distributed variables (Y, W), W has relative pseudoentropy given Y if it is hard for a polynomial time adversary to approximate—measured in relative entropy—the conditional distribution W given Y. Formally:

Definition 3.5 (relative pseudoentropy, Def. 3.4 in [VZ12])**.** *Let (Y, W) be a pair of random variables, we say that W has* relative pseudoentropy (t, Δ) given Y *if for all time t randomized algorithm S, we have:*

$$\mathrm{KL}\left(Y, W \,\|\, Y, \mathsf{S}(Y)\right) > \Delta\,.$$

As discussed in Section 1.2, it was shown in [VZ12] that if $f : \{0,1\}^n \to \{0,1\}^n$ is a one-way function, then $\big(f(X), X_1, \ldots, X_n\big)$ has next-bit pseudoentropy for uniform $X \in \{0,1\}^n$ (see Theorem 1.2). The first step in proving this result was to prove that X has relative pseudoentropy given $f(X)$ (see Lemma 1.4).

We observe that when (Y, W) is of the form $(f(X), X)$ for some function $f : \{0,1\}^n \to \{0,1\}^n$ and variable X over $\{0,1\}^n$, then relative pseudoentropy is implied by hardness in relative entropy by simply fixing $\widetilde{\mathsf{G}}$ to be the "honest sampler" $\widetilde{\mathsf{G}}(X) = (f(X), X)$. Indeed, in this case we have:

$$\mathrm{KL}\left(X, \widetilde{\mathsf{G}}_1(X) \,\middle\|\, \mathsf{S}(Y), Y\right) = \mathrm{KL}\left(X, f(X) \,\|\, \mathsf{S}(Y), Y\right)\,.$$

We can thus recover Lemma 1.4 as a direct corollary of Theorem 3.3.

Corollary 3.6. *Consider a function $f : \{0,1\}^n \to \{0,1\}^n$ and define $\Pi^f \stackrel{\text{def}}{=} \{(f(x), x) : x \in \{0,1\}^n\}$ and $Y \stackrel{\text{def}}{=} f(X)$ for X uniform over $\{0,1\}^n$. If f is (t, ε)-one-way, then (Π^f, Y) has hardness $\big(t', \log(1/\varepsilon)\big)$ in relative entropy and X has relative pseudoentropy $\big(t', \log(1/\varepsilon)\big)$ given Y with $t' = \Omega(t)$.*

Witness hardness in relative entropy. We also introduce a relaxed notion of hardness in relative entropy called witness hardness in relative entropy. In this notion, we further require $(\widetilde{\mathsf{G}}, \mathsf{S})$ to approximate the joint distribution of (instance, witness) pairs rather than only instances. For example, the problem of inverting a function f over a random input X is naturally associated with the distribution $\big(f(X), X\big)$. The relaxation in this case is analogous to the notion of *distributional one-way function* for which the adversary is required to approximate the uniform distribution over preimages.

Definition 3.7 (witness hardness in relative entropy)**.** *Let Π be a binary relation and (Y, W) be a pair of random variables supported on Π. We say that (Π, Y, W) has* witness hardness (t, Δ) in relative entropy *if for all pairs of time t algorithms $(\widetilde{\mathsf{G}}, \mathsf{S})$ where $\widetilde{\mathsf{G}}$ is a two-block generator supported on Π, for uniform $\widetilde{R}$:*

$$\mathrm{KL}\left(\widetilde{R}, \widetilde{\mathsf{G}}_1(\widetilde{R}), \widetilde{\mathsf{G}}_{\mathsf{w}}(\widetilde{R}) \,\middle\|\, \mathsf{S}(Y), Y, W\right) > \Delta\,.$$

Similarly, for $\delta \in [0,1]$, (Π, Y, W) has witness hardness (t, Δ) in δ-min relative entropy*, if for all such pairs:*

$$\mathrm{KL}^{\delta}_{\min}\left(\widetilde{R}, \widetilde{\mathsf{G}}_1(\widetilde{R}), \widetilde{\mathsf{G}}_{\mathsf{w}}(\widetilde{R}) \,\middle\|\, \mathsf{S}(Y), Y, W\right) > \Delta\,.$$

[5] As already mentioned in the introduction, this notion was in fact called "KL-hardness for sampling" in [VZ12] but we rename it here to unify the terminology between the various notions discussed here.

We introduced hardness in relative entropy first, since it is the notion which is most directly obtained from the hardness of distribution search problems. Observe that by the data processing inequality for relative entropy (Proposition 2.7), dropping the third variable on both sides of the relative entropies in Definition 3.7 only decreases them. Hence, hardness in relative entropy implies witness hardness as stated in (Theorem 3.8). As we will see in Section 4 witness hardness in relative entropy is the "correct" notion to obtain inaccessible entropy from: it is in fact equal to inaccessible entropy up to $1/\operatorname{poly}$ losses.

Theorem 3.8. *Let Π be a binary relation and (Y, W) be a pair of random variables supported on Π. If (Π, Y) is (t, ε)-hard, then (Π, Y, W) has witness hardness (t', Δ') in relative entropy and (t', Δ'') in δ-min relative entropy for every $\delta \in [0, 1]$ where $t' = \Omega(t)$, $\Delta' = \log(1/\varepsilon)$ and $\Delta'' = \log(1/\varepsilon) - \log(1/\delta)$.*

Remark. The data processing inequality does not hold exactly for $\mathrm{KL}_{\min}$, hence the statement about δ-min relative entropy in Theorem 3.8 does not follow with the claimed parameters in a black-box manner from Theorem 3.3. However, an essentially identical proof given in Appendix A yields the result.

4 Inaccessible Entropy and Hardness in Relative Entropy

In this section, we relate our notion of witness hardness in relative entropy to the inaccessible entropy definition of [HRVW16]. Roughly speaking, we "split" the relative entropy into blocks and obtain the intermediate notion of next-block inaccessible relative entropy (Section 4.1) which we then relate to inaccessible entropy (Section 4.2). Together, these results show that if f is a one-way function, the generator $\mathsf{G}^f(X) = \big(f(X)_1, \ldots, f(X)_n, X\big)$ has superlogarithmic inaccessible entropy.

4.1 Next-block hardness and rejection sampling

For an online (adversarial) generator $\widetilde{\mathsf{G}}$, it is natural to consider simulators S that also operate in an online fashion. That is:

Definition 4.1 (online simulator)**.** *Let $\widetilde{\mathsf{G}} : \prod_{i=1}^m \{0,1\}^{s_i} \to \prod_{i=1}^m \{0,1\}^{\ell_i}$ be an online m-block generator. An* online simulator *for $\widetilde{\mathsf{G}}$ is a PPT algorithm S such that for all $y = (y_1, \ldots, y_m) \in \prod_{i=1}^m \{0,1\}^{\ell_i}$, defining inductively $\widehat{r}_i \stackrel{\text{def}}{=} \mathsf{S}(\widehat{r}_{<i}, y_i) \in \{0,1\}^{s_i}$, we have for all $i \in [m]$:*

$$\widetilde{\mathsf{G}}_i(\widehat{r}_{\le i}) = y_i \quad \text{or} \quad \widehat{r}_i = \perp .$$

The running time *of S is the total amount of time required to compute $\widehat{r}_1, \ldots, \widehat{r}_m$.*

The goal of such an online simulator S is to ensure that the distribution of $\widehat{R}_i = \mathsf{S}(\widehat{r}_{<i}, y_i)$ is close to that of $\widetilde{R}_i | \widetilde{R}_{<i} = \widehat{r}_{<i}, \widetilde{Y}_i = y_i$ where $(\widetilde{Y}_1, \ldots, \widetilde{Y}_m) \stackrel{\text{def}}{=} \widetilde{\mathsf{G}}(\widetilde{R}_{\le m})$ for uniformly random $(\widetilde{R}_1, \ldots, \widetilde{R}_m)$. Equivalently, $\widehat{R}_i$ should be close to uniform on $\{\widehat{r}_i : \widetilde{\mathsf{G}}_i(\widehat{r}_{\le i}) = y_i\}$. Measuring closeness with relative entropy, we have:

Definition 4.2 (next-block hardness in relative entropy)**.** *The joint distribution $Y = (Y_1, \ldots, Y_m)$ has* next-block hardness (t, Δ) in relative entropy *if the following holds for every time t online m-block generator $\widetilde{\mathsf{G}}$ and every time t online simulator S for $\widetilde{\mathsf{G}}$.*

Write $\widetilde{Y}_{\leq m} \stackrel{\text{def}}{=} \widetilde{\mathsf{G}}(\widetilde{R}_{\leq m})$ *for uniform* $\widetilde{R}_{\leq m}$*, and define inductively* $\widehat{R}_i \stackrel{\text{def}}{=} \mathsf{S}(\widehat{R}_{<i}, Y_i)$*. Then we require:*

$$\sum_{i=1}^{m} \mathrm{KL}\left(\widetilde{R}_i, \widetilde{Y}_i | \widetilde{R}_{<i}, \widetilde{Y}_{<i} \,\middle\|\, \widehat{R}_i, Y_i | \widehat{R}_{<i}, Y_{<i}\right) > \Delta\,.$$

Similarly, for $\delta \in [0,1]$*, we say that* $(Y_1, \ldots, Y_m)$ *has* next-block hardness (t, Δ) in δ-min relative entropy *if, with the same notations as above:*

$$\Pr_{\substack{r_{\leq m} \leftarrow \widetilde{R}_{\leq m} \\ y_{\leq m} \leftarrow \widetilde{\mathsf{G}}(r_{\leq m})}} \left[\sum_{i=1}^{m} \mathrm{KL}^{*}_{y_i, r_{<i}, y_{<i}}\left(\widetilde{R}_i, \widetilde{Y}_i | \widetilde{R}_{<i}, \widetilde{Y}_{<i} \,\middle\|\, \widehat{R}_i, Y_i | \widehat{R}_{<i}, Y_{<i}\right) \leq \Delta\right] < \delta\,.$$

Observe that using the chain rule for relative entropy, the sum of relative entropies appearing in Definition 4.2 is exactly equal to the relative entropies appearing in Definition 3.2. Since, furthermore considering an online generator $\widetilde{\mathsf{G}}$ and online simulator S is only less general than arbitrary pairs $(\widetilde{\mathsf{G}}, \mathsf{S})$, we immediately obtain the following theorem.

Theorem 4.3. *Let* (Π, Y) *be a distributional search problem. If* (Π, Y) *has hardness* (t, Δ) *in relative entropy then* $(Y_1, \ldots, Y_m)$ *has next-block hardness* (t, Δ) *in relative entropy.*

Similarly, for any $\delta \in [0,1]$*, if* (Π, Y) *has hardness* (t, Δ) *in* δ*-min relative entropy then* $(Y_1, \ldots, Y_m)$ *has next-block hardness* (t, Δ) *in* δ*-min relative entropy.*

Proof. Immediate using the chain rule for relative (sample) entropy. □

The next step is to obtain a notion of hardness that makes no reference to simulators by considering, for an online block generator $\widetilde{\mathsf{G}}$, a specific simulator $\mathsf{Sim}^{\widetilde{\mathsf{G}},T}$ which on input $(\widehat{r}_{<i}, y_i)$, generates $\widehat{R}_i$ using rejection sampling until $\widetilde{\mathsf{G}}_i(\widehat{r}_{<i}, \widehat{R}_i) = y_i$. The superscript T is the maximum number of attempts after which $\mathsf{Sim}^{\widetilde{\mathsf{G}},T}$ gives up and outputs $\perp$. The formal definition of $\mathsf{Sim}^{\widetilde{\mathsf{G}},T}$ is given in Algorithm 1.

Algorithm 1 Rejection sampling simulator $\mathsf{Sim}^{\widetilde{\mathsf{G}},T}$ for $1 \leq i \leq m$

Input: $y_i \in \{0,1\}^*$, $\widehat{r}_{<i} \in (\{0,1\}^v \cup \{\perp\})^{i-1}$
Output: $\widehat{r}_i \in \{0,1\}^v \cup \{\perp\}$
  if $\widehat{r}_{i-1} = \perp$ **then**
    $\widehat{r}_i \leftarrow \perp$; **return**
  end if
  repeat
    sample $\widehat{r}_i \leftarrow \{0,1\}^v$
  until $\widetilde{\mathsf{G}}_i(\widehat{r}_{\leq i}) = y_i$ or $\geq T$ attempts
  if $\widetilde{\mathsf{G}}_i(\widehat{r}_{\leq i}) \neq y_i$ **then**
    $\widehat{r}_i \leftarrow \perp$
  end if

For the rejection sampling simulator $\mathsf{Sim}^{\widetilde{\mathsf{G}},T}$, we will show in Lemma 4.5 that the next-block hardness in relative entropy in Definition 4.2 decomposes as the sum of two terms:

1. A term measuring how well $\widetilde{\mathsf{G}}_{\leq m}$ approximates the distribution Y in an online manner, without any reference to a simulator.

2. An error term measuring the failure probability of the rejection sampling procedure due to having a finite time bound T.

As we show in Lemma 4.6, the error term can be made arbitrarily small by setting the number of trials T in $\mathsf{Sim}^{\widetilde{\mathsf{G}},T}$ to be a large enough multiple of $m \cdot 2^{\ell}$ where ℓ is the length of the blocks of $\widetilde{\mathsf{G}}_{\leq m}$. This leads to a $\mathrm{poly}(m)$ time algorithm whenever ℓ is logarithmic in m. That is, given an online block generator $\widetilde{\mathsf{G}}$ for which $\widetilde{\mathsf{G}}_{\leq m}$ has short blocks, we obtain a corresponding simulator "for free". Thus, considering only the first term leads to the following clean definition of next-block inaccessible relative entropy that makes no reference to simulators.

Definition 4.4 (next-block inaccessible relative entropy)**.** *The joint distribution $(Y_1, \ldots, Y_m)$ has* next-block inaccessible relative entropy (t, Δ)*, if for every time t online m-block generator $\widetilde{\mathsf{G}}$ supported on $Y_{\leq m}$, writing $\widetilde{Y}_{\leq m} \stackrel{\text{def}}{=} \widetilde{\mathsf{G}}(\widetilde{R}_{\leq m})$ for uniform $\widetilde{R}_{\leq m}$, we have:*

$$\sum_{i=1}^{m} \mathrm{KL}\left(\widetilde{Y}_i | \widetilde{R}_{<i}, \widetilde{Y}_{<i} \,\middle\|\, Y_i | R_{<i}, Y_{<i}\right) > \Delta\,,$$

where R_i is a "dummy" random variable over the domain of $\widetilde{\mathsf{G}}_i$ and independent of $Y_{\leq m+1}$. Similarly, for $\delta \in [0,1]$, we say that $(Y_1, \ldots, Y_{m+1})$ has next-block inaccessible δ-min relative entropy (t, Δ) *if for every $\widetilde{\mathsf{G}}$ as above:*

$$\Pr_{\substack{r_{\leq m} \leftarrow \widetilde{R}_{\leq m} \\ y_{\leq m} \leftarrow \widetilde{\mathsf{G}}(r_{\leq m})}} \left[\sum_{i=1}^{m} \mathrm{KL}^{*}_{y_i, r_{<i}, y_{<i}}\left(\widetilde{Y}_i | \widetilde{R}_{<i}, \widetilde{Y}_{<i} \,\middle\|\, Y_i | R_{<i}, Y_{<i}\right) \leq \Delta\right] < \delta\,,$$

where $(\widetilde{Y}_{\leq m}, \widetilde{R}_{\leq m})$ are defined as above.

Remark. Since $\widetilde{Y}_{<i}$ is a function of $\widetilde{R}_{<i}$, the first conditional distribution in the KL is effectively $\widetilde{Y}_i | \widetilde{R}_{<i}$. Similarly the second distribution is effectively $Y_i | Y_{<i}$. The extra random variables are there for syntactic consistency.

With this definition in hand, we can make formal the claim that, even as sample notions, the next-block hardness in relative entropy decomposes as next-block inaccessible relative entropy plus an error term.

Lemma 4.5. *For a joint distribution $(Y_1, \ldots, Y_m)$, let $\widetilde{\mathsf{G}}$ be an online m-block generator supported on $Y_{\leq m}$. Define $(\widetilde{Y}_1, \ldots, \widetilde{Y}_m) \stackrel{\text{def}}{=} \widetilde{\mathsf{G}}(\widetilde{R})$ for uniform random variable $\widetilde{R} = (\widetilde{R}_1, \ldots, \widetilde{R}_m)$ and let R_i be a "dummy" random variable over the domain of $\widetilde{\mathsf{G}}_i$ and independent of $Y_{\leq m}$. We also define $\widehat{R}_i \stackrel{\text{def}}{=} \mathsf{Sim}^{\widetilde{\mathsf{G}},T}(\widehat{R}_{<i}, Y_i)$ and $\widehat{Y}_i = \widetilde{\mathsf{G}}(\widehat{R}_{\leq i})$. Then, for all $r \in \mathrm{Supp}(\widetilde{R})$ and $y \stackrel{\text{def}}{=} \widetilde{\mathsf{G}}(r)$:*

$$\begin{aligned}
&\sum_{i=1}^{m} \mathrm{KL}^{*}_{r,y}\left(\widetilde{R}_i, \widetilde{Y}_i | \widetilde{R}_{<i}, \widetilde{Y}_{<i} \,\middle\|\, \widehat{R}_i, Y_i | \widehat{R}_{<i}, Y_{<i}\right) \\
&\quad = \sum_{i=1}^{m} \mathrm{KL}^{*}_{r,y}\left(\widetilde{Y}_i | \widetilde{R}_{<i}, \widetilde{Y}_{<i} \,\middle\|\, Y_i | R_{<i}, Y_{<i}\right) + \sum_{i=1}^{m} \log\left(\frac{1}{\Pr\left[\widehat{Y}_i = y_i | Y_i = y_i, \widehat{R}_{<i} = r_{<i}\right]}\right).
\end{aligned}$$

Moreover, the running time of $\mathsf{Sim}^{\widetilde{\mathsf{G}},T}$ *on input* $\widehat{R}_{<i}, Y_i$ *is* $O(|r_i| \cdot T)$*, with at most* T *oracle calls to* $\widetilde{\mathsf{G}}$*.*

Proof. Consider $r \in \mathrm{Supp}(\widetilde{R})$ and $y \stackrel{\text{def}}{=} \widetilde{\mathsf{G}}(r)$. Then:

$$\begin{aligned}
&\sum_{i=1}^{m} \mathrm{KL}^*_{r,y}\left(\widetilde{R}_i, \widetilde{Y}_i | \widetilde{R}_{<i}, \widetilde{Y}_{<i} \,\middle\|\, \widehat{R}_i, Y_i | \widehat{R}_{<i}, Y_{<i}\right) \\
&\quad = \sum_{i=1}^{m} \mathrm{KL}^*_{r,y}\left(\widetilde{R}_i, \widetilde{Y}_i | \widetilde{R}_{<i}, \widetilde{Y}_{<i} \,\middle\|\, \widehat{R}_i, \widehat{Y}_i | \widehat{R}_{<i}, \widehat{Y}_{<i}\right) \\
&\quad = \sum_{i=1}^{m} \left(\mathrm{KL}^*_{r,y}\left(\widetilde{R}_i | \widetilde{R}_{<i}, \widetilde{Y}_{\le i} \,\middle\|\, \widehat{R}_i | \widehat{R}_{<i}, \widehat{Y}_{\le i}\right) + \mathrm{KL}^*_{r,y}\left(\widetilde{Y}_i | \widetilde{R}_{<i}, \widetilde{Y}_{<i} \,\middle\|\, \widehat{Y}_i | \widehat{R}_{<i}, \widehat{Y}_{<i}\right)\right) \\
&\quad = \sum_{i=1}^{m} \mathrm{KL}^*_{r,y}\left(\widetilde{Y}_i | \widetilde{R}_{<i}, \widetilde{Y}_{<i} \,\middle\|\, \widehat{Y}_i | \widehat{R}_{<i}, \widehat{Y}_{<i}\right) = \sum_{i=1}^{m} \mathrm{KL}^*_{r,y}\left(\widetilde{Y}_i | \widetilde{R}_{<i} \,\middle\|\, \widehat{Y}_i | \widehat{R}_{<i}\right) .
\end{aligned}$$

The first equality is because $Y_i = \widehat{Y}_i$ since we are only considering non-failure cases ($r_i \neq \perp$). The second equality is the chain rule. The penultimate equality is by definition of rejection sampling: $\widetilde{R}_i | \widetilde{R}_{<i}, \widetilde{Y}_{\le i}$ and $\widehat{R}_i | \widehat{R}_{<i}, \widehat{Y}_{\le i}$ are identical on $\mathrm{Supp}(\widetilde{R}_i)$ since conditioning on $\widehat{Y}_i = y$ implies that only non-failure cases ($r_i \neq \perp$) are considered. The last equality is because $\widetilde{Y}_{<i}$ (resp. $\widehat{Y}_{<i}$) is a deterministic function of $\widetilde{R}_{<i}$ (resp. $\widehat{R}_{<i}$).

We now relate $\widehat{Y}_i | \widehat{R}_{<i}$ to $Y_i | Y_{<i}$:

$$\begin{aligned}
\Pr\left[\widehat{Y}_i = y_i | \widehat{R}_{<i} = r_{<i}\right] &= \Pr\left[\widehat{Y}_i = y_i, Y_i = y_i | \widehat{R}_{<i} = r_{<i}\right] \quad (\widehat{Y}_i = y_i \Leftrightarrow \widehat{Y}_i = y_i \wedge Y_i = y_i) \\
&= \Pr\left[\widehat{Y}_i = y_i | Y_i = y_i, \widehat{R}_{<i} = r_{<i}\right] \cdot \Pr\left[Y_i = y_i | \widehat{R}_{<i} = r_{<i}\right] \qquad \text{(Bayes' Rule)} \\
&= \Pr\left[\widehat{Y}_i = y_i | Y_i = y_i, \widehat{R}_{<i} = r_{<i}\right] \cdot \Pr\left[Y_i = y_i | Y_{<i} = y_{<i}\right] ,
\end{aligned}$$

where the last equality is because when $r \in \mathrm{Supp}(\widetilde{R})$, $\widehat{R}_{<i} = r_{<i} \Rightarrow Y_{<i} = y_{<i}$ and because Y_i is independent of $\widehat{R}_{<i}$ given $Y_{<i}$ (as $\widehat{R}_{<i}$ is simply a randomized function of $Y_{<i}$). The conclusion of the lemma follows by combining the previous two derivations. □

Observe that taking expectations with respect to a uniform $\widetilde{R}$ on both sides in the conclusion of Lemma 4.5, we get that next-block hardness in relative entropy is equal to the sum of next-block inaccessible relative entropy and the expectation of the error term coming from the rejection sampling procedure. The following lemma upper bounds this expectation.

Lemma 4.6. *Let* $\widetilde{\mathsf{G}}$ *be an online* m*-block generator, and let* $L_i \stackrel{\text{def}}{=} 2^{|\widetilde{\mathsf{G}}_i|}$ *be the size of the codomain of* $\widetilde{\mathsf{G}}_i$*,* $i \in [m]$*. Then for all* $i \in [m]$*,* $r_{<i} \in \mathrm{Supp}(\widetilde{R}_{<i})$ *and uniform* $\widetilde{R}_i$*:*

$$\mathop{\mathrm{E}}_{y_i \leftarrow \widetilde{\mathsf{G}}_i(r_{<i}, \widetilde{R}_i)}\left[\log \frac{1}{\Pr\left[\widehat{Y}_i = y_i | Y_i = y_i, \widehat{R}_{<i} = r_{<i}\right]}\right] \le \log\left(1 + \frac{L_i - 1}{T}\right) .$$

Proof of Lemma 4.6. By definition of $\mathsf{Sim}^{\widetilde{\mathsf{G}},T}$, we have:

$$\Pr\left[\widehat{Y}_i = y_i | Y_i = y_i, \widehat{R}_{<i} = r_{<i}\right] = 1 - \left(1 - \Pr\left[\widetilde{\mathsf{G}}_i(r_{<i}, \widetilde{R}_i) = y_i\right]\right)^T .$$

Applying Jensen's inequality, we have:

$$\begin{aligned}
&\mathop{\mathrm{E}}_{y_i \leftarrow \widetilde{\mathsf{G}}_i(r_{<i}, \widetilde{R}_i)} \left[\log \left(\frac{1}{\Pr\left[\widehat{Y}_i = y_i | Y_i = y_i, \widehat{R}_{<i} = r_{<i} \right]} \right) \right] \\
&\qquad \leq \log \mathop{\mathrm{E}}_{y_i \leftarrow \widetilde{\mathsf{G}}_i(r_{<i}, \widetilde{R}_i)} \left[\frac{1}{\Pr\left[\widehat{Y}_i = y_i | Y_i = y_i, \widehat{R}_{<i} = r_{<i} \right]} \right] \\
&\qquad = \log \left(\sum_{y \in \mathrm{Im}(\widetilde{\mathsf{G}}_i(r_{<i}, \cdot))} \frac{p_y}{1 - (1 - p_y)^T} \right)
\end{aligned}$$

where $p_y = \Pr\left[\widetilde{\mathsf{G}}_i(r_{<i}, \widetilde{R}_i) = y \right]$. Since the function $x / \left(1 - (1-x)^T\right)$ is convex (see Lemma A.1 in the appendix), the maximum of the expression inside the logarithm over probability distributions $\{p_y\}$ is achieved at the extremal points of the standard probability simplex. Namely, when all but one $p_y \to 0$ and the other one is 1. Since $\lim_{x \to 0} x / \left(1 - (1-x)^T\right) = 1/T$:

$$\log \left(\sum_{y \in \mathrm{Im}(\widetilde{\mathsf{G}}_i)} \frac{p_y}{1 - (1 - p_y)^T} \right) \leq \log \left(1 + (L_i - 1) \cdot \frac{1}{T} \right). \qquad \square$$

By combining Lemmas 4.5 and 4.6, we are now ready to state the main result of this section, relating witness hardness in relative entropy to next-block inaccessible relative entropy.

Theorem 4.7. *Let Π be a binary relation and let (Y, W) be a pair of random variables supported on Π. Let $Y = (Y_1, \ldots, Y_m)$ be a partition of Y into blocks of at most ℓ bits. Then we have:*

1. *if (Π, Y, W) has witness hardness (t, Δ) in relative entropy, then for every $0 < \Delta' \leq \Delta$, $(Y_1, \ldots, Y_m, W)$ has next-block inaccessible relative entropy $(t', \Delta - \Delta')$ where $t' = \Omega(t\Delta'/(m^2 2^\ell))$.*

2. *if (Π, Y, W) has witness hardness (t, Δ) in δ-min relative entropy then for every $0 < \Delta' \leq \Delta$ and $0 \leq \delta' \leq 1 - \delta$, we have that $(Y_1, \ldots, Y_m, W)$ has next-block inaccessible $(\delta + \delta')$-min relative entropy $(t', \Delta - \Delta')$ where $t' = \Omega(t\delta'\Delta'/(m^2 2^\ell))$.*

Proof. We consider an online generator $\widetilde{\mathsf{G}}$ supported on $(Y_1, \ldots, Y_m, W)$ and the simulator $\mathsf{Sim}^{\widetilde{\mathsf{G}}, T}$. For convenience, we sometimes write Y_{m+1} for W. Define $\widetilde{R} \stackrel{\text{def}}{=} \widetilde{R}_{\leq m}$ where $\widetilde{R}_{\leq m}$ is a sequence of independent and uniformly random variables, $\widetilde{Y}_{\leq m+1} \stackrel{\text{def}}{=} \widetilde{\mathsf{G}}(\widetilde{R})$, $\widetilde{\mathsf{G}}_1(\widetilde{R}) \stackrel{\text{def}}{=} \widetilde{Y}_{\leq m}$ and $\widetilde{\mathsf{G}}_{\mathsf{w}}(\widetilde{R}) \stackrel{\text{def}}{=} \widetilde{Y}_{m+1}$. We also write for $1 \leq i \leq m$, $\widehat{R}_i \stackrel{\text{def}}{=} \mathsf{Sim}^{\widetilde{\mathsf{G}}, T}(\widehat{R}_{<i}, Y_i)$, $\widehat{Y}_i \stackrel{\text{def}}{=} \widetilde{\mathsf{G}}(\widehat{R}_{\leq i})_i$. Finally we define $\mathsf{S}^{\widetilde{\mathsf{G}}, T}(Y) \stackrel{\text{def}}{=} \widehat{R}_{\leq m}$.

Observe that $(\widetilde{\mathsf{G}}_1, \widetilde{\mathsf{G}}_{\mathsf{w}})$ is a two-block generator supported on Π, so the pair $(\widetilde{\mathsf{G}}, \mathsf{S}^{\widetilde{\mathsf{G}}, T})$ forms a pair a algorithms as in the definition of witness hardness in relative entropy

(Definition 3.7). We focus on sample notions first, and consider $r \in \mathrm{Supp}(\widetilde{R})$, $y \in \mathrm{Supp}(\widetilde{Y}_{\leq m})$ and $w \in \mathrm{Supp}(\widetilde{Y}_{m+1})$. First we use the chain rule to isolate the witness block:

$$\begin{aligned}
&\mathrm{KL}^*_{r,y,w}\left(\widetilde{R}, \widetilde{\mathsf{G}}_1(\widetilde{R}), \widetilde{\mathsf{G}}_{\mathsf{w}}(\widetilde{R}) \,\middle\|\, \mathsf{S}^{\widetilde{\mathsf{G}},T}(Y), Y, W\right) \\
&\quad = \mathrm{KL}^*_{r,y,w}\left(\widetilde{\mathsf{G}}_{\mathsf{w}}(\widetilde{R}) | \widetilde{R}, \widetilde{\mathsf{G}}_1(\widetilde{R}) \,\middle\|\, W | \mathsf{S}^{\widetilde{\mathsf{G}},T}(Y), Y\right) + \mathrm{KL}^*_{r,y,w}\left(\widetilde{R}, \widetilde{\mathsf{G}}_1(\widetilde{R}) \,\middle\|\, \mathsf{S}^{\widetilde{\mathsf{G}},T}(Y), Y\right) \\
&\quad = \mathrm{KL}^*_{r,y,w}\left(\widetilde{Y}_{m+1} | \widetilde{R}_{\leq m}, \widetilde{Y}_{\leq m} \,\middle\|\, Y_{m+1} | R_{\leq m}, Y_{\leq m}\right) + \mathrm{KL}^*_{r,y,w}\left(\widetilde{R}, \widetilde{\mathsf{G}}_1(\widetilde{R}) \,\middle\|\, \mathsf{S}^{\widetilde{\mathsf{G}},T}(Y), Y\right) .
\end{aligned}$$

Next, as in Theorem 4.3 we apply the chain rule to decompose the second term on the right-hand side and obtain next-block hardness in relative entropy:

$$\mathrm{KL}^*_{r,y,w}\left(\widetilde{R}, \widetilde{\mathsf{G}}_1(\widetilde{R}) \,\middle\|\, \mathsf{S}^{\widetilde{\mathsf{G}},T}(Y), Y\right) = \sum_{i=1}^{m} \mathrm{KL}^*_{r,y,w}\left(\widetilde{R}_i, \widetilde{Y}_i | \widetilde{R}_{<i}, \widetilde{Y}_{<i} \,\middle\|\, \widehat{R}_i, Y_i | \widehat{R}_{<i}, Y_{<i}\right) .$$

Finally, we use Lemma 4.5 to further decompose the right-hand side term into inaccessible relative entropy and the rejection sampling error:

$$\begin{aligned}
&\sum_{i=1}^{m} \mathrm{KL}^*_{r,y,w}\left(\widetilde{R}_i, \widetilde{Y}_i | \widetilde{R}_{<i}, \widetilde{Y}_{<i} \,\middle\|\, \widehat{R}_i, Y_i | \widehat{R}_{<i}, Y_{<i}\right) \\
&\quad = \sum_{i=1}^{m} \mathrm{KL}^*_{r,y}\left(\widetilde{Y}_i | \widetilde{R}_{<i}, \widetilde{Y}_{<i} \,\middle\|\, Y_i | R_{<i}, Y_{<i}\right) + \sum_{i=1}^{m} \log\left(\frac{1}{\Pr\left[\widehat{Y}_i = y_i | Y_i = y_i, \widehat{R}_{<i} = r_{<i}\right]}\right) .
\end{aligned}$$

Combining the previous derivations, we obtain:

$$\begin{aligned}
&\mathrm{KL}^*_{r,y,w}\left(\widetilde{R}, \widetilde{\mathsf{G}}_1(\widetilde{R}), \widetilde{\mathsf{G}}_{\mathsf{w}}(\widetilde{R}) \,\middle\|\, \mathsf{S}^{\widetilde{\mathsf{G}},T}(Y), Y, W\right) \\
&\quad = \sum_{i=1}^{m+1} \mathrm{KL}^*_{r,y}\left(\widetilde{Y}_i | \widetilde{R}_{<i}, \widetilde{Y}_{<i} \,\middle\|\, Y_i | R_{<i}, Y_{<i}\right) + \sum_{i=1}^{m} \log\left(\frac{1}{\Pr\left[\widehat{Y}_i = y_i | Y_i = y_i, \widehat{R}_{<i} = r_{<i}\right]}\right) .
\end{aligned}$$

Now, the first claim of the theorem follows by taking expectations on both sides and observing that when $T = m \cdot 2^\ell/(\Delta' \ln 2)$, Lemma 4.6 implies that the expected value of the rejection sampling error is smaller than Δ'.

For the second claim, we first establish using Lemma 4.6 and Markov's inequality that:

$$\Pr_{\substack{y_{\leq m+1} \leftarrow \widetilde{Y}_{\leq m+1} \\ r \leftarrow \widetilde{R}}}\left[\sum_{i=1}^{m} \log\left(\frac{1}{\Pr\left[\widehat{Y}_i = y_i | \widehat{R}_{<i} = r_{<i}, \widehat{Y}_{<i} = y_{<i}\right]}\right) \geq \frac{m \cdot 2^\ell}{T\delta' \ln 2}\right] \leq \delta'$$

and we reach a similar conclusion by setting $T = m \cdot 2^\ell/(\delta'\Delta' \ln 2)$. □

Remark. For fixed distribution and generators, in the limit where T grows to infinity, the error term caused by the failure of rejection sampling in time T vanishes. In this case, hardness in relative entropy implies next-block inaccessible relative entropy without any loss in the hardness parameters.

4.2 Next-block inaccessible relative entropy and inaccessible entropy

We first recall the definition from [HRVW16], slightly adapted to our notations.

Definition 4.8 (Inaccessible Entropy)**.** *Let $(Y_1, \ldots, Y_{m+1})$ be a joint distribution.*[6] *We say that $(Y_1, \ldots, Y_{m+1})$ has* inaccessible entropy (t, Δ) *if for all $(m+1)$-block online generators $\widetilde{\mathsf{G}}$ running in time t and consistent with $(Y_1, \ldots, Y_{m+1})$:*

$$\sum_{i=1}^{m+1} \left(\mathrm{H}(Y_i|Y_{<i}) - \mathrm{H}(\widetilde{Y}_i|\widetilde{R}_{<i}) \right) > \Delta \, .$$

where $(\widetilde{Y}_1, \ldots, \widetilde{Y}_{m+1}) = \widetilde{\mathsf{G}}(\widetilde{R}_1, \ldots, \widetilde{R}_{m+1})$ for a uniform $\widetilde{R}_{\leq m+1}$.

Similarly $(Y_1, \ldots, Y_{m+1})$ has inaccessible δ-max entropy (t, Δ) *if for all $(m+1)$-block online generators $\widetilde{\mathsf{G}}$ running in time t and consistent with $(Y_1, \ldots, Y_{m+1})$:*

$$\Pr_{\substack{r_{\leq m+1} \leftarrow \widetilde{R}_{\leq m+1} \\ y_{\leq m+1} \leftarrow \widetilde{\mathsf{G}}(r_{\leq m+1})}} \left[\sum_{i=1}^{m+1} \left(\mathrm{H}^*_{y_i, y_{<i}}(Y_i|Y_{<i}) - \mathrm{H}^*_{y_i, r_{<i}}\left(\widetilde{Y}_i|\widetilde{R}_{<i}\right) \right) \leq \Delta \right] < \delta \, .$$

Unfortunately, one unsatisfactory aspect of Definition 4.8 is that inaccessible entropy can be negative since the generator $\widetilde{\mathsf{G}}$ could have more entropy than $(Y_1, \ldots, Y_{m+1})$: if all the Y_i are independent biased random bits, then a generator $\widetilde{\mathsf{G}}$ outputting unbiased random bits will have negative inaccessible entropy. On the other hand, next-block inaccessible relative entropy (Definition 4.4) does not suffer from this drawback.

Moreover, in the specific case where $(Y_1, \ldots, Y_{m+1})$ is a flat distribution[7], then no distribution with the same support can have higher entropy and in this case Definitions 4.4 and 4.8 coincide as stated in the following theorem.

Theorem 4.9. *Let $(Y_1, , \ldots, Y_{m+1})$ be a flat distribution and $\widetilde{\mathsf{G}}$ be an $(m+1)$-block generator consistent with $Y_{\leq m+1}$. Then for $\widetilde{Y}_{\leq m+1} = \widetilde{\mathsf{G}}(\widetilde{R}_{\leq m+1})$ for uniform $\widetilde{R}_{\leq m+1}$:*

1. *For every $y_{\leq m+1}, r_{\leq m+1} \in \mathrm{Supp}(\widetilde{Y}_{\leq m+1}, \widetilde{R}_{\leq m+1})$, it holds that*

$$\sum_{i=1}^{m+1} \left(\mathrm{H}^*_{y_i, y_{<i}}(Y_i|Y_{<i}) - \mathrm{H}^*_{y_i, r_{<i}}\left(\widetilde{Y}_i|\widetilde{R}_{<i}\right) \right) = \sum_{i=1}^{m+1} \mathrm{KL}^*_{y_i, y_{<i}, r_{<i}}\left(\widetilde{Y}_i|\widetilde{R}_{<i}, \widetilde{Y}_{<i} \,\middle\|\, Y_i|R_{<i}, Y_{<i}\right)$$

   In particular, $(Y_1, \ldots, Y_{m+1})$ has next-block inaccessible δ-min relative entropy (t, Δ) if and only if it has inaccessible δ-max entropy (t, Δ).

[6] We write $m+1$ the total number of blocks, since in this section we will think of Y_{m+1} (also written as W) as the witness of a distributional search problem and $(Y_1, \ldots, Y_m)$ are the blocks of the instance as in the previous section.

[7] For example, the distribution $(Y_{\leq m}, Y_{m+1}) = (f(U), U)$ for a function f and uniform input U is always a flat distribution even if f itself is not regular.

2. *Furthermore,*

$$\sum_{i=1}^{m+1}\left(\mathrm{H}\left(Y_i|Y_{<i}\right)-\mathrm{H}\left(\widetilde{Y}_i|\widetilde{R}_{<i}\right)\right)=\sum_{i=1}^{m+1}\mathrm{KL}\left(\widetilde{Y}_i|\widetilde{R}_{<i},\widetilde{Y}_{<i}\,\middle\|\,Y_i|R_{<i},Y_{<i}\right),$$

so in particular, $(Y_1,\ldots,Y_{m+1})$ *has next-block inaccessible relative entropy* (t,Δ) *if and only if it has inaccessible entropy* (t,Δ)*.*

Proof. For the sample notions, the chain rule (Proposition 2.6) gives:

$$\sum_{i=1}^{m+1}\mathrm{H}^*_{y_i,y_{<i}}\left(Y_i|Y_{<i}\right)=\mathrm{H}^*_y\left(Y_{\le m+1}\right)=\log|\operatorname{Supp}(Y_{\le m+1})|$$

for all y since Y is flat. Hence:

$$\begin{aligned}\log|\operatorname{Supp}(Y_{\le m+1})|-\sum_{i=1}^{m+1}\mathrm{H}^*_{y_i,y_{<i}}\left(\widetilde{Y}_i|\widetilde{R}_{<i}\right)&=\sum_{i=1}^{m+1}\left(\mathrm{H}^*_{y_i,y_{<i}}\left(Y_i|Y_{<i}\right)-\mathrm{H}^*_{y_i,r_{<i}}\left(\widetilde{Y}_i|\widetilde{R}_{<i}\right)\right)\\&=\sum_{i=1}^{m+1}\mathrm{KL}^*_{y_i,y_{<i},r_{<i}}\left(\widetilde{Y}_i|\widetilde{R}_{<i},\widetilde{Y}_{<i}\,\middle\|\,Y_i|R_{<i},Y_{<i}\right),\end{aligned}$$

so the second claim follows by taking the expectation over $(\widetilde{Y}_{\le m+1},\widetilde{R}_{\le m+1})$ on both sides. □

By chaining the reductions between the different notions of hardness considered in this work (hardness in relative entropy, next-block inaccessible relative entropy and inaccessible entropy), we obtain a more modular proof of the theorem of Haitner *et al.* [HRVW16], obtaining inaccessible entropy from any one-way function.

Theorem 4.10. *Let* n *be a security parameter,* $f:\{0,1\}^n\to\{0,1\}^n$ *be a* (t,ε)*-one-way function, and* X *be uniform over* $\{0,1\}^n$*. For* $\ell\in\{1,\ldots,n\}$*, decompose* $f(X)\stackrel{\text{def}}{=}(Y_1,\ldots,Y_{n/\ell})$ *into blocks of length* ℓ*. Then:*

1. *For every* $0\le\Delta\le\log(1/\varepsilon)$*,* $(Y_1,\ldots,Y_{n/\ell},X)$ *has inaccessible entropy* $(t',\log(1/\varepsilon)-\Delta)$ *for* $t'=\Omega\left(t\cdot\Delta\cdot\ell^2/(n^2\cdot 2^\ell)\right)$*.*
2. *For every* $0<\delta\le 1$ *and* $0\le\Delta\le\log(1/\varepsilon)-\log(2/\delta)$*,* $(Y_1,\ldots,Y_{n/\ell},X)$ *has inaccessible* δ*-max entropy* $(t',\log(1/\varepsilon)-\log(2/\delta)-\Delta)$ *for* $t'=\Omega\left(t\cdot\delta\cdot\Delta\cdot\ell^2/(n^2\cdot 2^\ell)\right)$*.*

Proof. Since f is (t,ε)-one-way, the distributional search problem $\left(\Pi^f,f(X)\right)$ where $\Pi^f=\{(f(x),x):x\in\{0,1\}^n\}$ is (t,ε)-hard. Clearly, $(f(X),X)$ is supported on Π^f, so by applying Theorem 3.8, we have that $(\Pi^f,f(X),X)$ has witness hardness $(\Omega(t),\log(1/\varepsilon))$ in relative entropy and $(\Omega(t),\log(1/\varepsilon)-\log(2/\delta))$ in $\delta/2$-min relative entropy. Thus, by Theorem 4.7 we have that $(Y_1,\ldots,Y_{n/\ell},X)$ has next-block inaccessible relative entropy $\left(\Omega\left(t\cdot\Delta\cdot\ell^2/(n^2\cdot 2^\ell)\right),\log(1/\varepsilon)-\Delta\right)$ and next-block inaccessible δ-min relative entropy $\left(\Omega\left(t\cdot\delta\cdot\Delta\cdot\ell^2/(n^2\cdot 2^\ell)\right),\log(1/\varepsilon)-\log(2/\delta)-\Delta\right)$, and we conclude by Theorem 4.9. □

Remark. For comparison, the original proof of [HRVW16] shows that for every $0 < \delta \leq 1$, $(Y_1, \ldots, Y_{n/\ell}, X)$ has inaccessible δ-max entropy $(t', \log(1/\varepsilon) - 2\log(1/\delta) - O(1))$ for $t' = \tilde{\Omega}\left(t \cdot \delta \cdot \ell^2/(n^2 \cdot 2^\ell)\right)$, which in particular for fixed t' has quadratically worse dependence on δ in terms of the achieved inaccessible entropy: $\log(1/\varepsilon) - 2 \cdot \log(1/\delta) - O(1)$ rather than our $\log(1/\varepsilon) - 1 \cdot \log(1/\delta) - O(1)$.

Corollary 4.11 (Theorem 4.2 in [HRVW16])**.** *Let n be a security parameter, $f : \{0,1\}^n \to \{0,1\}^n$ be a strong one-way function, and X be uniform over $\{0,1\}^n$. Then for every $\ell = O(\log n)$, $(f(X)_{1\ldots\ell}, \ldots, f(X)_{n-\ell+1\ldots n}, X)$ has inaccessible entropy $\left(n^{\omega(1)}, \omega(\log n)\right)$ and inaccessible $1/n^{\omega(1)}$-max entropy $\left(n^{\omega(1)}, \omega(\log n)\right)$.*

Acknowledgements

We thank Muthuramakrishnan Venkitasubramaniam for an inspiring conversation which sparked this work.

References

[BCC88] Gilles Brassard, David Chaum, and Claude Crépeau. Minimum disclosure proofs of knowledge. *Journal of Computer and System Sciences*, 37(2):156–189, 1988.

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

[DH76] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.

[DHRS04] Yan Zong Ding, Danny Harnik, Alon Rosen, and Ronen Shaltiel. Constant-round oblivious transfer in the bounded storage model. In *Theory of Cryptography, First Theory of Cryptography Conference, TCC 2004*, pages 446–472. Springer, 2004.

[GGM86] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *Journal of the ACM*, 33(4):792–807, October 1986.

[GMW87] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game or a completeness theorem for protocols with honest majority. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computing (STOC)*, pages 218–229. ACM Press, 1987.

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

[HHR+10] Iftach Haitner, Thomas Holenstein, Omer Reingold, Salil P. Vadhan, and Hoeteck Wee. Universal one-way hash functions via inaccessible entropy. In

Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Monaco / French Riviera, May 30 - June 3, 2010. Proceedings, pages 616–637, 2010.

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999.

[HNO+09] Iftach Haitner, Minh Nguyen, Shien Jin Ong, Omer Reingold, and Salil Vadhan. Statistically hiding commitments and statistical zero-knowledge arguments from any one-way function. *SIAM Journal on Computing*, 39(3):1153–1218, 2009.

[HRV10] Iftach Haitner, Omer Reingold, and Salil Vadhan. Efficiency improvements in constructing pseudorandom generators from one-way functions. In *Proceedings of the 42nd Annual ACM Symposium on Theory of Computing (STOC)*, pages 437–446, 2010.

[HRV13] Iftach Haitner, Omer Reingold, and Salil P. Vadhan. Efficiency improvements in constructing pseudorandom generators from one-way functions. *SIAM J. Comput.*, 42(3):1405–1430, 2013.

[HRVW09] Iftach Haitner, Omer Reingold, Salil Vadhan, and Hoeteck Wee. Inaccessible entropy. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing (STOC '09)*, pages 611–620, 31 May–2 June 2009.

[HRVW16] Iftach Haitner, Omer Reingold, Salil P. Vadhan, and Hoeteck Wee. Inaccessible entropy I: Inaccessible entropy generators and statistically hiding commitments from one-way functions. To appear. Preliminary version, named Inaccessible Entropy, appeared in STOC 09, 2016.

[IL89] Russell Impagliazzo and Michael Luby. One-way functions are essential for complexity based cryptography. In *Proceedings of the 30th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 230–235, 1989.

[Nao91] Moni Naor. Bit commitment using pseudorandomness. *Journal of Cryptology*, 4(2):151–158, 1991.

[NOVY98] Moni Naor, Rafail Ostrovsky, Ramarathnam Venkatesan, and Moti Yung. Perfect zero-knowledge arguments for NP using any one-way permutation. *Journal of Cryptology*, 11(2):87–108, 1998. Preliminary version in *CRYPTO'92*.

[NY89] Moni Naor and Moti Yung. Universal one-way hash functions and their cryptographic applications. In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing (STOC)*, pages 33–43. ACM Press, 1989.

[NZ96] Noam Nisan and David Zuckerman. Randomness is linear in space. *Journal of Computer and System Sciences*, 52(1):43–52, February 1996.

[Rom90] John Rompel. One-way functions are necessary and sufficient for secure signatures. In *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing (STOC)*, pages 387–394, 1990.

[VZ12] Salil P. Vadhan and Colin Jia Zheng. Characterizing pseudoentropy and simplifying pseudorandom generator constructions. In *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012*, pages 817–836, 2012.

[Yao82] Andrew C. Yao. Theory and applications of trapdoor functions. In *Proceedings of the 23th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 80–91, 1982.

A Missing Proofs

Lemma A.1. *For all $t \geq 1$, $f : x \mapsto \frac{x}{1-(1-x)^t}$ is convex over $[0,1]$.*

Proof. We instead show convexity of $\tilde{f} : x \mapsto f(1-x)$. A straightforward computation gives:

$$\tilde{f}''(x) = \frac{x^{t-2}t\big(t(1-x)(x^t+1) - (1+x)(1-x^t)\big)}{(1-x^t)^3}$$

so that it suffices to show the non-negativity of $g(x) = t(1-x)(x^t+1) - (1+x)(1-x^t)$ over $[0,1]$. The function g has second derivative $t(1-x)(t^2-1)x^{t-2}$, which is non-negative when $x \in [0,1]$, and thus the first derivative g' is non-decreasing. Also, the first derivative at 1 is equal to zero, so that g' is non-positive over $[0,1]$ and hence g is non-increasing over this interval. Since $g(1) = 0$, this implies that g is non-negative over $[0,1]$ and f is convex as desired. □

Theorem A.2 (Theorem 3.8 restated)**.** *Let Π be a binary relation and let (Y, W) be pair of random variables supported on Π. If (Π, Y) is (t,ε)-hard, then (Π, Y, W) is (t', Δ') witness hard in relative entropy and (t', Δ'') witness hard in δ-min relative entropy for every $\delta \in [0,1]$ where $t' = \Omega(t)$, $\Delta' = \log(1/\varepsilon)$ and $\Delta'' = \log(\delta/\varepsilon)$.*

Proof. We proceed similarly to the proof of Theorem 3.3. Let $(\widetilde{\mathsf{G}}, \mathsf{S})$ be a pair of algorithms with $\widetilde{\mathsf{G}} = (\widetilde{\mathsf{G}}_1, \widetilde{\mathsf{G}}_{\mathsf{w}})$ a two-block generator supported on Π. Define the linear-time oracle

algorithm $\mathsf{A}^{\widetilde{\mathsf{G}}_{\mathsf{w}},\mathsf{S}}(y) \stackrel{\text{def}}{=} \widetilde{\mathsf{G}}_{\mathsf{w}}(\mathsf{S}(y))$. Then

$$\begin{aligned}
\Pr\left[\Pi\big(Y,\mathsf{A}^{\widetilde{\mathsf{G}}_{\mathsf{w}},\mathsf{S}}(Y)\big)\right] &= \Pr\left[\Pi(Y,\widetilde{\mathsf{G}}_{\mathsf{w}}(\mathsf{S}(Y)))\right] \\
&\geq \Pr\left[\widetilde{\mathsf{G}}_1(\mathsf{S}(Y)) = Y\right] && (\widetilde{\mathsf{G}} \text{ is supported on } \Pi) \\
&= \sum_{r\in\mathrm{Supp}(\widetilde{R})} \Pr\left[\mathsf{S}(Y) = r \wedge Y = \widetilde{\mathsf{G}}_1(r)\right] \\
&\geq \sum_{\substack{r\in\mathrm{Supp}(\widetilde{R})\\ w\in\mathrm{Supp}(\widetilde{\mathsf{G}}_2(\widetilde{R}))}} \Pr\left[\mathsf{S}(Y) = r \wedge Y = \widetilde{\mathsf{G}}_1(r) \wedge W = w\right] \\
&= \mathop{\mathrm{E}}_{\substack{r\leftarrow\widetilde{R}\\ w\leftarrow\widetilde{\mathsf{G}}_2(r)}} \left[\frac{\Pr\left[\mathsf{S}(Y) = r \wedge Y = \widetilde{\mathsf{G}}_1(r) \wedge W = w\right]}{\Pr\left[\widetilde{R} = r \wedge \widetilde{\mathsf{G}}_2(r) = w\right]}\right] \\
&= \mathop{\mathrm{E}}_{\substack{r\leftarrow\widetilde{R}\\ y\leftarrow\widetilde{\mathsf{G}}_1(r)\\ w\leftarrow\widetilde{\mathsf{G}}_2(r)}} \left[2^{-\mathrm{KL}^*_{r,y,w}\left(\widetilde{R},\widetilde{\mathsf{G}}_1(\widetilde{R}),\widetilde{\mathsf{G}}_2(\widetilde{R}) \,\|\, \mathsf{S}(Y),Y,W\right)}\right],
\end{aligned}$$

The witness hardness in relative entropy then follows by applying Jensen's inequality (since 2^{-x} is convex) and the witness hardness in δ-min relative entropy follows by Markov's inequality by considering the event that the sample relative entropy is smaller than Δ (this event has density at least δ). □