Characterizing Pseudoentropy and Simplifying Pseudorandom Generator Constructions

Citation	Vadhan, Salil, and Colin Jia Zheng. 2012. Characterizing Pseudoentropy and Simplifying Pseudorandom Generator Constructions. In Proceedings of the 44th Symposium on Theory of Computing (STOC '12, New York, NY, USA, May 19 - 22, 2012), ed. ACM Special Interest Group on Automata and Computability Theory, 817-836. New York: ACM Press.
Published Version	doi:10.1145/2213977.2214051
Accessed	February 16, 2015 5:37:31 PM EST
Citable Link	http://nrs.harvard.edu/urn-3:HUL.InstRepos:12724017
Terms of Use	This article was downloaded from Harvard University's DASH repository, and is made available under the terms and conditions applicable to Other Posted Material, as set forth at http://nrs.harvard.edu/urn-3:HUL.InstRepos:dash.current.terms-of-use#LAA

Characterizing Pseudoentropy and Simplifying Pseudorandom Generator Constructions [*]

Salil Vadhan[†] Colin Jia Zheng[‡]
School of Engineering and Applied Sciences
Harvard University
Cambridge, Massachusetts
{salil,colinz}@seas.harvard.edu

ABSTRACT

We provide a characterization of pseudoentropy in terms of hardness of sampling: Let (X, B) be jointly distributed random variables such that B takes values in a polynomial-sized set. We show that B is computationally indistinguishable from a random variable of higher Shannon entropy given X if and only if there is no probabilistic polynomial-time S such that $(X, S(X))$ has small KL divergence from (X, B). This can be viewed as an analogue of the Impagliazzo Hardcore Theorem (FOCS '95) for Shannon entropy (rather than min-entropy).

Using this characterization, we show that if f is a one-way function, then $(f(U_n), U_n)$ has "next-bit pseudoentropy" at least $n+\log n$, establishing a conjecture of Haitner, Reingold, and Vadhan (STOC '10). Plugging this into the construction of Haitner et al., this yields a simpler construction of pseudorandom generators from one-way functions. In particular, the construction only performs hashing once, and only needs the hash functions that are randomness extractors (e.g. universal hash functions) rather than needing them to support "local list-decoding" (as in the Goldreich–Levin hardcore predicate, STOC '89).

With an additional idea, we also show how to improve the seed length of the pseudorandom generator to $\tilde{O}(n^3)$, compared to $\tilde{O}(n^4)$ in the construction of Haitner et al.

Categories and Subject Descriptors

F.0 [**Theory of Computation**]: [General]

General Terms

Theory

Keywords

cryptography, computational complexity, pseudorandomness, entropy, KL divergence, min-max theorem, hardcore lemma

1. INTRODUCTION

Computational analogues of information-theoretic notions have given rise to some of the most interesting phenomena in complexity and cryptography. For example, *computational indistinguishability* [GM2], which is the computational analogue of statistical distance, enabled bypassing Shannon's impossibility results on perfectly secure encryption [Sha], and provided the basis for the computational theory of pseudorandomness [BM, Yao1].

Computational analogues of *entropy* were introduced by Yao [Yao1] and Håstad, Impagliazzo, Levin, and Luby [HILL]. The Håstad et al. notion, known as *pseudoentropy*, was key to their fundamental result establishing the equivalence of pseudorandom generators and one-way functions, and has also now become a basic concept in complexity theory and cryptography.

A more relaxed notion, called *next-bit pseudoentropy*, was recently introduced by Haitner, Reingold, and Vadhan [HRV], who used it to give a simpler and more efficient construction of pseudorandom generators from one-way functions. From a one-way function on n-bit strings, they construct a pseudorandom generator with seed length $\tilde{O}(n^4)$, improving the bound of $\tilde{O}(n^8)$ from [HILL, Hol2].

In this work, we provide new characterizations of pseudoentropy and next-bit pseudoentropy, and use these to further simplify the construction of pseudorandom generators from one-way functions. In addition, we show how to save another factor of n in the seed length, yielding a pseudorandom generator with seed length $\tilde{O}(n^3)$ from a one-way function on n bits.

1.1 Characterizing Pseudoentropy

The Håstad et al. notion of pseudoentropy is the following:

Definition 1.1 (pseudoentropy [HILL], informal)**.** A random variable X has *pseudoentropy at least k* if there exists a random variable Y such that:

1. X is computationally indistinguishable from Y.

[*] Supported by NSF grant CCF-1116616 and US-Israel BSF grant 2010196. A full version of this paper appears on ECCC [VZ2].

[†] http://seas.harvard.edu/~salil. Currently on leave as a Visiting Researcher at Microsoft Research SVC and a Visiting Scholar at Stanford University.

[‡] http://seas.harvard.edu/~colinz. Currently an Exchange Scholar at Stanford University.

STOC12, May 19–22, 2012, New York, New York, USA.
Copyright 2012 ACM 978-1-4503-1245-5/12/05 ...$10.00.

2. $\mathrm{H}(Y) \geq k$, where $\mathrm{H}(\cdot)$ denotes Shannon entropy.[1]

Pseudoentropy is interesting because a random variable can have much higher pseudoentropy than its Shannon entropy. Indeed, if $G : \{0,1\}^n \to \{0,1\}^m$ is a pseudorandom generator, then $G(U_n)$ has Shannon entropy at most n, but is indistinguishable from U_m (by definition) and hence has pseudoentropy $m > n$. (Here and throughout, U_n denotes a random variable uniformly distributed over $\{0,1\}^n$.)

A useful generalization is the notion of *conditional pseudoentropy*, analogous to the notion of conditional pseudo-min-entropy studied by Hsiao, Lu, and Reyzin [HLR]:

Definition 1.2 (conditional pseudoentropy, informal)**.** Let (X, B) be jointly distributed random variables. We say that B has *(conditional) pseudoentropy at least k given X* if there exists a random variable C, jointly distributed with X such that

1. (X, B) is computationally indistinguishable from (X, C).

2. $\mathrm{H}(C|X) \geq k$, where $\mathrm{H}(\cdot|\cdot)$ denotes conditional Shannon entropy.[2]

Note that if B has pseudoentropy at least k given X, then (X, B) has pseudoentropy at least $\mathrm{H}(X)+k$, but the converse is false (consider X that has pseudoentropy $\mathrm{H}(X)+k$ on its own, with a B that has no pseudoentropy).

Intuitively, a random variable B should have high pseudoentropy given X iff B is hard to predict from X, and indeed there are results of this type known in special cases involving pseudo-min-entropy (to be discussed later). Our main result is such a characterization for pseudoentropy (i.e. pseudo-Shannon-entropy).

Before getting to the formal statement, note that both pseudoentropy and unpredictability may occur for information-theoretic reasons, as $\mathrm{H}(B|X)$ may be larger than 0. For example, suppose that B is a uniform random bit, independent of X. Then B has 1 bit of pseudoentropy given X and cannot be predicted better than random guessing from X, but these are not for computational reasons (i.e. they also hold for computationally unbounded algorithms). We would like to focus on the computational randomness in B. For pseudoentropy we can do this by simply subtracting $\mathrm{H}(B|X)$. For unpredictability, we do this by considering the feasibility of *sampling* the distribution $B|_{X=x}$ given a sample $x \sim X$. Thus, in the example that B is a random bit independent of X, this sampling is easy to do (in contrast to the task of predicting B from X).

With these choices, we can indeed prove that pseudoentropy and hardness of sampling are equivalent:

Theorem 1.3 (characterizing pseudoentropy, informal)**.** *Let (X, B) be jointly distributed random variables where B takes values in a polynomial-sized set. Then B has pseudoentropy at least $\mathrm{H}(B|X) + \delta$ given X if and only if there is no probabilistic polynomial-time algorithm S such that the KL divergence from (X, B) to $(X, S(X))$ is at most δ.*[3]

KL divergence is a common information-theoretic measure of "distance" between random variables (though it is not a metric).

The constraint that B takes values in a polynomial-sized set is essential for this theorem: If f is a one-way permutation and X is a uniformly random output, then it is very hard to sample $f^{-1}(X)$ given X, but the pseudoentropy of $f^{-1}(X)$ given X is negligible (since we can efficiently recognize $f^{-1}(X)$ given X). However, we do have an alternative version of our result that holds for B taking values in an exponentially large range (when considering nonuniform complexity). In that version, we replace the task of sampling a distribution $S(X)$ from X with that of computing a "measure" that, when normalized to be a distribution, has small KL divergence from (X, B). In particular, this alternative formulation is interesting even when X is empty and gives a characterization of pseudoentropy of an arbitrary random variable B (with respect to nonuniform complexity).

To provide some more intuition for our theorem and the proof techniques, we compare it to previous results relating forms of pseudoentropy and unpredictability.

1. Yao [Yao2] showed that if B is a single bit, then (X, B) is indistinguishable from (X, U_1) (i.e. B has pseudoentropy at least 1 given X) iff B cannot be predicted from X with probability noticeably more than 1/2. This can be generalized to B taking values in a polynomial-sized alphabet Σ: $B \in \Sigma$ has pseudoentropy $\log|\Sigma|$ given X iff B cannot be predicted with probability noticeably more than $1/|\Sigma|$. Thus, in the extreme case of maximal pseudoentropy (equal to $\log|\Sigma|$), we have an equivalence with unpredictability.

2. For B that takes values in larger (say exponentially large) alphabets, Goldreich and Levin [GL] showed that if B is very hard to predict from X (i.e. cannot be predicted with nonnegligible probability), then we can choose a random hash function H whose range is a polynomial-sized set Σ and it will hold that $H(B) \in \Sigma$ has pseudoentropy $\log|\Sigma|$ given X and H. While this is very useful and has many applications, it does not characterize the pseudoentropy of B itself (but rather a hash of it), requires a hash function that supports "local list-decoding," and again only talks about maximal pseudoentropy ($\log|\Sigma|$).

3. As noted in [STV], the Hardcore Theorem of Impagliazzo [Imp] (and subsequent strengthenings [KS, Hol1, BHK]) can be interpreted as relating unpredictability and a kind of pseudoentropy. Specifically, when B is a single bit, the Hardcore Theorem tells us that B cannot be predicted from X with probability greater than $1-\delta$ iff "B is indistinguishable from a random bit on a 2δ fraction of the probability space (X, B)" (this fraction of the probability space is typically called the "hardcore measure"). One formalization of the latter condition is to say that (X, B) is indistinguishable from (X, C) where C has *average min-entropy* [DORS] at least $\log(1/\delta)$ given X. This result is of the same spirit as Theorem 1.3, but refers to average min-entropy rather than Shannon entropy, and does not distinguish between information-theoretic hardness and computational hardness.

[1] The *Shannon entropy* of a discrete random variable X is defined as $\mathrm{H}(X) = \mathbb{E}_{x\sim X}[\log(1/\Pr[X = x])]$.

[2] The *conditional (Shannon) entropy* of random variable Y given random variable Z is defined as $\mathrm{H}(Y|Z) = \mathbb{E}_{z\sim Z}[\mathrm{H}(Y|_{Z=z})]$.

[3] The *KL divergence* (a.k.a. *relative entropy*) from random variable Y to random variable Z is defined as $\mathbb{E}_{y\sim Y}[\log(\Pr[Y = y]/\Pr[Z = z])]$.

In light of the above similarities, it is natural that the proof of Theorem 1.3 follows the same overall structure as existing proofs of the Hardcore Theorem when showing that the hardness of sampling B given X implies the pseudoentropy of B given X. Specifically, our proof for the case of nonuniform complexity (i.e. circuit size) has the same structure as Nisan's proof of the Hardcore Theorem [Imp]. We assume for contradiction that B does not have pseudoentropy $\mathrm{H}(B|X) + \delta$ given X, i.e. B is distinguishable from every C such that $\mathrm{H}(C|X) \geq \mathrm{H}(B|X) + \delta$. Using the Min-Max Theorem, we deduce that there is a convex combination D of small circuits that is a universal distinguisher, i.e. $\Pr[D(X, B) = 1] - \Pr[D(X, C) = 1] > \epsilon$ for every C such that $\mathrm{H}(C|X) \geq \mathrm{H}(B|X) + \delta$. Next we show how to use such a D to sample a distribution $S(X)$ at small KL divergence from B (given X). It turns out that we can do this by exponentiating D — we take $S(X)$ to be such that $\Pr[S(X) = b] \propto 2^{kD(x,b)}$ where $k \in \mathbb{R}$ is the largest number such that $\mathrm{H}(S(X)|X) \geq \mathrm{H}(B|X) + \delta$. In statistical physics $C = S(X)$ is known as a Boltzmann distribution associated with D, and can be shown to minimize $\Pr[D(X, C) = 1]$ among all high-entropy C [LL]. Thus it is the "hardest" high-entropy distribution for D to distinguish from B. The proof that $S(X)$ has small KL divergence from B uses a new information-theoretic lemma saying that if C is a random variable obtained from exponentiating D in this way, then the KL divergence from (X, B) to (X, C) can be expressed exactly in terms of D's advantage in distinguishing (X, B) and (X, C).

For the case of uniform complexity (namely, probabilistic polynomial-time algorithms), we replace the use of the Min-Max Theorem with a new Uniform Min-Max Theorem, which constructively builds a near-optimal strategy of the second player in a 2-player game from several best-responses of the second player to strategies of the first player. We defer a detailed discussion of the Uniform Min-Max Theorem and its other applications to a forthcoming paper [VZ1], but we include the proof of the Uniform Min-Max Theorem in our technical report [VZ2] for reference. We note that the proof of the Uniform Min-Max Theorem also uses ideas from the proof of the Uniform Hardcore Theorem due to Barak, Hardt, and Kale [BHK].

1.2 Next-Bit Pseudoentropy from One-Way Functions

The Håstad, Impagliazzo, Levin, and Luby [HILL] construction of pseudorandom generators from one-way functions begins by showing how to use a one-way function to construct an efficiently samplable distribution X whose pseudoentropy is noticeably larger than its Shannon entropy. This approach was refined by Haitner et al. [HRV] using the following variant of pseudoentropy:

Definition 1.4 (next-block pseudoentropy [HRV], informal)**.** A sequence of jointly distributed random variables $(X_1, \ldots, X_m)$ has *next-block pseudoentropy* at least k iff there exist random variables $(Y_1, \ldots, Y_m)$, jointly distributed with $(X_1, \ldots, X_m)$ such that:

1. $(X_1, \ldots, X_{i-1}, X_i)$ is computationally indistinguishable from $(X_1, \ldots, X_{i-1}, Y_i)$, and
2. $\sum_i \mathrm{H}(Y_i|X_1, \ldots, X_{i-1}) \geq k$.

Equivalently, X_I has pseudoentropy at least k/m given $X_1, \ldots, X_{I-1}$, where I is uniformly distributed in $[m]$.

We say that a random variable X taking values in $\{0,1\}^m$ has *next-bit pseudoentropy* at least k iff when we break X into 1-bit blocks, then $X = (X_1, \ldots, X_m)$ has next-block pseudoentropy at least k.

Intuitively, next-bit pseudoentropy captures the pseudoentropy from the perspective of an adversary who gets the bits one at a time (from left to right), instead of all at once. Thus, the next-bit pseudoentropy of a random variable can be much larger than its pseudoentropy. For example, if $G : \{0,1\}^n \to \{0,1\}^m$ is a pseudorandom generator, then $(G(U_n), U_n)$ has next-bit pseudoentropy at least $m > n$, but does not have pseudoentropy larger than n.

Haitner, Reingold, and Vadhan [HRV] showed that if $f : \{0,1\}^n \to \{0,1\}^m$ is a one-way function, $X \in_R \{0,1\}^n$, and $H : \{0,1\}^n \to \{0,1\}^n$ is a random hash function from an appropriate family, then $(f(X), H, H(X))$ has next-bit pseudoentropy $n+r+\log n$, where r is the number of random bits used to describe the hash function H. The intuition for this is as follows: Condition on $f(X) = y$ for some $y \in \{0,1\}^n$. Given that $f(X) = y$, X is uniformly distributed in a set of size $|f^{-1}(y)|$. Thus, by the Leftover Hash Lemma [HILL], the first $\approx \log |f^{-1}(y)|$ bits of $H(X)$ are statistically close to uniform given the prefix preceding them. In addition, it is still difficult to invert f and predict X given these bits (since a uniform random string can't help in inverting). Thus, by the Goldreich–Levin Theorem [GL], the next $\approx \log n$ bits of $H(X)$ are computationally indistinguishable from uniform given the preceding bits. Therefore the next-bit pseudoentropy of $(f(X), H, H(X))$ is at least

$$\begin{aligned}\mathrm{H}(f(X)) + r + \mathop{\mathbb{E}}_{y \leftarrow f(X)}[\log |f^{-1}(y)|] + \log n \\ = \mathrm{H}(f(X)) + r + \mathrm{H}(X|f(X)) + \log n = n + r + \log n.\end{aligned}$$

Haitner, Reingold, and Vadhan [HRV] conjectured that the hashing in the above construction is not necessary, and the hardness of inverting a one-way function directly provides (next-bit) pseudoentropy. We prove their conjecture:

Theorem 1.5 (one-way functions ⇒ next-bit pseudoentropy)**.** *If $f : \{0,1\}^n \to \{0,1\}^m$ is a one-way function and $X \in_R \{0,1\}^n$, then $(f(X), X)$ has next-bit pseudoentropy at least $n + \log n$.*

The proof of this theorem starts by showing that the onewayness of f implies that for every probabilistic polynomial-time algorithm A, the KL divergence from $(f(X), X)$ to $(f(X), A(f(X)))$ is at least $\log n$; otherwise A would invert f with nonnegligible probability. Then we show that the same holds also in a "next-bit" sense: if we break X into bits $X = X_1 \cdots X_n$ and choose $I \in_R [n]$, then for every probabilistic polynomial-time S, the KL divergence from $(f(X), X_1, \ldots, X_I)$ to $(f(X), X_1, \ldots, X_{I-1}, S(f(X), X_1, \ldots, X_{I-1}))$ is at least $(\log n)/n$. (Otherwise by iteratively applying S n times, we can obtain a probabilistic polynomial-time A such that $(f(X), A(f(X)))$ has KL divergence at most $\log n$ from $(f(X), X)$.) By Theorem 1.3, we deduce that X_I has pseudoentropy at least $\mathrm{H}(X_I|f(X), X_1, \ldots, X_{I-1}) + (\log n)/n$ given $f(X), X_1, \ldots, X_{I-1}$. That is, on average, the individual bits of X have $(\log n)/n$ extra bits of pseudoentropy (beyond their Shannon entropy) given $f(X)$ and the previous bits of X. Summing over all n bits of X, the next-bit pseudoentropy is at least $\log n$ bits larger than the Shannon entropy of $(f(X), X)$, which is n.

1.3 Pseudorandom Generators

Given the next-bit pseudoentropy generator $(f(X), X) \in \{0,1\}^{m+n}$ of Theorem 1.5, we can apply the construction of Haitner et al. [HRV] to obtain a pseudorandom generator through the following three steps:

- **Entropy Equalization:** To spread the pseudoentropy out evenly among the bits, we concatenate $u = \tilde{\Theta}(n)$ independent random evaluations of $(f(X), X)$, then drop the first I bits and the last $m + n - I$ bits of the $u \cdot (n + m)$-bit long result, for $I \in_R [m + n]$.
- **Converting Shannon Entropy to Min-Entropy and Amplifying the Gap:** Next, we take $t = \tilde{\Theta}(n^2)$ copies of the above next-bit pseudoentropy generator (after entropy equalization), but concatenate them "vertically" to obtain blocks, each of which consists of t bits. It can be shown that each of the blocks is indistinguishable from having high *min-entropy* conditioned on the previous ones.
- **Randomness Extraction:** Finally, we use a single random universal hash function to extract the pseudo-min-entropy from each of the blocks, and concatenate the results to produce our output.

Thus, to obtain a pseudorandom generator from a one-way function f, we simply need to evaluate f on $u \cdot t = \tilde{O}(n^3)$ random inputs, arrange the input and output bits into a matrix consisting of $(u - 1) \cdot (m + n)$ columns and t rows, and apply a universal hash function to each column. (The seed of the pseudorandom generator consists of the $u \cdot t$ inputs to f, the t random shifts used for entropy equalization, and the description of the universal hash function.) The construction is illustrated in Figure 1. Note that we only need to hash once in the construction and the only property we need of our hash function is randomness extraction (e.g. via the Leftover Hash Lemma). In contrast, all previous constructions of pseudorandom generators from one-way functions (even from one-way permutations) required hash functions with "local list-decoding" properties (e.g. the Goldreich–Levin hardcore predicate) in addition to randomness extraction. As pointed out to us by Yuval Ishai, an advantage of using only universal hash functions is that they can be implemented by linear-size boolean circuits [IKOS], and thus we can obtain PRGs computable by circuits of size linear in their stretch (from one-way functions that are computable by linear-size circuits but exponentially hard to invert). Such PRGs have applications to "cryptography with constant computational overhead".

While simpler, the aforementioned construction achieves essentially the same parameters as [HRV]. Using an additional idea, we show how to save a factor of roughly $u = \tilde{\Theta}(n)$ in the seed length. The idea is that to extract the randomness from a column of the aforementioned matrix, we do not need to construct the entire matrix. We can use just enough seed to fill a single column, and then we can use randomness extracted from that column to help generate more columns, and iterate. (This idea is independent of our simplifications above, and can also be applied to the construction based on the [HRV] pseudoentropy generator.) Thus we show:

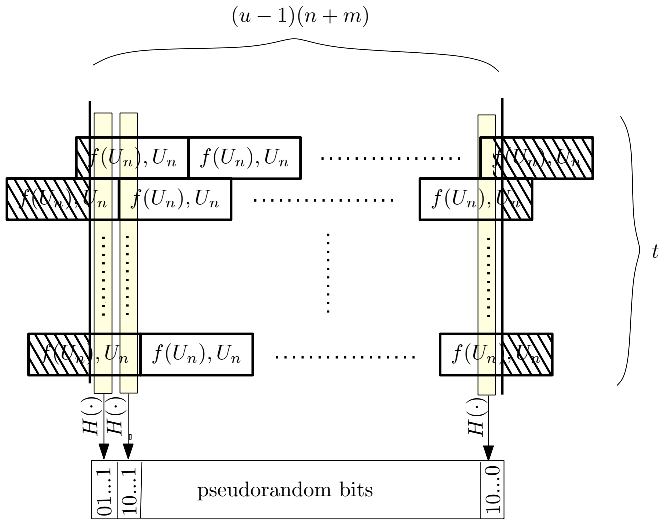

Figure 1: Simplified construction of PRG from one-way function $f : \{0,1\}^n \to \{0,1\}^m$
Each row contains iid copies of $(f(U_n), U_n)$, shifted by a random offset $I \in [n+m]$. To extract pseudorandom bits, an arbitrary universal hash function H (with a proper output length) is applied to all bits in the same column.

Theorem 1.6 (one-way functions ⇒ pseudorandom generators, informal)**.** *Given a one-way function $f : \{0,1\}^n \to \{0,1\}^n$, we can construct a pseudorandom generator with seed length $\tilde{O}(n^3)$.*

This theorem improves the seed length of $O(u \cdot t \cdot n) = \tilde{O}(n^4)$ from Haitner et al. [HRV]. We note that Haitner et al. gave a *nonuniform* construction of seed length $\tilde{O}(n^3)$, requiring $\text{poly}(n)$ bits of nonuniform advice to compute the pseudorandom generator. (Entropy equalization can be avoided by nonuniformly hardwiring the amount of entropy contributed by each bit.) Also, our construction still requires evaluating the one-way function at least $u \cdot t = \tilde{\Theta}(n^3)$ times; we just no longer need these evaluations to be independent. Finally, like [HRV], the construction obtains $\Theta(\log n)$ bits of additive stretch per invocation of the one-way function, which is optimal by [GGKT].

With Theorem 1.6, now the only blow-up in seed length in constructing pseudorandom generators from one-way functions is due to converting Shannon entropy to min-entropy. It is an intriguing open problem whether that blow-up can be avoided or shown to be necessary.

1.4 Relation to Inaccessible Entropy

A variety of computational notions of entropy have been studied in the cryptography and complexity literature, e.g. [Yao2, HILL, BSW, HLR, HRVW, HRV, HHR$^+$, FR, Rey]. In addition to the notions discussed in Sections 1.1 and 1.2, our work was also inspired by the works on *inaccessible entropy* [HRVW, HHR$^+$].

Like our characterization of conditional pseudoentropy, inaccessible entropy refers to a difficulty of sampling a random variable B from a jointly distributed random variable X. However, there are important differences. In our characterization (Theorem 1.3), the sample of X is generated externally and fed to the adversary, who tries then to sample the conditional distribution $B|X$. In the [HHR$^+$] notion of inaccessible entropy, the adversary is also given the random coins used to generate X, and we compare its output distribution conditioned on those coins to $B|X$. And in the original notion of inaccessible entropy, from [HRVW], the adversary is the one who generates X (or some approximation to it). These three notions are analogous to the security conditions for one-way functions, target collision-resistant hash functions (i.e. UOWHFs), and collision-resistant hash functions, respectively (thinking of $X = f(B)$ for $B \in_R \{0,1\}^n$). We note that the hardness of sampling we consider also differs from inaccessible entropy in the way it measures how well an adversary approximates the conditional distribution $B|X$. Roughly speaking, in our notion (measuring the KL divergence from $B|X$ to the adversary's output), the adversary's goal is to produce an output distribution that *contains* $B|X$ as tightly as possible. In the notions of inaccessible entropy, the adversary's goal is to produce an output distribution that is *contained within* $B|X$ as tightly as possible.

There is also significant similarity between our construction and those involving inaccessible entropy. In [HRVW], it is shown that if f is a one-way function, then $(f(U_n), U_n)$ is a next-bit inaccessible entropy generator, just like we show that it is a next-bit pseudoentropy generator (Theorem 1.5). However, for inaccessible entropy, it is only necessary to break $f(U_n)$ into bits (U_n can be treated as a single block), and for pseudoentropy it is only necessary to break U_n into bits ($f(U_n)$ can be treated as a single block). Nevertheless, there are enough similarities to suggest that there may be a deeper connection between inaccessible entropy and pseudoentropy; trying to formalize this connection is an interesting question for future work.

1.5 Paper Organization

Basic notions of information theory and computational randomness are defined in Section 2. In Section 3 we describe and prove our characterization of pseudoentropy. In Section 4 we show how to generate next-bit pseudoentropy from any one-way function. In Section 5 we describe the PRG construction and how to save the seed length.

2. PRELIMINARIES

2.1 Entropy

Shannon entropy plays a central role in this paper. For more background on entropy and proofs of the lemmas stated here, see [CT].

Definition 2.1 (entropy)**.** For a random variable X, the *(Shannon) entropy of* X is defined to be

$$\mathrm{H}(X) = \mathop{\mathbb{E}}_{x \sim X}\left[\log \frac{1}{\Pr[X = x]}\right].$$

For jointly distributed random variables X and B, the *conditional (Shannon) entropy of* B *given* X (or, *conditional (Shannon) entropy of* B when X is clear from the context) is defined to be

$$\mathrm{H}(B|X) = \mathop{\mathbb{E}}_{x \sim X}\left[\mathrm{H}(B|_{X=x})\right].$$

Fact 2.2 (chain rule for Shannon entropy)**.** $\mathrm{H}(X, B) = \mathrm{H}(X) + \mathrm{H}(B|X)$.

The notion of *KL divergence* from random variable A to random variable B is closely related to Shannon entropy; intuitively it measures how dense A is within B, on average (with 0 divergence representing maximum density, i.e. $A = B$, and large divergence meaning that A is concentrated in a small portion of B).

Definition 2.3 (KL divergence)**.** For random variables A and B, the *KL divergence from* A *to* B is defined to be

$$\mathrm{KL}(A||B) = \mathop{\mathbb{E}}_{a \sim A}\left[\log \frac{\Pr[A = a]}{\Pr[B = a]}\right]$$

or conventionally $+\infty$ if $\Pr[A = a] > 0$ and $\Pr[B = a] = 0$ for some a.

For random variables (X, A) and (Y, B), the *conditional KL divergence from* $A|X$ *to* $B|Y$ is defined to be

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

Thus, conditional KL divergence captures the expected KL divergence from $A|_{X=x}$ to $B|_{Y=x}$, over $x \sim X$. Like Shannon entropy, it has a chain rule:

Fact 2.4 (chain rule for KL divergence)**.** $\mathrm{KL}(X, A||Y, B) = \mathrm{KL}(X||Y) + \mathrm{KL}((A|X)||(B|Y))$.

Like other distance measures between distributions, applying any (deterministic) function never increases the KL divergence:

Fact 2.5 (entropy-like property of KL divergence)**.** [4] $\mathrm{KL}(g(A)||g(B)) \leq \mathrm{KL}(A||B)$ *for any function* g.

Note however, that the KL divergence is *not* a metric; it is not symmetric and does not satisfy the triangle inequality.

2.2 Pseudorandom Generators

First, we define the computational analogue of two random variables being statistically close:

Definition 2.6 (indistinguishability)**.** Let n be a security parameter. Two $\{0,1\}^n$-valued random variables $X = X(n)$ and $Y = Y(n)$ are (T, ϵ) *indistinguishable* for $T = T(n)$, $\epsilon = \epsilon(n)$ if for all time T randomized algorithm D and all sufficiently large n, $|\Pr[D(X) = 1] - \Pr[D(Y) = 1]| \leq \epsilon$.

$|\Pr[D(X) = 1] - \Pr[D(Y) = 1]|$ is called D's *distinguishing advantage for* X, Y.

A *pseudorandom generator* is an algorithm that stretches a short uniformly random string to a longer *pseudorandom* string, one which *looks* random even to algorithms more powerful than the generator itself:

Definition 2.7 (pseudorandom)**.** Let n be a security parameter, $q = q(n)$. A $[q]$-valued random variable X is (T, ϵ) *pseudorandom* for $T = T(n)$, $\epsilon = \epsilon(n)$ if X and $U_{[q]}$ are (T, ϵ) indistinguishable. A polynomial time computable function $G : \{0,1\}^{d=d(n)} \to \{0,1\}^{\ell=\ell(n)}$ is a (T, ϵ) *pseudorandom generator* (PRG), if $G(U_d)$ is (T, ϵ) pseudorandom.

We say G is a *pseudorandom generator* if G is a $(n^c, 1/n^c)$ pseudorandom generator for every constant c. The input to a pseudorandom generator is called the *seed.* The number of extra bits, $\ell - d$, is called the *stretch.*

While the notions of indistinguishability and pseudorandom generators here are defined for uniform algorithms, nonuniform indistinguishability and nonuniform pseudorandomness can be defined by replacing time T algorithms with size T boolean circuits.

[4] This is in fact equivalent to the *log-sum inequality* [CT]. For a more direct proof, see [GV].

2.3 Pseudoentropy and Next-Bit Pseudoentropy

The computational analogue of entropy, *pseudoentropy*, was first introduced by Hastad et al. [HILL]. We begin with the nonuniform definition because it is simpler:

Definition 2.8 (pseudoentropy, nonuniform setting)**.** Let X be a random variable. We say X has (T, ϵ) *nonuniform pseudoentropy at least* k if there exists a random variable Y with $\mathrm{H}(Y) \geq k$ such that X and Y are (T, ϵ) nonuniformly indistinguishable.

If $X = X(n)$ for a security parameter n, we say X has *nonuniform pseudoentropy at least* $k = k(n)$ if for every constant c, $X(n)$ has $(n^c, 1/n^c)$ nonuniform pseudoentropy at least $k(n) - 1/n^c$ for all sufficiently large n.

A natural generalization of *pseudoentropy* is the notion of *conditional* pseudoentropy.

Definition 2.9 (conditional pseudoentropy, nonuniform setting)**.** Let B be a random variable jointly distributed with X. We say B has (T, ϵ) *nonuniform (conditional) pseudoentropy at least* k (or *pseudoentropy gap at least* $k - \mathrm{H}(B|X)$) *given* X if there exists a random variable C jointly distributed with X such that the following holds:

- $\mathrm{H}(C|X) \geq k$;
- (X, B) and (X, C) are (T, ϵ)-indistinguishable.

If $B = B(n)$ for a security parameter n, we say B has *nonuniform (conditional) pseudoentropy at least* $k = k(n)$ *given* X if for every constant c, $B(n)$ has $(n^c, 1/n^c)$ nonuniform (conditional) pseudoentropy at least $k(n) - 1/n^c$ given $X(n)$ for all sufficiently large n.

In the uniform setting (i.e. randomized algorithms instead of circuits), the right definitions are more subtle to come by. It turns out that we must require indistinguishability even against algorithms equipped with an sampling oracle. (See remark below for more discussion.)

Notation. For a distribution Z, let O_Z denote the oracle that gives a random sample from Z when queried.

Definition 2.10 (pseudoentropy, uniform setting)**.** Let n be a security parameter, $T = t(n)$, $\epsilon = \epsilon(n)$, $k = k(n)$, $q = q(n)$. Let X be a $[q]$-valued random variable. We say X has (T, ϵ) *uniform pseudoentropy at least* k if for all time T randomized oracle algorithm A there exists a random variable Y jointly distributed[5] with X such that the following holds for all sufficiently large n:

- $\mathrm{H}(Y) \geq k$;
- X, Y are indistinguishable by $A^{O_{X,Y}}$:
$$\left|\Pr[A^{O_{X,Y}}(X) = 1] - \Pr[A^{O_{X,Y}}(Y) = 1]\right| \leq \epsilon.$$

We say X has *uniform pseudoentropy at least* $k = k(n)$ if for every constant c, $X(n)$ has $(n^c, 1/n^c)$ uniform pseudoentropy at least $k(n) - 1/n^c$.

The reason to give the distinguishers oracle access to $O_{X,Y}$ is to ensure that the definition composes: if X_1 and X_2 are iid copies of X, we'd like to say that (X_1, X_2) has pseudoentropy at least $2k$. Indeed we'd want to say that (X_1, X_2) is indistinguishable from (Y_1, Y_2) where Y_1, Y_2 are iid copies of Y. However, indistinguishability against uniform algorithms is not preserved under taking multiple independent samples in general [GM1]. Requiring indistinguishability against distinguishers with oracle access to $O_{X,Y}$ ensures that indistinguishability will be preserved under taking multiple independent samples.

Definition 2.11 (conditional pseudoentropy, uniform setting)**.** Let n be a security parameter, $T = t(n)$, $\epsilon = \epsilon(n)$, $k = k(n)$, $q = q(n)$. Let B be a $[q]$-valued random variable jointly distributed with X. We say B has (T, ϵ) *uniform (conditional) pseudoentropy at least* k *given* X if for every randomized oracle algorithm A computable in time T, there is a random variable C jointly distributed with X, B such that the following holds for all sufficiently large n:

- $\mathrm{H}(C|X) \geq k$;
- (X, B) and (X, C) are indistinguishable by $A^{O_{X,B,C}}$:
$$|\Pr[A^{O_{X,B,C}}(X, B) = 1] - \Pr[A^{O_{X,B,C}}(X, C) = 1]| \leq \epsilon.$$

We say B has *uniform (conditional) pseudoentropy at least* $k = k(n)$ *given* X if for every constant c, $B(n)$ has $(n^c, 1/n^c)$ uniform (conditional) pseudoentropy at least $k(n) - 1/n^c$ given $X(n)$.

We give the distinguishers oracle access to $O_{X,B,C}$ for the same reason as we give oracle access to $O_{X,Y}$ in Definition 2.10. However, a consequence of our results is that the definition with oracle $O_{X,B,C}$ is equivalent to the definition with oracle $O_{X,B}$ provided B comes from a polynomial-sized alphabet. In particular, if (X, B) is also polynomial-time samplable (which will be the case in our applications), the definition is equivalent to one without oracle $O_{X,B,C}$. (See Corollary 3.23.)

Finally, it is useful to talk about the *total* conditional pseudoentropy of a sequence of random variables, called the *next-block pseudoentropy*:

Definition 2.12 (next-block pseudoentropy)**.** Let n be a security parameter, $k = k(n)$, and $B^{(i)}$ be a random variable for each $i = 1, \ldots, m = m(n)$. We say $\left(B^{(1)}, B^{(2)}, \ldots\right)$ has *(non)uniform next-block* (or *next-bit*, if each $B^{(i)}$ is a bit) *pseudoentropy at least* k if $B^{(I)}$ has (non)uniform pseudoentropy at least k/m given $B^{(1)} \ldots B^{(I-1)}$, for $I \in_R [m]$.

It is easy to see that next-bit pseudoentropy is a weaker notion than pseudoentropy. Therefore we would like "blocks" to be small, ideally bits, to increase the next-block pseudoentropy. Note that the next-bit pseudoentropy is sensitive to the order of the bits; for example, for any one-way function f, $(U_n, f(U_n))$ does not have next-bit pseudoentropy $n + 1$, but $(f(U_n), U_n)$ has next-bit pseudoentropy at least $n + \Omega(\log n)$ as we show in Section 3.

3. CHARACTERIZING PSEUDOENTROPY

In this section, we show that a random variable B having pseudoentropy given X, is equivalent to B being *KL-hard*

[5] In Definition 2.10, Y can be taken to be independent of X without loss of generality, but allowing dependence is important in the definition of conditional pseudoentropy.

given X, which roughly captures the hardness of generating B from X in terms of KL divergence. We prove the equivalence in both nonuniform and uniform models of computation.

To state the mains results precisely, we begin with basic conventions and definitions. We will work with random variables taking values in $[q]$, which are jointly distributed with a $\{0,1\}^n$-valued random variable X. For any $[q]$-valued random variable C jointly distributed with X, we write $C(a|x) = \Pr[C = a|X = x]$. We will drop "jointly distributed with X" when it is clear from the context.

Such a jointly distributed r.v. C can be algorithmically represented in two ways: (i) By a randomized algorithm S that samples C from X, i.e. $C = S(X)$; (ii) By an algorithm P that computes the (conditional) probability mass function (pmf) of C, i.e. $P(x,a) = \Pr[C = a|X = x]$. In general, having an efficient algorithm for one representation does not imply having an efficient algorithm for the other (under some complexity assumptions) [KMR$^+$, Nao]. But when the alphabet size q is small, approximating the pmf of C given X (say to within $\pm\epsilon$) is equivalent to approximately sampling C given X (say to within statistical distance ϵ), up to a factor of $\mathrm{poly}(q, 1/\epsilon)$ in running time. (See Lemmas 3.6, 3.7 below.)

A drawback of the pmf representation is that it can be infeasible to maintain the normalization $\sum_a P(x,a) = 1$ when manipulating the random variable if the alphabet size q is large. Thus it is convenient to work with *measures* instead of pmf. A function $P : \{0,1\}^n \times [q] \to (0,+\infty)$ is called a (conditional) *measure* of the random variable C_P defined as follows:

$$C_P(a|x) = \frac{P(x,a)}{\sum_b P(x,b)}.$$

Thus a measure is just some scalar multiple of the pmf. In this section, we generalize the pmf representation so that P only has to compute some (conditional) measure of C.

Definition 3.1 (KL predictors)**.** Let (X,B) be a $\{0,1\}^n \times [q]$-valued random variable, and $P : \{0,1\}^n \times [q] \to (0,+\infty)$ a deterministic function. We say that P is a *δ-KL predictor* of B given X if

$$\mathrm{KL}(X,B||X,C_P) \le \delta.$$

If P is randomized, we say that P is a *δ-KL predictor* of B given X if

$$\mathop{\mathbb{E}}_{p\sim P}[\mathrm{KL}(X,B||X,C_p)] \le \delta,$$

where we view P as a distribution over functions $p : \{0,1\}^n \times [q] \to (0,+\infty)$.

Definition 3.2 (KL-hard, nonuniform setting)**.** Let (X,B) be a $\{0,1\}^n \times [q]$-valued random variable, for $q = q(n)$. We say B is *nonuniformly (t,δ) KL-hard given X* if there is no circuit P of size t that is a δ-KL predictor of B given X.

We say B is *nonuniformly δ KL-hard given X* if for every constant c, B is nonuniformly $(n^c, \delta - 1/n^c)$ KL-hard given X for all sufficiently large n.

Analogously to pseudoentropy, the nonuniform and uniform definitions differ in whether we need to give a sampling oracle to the adversary.

Definition 3.3 (KL-hard, uniform setting)**.** Let n be a security parameter, $\delta = \delta(n) > 0$, $t = t(n) \in \mathbb{N}$, $q = q(n)$. Let (X,B) be a $\{0,1\}^n \times [q]$-valued random variable. We say B is *uniformly (t,δ) KL-hard given X* if for all time t randomized oracle algorithms $P : \{0,1\}^n \times [q] \to (0,+\infty)$ and all sufficiently large n, $P^{O_{X,B}}$ is not a δ-KL predictor of B given X (where the randomness of $P^{O_{X,B}}$ consists both of its internal coin tosses and the samples it gets from the oracle $O_{X,B}$).

We say B is *uniformly δ KL-hard given X* if for every constant c, B is uniformly $(n^c, \delta - 1/n^c)$ KL-hard given X.

Note that by letting $P(x,a) = 1$, we already get $C = U_{[q]}$ i.e. $\mathrm{KL}(X,B||X,C) = \log q - \mathrm{H}(B|X) \le \log q$. Thus it only makes sense to talk about KL-hardness for $\delta \le \log q$.

The following related definition may be more natural, as a closer parallel to the familiar notion of average-case hardness:

Definition 3.4 (KL-hard for sampling, nonuniform setting)**.** Let (X,B) be a $\{0,1\}^n \times [q]$-valued random variable, for $q = q(n)$. We say B is *nonuniformly (t,δ) KL-hard for sampling given X* if for all size t randomized circuits $S : \{0,1\}^n \to [q]$ it holds that $\mathrm{KL}(X,B||X,S(X)) > \delta$.

Definition 3.5 (KL-hard for sampling, uniform setting)**.** Let n be a security parameter, $\delta = \delta(n) > 0$, $t = t(n) \in \mathbb{N}$, $q = q(n)$. Let (X,B) be a $\{0,1\}^n \times [q]$-valued random variable. We say B is *uniformly (t,δ) KL-hard for sampling given X* if for all time t randomized oracle algorithms S, for all sufficiently large n, it holds that $\mathrm{KL}(X,B||X,S^{O_{X,B}}(X)) > \delta$.

These two notions are equivalent up to a polynomial factor in t, provided that size of the alphabet q is a polynomial:

Lemma 3.6. *Let (X,B) be a $\{0,1\}^n \times [q]$-valued random variable. If B is nonuniformly (t,δ) KL-hard for sampling given X, then B is nonuniformly $(\Omega(t/q),\delta)$ KL-hard given X. Conversely, if B is nonuniformly (t,δ) KL-hard given X, then B is nonuniformly $(t', \delta - \epsilon)$ KL-hard for sampling given X for $t' = t/\mathrm{poly}(n,q,1/\epsilon)$, for every $\epsilon > 0$.*

Proof. Suppose B is not nonuniformly (t',δ) KL-hard given X. That is, there exists a size t' circuit P such that $\mathrm{KL}(X,B ||X,C_P) \le \delta$. Then we can sample $S(x) = a$ w.p. $C_P(a|x)$ so that $\mathrm{KL}(X,B||X,S(X)) \le \delta$. S has circuit size $O(q \cdot t')$. This contradicts the fact that B is nonuniformly (t,δ) KL-hard for sampling, for $t' = \Omega(t/q)$.

Conversely, suppose $\mathrm{KL}(X,B||X,S(X)) \le \delta - \epsilon$ for some size t' circuit S. We will construct a size t *randomized* δ-KL predictor P (so that it will be useful for the uniform setting, Lemma 3.7, as well) as follows. We compute $E(x,a)$ such that w.p. at least $1-\gamma$, $|\Pr[S(x) = a] - E(x,a)| \le \epsilon^2/c^2q$ for all x, a, where c is a large enough constant. This is done by taking $m = O(n + \log q + \log(1/\gamma)) \cdot q^2/\epsilon^4$ samples of the randomness of S. We then output $P(x,a) = \max\{E(x,a), \epsilon/cq\} \in (\epsilon/cq, 1]$.

We view P as a distribution over functions $p : \{0,1\}^n \times [q] \to (\epsilon/cq, 1]$. Consider any $p \in \mathrm{supp}(P)$ such that $|\Pr[S(x) = a] - E(x,a)| \le \epsilon^2/c^2q$ for all x, a. Notice that $\sum_b p(x,b) \le 1 + q \cdot (\epsilon/cq) = 1 + \epsilon/c$. If $\Pr[S(x) = a] > \epsilon/cq$, then

$$\begin{aligned}\log \frac{\Pr[S(x) = a]}{C_p(a|x)} &\le \log \frac{p(x,a) + \epsilon^2/c^2}{p(x,a)} + \log \sum_b p(x,b) \\ &\le \log(1+\epsilon/c) + \log(1+\epsilon/c) \le \frac{\epsilon}{2}.\end{aligned}$$

If $\Pr[S(x) = a] \leq \epsilon/cq$, then

$$\begin{aligned}\log \frac{\Pr[S(x) = a]}{C_p(a|x)} &= \log \frac{\Pr[S(x) = a]}{p(x,a)} + \log \sum_b p(x,b) \\ &\leq \log(1+\epsilon/c) \leq \frac{\epsilon}{2}.\end{aligned}$$

Thus we get

$$\begin{aligned}\mathrm{KL}(X,B||X,C_p) &= \mathrm{KL}(X,B||X,S(X)) \\ &\quad + \mathop{\mathbb{E}}_{x\sim X}\left[\sum_a B(a|x)\log\frac{\Pr[S(x)=a]}{C_p(a|x)}\right] \\ &\leq \delta - \epsilon + \frac{\epsilon}{2}.\end{aligned}$$

On the other hand, for every $p : \{0,1\}^n \times [q] \to (\epsilon/cq, 1]$ it holds that

$$\begin{aligned}\mathrm{KL}(X,B||X,C_p) &= \mathbb{E}\left[\sum_a B(a|X)\log\left(B(a|X)/C_p(a|X)\right)\right] \\ &\leq \max_{x,a} \log\left(1/C_p(a|x)\right) \\ &= O\left(\log q + \log\frac{1}{\epsilon}\right).\end{aligned}$$

Thus,

$$\begin{aligned}&\mathop{\mathbb{E}}_{p\sim P}[\mathrm{KL}(X,B||X,C_p)] \\ &\leq (1-\gamma)\cdot(\delta - \frac{\epsilon}{2}) + \gamma \cdot O\left(\log q + \log\frac{1}{\epsilon}\right) \leq \delta\end{aligned}$$

for an appropriate choice of $\gamma = O(\epsilon/(\log q + \log(1/\epsilon)))$. Furthermore, P has circuit size $O(t'm) = t$. Thus B is not nonuniformly (t,δ) KL-hard given X. □

Lemma 3.7. *Let n be a security parameter, $\delta = \delta(n) > 0$, $t = t(n) \in \mathbb{N}$, $p = p(n)$, $\epsilon = \epsilon(n) > 0$, $q = q(n)$ all computable in time* poly(n). *Let (X,B) be a $\{0,1\}^n \times [q]$-valued random variable. If B is uniformly (t,δ) KL-hard for sampling given X, then B is uniformly $(\Omega(t/(q+n)),\delta)$ KL-hard given X. Conversely, if B is uniformly (t,δ) KL-hard given X, then B is uniformly $(t', \delta-\epsilon)$ KL-hard for sampling given X, for $t' = t/\mathrm{poly}(n,q,1/\epsilon)$.*

Proof. The proof for the second part is identical to Lemma 3.6. For the first part, suppose B is not uniformly (t',δ) KL-hard given X. That is, there is a time t' oracle algorithm P such that when $P^{O_{X,B}}$ is viewed as a distribution over functions $p : \{0,1\}^n \times [q] \to (0,+\infty)$, for infinitely many n,

$$\mathbb{E}_{p\ \sim P^{O_{X,B}}}\left[\mathrm{KL}(X,B||X,C_p)\right] \leq \delta.$$

Then we can sample $S(x) = a$ w.p. $\mathbb{E}_{p\sim P^{O_{X,B}}}[C_p(a|x)]$, where we first pick $p \sim P^{O_{X,B}}$ by fixing the internal coin tosses of P and samples from oracle $O_{X,B}$. By convexity of $\mathrm{KL}(X,B||X,\cdot)$,

$$\begin{aligned}\mathrm{KL}(X,B||X,S(X)) &= \mathrm{KL}\left(X,B||X,C_{P^{O_{X,B}}}\right) \\ &\leq \mathbb{E}_{p\sim P^{O_{X,B}}}\left[\mathrm{KL}\left(X,B||X,C_p\right)\right] \leq \delta.\end{aligned}$$

This contradicts the fact that B is uniformly (t,δ) KL-hard for sampling, for $t' = \Omega(t/(q+n))$. □

In this section, it is more convenient to work with the first version of KL-hardness (i.e. not for sampling). We show the following main results which establish equivalence between (conditional) pseudoentropy and KL-hardness in both nonuniform and uniform settings.

Theorem 3.8 (Main Theorem, nonuniform setting)**.** *Let (X,B) be a $\{0,1\}^n \times [q]$-valued random variable, $\delta > 0$, $\epsilon > 0$.*

1. *If B is nonuniformly (t,δ) KL-hard given X, then for every $\epsilon > 0$, B has nonuniform (t',ϵ) pseudoentropy at least* $\mathrm{H}(B|X)+\delta-\epsilon$ *given X, for $t' = t^{\Omega(1)}/\mathrm{poly}(n,\log q, 1/\epsilon)$.*
2. *Conversely, if B has nonuniform (t,ϵ) pseudoentropy at least* $\mathrm{H}(B|X)+\delta$ *given X, then for every $\sigma > 0$, B is nonuniformly (t',δ') KL-hard given X, for $t' = \min\{t^{\Omega(1)}/\mathrm{polylog}\,(1/\sigma)\,, \Omega(\sigma/\epsilon)\}$ and $\delta' = \delta - \sigma$.*

Corollary 3.9. *Let (X,B) be a $\{0,1\}^n \times [q]$-valued random variable. Then B has nonuniform pseudoentropy at least* $\mathrm{H}(B|X)+\delta$ *given X if and only if B is nonuniformly δ KL-hard given X.*

By dropping X, the polylog(q) dependence gives us a characterization of *nonuniform pseudoentropy* for an n-bit random variables: (Note that without conditioning on X, the definition of KL-hard still makes sense, expressing the hardness of computing a measure that approximates the distribution B.)

Corollary 3.10. *An n-bit random variable B has nonuniform pseudoentropy at least* $\mathrm{H}(B)+\delta$ *if and only if B is nonuniformly δ KL-hard.*

We now state the uniform versions of our results, which are analogous to the nonuniform versions but have a *polynomial* dependence on q (we do not know whether it can be made polylogarithmic like in Theorem 3.8, so we don't have a uniform analogue of Corollary 3.10.)

Theorem 3.11 (Main Theorem, uniform setting)**.** *Let n be a security parameter, $\delta = \delta(n) > 0$, $t = t(n) \in \mathbb{N}$, $\epsilon = \epsilon(n) > 0$, $q = q(n)$, $\sigma = \sigma(n)$ all computable in time* poly(n). *Let (X,B) be a $\{0,1\}^n \times [q]$-valued random variable.*

1. *If B is uniformly (t,δ) KL-hard given X, then B has uniform (t',ϵ) pseudoentropy at least* $\mathrm{H}(B|X)+\delta-\epsilon$ *given X, for $t' = t^{\Omega(1)}/\mathrm{poly}(n,q,1/\epsilon)$.*
2. *Conversely, if B has uniform (t,ϵ) pseudoentropy at least* $\mathrm{H}(B|X)+\delta$ *given X, then B is uniformly (t',δ') KL-hard given X, for $t' = \min\{t^{\Omega(1)}/\mathrm{poly}(n,\log(1/\sigma),$ $\Omega(\sigma/\epsilon)\}$ and $\delta' = \delta - \sigma$.*

Corollary 3.12. *Let n be a security parameter, $\delta = \delta(n) > 0$, $q = \mathrm{poly}(n)$ computable in time* poly(n). *Let (X,B) be a $\{0,1\}^n \times [q]$-valued random variable. Then B has uniform pseudoentropy at least* $\mathrm{H}(B|X)+\delta$ *given X if and only if B is uniformly δ KL-hard given X.*

Note that we do not make any samplability assumption on X (in both nonuniform and uniform settings).

Distinguishers are a central object in studying pseudoentropy. A distinguisher D is a $\{0,1\}$-valued randomized function, and $D(x)$ denotes the probability that the function outputs 1 on input $x \in \{0,1\}^*$. A *generalized distinguisher* D is a $\mathbb{R}^+$-valued randomized function, and $D(x)$ denotes the expectation of the output on input x. For generalized distinguishers D_1 and D_2, the scalar multiple kD_1 ($k \geq 0$) and the sum $D_1 + D_2$ are also generalized distinguishers.

A generalized distinguisher D is said to have distinguishing advantage $\mathrm{Adv}_D(X, Y) = \mathbb{E}[D(X)] - \mathbb{E}[D(Y)]$ between random variables X, Y. Thus for random variables (X, B), (X, C):

$$\begin{aligned}&\mathrm{Adv}_D((X,B),(X,C))\\&= \mathbb{E}[D(X,B) - D(X,C)]\\&= \mathop{\mathbb{E}}_X\left[\sum_a D(X,a)(B(a|X) - C(a|X))\right].\end{aligned}$$

A key idea in our argument is to analyze the random variable $\mathbf{2}^D$ for a generalized distinguisher D, defined as

$$\mathbf{2}^D(a|x) = \frac{2^{D(x,a)}}{\sum_b 2^{D(x,b)}}.$$

This is a conditional version of the *Boltzmann distribution* (or *Gibbs distribution*; *canonical ensemble*) in statistical physics [LL], which is the unique distribution that achieves maximum entropy under a linear constraint on the pmf. We consider the conditional Boltzmann distribution in our context for a similar reason: for any distinguisher D, it turns out that $C = \mathbf{2}^{kD}$ ($k \geq 0$) minimizes $\mathrm{Adv}_D((X, B), (X, C))$ among all C with $\mathrm{H}(C|X) \geq r = \mathrm{H}(\mathbf{2}^{kD}|X)$. (The unconditional version is well known in statistical physics [LL]. We give a simple proof for the conditional version in Lemma 3.18). Thus a lower bound on $\mathrm{Adv}_D((X, B), (X, C))$ for all C with $\mathrm{H}(C|X) \geq r$ is equivalent to a lower bound for $C = \mathbf{2}^{kD}$.

In particular, we are able to relate $\mathrm{Adv}_D((X, B), (X, \mathbf{2}^D))$ to the KL divergence from (X, B) to $(X, \mathbf{2}^D)$ and the entropies of these random variables by the following key lemma:

Lemma 3.13. *Let (X, B) be a $\{0, 1\}^n \times [q]$-valued random variable, D be a generalized distinguisher. Then*

$$\mathrm{KL}(X, B||X, \mathbf{2}^D) = \mathrm{H}(\mathbf{2}^D|X) - \mathrm{H}(B|X) - \mathrm{Adv}_D((X, B), (X, \mathbf{2}^D)).$$

Proof.

$$\begin{aligned}&\mathrm{KL}(X,B||X,\mathbf{2}^D)\\&= \mathop{\mathbb{E}}_X\left[\sum_a B(a|X)\log\frac{B(a|X)}{\mathbf{2}^D(a|X)}\right]\\&= \mathrm{H}(\mathbf{2}^D|X) - \mathrm{H}(B|X)\\&\quad + \mathop{\mathbb{E}}_X\left[\sum_a (B(a|X) - \mathbf{2}^D(a|X))\log\frac{1}{\mathbf{2}^D(a|X)}\right]\\&= \mathrm{H}(\mathbf{2}^D|X) - \mathrm{H}(B|X)\\&\quad + \mathop{\mathbb{E}}_X\left[\sum_a (B(a|X) - \mathbf{2}^D(a|X))\left(\log\sum_b 2^{D(X,b)} - D(X,a)\right)\right]\\&= \mathrm{H}(\mathbf{2}^D|X) - \mathrm{H}(B|X)\\&\quad + \mathop{\mathbb{E}}_X\left[-\sum_a D(X,a)(B(a|X) - \mathbf{2}^D(a|X))\right]\\&= \mathrm{H}(\mathbf{2}^D|X) - \mathrm{H}(B|X) - \mathrm{Adv}_D((X,B),(X,\mathbf{2}^D)).\end{aligned}$$

□

We note that with $D(x, a) \equiv 0$, this becomes the familiar $\mathrm{KL}(X, B||X, U_{[q]}) = \log q - \mathrm{H}(B|X)$. To quickly see why this lemma is useful: suppose D has good performance distinguishing $\mathbf{2}^D$ from B, then we can use $\mathbf{2}^D$ to predict B within small KL divergence; this is essentially the idea why KL-hardness implies pseudoentropy, at least in the nonuniform setting (Part 1 of Theorem 3.8).

3.1 KL-hardness Implies Pseudoentropy, Nonuniform Setting

We begin with the main technical ingredient of pseudoentropy implying KL-hardness.

Lemma 3.14. *Let (X, B) be a $\{0, 1\}^n \times [q]$-valued random variable where $\mathrm{H}(B|X) \leq \log q - \delta$ for some $\delta \geq 0$. Let $\epsilon > 0$, and D be a distinguisher such that $\mathrm{Adv}_D((X, B), (X, C)) > \epsilon$ for all C with $\mathrm{H}(C|X) \geq \mathrm{H}(B|X) + \delta$. Then there exists $k \in [0, (\log q)/\epsilon]$ such that $\mathrm{KL}(X, B||X, \mathbf{2}^{kD}) \leq \delta$.*

This lemma says that a *universal* distinguisher D — one that distinguishes B from *all* high-entropy C's — can be used to approximate B to within small KL divergence.

Proof of Lemma 3.14. Let $k_0 = (\log q)/\epsilon$. First we show there exists $k \in [0, k_0]$ such that $\mathrm{H}(\mathbf{2}^{kD}|X) = \mathrm{H}(B|X) + \delta$. By Lemma 3.13,

$$\begin{aligned}&\mathrm{Adv}_D((X,B),(X,\mathbf{2}^{k_0 D}))\\&= \frac{1}{k_0}\left(\mathrm{H}(\mathbf{2}^{k_0 D}|X) - \mathrm{H}(B|X) - \mathrm{KL}(X,B||X,\mathbf{2}^{k_0 D})\right)\\&\leq \frac{\log q}{k_0} = \epsilon,\end{aligned}$$

where we use nonnegativity of entropy and KL divergence. Thus, by assumption $\mathrm{H}(\mathbf{2}^{k_0 D}|X) < \mathrm{H}(B|X) + \delta$. Now we know (i) $\mathrm{H}(\mathbf{2}^{k_0 D}|X) < \mathrm{H}(B|X) + \delta$, (ii) $\mathrm{H}(\mathbf{2}^0|X) = \log q \geq \mathrm{H}(B|X) + \delta$, and (iii) $\mathrm{H}(\mathbf{2}^{kD}|X)$ is continuous as a funciton of $k \in [0, +\infty)$. By the Intermediate Value Theorem, there exists $k \in [0, k_0]$ such that $\mathrm{H}(\mathbf{2}^{kD}|X) = \mathrm{H}(B|X) + \delta$.

Given such k, applying Lemma 3.13 again we get:

$$\begin{aligned}&\mathrm{KL}(X,B||X,\mathbf{2}^{kD})\\&= \mathrm{H}(\mathbf{2}^{kD}|X) - \mathrm{H}(B|X) - \mathrm{Adv}_{kD}((X,B),(X,\mathbf{2}^{kD}))\\&= \delta - k\mathrm{Adv}_D((X,B),(X,\mathbf{2}^{kD}))\\&\leq \delta - k\epsilon \leq \delta.\end{aligned}$$

□

To prove Part 1 of Theorem 3.8, we use the Min-Max Theorem to get a universal distinguisher from the assumption that B has low conditional pseudoentropy, and then apply Lemma 3.14.

Theorem 3.15 (KL-hardness ⇒ pseudoentropy, nonuniform setting). *Let (X, B) be a $\{0, 1\}^n \times [q]$-valued random variable, $\delta > 0$. If B is nonuniformly (t, δ) KL-hard given X, then for every $\epsilon > 0$, B has nonuniform (t', ϵ) pseudoentropy at least $\mathrm{H}(B|X) + \delta - \epsilon$ given X for $t' = t^{\Omega(1)}/\mathrm{poly}(n, 1/\epsilon, \log q)$.*

Proof. Suppose for contradiction that B does not have nonuniform (t', ϵ) conditional pseudoentropy at least $\mathrm{H}(B|X)+\delta-\epsilon$. By definition, for any $[q]$-valued random variable C with $\mathrm{H}(C|X) \geq \mathrm{H}(B|X) + \delta - \epsilon$, there is a size t' distinguisher D between (X, B) and (X, C), with $\mathrm{Adv}_D((X, B)(X, C)) > \epsilon$.

Consider the following two player zero-sum game. Player 1 picks a $[q]$-valued random variable C with $\mathrm{H}(C|X) \geq \mathrm{H}(B|X) + \delta - \epsilon$. Player 2 picks a size t' distinguisher D. The payoff for Player 2 is $\mathrm{Adv}_D((X,B)(X,C))$.

Player 1 has no mixed strategy to force Player 2 to achieve payoff at most ϵ, because a convex combination of random variables with conditional entropy at least $\mathrm{H}(B|X) + \delta - \epsilon$ also has conditional entropy at least $\mathrm{H}(B|X) + \delta - \epsilon$. So, by the Min-Max Theorem, Player 2 has a mixed strategy that achieves expected payoff greater than ϵ regardless of Player 1's move. Rephrasing, there is a convex combination D of size t' distinguishers that is a universal distinguisher, in the sense that $\mathrm{Adv}_D((X,B),(X,C)) > \epsilon$ for all C with $\mathrm{H}(C|X) \geq \mathrm{H}(B|X) + \delta - \epsilon$.

By Lemma 3.14, there exists $k \in [0, (\log q)/\epsilon]$ such that $\mathrm{KL}(X,B||X,\mathbf{2}^{kD}) \leq \delta - \epsilon$. In other words, $P(x,a) = 2^{kD(x,a)}$ satisfies $\mathrm{KL}(X,B||X,C_P) \leq \delta - \epsilon$.

Efficiency.

P may not have small circuit size since D is a *distribution* on size t' circuits. Thus we replace D with $\tilde{D}$ which computes the mean of $O\left((n + \log q)/\epsilon^2\right)$ samples of D, such that $\forall x, a, \left|\tilde{D}(x,a) - D(x,a)\right| \leq \epsilon/2$. As $|\mathrm{Adv}_{\tilde{D}}((X,B),(X,C)) - \mathrm{Adv}_D((X,B),(X,C))| \leq \epsilon/2$, $\tilde{D}$ is an efficient universal $\epsilon/2$-distinguisher. By Lemma 3.14 there exists $k \in [0, O(\log q/\epsilon)]$ such that $\mathrm{KL}(X,B||X,\mathbf{2}^{k\tilde{D}}) \leq \delta - \epsilon$. We then approximate k by some rational $\tilde{k}$ to $\Theta(\epsilon/c)$ precision so that $\forall x, a$, $\left|\tilde{k}\tilde{D}(x,a) - k\tilde{D}(x,a)\right| \leq \epsilon/c$, for a sufficiently large constant c. Since $\tilde{k}\tilde{D}$ is rational valued, we can use Newton's method to construct a circuit $\tilde{P}$ approximating $\mathbf{2}^{\tilde{k}\tilde{D}}$. This can be done in such a way that

$$\mathrm{KL}\left(X,B||X,C_{\tilde{P}}\right) \leq \mathrm{KL}(X,B||X,\mathbf{2}^{k\tilde{D}}) + \epsilon \leq \delta$$

and $\tilde{P}$ has size $t = \mathrm{poly}\left(t', n, 1/\epsilon, \log q\right)$. See Lemma A.3 for details. This contradicts the hypothesis that B is nonuniformly (t,δ) KL-hard given X. □

3.2 KL-hardness Implies Pseudoentropy, Uniform Setting

To prove the uniform complexity version of Theorem 3.15, we replace the use of the Min-Max Theorem in the proof of Theorem 3.15 with a Uniform Min-Max Theorem from our forthcoming paper [VZ1]. The Uniform Min-Max Theorem *constructively* builds a near-optimal strategy of the first player in a 2-player game from several best-responses of the first player to strategies of the second player.

KL projection. Let $\mathcal{C}$ be a non-empty closed convex set of Γ-valued random variables and let N be a Γ-valued random variable. One can show that there exists a unique

$$M^* = \arg\min_{M \in \mathcal{C}} \mathrm{KL}(M||N).$$

M^* is called the *KL projection* of N on $\mathcal{C}$.

A nice property of KL projection is the following geometric structure (see [CT], Chap 11, Section 6):

Theorem 3.16 (Pythagorean theorem)**.** *Let $\mathcal{C}$ be a non-empty closed convex set of Γ-valued random variables. Let M^* be the KL projection of N on $\mathcal{C}$. Then for all $M \in \mathcal{C}$,*

$$\mathrm{KL}(M||M^*) + \mathrm{KL}(M^*||N) \leq \mathrm{KL}(M||N).$$

In particular,

$$\mathrm{KL}(M||M^*) \leq \mathrm{KL}(M||N).$$

Assuming $\mathrm{KL}(M^*||N)$ is finite, then Pythagorean theorem implies the KL projection M^* is unique: for any $M \in \mathcal{C}$ which is also a KL projection, the theorem implies $\mathrm{KL}(M||M^*) = 0$, which holds only when $M = M^*$.

Finding the exact KL projection is often computationally imfeasible, so we consider *approximate KL projection.* We say M^* is a *σ-approximate KL projection* of N on $\mathcal{C}$, if $M^* \in \mathcal{C}$ and for all $M \in \mathcal{C}$,

$$\mathrm{KL}(M||M^*) \leq \mathrm{KL}(M||N) + \sigma.$$

In our context, let $\mathcal{C}_r$ denote the set of distributions (X,C) over $\{0,1\}^n \times [q]$ for all C with $\mathrm{H}(C|X) \geq r$. We state here the Uniform Min-Max Theorem specialized to the case where the strategies for Player 2 are distinguishers:

```
Let $C^{(1)} = U_{[q]}$, $c$ a sufficiently large constant
for $i \leftarrow 1$ to $S$ do
    Obtain an arbitrary $D^{(i)}$ with
        $\mathrm{Adv}_{D^{(i)}}((X,B),(X,C^{(i)})) > c\epsilon$
    Weight Update: Let $C^{(i)'}$ be such that
        $C^{(i)'}(a|x) \propto e^{\epsilon \cdot D^{(i)}(x,a)} \cdot C^{(i)}(a|x)$
    $(X, C^{(i+1)}) \leftarrow$ an arbitrary $\sigma$-approximate KL
        projection of $(X, C^{(i)'})$ on $\mathcal{C}_r$
end
Let $D^*$ compute the average of $D^{(1)}, \ldots, D^{(S)}$
```

Algorithm 1: Finding Universal Distinguisher

Theorem 3.17 (Uniform Min-Max Theorem for distinguishers [VZ1])**.** *Consider the two-player zero-sum game where $\mathcal{V} = \mathcal{C}_r$ for some r, W is a set of* randomized boolean circuits*, and for every $(X,C) \in \mathcal{C}_r$, randomized circuit $D \in W$, $f((x,a),D) = \mathbb{E}[D(X,B)] - D(x,a)$ so that $F((X,C),D) = \mathrm{Adv}_D((X,B),(X,C))$. Then for any $0 < \epsilon \leq 1$, $\sigma = \epsilon^2$, after*

$$S = O\left(\max_{(X,C) \in \mathcal{C}_r} \mathrm{KL}(X,C||X,U_{[q]})/\epsilon^2\right) = O\left((\log q - r)/\epsilon^2\right)$$

iterations Algorithm Finding Universal Distinguisher always outputs some D^* such that for all $(X,C) \in \mathcal{C}_r$,

$$\mathrm{Adv}_{D^*}((X,B),(X,C)) = \Omega(\epsilon).$$

The proof of Theorem 3.17 can be found in our technical report [VZ2]. To implement Algorithm Finding Universal Distinguisher, in particular, we need to compute σ-approximate KL projections on the conditional entropy ball $\mathcal{C}_r$.

3.2.1 Approximate KL Projection on the Conditional Entropy Ball

In this section we describe how to efficiently find (X,C) as a σ-approximate KL projection of (X,C') on $\mathcal{C}_r$. We first describe the exact KL projection of random variable (X,C) on a conditional entropy ball $\mathcal{C}_r$, then show how to approximate it.

Recall that for a generalized distinguisher $D : \{0,1\}^n \times [q] \to \mathbb{R}^+$, $k \in \mathbb{R}$, and a $\{0,1\}^n$-valued random variable

X, we define a $[q]$-valued random variable $\mathbf{2}^{kD}$ (jointly distributed with X) as follows:

$$\mathbf{2}^{kD}(a|x) = \frac{2^{kD(x,a)}}{\sum_b 2^{kD(x,b)}}.$$

We begin by showing that $C = \mathbf{2}^{kD}$ ($k \geq 0$) minimizes $\mathrm{Adv}_D((X,B),(X,C))$ among all C with $\mathrm{H}(C|X) \geq \mathrm{H}(\mathbf{2}^{kD}|X)$. As mentioned above, $\mathbf{2}^{kD}$ is a conditional version of the Boltzmann distribution in statistical physics [LL], for which a similar property is well known. While this was our motivation to consider the random variable $\mathbf{2}^{kD}$, we did not explicitly need it for the nonuniform theorem (Theorem 3.15). But why are distinguishers relevant at all, when all we want is to KL-project an arbitrary (X,C) on some entropy ball? The reason is that when viewing C as $\mathbf{2}^D$ for some generalized distinguisher D, Lemma 3.13 says we can minimize KL by maximizing the distinguishing advantage, assuming that the entropy difference is fixed. This will be clear in the proof of Lemma 3.19 below.

Lemma 3.18. *For every C it holds that $\mathbb{E}\left[D(X,\mathbf{2}^{kD})\right] \geq \mathbb{E}\left[D(X,C)\right]$ for all $k \geq 0$ such that $\mathrm{H}(\mathbf{2}^{kD}|X) \leq \mathrm{H}(C|X)$.*

Proof. Consider any C where $\mathrm{H}(C|X) \geq \mathrm{H}(\mathbf{2}^{kD}|X)$. If $k = 0$, then $\mathrm{H}(\mathbf{2}^{kD}|X) = \log q$, so C and $\mathbf{2}^{kD}$ must both be uniform on $[q]$ given X and the result holds vacuously. Thus assume $k > 0$. By Lemma 3.13,

$$\begin{aligned}&\mathrm{H}(\mathbf{2}^{kD}|X) - \mathrm{H}(C|X) - \mathrm{Adv}_{kD}((X,C),(X,\mathbf{2}^{kD}))\\ &\quad = \mathrm{KL}(X,C||X,\mathbf{2}^{kD}) \geq 0,\end{aligned}$$

where we use nonnegativity of KL divergence. Thus,

$$\begin{aligned}&\mathbb{E}\left[D(X,C)\right] - \mathbb{E}\left[D(X,\mathbf{2}^{kD})\right]\\ &\quad = \frac{1}{k}\left(\mathrm{Adv}_{kD}((X,C),(X,\mathbf{2}^{kD}))\right)\\ &\quad \leq \frac{1}{k}\left(\mathrm{H}(\mathbf{2}^{kD}|X) - \mathrm{H}(C|X)\right) \leq 0,\end{aligned}$$

as desired. □

Lemma 3.19 (KL projection on the conditional entropy ball)**.** *Let (X,C) be a $\{0,1\}^n \times [q]$-valued random variable such that $C(a|x) \neq 0$ for all x, a. Let (X,C^*) be the KL projection of (X,C) on $\mathcal{C}_r$, where $0 \leq r < \log q$. Let*

$$D(x,a) = \log \frac{C(a|x)}{\min_b \{C(b|x)\}}.$$

Then $C = \mathbf{2}^D$ and $C^* = \mathbf{2}^{\alpha D}$ for some $\alpha \in (0,1]$ such that $\mathrm{H}(\mathbf{2}^{\alpha D}|X) \geq r$ (in fact $\mathrm{H}(\mathbf{2}^{\alpha D}|X) = r$ whenever $(X,C) \notin \mathcal{C}_r$).

Proof. One can readily verify that D is a generalized distinguisher and $C = \mathbf{2}^D$. Moreover, if $(X,C) \in \mathcal{C}_r$ then the KL projection is $(X,C) = (X,\mathbf{2}^D)$ itself, i.e. $\alpha = 1$.

To find the KL projection for $(X,C) \notin \mathcal{C}_r$, we first note there exists $\alpha \in (0,1)$ such that $\mathrm{H}(\mathbf{2}^{\alpha D}|X) = r$ (by the Intermediate Value Theorem, because $\mathrm{H}(\mathbf{2}^D|X) < r$, $\mathrm{H}(\mathbf{2}^0|X) = \log q \geq r$ and $\mathrm{H}(\mathbf{2}^{kD}|X)$ is continuous as a function of $k \in (0,1)$). By definition of KL projection, we want to minimize $\mathrm{KL}(X,C'||X,\mathbf{2}^D)$ over all C' where $\mathrm{H}(C'|X) = r$ (as KL projection is always on the boundary of $\mathcal{C}_r$; see Lemma A.1). By Lemma 3.13,

$$\begin{aligned}&\mathrm{KL}(X,C'||X,\mathbf{2}^D)\\ &\quad = \mathrm{H}(\mathbf{2}^D|X) - \mathrm{H}(C'|X) - \mathrm{Adv}_D((X,C'),(X,\mathbf{2}^D)).\end{aligned}$$

Hence minimizing $\mathrm{KL}(X,C'||X,\mathbf{2}^D)$ is equivalent to maximizing $\mathrm{Adv}_D((X,C'),(X,\mathbf{2}^D)) = \mathbb{E}\left[D(X,C')\right] - \mathbb{E}\left[D(X,\mathbf{2}^D)\right]$, and the result follows from Lemma 3.18. □

Lemma 3.20 (approximating KL projection on the conditional entropy ball)**.** *There exists a $\mathrm{poly}(\kappa, n, q, 1/\sigma, \log(1/\gamma))$ time algorithm that given oracle access to $D : \{0,1\}^n \times [q] \to [0,\kappa]$ and O_X, given $\sigma > 0$ and $0 \leq r \leq \log q - \sigma$, with probability $1-\gamma$ (over its internal randomness) outputs some $\beta \in (0,1]$ of bit length $\log(\kappa/\sigma) + \log\log q + O(1)$ such that $(X,\mathbf{2}^{\beta D})$ is a σ-approximate KL projection of $(X,\mathbf{2}^D)$ on $\mathcal{C}_r$.*

Proof. We compute an estimate $H_\beta \in [\mathrm{H}(\mathbf{2}^{\beta D}|X) \pm \sigma/6]$ for a discrete β ranging from 0 to 1 in steps of $\sigma/(c\kappa \log q)$ for some large enough constant c, and output $D' = \beta D$ for any β satisfying $H_\beta \in [r + \sigma/6, r + 5\sigma/6]$. This can be done in time $\mathrm{poly}(\kappa, n, q, 1/\sigma, \log(1/\gamma))$ and with success probability $1-\gamma$ after a union bound over all $c\kappa \log q/\sigma$ values of β (see Lemma A.3). If we fail to find such β, then we output $\beta = 1$. We now argue correctness of the algorithm.

If we fail to find such β, then it must be only because $\mathrm{H}(\mathbf{2}^D|X) \geq r$, in other words KL projection of $\left(X,\mathbf{2}^D\right) \in \mathcal{C}_r$ is simply itself. To see that, suppose $\mathrm{H}(\mathbf{2}^D|X) < r$. Since any $\sigma/(c\kappa \log q)$ variation in β causes at most $\sigma/3$ variation in $\mathrm{H}(\mathbf{2}^{\beta D}|X)$ (Lemma A.4), and that $\mathrm{H}(\mathbf{2}^{0D}|X) = \log q \geq r + \sigma$, $\mathrm{H}(\mathbf{2}^{1D}|X) < r$, a discrete Intermediate Value Theorem says there exists a discrete $\beta \in [0,1]$ with $\mathrm{H}(\mathbf{2}^{\beta D}|X) \in [r + \sigma/3, r + 2\sigma/3]$. In other words, we can find such β.

Hence WLOG assume we have found such β. Closeness of H_β to both r and $\mathrm{H}(\mathbf{2}^{\beta D}|X)$ ensures that

$$r \leq \mathrm{H}(\mathbf{2}^{\beta D}|X) \leq r + \sigma.$$

Thus $(X,\mathbf{2}^{\beta D}) \in \mathcal{C}_r$. Recall that the exact KL projection of $(X,\mathbf{2}^D)$ on $\mathcal{C}_r$ is $(X,\mathbf{2}^{\alpha D})$ where $\alpha = 1$ if $(X,\mathbf{2}^D) \in \mathcal{C}_r$, or $0 < \alpha < 1$ and $\mathrm{H}(\mathbf{2}^{\alpha D}) = r$ if $(X,\mathbf{2}^D) \notin \mathcal{C}_r$ (Lemma 3.19). To prove $(X,\mathbf{2}^{\beta D})$ is a σ-approximate KL projection, it suffices to show that for any $(X,C) \in \mathcal{C}_r$,

$$\mathrm{KL}(X,C||X,\mathbf{2}^{\beta D}) - \mathrm{KL}(X,C||X,\mathbf{2}^{\alpha D}) \leq \sigma.$$

Then we are done, as it will follow from Pythagorean Theorem (Theorem 3.16) that

$$\begin{aligned}\mathrm{KL}(X,C||X,\mathbf{2}^{\beta D}) &\leq \mathrm{KL}(X,C||X,\mathbf{2}^{\alpha D}) + \sigma\\ &\leq \mathrm{KL}(X,C||X,\mathbf{2}^D) + \sigma.\end{aligned}$$

By Lemma 3.13,

$$\begin{aligned}&\mathrm{KL}(X,C||X,\mathbf{2}^{\beta D}) - \mathrm{KL}(X,C||X,\mathbf{2}^{\alpha D})\\ &= \mathrm{H}(\mathbf{2}^{\beta D}|X) - \mathrm{H}(\mathbf{2}^{\alpha D}|X)\\ &\quad - \left(\mathrm{Adv}_{\beta D}((X,C),(X,\mathbf{2}^{\beta D})) - \mathrm{Adv}_{\alpha D}((X,C),(X,\mathbf{2}^{\alpha D}))\right)\\ &\leq (r+\sigma) - r\\ &\quad - \left(\mathrm{Adv}_{\beta D}((X,C),(X,\mathbf{2}^{\beta D})) - \mathrm{Adv}_{\alpha D}((X,C),(X,\mathbf{2}^{\alpha D}))\right)\\ &= \sigma + (\alpha - \beta)\,\mathbb{E}\left[D(X,C)\right] + \beta\mathbb{E}[D(X,\mathbf{2}^{\beta D})] - \alpha\mathbb{E}[D(X,\mathbf{2}^{\alpha D})].\end{aligned}$$

Note that $\alpha \geq \beta$, because either $\alpha = 1 \geq \beta$ (when $(X, \mathbf{2}^D) \in \mathcal{C}_r$), or $\mathrm{H}(\mathbf{2}^{\alpha D}|X) = r \leq \mathrm{H}(\mathbf{2}^{\beta D}|X)$ (when $(X, \mathbf{2}^D) \notin \mathcal{C}_r$) and it follows from monotonicity of $\mathrm{H}(\mathbf{2}^{kD}|X)$ as a function of k in $[0, +\infty)$ (Lemma A.2). Thus by Lemma 3.18, $(\alpha - \beta)\,\mathbb{E}\left[D(X,C)\right] \leq (\alpha-\beta)\,\mathbb{E}\left[D(X, \mathbf{2}^{\alpha D})\right]$, and the above inequality becomes

$$
\begin{aligned}
&\mathrm{KL}(X, C||X, \mathbf{2}^{\beta D}) - \mathrm{KL}(X, C||X, \mathbf{2}^{\alpha D}) \\
&\leq \sigma + \beta\left(\mathbb{E}\left[D(X, \mathbf{2}^{\beta D})\right] - \mathbb{E}\left[D(X, \mathbf{2}^{\alpha D})\right]\right) \\
&= \sigma + \beta \cdot \mathrm{Adv}_D((X, \mathbf{2}^{\beta D}), (X, \mathbf{2}^{\alpha D})).
\end{aligned}
$$

Now applying Lemma 3.13 again on $\mathbf{2}^{\alpha D}$ and $\mathbf{2}^{\beta D}$ gives

$$
\begin{aligned}
&\mathrm{Adv}_{\alpha D}((X, \mathbf{2}^{\beta D}), (X, \mathbf{2}^{\alpha D})) \\
&= \mathrm{H}(\mathbf{2}^{\alpha D}) - \mathrm{H}(\mathbf{2}^{\beta D}) - \mathrm{KL}(X, \mathbf{2}^{\beta D}||X, \mathbf{2}^{\alpha D}) \\
&\leq \mathrm{H}(\mathbf{2}^{\alpha D}) - \mathrm{H}(\mathbf{2}^{\beta D}) \leq 0,
\end{aligned}
$$

where we used nonnegativity of KL divergence. Therefore

$$
\mathrm{KL}(X, C||X, \mathbf{2}^{\beta D}) - \mathrm{KL}(X, C||X, \mathbf{2}^{\alpha D}) \leq \sigma.
$$

□

3.2.2 Putting it Together

We now have all the tools ready to prove Theorem 3.11 (KL hardness implies pseudoentropy, uniform setting). We just will replace the use of the Min-Max Theorem in the proof of Theorem 3.15 with the Uniform Min-Max Theorem for distinguishers (Theorem 3.17), using Lemma 3.20 to implement the approximate KL projection. However, notice that $\mathrm{H}(B|X)$ hence the "radius" of the conditional entropy ball $\mathcal{C}_r$ is unknown. We will simply try all radii (with quantization) and pick the distinguisher that results in the best KL predictor, which can be tested by sampling (X, B).

Theorem 3.21 (KL-hardness ⇒ pseudoentropy, uniform setting)**.** *Let n be a security parameter, $\delta = \delta(n) > 0$, $t = t(n) \in \mathbb{N}$, $\epsilon = \epsilon(n) > 0$, $q = q(n)$ all computable in time* poly(n)*. Let (X, B) be a $\{0,1\}^n \times [q]$-valued random variable. If B is uniformly (t, δ) KL-hard given X, then B has uniform (t', ϵ) pseudoentropy at least* $\mathrm{H}(B|X) + \delta - \epsilon$ *given X, for $t' = t^{\Omega(1)}/\mathrm{poly}(n, q, 1/\epsilon)$.*

Proof. Suppose for contradiction that B does not have uniform (t', ϵ) conditional pseudoentropy at least $\mathrm{H}(B|X) + \delta - \epsilon$. By definition, there is a time t' randomized oracle algorithm D such that for infinitely many n and every C with $\mathrm{H}(C|X) \geq \mathrm{H}(B|X) + \delta - \epsilon$, $D^{O_{X,B,C}}$ ϵ-distinguishes (X, B) and (X, C).

Let $\mathcal{C}_r$ denote the entropy ball $\{(X, C) : \mathrm{H}(C|X) \geq r\}$. Let $\gamma > 0$ be an error parameter to be fixed later. Assume that given any $r \geq \mathrm{H}(B|X) + \delta - \epsilon/2$, we can implement Algorithm Finding Universal Distinguisher on $\mathcal{C} = \mathcal{C}_r$ using oracle $O_{X,B}$, to output a circuit D^* of size $\mathrm{poly}(t', n, \log q, 1/\epsilon, \log(1/\gamma))$ w.p. at least $1-\gamma$, in time $\mathrm{poly}(t', n, q, 1/\epsilon, \log(1/\gamma))$. We show how to do in the end.

Let c be a large enough constant. We show that the following time t oracle algorithm P violates the hypothesis that B is uniformly (t, δ) KL-hard given X:

```
INPUT: $(x, a) \in \{0,1\}^n \times [q]$
ORACLE: $O_{X,B}$
for $r \leftarrow 0$ to $\log q$ in steps of $\epsilon/c$ do
    $D^* \leftarrow$ Run Algorithm Finding Universal
        Distinguisher on $\mathcal{C} = \mathcal{C}_r$ using oracle $O_{X,B}$
    for $k \leftarrow 0$ to $(\log q)/\epsilon$ in steps of $\epsilon/c$ do
        Add the generalized distinguisher $kD^*$ to list $L$
    end
end
for each generalized distinguisher $D' \in L$ do
    Estimate $\mathrm{KL}(X, B||X, \mathbf{2}^{D'}) + \mathrm{H}(B|X)$ within $\epsilon/c$
        error using oracle $O_{X,B}$
end
Let $\tilde{D} \in L$ have the lowest estimate
$P^{O_{X,B}}(x, a)$ outputs an approximation of $\mathbf{2}^{\tilde{D}}(a|x)$
```

Algorithm 2: A predictor P violating the KL-hardness of B given X

To prove correctness, first we claim w.p. at least $1 - \gamma$ there exists a generalized distinguisher $kD^* \in L$ satisfying

$$
\mathrm{KL}(X, B||X, \mathbf{2}^{kD^*}) \leq \delta - \epsilon/3 + \epsilon/c.
$$

Consider an iteration where $r \in [\mathrm{H}(B|X) + \delta - \epsilon/2, \mathrm{H}(B|X) + \delta - \epsilon/3]$. Recall that we assume Algorithm Finding Universal Distinguisher can be implemented on $\mathcal{C} = \mathcal{C}_r$ to output a circuit D^* of size $\mathrm{poly}(t', n, \log q, 1/\epsilon, \log(1/\gamma))$ w.p. at least $1-\gamma$. Theorem 3.17 says that $\mathrm{Adv}_{D^*}((X, B), (X, C)) = \Omega(\epsilon)$ for all C with

$$
\mathrm{H}(C|X) \geq \mathrm{H}(B|X) + \delta - \epsilon/3 \geq r.
$$

Lemma 3.14 says there exists $k^* \in [0, (\log q)/\epsilon]$ such that $\mathrm{KL}(X, B||X, \mathbf{2}^{k^* D^*}) \leq \delta - \epsilon/3$. Thus in any inner iteration where $k \in [k^* - \epsilon/c, k^*]$, it follows from Lemma A.4 that

$$
\begin{aligned}
\mathrm{KL}(X, B||X, \mathbf{2}^{kD^*}) &\leq \mathrm{KL}(X, B||X, \mathbf{2}^{k^* D^*}) + \epsilon/c \\
&\leq \delta - \epsilon/3 + \epsilon/c.
\end{aligned}
$$

It turns out that by sampling, for each $D' \in L$ we can estimate $\mathrm{KL}(X, B||X, \mathbf{2}^{D'}) + \mathrm{H}(B|X)$ within ϵ/c error w.p. at least $1 - \gamma/|L|$, in time $\mathrm{poly}\left(t', n, 1/\epsilon, q, \log(1/\gamma)\right)$ (See Lemma A.3). Thus, w.p. at least $1 - \gamma$ the generalized distinguisher $\tilde{D} : \{0,1\} \times [q] \rightarrow [0, (\log q)/\epsilon]$ in L with the least estimate satisfies

$$
\mathrm{KL}(X, B||X, \mathbf{2}^{\tilde{D}}) \leq \mathrm{KL}(X, B||X, \mathbf{2}^{kD^*}) + 2\epsilon/c.
$$

Finally, approximating $\mathbf{2}^{\tilde{D}}$ can be done using Newton's method to produce a predictor $p : \{0,1\}^n \times [q] \rightarrow [1, q^{1/\epsilon}]$ such that w.p. at least $1 - 3\gamma$ the random variable C_p satisfies

$$
\begin{aligned}
\mathrm{KL}\left(X, B||X, C_p\right) &\leq \mathrm{KL}(X, B||X, \mathbf{2}^{\tilde{D}}) + \epsilon/c \\
&\leq \delta - \epsilon/3 + 4\epsilon/c \leq \delta - \epsilon/4,
\end{aligned}
$$

and the total running time is $t = \mathrm{poly}\left(t', n, 1/\epsilon, \log q, \log(1/\gamma)\right)$. See Lemma A.3 for details.

We view $P^{O_{X,B}}$ as a distribution over functions $p : \{0,1\}^n \times [q] \rightarrow [1, q^{1/\epsilon}]$, and it remains show that

$$
\mathop{\mathbb{E}}_{p\,\sim P^{O_{X,B}}}\left[\mathrm{KL}(X, B||X, C_p)\right] \leq \delta.
$$

By the earlier analysis we know that $\mathrm{KL}\left(X, B||X, C_p\right) < (\delta - \epsilon/4)$ with probability at least $1 - 3\gamma$ over $p \sim P^{O_{X,B}}$, and for every $p : \{0,1\}^n \times [q] \to [1, q^{1/\epsilon}]$,

$$\begin{aligned}\mathrm{KL}(X, B||X, C_p) &= \mathbb{E}\left[\sum_a B(a|X) \log\left(B(a|X)/C_p(a|X)\right)\right] \\ &\le \max_{x,a} \log\left(1/C_p(a|x)\right) = O(\log q + 1/\epsilon).\end{aligned}$$

Thus

$$\begin{aligned}&\mathop{\mathbb{E}}_{p \sim P^{O_{X,B}}} [\mathrm{KL}(X, B||X, C_p)] \\ &\le (1 - 3\gamma) \cdot (\delta - \epsilon/4) + (3\gamma) \cdot O(\log q + 1/\epsilon) \le \delta\end{aligned}$$

for an appropriate choice of $\gamma = \Omega(\epsilon/(\log q + 1/\epsilon))$, as desired.

Implementing Finding Universal Distinguisher.
Given any $r \ge \mathrm{H}(B|X) + \delta - \epsilon/2$ and oracle access to $O_{X,B}$, we show how to implement each of the $S = O((\log q)/\epsilon^2)$ iterations of Algorithm Finding Universal Distinguisher on $\mathcal{C} = \mathcal{C}_r$ efficiently and output a size $\mathrm{poly}(t', n, \log q, 1/\epsilon, \log(1/\gamma))$ circuit D^* w.p. at least $1-\gamma$, in time $\mathrm{poly}(t', n, q, 1/\epsilon, \log(1/\gamma))$.

Let $\gamma' > 0$ be an error parameter to be fixed later. For each iteration $j \in [S]$, we will implement $C^{(j)}$ in Algorithm Finding Universal Distinguisher by constructing a generalized distinguisher D_j as a circuit of size $\mathrm{poly}(t', n, \log q, 1/\epsilon, \log(1/\gamma'))$ such that $C^{(j)} = \mathbf{2}^{D_j}$. We do this for $j = 1$ by setting $D_1 = 0$. Assuming we have constructed D_j, we can construct D_{j+1} in time $\mathrm{poly}(t', n, q, 1/\epsilon)$ as follows:

1. *We can obtain a size* $t'' = \mathrm{poly}(t', n, \log q, 1/\epsilon, \log(1/\gamma'))$ *distinguisher* $D^{(j)}$ *from* D_j *such that*

$$\mathrm{Adv}_{D^{(j)}}((X, B), (X, C^{(j)})) > \epsilon' = c\epsilon$$

in time $\mathrm{poly}(t', n, q, 1/\epsilon)$ *w.p. at least* $1 - 2\gamma'$, where c is the constant in Algorithm Finding Universal Distinguisher.

By using Newton's method to approximate $\mathbf{2}^{D_j}$, we can construct a circuit $\tilde{P}$ such that the random variable $\tilde{C}(a|x) = \tilde{P}(x, a)/\sum_b \tilde{P}(x, b)$ satisfies (i) $\mathrm{H}(\tilde{C}|X) \ge \mathrm{H}(C^{(j)}|X) - \epsilon/2$; (ii) For any distinguisher D', $\mathrm{Adv}_{D'}((X, B), (X, C^{(j)})) \ge \mathrm{Adv}_{D'}((X, B), (X, \tilde{C})) - \epsilon/3$. This can be done in time $\mathrm{poly}(t', n, \log q, 1/\epsilon, \log(1/\gamma'))$ w.p. at least $1 - \gamma'$ (See Lemma A.3).

We then generate $m = O((\log(1/\gamma') + n + \log q)/\epsilon^2)$ random samples of $(X, B, \tilde{C})^{t'}$ and $U_{t'}$, where $\tilde{C}$ is samplable from X in time $\mathrm{poly}(t', n, q, 1/\epsilon, \log(1/\gamma'))$. Finally let $D^{(j)}$ be the distinguisher that given (x, a), chooses $I \in_R [m]$ and outputs $D^{O_{X,B,\tilde{C}}}(x, a)$ using the Ith copy of $(X, B, \tilde{C})^{t'}$ to answer oracle queries and the Ith copy of $U_{t'}$ as the internal randomness of D. Note that the size of $D^{(j)}$ does not depend on the size of D_j (but the size of D_{j+1} will additively depend on the size of $D^{(j)}$).

By a Chernoff bound and union bound, w.p. at least $1 - \gamma'$ for every (x, a) we have

$$\left|D^{(j)}(x, a) - D^{O_{X,B,\tilde{C}}}(x, a)\right| \le \epsilon/3.$$

Thus,

$$\begin{aligned}&\left|\mathrm{Adv}_{D^{(j)}}((X, B), (X, \tilde{C})) - \mathrm{Adv}_{D^{O_{X,B,\tilde{C}}}}((X, B), (X, \tilde{C}))\right| \\ &\le \epsilon/3.\end{aligned}$$

Since

$$\begin{aligned}\mathrm{H}(\tilde{C}|X) &\ge \mathrm{H}(C^{(j)}|X) - \epsilon/4 \\ &\ge (\mathrm{H}(B|X) + \delta - \epsilon/2) - \epsilon/2 = \mathrm{H}(B|X) + \delta - \epsilon\end{aligned}$$

the conditional pseudoentropy of B guarantees that

$$\begin{aligned}&\mathrm{Adv}_{D^{(j)}}((X, B), (X, C^{(j)})) \\ &\ge \mathrm{Adv}_{D^{(j)}}((X, B), (X, \tilde{C})) - \epsilon/3 \\ &\ge \mathrm{Adv}_{D^{O_{X,B,\tilde{C}}}}((X, B), (X, \tilde{C})) - \epsilon/3 - \epsilon/3 \\ &> \epsilon - 2\epsilon/3 = \epsilon'.\end{aligned}$$

2. $C^{(j+1)'} = \mathbf{2}^{D_j + (\log e)\epsilon \cdot D^{(j)}}$. This is just the consequence of multiplicative weight update.

3. *We can obtain a size* $\mathrm{poly}(t', n, \log q, 1/\epsilon, \log(1/\gamma'))$ *generalized distinguisher* D_{j+1} *such that* $(X, \mathbf{2}^{D_{j+1}})$ *is an* $O(\epsilon^2)$*-approximate KL projection of* $(X, C^{(j+1)'})$ *on* $\mathcal{C}_r$ *w.p. at least* $1 - \gamma'$*, in time* $\mathrm{poly}(t', n, q, 1/\epsilon, \log(1/\gamma'))$. Indeed, using Lemma 3.20 we can obtain an $O(\epsilon^2)$-approximate KL projection $(X, \mathbf{2}^{D_{j+1}})$ where $D_{j+1} = \beta_{j+1} \cdot (D_j + \epsilon'/2 \cdot D^{(j)})$ for some $\beta_{j+1} \in (0, 1]$. Notice that $D_j + \epsilon'/2 \cdot D^{(j)}$ is a $[0, O(S\epsilon)]$-valued generalized distinguisher as each iteration increases the range of D_j by at most $O(\epsilon)$. Thus by Lemma 3.20, β_{j+1} is of bit length $\log(S/\epsilon) + \log\log q + O(1)$. Consequently, D_{j+1} is of size $\mathrm{poly}(t', n, \log q, 1/\epsilon)$, as each iteration increases size of D_j by at most $t'' + \mathrm{poly}(\log(S/\epsilon), \log\log q)$.

Let $\gamma' = \gamma/c'S$ for a large enough constant c'. By induction, w.p. at least $1 - O(S\gamma') = 1 - \gamma$, we can construct the required D_j for every j and every iteration is correctly implemented in time $\mathrm{poly}(t', n, q, 1/\epsilon, \log(1/\gamma))$. Furthermore, D^* which computes the average of $D^{(j)}$ has circuit size $O(S \cdot t'') = \mathrm{poly}(t', n, \log q, 1/\epsilon, \log(1/\gamma))$. □

3.3 Pseudoentropy Implies KL-hardness

Theorem 3.22 (pseudoentropy ⇒ KL-hardness, nonuniform and uniform settings)**.** *Let n be a security parameter, $\delta = \delta(n) > 0$, $t = t(n) \in \mathbb{N}$, $\epsilon = \epsilon(n) > 0$, $q = q(n)$, $\sigma = \sigma(n)$ all computable in time* $\mathrm{poly}(\mathrm{n})$*. Let (X, B) be a $\{0,1\}^n \times [q]$-valued random variable. If B has (non)uniform (t, ϵ) pseudoentropy at least* $\mathrm{H}(B|X) + \delta$ *given X, then B is (non)uniformly (t', δ') KL-hard given X, for $t' = \min\{t^{\Omega(1)}/\mathrm{poly}(n, \log(1/\sigma)), \Omega(\sigma/\epsilon)\}$ and $\delta' = \delta - \sigma$.*

In the uniform case, this implication holds even for a weaker definition of conditional pseudoentropy where we only require indistinguishability against distinguishers with oracle access to $O_{X,B}$.

Proof. We shall prove the nonuniform version. Once so it will be clear that the uniform version follows.

Suppose for contradiction that B is not nonuniformly $(t', \delta - \sigma)$ KL-hard. Then there is a $(\delta - \sigma)$-KL predictor $P : \{0,1\}^n \times [q] \to [1, +\infty)$ as a circuit of size t'. We show that there is a size $\mathrm{poly}(t', \log(1/\sigma)) \le t$ universal distinguisher D such that $\mathrm{Adv}_D((X, B), (X, C)) > \epsilon$ for every C with $\mathrm{H}(C|X) \ge \mathrm{H}(B|X) + \delta$. Specifically, we show that if P is a λ-KL predictor of size t', then there is a size $\mathrm{poly}(t', \log(1/\sigma))$ distinguisher D such that $\mathrm{Adv}_D((X, B), (X, C)) \ge (\delta - \lambda - \sigma/2)/2t'$ for every C with $\mathrm{H}(C|X) \ge \mathrm{H}(B|X) + \delta$.

We claim the following distinguisher D is a desired universal distinguisher:

$$D(x,a) = \frac{1}{2t'}\left(\log P(x,a) + t'\right).$$

Note that D is a distinguisher i.e. $D(x,a) \in [0,1]$, because $2^{-t'} \leq P(x,a) \leq 2^{t'}$. Moreover, one can verify that $\mathbf{2}^{2t'D} = C_P$.

Now consider any C with $\mathrm{H}(C|X) \geq \mathrm{H}(B|X) + \delta$. Applying Lemma 3.13 twice, we obtain

$$\begin{aligned}&\mathrm{H}(\mathbf{2}^{2t'D}|X) - \mathrm{H}(B|X) - \mathrm{Adv}_{2t'D}((X,B),(X,\mathbf{2}^{2t'D}))\\ &\quad = \mathrm{KL}(X,B||X,\mathbf{2}^{2t'D}) \leq \lambda,\end{aligned}$$

where the inequality by definition of λ-KL predictor, as well as

$$\begin{aligned}&\mathrm{H}(\mathbf{2}^{2t'D}|X) - \mathrm{H}(C|X) - \mathrm{Adv}_{2t'D}((X,C),(X,\mathbf{2}^{2t'D}))\\ &\quad = \mathrm{KL}(X,C||X,\mathbf{2}^{2t'D}) \geq 0,\end{aligned}$$

where the inequality is by nonnegativity of KL divergence. Taking the difference yields

$$\begin{aligned}&\mathrm{Adv}_D((X,B),(X,C))\\ &= \mathrm{Adv}_D((X,B),(X,\mathbf{2}^{2t'D})) - \mathrm{Adv}_D((X,C),(X,\mathbf{2}^{2t'D}))\\ &= \frac{\mathrm{Adv}_{2t'D}((X,B),(X,\mathbf{2}^{2t'D})) - \mathrm{Adv}_{2t'D}((X,C),(X,\mathbf{2}^{2t'D}))}{2t'}\\ &\geq \frac{\mathrm{H}(C|X) - \mathrm{H}(B|X) - \lambda}{2t'}\\ &\geq \frac{\delta - \lambda}{2t'}.\end{aligned}$$

Efficiency.

We approximate D by $\tilde{D}$, where $\log P(x,a)$ is computed to precision $\sigma/2$. Since $P(x,a)$ is represented as a rational p_1/p_2 where $p_1, p_2 \leq 2^{t'}$, the logarithm can be approximated to that precision in time $\mathrm{poly}(t', \log(1/\sigma))$ using Taylor series. Thus D has circuit size $\mathrm{poly}(t', \log(1/\sigma)) \leq t$. Moreover, for any C with $\mathrm{H}(C|X) \geq \mathrm{H}(B|X) + \delta$, we have

$$\begin{aligned}\mathrm{Adv}_{\tilde{D}}((X,B),(X,C)) &\geq \mathrm{Adv}_D((X,B),(X,C)) - \frac{1}{2t'}\cdot\frac{\sigma}{2}\\ &\geq \frac{\delta - \lambda - \sigma/2}{2t'}.\end{aligned}$$

This completes the proof for the nonuniform case.

At this point, the uniform version also follows quite naturally: Given P such that when $P^{O_{X,B}}$ is viewed as a distribution over functions $p : \{0,1\}^n \times [q] \to (0,+\infty)$,

$$\mathop{\mathbb{E}}_{p\sim P^{O_{X,B}}}[\mathrm{KL}(X,B||X,C_p)] \leq \delta - \sigma.$$

We let D be the randomized oracle algorithm such that $D^{O_{X,B}}$ performs the above conversion from a λ-KL predictor to a universal $(\delta - \lambda - \sigma/2)/2t'$-distinguisher, replacing the $P(x,a)$ there with the output of simulating $P^{O_{X,B}}$ on (x,a) (using random coin tosses and $O_{X,B}$). Thus for every C with $\mathrm{H}(C|X) \geq \mathrm{H}(B|X) + \delta$,

$$\begin{aligned}&\mathbb{E}\left[\mathrm{Adv}_{D^{O_{X,B}}}((X,B),(X,C))\right]\\ &\geq \mathop{\mathbb{E}}_{p\sim P^{O_{X,B}}}\left[\frac{\delta - \mathrm{KL}(X,B||X,C_p) - \sigma/2}{2t'}\right]\\ &\geq \frac{\sigma}{4t'} > \epsilon.\end{aligned}$$

Furthermore, D runs in time $\mathrm{poly}(n, t', \log(1/\sigma)) \leq t$. □

Since Theorem 3.22 only requires a weaker version of conditional pseudoentropy, we obtain the following equivalence:

Corollary 3.23. *Let n be a security parameter, $\delta = \delta(n) > 0$, $q = q(n)$ computable in time* $\mathrm{poly}(n)$*. Let (X,B) be a $\{0,1\}^n \times [q]$-valued random variable that is polynomial-time samplable. Then the following are equivalent:*

1. *B is uniformly δ KL-hard given X;*
2. *B has uniform pseudoentropy at least* $\mathrm{H}(B|X) + \delta$ *given X;*
3. *B has "weak" uniform pseudoentropy at least* $\mathrm{H}(B|X) +$ *δ given X: For every probabilistic polynomial time algorithm A and every constant c, there is a random variable C jointly distributed with X, B such that the following holds for all sufficiently large n:*
   - $\mathrm{H}(C|X) \geq \mathrm{H}(B|X) + \delta - 1/n^c$*;*
   - *(X,B) and (X,C) are indistinguishable by A:*

$$|\Pr[A(X,B) = 1] - \Pr[A(X,C) = 1]| \leq \frac{1}{n^c}.$$

Proof. 1 ⇒ 2 by Theorem 3.21. 2 ⇒ 3 by definition. 3 ⇒ 1 by Theorem 3.22 and the fact that (X,B) is polynomial-time samplable. □

4. FROM ONE-WAY FUNCTIONS TO NEXT-BIT PSEUDOENTROPY

In this section, we show how to obtain a next-bit pseudoentropy generator from an arbitrary one-way function f. One-way functions are functions easy to compute but hard to invert:

Definition 4.1 (one-way functions)**.** $f : \{0,1\}^* \to \{0,1\}^*$ is *a* (T, γ) *one-way function* for $T = T(n)$, $\gamma = \gamma(n)$ if f is computable in polynomial time, and for every time T randomized algorithm A, for all sufficiently large n, it holds that $\Pr_{y\sim f(U_n)}[f(A(y)) = y] < \gamma$. We say f is one-way if f is $(n^c, 1/n^c)$ one-way for every constant c.

This section is structured as follows. Given a one-way function f, we first show that U_n is KL-hard for sampling given $f(U_n)$. By a chain rule for KL-hardness, we then argue it is KL-hard to sample the next bit of U_n given $f(U_n)$ and all previous bits of U_n. Finally, we use the equivalences between KL-hardness for sampling, KL-hardness, and conditional pseudoentropy (for small q) to derive that $(f(U_n), U_n)$ has a lot of total next-bit pseudoentropy.

Lemma 4.2 (KL-hardness from one-way functions)**.** *Let n be a security parameter, and $f : \{0,1\}^n \to \{0,1\}^n$ be (t,γ) one-way, for $t = t(n)$, $\gamma = \gamma(n)$. Then U_n is uniformly $(t', \log(1/\gamma))$ KL-hard for sampling given $f(U_n)$, for $t' = t/\mathrm{poly}(n)$.*

Proof. Suppose for contradiction that U_n is not uniformly $(t', \log(1/\gamma))$ KL-hard for sampling given $f(U_n)$, i.e. there exists a time t' randomized oracle algorithm S such that

$$\mathrm{KL}\left(f(U_n), U_n || f(U_n), S^{O_{f(U_n),U_n}}(f(U_n))\right) \leq \log\frac{1}{\gamma}.$$

Let $g(y, x)$ be the indicator function that $f(x) = y$. Since applying a (deterministic) function does not increase KL divergence (Lemma 2.5),

$$\mathrm{KL}\left(g\left(f(U_n), U_n\right) || g\left(f(U_n), S^{O_{f(U_n),U_n}}(f(U_n))\right)\right) \leq \log \frac{1}{\gamma}$$

where $g(f(U_n), U_n) \equiv 1$, and $g\left((f(U_n), S^{O_{f(U_n),U_n}}(f(U_n))\right)$ equals 1 w.p. $p = \Pr[S^{O_{f(U_n),U_n}}(f(U_n)) = U_n]$. Since the KL divergence from Bernoulli(1) to Bernoulli(p) is $\log(1/p)$, we must have $p \geq \gamma$. That is,

$$\Pr[S^{O_{f(U_n),U_n}}(f(U_n)) = U_n] \geq \gamma.$$

Since $O_{f(U_n),U_n}$ can be simulated in time poly(n), this violates the fact that f is (t, γ) one-way for $t = t' \cdot \mathrm{poly}(n)$. □

Lemma 4.3 (chain rule for KL-hardness)**.** *Let Y be a distribution over $\{0, 1\}^n$, jointly distributed with Z. If Y is uniformly (t, δ) KL-hard for sampling given Z, then Y_I is uniformly $(t', \delta/n)$ KL-hard for sampling given $(Z, Y_1, \ldots, Y_{I-1})$, for $I \in_R [n]$, $t' = t/O(n)$.*

Proof. Suppose Y_I is not uniformly $(t', \delta/n)$ KL-hard for sampling given $(Z, Y_1, \ldots, Y_{I-1})$, that is there exists a time t' randomized oracle algorithm S such that

$$\mathrm{KL}\left(Z, Y_1, \ldots, Y_I || Z, Y_1, \ldots, Y_{I-1}, S^{O_{Z,Y_1,\ldots,Y_I}}(Z, Y_1, \ldots, Y_{I-1})\right)$$
$$\leq \frac{\delta}{n}.$$

Consider the time $O(nt') = t$ algorithm that samples $W_1, \ldots, W_n$ from Z using oracle $O_{Z,Y}$, where W_i is inductively defined to be $S^{O_{Z,Y_1,\ldots,Y_I}}(Z, W_1, \ldots, W_{i-1})$. By the chain rule for KL divergence (Fact 2.4),

$$\mathrm{KL}(Z, Y_1, \ldots, Y_j || Z, W_1, \ldots, W_j)$$
$$- \mathrm{KL}(Z, Y_1, \ldots, Y_{j-1} || Z, W_1, \ldots, W_{j-1})$$
$$= \mathrm{KL}((Y_j | Z, Y_1, \ldots, Y_{j-1}) || (W_j | Z, W_1, \ldots, W_{j-1}))$$
$$= \mathrm{KL}(Z, Y_1, \ldots, Y_j || Z, Y_1, \ldots, Y_{j-1}, S^{O_{Z,Y_1,\ldots,Y_I}}(Z, Y_1, \ldots, Y_{j-1}))$$

where the last equality follows from definition of conditional KL divergence. Telescoping over $j = 1, \ldots, n$,

$$\mathrm{KL}(Z, Y || Z, W_1, \ldots, W_n)$$
$$= \sum_{i=1}^{n} \mathrm{KL}$$
$$(Z, Y_1, \ldots, Y_i || Z, Y_1, \ldots, Y_{i-1}, S^{O_{Z,Y_1,\ldots,Y_I}}(Z, Y_1, \ldots, Y_{i-1}))$$
$$= n \cdot \mathrm{KL}$$
$$(Z, Y_1, \ldots, Y_I || Z, Y_1, \ldots, Y_{I-1}, S^{O_{Z,Y_1,\ldots,Y_I}}(Z, Y_1, \ldots, Y_{I-1}))$$
$$\leq n \cdot \frac{\delta}{n} = \delta.$$

This violates Y being uniformly (t, δ) KL-hard for sampling given Z. □

Now the remainder of showing next-bit pseudoentropy of $(f(U_n), U_n)$ follows from (i) KL-hard for sampling implies KL-hard; (ii) KL-hard implis conditional pseudoentropy (for small q). Formally,

Theorem 4.4 (next-bit pseudoentropy from one-way functions)**.** *Let n be a security parameter, $t = t(n)$, $\gamma = \gamma(n)$, $\epsilon = \epsilon(n)$ all computable in polynomial time. Let $f : \{0, 1\}^n \to \{0, 1\}^n$ be (t, γ) one-way. Then $(f(U_n), U_n)$ has (t', ϵ) uniform next-bit pseudoentropy at least $n + \log(1/\gamma) - \epsilon$, for $t' = t^{\Omega(1)}/\mathrm{poly}(n, 1/\epsilon)$.*

Proof. Let $Z = f(U_n)$, $Y = U_n$ and $I \in_R [n]$. By Lemma 4.2 and 4.3, Y_I is uniformly $(t/\mathrm{poly}(n), \log(1/\gamma)/n)$ KL-hard for sampling given $(Z, Y_1, \ldots, Y_{I-1})$. By Lemma 3.7, Y_I is uniformly $(t/\mathrm{poly}(n), \log(1/\gamma)/n)$ KL-hard given $(Z, Y_1, \ldots, Y_{I-1})$. By Theorem 3.21, Y_I has (t', ϵ) uniform conditional pseudoentropy at least $\mathrm{H}(Y_I | Z, Y_1, \ldots, Y_{I-1}) + \log(1/\gamma)/n - \epsilon/n$, for $t' = t^{\Omega(1)}/\mathrm{poly}(n, 1/\epsilon)$. Equivalently, (Z, Y) has (t', ϵ) uniform next-bit pseudoentropy at least $\mathrm{H}(Y, Z) + \log(1/\gamma) - \epsilon = n + \log(1/\gamma) - \epsilon$. □

Remark 4.5. The argument in this section says $(f(U_n), U_n)$ has a lot of next-bit pseudoentropy as long as U_n is KL-hard to sample from $f(U_n)$. The KL-hardness of sampling U_n from $f(U_n)$ is similar to the notion of a *distributional one-way function* [IL] which amounts to replacing KL divergence with statistical distance.

For U_n to be KL-hard to sample from $f(U_n)$, it is not necessary that f is one-way. For example, given any one-way function $h : \{0, 1\}^n \to \{0, 1\}^{n/2}$, define

$$f(x) = \begin{cases} x_{1,\ldots,n/2} & (x_{n/2+1,\ldots,n} = 0^{n/2}) \\ h(x) & (\text{otherwise}) \end{cases}.$$

Clearly f is not one-way, but U_n is still KL-hard to sample from $f(U_n)$. Thus, our construction of next-bit pseudoentropy generators (and later on, pseudorandom generators) can be based on a larger class of functions.

5. FROM NEXT-BIT PSEUDOENTROPY TO PSEUDORANDOMNESS

In this section, for brevity, we always assume the uniform setting whenever referring to one-way functions and computational notions of (conditional) entropy. Nonetheless, these results hold in the nonuniform setting too, with little or no change in the argument.

5.1 The Construction

Haitner et al. show a construction of a pseudorandom generator from any next-bit pseudoentropy generator G_{nb}. Their result can be stated as follows:

Theorem 5.1 (pseudorandomness from next-bit pseudoentropy [HRV])**.** *Let n be a security parameter. Let $\Delta = \Delta(n) \in [1/poly(n), n]$, $m = m(n)$, $\kappa = \kappa(n) \in [n/2]$ be polynomial time computable. For every polynomial time computable $G_{nb} : \{0, 1\}^n \to \{0, 1\}^m$ such that $G_{nb}(U_n)$ has (T, ϵ) next-bit pseudoentropy at least $n + \Delta$, there exists a $(T - n^{O(1)}, n^{O(1)} \cdot (\epsilon + 2^{-\kappa}))$ pseudorandom generator $G : \{0, 1\}^d \to \{0, 1\}^{d \cdot (1+\Omega(\Delta/n))}$ with seed length*

$$d = O\left(\frac{m^2 n^2 \kappa \log^2 n}{\Delta^3}\right).$$

Moreover, G is computable in $\mathbf{NC}^1$ with $O(d/n)$ (uniformly random) oracle calls to G_{nb}.

By Theorem 4.4, we can simply use $U_n \to (f(U_n), U_n)$ as the next-bit pseudoentropy generator, and obtain the following construction of PRG G from one-way functions f (illustrated in Figure 1), by applying the construction in Theorem 5.1:

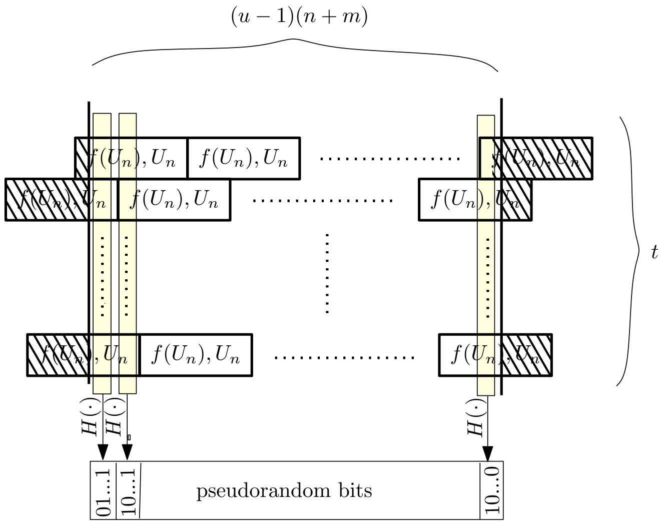

Figure 1: Simplified construction of PRG from one-way function $f : \{0,1\}^n \to \{0,1\}^m$. Each row contains $u = \Theta(n/\log n)$ iid copies of $(f(U_n), U_n)$, shifted by a random offset $I \in [n+m]$. To extract pseudorandom bits, an arbitrary universal hash function H (with a proper output length) is applied to all $t = \Theta(d/(u \cdot (n+m)))$ bits in the same column.

Construction 1. Given input U_d, the pseudorandom generator outputs

$$h, h(G_1^1 G_1^2 \dots G_1^t), h(G_2^1 G_2^2 \dots G_2^t), \dots$$

where h is a universal hash function, and for each $1 \le i \le t$, G^i consists of $u = \Theta(n/\Delta)$ iid copies of $(f(U_n), U_n)$, with the first I bits of the first copy and the last $m+n-I$ bits of the last copy discarded, for $I \in_R [n+m]$ (using a new copy of I for each G^i). We let $t = \Theta(d/(u \cdot n))$.

If $f : \{0,1\}^n \to \{0,1\}^n$ is one-way, then setting parameters $m = n$, $\Delta = \log n$ and $\kappa = \omega(\log n)$, G is a PRG with seed length any $d = \omega(n^4)$ and stretch $d \cdot \Omega((\log n)/n)$.

The following corollary was pointed out to us by Yuval Ishai: If f is a one-way function with exponential security and linear circuit size, by using universal hash functions that have linear circuit size as constructed in [IKOS], we can obtain a PRG whose circuit complexity is linear in its stretch. Such pseudorandom generators (with circuit complexity linear in their stretch) are useful for cryptography with constant computational overhead [IKOS].

Corollary 5.2 (pseudorandom generators with constant overhead)**.** *Suppose that there is a function $f : \{0,1\}^n \to \{0,1\}^n$ computable by uniform circuits of size $O(n)$ and such that for some constant $\alpha > 0$ and every constant c, f is $(n^c, 2^{-\alpha n})$ one-way. Then there exists a pseudorandom generator $G : \{0,1\}^d \to \{0,1\}^{2d}$ computable by uniform boolean circuits of size $O(d)$, for $d = O(n \cdot \mathrm{polylog}(n))$.*

Proof. By Theorem 4.4, $G_{nb}(U_n)$ has uniform next-bit pseudoentropy at least $(1+\alpha)n$. By Theorem 5.1, there exists a pseudorandom generator $G : \{0,1\}^d \to \{0,1\}^{d\cdot(1+\alpha)}$ with seed length $d = O(n \log^3 n)$. We see from the construction (Construction 1) that G (i) performs $O(d/n)$ evaluations of f, for a total circuit size of $O(d)$ since f has $O(n)$ circuit size; (ii) applies hashing on all $\Theta(n/\alpha)$ columns and a total of $O(d)$ bits, for a total circuit size of $O(d)$ using universal hash functions computable by uniform circuits of linear size [IKOS]. Thus G has circuit size $O(d)$. We then do iterative composition [Gol] $\lceil 1/\alpha \rceil$ times to increase the output length to $2d$; this increases the circuit size by a constant factor. □

This result does not follow from the [HRV] construction alone, since their next-bit pseudoentropy generator requires hash functions that support "local list-decoding" and are not known to be implementable in linear size.

5.2 Saving Seed Length

In this section, we show how to save the seed length of [HRV]'s construction of pseudorandom generators from next-bit pseudoentropy generators, by a factor of $\Theta(n)$.

There are three steps in the construction:

1. Entropy equalization — discarding the first I bits of the first copy and the last $m-I$ bits of the last copy of G_{nb}. Since G_{nb} is highly unstructured, nothing can be said about the conditional pseudoentropy in any fixed bit, yet by discarding a random prefix, each position is now a random bit in G_{nb}. By taking many copies of G_{nb}, the amortized loss of next-bit pseudoentropy is small.

   Lemma 5.3. [HRV] *Let n be a security parameter, $m = m(n) = poly(n)$ and $\ell = \ell(n) = poly(n)$ be* poly(n) *time computable integer functions, where $\ell(n) > 1$. Let X be random variable over $\{0,1\}^m$ with (T,ϵ)-next-bit pseudoentropy at least k, for $T = T(n)$, $\epsilon = \epsilon(n)$ and $k = k(n)$. Let J be uniformly distributed over $[m]$ and let $\tilde{X} = X_J^{(1)}, \dots, X_m^{(1)}, \dots, X_1^{(\ell)}, \dots, X_{J-1}^{(\ell)}$, where $X^{(i)}$'s are iid copies of X. Then every bit of $\tilde{X}$ has $(T - O(\ell \cdot m), \ell \cdot \epsilon)$ conditional pseudoentropy at least $(\ell-1)k/(\ell m)$, conditioned on previous bits of $\tilde{X}$ and J.* [6]

2. Converting conditional Shannon entropy to conditional min-entropy — taking multiple (parallel) copies. This generalizes the standard procedure of converting Shannon entropy to min-entropy by taking sufficiently many copies. Conditional pseudo-min-entropy is defined analogously to conditional pseudoentropy; see [HRV].

   Lemma 5.4. [HRV] *Let n be a security parameter, $m = m(n) = poly(n)$ and $t = t(n) = poly(n)$ be* poly(n) *time computable integer functions. Let X be a random variable over $\{0,1\}^m$ where every bit of X has (T,ϵ) conditional pseudoentropy at least α, for $T = T(n)$, $\epsilon = \epsilon(n)$, $\alpha = \alpha(n)$. Then for every $\kappa = \kappa(n) > 0$ it holds that every block of $(X_1^{(1)}, X_1^{(2)}, \dots), \dots, (X_m^{(1)}, X_m^{(2)}, \dots)$ conditioned on previous blocks, has (T', ϵ') conditional pseudo*-min-*entropy α', where $X^{(i)}$'s are iid copies of X, and*

   - $T' = T'(n) = T - O(m \cdot t)$,
   - $\epsilon' = \epsilon'(n) = t^2 \cdot (\epsilon + 2^{-\kappa} + 2^{-ct})$ *for a universal constant $c > 0$, and*
   - $\alpha' = \alpha'(n) = t \cdot \alpha - \Gamma(t,\kappa)$, *for $\Gamma(t,k) \in O(\sqrt{t \cdot \kappa} \cdot \log t)$.*

[6] This is slightly stronger than the version in [HRV], which does not condition on J. However, it is easy to see from their proof that one can additionally condition on J.

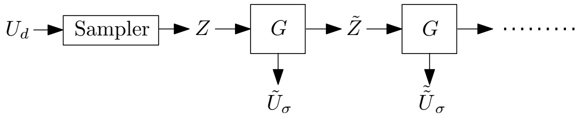

Figure 2: Iterative composition for Z-seeded PRG G

3. Randomness extraction. This step is essentially a computational version of block source extraction. At the previous step, the amount of next-bit pseudo-min-entropy in each block is known. So we may choose hash functions of fixed output length to make the output pseudorandom.

   Lemma 5.5. [HRV] *Let n be a security parameter, $m = m(n) = poly(n)$, $t = t(n) = poly(n)$, $\alpha = \alpha(n) \in [t(n)]$ and $\kappa = \kappa(n) \in [\alpha(n)]$ be* poly(n) *time computable integer functions. Let $\{h_s : \{0,1\}^t \to \{0,1\}^{\alpha-\kappa}\}$ be some family of universal hash functions. Let $X_1, \ldots, X_m$ be random variables over $\{0,1\}^t$ such that every X_i conditioned on $X_1, \ldots, X_{i-1}$ has (T, ϵ) conditional pseudo-min-entropy α, for $T = T(n)$ and $\epsilon = \epsilon(n)$. Then $(h, h(X_1), \ldots, h(X_m))$ is $(T - m \cdot t^{O(1)}, m \cdot (\epsilon + 2^{-\kappa/2}))$ pseudorandom, where h is a random hash function from the family.*

We refer to [HRV] for the proofs and detailed explanation of intuition behind these steps.

The seed length blow up in [HRV] comes from Step 1 (Entropy Equalization) and Step 2 (Converting to conditional min-entropy), as each involves repeating the current generator on many independent seeds. We show how to save the blow up due to Entropy Equalization, by showing how randomness from a "few" copies of G_{nb} can be used to generate more copies of G_{nb}, and iteratively.

Specifically, we show that the [HRV] construction above, but taking only $\ell = 2$ copies in Entropy Equalization, gives rise to a "Z-seeded" PRG, one that given input distribution Z outputs some $(\tilde{Z}, \tilde{U}_\sigma)$ indistinguishable from (Z, U_σ). (If Z were uniformly distributed in $\{0,1\}^d$, this would be a standard PRG.) Then we apply iterative composition (just like iterative composition for standard PRGs [Gol]) to increase the number of pseudorandom bits (without changing the seed distribution Z).

We begin by describing the iterative composition of Z-seeded PRGs, illustrated in Figure 2.

Lemma 5.6 (iterative composition of Z-seeded PRGs)**.** *Let n be a security parameter. Let $\sigma = \sigma(n), \ell = \ell(n) =$* poly$(n)$ *be* poly(n) *time computable functions. Let $Z = Z(n)$ be a distribution samplable in* poly(n) *time using $d = d(n)$ bits of randomness. Let G be a generator computable in* poly(n) *time such that $G(Z) = (\tilde{Z}, \tilde{U}_\sigma)$ is (T, ϵ)-indistinguishable from (Z, U_σ), for $T = T(n), \epsilon = \epsilon(n)$. Then there is a $(T -$* poly$(n), \ell\epsilon)$ *pseudorandom generator $G' : \{0,1\}^d \to \{0,1\}^{\ell\sigma}$ computable in* poly(n) *time.*

Proof. Consider the following algorithm $G_\ell(z)$: If $\ell = 0$ then output ϵ (the empty string). If $\ell \geq 0$ then let $(\tilde{z}, \tilde{u}) = G(z)$ and output $G_{\ell-1}(\tilde{z}) \circ \tilde{u}$.

We claim that $G_\ell(Z)$ is pseudorandom, so we obtain the desired PRG G' by composing G_ℓ with algorithm that samples Z given d random bits. Clearly G' runs in poly(n) time. We show the pseudorandomness of $G_\ell(Z)$ by a hybrid argument.

Suppose for contradiction that $G_\ell(Z)$ is not $(T', \ell\epsilon)$-pseudorandom, i.e. there exists a T' time $\ell\epsilon$-distinguisher D between $G_\ell(Z)$ and $U_{\ell\sigma}$. For each $0 \leq i \leq \ell$ define a hybrid distribution $H_i = (G_i(Z), U_{(\ell-i)\sigma})$. Thus $H_0 = U_{\ell\sigma}$ and $H_\ell = G_\ell(Z)$. Let $I \in_R [\ell]$. Then

$$\begin{aligned}\mathbb{E}\left[D(H_I) - D(H_{I-1})\right] &= \frac{1}{\ell}\sum_{k=1}^{\ell} \mathbb{E}\left[D(H_k) - D(H_{k-1})\right] \\ &= \frac{1}{\ell}\mathbb{E}[D(G_\ell(Z)) - D(U_{\ell\sigma})] > \epsilon.\end{aligned}$$

We use this to break the pseudorandomness property of G. Denote $G(Z) = (\tilde{Z}, \tilde{U}_\sigma)$. We claim that $D'(z, u) = D(G_{I-1}(z) \circ u \circ U_{(\ell-I)\sigma})$, where $I \in_R [\ell]$ and $|u| = \sigma$, ϵ'-distinguishes (Z, U_σ) from $(\tilde{Z}, \tilde{U}_\sigma)$. Notice that given $(\tilde{z}, \tilde{u}) = G(z)$, we have $(G_{I-1}(\tilde{z}), \tilde{u}) = G_I(z)$ by definition of G_ℓ. Thus, $D'(\tilde{Z}, \tilde{U}_\sigma) = D(G_I(Z) \circ U_{(\ell-I)\sigma}) = D(H_I)$ whereas $D'(Z, U_\sigma) = D(G_{I-1}(Z) \circ U_\sigma \circ U_{(\ell-I)\sigma}) = D(H_{I-1})$. It follows that

$$\mathbb{E}[D'(Z, U_\sigma) - D'(\tilde{Z}, \tilde{U}_\sigma)] = \mathbb{E}[D(H_I) - D(H_{I-1})] > \epsilon.$$

Moreover, D' is computable in $T' +$ poly(n) time. For an appropriate $T' = T -$ poly(n), this contradicts that (Z, U_σ) and $(\tilde{Z}, \tilde{U}_\sigma)$ are (T, ϵ) indistinguishable. Therefore, $G_\ell(Z)$ is $(T -$ poly$(n), \ell\epsilon)$-pseudorandom. □

We now show how to construct a Z-seeded PRG G from any next-bit pseudoentropy generator G_{nb}, as demonstrated in Figure 3. By applying iterative composition, this gives rise to a seed-efficient construction of PRG from a pseudoentropy generator G_{nb} which should be compared to the original construction illustrated in Figure 1.

Theorem 5.7 (Z-seeded PRGs from next-bit pseudoentropy)**.** *Let n be a security parameter. Let $\Delta = \Delta(n) \in [1/\text{poly}(n), n]$, $m = m(n)$, $\kappa = \kappa(n) \in [n/2]$ be polynomial-time computable functions. For every polynomial-time computable $G_{nb} : \{0,1\}^n \to \{0,1\}^m$ such that $G_{nb}(U_n)$ has (T, ϵ) next-bit pseudoentropy at least $n + \Delta$ (for $T = T(n)$ and $\epsilon = \epsilon(n)$), there exists distribution $Z = Z(n)$ and generator G such that:*

1. *Z is samplable in polynomial time using*

   $$d = O\left(\frac{m^2 n\kappa \log^2\left(\frac{n\kappa}{\Delta}\right)}{\Delta^2}\right)$$

   bits of randomness;

2. *G is computable in polynomial time and $G(Z)$ is $(T - n^{O(1)}, n^{O(1)} \cdot (\epsilon + 2^{-\kappa}))$ indistinguishable from (Z, U), U being uniformly random string of length $\Omega(d \cdot \Delta/n)$.*

Moreover, G is computable with $O(d/n)$ (uniform and independent) oracle calls to G_{nb}.

Proof. Let t be a parameter to be fixed later. Let $J^{(1)}, \ldots, J^{(t)}$ be t iid copies of $J \in_R [m]$, and $H \in_R \{0,1\}^t$. Consider

$$\begin{aligned}Z =& (H \circ J^{(1)} \ldots J^{(t)} \\ &\circ G_{nb}(U^{(1)})_{1,\ldots,J^{(1)}-1} \ldots G_{nb}(U^{(t)})_{1,\ldots,J^{(t)}-1} \\ &\circ G_{nb}(U^{(t+1)}) \ldots G_{nb}(U^{(2t)}))\end{aligned}$$

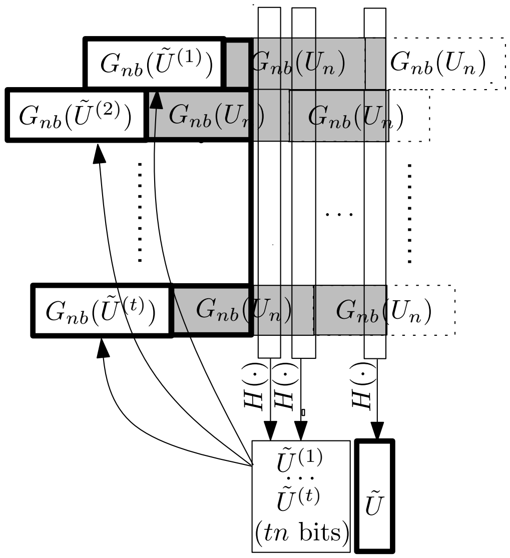

Figure 3: Construction of Z-seeded PRG G from any next-bit pseudoentropy generator G_{nb}
The shaded area represents input Z. The bold boxes are the output $G(Z) = (\tilde{Z}, \tilde{U})$.
The ith row is shifted by a random offset $J^{(i)} \in [n+m]$. An arbitrary universal hash function H (with a proper output length) is then applied to all bits in the same column, producing pseudorandom bits $(\tilde{U}^{(1)}, \dots, \tilde{U}^{(t)}, \tilde{U})$ where each $\tilde{U}^{(i)}$ is of length n. We then apply G_{nb} to each $\tilde{U}^{(i)}$. Together with unused bits of Z they form $\tilde{Z}$.
We ignore H, $J^{(1)}$, …, $J^{(t)}$ in the figure since they are the same in the input and output of G.

where $U^{(i)}$'s are iid copies of U_n. Z is clearly samplable in polynomial time using $d = t + t \cdot (\log m + 2n) = O(tn)$ bits of randomness.

We now define G. Interpret G's input as $h \circ j^{(1)} \dots j^{(t)} \circ G_{nb}(u^{(1)})_{1,\dots,j^{(1)}-1} \dots G_{nb}(u^{(t)})_{1,\dots,j^{(t)}-1} \circ G_{nb}(u^{(t+1)}) \dots G_{nb}(u^{(2t)})$, where $h, j^{(i)}, u^{(i)}$ are strings of length t, $\log m$ and n respectively. G is defined as follows:

1. For each $i \in [t]$ (that is, for each "row"), we set $y^{(i)} = \left(G_{nb}(u^{(t+i)})_{j^{(i)},\dots,m} \circ G_{nb}(u^{(i)})_{1,\dots,j^{(i)}-1}\right)$ (Entropy Equalization);

2. Apply a universal hash function $h : \{0,1\}^t \to \{0,1\}^{t'}$ where t' will be chosen later so that $t'm > tn$, on $y_j^{(1)} \circ \dots \circ y_j^{(t)}$, for each $j \in [m]$ (that is, for each "column"). Thus m calls to h produce $t'm$ bits in total:

$$\begin{aligned}&\tilde{u}^{(1)} \dots \tilde{u}^{(t)} \circ \tilde{u}\\ &= h(y_1^{(1)}, \dots, y_1^{(t)}) \circ h(y_2^{(1)}, \dots, y_2^{(t)}) \dots h(y_m^{(1)}, \dots, y_m^{(t)})\end{aligned}$$

   where $\tilde{u}^{(1)}, \dots, \tilde{u}^{(t)}$ are n-bit strings, and $\tilde{u}$ is the remaining $t'm - tn$ bits.

3. Output

$$\begin{aligned}&h \circ j^{(1)} \dots j^{(t)}\\ &\quad \circ G_{nb}(u^{(t+1)})_{1,\dots,j^{(1)}-1} \dots G_{nb}(u^{(2t)})_{1,\dots,j^{(t)}-1}\\ &\quad \circ G_{nb}(\tilde{u}^{(1)}) \dots G_{nb}(\tilde{u}^{(t)}) \circ \tilde{u}.\end{aligned}$$

We now prove that $G(Z)$ is computationally indistinguishable from $(Z \circ U)$ where $U = U_{t'm-tn}$ (i.e. a $t'm - tn$ bit random string). Suppose we run $G(Z)$ to obtain

$$G(Z) = \left(H \circ W \circ G_{nb}(\tilde{U}^{(1)}) \dots G_{nb}(\tilde{U}^{(t)}) \circ \tilde{U}\right)$$

where $\tilde{U}$ is of length $t'm - tn$, and

$$\begin{aligned}&W =\\ &\left(J^{(1)} \dots J^{(t)} \circ G_{nb}(U^{(t+1)})_{1,\dots,J^{(1)}-1} \dots G_{nb}(U^{(2t)})_{1,\dots,J^{(t)}-1}\right).\end{aligned}$$

In the following, we will show that $G(Z) = (H \circ W \circ G_{nb}(\tilde{U}^{(1)}) \dots G_{nb}(\tilde{U}^{(t)}) \circ \tilde{U})$ is computationally indistinguishable from $(Z \circ U) = \left(H \circ W \circ G_{nb}^{(1)} \dots G_{nb}^{(t)} \circ U\right)$, where $G_{nb}^{(i)}$'s are iid copies of $G_{nb}(U_n)$. The proof is essentially the same 3-step analysis as in Haitner et. al, with the tweak that the conditional pseudoentropy and conditional pseudo-min-entropy are now additionally conditioned on W, and the final indistinguishablility holds for W taking any value.

In Step 1, we set

$$Y^{(i)} = G_{nb}(U^{(t+i)})_{J^{(i)},\dots,m} \circ G_{nb}(U^{(i)})_{1,\dots,J^{(i)}-1}.$$

Recall that $G_{nb}(U_n)$ has (T, ϵ) next-bit pseudoentropy at least $n + \Delta$. Applying Lemma 5.3 (Entropy Equalization) with $\ell = 2$, $X^{(1)} = G_{nb}(U^{(t+i)})$ and $X^{(2)} = G_{nb}(U^{(i)})$, we obtain that every bit of $Y^{(i)}$ conditioned on previous bits of $Y^{(i)}$, $G_{nb}(U^{(t+i)})_{1,\dots,J^{(i)}-1}$ and $J^{(i)}$, has $(T - O(m), 2\epsilon)$ conditional pseudoentropy at least $(\Delta + n)/m$.

Recall that $Y^{(1)}, \dots, Y^{(t)}$ are t independent rows. By Lemma 5.4 (t-fold parallel repetition), $Y_j^{(1)}, \dots, Y_j^{(t)}$ has $(T - O(mt), t^2 \cdot (2\epsilon + 2^{-\kappa} + 2^{-ct}))$ conditional pseudo-min-entropy at least $\alpha = t(\Delta + n)/m - O(\sqrt{t\kappa} \log t)$, conditioned on W and all $Y_k^{(1)}, \dots, Y_k^{(t)}$ where $k < j$.

In Step 2, we apply hashing to each "column". By Lemma 5.5, if we set $t' = \alpha - 2\kappa$, then $(H \circ \tilde{U}^{(1)} \dots \tilde{U}^{(t)} \circ \tilde{U})$ and $(H \circ U_{tn} \circ U)$ are $(T - O(mt) - mt^{O(1)}, mt^2 \cdot (2\epsilon + 2^{-\kappa} + 2^{-\Omega(t)}) + m \cdot 2^{-\kappa})$ indistinguishable, for W taking any value. Thus the same can be said about $(H \circ G_{nb}(\tilde{U}^{(1)}) \dots G_{nb}(\tilde{U}^{(t)}) \circ \tilde{U})$ and $(H \circ G_{nb}^{(1)} \dots G_{nb}^{(t)} \circ U)$. Thus we conclude that

$$G(Z) = \left(H \circ W \circ G_{nb}(\tilde{U}^{(1)}) \dots G_{nb}(\tilde{U}^{(t)}) \circ \tilde{U}\right)$$

is $(T - O(mt) - mt^{O(1)}, mt^2 \cdot (2\epsilon + 2^{-\kappa} + 2^{-\Omega(t)}) + m \cdot 2^{-\kappa})$ indistinguishable from

$$\left(H \circ W \circ G_{nb}^{(1)} \dots G_{nb}^{(t)} \circ U\right) = (Z \circ U).$$

We are left to set the parameters. We need to guarantee

$$\begin{aligned}\Omega\left(\frac{\Delta}{n} d\right) &\leq t'm - tn\\ &= \left(\frac{t(\Delta + n)}{m} - O(\sqrt{t\kappa} \log t) - 2\kappa\right) m - tn\end{aligned}$$

where $d = O(tn)$. Assuming $\kappa \le O(t)$, this can be simplified to

$$\frac{\sqrt{t}}{\log t} \ge O\left(\frac{m\sqrt{\kappa}}{\Delta}\right)$$

which is guaranteed for an appropriate choice of

$$t = O\left(\frac{m^2\kappa\log^2\left(\frac{m\kappa}{\Delta}\right)}{\Delta^2}\right),$$

and consequently

$$d = O(tn) = O\left(\frac{m^2n\kappa\log^2\left(\frac{m\kappa}{\Delta}\right)}{\Delta^2}\right) = O\left(\frac{m^2n\kappa\log^2\left(\frac{n\kappa}{\Delta}\right)}{\Delta^2}\right).$$

So (Z,U) and $G(Z)$ are $(T - O(ts) - mt^{O(1)}, mt^2 \cdot (2\epsilon + 2^{-\kappa} + 2^{-\Omega(t)}) + m2^{-\kappa}) = (T - n^{O(1)}, n^{O(1)} \cdot (\epsilon + 2^{-\kappa}))$ indistinguishable. Moreover, G makes $O(d/n)$ uniformly random oracle calls to G_{nb}. □

Combining Lemma 5.6 and Theorem 5.7, we obtain a seed length efficient construction of pseudorandom generators:

Corollary 5.8 (pseudorandomness from next-bit pseudoentropy)**.** *Let n be a security parameter. Let $\Delta = \Delta(n) \in [1/\mathrm{poly}(n), n]$, $m = m(n)$, $\kappa = \kappa(n) \in [n/2]$, $\ell = \ell(n) = \mathrm{poly}(n)$ be computable in time $\mathrm{poly}(n)$. For every polynomial time computable $G_{nb} : \{0,1\}^n \to \{0,1\}^m$ such that $G_{nb}(U_n)$ has (T,ϵ) next-bit pseudoentropy at least $n + \Delta$ (for $T = T(n)$ and $\epsilon = \epsilon(n)$), there exists a polynomial-time computable $(T - n^{O(1)}, n^{O(1)} \cdot (\epsilon + 2^{-\kappa}))$ pseudorandom generator $G : \{0,1\}^d \to \{0,1\}^{d\cdot(\ell\Delta/n)}$ with seed length*

$$d = O\left(\frac{m^2n\kappa\log^2\left(\frac{n\kappa}{\Delta}\right)}{\Delta^2}\right).$$

Moreover, G is computable with $O(\ell d/n)$ (uniformly random) oracle calls to G_{nb}.

Proof. By Theorem 5.7, there is a Z-seeded PRG G' where Z is samplable in polynomial time from U_d, and $G'(Z)$ is $(T - n^{O(1)}, n^{O(1)} \cdot (\epsilon + 2^{-\kappa}))$ indistinguishable from (Z,U). By Lemma 5.6 there exists a pseudorandom generator G with the above parameters. □

In particular, from a one-way function $f : \{0,1\}^n \to \{0,1\}^n$ and setting $m = n$, $\Delta = \log n$, $\kappa = \omega(\log n)$, $\ell = 2n/\Delta$ we can construct a pseudorandom generator of seed length any $d = \omega(n^3 \log n)$. Like [HRV], the construction obtains $\Theta(\log n)$ bits of additive stretch per invocation of the one-way function, which is optimal by [GGKT].

Acknowledgments

We thank Kai-Min Chung, Oded Goldreich, Iftach Haitner, Thomas Holenstein, Yuval Ishai, Bruce Kapron, Omer Reingold, Jonathan Ullman, and Udi Wieder for helpful discussions.

6. REFERENCES

[BHK] Boaz Barak, Moritz Hardt, and Satyen Kale. The uniform hardcore lemma via approximate bregman projections. In *SODA '09: Proceedings of the Nineteenth Annual ACM -SIAM Symposium on Discrete Algorithms*, pages 1193–1200, Philadelphia, PA, USA, 2009. Society for Industrial and Applied Mathematics.

[BM] Manuel Blum and Silvio Micali. How to generate cryptographically strong sequences of pseudo random bits. pages 112–117, 1982.

[BSW] Boaz Barak, Ronen Shaltiel, and Avi Wigderson. Computational analogues of entropy. In *RANDOM-APPROX*, pages 200–215, 2003.

[CT] Thomas M. Cover and Joy A. Thomas. *Elements of information theory (2. ed.)*. Wiley, 2006.

[DORS] Yevgeniy Dodis, Rafail Ostrovsky, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM J. Comput.*, 38(1):97–139, 2008.

[FR] Benjamin Fuller and Leonid Reyzin. Computational entropy and information leakage. 2011. (available at http://www.cs.bu.edu/fac/reyzin).

[GGKT] Rosario Gennaro, Yael Gertner, Jonathan Katz, and Luca Trevisan. Bounds on the efficiency of generic cryptographic constructions. *SIAM Journal on Computing*, 35(1):217–246, 2005.

[GL] Oded Goldreich and Leonid A. Levin. A hard-core predicate for all one-way functions. In *Proceedings of the Twenty First Annual ACM Symposium on Theory of Computing*, pages 25–32, Seattle, Washington, 15–17 May 1989.

[GM1] Oded Goldreich and Bernd Meyer. Computational indistinguishability: algorithms vs. circuits. *Theoretical Computer Science*, 191(1-2):215–218, 1998.

[GM2] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, April 1984.

[Gol] Oded Goldreich. *Computational Complexity: A Conceptual Perspective*. 2006.

[GV] Oded Goldreich and Salil Vadhan. Comparing entropies in statistical zero knowledge with applications to the structure of szk. In *In Proceedings of the Fourteenth Annual IEEE Conference on Computational Complexity*, pages 54–73. IEEE Computer Society Press, 1998.

[HHR$^+$] Iftach Haitner, Thomas Holenstein, Omer Reingold, Salil P. Vadhan, and Hoeteck Wee. Universal one-way hash functions via inaccessible entropy. In *EUROCRYPT*, pages 616–637, 2010.

[HILL] 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 (electronic), 1999.

[HLR] Chun-Yuan Hsiao, Chi-Jen Lu, and Leonid Reyzin. Conditional computational entropy, or toward separating pseudoentropy from compressibility. In *EUROCRYPT*, pages 169–186, 2007.

[Hol1] Thomas Holenstein. Key agreement from weak bit agreement. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing (STOC)*, pages 664–673, 2005.

[Hol2] Thomas Holenstein. Pseudorandom generators from one-way functions: A simple construction for any hardness. In *TCC*, pages 443–461, 2006.

[HRV] 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.

[HRVW] 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.

[IKOS] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Cryptography with constant computational overhead. In *STOC*, pages 433–442, 2008.

[IL] Russell Impagliazzo and Michael Luby. One-way functions are essential for complexity based cryptography. In *FOCS*, pages 230–235, 1989.

[Imp] Russell Impagliazzo. Hard-core distributions for somewhat hard problems. In *36th Annual Symposium on Foundations of Computer Science*, pages 538–545, Milwaukee, Wisconsin, 23–25 October 1995. IEEE.

[KMR+] Michael J. Kearns, Yishay Mansour, Dana Ron, Ronitt Rubinfeld, Robert E. Schapire, and Linda Sellie. On the learnability of discrete distributions. In *STOC*, pages 273–282, 1994.

[KS] Adam R. Klivans and Rocco A. Servedio. Boosting and hard-core sets. In *In Proceedings of the Fortieth Annual Symposium on Foundations of Computer Science*, pages 624–633, 1999.

[LL] L.D. Landau and E.M. Lifshitz. *Statistical physics*, volume 5 of *Statistical Physics*. Oxford: Pergamon Press, 1980.

[Nao] Moni Naor. Evaluation may be easier than generation. In *STOC*, pages 74–83, 1996.

[Rey] Leonid Reyzin. Some notions of entropy for cryptography. In *ICITS*, pages 138–142, 2011.

[Sha] Claude Shannon. Communication theory of secrecy systems. *Bell System Technical Journal*, 28(4):656–715, 1949.

[STV] Madhu Sudan, Luca Trevisan, and Salil Vadhan. Pseudorandom generators without the XOR lemma. *Journal of Computer and System Sciences*, 62:236–266, 2001.

[VZ1] Salil Vadhan and Colin Jia Zheng. A uniform minmax theorem and its applications. In preparation, 2012.

[VZ2] Salil P. Vadhan and Colin Jia Zheng. Characterizing pseudoentropy and simplifying pseudorandom generator constructions. *Electronic Colloquium on Computational Complexity (ECCC)*, 18:141, 2011.

[Yao1] Andrew C. Yao. Theory and applications of trapdoor functions. pages 80–91, 1982.

[Yao2] Andrew C. Yao. Theory and applications of trapdoor functions (extended abstract). In *23rd Annual Symposium on Foundations of Computer Science*, pages 80–91, Chicago, Illinois, 3–5 November 1982. IEEE.

APPENDIX

A. INFORMATION-THEORETIC FACTS AND APPROXIMATION

Proofs for these facts can be found in the technical report [VZ2].

Lemma A.1. *The KL projection of any C on any convex set $\mathcal{C} \not\ni C$ is in the boundary of $\mathcal{C}$.*

Lemma A.2. *For every generalized distinguisher D, $\mathrm{H}(\mathbf{2}^{kD}|X)$ is monotone decreasing in k for $k \in [0, +\infty)$.*

Lemma A.3 (approximation lemma)**.**

1. *There is a $\mathrm{poly}(t, n, \log q, 1/\sigma, \log(1/\gamma))$ time algorithm $\tilde{P} : \{0,1\}^n \times [q] \to [1, 2^{\tilde{\kappa}}]$ that, given a generalized distinguisher $\tilde{D} : \{0,1\}^n \times [q] \to [0, \tilde{\kappa}]$ as a circuit of size t, $\sigma > 0$, $\gamma > 0$, with probability $1 - \gamma$ (over its internal randomness) the following holds: for the random variable*
$$\tilde{C}(a|x) = \frac{\tilde{P}(x,a)}{\sum_b \tilde{P}(x,b)},$$
for any generalized distinguisher D where $\forall x, a, |D(x,a) - \tilde{D}(x,a)| \le \sigma$, and any $D' : \{0,1\}^n \to [0, \kappa]$:
$$\left|\mathbb{E}\left[D'(X, \tilde{C})\right] - \mathbb{E}\left[D'(X, \mathbf{2}^D)\right]\right| = \kappa \cdot O(\sigma),$$
$$\left|\mathrm{KL}(X, B||X, \tilde{C}) - \mathrm{KL}(X, B||X, \mathbf{2}^D)\right| = O(\sigma),$$
$$\left|\mathrm{H}(\tilde{C}|X) - \mathrm{H}(\mathbf{2}^D|X)\right| = \left(\mathrm{H}(\mathbf{2}^D|X) + 1\right) \cdot O(\sigma).$$
2. *There is a $\mathrm{poly}(t, n, q, 1/\epsilon, \log(1/\gamma))$ time algorithm that given a generalized distinguisher $D : \{0,1\}^n \times [q] \to \mathbb{R}^+$ as a circuit of size t, $\epsilon > 0$, $\gamma > 0$, with probability $1-\gamma$ (over its internal randomness) estimates $\mathrm{H}(\mathbf{2}^D|X)$ within $O(\epsilon)$ additive error.*
3. *There is a $\mathrm{poly}(\kappa, t, n, q, 1/\epsilon, \log(1/\gamma))$ time oracle algorithm that given a generalized distinguisher $D : \{0,1\}^n \times [q] \to [0, \kappa]$ as a circuit of size t, $\epsilon > 0$, $\gamma > 0$, for any $\{0,1\}^n \times [q]$-valued random variable (X, B), with probability $1 - \gamma$ (over its internal randomness) estimates $\mathrm{Adv}_D((X, B), (X, \mathbf{2}^D))$ and $\mathrm{KL}(X, B||X, \mathbf{2}^D) + \mathrm{H}(B|X)$ within $O(\epsilon)$ additive error using oracle $O_{X,B}$.*

Lemma A.4. *For any generalized distinguishers D_1, D_2, any $\{0,1\}^n \times [q]$-valued random variable (X, B),*
$$\left|\mathrm{H}(\mathbf{2}^{D_1}|X) - \mathrm{H}(\mathbf{2}^{D_2}|X)\right|$$
$$= \left(\mathrm{H}(\mathbf{2}^{D_2}|X) + 1\right) \cdot O\left(\max_{x,a} |D_1(x,a) - D_2(x,a)|\right),$$
$$\left|\mathrm{KL}(X, B||X, \mathbf{2}^{D_1}) - \mathrm{KL}(X, B||X, \mathbf{2}^{D_2})\right|$$
$$= O\left(\max_{x,a} |D_1(x,a) - D_2(x,a)|\right).$$