Conditional Computational Entropy, or Toward Separating Pseudoentropy from Compressibility

Chun-Yuan Hsiao[1], Chi-Jen Lu[2], and Leonid Reyzin[1]

[1] Boston University, Boston, MA 02215, USA
{cyhsiao,reyzin}@cs.bu.edu
Work performed in part while visiting the
Institute for Pure and Applied Mathematics at UCLA
[2] Academia Sinica, 128 Academia Road, Section 2, Nankang, Taipei 115, Taiwan
cjlu@iis.sinica.edu.tw

Abstract. We study conditional computational entropy: the amount of randomness a distribution appears to have to a computationally bounded observer who is given some correlated information. By considering conditional versions of HILL entropy (based on indistinguishability from truly random distributions) and Yao entropy (based on incompressibility), we obtain:

- a separation between conditional HILL and Yao entropies (which can be viewed as a separation between the traditional HILL and Yao entropies in the shared random string model, improving on Wee's 2004 separation in the random oracle model);
- the first demonstration of a distribution from which extraction techniques based on Yao entropy produce more pseudorandom bits than appears possible by the traditional HILL-entropy-based techniques;
- a new, natural notion of unpredictability entropy, which implies conditional Yao entropy and thus allows for known extraction and hard-core bit results to be stated and used more generally.

1 Introduction

The various information-theoretic definitions of entropy measure the amount of randomness a probability distribution has. As cryptography is able to produce distributions that appear, for computationally bounded observers, to have more randomness than they really do, various notions of *computational* entropy attempt to quantify this *appearance* of entropy. The commonly used HILL entropy (so named after [HILL99]) says that a distribution has computational entropy k if it is indistinguishable (in polynomial time) from a distribution that has information-theoretic entropy k.[3] The so-called Yao entropy [Yao82,BSW03], says that a distribution has computational entropy k if it cannot be efficiently compressed to below k bits and then efficiently decompressed. Other computational notions of entropy have been considered as well [BSW03,HILL99].

[3] The specific notion of information-theoretic entropy depends on the desired application; for the purposes of this paper, we will use min-entropy, defined in Section 2.

Computational notions of entropy are useful, in particular, for extracting strings that are pseudorandom (i.e., look uniform to computationally bounded observers) from distributions that appear to have entropy. Indeed, generation of pseudorandom bits is the very purpose of computational entropy defined in [HILL99], and its variant considered in [GKR04]. Pseudorandom bits have many uses, for example, as keys in cryptographic applications.

The adversary in cryptographic applications (or, more generally, an observer) often possesses information related to the distribution whose entropy is being measured. For example, in the case of Diffie-Hellman key agreement [DH76] the adversary has g^x and g^y, and the interesting question is the amount of computational entropy of g^{xy}. Thus, the entropy of a distribution for a particular observer (and thus the pseudorandomness of the extracted strings) depends on what other information the observer possesses. Because notions of computational entropy necessarily refer to computationally-bounded machines (e.g., the distinguisher for the HILL entropy or the compressor and decompressor for the Yao entropy), they must also consider the information available to these machines. This has sometimes been done implicitly (e.g., in [GKR04]); however, most commonly used definitions do not do so explicitly.

In this work, we explicitly put forward notions of *conditional* computational entropy. This allows us to:

1. Separate conditional Yao entropy from conditional HILL entropy by demonstrating a joint distribution (X, Z) such that X has high Yao entropy but low HILL entropy when conditioned on Z.
2. Demonstrate (to the best of our knowledge, first) application of Yao entropy by extracting more pseudorandom bits from a distribution using Yao-entropy-based techniques than seems possible from HILL-entropy-based techniques.
3. Define a new, natural notion of unpredictability entropy, which can be used, in particular, to talk about the entropy of a value that is unique, such as g^{xy} where g^x and g^y are known to the observer, and possibly even verifiable, such as the preimage x of a one-way permutation f, where $y = f(x)$ is known to the observer.

HILL-Yao Separation. The first contribution (Section 3) can be seen as making progress toward the open question of whether Yao entropy implies HILL entropy, attributed in [TVZ05] to Impagliazzo [Imp99] (the converse is known to be true: HILL entropy implies Yao entropy, because compressibility implies distinguishability). Wee [Wee04] showed that Yao entropy does not imply HILL entropy in the presence of a random oracle and a membership testing oracle. Our separation of conditional Yao entropy from conditional HILL entropy can be seen as an improvement of the result of [Wee04]: it shows that Yao entropy does not imply HILL entropy in the presence of a (short) random string, because the distribution Z on which X is conditioned is simply the uniform distribution on strings of polynomial length. The separation holds under the quadratic residuosity assumption.

Randomness Extraction. Usually, pseudorandomness extraction is analyzed via HILL entropy, because distributions with HILL entropy are indistinguishable from distributions with the same statistical entropy, and we have tools (namely, randomness extractors [NZ96]) to obtain uniform strings from the latter. Tools are also available to extract from Yao entropy: namely, extractors with a special *reconstruction* property [BSW03]. Our second contribution (Section 4) is to show that considering the Yao entropy and applying a reconstructive extractor can yield many more pseudorandom bits than the traditional analysis, because, according to our first result, Yao entropy can be much higher than HILL entropy. This appears to be the first application of Yao entropy, and also demonstrates the special power of reconstructive extractors.

It is worth mentioning that while our separation of entropies is conditional, the extraction result holds even for the traditional (unconditional) notion of pseudorandomness. The analysis of pseudorandomness of the resulting string, however, relies on the notion of conditional entropy, thus demonstrating that it can be a useful tool even in the analysis of pseudorandomness of unconditional distributions.

Unpredictability Entropy. Unpredictability entropy is a natural formalization of a previously nameless notion that was implicitly used in multiple works.. Our definition essentially says that if some value cannot be predicted from other information with probability higher than 2^{-k}, then it has entropy k when conditioned on that information. For example, when a one-way permutation f is hard to invert with probability higher than 2^{-k}, then conditioned on $f(x)$, the value x has entropy k. The use of *conditional* entropy is what makes this definition meaningful for cryptographic applications.

We demonstrate that almost k pseudorandom bits can be extracted from distributions with unpredictability entropy k, by showing that unpredictability entropy implies conditional Yao entropy, to which reconstruction extractors can be applied. Thus, unpredictability entropy provides a simple language that allows, in particular, known results on hardcore bits of one-way functions to be stated more generally.

We also prove other (fairly straightforward) relations between unpredictability entropy and HILL and Yao conditional entropies.

2 Definitions and Notation

In this section we recall the HILL and Yao definitions of computational entropy (or pseudoentropy) and provide the new, conditional definitions.

Notation. We will use n for the length parameter; our distributions will be on strings of length polynomial in n. We will use s as the circuit size parameter (or running time bound when dealing with Turing machines instead of circuits). To denote a value x sampled from a distribution X, we write $x \leftarrow X$. We denote by $M(X)$ the probability distribution on the outputs of a Turing machine M, taken over the coin tosses (if any) of M and the random choice of the input x according to the distribution X. We use U_n to denote the uniform distribution on $\{0,1\}^n$. For a joint distribution (X, Z), we write X_z to denote the conditional distribution of X when $Z = z$; conversely, given a collection of distributions X_z and a distribution Z, we use (X, Z) to denote the joint distribution given by $\mathbf{Pr}[(X, Z) = (x, z)] = \mathbf{Pr}[Z = z]\,\mathbf{Pr}[X_z = x]$.

We may describe more complicated distributions by describing the sampling process and then the sampled outcome. For example, $\{a \leftarrow X; b \leftarrow X : (a, b)\}$ denotes two independent samples from X, while $\{a \leftarrow X : (a, M(a, Y))\}$ denotes the distribution obtained by sampling X to get a, sampling Y to get b, running $M(a, b)$ to get c, and outputting (a, c).

The statistical distance between two distributions X and Y, denoted by $\mathtt{dist}(X, Y)$, is defined as $\max_T |\,\mathbf{Pr}[T(X) = 1] - \mathbf{Pr}[T(Y) = 1]|$ where T is any test (function). (This is equivalent to the commonly seen $\mathtt{dist}(X, Y) = \frac{1}{2}\sum_a |\,\mathbf{Pr}[X = a] - \mathbf{Pr}[Y = a]|$.) The computational distance with respect to size s circuits, denoted by $\mathtt{cdist}_s(X, Y)$, limits T to be any circuit of size s.

Unconditional Computational Entropy. The min-entropy of a distribution X, denoted by $\mathbf{H}_\infty(X)$, is defined as $-\log(\max_x \mathbf{Pr}[X = x])$. Although min-entropy provides a rather pessimistic view of a distribution (looking only at its worst-case element), this notion is useful in cryptography, because even a computationally unbounded predictor can guess the value of a sample from X with probability at most $2^{-\mathbf{H}_\infty(X)}$. Most results on randomness extractors are formulated in terms of min-entropy of the source distribution.

The first definition says that a distribution has high computational min-entropy if it is *indistinguishable* from some distribution with high statistical min-entropy. It can thus be seen as generalization of the notion of pseudorandomness of [Yao82], which is defined as indistinguishability from uniform.

Definition 1 ([HILL99,BSW03]). *A distribution X has* **HILL entropy** *at least k, denoted by $\mathbf{H}^{\mathsf{HILL}}_{\epsilon,s}(X) \geq k$, if there exists a distribution Y such that $\mathbf{H}_\infty(Y) \geq k$ and $\mathtt{cdist}_s(X, Y) \leq \epsilon$.*

(In [HILL99] Y needs to be efficiently samplable; however, for our application, as well as for [BSW03], samplability is not required.)

Another definition of computational entropy considers compression length. Shannon's theorem [Sha48] says that the minimum compression length of a distribution, by all possible compression and decompression functions, is equal to its average entropy (up to small additive terms). Yao [Yao82] proposed to measure computational entropy by imposing computational constraints on the compression and decompression algorithms.[4] In order to convert this into a worst-case (rather than average-case) metric similar to min-entropy, Barak et al. [BSW03] require that any subset in the support of X (instead of only the entire X) be hard to compress.

[4] Yao called it "effective" entropy.

Definition 2 ([Yao82,BSW03]). *A distribution X has* **Yao entropy** *at least k, denoted by $\mathbf{H}^{\mathsf{Yao}}_{\epsilon,s}(X) \geq k$, if for every pair of circuits c, d (called "compressor" and "decompressor") of total size s with the outputs of c having length ℓ,*

$$\underset{x \leftarrow X}{\mathbf{Pr}}[d(c(x)) = x] \leq 2^{\ell-k} + \epsilon.$$

Note that just like HILL entropy, for $\epsilon = 0$ this becomes equivalent to min-entropy (this can be seen by considering the singleton set of the most likely element).

Conditional Computational Entropy. Before we provide the new conditional definitions of computational entropy, we need to consider the information-theoretic notion of conditional min-entropy.

Let (Y, Z) be a distribution. If we take the straightforward average of the min-entropies $\mathbb{E}_{z \leftarrow Z}[\mathbf{H}_\infty(Y_z)]$ to be the conditional min-entropy, we will lose the relation between min-entropy and prediction probability, which is important for many applications (see e.g. Lemma 4 and Lemma 7). For instance, if for half of Z, $\mathbf{H}_\infty(Y_z) = 0$ and the other half $\mathbf{H}_\infty(Y_z) = 100$, then, given a random z, Y can be predicted with probability over $1/2$, much more than 2^{-50} the average would suggest. A conservative approach, taken in [RW05], would be to take the minimum (over z) of $\mathbf{H}_\infty(Y_z)$. However, this definition may kill "good" distributions like $Y_z = U_n$ for all $z \neq 0^n$ and $Y_z = 0^n$ for $z = 0^n$; although this problem can be overcome by defining a so-called "smooth" version [RW05,RW04], we follow a different approach.

For the purposes of randomness extraction, Dodis et al. [DORS06] observed that because Z is not under adversarial control, it suffices that the *average*, over Z, of the maximum probability is low. They define average min-entropy: $\tilde{\mathbf{H}}_\infty(Y|Z) \stackrel{\text{def}}{=} -\log(\mathbb{E}_{z \leftarrow Z}[2^{-\mathbf{H}_\infty(Y|Z=z)}] = -\log(\mathbb{E}_{z \leftarrow Z}[\max_y \mathbf{Pr}[Y_z = y]])$. This definition averages prediction probabilities before taking the logarithm and ensures that for any predictor P, $\mathbf{Pr}_{(y,z) \leftarrow (Y,Z)}[P(z) = y] \leq 2^{-\tilde{\mathbf{H}}_\infty(Y|Z)}$. It also ensures that randomness extraction works almost as well as it does for unconditional distributions; see Section 4.

Using this definition of conditional min-entropy, defining conditional HILL-entropy is straightforward.

Definition 3 (Conditional HILL entropy). *For a distribution (X, Z), we say X has HILL entropy at least k conditioned on Z, denoted by $\mathbf{H}^{\mathsf{HILL}}_{\epsilon,s}(X|Z) \geq k$, if there exists a collection of distributions Y_z (giving rise to a joint distribution (Y, Z)) such that $\tilde{\mathbf{H}}_\infty(Y|Z) \geq k$ and $\mathtt{cdist}_s((X, Z), (Y, Z)) \leq \epsilon$.*

For conditional Yao entropy, we simply let the compressor and decompressor have z as input.

Definition 4 (Conditional Yao entropy). *For a distribution (X, Z), we say X has Yao entropy at least k conditioned on Z, denoted by $\mathbf{H}^{\mathsf{Yao}}_{\epsilon,s}(X|Z) \geq k$, if for every pair of circuits c, d of total size s with the outputs of c having length ℓ,*

$$\underset{(x,z) \leftarrow (X,Z)}{\mathbf{Pr}}[d(c(x, z), z) = x] \leq 2^{\ell-k} + \epsilon.$$

We postpone the discussion of unpredictability entropy until Section 5.

Asymptotic Definitions. All above definitions are with respect to a single distribution and fixed-size circuits. We are also interested in their asymptotic behaviors, so we consider *distribution ensembles*. In this case, everything is parameterized by n: $X^{(n)}$, $s(n)$, and $\epsilon(n)$. In such a case, whether circuits in our definitions are determined after n is chosen (the nonuniform setting), or whether an algorithm of running time $s(n)$ is chosen independent of n (the uniform setting) makes a difference. We consider the nonuniform setting.

We omit the subscripts $s(n)$ and $\epsilon(n)$ when they "denote" any polynomial and negligible functions, respectively ($\epsilon(n)$ is negligible if $\epsilon(n) \in n^{-\omega(1)}$). More precisely, we write $\mathbf{H}^{\mathsf{HILL}}(X^{(n)}) \geq k(n)$, if there is a distribution ensemble $Y^{(n)}$ such that $\mathbf{H}_\infty(Y^{(n)}) \geq k(n)$ for all n, and for every polynomial $s(n)$, there exists a negligible $\epsilon_s(n)$ such that $\texttt{cdist}_{s(n)}(X^{(n)}, Y^{(n)}) \leq \epsilon_s(n)$. Similarly for the other definitions.

3 Separating HILL Entropy from Yao Entropy

In this section we construct a joint distribution (X, Z),[5] such that given Z, the distribution X has high Yao but low HILL entropy; namely, $\mathbf{H}^{\mathsf{Yao}}(X|Z) \gg \mathbf{H}^{\mathsf{HILL}}(X|Z)$. This is a separation of conditional HILL and Yao entropies. Since Z will be simply a polynomially long random string, this result can also be viewed as a separation of Yao entropy and HILL entropy in the Common Reference String (CRS) model. (In this model one assumes that a uniformly-distributed string of length $q(n)$, for some fixed polynomial q, is accessible to everyone.)

Our construction uses a non-interactive zero knowledge proof system, so we describe it briefly in the following subsection.

3.1 Non-Interactive Zero Knowledge (NIZK)

NIZK was introduced by Blum et al. [BFM88,BDMP91]. For our purposes, a single-theorem variant suffices. Let λ be a positive polynomial and $L \in \mathbf{NP}$ be a language that has witnesses of length n for theorems of lengths $(\lambda(n-1), \lambda(n)]$. (It is easier for us to measure everything in terms of witness length rather than the more traditional theorem length, but they are anyway polynomially related for the languages we are interested in.) NIZK works in the CRS model. Let q be a positive polynomial, and let the CRS be $r \leftarrow U_{q(n)}$ when witnesses are of length n. A NIZK proof system for L is a pair of polynomial-time Turing machines (P, V), called the *prover* and the *verifier* (as well as the polynomial q) such that the following three conditions hold.

1. Completeness: $\forall \phi \in L$ with NP witness w, if $\pi = \mathsf{P}(\phi, w, r)$ is the proof generated by P, then $\mathbf{Pr}_{r \leftarrow U_{q(n)}}[\mathsf{V}(\phi, \pi, r) = 1] = 1$.[6]

[5] Actually, (X, Z) should be defined as a distribution ensemble $(X^{(n)}, Z^{(n)})$, but we'll omit the superscript for ease of notation.

[6] If P is probabilistic, the probability is taken over the choice r and random choices made by P.

2. Soundness: Call r *bad* if $\exists \phi \notin L$, $\exists \pi'$, such that $\mathsf{V}(\phi, \pi', r) = 1$ (and *good* otherwise). Then $\mathbf{Pr}_{r \leftarrow U_{q(n)}}[r \text{ is bad}]$ is negligible in n.
3. Zero-knowledgeness: There is a probabilistic polynomial time Turing machine SIM called the simulator, such that $\forall \phi \in L$ and every witness w for ϕ, $\mathsf{SIM}(\phi) = (\phi, \Pi_{\mathsf{SIM}}, R_{\mathsf{SIM}})$ is computationally indistinguishable from $(\phi, \Pi, R) = \{r \leftarrow U_{q(n)}\ ;\ \pi \leftarrow \mathsf{P}(\phi, w, r) : (\phi, \pi, r)\}$.

For our analysis, we need two additional properties. First, we need the proofs π not to add too much entropy. For this, we use ideas on unique NIZK by Lepinski, Micali and shelat [LMS05]. We do not need the full-fledged uniZK system; rather, the single-theorem system described as the first part of the proof of [LMS05, Theorem 1] suffices (it is based on taking away most of the prover freedom for the single-theorem system of [BDMP91]). The protocol of [LMS05] is presented in the public-key model, in which the prover generates the public key (x, y) consisting of an n-bit modulus x and n-bit value $y \in \mathbb{Z}_x^*$. To make it work for our setting, we simply have the prover generate the public key during the proof and put it into π. Once the public key is fixed, the prover has no further choices in generating π, except choosing a witness w for $\phi \in L$ (note that this actually requires a slight modification to the proof of [LMS05], which we describe in Appendix A).

The second property we need is that the simulated shared randomness R_{SIM} is independent of the simulator input ϕ. It is satisfied by the [LMS05] proof system (as well as by the [BDMP91] system on which it is based).

The zero-knowledge property of the [LMS05] proof system is based on the following assumption (the other properties are unconditional).

Assumption 1 (Quadratic Residuousity [GM84] for Blum Integers) *For all probabilistic polynomial time algorithms P, if p_1 and p_2 are random $n/2$-bit primes congruent to 3 modulo 4, y is a random integer between* 1 *and p_1p_2 with Jacobi symbol $\left(\frac{y}{p_1p_2}\right) = 1$, and $b = 1$ if y is a quadratic residue modulo p_1p_2 and 0 otherwise, then $|1/2 - \mathbf{Pr}[P(y, p_1p_2) = b]|$ is negligible in n.*

The formal statement of the properties we need from [LMS05] follows.

Lemma 1 ([LMS05]+Appendix A). *If the above assumption holds, then there exists an NIZK proof system for any language $L \in$ **NP** with the following additional properties: (1) if r is good and ϕ has t distinct witnesses w, then the number of proofs π for ϕ that are accepted by V is at most $t2^{2n}$, and (2) the string R_{SIM} output by the simulator is independent of the simulator input ϕ.*

3.2 The Construction

Our intuition is based on the separation by Wee [Wee04], who demonstrated an oracle relative to which there is a random variable that has high Yao and low HILL entropy. His oracle consists of a random length-increasing function and an oracle for testing membership in the sparse range of this function. The random variable is simply the range of the function. The ability to test membership in the range helps distinguish it from uniform, hence HILL entropy is low. On the other hand, knowing that a random variable is in the range of a random function does not help to compress it, hence Yao entropy is high.

We follow this intuition, but replace the length-increasing random function and the membership oracle with a pseudorandom generator and an NIZK proof of membership, respectively. Our distribution X consists of two parts: 1) output of a pseudorandom generator and, 2) an NIZK proof that the first part is as alleged. However, an NIZK proof requires a polynomially long random string (shared, but not controlled, by the prover and the verifier). So we consider the computational entropy of X, *conditioned* on a polynomially long random string r chosen from the uniform distribution $Z = U_{q(n)}$.

Let $G : \{0,1\}^n \to \{0,1\}^{\lambda(n)}$, for some polynomial λ, be a pseudorandom generator (in order to avoid adding assumptions, we can build based on Assumption 1), and let $((\mathsf{P},\mathsf{V}),q)$ be the NIZK proof system guaranteed by Lemma 1 for the **NP** language $L = \{\phi \mid \exists\alpha \text{ such that } \phi = G(\alpha)\}$. Let $Z = R = U_{q(n)}$. Our random variable X consists of two parts $(G(U_n),\pi)$, where π is the proof, generated by P, that the first part is an output of G. More precisely, the joint distribution (X,Z) is defined as $\{\alpha \leftarrow U_n \ ;\ r \leftarrow U_{q(n)} \ ;\ \pi \leftarrow \mathsf{P}(G(\alpha),\alpha,r) : ((G(\alpha),\pi),r)\}$. Note that because X contains a proof relative to the random string r, it is defined only after the value r of Z is fixed.

Lemma 2 (Low HILL entropy). $\mathbf{H}^{\mathsf{HILL}}(X|Z) < 3n+1$.

Proof. Suppose there is some collection $\{Y_r\}_{r\in Z}$ for which $\tilde{\mathbf{H}}_\infty(Y|Z) \geq 3n+1$. We will show that there is a distinguisher that distinguishes (X,Z) from (Y,Z). In fact, we will use the verifier V of the NIZK proof system as a universal distinguisher, which works for every such Y.

Let $p(r) \stackrel{\text{def}}{=} \max_y \mathbf{Pr}[Y_r = y]$ be the probability of most likely value of the random variable Y_r.

When r is good, the number of (ϕ,π) pairs for which $\mathsf{V}(\phi,\pi,r) = 1$ is at most 2^{3n}: the total number 2^n of witnesses times the number of proofs 2^{2n} for each witness. Now, parse y as a theorem-proof pair. The number of y such that $\mathsf{V}(y,r) = 1$ is at most 2^{3n}, and each of these y happens with probability at most $p(r)$. Therefore, when r is good, $\mathbf{Pr}_{y\leftarrow Y_r}[\mathsf{V}(y,r) = 1] \leq 2^{3n}p(r)$, by the union bound. Hence, for any r, $\mathbf{Pr}_{y\leftarrow Y_r}[\mathsf{V}(y,r) = 1 \ \wedge\ r \text{ is good}] \leq 2^{3n}p(r)$ (for good r this is the same as above, and for bad r this probability is trivially 0, because of the conjunction).

Now consider running V on a sample from (Y,Z).

$$\begin{aligned}
\mathop{\mathbf{Pr}}_{(y,r)\leftarrow(Y,Z)}[\mathsf{V}(y,r)=1] &\leq \mathop{\mathbf{Pr}}_{r\leftarrow Z}[r \text{ is bad}] + \mathop{\mathbf{Pr}}_{(y,r)\leftarrow(Y,Z)}[\mathsf{V}(y,r)=1 \ \wedge\ r \text{ is good}] \\
&\leq \mathrm{negl}(n) + \mathop{\mathbb{E}}_{r\leftarrow Z}[\mathop{\mathbf{Pr}}_{y\leftarrow Y_r}[\mathsf{V}(y,r)=1 \ \wedge\ r \text{ is good}]] \\
&\leq \mathrm{negl}(n) + \mathop{\mathbb{E}}_{r\leftarrow Z}[2^{3n}p(r)] \\
&\leq \mathrm{negl}(n) + \frac{1}{2}
\end{aligned}$$

(the last inequality follows from the definition of $\tilde{\mathbf{H}}_\infty$: $2^{-\tilde{\mathbf{H}}_\infty(Y|Z)} = \mathbb{E}_{r\leftarrow Z}[p(r)] \leq 2^{-(3n+1)}$).

Since $\mathbf{Pr}_{(x,r)\leftarrow(X,Z)}[\mathsf{V}(x,r)=1]=1$, V distinguishes (X,Z) from (Y,Z) with advantage close to $1/2$. □

Lemma 3 (High Yao entropy). *If Assumption 1 holds, then* $\mathbf{H}^{\mathsf{Yao}}(X|Z) \geq \lambda(n)$.

Proof. Let $s(n)$ be a polynomial. The following two statements imply that under Assumption 1, $\epsilon_s(n) \stackrel{\text{def}}{=} \mathtt{cdist}_{s(n)}((X,Z), \mathsf{SIM}(U_{\lambda(n)}))$ is negligible, by the triangle inequality.

1. $\mathtt{cdist}_{s(n)}((X,Z), \mathsf{SIM}(G(U_n)))$ is negligible. Indeed, fix a seed $\alpha \in \{0,1\}^n$ for G, and let $(X_\alpha, Z) = \{r \leftarrow U_{q(n)}; \pi \leftarrow \mathsf{P}(G(\alpha), \alpha, r) : ((G(\alpha), \pi), r)\}$. By the zero-knowledge property, we know that $\mathtt{cdist}_{s(n)}((X_\alpha, Z), \mathsf{SIM}(G(\alpha)))$ is negligible. Since it holds for every $\alpha \in \{0,1\}^n$, it also holds for a random α; we conclude that $\mathtt{cdist}_{s(n)}((X,Z), \mathsf{SIM}(G(U_n)))$ is negligible.
2. $\mathtt{cdist}_{s(n)}(\mathsf{SIM}(U_{\lambda(n)}), \mathsf{SIM}(G(U_n)))$ is negligible, because G is a pseudorandom generator.

By definition of $\epsilon_s(n)$, if the compressor and decompressor c and d have total size t, then

$$\left| \mathop{\mathbf{Pr}}_{(x,z)\leftarrow(X,Z)}[d(c(x,z),z)=x] \;-\; \mathop{\mathbf{Pr}}_{(x,z)\leftarrow\mathsf{SIM}(U_{\lambda(n)})}[d(c(x,z),z)=x] \right| \leq \epsilon_s(n)\,,$$

where $s = t +$ (size of circuit to check equality of strings of length $|x|$), because we can use $d(c(\cdot,\cdot),\cdot)$ together with the equality operator as a distinguisher.

Let the output length of c be ℓ. Then $\mathbf{Pr}_{(x,z)\leftarrow\mathsf{SIM}(U_{\lambda(n)})}[d(c(x,z),z)=x] \leq 2^{\ell-\lambda(n)}$, because for every fixed z, x contains $\phi \in U_{\lambda(n)}$ (because by Lemma 1, z is independent of ϕ in the NIZK system we use). Hence $\mathbf{Pr}_{(x,z)\leftarrow(X,Z)}[d(c(x,z),z) = x] \leq 2^{\ell-\lambda(n)} + \epsilon_s(n)$, and $\mathbf{H}^{\mathsf{Yao}}_{\epsilon_s(n),t(n)}(X|Z) \geq \lambda(n)$. For every polynomial $t(n)$, the value $s(n)$ is polynomially bounded, and therefore $\epsilon_s(n)$ is negligible, so $\mathbf{H}^{\mathsf{Yao}}(X|Z) \geq \lambda(n)$. □

Remark 1. In the previous paragraph, we could consider also the simulated proof π (recall $x = (\phi, \pi)$) when calculating $\mathbf{Pr}_{(x,z)\leftarrow\mathsf{SIM}(U_{\lambda(n)})}[d(c(x,z),z) = x]$ for even higher Yao entropy. A simulated proof π contains many random choices made by the simulator. Although the simulator algorithm for [LMS05] is not precisely specified, but rather inferred from the simulator in [BDMP91], it is quite clear that the simulator will get to flip at least three random coins per clause in the 3-CNF formula produced out of ϕ in the reduction to 3-SAT (these three coins are needed in order to simulate the location of the $(0,0,0)$ triple [LMS05, proof of Theorem 1, step 9] among the eight triples). This more careful calculation of $\mathbf{Pr}_{(x,z)\leftarrow\mathsf{SIM}(U_{\lambda(n)})}[d(c(x,z),z) = x]$ will yield the slightly stronger statement $\mathbf{H}^{\mathsf{Yao}}(X|Z) \geq \lambda(n) + 3\gamma(n)$, where $\gamma(n)$ is the number of clauses in the 3-CNF formula. This more careful analysis is not needed here, but will be used in Section 4.3.

Since for any polynomial $\lambda(n)$, we have pseudorandom generators of stretch λ, Lemma 2 and Lemma 3 yield the following theorem.

Theorem 1 (Separation). *Under the Quadratic Residuosity Assumption, for every polynomial λ, there exists a joint distribution ensemble $(X^{(n)}, Z^{(n)})$ such that $\mathbf{H}^{\mathsf{Yao}}(X^{(n)} \mid Z^{(n)}) \geq \lambda(n)$ and $\mathbf{H}^{\mathsf{HILL}}(X^{(n)} \mid Z^{(n)}) \leq 3n+1$. Moreover, $Z^{(n)} = U_{q(n)}$ for some polynomial $q(n)$.*

4 Randomness Extraction

As mentioned in the introduction, one of the main applications of computational entropy is the extraction of pseudorandom bits. Based on Theorem 1, in this section we show that the analysis based on Yao entropy can yield many more pseudorandom bits than the traditional analysis based on HILL entropy. Although Theorem 1 is for the conditional setting, we will see an example of extraction that benefits from the conditional-Yao-entropy analysis for the unconditional setting as well.

Before talking about extracting pseudorandom bits from computational entropy, let us look at a tool for analogous task in the information-theoretic setting: an *extractor* takes a distribution Y of min-entropy k, and with the help of a uniform string called the seed, "extracts" the randomness contained in Y and outputs a string of length m that is *almost uniform* even given the seed.

Definition 5 ([NZ96]). *A polynomial-time computable function $E : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m \times \{0,1\}^d$ is a strong (k, ϵ)-extractor if the last d outputs of bits of E are equal to the last d input bits (these bits are called* seed*), and $\mathtt{dist}((E(X, U_d), U_m \times U_d) \leq \epsilon$ for every distribution X on $\{0,1\}^n$ with $\mathbf{H}_\infty(X) \geq k$. The number of extracted bits is m, and the entropy loss is $k - m$.*

There is a long line of research on optimizing the parameters of extractors: minimizing seed length, minimizing ϵ, and maximizing m. For applications of primary interest here—using extracted randomness for cryptography—seed length is less important, because strong extractors can use non-secret random seeds, which are usually much easier to create than the secret from which the pseudorandom bits are being extracted. It is more important to maximize m (as close to k as possible), while keeping ϵ negligible.[7]

4.1 Extracting from Conditional HILL Entropy

It is not hard to see that applying an extractor on distributions with HILL entropy yields pseudorandom bits; because otherwise the extractor together with

[7] This is in contrast to the derandomization literature, where a small constant ϵ suffices, and one is more interested in (simultaneously) maximizing m and minimizing d.

the distinguisher violate the definition of HILL entropy. We show the same for the case of conditional HILL entropy. We reiterate that in the conditional case, the variable Z is given to the distinguisher who is trying to tell the output of the extractor from random.

Lemma 4. *If $\mathbf{H}^{\mathsf{HILL}}_{\epsilon_1,s}(X|Z) \geq k$, then for any $(k - \log\frac{1}{\delta}, \epsilon_2)$-extractor $E : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$,*

$$\underset{s'}{\mathtt{cdist}}\left(\{(x,z) \leftarrow (X,Z) : (E(x,U_d),z)\}, U_m \times U_d \times Z\right) \leq \epsilon_1 + \epsilon_2 + \delta\,,$$

where $s' = s - size(E)$.

Proof. $\mathbf{H}^{\mathsf{HILL}}_{\epsilon_1,s}(X|Z) \geq k$ means that there exists a collection of $\{Y_z\}_{z\in Z}$ such that $\mathtt{cdist}_s((X,Z)(Y,Z)) \leq \epsilon_1$, and $\tilde{\mathbf{H}}_\infty(Y|Z) \geq k$. By Markov's inequality, $\mathbf{Pr}_{z\in Z}[\mathbf{H}_\infty(Y_z) \leq k - \log\frac{1}{\delta}] \leq \delta$. Hence, the extractor works as expected in all but δ fraction of the cases; that is, for all but δ fraction of z values, $\mathtt{dist}(E(Y_z,U_d), U_m \times U_d) \leq \epsilon_2$. Taking expectation over $z \in Z$, we get

$$\mathtt{dist}\left(\{(y,z) \leftarrow (Y,Z) : (E(y,U_d),z)\}, U_m \times U_d \times Z\right) \leq \epsilon_2 + \delta\,,$$

because $\mathtt{dist}$ is bounded by 1. The desired result follows by triangle inequality. □

Remark 2. The entropy loss of E is at least $2\log\frac{1}{\epsilon_2} - O(1)$, by a fundamental constraint on extractors [RT00], giving us a total entropy loss of at least $\log\frac{1}{\delta} + 2\log\frac{1}{\epsilon_2} - O(1)$. The loss of $\log\frac{1}{\delta}$ can be avoided for some specific E, such as pairwise-independent (a.k.a. strongly universal) hashing [CW79], as shown in [DORS06, Lemma 4.2]; because pairwise-independent hashing has optimal entropy loss of $2\log\frac{1}{\epsilon_2} - 2$, this gives us the maximum possible number of extracted bits. The loss of $\log\frac{1}{\delta}$ can be also avoided when $\min_{z\in Z}\mathbf{H}_\infty(Y_z) \geq k$ (as is the case in, e.g., [GKR04]).

Using an extractor on distributions with HILL entropy (the method that we just showed extends to conditional HILL entropy) is a common method for extracting pseudorandom bits. HILL entropy is used, in particular, because it is easier to analyze than Yao entropy. In fact, in the unconditional setting, the only way we know how to show that a distribution has high Yao entropy (incompressibility) is by arguing that it has high HILL entropy (indistinguishability). Nevertheless, Barak et al. [BSW03] showed that some extractors can also extract from Yao entropy.

4.2 Extracting from Conditional Yao Entropy

Barak et al. [BSW03] observed that extractors with the so-called reconstruction procedure can be used to extract from Yao Entropy. Thus, Theorem 1 ($\mathbf{H}^{\mathsf{Yao}}(X|Z) \gg \mathbf{H}^{\mathsf{HILL}}(X|Z)$) suggests that such a *reconstructive* extractor with a Yao-entropy-based analysis may yield more pseudorandom bits than a generic extractor with a traditional HILL-entropy-based analysis. We begin with a definition from [BSW03].

Definition 6 (Reconstruction procedure). *An (ℓ,ϵ)-reconstruction for a function $E : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m \times \{0,1\}^d$ (where the last d outputs are equal to the last d inputs bits) is a pair of machines C and D, where $C : \{0,1\}^n \to \{0,1\}^\ell$ is a randomized Turing machine, and $D^{(\cdot)} : \{0,1\}^\ell \to \{0,1\}^n$ is a randomized oracle Turing machine which runs in time polynomial in n. Furthermore, for every x and T, if $|\,\mathbf{Pr}[T(E(x,U_d))=1]-\mathbf{Pr}[T(U_m\times U_d)=1]| > \epsilon$, then $\mathbf{Pr}[D^T(C^T(x))=x] > 1/2$ (the probability is over the random choices of C and D).*

Trevisan [Tre99] showed, implicitly, that any E with an (ℓ,ϵ)-reconstruction is an $(\ell+\log\frac{1}{\epsilon}, 3\epsilon)$-extractor, and Barak et al. [BSW03] showed that such extractors can be used to extract pseudorandom bits from distributions with Yao entropy. We extend the proof of Barak et al. so that their result holds for the conditional version of Yao entropy.

Lemma 5. *Let X be a distribution with $\mathbf{H}^{\mathsf{Yao}}_{\epsilon,s}(X|Z) \geq k$, and let E be an extractor with a $(k-\log\frac{1}{\epsilon},\epsilon)$-reconstruction (C,D). Then $\mathtt{cdist}_{s'}((E(X,U_d),Z), U_m \times U_d \times Z) \leq 5\epsilon$, where $s' = s/(size(C)+size(D))$.*

Proof. Assume, for the purpose of contradiction, that there is a distinguisher T of size s' such that $\mathbf{Pr}[T(E(X,U_d),Z)=1] - \mathbf{Pr}[T(U_m \times U_d \times Z)=1] > 5\epsilon$. By the Markov inequality, there is a subset S in the support of (X,Z) such that $\mathbf{Pr}[(X,Z)\in S] \geq 4\epsilon$, and $\forall (x,z) \in S$, $\mathbf{Pr}[T(E(x,U_d),z)=1] - \mathbf{Pr}[T(U_m \times U_d, z)=1] > \epsilon$. For every pair $(x,z)\in S$, $\mathbf{Pr}[D^{T(\cdot,z)}(C(x)) = x] > 1/2$, where the probability is over the random choices of C and D. Thus, there is a fixing of the random choices of C and D, denoted by circuits $\bar{C},\bar{D}$, such that $\mathbf{Pr}_{(x,z)\leftarrow(X,Z)}[\bar{D}^{T(\cdot,z)}(\bar{C}(x)) = x] > 2\epsilon$. Let $c(x,z) = \bar{C}(x)$ and $d(y,z) = \bar{D}^{T(\cdot,z)}(y)$ be the compression and decompression circuits, respectively. Then $\mathbf{Pr}_{(x,z)\leftarrow(X,Z)}[d(c(x,z),z)=x] > 2\epsilon = 2^{\ell-k}+\epsilon$, a contradiction. □

The above lemma does not yield more pseudorandom bits when given a distribution that has high Yao but low HILL entropy, unless we have a reconstructive extractor with long output length (compared to generic extractors, which work for HILL entropy). Fortunately, there is a simple way to increase the output length of a reconstructive extractor, at the expense of increasing the seed length; namely, by applying the extractor multiple times on the same input distribution but each time with an independent fresh seed. Furthermore, there do exist reconstructive extractors; e.g., the Goldreich-Levin extractor: $GL(x,y) \stackrel{\text{def}}{=} (x\cdot y)\circ y$, where $\circ$ denotes concatenation and $\cdot$ denotes inner product. Below, we describe more precisely how to increase the output length. For a proof, we refer the readers to Section 3.5 in the survey by Shaltiel [Sha02].

Proposition 1. *Let $GL : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}\times\{0,1\}^n$ be an extractor with (ℓ,ϵ)-reconstruction. Then $E : \{0,1\}^n \times \{0,1\}^{mn} \to \{0,1\}^m \times \{0,1\}^{mn}$ defined below is an extractor with $(m+\ell, m\epsilon)$-reconstruction. Let $\circ$ denote component-wise concatenation (i.e., to agree syntactically with the definition of extractor, we concatenate the 1-bit outputs and the n-bit seeds separately)*

$$E(x,y_1,\ldots,y_m) \stackrel{\text{def}}{=} GL(x,y_1)\circ\cdots\circ GL(x,y_m)\,.$$

For the Goldreich-Levin extractor, $\ell = O(\log \frac{1}{\epsilon})$. Then Lemma 5 implies that E extracts m pseudorandom bits out of any distribution that has Yao entropy $m + \ell + \log \frac{1}{\epsilon} = m + O(\log \frac{1}{\epsilon})$. This shows that it is possible to extract almost all of Yao entropy (e.g., if the negligible $\epsilon = 2^{-\mathrm{polylog}(n)}$ suffices, then all but a polylogarithmic amount of entropy can be extracted).

Using the distribution of Theorem 1, we can set $\epsilon = 2^{-n}$ to extract $\lambda(n) - O(n)$ bits from X that are pseudorandom even given Z. This is more than the linear number of bits extractable from X using the analysis based on conditional HILL entropy.

4.3 Unconditional Extraction

In this subsection, let $(X, Z) = ((G(U_n), \mathit{\Pi}), R) = \{\alpha \leftarrow U_n\ ;\ r \leftarrow U_{q(n)}\ ;\ \pi \leftarrow \mathsf{P}(G(\alpha), \alpha, r) : ((G(\alpha), \pi), r)\}$ as defined in Section 3.2. The question is: how many pseudorandom bits can we extract from the unconditional distribution (X, Z)? Surprisingly, analysis based on conditional entropy yields more bits than unconditional analysis, demonstrating that the notion of conditional entropy may be a useful tool even in the analysis of pseudorandomness of unconditional distributions.

Analysis based on unconditional entropy. The straightforward way is to apply an extractor on (X, Z). This gives us almost k pseudorandom bits provided that $\mathbf{H}^{\mathsf{HILL}}(X, Z) \geq k$, or $\mathbf{H}^{\mathsf{Yao}}(X, Z) \geq k$ for reconstructive extractors (see previous subsections). However, the best we can show is that $\mathbf{H}^{\mathsf{HILL}}(X, Z) = \lambda(n) + q(n) + O(n)$ (the analysis appears in Appendix B), and hence we cannot prove, using HILL entropy, that more than $\lambda(n) + q(n) + O(n)$ bits can be extracted. On the other hand, we do not know if $\mathbf{H}^{\mathsf{Yao}}(X, Z)$ is higher; this is closely related to the open problem of whether HILL entropy is equivalent to Yao entropy, and appears to be difficult.[8] Thus, analysis based on unconditional entropy does not seem to yield more than $\lambda(n) + q(n) + O(n)$ bits.

More bits from conditional Yao entropy. Analysis based on conditional HILL entropy seems to yield even fewer bits (see Lemma 2). But using conditional Yao entropy, we get the following result.

Lemma 6. *It is possible to extract* $4\lambda(n) + q(n) - O(n)$ *pseudorandom bits out of* (X, Z).

[8] To show that $\mathbf{H}^{\mathsf{Yao}}(X, Z)$ is high, one would have to show that the pair (X, Z) cannot be compressed; the same indistinguishability argument as in Lemma 3 does not work for the pair (X, Z), because in the simulated distribution, Z is simulated and thus has less entropy. It is thus possible that both the real distribution (where Z is random and ϕ in X is pseudorandom) and the simulated distribution (where ϕ is random and Z is pseudorandom), although indistinguishable, can be compressed with the help of the proof π.

Proof (Sketch). According to Remark 1 following Lemma 3, we can show that the conditional Yao entropy $\mathbf{H}^{\mathsf{Yao}}(X|Z) \geq \lambda(n) + 3\gamma(n)$, where $\gamma(n)$ is the number of clauses in the 3-CNF formula produced from ϕ in the reduction from L to 3-SAT. Since $\gamma(n) \geq \lambda(n)$, we can extract $4\lambda(n) - O(n)$ bits from X that are pseudorandom even given Z, by the last paragraph of Section 4.2. Noting that Z is simply a uniform string[9], we can append it to the pseudorandom bits extracted from X and obtain an even longer pseudorandom string. Thus, we get $4\lambda(n) + q(n) - O(n)$ pseudorandom bits using the analysis based on conditional Yao entropy. □

5 Unpredictability Entropy

In this section, we introduce a new computational entropy, which we call unpredictability entropy. Analogous to min-entropy, which is the logarithm of the maximum predicting probability, unpredictability entropy is the logarithm of the maximum predicting probability where the predictor is restricted to be a circuit of polynomial size. Note that in the unconditional setting, unpredictability entropy is just min-entropy; a small circuit can have the most likely value hardwired. In the conditional setting, however, this new definition can be very different from min-entropy, and in particular, allows us to talk about the entropy of a value that is unique, such as g^{xy} where g^x and g^y are known to the observer, and possibly even verifiable, such as the preimage x of a one-way permutation f, where $y = f(x)$ is known to the observer.

Definition 7 (Unpredictability entropy). *For a distribution (X, Z), we say that X has* **unpredictability entropy** *at least k conditioned on Z, denoted by $\mathbf{H}^{\mathsf{unp}}_{\epsilon,s}(X|Z) \geq k$, if there exists a collection of distributions Y_z (giving rise to a joint distribution (Y, Z)) such that $\mathtt{cdist}_s((X, Z), (Y, Z)) \leq \epsilon$, and for all circuits C of size s,*

$$\mathbf{Pr}[C(Z) = Y] \leq 2^{-k}.$$

Remark 3. The parameter ϵ and the variable Y do not seem to be necessary in the definition; we can simply require $\mathbf{Pr}[C(Z) = X] \leq 2^{-k}$. However, they make this definition *smooth* [RW04] and easier to compare with existing definitions of HILL and Yao entropy.

Remark 4. Note that our entropy depends primarily on the predicting probability, as opposed to on the size of the predicting circuit or the combination of both (see e.g., [TSZ01,HILL99]). We choose to have s fixed, in order to accommodate distributions with nonzero information-theoretic entropy; otherwise the computational entropy of such distribution would be infinite because the predicting probability doesn't increase no matter how big the predicting circuit grows. For the case of one-way function, unpredictability entropy is what is often called "hardness." This notion is more general, and provides a simple language

[9] In case Z is not uniform but contains some amount of entropy, we can apply another extractor on it.

for pseudorandomness extraction: namely, a distribution with computational entropy k contains k pseudorandom bits that can be extracted (see below).

5.1 Relation to Other Notions and Bit Extraction

In this subsection we show that high conditional HILL entropy implies high unpredictability entropy, which in turn implies high conditional Yao entropy. Note that, assuming exponentially strong one-way permutations f exist, unpredictability entropy does not imply conditional HILL entropy: simply let $(X, Z) = (x, f(x))$.

Lemma 7. $\mathbf{H}^{\mathsf{HILL}}_{\epsilon,s}(X|Z) \geq k \Rightarrow \mathbf{H}^{\mathsf{unp}}_{\epsilon,s}(X|Z) \geq k.$

Proof. $\mathbf{H}^{\mathsf{HILL}}_{\epsilon,s}(X|Z) \geq k$ means that there is a Y such that $\tilde{\mathbf{H}}_\infty(Y|Z) \geq k$ and $\texttt{cdist}_s((X,Z),(Y,Z)) \leq \epsilon$. And $\tilde{\mathbf{H}}_\infty(Y|Z) \geq k$ means that $\mathbb{E}_{z\leftarrow Z}[\max_y \mathbf{Pr}[Y = y|Z = z]] \leq 2^{-k}$, which implies that for all circuits C of size s, $\mathbf{Pr}[C(Z) = Y] \leq 2^{-k}$. □

Lemma 8. $\mathbf{H}^{\mathsf{unp}}_{\epsilon,s}(X|Z) \geq k \Rightarrow \mathbf{H}^{\mathsf{Yao}}_{\epsilon,s}(X|Z) \geq k.$

Proof. $\mathbf{H}^{\mathsf{unp}}_{\epsilon,s}(X|Z) \geq k$ means that there is a collection of $\{Y_z\}_{z\in Z}$ such that $\texttt{cdist}_s((X,Z),(Y,Z)) \leq \epsilon$, and for all circuits C of size s, $\mathbf{Pr}[C(Z) = Y] \leq 2^{-k}$. We will show that $\mathbf{H}^{\mathsf{Yao}}_{0,s}(Y|Z) \geq k$, which in turn implies $\mathbf{H}^{\mathsf{Yao}}_{\epsilon,s}(X|Z) \geq k$.

Suppose for contradiction that $\mathbf{H}^{\mathsf{Yao}}_{0,s}(Y|Z) < k$. Then there exists a pair of circuits c, d of total size s with the outputs of c having length ℓ, such that $\mathbf{Pr}_{(y,z)\leftarrow(Y,Z)}[d(c(y,z),z) = y] > 2^{\ell-k}$. Because $|c(y,z)| = \ell$, guessing the correct value is at least $2^{-\ell}$, so $\mathbf{Pr}_{(a,y,z)\leftarrow(U_\ell,Y,Z)}[d(a,z) = y] > 2^{\ell-k} \cdot 2^{-\ell} = 2^{-k}$, a contradiction since $d(a,\cdot)$ (with some fixing of a) is a circuit of size at most s. So $\mathbf{H}^{\mathsf{Yao}}_{0,s}(Y|Z) \geq k$.

Next, suppose for contradiction that $\mathbf{H}^{\mathsf{Yao}}_{\epsilon,s}(X|Z) < k$. Then there exists a pair of circuits c, d of total size s with the outputs of c having length ℓ, such that $\mathbf{Pr}_{(x,z)\leftarrow(X,Z)}[d(c(x,z),z) = x] > 2^{\ell-k} + \epsilon$. But $\mathbf{Pr}_{(y,z)\leftarrow(Y,Z)}[d(c(y,z),z) = y] \leq 2^{\ell-k}$, which means that $d(c(\cdot,\cdot),\cdot)$ can be used to distinguish (X,Z) from (Y,Z) with advantage more than ϵ, a contradiction to $\texttt{cdist}_s((X,Z),(Y,Z)) \leq \epsilon$. Hence $\mathbf{H}^{\mathsf{Yao}}_{\epsilon,s}(X|Z) \geq k$. □

From Section 4, we know how to extract almost k bits from distributions with Yao entropy k, by using reconstructive extractors. Lemma 8 implies that the same method works for unpredictability entropy. Thus, the notion of unpredictability entropy allows for more general statements of results on hardcore bits (such as, for example, [GL89,TSZ01]), which are usually formulated in terms of one-way functions. Most often these results generalize easily to other conditionally unpredictable distributions, for instance, the Diffie-Hellman distribution $(g^{xy} \mid g, g^x, g^y)$. However, such generalization is not automatic, because a prediction of a one-way function inverse is verifiable (namely, knowing y, one can check if the guess for $f^{-1}(y)$ is correct), while a guess of a value of a conditionally unpredictable distribution in general is not (indeed, the Diffie-Hellman distribution does not have it unless the decisional Diffie-Hellman problem is easy). Thus, it would be beneficial if results were stated for the more general case of unpredictable distributions whenever such verifiability is not crucial. Unpredictability entropy provides a simple language for doing so.

6 Acknowledgments

We thank anonymous referees for their helpful comments, and Moni Naor for pointing out related work. This work supported was in part by the US National Science Foundation grants CCR-0311485, CCF-0515100 and CNS-0546614, the Taiwan National Science Council grants NSC95-2218-E-001-001, NSC95-2218-E-011-015 and NSC95-3114-P-001-002-Y02, and the Institute for Pure and Applied Mathematics at UCLA.

References

[BDMP91] Manuel Blum, Alfredo De Santis, Silvio Micali, and Giuseppe Persiano. Noninteractive zero-knowledge. *SIAM Journal on Computing*, 20(6):1084–1118, December 1991.

[BFM88] Manuel Blum, Paul Feldman, and Silvio Micali. Non-interactive zero-knowledge and its applications (extended abstract). In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, pages 103–112, Chicago, Illinois, 2–4 May 1988.

[BSW03] Boaz Barak, Ronen Shaltiel, and Avi Wigderson. Computational analogues of entropy. In Sanjeev Arora, Klaus Jansen, José D. P. Rolim, and Amit Sahai, editors, *RANDOM-APPROX 2003*, volume 2764 of *LNCS*, pages 200–215. Springer, 2003.

[CW79] J.L. Carter and M.N. Wegman. Universal classes of hash functions. *Journal of Computer and System Sciences*, 18:143–154, 1979.

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

[DORS06] Yevgeniy Dodis, Rafail Ostrovsky, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. Technical Report 2003/235, Cryptology ePrint archive, `http://eprint.iacr.org`, 2006. Previous version appeared at *EUROCRYPT 2004*.

[GKR04] Rosario Gennaro, Hugo Krawczyk, and Tal Rabin. Secure hashed Diffie-Hellman over non-DDH groups. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology—EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 361–381. Springer-Verlag, 2004.

[GL89] O. Goldreich and L. 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.

[GM84] S. Goldwasser and S. Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, April 1984.

[HILL99] J. Håstad, R. Impagliazzo, L.A. Levin, and M. Luby. Construction of pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999.

[Imp99] Russell Impagliazzo. Remarks in open problem session at the dimacs workshop on pseudorandomness and explicit combinatorial constructions, 1999.

[LMS05] Matt Lepinski, Silvio Micali, and Abhi Shelat. Fair-zero knowledge. In Joe Kilian, editor, *TCC*, volume 3378 of *LNCS*, pages 245–263. Springer-Verlag, 2005.

[Nao96] Moni Naor. Evaluation may be easier than generation. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing*, pages 74–83, Philadelphia, Pennsylvania, 22–24 May 1996.

[NZ96] Noam Nisan and David Zuckerman. Randomness is linear in space. *Journal of Computer and System Sciences*, 52(1):43–53, 1996.

[RT00] Jaikumar Radhakrishnan and Amnon Ta-Shma. Bounds for dispersers, extractors, and depth-two superconcentrators. *SIAM Journal on Computing*, 13(1):2–24, 2000.

[RW04] Renato Renner and Stefan Wolf. Smooth rényi entropy and applications. In *Proceedings of IEEE International Symposium on Information Theory*, page 233, June 2004.

[RW05] Renato Renner and Stefan Wolf. Simple and tight bounds for information reconciliation and privacy amplification. In Bimal Roy, editor, *Advances in Cryptology—ASIACRYPT 2005*, LNCS, Chennai, India, 4–8 December 2005. Springer-Verlag.

[Sha48] Claude E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423 and 623–656, July and October 1948. Reprinted in D. Slepian, editor, *Key Papers in the Development of Information Theory*, IEEE Press, NY, 1974.

[Sha02] Ronen Shaltiel. Recent developments in explicit constructions of extractors. *Bulletin of the EATCS*, 77:67–95, 2002.

[Tre99] Luca Trevisan. Construction of extractors using pseudo-random generators (extended abstract). In *STOC*, pages 141–148, 1999.

[TSZ01] Amnon Ta-Shma and David Zuckerman. Extractor codes. In *STOC*, pages 193–199, 2001.

[TVZ05] Luca Trevisan, Salil P. Vadhan, and David Zuckerman. Compression of samplable sources. Technical Report TR05-012, Electronic Colloquium on Computational Complexity (ECCC), 2005.

[Wee04] Hoeteck Wee. On pseudoentropy versus compressibility. In *IEEE Conference on Computational Complexity*, pages 29–41. IEEE Computer Society, 2004.

[Yao82] A. C. Yao. Theory and applications of trapdoor functions. In *23rd Annual Symposium on Foundations of Computer Science*, pages 80–91, Chicago, Illinois, 3–5 November 1982. IEEE.

A Modifications to the Proof of [LMS05]

The proof of Theorem 1 in [LMS05] requires the n-bit modulus x chosen by the prover (and, in our case, included as part of the proof) to be a Blum integer, i.e., a product of two primes that are each congruent to 3 modulo 4. However, the proof π (using the techniques from [BDMP91]) guarantees only that x is "Regular(2)," i.e., is square-free and has exactly two distinct odd prime divisors. In other words, we are assured only that x is of the form $p^i q^j$ for some odd primes

p, q and some i, j not simultaneously even. Soundness does not suffer if a prover maliciously chooses such an x that is not a Blum integer, but the uniqueness property does: there may be more than one valid proof π, because π consists of square roots s of values in $\mathbb{Z}_x^*$ such that the Jacobi symbol $\left(\frac{s}{x}\right) = 1$ and $s < x/2$, and there may be more than one such square root if x is not a Blum integer.

One approach to remedy this problem is to use the technique proposed in countable zero-knowledge of Naor [Nao96, Theorem 4.1]: to include into π the proof that x is a Blum integer. Another, simpler, approach (which does not seem to work for the problem in [Nao96], because the length of the primes is important there) is to require the verifier to check that $x \equiv 1 \pmod 4$. This guarantees that either $p \equiv q \equiv 3 \bmod 4$ and i, j are odd, in which case uniqueness of a square root $r < x/2$ with $\left(\frac{r}{x}\right) = 1$ is guaranteed, or $p^i \equiv q^j \equiv 1 \bmod 4$, in which case simple number theory (case analysis by the parity of i, j) shows that half the quadratic residues in $\mathbb{Z}_x^*$ have *no* square root r with $\left(\frac{r}{x}\right) = 1$. Thus, such an x that allows for non-unique proofs is very unlikely to work for a shared random string r, and we can simply add strings r for which such an x exists to the set of bad strings (which will remain of negligible size).

B Unconditional HILL Entropy of (X, Z)

Recall that $(X, Z) = ((G(U_n), \Pi), R) = \{\alpha \leftarrow U_n \; ; \; r \leftarrow U_{q(n)} \; ; \; \pi \leftarrow \mathsf{P}(G(\alpha), \alpha, r) : ((G(\alpha), \pi), r)\}$. Below, we show that $\mathbf{H}^{\mathsf{HILL}}(X, Z) \geq \lambda(n) + q(n) + O(n)$; it is unclear if higher HILL entropy can be shown. The discussion assumes some familiarity with the NIZK system for 3-SAT, by Lepinski, Micali, and shelat [LMS05].

By the zero-knowledgeness, the output distribution $(X_{\mathsf{SIM}}, Z_{\mathsf{SIM}})$ of the simulator is indistinguishable from (X, Z). So $\mathbf{H}^{\mathsf{HILL}}(X, Z)$ is no less than the min-entropy of $(X_{\mathsf{SIM}}, Z_{\mathsf{SIM}})$. We count how many choices the simulator SIM has: there are,

- $2^{\lambda(n)}$ theorems to prove,
- fewer than 2^{2n} proving pairs to choose from (a proving pair is an n-bit Blum integer x and an n-bit quadratic residue $y \in \mathbb{Z}_x^*$),
- $2^{q(n) - \kappa(n)}$ choices for shared "random" string r, where $\kappa(n)$ is the number of Jacobi symbol 1 elements of $\mathbb{Z}_x^*$ included in r (because in the simulated r, these elements must be quadratic residues in $\mathbb{Z}_x^*$),
- $2^{\kappa(n)}$ choices for claiming, in the simulated proof, whether each of the Jacobi symbol 1 elements in r is a quadratic residue or a quadratic nonresidue (the simulator gets to make false claims about that, because in the simulated r, they are all residues).

Taking the logarithm of the number of choices, we have $\mathbf{H}^{\mathsf{HILL}}(X, Z) \geq \lambda(n) + q(n) + O(n)$. This seems to be the best we can do, as we do not know whether there are other distribution that is indistinguishable from (X, Z).