Inaccessible Entropy

Salil Vadhan

(joint work with Iftach Haitner, Omer Reingold, and Hoeteck Wee)

Computational analogues of information-theoretic notions have given rise to some of the most interesting phenomena in the theory of computation. For example, a computational analogue of entropy, known as *pseudoentropy*, introduced by Håstad, Impagliazzo, Levin, and Luby [HILL], was the 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.

In this work, we introduce another computational analogue of entropy, which we call *accessible entropy*, and present several applications of it to the foundations of cryptography. Before describing accessible entropy (and a complementary notion of *inaccessible entropy*), we recall the standard information-theoretic notion of entropy and the computational notion of pseudoentropy of Håstad et al.

Entropy and Pseudoentropy. Recall that the *entropy* of a random variable X is defined to be $\mathrm{H}(X) := \mathrm{E}_{x \overset{\mathrm{R}}{\leftarrow} X}[\log(1/\Pr[X = x])$, which measures the number of "bits of randomness" in X (on average). We will refer to $\mathrm{H}(X)$ as the *real entropy* of X to contrast with the computational analogues that we study. Håstad et al. [HILL] say that a random variable X has *pseudoentropy* (at least) k if there exists a random variable Y of entropy (at least) k such that X and Y are computationally indistinguishable.

The reason that pseudoentropy is interesting and useful is that there exist random variables X whose pseudoentropy is larger than their real entropy. For example, the output of a pseudorandom generator $G : \{0,1\}^\ell \to \{0,1\}^n$ on a uniformly random seed has entropy at most ℓ, but has pseudoentropy n (by definition). Håstad et al. proved that in fact, from *any* efficiently samplable distribution X whose pseudoentropy is noticeably larger than its real entropy, it is possible to construct a pseudorandom generator. By showing, in addition, how to construct such a distribution X from any one-way function, Håstad et al. prove their theorem that the existence of one-way functions implies the existence of pseudorandom generators.

The notion of pseudoentropy is only useful, however, as a lower bound on the "computational entropy" in a distribution. Indeed, it can be shown that every distribution on $\{0,1\}^n$ is computationally indistinguishable from a distribution of entropy at most $\mathrm{poly}(\log n)$. While several other computational analogues of entropy have been studied in the literature (cf., [BSW]), all of these are also meant to serve as ways of capturing the idea that a distribution "behaves like" one of higher entropy. In this paper, we explore a way in which a distribution can "behave like" one of much *lower* entropy.

Accessible Entropy. We motivate the idea of accessible entropy with an example. Consider an algorithm G that gets as input a random function h :

$\{0,1\}^n \to \{0,1\}^m$ from a family of collision-resistant hash functions (where $m \ll n$), chooses a random $x \overset{\text{R}}{\leftarrow} \{0,1\}^n$, sets $y = h(x)$, and outputs the pair (y, x).

Now, information-theoretically, the second block of G's output (namely x) has entropy at least $n-m$ conditioned on the input h and the first block y, because $y = h(x)$ reveals only m bits of information about x. However, the collision-resistance property says that given the *state* of G after the first block, there is at most one consistent value of x that G can reveal with nonnegligible probability. (Otherwise, G would be able find two distinct messages $x \neq x'$ such that $h(x) = h(x')$.) This holds even if G is replaced by any polynomial-time adversary G^*. Thus, there is "real entropy" in x (conditioned on the history) but it is "computationally inaccessible" to G^*, to whom x effectively has entropy 0.

We generalize this basic idea to allow the upper bound on the "accessible entropy" to be a parameter k, and to consider both the real and accessible entropy accumulated over several blocks. In more detail, consider an m-block generator G that on input z, outputs a sequence $(y_1, \ldots, y_m)$ of blocks, and let $(Z, Y_1, \ldots, Y_m)$ be random variables denoting a random input Z to G and the output blocks of $\mathsf{G}(Z)$ (when G's coin tosses are chosen uniformly at random). We define the *real entropy* of G to be

$$\sum_i \mathrm{H}(Y_i | Z, Y_1, \ldots, Y_{i-1}),$$

where $\mathrm{H}(X|Y) = \mathrm{E}_{y \overset{\text{R}}{\leftarrow} Y}[\mathrm{H}(X|_{Y=y})]$ is the standard notion of conditional entropy.

To define *accessible entropy*, consider a probabilistic polynomial-time adversary G^* that receives an input z, and then in sequence of m stages, tosses some fresh random coins s_i and computes and outputs a block y_i. At the end it should also justify that it has behaved consistently with the honest algorithm G by producing coin tosses r for G such that G would have output $(y_1, \ldots, y_m)$ on input z and coin tosses r. (For simplicity we restrict attention to G^* that always produce correct justifications, though our definitions and results can be generalized also to handle G^* that sometimes fail to do so.) Now, let $(Z, S_1, Y_1, S_2, Y_2, \ldots, S_m, Y_m)$ be random variables corresponding to the sequence of coins S_i and outputs Y_i of G^* on a random input Z. Then we define the *accessible entropy* achieved by G^* to be

$$\sum_i \mathrm{H}(Y_i | Z, S_1, \ldots, S_{i-1}).$$

The key point is that now we compute the entropy conditioned not just on the previous blocks, but on the entire local state of G^* prior to generating the i'th block. (We don't need to include Y_j for $j < i$ since these are determined by Z and $S_1, \ldots, S_j$.)

The collision resistance example given earlier shows that there can be generators G whose computationally accessible entropy is much smaller than the real Shannon entropy. Indeed, in that protocol, the real entropy of G's blocks is n (namely, the total entropy in x), but the computationally accessible entropy is at most $m + \mathrm{neg}(n)$, where $m \ll n$ is the output length of the collision-resistant hash function. (Here we are counting the conditional entropy in all of G's blocks for simplicity, but the definitions generalize naturally if we only want to sum the conditional entropies over some subset of blocks.) Thus, in contrast to pseudoentropy, accessible entropy is useful for expressing the idea that the "computational entropy" in a distribution is *smaller* than its real entropy. We refer to the difference (real entropy) − (accessible entropy) as the *inaccessible entropy* of G.

Applications. We have used the notion of inaccessible entropy and variants to:

- Give a much simpler and more efficient construction of statistically hiding commitment schemes from arbitrary one-way functions.
- Prove that constant-round statistically hiding commitments are necessary for constructing constant-round zero-knowledge proof systems for NP that remain secure under parallel composition (assuming the existence of one-way functions).
- Give a simpler construction of universal one-way hash functions and hence digital signature schemes from one-way functions. This appears in a follow-up subsequent paper [HRVW2]
- Inspire a simpler and more efficient construction of pseudorandom generators from one-way functions [HRV].

Bibliographic Note. Our paper [HRVW1] utilizes a more general (and more involved) notion of inaccessible entropy for *protocols*. The simpler notion of inaccessible entropy generators described above and the simple construction of such generators from one-way functions described in the talk will eventually be incorporated into the paper.

References

[BSW] B. Barak, R. Shaltiel, and A. Wigderson. Computational Analogues of Entropy. In *RANDOM-APPROX*, 2003.

[HRVW1] 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. Full version available as *Electronic Colloquium on Computational Complexity* TR09-045, May 2009.

[HRVW2] Iftach Haitner, Omer Reingold, Salil Vadhan, and Hoeteck Wee. Universal one-way hash functions via inaccessible entropy. Unpublished manuscript, October 2009.

[HRV] Iftach Haitner, Omer Reingold, Salil Vadhan. Efficiency improvements in constructing pseudorandom generators from one-way functions. Unpublished manuscript, November 2009.

[HILL] J. Håstad, R. Impagliazzo, L. A. Levin, and M. Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999. Preliminary versions in *STOC'89* and *STOC'90*.