COS598D Lecture 3
Pseudorandom generators from one-way functions

Scribe: Moritz Hardt, Srdjan Krstic

February 22, 2008

In this lecture we prove the existence of pseudorandom-generators assuming that one-way functions exist (Hastad, Impagliazzo, Levin and Luby '99). Our proof is with respect to non-uniform security. We also sketch the uniform case which requires a uniform version of Impagliazzo's hardcore set lemma that uses ideas from learning theory.

1 Basic Concepts

We briefly recall some standard notions from cryptography. For details the reader is referred to one of the many textbooks on this topic.

1.1 Pseudorandom generators

In order to define pseudorandom generators, we need the notion of computational indistinguishability.

Definition 1 We say two families of distributions $\{X_n\}$ and $\{Y_n\}$ are *computationaly indistinguishable* and write $\{X_n\} \approx \{Y_n\}$ if for every family of polynomially sized circuits $\{C_n\}$, there exists a negligible function $\epsilon : \mathbb{N} \to [0,1]$, i.e., $\epsilon = n^{-\omega(1)}$, such that

$$|Pr[C_n(X_n) = 1] - Pr[C_n(Y_n) = 1]| < \epsilon(n).$$

Definition 2 We say that a family $\{X_n\}$ is *pseudorandom* if it is computationally indistinguishable from the uniform distribution, i.e., $\{X_n\} \approx \{U_n\}$.

A function $G : \{0,1\}^n \to \{0,1\}^{l(n)}$ is called a *pseudorandom generator* (PRG) if G is computable in polynomial time and $G(U_n)$ is pseudorandom.

We are only interested in PRGs where $l(n) > n$. As soon as this is the case, we can get pseudorandom generators of arbitrary stretch $l'(n) > l(n)$. Hence, it suffices to construct PRGs that map n bits to $n+1$ bits.

1.2 One-way functions and permutations

Definition 3 We say that a function $f : \{0,1\}^* \rightarrow \{0,1\}^*$ is a *one-way function* (OWF) if f is computable in polynomial time and for every family of polynomial sized circuits $\{C_n\}$, there exists a negligible function ϵ such that

$$P_{x \sim U_n}[f(C_n(f(x))) = f(x)] < \epsilon.$$

In case f is one-to-one, we will call f a *one-way permutation.*

For convenience and without loss of generality, we will assume that $|f(x)| = |x|$.

1.3 Goldreich-Levin theorem, informally

Suppose we have a one-way *permutation* f and consider the function $G(x,r) = f(x)r(x \odot r)$, where $x \odot r$ the inner product $\sum_i x_i r_i \mod 2$. What Goldreich and Levin '89 show is that no efficient algorithm can "predict" the bit $x \odot r$ given r and $f(x)$ with more than negligible success probability for random x and r. This kind of unpredictability was known to imply pseudorandomness (Blum-Micali, Yao '82). Therefore, $G(x,r)$ is in fact a pseudorandom generator mapping strings of length $2n$ into strings of length $2n+1$. In other words, the existence of pseudorandom generators is implied by the existence of one-way permutation.

Unfortunately, G need not be secure when f is not a permutation. A one-way function could, for instance, always map into strings ending with a zero. As we will see next, the case of one-way functions requires significantly more work.

2 Main theorem and proof outline

Around 1989, Impagliazzo and Luby [IL89] showed that the existence of various cryptographic primitives, such as encryption schemes, message authentication codes, signature schemes follows from the existence of one-way functions. A major challenge was to obtain pseudorandom generators from the same assumption. This result was first achieved by Impagliazzo, Luby and Levin [ILL89] with respect to nonuniform security. Later, Hastad extended the result to the uniform case. A joint journal version appeared in '99 [HILL99].

Theorem 1 ([HILL99]) *Assuming the existence of a one-way function, there exists a pseudorandom generator.*

The original construction was highly impractical. An n-bit one-way function (think of $n > 100$) would only give a pseudorandom generator of seed length n^{32}. Recently, Holenstein [Hol06] improved (giving n^8) and simplified the construction. The best known seed length based on general one-way functions is n^7 due to Haitner et al. '06. Our proof is similar to that of Holenstein, although we only sketch the uniform case. The proof strategy can be summarized as follows.

1. We define a notion of pseudo-entropy in analogy with the notion of pseudorandomness.

2. We show that it is not dificult to generate pseudorandomness once we can generate sufficient amount of pseudo-entropy. The crucial tool in this step are extractors. Hence, our task of proving the existence of a pseudorandom generator reduces to that of giving a pseudo-entropy generator.

3. We construct a pseudo-entropy generator from scratch with very weak parameters. This step relies on the Goldreich-Levin lemma and pairwise independent hash functions.

4. We prove that these weak guarantees are sufficient to obtain a full pseudo-entropy generator. This part is fairly simple with respect to a non-uniform notion of security. The uniform proof is discussed separately.

3 Proof

We say a distribution X has *pseudo-entropy* at least k, written $\widetilde{H}_\infty(X) \geq k$, if there exists a distribution Y with min-entropy $H_\infty(Y) \geq k$ such that $X \approx Y$.

A function $F : \{0,1\}^m \to \{0,1\}^n$ is called a *pseudo-entropy generator* (PEG), if there exists a k such that

1. $\widetilde{H}_\infty(F(U_m)) \geq k + n^\epsilon$,

2. $H_\infty(F(U_m)) \leq k$ with probability $1-\epsilon$ for negligible ϵ. More precisely, there is a $Y \subseteq F(U_m)$ with $H_\infty(Y) \leq k$ such that $P_x(F(x) \in Y) \geq 1-\epsilon$.

3.1 From Pseudo-entropy to Pseudorandomness

We will show how to construct a pseudorandom generator from a pseudo-entropy generator. The crucial tool are extractors. For details on extractors the reader is referred to the material from the previous lecture. We quickly recall that a function $\mathrm{EXT} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^l$ is a *strong* (k,ϵ)*-extractor*, if for all distributions X with min-entropy $H_\infty(X) \geq k$, we have

$$|U_d, \mathrm{EXT}(X, U_d) - U_d, U_l| \leq \epsilon.$$

A construction based on the Left-Over-Hashing Lemma achieves $l \geq k - k^{0.01}$ with $d = 2n$ and ϵ exponentially small in k.

Construction Now, assume that we have a pseudo-entropy generator $F : \{0,1\}^m \to \{0,1\}^n$ for some $k < m \leq n$. Also assume we have two extractors $EXT_1 \colon \{0,1\}^m \times \{0,1\}^{2m} \to \{0,1\}^{l_1}$ and $EXT_2 \colon \{0,1\}^n \times \{0,1\}^{2m} \to \{0,1\}^{l_2}$.

Claim 1. The function $G : \{0,1\}^{m+2m+2m} \to \{0,1\}^{4m+l_1+l_2}$, defined as

$$G(x, s_1, s_2) = \mathrm{EXT}_1(x, s_1) \parallel s_1 \parallel \mathrm{EXT}_2(F(x), s_2) \parallel s_2$$

is a pseudorandom generator.

Proof. Obtain the distribution D' from $D = G(U_m, U_{2m}, U'_{2m})$ by replacing the output segment $\mathrm{EXT}_1(U_m, U_{2m})U_{2m}$ with the uniform distribution. Our claim is proven, if we can argue that $D \approx D'$ and $D' \approx U_l$ where $l = 4m + l_1 + l_2$.

To see the first part, suppose we sample x in the following way. Conditioned on $y \sim F(U_m)$, we draw x uniformly at random from the set $F^{-1}(y)$. Notice, by the second condition of a pseudo-entropy generator, for all but a negligible fraction of the y's, we have that $|F^{-1}(y)| \geq 2^{m-k}$. In other words, in this case EXT_1 will be given a sample from a distribution that is uniform on at least $m - k$ bits. Hence, the first extractor provides us with

$$l_1 \geq m - k - (m-k)^{0.01} \geq m - k - n^{\epsilon/100}$$

bits that are statistically close to uniform.

For the second part, we use the first condition of a pseudo-entropy generator. Namely, $F(U_m)$ is computationally indistinguishable from a distribution of min-entropy $k + n^\epsilon$. Hence, the second extractor gives us

$$l_2 \geq k + n^\epsilon - (k + n^\epsilon)^{0.01} \geq k + n^{\epsilon/2}$$

bits that are *computationally* indistinguishable from uniform random bits. □

To complete this section it remains to observe that $l_1 + l_2 > m$. That is, we have constructed pseudorandom generator of nontrivial stretch.

3.2 The Existence of a Weak PEG

We will now relax the requirements of a pseudo-entropy generator and see how to construct such an object. We say $F : \{0,1\}^m \to \{0,1\}^m$ is a *weak pseudo-entropy generator* if there exists a k such that

1. $\widetilde{H}(F(U_m)) \geq k + \frac{1}{100m}$, and,
2. $H(F(U_m)) \leq k$.

Here, H denotes the Shannon entropy, and $\widetilde{H}$ is the "pseudo Shannon entropy" that we define in the same way we defined pseudo-entropy for min entropy.

Now, suppose $f : \{0,1\}^n \to \{0,1\}^n$ is a one-way function and define the function

$$F(x, i, h, r) = f(x) \parallel i \parallel h \parallel h(x) \parallel r \parallel x \odot r.$$

Consider the output distribution of F for random $x, r \in \{0,1\}^n$, $i \in \{1, \ldots, n+10\}$ and a pairwise independent hash function $h : \{0,1\}^n \to \{0,1\}^i$. Technically speaking, F will be given the seed for such a family of hash functions. The seed can have the same length regardless of i. Let Y denote the entire output except for the last bit which we denote by Z. Suppose we modify the distribution Z as follows.

$$Z' = \begin{cases} \text{random bit} & \text{if } i = i^* \\ x \odot r & \text{otherwise} \end{cases}$$

where $i^* = \lceil \log(|f^{-1}(w)|) \rceil + 1$. We will use the distribution YZ' to argue that F is in fact a weak pseudo-entropy generator.

Claim 2.

$$H(YZ') \geq H(YZ) + \frac{1}{2m}$$

Proof. We have $H(YZ') = H(Y) + H(Z|Y)$ and $H(YZ) = H(Y) + H(Z|Y)$. So, it suffices to prove $H(Z'|Y) \geq H(Z|Y) + \frac{1}{2m}$. Clearly, whenever $i \neq i^*$, we have $Z = Z'$. However, the case $i = i^*$ occurs with probability $\frac{1}{m}$ and here we will show that Z has conditional entropy zero with probability 1/2. Condition on $w \sim f(U_n)$ and let $x \in S = f^{-1}(w)$. We claim with probability 1/2 over random h, $h(x)$ determines x. Indeed, since $|S| \leq 2^{i^*-1}$, we can bound the collision probability of h as

$$P(\exists x' \in S \backslash \{x\} \colon h(x) = h(x')) \leq \frac{2^{i^*-1}}{2^{i^*}} \leq \frac{1}{2}. \qquad \square$$

Claim 3.

$$YZ \approx YZ'$$

Proof. Suppose otherwise. Then, we have a distinguishing algorithm A such that

$$A(f(x), i^*, h, h(x), r) = x \odot r$$

with probability $\frac{1}{2} + \epsilon$. We want to use this algorithm to invert f using the Goldreich-Levin algorithm. The string i^* we can guess correctly with probability $1/n$. The same is not immediately clear for $h(x)$. However, since h is collision resistant, we can argue that the first $\log(|f^{-1}(w)|) - \log(1/\epsilon)$ bits of $h(x)$ are close to uniform. The remaining bits can be brute-forced efficiently. □

3.3 PEGs from weak PEGs

It remains to show how to construct a pseudo-entropy generator from a weak one. In this step we will rely on our non-uniform notion of security. The uniform case is treated separately in the following section.

Let $F\colon \{0,1\}^m \to \{0,1\}^n$ denote a weak pseudo-entropy generator and consider the function

$$F'(x_1 \dots x_l) = F(x_1) \parallel F(x_2) \parallel \cdots \parallel F(x_l)$$

where l is of order, say, n^{10}. In order to show that F' is a (strong) pseudo-entropy generator, we prove the following two claims.

Claim 4. For all except a negligible fraction of the strings $\bar{y}$ in the image of F', we have

$$2^{-kl(1+\epsilon)} \leq P(F'(x_1, \dots, x_l) = \bar{y}) \leq 2^{-kl(1-\epsilon)}$$

where each x_i is drawn uniformly and independently at random.

Proof. Notice, by assumption,

$$H(F(U_m)) = E_{y\sim\mu}(\log(1/\mu(y))) = k.$$

where $\mu = F(U_m)$. Hence,

$$\begin{aligned} & P_{\bar{y}=y_1,\ldots,y_l}\left(2^{-kl(1+\epsilon)} \leq \prod_{i=1}^{l} \mu(y_i) \leq 2^{-kl(1-\epsilon)}\right) \\ & = 1 - P\left(\left|\sum_{i=1}^{l} \log(1/\mu(y_i)) - kl\right| > \epsilon kl\right) \\ & \geq 1 - 2^{-\Omega(\epsilon^2 kl/n)}. \end{aligned}$$

Here, we used that $\log(1/\mu(y_i)) \leq n$, since $y_i \in F(U_m)$. □

Claim 5. If two distributions X, Y satisfy $X \approx Y$, then $X_1 \parallel \cdots \parallel X_l \approx Y_1 \parallel \cdots \parallel Y_l$ where X_i and Y_i denote independent copies of X and Y, respectively.

Proof. Consider the following sequences.

$$\begin{gathered} X_1 \parallel X_2 \parallel X_3 \parallel \cdots \parallel X_l \\ Y_1 \parallel X_2 \parallel X_3 \parallel \cdots \parallel X_l \\ Y_1 \parallel Y_2 \parallel X_3 \parallel \cdots \parallel X_l \\ \vdots \\ Y_1 \parallel Y_2 \parallel Y_3 \parallel \ldots \parallel Y_l \end{gathered}$$

Suppose a circuit can distinguish two subsequent distributions, say the i-th and $(i+1)$-th. Then, by an averaging argument, we can fix the values for $Y_1, \ldots, Y_{i-1}$ and $X_{i+1}, \ldots, X_l$ such that the circuit actually distinguishes between X_i and Y_i. In fact, we can hardwire these values into the circuit so as to obtain a new circuit that distinguishes between X and Y. □

Given these two claims, the proof is easily completed. By assumption there is a distribution $Y \approx F(U_n)$ with $H(Y) \geq k + \frac{1}{100m}$. Hence, by Claim 5, $F'(U_{lm})$ is indistinguishable from a distribution with *Shannon* entropy $kl + \frac{l}{100m}$. Still, we need a bound on the pseudo min-entropy of F'. But, by Claim 4, we have for most $\bar{y}$ that $P(F' = \bar{y}) \leq 2^{-(k+1/100m)l(1-l^{0.1})}$. We simple remove every element $\bar{y}$ for which this is not true from the support of F'. For large enough l this will give us the required entropy.

The second requirement of a pseudo-entropy generator follows directly from Claim 4.

Remark 1 In the proof of Claim 5, we used the fact that we defined security with respect to non-uniform circuits. The claim is actually false in general when it comes to uniform security. However, Holenstein [Hol06] showed how one can prove the security of our construction using what is known as a *uniform hardcore lemma.* This lemma will be stated and proven next.

4 Uniform Hardcore Lemma

In this section, we sketch the proof of a uniform hard-core lemma. Our presentation is based on a recent approach due to Kale [Kal07].

To state the theorem, we say a distribution $\mathbf{p}$ over a set $S \subseteq \{0,1\}^n$ is *ϵ-smooth* if for every $x \in S$, we have $\mathbf{p}(x) \leq \frac{1}{\epsilon|S|}$. For instance, a distribution uniform over a subset of S of size $\epsilon|S|$ is is ϵ-smooth with respect to this definition. Also, notice the set of ϵ-smooth distributions is convex.

Theorem 2 *Let $\{f_n\}_{n\in\mathbb{N}}$ denote a family of boolean function and let $\epsilon, \gamma \colon \mathbb{N} \to (0,1)$. Suppose, there exists an algorithm A which given oracle access to any ϵ-smooth distribution $\mathbf{p}$ over $\{0,1\}^n$ returns a circuit C of size at most $s(n)$ such that $\Pr_{x\sim\mathbf{p}}[C(x) = f_n(x)] \geq \frac{1}{2} + \gamma(n)$.*

Then there is an algorithm B which for every n and oracle access to f_n returns a circuit C' such that C' computes f_n correctly on at least a $1-\epsilon(n)$ fraction of all inputs. Furthermore, the algorithm B makes $O(\frac{\ln(1/\epsilon)}{\gamma^2})$ calls to A and its runtime is polynomial in the number of calls to A and the cost of simulating A. Also, the circuit C' is the majority circuit of the circuits returned by A.

For details on how to apply such a theorem in the above context, we refer the reader to the paper of Holenstein [Hol06].

Before we can prove this theorem, we will need some prelimanaries. Specifically, we will be interested in Bregman projections. These are projections of probability distributions onto convex sets with respect to the relative entropy as distance function. We shall think of discrete distributions over a finite set X of cardinality N as real-valued vectors $\mathbf{p}, \mathbf{q}$ indexed by the set X. For two distributions $\mathbf{p}, \mathbf{q}$, define the *relative entropy* as

$$\mathrm{D}(\mathbf{p} \,||\, \mathbf{q}) = \sum_{x\in X} p_x \log \frac{p_x}{q_x}.$$

Further, let $\Gamma \subseteq \mathbb{R}^N$ be a closed convex set. The *Bregman projection* of $\mathbf{q}$ onto the set Γ is defined as

$$\mathrm{proj}_\Gamma \, \mathbf{q} = \arg\min_{\mathbf{p}\in\Gamma} \mathrm{D}(\mathbf{p} \,||\, \mathbf{q}).$$

Bregman showed that the projection associated with any divergence satisfies a generalized Pythagorean Theorem. We will only need this theorem in the case of relative entropy.

Theorem 3 (Bregman) *Let $\mathbf{p}, \mathbf{q}$ be two distributions and let $\Gamma \subseteq \mathbb{R}^N$ be a closed convex set. Then,*

$$\mathrm{D}(\mathbf{p} \,||\, \mathrm{proj}_\Gamma \, \mathbf{q}) \leq \mathrm{D}(\mathbf{p} \,||\, \mathbf{q}). \tag{1}$$

Notice, the Bregman projection is defined by a convex program and hence it can be computed (or approximated) efficiently when given some suitable representation of theset Γ (see [Kal07] for further references).

Figure 1: Uniform Smooth Boosting

Input: Oracle access to a boolean function $f\colon \{0,1\}^n \to \{0,1\}$; parameters $\epsilon, \gamma > 0$ and $T \in \mathbb{N}$; an algorithm A which, when given oracle access to a smooth distribution $\mathbf{p}$ over $\{0,1\}^n$, returns a circuit C such that $\Pr_{x\sim\mathbf{p}}[C(x) = f_n(x)] \geq \frac{1}{2} + \gamma$.
For $t = 1, 2, \ldots, T$:

1. Run algorithm A on input of $\mathbf{p}^{(t)}$ so as to obtain a circuit $C^{(t)}$ where $\mathbf{p}^{(1)}$ denotes the uniform distribution over $\{0,1\}^n$. When algorithm A asks a query x, compute and return $\mathbf{p}^{(t)}(x)$.
2. Define $\mathbf{m}^{(t)}$ by putting $m_x^{(t)} = 1$ if $C^{(t)}(x) = f(x)$ and 0 otherwise.
3. Define $\mathbf{q}^{(t+1)}$ coordinate-wise using the following update rule $q_x^{(t+1)} = \frac{(1-\gamma)^{m_x^{(t)}}}{Z^{(t)}} p_x^{(t)}$ where $Z^{(t)} = \sum_x p_x^{(t)} (1-\gamma)^{m_x^{(t)}}$ is the normalization factor.
4. Define $\mathbf{p}^{(t+1)} = \text{proj}_\Gamma\, \mathbf{q}^{(t+1)}$ where Γ denotes the set of ϵ-smooth distributions.

Output: The circuit $C' = \text{MAJORITY}(C^{(1)}, C^{(2)}, \ldots, C^{(T)})$.

Proof. To prove the theorem, we will run the algorithm depicted in Figure 1 for $T = \frac{2}{\gamma^2}\ln(1/\epsilon) + 1$ rounds. The runtime requirement follows from the observation that Step 1 in the t-th round can be implemented efficiently using $O(t)$ applications of the update rule described in Step 2–4. The projection in (4) cannot be computed exactly efficiently, but it is possible to approximate the projection accurately enough. We omit the details of this step.

Next, we claim that the circuit C' computes f_n correctly on a $1-\epsilon$ fraction of the inputs. To argue this point we will appeal to the following lemma of Kale [Kal07]. The proof uses standard techniques from learning theory combined with Bregman's theorem. We omit it from these scribe notes.

Lemma 6 *Our algorithm achieves the following bound for every fixed distribution* $\mathbf{p}$*,*

$$\sum_{t=1}^{T} \langle \mathbf{m}^{(t)}, \mathbf{p}^{(t)} \rangle \leq (1+\delta) \sum_{t=1}^{T} \langle \mathbf{m}^{(t)}, \mathbf{p} \rangle + \frac{1}{\delta} \mathrm{D}(\mathbf{p} \,||\, \mathbf{p}^{(1)}). \tag{2}$$

We will apply the lemma as follows.

First notice, it follows from our assumption that

$$\sum_{t=1}^{T} \langle \mathbf{m}^{(t)}, \mathbf{p}^{(t)} \rangle = \sum_{t=1}^{T} \Pr_{x\sim\mathbf{p}^{(t)}} [C^{(t)}(x) = f(x)] \geq \Big(\frac{1}{2} + \gamma\Big) T. \tag{3}$$

Now let us consider the set $E = \{x \in \{0,1\}^n \mid C(x) \neq f(x)\}$, i.e., those points on which the majority circuit C' errs. Notice that $\sum_{t=1}^{T} m_x^{(t)}$ is equal to the number of circuits $C^{(t)}$ which correctly compute f_n on input x. Since C' is a majority circuit, we have $\sum_{t=1}^{T} m_x^{(t)} \leq \frac{1}{2}T$ whenever $x \in E$. In particular, this is true in expectation taken over uniform $x \in E$. Hence, if we let $\mathbf{u}_E$ denote the uniform distribution over the set E, then we must have,

$$\sum_{t=1}^{T} \langle \mathbf{m}^{(t)}, \mathbf{u}_E \rangle = \mathop{\mathbf{E}}_{x \sim \mathbf{u}_E} \left[\sum_{t=1}^{T} m_x^{(t)} \right] \leq \frac{1}{2}T \tag{4}$$

Now, suppose $|E| > \epsilon 2^n$. This implies that $\mathbf{u}_E$ is an ϵ-smooth distribution. Moreover, $\mathrm{D}(\mathbf{u}_E \,||\, \mathbf{p}^{(1)}) \leq \ln(1/\epsilon)$. Thus, we can apply Lemma 6 to (3) and (4) with $\delta = \gamma$ and $\mathbf{p} = \mathbf{u}_E$ to conclude

$$\Big(\frac{1}{2} + \gamma\Big)T \leq \Big(\frac{1}{2} + \frac{\gamma}{2}\Big)T + \frac{1}{\gamma}\ln(1/\epsilon). \tag{5}$$

In particular,

$$T \leq \frac{2}{\gamma^2}\ln(1/\epsilon).$$

So, if we run our algorithm for $T \geq \frac{2}{\gamma^2}\ln(1/\epsilon) + 1$ rounds, then we can be sure that $|E|/2^n < \epsilon$. □

References

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

[Hol06] Thomas Holenstein. Pseudorandom generators from one-way functions: A simple construction for any hardness. In Shai Halevi and Tal Rabin, editors, *TCC*, volume 3876 of *Lecture Notes in Computer Science*, pages 443–461. Springer, 2006.

[IL89] Russell Impagliazzo and Michael Luby. One-way functions are essential for complexity based cryptography (extended abstract). In *FOCS*, pages 230–235. IEEE, 1989.

[ILL89] Russell Impagliazzo, Leonid A. Levin, and Michael Luby. Pseudo-random generation from one-way functions (extended abstracts). In *STOC*, pages 12–24. ACM, 1989.

[Kal07] Satyen Kale. Boosting and hard-core set constructions: a simplified approach. *Electronic Colloquium on Computational Complexity (ECCC)*, (131), 2007.