Pseudorandom generators from general one-way functions

15-859I
Spring 2003

Review:

- A function G : $\{0,1\}^n \rightarrow \{0,1\}^{\ell(n)}$ is a pseudorandom generator (PRG): if
  - G is polynomial time computable;
  - $\ell(n) > n$ for all n ; and
  - For all PPTM A,
    $| \Pr[A(G(U_n)) = 1] - \Pr[A(U_{\ell(n)}) = 1] |$
    is negligible.
- A PRG stretches a short random string into a longer string which is indistinguishable from a random string of the same length.

Review

- Recall the Goldreich-Levin theorem:
  - if f is a one-way function, then x•r is a hard-core bit for g(x,r) = (f(x),r)
  - A hard-core bit for f is a function b such that for any PPT A, $|\Pr_{x \leftarrow U_n}[A(f(x)) = b(x)] - ½|$ is negligible
- This leads to simple construction of a PRG from a OWP f: G(x,r) = (f(x),r, x•r)
- What if f is one-way, but not a permutation?

Very General Outline

- Proof has three steps:

1. Show how to construct a “False Entropy Generator” from a general OWF
2. Show how to construct a “Pseudoentropy Generator” from a False Entropy Generator
3. Show how to construct a Pseudorandom generator from a pseudoentropy generator

- We will work backwards.

Review: Entropy

- The Shannon Entropy H(X) of a random variable X with support [X] is:
  - $H(X) = \Sigma_x -\Pr[X=x] \log \Pr[X=x]$
- The Renyi entropy $H_R(D)$ is the log collision probability of D:
  - $H_R(X) = - \log \Pr_{X,Y \leftarrow D}[X=Y]$
- The statistical difference between two distributions E and D on set S is defined by:
  $L_1(D,E) = ½ \Sigma_{x \in S} |\Pr[D=x] - \Pr[E=x]|$
- If $L_1(D,E)$ is negligible, D and E are computationally indistinguishable.

Review: Universal Hashing

- A family h : $\{0,1\}^{p(n)} \times \{0,1\}^n \rightarrow \{0,1\}^{m(n)}$ is a *universal hash function* if $\forall x \neq x' \in \{0,1\}^n$, $a,a' \in \{0,1\}^{m(n)}$,
  $\Pr_Y[h_Y(x) = a \text{ AND } h_Y(x')=a'] = 1/2^{2m(n)}$
  where $Y \leftarrow U_{p(n)}$
- Can be built by linear functions over GF(2):
  - h : $\{0,1\}^{(n+1)m(n)} \times \{0,1\}^n \rightarrow \{0,1\}^{m(n)}$ defined by
  - $h_Y(x) = (x,1) \cdot y$

From Renyi to Shannon entropy: The “Leftover Hash Lemma”

Let:

- D_n:$\{0,1\}^n$ have $H_R(D_n) \geq m(n)$
- $e(n)$ be a positive integer
- $h : \{0,1\}^{l(n)} \times \{0,1\}^n \rightarrow \{0,1\}^{m(n)-2e(n)}$ be a universal hash function
- $X \leftarrow D_n$, $Y \leftarrow U_{l(n)}$, $Z \leftarrow U_{m(n)-2e(n)}$. then $L_1((h_Y(X), Y) , (Z, Y)) \leq 2^{-(e(n)+1)}$
- Proof: 15-855 Exercise 5.5

From Shannon to Renyi Entropy: Product distributions

- Let D:$\{0,1\}^n$ be a probability ensemble, and let $k(n)$ be integer-valued and poly(n) bounded. Then there exists an ensemble E:$\{0,1\}^{nk(n)}$ with:
  - $H_R(E) \geq k(n)H(D) – n(k(n))^{2/3}$
  - $L_1(D^{k(n)},E) \leq 2^{-k(n)^{1/3}}$
- Proof: [Shannon, 48]
- Applies to conditional entropy as well…

Pseudoentropy Generators

- $f : \{0,1\}^{t(n)} \rightarrow \{0,1\}^{\ell(n)}$ has *computational entropy* $s(n)$ if $\exists$ ptc f':$\{0,1\}^{m(n)} \rightarrow \{0,1\}^{\ell(n)}$:
  - $f(U_{t(n)})$ and $f'(U_{m(n)})$ are computationally indistinguishable; and
  - $H(f'(U_{m(n)}) \geq s(n)$.
- f is a *pseudoentropy generator* with *pseudoentropy* $s(n)$ if $f(U_{t(n)})$ has computational entropy $t(n)+s(n)$

Pseudoentropy generator to PRG

- Let f: $\{0,1\}^n \rightarrow \{0,1\}^{m(n)}$ be a pseudoentropy generator with pseudoentropy s(n).
- We have a function where the output is computationally indistinguishable from a distribution with more entropy… but it doesn’t look uniform.
- Solution: Extract the entropy with a hash function.

Pseudoentropy generator to PRG

- Let f: $\{0,1\}^n \rightarrow \{0,1\}^{m(n)}$ be a pseudoentropy generator with pseudoentropy s(n).
- Let $k(n) = ((2m(n) +1)/s(n))^3$
- Let $j(n) = k(n)(n + s(n)) – 2m(n)k(n)^{2/3}$
- Let $h : \{0,1\}^{p(n)} \times \{0,1\}^{k(n)m(n)} \rightarrow \{0,1\}^{j(n)}$ be a universal hash family
- Define $g : \{0,1\}^{p(n)} \times \{0,1\}^{nk(n)} \rightarrow \{0,1\}^{p(n)+j(n)}$ by $g(Y,X) = (h_Y(f^{k(n)}(X)),Y)$

Theorem

- g as above is a pseudorandom generator.
- Proof:
  - Let $f' : \{0,1\}^{n'(n)} \rightarrow \{0,1\}^{m(n)}$ be the function with $H(f'(U_{n'(n)}) > n + s(n)$ such that $f(U_n)$ and $f'(U_{n'(n)})$ are computationally indistinguishable.
  - Then $f^{k(n)}(U)$ and $f'^{k(n)}(U)$ are computationally indistinguishable.
  - Also, $(h_Y(f^{k(n)}(U)), Y)$ and $(h_Y(f'^{k(n)}(U), Y)$ are computationally indistinguishable.

Proof…

- So if we can show that $(h_Y(f'^{k(n)}(U)), Y)$ and $(U_{j(n)},Y)$ are computationally indistinguishable, we are done.
- Note: $f'^{k(n)}(U) \geq n + s(n)$.
- So, by Shannon-to-Renyi theorem, there exists an $E:\{0,1\}^{m(n)k(n)}$ with

$$\begin{aligned} H_R(E) &\geq k(n)(n + s(n)) - m(n)k(n)^{2/3} \\ &= j(n) + m(n)k(n)^{2/3}\text{; and} \\ L_1(E,f'^{k(n)}(U)) &\leq 2^{-k(n)^{1/3}} \end{aligned}$$

- Note that by the Leftover Hash Lemma,

$$L_1((h_Y(E),Y),(U_{j(n)},Y)) \leq 2^{1-(m(n)k(n)^{2/3}/2)}$$

So $L_1(U_{j(n)+p(n)},(h_Y(f'^{k(n)}(U)),Y)) \leq 2^{-k(n)^{1/3} + 1}$. QED.

Summary: PEG → PRG

- We can take a distribution that is computationally indistinguishable from a distribution with more entropy than its input, and make it into a PRG.
- We only need to use a hash function, and a product distribution to get "enough" Renyi entropy.
- This might be an idea that will come up later….

False Entropy Generators

- Recall, $f : \{0,1\}^{t(n)} \to \{0,1\}^{\ell(n)}$ is a pseudoentropy generator if $f(U_{t(n)})$ has computational entropy $> t(n)$.
- We say that f is a *false entropy generator* if $f(U_{t(n)})$ has computational entropy $> H(f(U_{t(n)}))$.
- f has false entropy s(n) if it has computational entropy at least $H(f(U_{t(n)})) + s(n)$.
- Notice that the computational entropy of a false entropy generator might not be greater than the entropy of its seed.

Degeneracy

- Let $f : \{0,1\}^n \to \{0,1\}^{\ell(n)}$
- The *degeneracy* of f is the information loss of f: $D_n(f) = H(X) - H(f(X)) = H(X \mid f(X))$
- The *approximate degeneracy* of f at $z = f(x)$ is

$$\tilde{\mathbf{D}}_f(z) = \lceil \log(|\{x' : f(x') = z\}|) \rceil$$

- Note that $|E_X[\tilde{\mathbf{D}}_f(X)] - D_n(f)| \leq 1$
- If f(X) has false entropy s(n), then if we could output $\tilde{\mathbf{D}}_f(X)$ extra bits of X, we would have pseudoentropy s(n).

Hmm…

Problem with this idea:

- How to calculate $d=\tilde{\mathbf{D}}_f(X)$?

  f could be highly nonregular, so that d has high variance and is hard to even guess with high accuracy
- But we do know that whp,

$$|\tilde{\mathbf{D}}_{fk(n)}(f^{k(n)}(U)) - k(n)D_n(f)| \leq k(n)$$

- So if we knew $H(f(U_n))$, we could extract at least $k(n)(n - \lceil H(f(U_n)) \rceil)$ bits of $X^{k(n)}$

FEG → PEG Theorem

- Let $f: \{0,1\}^n \to \{0,1\}^{\ell(n)}$ have false entropy s(n)
- Suppose we know e_n such that

$$|H(f(U_n)) - e_n| \leq s(n)/8$$

- Let $k(n) = \lceil (4n/s(n))^3 \rceil$,

$$j(n) = \lceil k(n)(n-e_n) - nk(n)^{2/3} \rceil$$

- Let $h : \{0,1\}^{p(n)} \times \{0,1\}^{nk(n)} \to \{0,1\}^{j(n)}$ be a universal hash function.
- Define $g(e_n,u,r) = (f^{k(n)}(u), h_r(u), r)$
- Then, g has nonuniform pseudoentropy 1.

FEG → PEG Proof

- $f: \{0,1\}^n \rightarrow \{0,1\}^{\ell(n)}$ has false entropy $s(n) \Rightarrow \exists$ $D:\{0,1\}^{\ell(n)} \cong f(X)$, with $H(D) \geq H(f(X)) + s(n)$
- Then, $(D^{k(n)},U_{j(n)},r) \cong (f^{k(n)}(X^{k(n)}),U_{j(n)},r)$ by hybrid argument.
- And since $H(X|f(X)) = n - H(f(X)) = n - e_n$

  $$L_1((f^{k(n)}(x),U_{j(n)},r), g(e_n,x,r)) \leq 2^{-k(n)^{1/3}}$$

  By the conditional version of the S-to-R thm,

  $$(D^{k(n)},U_{j(n)},r) \cong g(e_n,x,r)$$

FEG → PEG Proof

- How much entropy is in $D' = (D^{k(n)},U_{j(n)},r)$?

  $$\begin{aligned} H(D') &\geq k(n)(H(f(X)) + s(n))+j(n)+p(n) \\ &\geq k(n)(e_n + \tfrac{7}{8}s(n)) +j(n) +p(n) \\ &= \tfrac{7}{8}ks(n)+ke_n + k(n-e_n)-nk^{2/3} + p(n) \\ &= nk(n) + p(n) + \tfrac{7}{8}k(n)s(n) - nk(n)^{2/3} \\ &= H(x,r) + 40n^3/s(n)^2 \geq H(x,r) +1 \end{aligned}$$

- So with the correct value of e_n, $g(e_n,x,r)$ has pseudoentropy 1, QED.

FEG → PRG

- If we have the correct value of e_n, we can compose our theorems to get a PRG from a FEG
- Need to remove this "mild" nonuniformity
- Let let $k(n) < poly(n)$, and let $g: \{0,1\}^{\lceil \log k(n) \rceil} \times \{0,1\}^n \rightarrow \{0,1\}^{\ell(n)}$ be such that $g(a_n, \cdot)$ is a PRG, with $\ell(n) > nk(n)$
- Define $g' : \{0,1\}^{n \times k(n)} \rightarrow \{0,1\}^{\ell(n)}$ by

  $$g'(x') = \oplus_i g(i,x_i')$$

Nonuniform PRG theorem

- Theorem: g' is a PRG.
- Proof: Suppose we have PPTM A with

  $$|\Pr[A(g'(x')) = 1] - \Pr[A(U_{\ell(n)})=1]| = p(n)$$

- We construct PPTM A' with

  $$|\Pr[A'(g(a_n,x))=1] - \Pr[A'(U_{\ell(n)})=1]| = p(n)$$

- $A'(u) =$ choose $x_1,\ldots,x_{k(n)} \leftarrow U_n$.

  Output $A(u \oplus_{i \neq a_n} g(i,x_i))$.
- So if $g(a_n,\cdot)$ is a PRG, so is g'.

Recap

- To turn a PEG into a PRG, we take a product distribution and output the universal hash of the PEG output.
- To turn a FEG into a (mildly nonuniform) PEG, we take a product distribution and output the universal hash of the FEG input.
- To turn a mildly nonuniform PRG into a uniform PRG, take XOR over all possible advice strings.
- Next lecture: OWF → FEG.