Pseudorandom generators from general one-way functions II

15-859I
Spring 2003

Review:

- Our goal is to construct a PRG from any OWF
- A False Entropy Generator is a function $f:\{0,1\}^n \to \{0,1\}^{\ell(n)}$ that has $f(U_n)$ computationally indistinguishable from some ptc ensemble D_n: $\{0,1\}^{\ell(n)}$ where $H(D) > H(f(U))$.
- Using universal hash functions and product distributions, we can construct a PRG from a F.E.G. (4 pages from [HILL99])

Today – Next Monday

- How do we construct a F.E.G. from a OWF? (10 pages from [HILL99])

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)
- i.e., when f(x) is a permutation, x•r gives us 1 bit of computational entropy…

Idea

- Why not construct a false entropy generator from f in the same way, ie g(x,r) = (f(x),r,x•r)
- Then the p-time distribution (f(x),r,b) with uniform bit b is indistinguishable from g(x,r), but…
- Problem: suppose f is one-way. Construct f'(x,y) = f(x). Then g(x,y,r) is statistically indistinguishable from (f(x,y),r,b)… So H(f(x,y),r,b) = H(g(x,y,r)), and g is not a false entropy generator.

Idea

- If we could somehow force f to be one-to-one, we would be OK, since then H(g(x,r)) < H(f(x),r,b).
- HW: if we could compute $r_f(x)$, then f'(x) = f(x),r_f(x) is one-to-one and one-way
- Since r(x) is hard to compute, replace it by

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

  bits of h(x), for h a universal hash function.

Theorem

- Let $f : \{0,1\}^n \to \{0,1\}^{\ell(n)}$ be ptc and suppose that $\tilde{\mathbf{D}}_f$ is ptc. Let $h:\{0,1\}^{p(n)}\times\{0,1\}^n\to\{0,1\}^{n+2}$ be a universal hash function. Define

  $$f'(x,y) = (f(x),h_y(x)|_{1\ldots\tilde{\mathbf{D}}f(f(x))+2},y)$$

- Theorem:

1. if f is one-way then f' is one-way
2. Let $X\leftarrow U_n$, $Y\leftarrow U_{p(n)}$. Then

   $$H(f'(X,Y)) \geq n + p(n) - \tfrac{1}{2}.$$

Proof of (1)

- Suppose we have A which inverts f'(X,Y). Define the algorithm M^A as follows:
- $M^A(z)$ = on input z = f(x):
  Compute $d=\tilde{\mathbf{D}}_f(z)$
  Choose $a \leftarrow U_{d+2}$
  Choose $y \leftarrow U_{p(n)}$.
  Let x' = A(z,a,y). If f(x') = z output x'.
- Lemma: if A inverts f'(X,Y) with probability $\delta(n)$ then M^A inverts f with probability at least $\delta(n)^3/128$.

Proof of lemma

- Define the conditional random variable

  $$B_{z,y} \leftarrow h_y(W \in_U \{ w : f(w) = z \})|_{1\ldots\tilde{\mathbf{D}}f(z)-j(n)}$$

  where $j(n) = 2\lceil \log(2/\delta(n))\rceil$.
  The probability, over X,Y, and $B_{f(X),Y}$, that there is a γ such that $A(f(X),(B_{f(X),Y},\gamma),Y)$ inverts f is at least the probability that A inverts f'(X,Y), or $\delta(n)$

Lemma, cont…

- Now for a fixed z, define the random variable

  $$B_z \leftarrow U_{d(z)+2}.$$

  Notice that in $B_{z,y}$, $H_R(W|z) = d(z)$.
  Thus, by the leftover hash lemma, we get:

  $$L_1((B_{z,Y},Y), (B_z,Y)) \leq 2^{-\log 2/\delta(n)} = \delta(n)/2.$$

  And, since for any predicate P,

  $$|Pr[P(X) = 1] - Pr[P(Y) = 1]| \leq L_1(X,Y),$$

  It must be the case that

  $$Pr_{X,Y,B}[\exists\gamma{:}A(f(X),(B_{f(X)},\gamma),Y) \text{ inverts } f] \geq \delta(n)-\delta(n)/2$$

Lemma, cont

- When we are so lucky in the choice of X,Y, and $B_{f(X)}$, then if we choose a $\gamma \leftarrow U_{j(n)+2}$ we will hit the correct γ with probability

  $$2^{-(j(n)+2)} = 2^{-2-2\lceil\log(2/\delta(n))\rceil}$$
  $$\geq \tfrac{1}{4}\, 2^{-2(\log(2/\delta(n))+1)}$$
  $$= (\delta(n)/2)^2/16 = \delta(n)^2/64$$

- Since the probability of getting lucky is at least $\delta(n)/2$, this gives us probability at least $\delta(n)^3/128$ of inverting f, as claimed.

Theorem

- Let $f : \{0,1\}^n \to \{0,1\}^{\ell(n)}$ be ptc and suppose that $\tilde{\mathbf{D}}_f$ is ptc. Let $h:\{0,1\}^{p(n)}\times\{0,1\}^n\to\{0,1\}^{n+2}$ be a universal hash function. Define

  $$f'(x,y) = (f(x),h_y(x)|_{1\ldots\tilde{\mathbf{D}}f(f(x))+2},y)$$

- Theorem:

1. if f is one-way then f' is one-way ✓
2. Let $X\leftarrow U_n$, $Y\leftarrow U_{p(n)}$. Then

   $$H(f'(X,Y)) \geq n + p(n) - \tfrac{1}{2}.$$

Proof of (2)

- Fix z, and let $x \neq x'$ satisfy $f(x) = f(x') = z$. By the universality of h,
  $\Pr[h_Y(x) = h_Y(x')|_{1\ldots d(z)+2}] = 2^{-(d(z)+2)}$
  $\leq 1/(4|\{x:f(x)=z\}|)$
- $\Pr[f'(X,Y) = f'(X',Y')] \leq$
  $2^{-(n+p(n))}+2^{-p(n)}\Sigma_x \Sigma_{x' \in f^{-1}(x)} 1/(4\,|f^{-1}(x)|)\Pr[X=x, X'=x']$
  $\leq 5/4\ 2^{-(n+p(n))}$
- So $H(f'(X,Y)) \geq H_R(f'(X,Y)) \geq -\log(5/4\ 2^{-(n+p(n))})$
  $\geq n + p(n) + 2 - \log(5)$ ✓

Corollary

- If $f:\{0,1\}^n \to \{0,1\}^{\ell(n)}$ is one-way and $\tilde{\mathbf{D}}_f$ is ptc, and f' is defined as above, then $g(x,y,r) = (f'(x,y),r,x\bullet r)$ is a pseudoentropy generator with pseudoentropy ½.
- Proof: It follows from (HW) that
  $(f'(X,Y),R, X\bullet R) \cong (f'(X,Y),R,U_1)$
  But $H(f'(X,Y),R,U_1) = H(f'(X,Y))+ H(R) +H(U_1)$
  $\geq 2n + p(n) + ½,$
  while $H(X,Y,R) = 2n + p(n)$. QED.

More Corollaries

- Corollary: If there is a OWF f with ptc $\tilde{\mathbf{D}}_f$ then there is a pseudorandom generator.
  - Proof: Compose previous theorem with results of previous class.
- Corollary: If there is a s(n)-regular OWF f for s(n) a ptc function, then there is a pseudorandom generator.
  - Proof: $\tilde{\mathbf{D}}_f(f(x)) = \lceil \log s(n) \rceil$. So it is ptc. Apply previous corollary.

The problem here…

- How to compute $\tilde{\mathbf{D}}_f$? No reason in general to assume it is ptc.
- Basic approach: $0 \leq d = \tilde{\mathbf{D}}_f(z) < n$ for all z. So if we take a uniform guess for d, we will be right with probability 1/n.
- If we take enough inputs to f', and enough guesses for d, we should on average get about 1/n bit of false entropy per copy.
- Proving it is the hard part… (2pp down, 8 to go)

A new construction.

- Let $f:\{0,1\}^n \to \{0,1\}^{\ell(n)}$ be a one-way function, and let $h: \{0,1\}^{p(n)} \times \{0,1\}^n \to \{0,1\}^{n+\lceil \log 2n \rceil}$ be a universal hash function. Define
  $f'(x,i,r) = (f(x),h_r(x)|_{1\ldots i+\lceil \log 2n \rceil}, i, r)$
- Let $Y \leftarrow U_n$, then when $I < \tilde{\mathbf{D}}_f(f(X))$, we will have $(f'(X,I,R),Y,X\bullet Y) \cong (f'(X,I,R),Y,U_1)$.
- To formalize, define two sets:
  - $T = \{(x,i) : x \in \{0,1\}^n, i \in \{0,\ldots, \tilde{\mathbf{D}}_f(f(x))\}\}$
  - $T^C = \{(x,i) : x \in \{0,1\}^n, i \in \{\tilde{\mathbf{D}}_f(f(x))+1, \ldots,n-1\}\}$

Lemma

- Lemma: Let $W = (X,I) \in_U T$, $R \leftarrow U_{p(n)}$, $Y \leftarrow U_n$ and $B \leftarrow U_1$. Then:
  $(f'(W,R),X\bullet Y,Y) \cong (f'(W,R),B,Y)$
- Proof: By (HW), if there exists an adversary A distinguishing $X\bullet Y$ from B with probability δ, then there is an adversary A' predicting $X\bullet Y$ from $f'(W,R)$ with advantage δ.
  By the Goldreich-Levin theorem, If A can predict $X\bullet Y$ given $f'(W,R)$ with advantage δ, there is a machine M^A which can invert $f'(W,R)$ with probability $(\delta^2/2)$

Proof, con't.

- So we need to show that given M^A for inverting f'(W,R), we can construct N^A which inverts f(X).
- $N^A(f(x))$ =
  - Choose $i \in_U \{0,\ldots,n-1\}$, $r \leftarrow U_{p(n)}$, $a \leftarrow U_{i+\lceil \log 2n \rceil}$
  - Return $M^A(f(x),a,i,r)$
- What is $Pr[f(N^A(f(x))) = f(x)]$?
- First, notice that $Pr[(x,i) \in T] \geq 1/n$. So

  $Pr[f(N^A(f(x))) = f(x)] \geq$

  $1/n \; Pr[M^A(f(x),a,i,r) \text{ inverts } f \text{ when } (x,i) \in T]$

Proof, con't, 2

- But now we can use the same techniques as for the previous theorem, to show that

  $Pr[M^A(f(x),a,i,r) \in f^{-1}(f(x)) : (x,i) \in T] \geq$

  $1/4n \; (Pr[M^A(f'(W,R)) \text{ inverts } f'])^3$.
- So, if A distinguishes X•Y from B given (f'(W,R), Y) with probability δ, then N^A inverts f(X) with probability at least $1/4n^2 \; (\delta^2/2)^3 = 1/\text{poly}(n)$ when $\delta = 1/\text{poly}(n)$.
- i.e., if f is one-way, then $(f'(W,R),X\bullet Y,Y) \cong (f'(W,R),B,Y)$, Q.E.D.

Recap

- We have that if f is one-way then when we guess $i \leq \tilde{\mathbf{D}}_f(f(x))$, x•y is indistinguishable from a uniform bit.
- When we guess i too small, this is the case anyway. So exactly when we guess the correct value of i, we will get a bit of false entropy.
- Suppose we take a large product of f'(X,I,R),Y,X•Y. How do we know which of the X•Y give us false entropy, and should be output, and which don't?
- Solution: take all of the inner product bits and hash down to the expected number of false entropy bits.

Construction: Parameters

- Let $k(n) \geq 125n^3$, $I \in_U \{0,\ldots,n-1\}$, and define

  $p_n = Pr[I \leq \tilde{\mathbf{D}}_f(f(x))]$

  $m(n) = k(n)p_n - 2k(n)^{2/3}$
- Let $X',Y' \leftarrow U_{nk(n)}, I' \in_U \{0,\ldots,n-1\}^{k(n)}$, $R' \leftarrow U_{k(n)p(n).}$, $Z \leftarrow U_{m(n)}$.
- Let $h' : \{0,1\}^{p'(n)} \times \{0,1\}^{k(n)} \rightarrow \{0,1\}^{m(n)}$ be a universal hash function, and $V \leftarrow U_{p'(n)}$
- Define $g(p_n,X',Y',I',R',V) =$

  $(h'_V(X'\bullet Y'),f'^{k(n)}(X',I',R'),V,Y')$

Main Theorem

- False Entropy Theorem: g is a mildly nonuniform false entropy generator.
- Proof: Delayed…
- Main Theorem: If there exists a one-way function, then there exists a pseudorandom generator.
- Proof: Compose previous theorems: False Entropy Theorem, FEG → (mildly nonuniform) PEG theorem, PEG → PRG theorem, mildly nonuniform PRG → PRG theorem.
- We're done! Oh wait, that pesky False entropy theorem…

False Entropy Theorem

- Proof: Consider the distributions:

  D = $g(p_n,X',Y',I',R',V)$ and

  E = $(Z,f'^{k(n)}(X',I',R'),V,Y')$

  Lemma 1: $H(E) \geq H(D) + 10n^2$.

  Lemma 2: $D \cong E$

  Thus, g is a false entropy generator given p_n. We will show in the proof of lemma 2 that it is OK to use a value ρ with $p_n \leq \rho \leq p_n + 1/n$. Therefore we only need log n bits of advice. So g is a mildly nonuniform false entropy generator. QED

Proof of Lemma 1

- Lemma 1: $H(E) - H(D) \geq 10n^2$. Proof:

$H(E) = H(Z| E_{m(n)\ldots|E|}) + H(E_{m(n)\ldots|E|})$, and

$H(D) = H(h_V(X'\bullet Y') \mid D_{m(n)\ldots|D|}) + H(D_{m(n)\ldots|D|})$; so

- $H(E) - H(D) = H(Z|E_{m(n)+}) - H(h_V(X'\bullet Y')|D_{m(n)+})$
- $H(Z|E_{m(n)+}) = m(n)$; $H(h_V(X'\bullet Y')|D_{m(n)+}) \leq H(X'\bullet Y'|D_{m(n)+})$.
- For each j : $l'_j < \tilde{\mathbf{D}}_f(f(X'_j))$, then $H(X'_j \bullet Y'_j) \leq 1$.
- When $l'_j \geq \tilde{\mathbf{D}}_f(f(X'_j))$, $H(X'_j\bullet Y'_j \mid Y'_j, f'(X'_j,l'_j,R'_j)) \leq 1/2n$, since $Pr[\exists x': f'(x',l'_j,R'_j) = f'(X'_j,l'_j,R'_j)] \leq 1/2n$

Proof of Lemma 1, cont.

$H(E) - H(D) \geq m(n) -$

$Pr[l'_j < \tilde{\mathbf{D}}_f(f(X'_j))]\, k(n) + (1-Pr[...])k(n)/2n$

$\geq m(n) - (p_n - 1/n)k(n) + k(n)/2n$

$= m(n) - k(n)(p_n - 1/n + 1/2n)$

$= m(n) - k(n)(p_n - 1/2n)$

$= (k(n)p_n - 2k(n)^{2/3}) - k(n)(p_n - 1/2n)$

$= k(n)/2n - 2k(n)^{2/3}$.

$= 125n^2/2 - 50n^2 > 10n^2$. Q.E.D. (5pp down)

Lemma 2: D ≅ E

- Recall:

  D = $h_V(X'\bullet Y')$, $f'^{k(n)}(X',l',R')$, V, Y'

  E = $(Z,f'^{k(n)}(X',l',R'),V,Y')$
- Another way to describe D:
  - For each j, choose C_j=1 with probability p_n
  - When C_j = 1, choose $(X'_j,l'_j) \in T$, else $(X'_j,l'_j) \in T^C$
- Define the distribution D':
  - Same as D, except when C_j = 1 replace j^{th} input to h $(X'_j\bullet Y'_j)$ by $B_j \leftarrow U_1$.

Lemma 2 intuition…

- Notice that by the Leftover Hash Lemma, $L_1(D',E) \leq 2^{-k(n)^{1/3}} = 2^{-5n}$, so $D' \cong E$.
- Intuitively, in D' we just replace $X'_j\bullet Y'_j$ by B_j when $(X'_j,l'_j) \in T$; and we have already shown that in this case $X'_j\bullet Y'_j \cong B_j$. So we would expect $D\cong D'$, giving $D\cong E$.
- This is a non-standard hybrid argument, we will need to check it.

Hybrid argument for D≅D'

- Suppose we have A such that

  $Pr[A(D)=1] - Pr[A(D')=1] = \delta(n)$
- Define the hybrid distributions $F^{(j)}$ so that $F^{(j)}$ is distributed identically to D' up to position j and D afterwards, i.e., $F^{(j)}$ is chosen like D except that for i≤j, when C_i=1 we replace $X'_i\bullet Y'_i$ by B_i. Thus $F^{(0)} = D$, $F^{(k(n))} = D'$
- If $J \in_U \{1,\ldots,k(n)\}$, then we have that

  $E_J[\, A(F^{(J-1)}) - A(F^{(J)})\,] = \delta(n)/k(n)$

Hybrid argument continued…

- So given A, we can construct M^A that distinguishes $(f'(W,R),X\bullet Y,Y)$ from $(f'(W,R),B,Y)$ as follows:
  - On input (f'(w,r),b,y):
  - Choose $j \in_U \{1,\ldots,k(n)\}$
  - Choose a sample $s = (h_V(B),f'(W',R'),Y', V,C) \leftarrow F^{(j)}$
  - If c_j=0, output a random bit and stop
  - Else replace $f'(W'_j,R'_j),B_j,Y'_j$ by f'(w,r),b,y, and output A(s)

Hybrid argument…

- $E[M^A(f'(w,r),x \bullet y,y) - M^A(f'(w,r),b,y)] =$
  $E[A(F^{(J-1)}) - A(F^{(J)})] = \delta(n)/k(n)$
- So the hybrid argument will work…
- But there is a problem: To sample from $F^{(J)}$, we have to sample from T or T^C.
- In general there may not be an efficient way to do this. So the argument fails, $M^{(A)}$ is not efficient.