Pseudorandom generators from general one-way functions III

Review:
- Our goal is to construct a PRG from any OWF.
- A False Entropy Generator is a function $f: \{0,1\}^n \rightarrow \{0,1\}^m$ that has $f(U_n)$ computationally indistinguishable from some ptc ensemble $D_n: \{0,1\}^m$ 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]).

Review: $f'$ construction
- Let $f: \{0,1\}^n \rightarrow \{0,1\}^m$ be a one-way function, and let $h: \{0,1\}^p \times \{0,1\}^n \rightarrow \{0,1\}^{n \log 2n}$ be a universal hash function. Define $f'(x,i,r) = (f(x), hr(x)|1…i+ \log 2n, i, r)$.
- Let $Y \leftarrow U_n$, then when $I < D_\tilde{f}(f(X))$, we will have $(f'(X,I,R), Y, X \cdot Y) \cong (f''(X,I,R), Y, U_1)$.
- To formalize, define two sets:
  - $T = \{(x,i) : x \in \{0,1\}^n, i \in \{0,\ldots, D_\tilde{f}(f(x))\}\}$
  - $T^c = \{(x,i) : x \in \{0,1\}^n, i \in \{D_\tilde{f}(f(x)) + 1, \ldots, n-1\}\}$

Review: FEG Construction
- Let $k(n) \geq 125n^3$, $I \in U \{0,\ldots,n-1\}$, and define $p_n = \Pr[I \leq D_\tilde{f}(f(x))]$.
- Define $m(n) = k(n)p_n - 2k(n)^{2/3}$.
- Let $X, Y' \leftarrow U_{k(n)p(n)}$, $I' \in U \{0,\ldots,n-1\}$, $R' \leftarrow U_{k(n)p(n)}$.
- Let $h': \{0,1\}^p \times \{0,1\}^n \rightarrow \{0,1\}^m$ be a universal hash function, and $V \leftarrow U_{p(n)}$.
- Define $g(p_n, X', Y', I', R', V) = (h'(X', Y'), f'_{k(n)}(X', I', R'), V, Y')$.

Review: 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 $\rightarrow$ (mildly nonuniform) PEG theorem, PEG $\rightarrow$ PRG theorem, mildly nonuniform PRG $\rightarrow$ PRG theorem.
- We're done! Oh wait, that pesky False entropy theorem...

Review: 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: $D \equiv E$.
- 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 $p$ with $p_n \leq p \leq p_n^{1/n}$. Therefore we only need log $n$ bits of advice. So $g$ is a mildly nonuniform false entropy generator. QED
Lemma 2: \( D \cong E \)

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

Lemma 2 intuition...

- Notice that by the Leftover Hash Lemma, \( L_1(D', E) \leq 2^{-k(n)/5} = 2^{-\delta n} \), so \( D' \cong E \).
- Intuitively, in \( D' \) we just replace \( X'_j \cdot Y'_j \) by \( B_j \) when \((X'_j, I'_j) \in T\); and we have already shown that in this case \( X'_j \cdot Y'_j \cong B_j \). So we would expect \( D \cong D' \), giving \( D \cong E \).
- The hybrid argument fails, however, because we can’t efficiently sample from \( D' \).

Hybrid argument for \( D \cong D' \)

- Suppose we have \( A \) such that
  \[
  \Pr[A(D) = 1] - \Pr[A(D') = 1] = \delta(n)
  \]
- Define the hybrid distributions \( F^{(i)} \) so that \( F^{(i)} \) is distributed identically to \( D' \) up to position \( j \) and \( D \) afterwards, i.e., \( F^{(i)} \) is chosen like \( D \) except that for isj, when \( C_j = 1 \) we replace \( X'_j \cdot Y'_j \) by \( B_j \). Thus \( F^{(i)} = D \), \( F^{(k(n))} = D' \).
- If \( j \in \{1, \ldots, k(n)\} \), then we have that
  \[
  E_{j}[(A(F^{(j-1)}) - A(F^{(j)}))] = \delta(n)/k(n)
  \]
- How to fix our Hybrid argument?
  - Notice that when \( C_j = 0 \), \( A \) has no advantage, yet when \( C_j = 1 \) \( A \) has significant advantage.
  - So \( A \) “knows” when an element \( W \in T \), given \( f'(W, R) \).
  - We will take advantage of this to build hybrid distributions which are “close” to \( F^{(i)} \) allowing us to get by the problem.
  - This is the last 4 technical pages of [HILL99]

New Hybrids...

- We will define two sets of hybrid distributions, \( E^{(0)}, D^{(0)} \) for \( j \in \{0, \ldots, k(n)\} \).
- We will have \( E^{(0)} = E, D^{(0)} = D \), and \( E^{(k(n))} = D^{(k(n))} \).
- Define \( \delta^{(0)} = \Pr[A(D^{(0)}) = 1] - \Pr[A(E^{(0)}) = 1] \).
- Then \( \delta^{(0)} = \delta(n) \) and \( \delta^{(k(n))} = 0 \).
- We will also have: \( E_{j}[(\delta^{(j-1)} - \delta^{(j)})] \geq \delta(n)/k(n) \).
- This will allow us to (indirectly) invert \( f' \) later.

Definition of \( D^{(0)}, E^{(0)} \)

- Define parameters:
  - \( \rho = \delta(n)/(16k(n)) \)
  - \( \tau = 64n^2/\rho \)
- Define: \( D^{(0)} = D; E^{(0)} = E; B \leftarrow U_{k(n)} \).
- Suppose \( D^{(j-1)} \) is defined. Then to sample from \( D^{(j)} \):
  - Choose \( C_j \in \{0,1\} \) so that \( \Pr[C_j = 1] = p_n \).
  - Sample \( x_m \leftarrow U_n, l_m \in \{1, \ldots, n\} \), let \( w_m = (x_m, l_m) \), \( 1 \leq m \leq \tau \).
D(j-1)(c_j,w_m) = \Pr[A(D(j-1)(c_j,w_m)) = 1] - \Pr[A(E(j-1)(w_m)) = 1].

Define \(\delta(j-1)(c_j,w_m)\) to be the same as \(\delta(j-1)\) except that \((X_j',I_j')\) is fixed to \(w_m\) and the \(j\)th input bit of \(h'\) is set to \(x_m\) if \(c_j=0\) and \(B_j\) otherwise.

Define \(\delta^\oplus(j)(c_j,w_m)\) to be \(\delta(j)(c_j,w_m)\) except \((X_j',I_j')\) is fixed to \(w_m\).

Using our hybrids

Define \(D(j)(w,r,b,y)\) to be \(D(j)\) with \(f'(X_{j+1}',Y_{j+1}',R_{j+1}')\) replaced by \(f'(w,r)\), the \(j+1\) input bit to \(h'\) replaced by \(b\), and \(Y_{j+1}'\) replaced by \(y\); Same for \(E(j)(w,r,y)\).

Define \(\epsilon(j) = E\left[\delta(j)(0,W) - \delta(j)(1,W)\right]\)

Define \(\epsilon^\oplus(j) = E\left[\delta^\oplus(0,W) - \delta^\oplus(1,W)\right]\)

So we just need to show that \(E[\epsilon(j)] \geq \delta(n)/8k(n)\)

Proof, con’t...

Notice that:

- \(E[d(j,w,R,x\cdot Y,Y) - e(j,w,R,Y)] = \delta^\oplus(0,w)\)
- \(E[d(j,w,R,B,Y) - e(j,w,R,Y)] = \delta^\oplus(1,w)\)

Define \(g(j) = E[\delta^\oplus(0,W) - \delta^\oplus(1,W)]\)

Then the advantage of \(M^A\) is:

\(E[M^A(f'(W,R),X\cdot Y,Y) - E[M^A(f'(W,R),B,Y)] = E[g(j)] - E[g(j)]/2 = E[\epsilon(j)]/2\)

So we just need to show that \(E[\epsilon(j)] \geq \delta(n)/8k(n)\)
Alternatively…

- Alternatively we can show that 
  \( E[\delta(\epsilon^0)] \geq 2\rho k(n) \)
- We will prove this by showing that:
  a) \( E[\delta(k(n))] \leq 2^{-n+1} \)
  b) \( E[\delta(j) - \delta(j+1)] \leq \epsilon(j) + 4\rho \)
- This will give us:
  \[ 8\rho k(n) = \delta(n)/2 < \delta(n) - E[\delta(k(n))] = \sum_j E[\delta(j) - \delta(j+1)] \leq 4k(n)\rho + E[\sum_j \epsilon(j)]. \]

Proof of (a) \( E[\delta(k(n))] \leq 2^{-n+1} \)

- Notice that \( E[k(n)] \) and \( D[k(n)] \) are identical except that the first \( m(n) \) bits of \( E[k(n)] \) are Z and the first \( m(n) \) bits of \( D[k(n)] \) are the output of \( h' \).
- But \( H_n(input to h') \) rest of \( D[k(n)] \) \( \geq \sum_c \epsilon_c \).
- A Chernoff bound gives us that with probability at least \( 1-2^{-n} \), 
  \[ \sum_j \epsilon(j) \geq k(n)p_j - k(n)2^3 = m(n) + k(n)2^3 \]
- When this is true, we get from the Leftover hash lemma that 
  \[ L_1(D[k(n)],E[k(n)]) \leq 2^{-k(n)/2} < 2^{-n}. \]
- This gives us \( E[\delta(k(n))] \leq 2^{-n+1}. \)

Proof of (b) \( E[\delta(j) - \delta(j+1)] \leq \epsilon(j) + 4\rho \)

- Recall that \( W \in U \). Define \( W \in U^T \). 
- Then since the \( j+1 \) input to \( h' \) in \( D^0 \) is always \( X_j' \cdot Y_j' \), we have 
  \[ \delta(j) = p_j E[\delta(j)(0, W)] + (1-p_j) E[\delta(j)(0, W^c)] \]
- We will complete the proof by showing that 
  \[ E[\delta(j)] + 4\rho \geq p_j E[\delta(j)(1, W)] + (1-p_j) E[\delta(j)(1, W^c)]. \]

To show: 
\[ E[\delta(j)] + 4\rho \geq p_j E[\delta(j)(1, W)] + (1-p_j) E[\delta(j)(1, W^c)] \]

- A Chernoff Bound gives us that with probability at least \( 1-2^{-n} \), for stage \( j \), at least \( n/\rho \) of the \( w_m \) are in \( T \) and at least \( n/\rho \) of the \( w_m \) are in \( T^c \).
- Thus with probability at least \( 1-2^{-n} \), we have: 
  \[ \max_m \{ \delta(j)(c, w_m) \} \geq \max_m \{ E[\delta(j)(c, W)] \}, E[\delta(j)(c, W^c)] \} - \rho \]
- Also recall that with probability at least \( 1-2^{-n} \), we have \( |\Delta(j)(c, w_m) - \delta(j)(c, w_m)| < \rho \)
- With probability at least \( 1 - 3\cdot2^{-n} \). Thus: 
  \[ E[\delta(j)] = E[\delta(j)(c, W)] \]
  \[ \geq \max \{ E[\delta(j)(c, W)], E[\delta(j)(c, W^c)] \} - 4\rho \]
  Giving the required inequality.

So we are done

- This completes the proof that \( A \) distinguishes \( f'(w, r), x \cdot y, y \) from \( f'(w, r), b \cdot y \).
- Thus completing the proof that a F.E. Generator can be constructed from any one-way function.
- HUGE issue: suppose we compose the various constructions to get a pseudorandom generator. Then to get inputs to \( f \) of size \( n \), the inputs to the resulting generator will have size \( n^{34} \). [HILL99]
Open problem

- Now we don’t actually require all of the intermediate product distributions… [HILL99] claim that the same techniques can chip it down to inputs of size $n^4$.
- Open problem: construct a pseudorandom generator from any one-way function $f$ such that the security of $f$ on inputs of size $n$ is related to the security of $g$ on inputs of size $n^2$ or $n^3$. 