Last time we built PRBG’s based on OWP’s and hardcore bits. Before moving on, we’ll mention a couple of other results.

1 Goldreich-Levin

Theorem (Goldreich-Levin): Any OWP can be modified into a OWP that has a HCB.

Suppose that $f : \{0, 1\}^l \rightarrow \{0, 1\}^l$ is a 1-way permutation. Then we can define $F^\prime : \{0, 1\}^{2l} \rightarrow \{0, 1\}^{2l}$ such that $F^\prime(x||y) = F(x)||y$ ($|x| = |y| = l$) and $B(x||y) = \langle x, y \rangle \in \{0, 1\}$, where $\langle x, y \rangle = \oplus_{i=1}^l x_i y_i$ is the inner product over $GF(2)$ of $x$ and $y$ (and where $x = x_1 x_2 \ldots x_l, y = y_1 y_2 \ldots y_l$). Then we have

**Theorem 1.** $F^\prime$ is a OWP and $B$ is a hardcore bit.

For a proof, see notes by Ostrovsky.

2 One Way Functions and the HILL (Haastad Impagliazzo Levin Luby) Theorem

Consider a one-way function $F : X \rightarrow Y$. It is easy to compute $F$, but hard to compute $F^{-1}$. That is, for any efficient algorithm $A$,

$\Pr[x \leftarrow X, y \leftarrow F(x), x' \leftarrow A(y) \mid F(x') = y]$ is negligible.

The notion of a one way function is a weaker and more general notion than one-way permutation, so one way functions are more likely to exist. The notions of the hardcore bit and the goal of the Goldreich-Levin theorem apply to one-way functions. However, you cannot plug a OWF into Goldreich-Levin’s PRBG: that is, their proof for a OWP used the fact that it was a permutation in all the hybrid argument steps! Nevertheless, HILL showed that you can construct a PRBG from a OWF. It is a constructive proof, but it is impractical, and is omitted here.

Note though that the converse is easy – a PRBG is clearly a OWF. Assume $G : \{0, 1\}^l \rightarrow \{0, 1\}^{l'}$ with $l' >> l$. If we could break the OWF, we could compute the seed and determine whether the value is pseudorandom. This would yield a distinguisher, and contradict the assumption of a PRBG.
3 Pseudo-Random Functions (PRF’s)

We wish to find a family of functions where we select an individual function by selecting a key $k$ that defines a function. The setup, given oracle access to $A$ with adaptive queries and $k$ fixed:

We define the security of a PRF in terms of an attack game between an adversary and a challenger. Adversary $A$ talks to one of two oracles; One oracle is the PRF oracle for $F_k$ (random key $k$). The other oracle is a random function oracle. Security is achieved is $A$ can’t tell which oracle he’s talking to.

A random function oracle is an oracle for a function chosen at random from the set of all functions. It can be implemented as a random number generator which keeps a table of all answers already generated, and returns the already given answer for repeated queries, and otherwise returns a random element.

3.1 Applications

A few applications of PRFs are presented here.

3.1.1 Multi-Message Attacks

Multi-message attacks are described in HW3 problem 6. In this model, the same bit $b$ is used for multiple encryptions.

Let the encryption be: $E(m) = (r, F_k(r) \oplus m)$, and the decryption be: $D(r, s) = F_k(r) \oplus s$. $r$ is called a nonce – meaning it is used only once. $r$ could be a counter, for example, but that would require that the sender maintain state, which doesn’t fit into our definitional framework, and could cause headaches if you accidentally reset $r$. So it is recommended to choose $r$ at random, and $l$ must be large enough to avoid collisions in choosing $r$’s.

We shall prove multi-message security via three games. Let $\ell$ be an upper bound on the number of decryption queries, $b$ be the hidden bit, and $\hat{b}$ the output of the adversary.

Game 0: Original attack game. $S_0 = \text{event that } b = \hat{b}$ in Game 0.

Game 1: Same as game 0, but if any nonce repeats, we halt the game. Let $S_1 = \text{event that } b = \hat{b}$ in Game 1.

We prove that these two games are indistinguishable, using the following lemma.

**Lemma 2.** Difference Lemma. If we have events $E_0, E_1, F$, with $E_0 \land \neg F \Leftrightarrow E_1 \land \neg F$, then $|Pr[E_0] - Pr[E_1]| \leq Pr[F]$. 

L9-2
Let $F$ be the event that a nonce repeats. If $F$ does not occur, games 0 and 1 proceed identically. By construction, $S_0 \land \neg F \iff S_1 \land \neg F$. So by the difference lemma, $|\Pr[S_0] - \Pr[S_1]| \leq \Pr[F] = O(2^{-r})$.

Game 2: same as Game 1 but with the output of $F_k$ replaced by uniformly distributed independent random bitstrings. Let $S_2 = \text{event that } b = \hat{b}$ in Game 2.

Now we are free to apply the definition of a pseudo-random function. Under the secure PRF assumption, we have $|\Pr[S_1] - \Pr[S_2]|$ is negligible, since otherwise we would have a distinguisher for the PRF. $\Pr[S_2] = \frac{1}{2}$, so we conclude that $|\Pr[S_0] - 1/2| \leq O(2^{-r}) + \nu$, where $\nu$ negligible, and security follows.

### 3.1.2 Message Authentication

The setup: Alice and Bob share a key $k$. Alice sends several messages $m_1, m_2, \ldots$ to Bob. The adversary can eavesdrop and inject messages.

Suppose Alice sends $(m_i, t_i = F_k(m_i))$ to Bob. If $F_k$ is a PRF, we have that $F_k$ is also an unpredictable function. Unpredictability can be defined as follows: The adversary chooses $x_1, \ldots, x_\ell$ adaptively, obtaining $F_k(x_1), \ldots F_k(x_\ell)$. It is hard to compute $F_k(x)$ for $x \not\in \{x_1, \ldots, x_\ell\}$.

### 3.1.3 Interactive Identification Scheme.

The problem: Friend or Foe Identification.

The solution: $A$ sends $r$, $B$ sends $F_k(r)$. This is a simple challenge-response protocol.
3.2 PRF from PRBG

We show how to compute a PRF from a PRBG. Suppose \( G : \{0, 1\}^t \rightarrow \{0, 1\}^t \) is a secure PRBG. The keyspace is \( \{0, 1\}^t \), the input space is \( \{0, 1\}^t \), and \( t \) is a parameter. The output space is \( \{0, 1\}^t \).

We will define \( G \) by a pair of functions \((G_0, G_1)\): \( G(s) = G_0(s)||G_1(s) \).

If \( t = 1 \), we are done: a PRF with 1-bit input space is a PRBG.

Say the input to the PRF is \( x = x_1x_2\ldots x_t \), with \( x_i \in \{0, 1\} \), \( x \in \{0, 1\}^t \).

The key is \( k \in \{0, 1\}^t \).

We shall consider a binary tree with each link of the form: \( x \rightarrow G_0(x), G_1(x) \), with \( k \) at the root. That is, each level has applications of \( G_0, G_1 \) respectively. e.g. \( G_0(k) \rightarrow G_0(G_0(k)), G_1(G_0(k)) \rightarrow G_0(G_1(k)), G_1(G_1(k)) \) are the second-level nodes.

We evaluate \( F_k(x) \) by tracing out a path in this tree; the leaf that we reach is the output.
That is, \( F_k(x) = G_{x_t}(\ldots(G_{x_1}(k))\ldots) \).

**Theorem 3.** \( F_k(x) \) is a secure PRF.

**Proof.** We use the hybrid argument. We replace successive levels (innermost, \( x_1 \), upper level of tree first) with random. More formally, we have to do a randomized hybrid argument and one further observation to make the proof work. Define a sequence of hybrid, intermediate games:

- **Game 0:** the original game. \( t \) levels. \( k, k_0, k_1, k_{00}, \ldots \)
- **Game 1:** same except start with random \( k_0, k_1 \).

Suppose the adversary makes up to \( q \) queries. Each query is a path through the tree. As the game progresses we can trace out and remember the paths. This way, in game \( i \), we only need to generate at most \( q \) random values at level \( i \), rather than \( 2^i \). So it is practical.

\[ \ldots \]

- **Game \( t \):** We have random outputs.

  We compare game \( i \) to \( i + 1 \). Distinguishing between \( i \) and \( i + 1 \) gives a distinguisher for \( q \) copies of \( G \).

  Suppose \( A \) distinguishes the PRF from random. Run the following algorithm: choose \( i \) at random from \( \{1, \ldots, t\} \). Input: \( k_{10}, k_{11}, k_{20}, k_{21}, \ldots, k_{q0}, k_{q1} \). That is, these are either random or outputs of a PRF.

  Choose \( i \) and run game \( i \) using these values to seed level \( i \).
The advantage that $A'$ has in distinguishing random from pseudorandom is equal to $(1/t)(\text{advantage that } A \text{ has in distinguishing game } i \text{ from game } i + 1)$. Then we plug in results from the homework 2 exercise which leads to another $1/q$ factor. Finally, we get an algorithm $A''$ that has a distinguishing advantage of $1/(qt)$ times the advantage of $A$. So $F_k$ is secure if the PRBG is secure.

\( \square \)