So far in this course, we have shown that:

\[ \text{PRF} \Rightarrow \text{CPA-secure encryption} \Rightarrow \text{authenticated encryption} \]  

Conceptually “simpler” object

From HW1, we saw how to construct a PRF from a (length-doubling) PRG:

\[ \text{can be built from any PRG with 2-bit stretch} \]

**Question:** Can we distill this further? Can we base symmetric cryptography on an even simpler primitive?  
- Cryptography is about exploiting some kind of asymmetry: we want an operation that is “easy” for honest users, but hard for adversaries.  
- Suggests a notion of “hard to invert”: (cannot recover seed from PRG, cannot decrypt without knowledge of secret, etc.)

**Definition.** A function \( f: X \rightarrow Y \) is one-way if 
1. \( f \) is efficiently computable.  
2. For all efficient adversaries \( A \):
   
   \[
   \Pr[X \leftarrow X, \ y \leftarrow A(f(x)) : f(x) = f(y)] = \text{negl}(\lambda)
   \]

"Function is hard to invert on average"

**Theorem (Hastad-Imaglische-Lein-Luby).** \( \text{OWF} \Rightarrow \text{PRG} \) [implies OWF is sufficient (and necessary) for symmetric cryptography]

We will consider a weaker statement: one-way permutation \( \Rightarrow \text{PRG} \)

**Definition.** A function \( f: X \rightarrow X \) is a one-way permutation if  
1. \( f \) is one-way  
2. \( f \) is a permutation.

**Goal:** given a OWP \( f: X \rightarrow X \), can we construct a PRG with one-bit stretch.  
**Idea:** if \( X \leftarrow X \), then \( f(x) \) is uniformly random  
moreover, given \( f(x) \), should be difficult to recover (all or) \( X \leftarrow \) leverage this to get 1 pseudorandom bit

**Definition.** Let \( f: X \rightarrow Y \) be a one-way function. Then \( h: X \rightarrow \mathbb{R} \) is a hard-core predicate for \( f \) if no efficient adversary can distinguish the following distributions:  
\[
D_0: \left\{ X \leftarrow X : (f(x), h(x)) \right\} \\
D_1: \left\{ X \leftarrow X, r \leftarrow R : (f(x), r) \right\}
\]

If a OWP has a hard-core predicate, that immediately implies a PRG:
\[
\text{PRG}(\lambda) := f(\lambda) \| h(\lambda)
\]

Typically, we will consider hard-core bits

\( \text{(i.e., } R = \{0,1\} \text{)} \)

**Lemma.** Let \( f: X \rightarrow Y \) be a one-way function. Suppose \( h: X \rightarrow \{0,1\} \) is unpredictable in the following sense: for all efficient adversaries \( A \):

\[
| \Pr[X \leftarrow X : A(f(x)) = h(x)] - \frac{1}{2} | = \text{negl}(\lambda)
\]

If \( h \) is unpredictable, then it is a hard-core bit.  

[Note: Converse of this is immediate]
Proof. Suppose there exists an efficient $A$ that can distinguish between $(f(x), h(x))$ and $(f(x), r)$ for $x \notin X$ and $b \in \{0,1\}$ with advantage $\varepsilon$. We use $A$ to build a predictor $B$:

1. On input $f(x)$, sample $b \xleftarrow{\$} \{0,1\}$ and run $A$ on input $(f(x), b)$.
2. If $A$ outputs 1, then output $b$. Otherwise, output 1-$b$.

Intuition: Suppose $A$ is more likely to output 1 given inputs from the “hard-core bit distribution”. This means that $A$ outputs 1 if we guess correctly.

Formally: $\Pr[B(f(x)) = h(x)] = \Pr[A(f(x), b) = h(x)]$

$$= \Pr[A(f(x), b) = 1 \mid b = h(x)] \Pr[b = h(x)] + \Pr[A(f(x), b) = 0 \mid b = 1 - h(x)] \Pr[b = 1 - h(x)]$$

$$= \frac{1}{2} + \frac{1}{2} \left( \Pr[A(f(x), b) = 1 \mid b = h(x)] - \Pr[A(f(x), b) = 1 \mid b = 1 - h(x)] \right)$$

Now, $\varepsilon = \left| \Pr[A(f(x), h(x)) = 1] - \Pr[A(f(x), b) = 1] \right|$ 

$$= \left| \alpha - \Pr[b = h(x)] \Pr[b = h(x)] \right| - \frac{1}{2} \left| \Pr[b = h(x)] \right|$$

$$= \frac{1}{2} \left| \alpha - \beta \right|$$

$$= \frac{1}{2} \left| \alpha - \beta \right|$$

Theorem (Goldreich-Levin). Let $f : \{0,1\}^n \rightarrow \{0,1\}^n$ be a one-way function. For a string $r \in \{0,1\}^n$, define the function $h_r : \{0,1\}^n \rightarrow \{0,1\}$ where $h_r(x) = \Sigma r_i x_i \mod 2$. Then the function $g(x, r) = (f(x), r)$ is one-way and $h_r$ is a hard-core predicate for $g$.

Observe that if $f$ is a OWF, then so is $g$.

Proof Idea. One-wayness of $g$ immediately follows from one-wayness of $f$. Suffices to show that $h_r$ is hard-core. Suppose that $h_r$ is not a hard-core predicate for $g$. This means that there is an adversary $A$ that can predict $h_r$ given $(f(x), r)$ with probability $\frac{1}{2} + \varepsilon$. We will use $g$ to construct an adversary $B$ that can invert $f$ (and thus $g$).

Here: Suppose $A$ succeeds with probability $\frac{1}{2}$:

$$\Pr[A(g(x, r) = h_r(x)] = \frac{1}{2} \quad \text{(for } x, r \in \{0,1\}^n)$$

Given $y = f(x)$, run $A$ on inputs $(y, e_1), ..., (y, e_n)$ where $e_i$ is the $i$th basis vector:

$$h_{e_i}(x) = \langle e_i, x \rangle \mod 2$$

$$= x_i \in \{0,1\}$$

Suppose now that $A$ succeeds with probability $\frac{3}{4} + \varepsilon$ for constant $\varepsilon > 0$:

Evaluating at $e_1, ..., e_n$ not guaranteed to work since $A$ could be wrong on all of these inputs.
Analysis proceeds in two steps:

1. Fix an \( x \in \mathbb{Z}_2^n \). Suppose we have a function \( t : \mathbb{Z}_2^n \rightarrow \{0,1\} \) where
   \[
   \Pr[r \in \mathbb{Z}_2^n : t(r) = \langle x, r \rangle] \geq \frac{3}{4} + \varepsilon
   \]
   We show that we can learn \( x \) by evaluating \( t \) on carefully chosen points.
   Similar to before, \( t \) could be wrong on \( e_1, \ldots, e_n \). Need evaluation points to be random.

   Sample \( r \in \mathbb{Z}_2^n \) and evaluate \( t \) at \( r \) and \( e_1 + r \).

   By assumption:
   \[
   \Pr[t(r) = \langle x, r \rangle] \geq \frac{3}{4} + \varepsilon
   \]
   \[
   \Pr[t(r + e_1) = \langle x, r + e_1 \rangle] \geq \frac{3}{4} + \varepsilon
   \]  
   (since \( r + e_1 \) with \( r \in \mathbb{Z}_2^n \) is uniform)

   But these events are not independent: inputs are correlated!

   Consider the complements:
   \[
   \Pr[t(r) \neq \langle x, r \rangle] < \frac{1}{4} - \varepsilon \quad \Rightarrow \quad \Pr[t(r) \neq \langle x, r + e_1 \rangle] < \frac{1}{4} - \varepsilon
   \]

   Thus, with prob. at least \( \frac{1}{2} + 2\varepsilon \), \( t(r) = \langle x, r \rangle \) and \( t(r + e_1) = \langle x, r + e_1 \rangle \)

   Set \( z = t(r) + t(r + e_1) \)

   If \( t(r) = \langle x, r \rangle \) and \( t(r + e_1) = \langle x, r + e_1 \rangle \),

   \[
   t(r + e_1) - t(r) = \langle x, r + e_1 \rangle - \langle x, r \rangle = \langle x, e_1 \rangle = x_1
   \]

   Idea: Sample \( k \) independent pairs \( (r_i, r_i + e_{i1}) \) for \( r_i \in \mathbb{Z}_2^n \) and compute estimates \( z_1, \ldots, z_k \)

   Take the first bit \( \hat{x}_1 \) to be Majority(\( z_1, \ldots, z_k \))

   Repeat this procedure to obtain estimates \( \hat{x}_2, \ldots, \hat{x}_n \). Output \( \hat{x}_1, \hat{x}_2, \ldots, \hat{x}_n \).

   Analysis will use a Chernoff bound. Simple version for our setting:

   Let \( X_1, \ldots, X_k \in \{0,1\} \) be independent random variables where
   \[
   \Pr[X_i = 1] \geq \frac{1}{2} + \varepsilon
   \]

   Then,
   \[
   \Pr[\text{Majority}(X_1, \ldots, X_k) = 1] \leq 2\varepsilon e^{-2\varepsilon^2 k}
   \]

   In particular, if \( \varepsilon = o(1) \),
   \[
   \Pr[\text{Majority}(X_1, \ldots, X_k) = 1] \leq 2^{-o(k)}
   \]

   for each bit of \( x \).

   By the Chernoff bound, \( \hat{x}_i = X_i \) with probability \( 1 - \text{neg}(\varepsilon) \). Repeating this \( n \) times yields the desired result.

   Total evaluations of \( t \): \( O(n^2) \)

2. Our setting is not quite this:

   \[
   \Pr[X_r \in \{0,1\} : A(f(x), r) = \langle x, r \rangle] \geq \frac{3}{4} + \varepsilon
   \]

   Randomness taken over both \( x \) and \( r \) while above analysis only looks at \( r \).

   Let's say an \( x \) is "good" if

   \[
   \Pr[r \in \{0,1\} : A(f(x), r) = \langle x, r \rangle] \geq \frac{3}{4} + \varepsilon
   \]

   If \( x \) is "good", then can recover \( x \) using above algorithm.

   How many \( x \)'s are good? If \( \Pr[X \in \{0,1\} : x \text{ is "good"}] \) is non-negligible, then we have proven the claim. Algorithm B runs above decoder on \( A \) and recovers \( x \) whenever \( x \) is good, which happens with non-negligible probability.

   If \( A \) succeeds on \( \frac{3}{4} + \varepsilon \)-fraction of \( x \)'s, cannot have too many 'bad' \( x \)'s. (Averaging argument)

   Suppose \( 8 \) fraction of \( x \)'s are bad. Then, probability of \( A \) succeeding over choice of \( x \)'s \( 8 \frac{3}{4} + \varepsilon \) is at most
\[
\delta \left( \frac{3}{4} + \frac{\delta}{2} \right) + (1-\delta) = 1 - \frac{\delta}{4} + \frac{\delta^2}{2}
\]

Require that 
\[
1 - \frac{\delta}{4} + \frac{\delta^2}{2} \geq \frac{3}{4} + \varepsilon \quad \Rightarrow \quad 1 - \delta + 2\delta\varepsilon \geq 4\varepsilon
\]

\[
\Rightarrow \quad 8(1-2\varepsilon) \leq 1-4\varepsilon \quad \Rightarrow \quad 8 \leq \frac{1-4\varepsilon}{1-2\varepsilon}
\]

**Conclusion:** At most constant fraction is “bad” so inversion will succeed on constant fraction of inputs.

**HW:** Show how to go from \( \frac{3}{4} + \varepsilon \) to \( \frac{1}{2} + \varepsilon \) for constant \( \varepsilon > 0 \).