Last time

• How to make PRGs stretch more
• How to use Cryptographic Hash Functions to get PRGs
• Chosen plain-text security

Today

• Pseudorandom Functions
• PRFs → CPA secure encryption
• Starting Authentication
Recalling CPA Security (and Randomized Enc)
Recalling how PRGs were useful for single-message security.

Relying on OTP’s idea.

\[ \text{Enc} \]

\[ \text{PRG} \]

\[ \text{Key} = \text{PRG}(\text{K}) \]

\[ \oplus \]

\[ \text{C} \]

\[ \text{C} \]

\[ \oplus \]

\[ m \]

\[ m_1 \oplus m_2 \]

pseudo-random
A useful lemma for indistinguishability

Lemma: Starting point

Conclusion: for all poly-time $Z$ (potentially randomized)

then

$Y_n, X_n : \text{randomized inputs.}$

$X_n, Y_n : \text{outputs.}$

$X_n \approx X_n'$

$Y_n \approx Y_n'$

$\exists \text{ neg \ e.}$

$P_1[A(x) = 1] - P_1[A(y) = 1] \leq \text{neg}(n).$

if $B$ breaks $X_n \approx Y_n$ then output are also indistinguishable.
Pseudo Random Functions:
A very long-output PRG
Pseudo-Random Functions (other definition)

**Definition 3.25** Let $F : \{0, 1\}^* \times \{0, 1\}^* \rightarrow \{0, 1\}^*$ be an efficient, length-preserving, keyed function. $F$ is a pseudorandom function if for all probabilistic polynomial-time distinguishers $D$, there is a negligible function $\text{negl}$ such that:

$$\left| \Pr[D^{F_k}(1^n) = 1] - \Pr[D^{f}(1^n) = 1] \right| \leq \text{negl}(n),$$

where the first probability is taken over uniform choice of $k \in \{0, 1\}^n$ and the randomness of $D$, and the second probability is taken over uniform choice of $f \in \text{Func}_n$ and the randomness of $D$.

$F$ is PRF if for all poly-time $A$...
How to Obtain Pseudorandom Functions?
(1: using length-doubling PRGs)
How to Obtain Pseudorandom Functions?
(2: again using hash functions)

\[ F(k, x) \in \{0,1\}^{256} \]
\[ k \in \{0,1\}^{128} \]

Without the key SHA256 is NOT a PRF.
How to use PRFs to encrypt CPA securely?

W.l.o.g.: enough to only encrypt $M = 0, 1^l$

Claim: $\text{Enc}(k, x): X = (x_1, x_2, \ldots, x_k) \xrightarrow{\ell \text{-bit}} X \xleftarrow{\ell \text{-bit}} y_1 \leftarrow \text{fresh}\ \text{Enc}(k, x_k)$

and $\Pr\left(y_1 = y_k\right) = \Omega$

needs a proof: (Book prove it!)
We have \( F(k, x) \rightarrow y \rightarrow \text{l-bit} \).

Goal: \( \text{Enc}(k, m) \rightarrow c \rightarrow \text{l-bit} \).  
\( \text{Dec}(k, c) \rightarrow m \)

1. \( \text{Enc}(k, m) = F(k, m) ? \text{Dec} ? \)
2. \( \text{Enc}(k, m) = F(k, m) \oplus m \)
3. \( \text{Enc}(k, m) = F(k, 1234) \oplus m \)  
\( \text{Dec}(k, c) = F(k, 1234) \oplus c \)
How to use PRFs to encrypt?
Recall: it has to be a randomized encryption!

$4^{th}:
\text{Enc}(k, m, r) = F(k, r) \oplus m, r$

Pick $i \leq \text{len} + 1 \oplus n$

$\text{Dec}(k[C]) = F(k, r) \oplus c = m$

$\text{Dec}(k, C) := c = [x, r]$

$F(k, r) \oplus x = m$

$\text{block of length } n$
Pseudo-Random Functions

→ CPA Secure Encryption

• PRF $F_k(x)$: For a randomly chosen $k$ no poly-time distinguisher $A$ can distinguish if it is “talking to” $F_k(\cdot)$ or a truly random function $R(\cdot)$

• Construction of CPA secure encryption using PRF $F_k(\cdot)$:
  1. Generate random key $k$ and use it as the key to the PRF
  2. To encrypt message $m$ of length $l_{output}$ take $c = [r, m \oplus F_k(r)]$ for random $r$
  3. To decrypt $c = [r, y]$ take $m = y \oplus F_k(r)$
Proof of Security:

\[ \text{Enc}(k, m) = (r, F_k(r) \oplus m) \]

- **Real 0**
- **Ideal 0**: Substitute with \( R \)
- **Ideal 1**: Substitute \( F \) with \( R \)
- **Real 1**

If \( |P_0 - q_0| \geq \epsilon \), we can make this into an attack against the security of \( PRF \).