Legal Notice

The Zoom session for this class will be recorded and made available asynchronously on Canvas to registered students.
Announcements

1. HW 1 is due today! Turn it in now if you haven’t yet!

2. HW 2 is out, due before class in 1 week, October 20.
Last time: Block ciphers

This time: Pseudorandom functions and chosen plaintext attacks
Pseudo-random functions (PRFs)

Deterministic algorithm $F$:

- $k \in K$, $x \in X$, $y \in Y$
- $F_k(x) = y$
- Should be computationally indistinguishable from a truly random function

In contrast to the (pseudo)random permutations from last lecture, this function is not required to be one-to-one.
Distinguishing experiment for PRFs

\[ D \quad 0 \]

1. Chooses \( f = \begin{cases} 
\text{random fn} \\
\text{PRF } F_k \text{ } k \in \{0,1\}^n
\end{cases} \)

2. query \( x_i \)

\[ f(x_i) \]

\[ x_2 \]

\[ f(x_2) \]

\[ \vdots \]

\[ b \in \{0,1\} \]

**Definition**

\( F_k \) is a secure PRF if \( \forall \) efficient \( D \)

\[ | \Pr[D(F_k) = 1] - \Pr[D(\text{random fn}) = 1] | \text{ negligible} \]
Is a PRP indistinguishable from a PRF?

Consider a distinguishing experiment between functions and permutations.

**Observation:** The only way to distinguish between a permutation and a non-permutation function is to find a collision, inputs so that \( f(x_1) = f(x_2) \).

Let \(|X| = N\). By the birthday bound, the adversary will observe a collision in outputs in a PRF with constant probability after \( \sqrt{N} \) inputs.

**Theorem**

An adversary that makes \( Q \) queries can distinguish a random permutation from a random function with probability at most \( Q^2 / 2N \).
Constructing PRGs from PRFs

**Theorem**

Let $x_1, \ldots, x_\ell$ be fixed, distinct elements, and $F$ be a PRF. $G(k) = (F_k(x_1), F_k(x_2), \ldots, F_k(x_\ell))$ is a secure PRG.

**Proof.**

Assume for contradiction that adversary $A_G$ can distinguish this PRG from random. Can construct a distinguisher $A_F$ for the PRF.
Counter Mode

In the previous construction \( x_1, \ldots, x_\ell \) just need to be distinct elements. So just choose \( r \) and let \( x_1 = r, x_2 = r + 1, \ldots \)

stream: \( F_{1c}(r), F_k(r+1), F_{1c}(r+2), \ldots \)

Then we can use this as a stream cipher to encrypt:

\[
\text{Enc}_k(m) = (r, F_k(r) \oplus m[0], F_k(r + 1) \oplus m[1], \ldots, F_k(r + \ell) \oplus m[\ell])
\]

\[
\text{Dec}_k(c) = (F_k(r) \oplus c[0], F_k(r + 1) \oplus c[1], \ldots, F_k(r + \ell) \oplus c[\ell])
\]

This is semantically secure.
Attack models we’ve seen so far:

Ciphertext-only attack
- Most restrictive attack model

Known plaintext attack
- Historical example: WWII Enigma-encoded messages from Germans ending in “Heil Hitler”
- Modern example: Observing ciphertext from someone visiting the main page of Wikipedia over HTTPS.

Both of these are covered by semantic security.

New attack model:
Chosen plaintext attack
- Historical example: British military would place mines in particular locations hoping Germans would send encrypted messages about that location.
- Modern example: Attacker-controlled Javascript on a web page causes victim web client to make a HTTPS connection.
Chosen plaintext attack

![Diagram of Chosen Plaintext Attack]

Definition
Enc is CPA-secure if ∀ efficient A, Pr[A succeeds] ≤ 1/2 + ε for ε negligible
Another definition of chosen plaintext attack

**Definition**

Enc is CPA-secure if

\[ | \Pr[A \text{ outputs } 1 \mid b = 1] - \Pr[A \text{ outputs } 1 \mid b = 0] | \text{ negligible} \]
Theorem

No deterministic cipher can be CPA-secure.
Theorem

No deterministic cipher can be CPA-secure.

Proof.
Adversary queries \((m_0, m_1)\) then \((m_0, m_0)\).  
\qed
Using a PRF for CPA-secure encryption

• Generate $k$ at random.

• Encryption:
  1. Generate $r$ uniformly at random.
  2. $\text{Enc}_k(m) = (r, F_k(r) \oplus m)$

• Decryption:
  1. Parse $c = (r, s)$
  2. $\text{Dec}_k(c) = F_k(r) \oplus s$. 
Theorem

The $\text{Enc}_k(m) = (r, F_k(r) \oplus m)$ construction on the previous slide is CPA-secure.

Proof.

By contradiction. Assume adversary $A$ can win CPA-security game \#1 with non-negligible advantage, construct a PRF distinguisher.
Proof.

Assume $A$ distinguishes pseudorandom $F$ with advantage $d > \text{negl.}$

1. If $F$ pseudorandom, $\Pr[A \text{ succeeds}] = 1/2 + d$

2. If $F$ is a truly random function $f$: $A$ makes $Q$ oracle queries.
   - If nonce $r_c$ used in challenge is repeated, $A$ learns value of $f(r_c)$ and succeeds with probability 1.

   $$\Pr[r_c \text{ repeated across oracle queries}] \leq \frac{Q}{2^n}$$

   - If $r_c$ not used in challenge, no information: $\Pr[\text{success}] = 1/2$

   $$\Pr[A \text{ succeeds}] \leq 1/2 + \frac{Q}{2^n}$$

   $$|\Pr[D | F] - \Pr[D | F_k]| = |1/2 + d - (1/2 + \frac{Q}{2^n})| = d - \frac{Q}{2^n} > \text{negl.}$$
Using stream ciphers in a CPA-secure way

Augment stream cipher with an initialization vector or IV.

- \( \text{Enc}_k(m) = (IV, G(k, IV) \oplus m) \)

For this to be secure, \( G(k, IV) \) needs to be pseudorandom when IV is known.

Insecure if IV is ever reused.

**WEP insecurity.** WEP is broken in multiple ways: it uses a 24-bit IV, which repeats with 50% probability after 5,000 packets.
If we use a block cipher in counter mode with a randomized starting value \( r \), this is CPA-secure.

\[
\text{Enc}_k(m) = (r, F_k(r) \oplus m[0], F_k(r + 1) \oplus m[1], \ldots, F_k(r + \ell) \oplus m[\ell])
\]

\[
\text{Dec}_k(c) = (F_k(r) \oplus c[0], F_k(r + 1) \oplus c[1], \ldots, F_k(r + \ell) \oplus c[\ell])
\]

The value \( r \) is the IV.

This is an ok choice of mode of operation for AES.
Cipher block chaining (CBC) mode

1. IV has same length as block length.
2. \( c_i = \text{Enc}_k(c_{i-1} \oplus m_i) \)
3. Output \((\text{IV}, c_0, c_1, c_2, \ldots)\).

IV should be random.

CBC mode is CPA-secure, but suffers from implementation vulnerabilities that you get to break in HW 3.
• HW 2 is due before class in 1 week, October 20.