Previously on COS 433...
Security Experiment/Game
(One-time setting)
Definition: \((\text{Enc}, \text{Dec})\) has \((t, \varepsilon)\)-ciphertext indistinguishability if, for all \(\mathcal{A}\) running in time at most \(t\)

\[
\left| \Pr[1 \leftarrow \text{IND-Exp}_0(\mathcal{A})] - \Pr[1 \leftarrow \text{IND-Exp}_1(\mathcal{A})] \right| \leq \varepsilon
\]
Construction with $|k| \ll |m|$

Idea: use OTP, but have key generated by some expanding function $G$
What Do We Want Out of $G$?

Definition: $G: \{0,1\}^\lambda \rightarrow \{0,1\}^n$ is a $\lambda$-secure pseudorandom generator (PRG) if:

- $n > \lambda$
- $G$ is deterministic
- For all running in time at most $t$,
  
  $\Pr[ (G(s))=1 : s \leftarrow \{0,1\}^\lambda ] - \Pr[ (x)=1 : x \leftarrow \{0,1\}^n ] \leq \varepsilon$
Reminder: Kerckhoffs’s Principle

Kerckhoffs’s Principle: A cryptosystem should be secure even if everything about the system, except the key, is public knowledge.

Applies to any crypto object we’ll see in this course

For PRGs, the “key” is just the input to the function
Secure PRG $\rightarrow$ Ciphertext Indistinguishability

$K = \{0,1\}^\lambda$
$M = \{0,1\}^n$
$C = \{0,1\}^n$

$Enc(k,m) = PRG(k) \oplus m$
$Dec(k,c) = PRG(k) \oplus c$
Security?

Intuitively, security is obvious:
• $\text{PRG}(k)$ "looks" random, so should completely hide $m$
• However, formalizing this argument is non-trivial.

Solution: reductions
• Assume toward contradiction an adversary for the encryption scheme, derive an adversary for the PRG
Security

Assume towards contradiction that there is a such that

\[ \Pr[W_0] - \Pr[W_1] \geq \epsilon, \text{ non-negligible} \]

\[ W_b : b' = 1 \text{ in IND-Exp}_b \]
Security

Use to build . will run as a subroutine, and pretend to be

\[ m_0, m_1 \in \mathcal{M} \]

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

\[ c \leftarrow x \oplus m_b \]

\[ 1 \oplus b \oplus b' \]
Security

Case 1: $x = \text{PRG}(s)$ for a random seed $s$

- “sees” $\text{IND-Exp}_b$ for a random bit $b$
Security

Case 1: $x = \text{PRG}(s)$ for a random seed $s$

- \( \text{\# "sees" IND-Exp}_b \) for a random bit $b$
- $\Pr[1 \oplus b \oplus b' = 1] = \Pr[b = b']$
  
  \[
  \begin{align*}
  &\quad = \frac{1}{2} \Pr[b' = 1 \mid b = 1] \\
  &\quad \quad + \frac{1}{2} \left( 1 - \Pr[b' = 1 \mid b = 0] \right) \\
  &\quad = \frac{1}{2} (1 + \Pr[W_0] - \Pr[W_1]) \\
  &\quad = \frac{1}{2} (1 \pm \varepsilon)
  \end{align*}
  \]
Security

Case 2: $\mathbf{x}$ is truly random

- $\mathbf{x}$ “sees” OTP encryption

$m_0, m_1 \in M_\lambda$

$b \leftarrow \{0,1\}$

$x \leftarrow \{0,1\}^n$

$c \leftarrow x \oplus m_b$

$b'$
Security

Case 2: $\times$ is truly random

- “sees” OTP encryption
- Therefore $\Pr[b'=1 \mid b=0] = \Pr[b'=1 \mid b=1]$
- $\Pr[1\oplus b \oplus b'=1] = \Pr[b=b']$
  
  \[
  = \frac{1}{2} \Pr[b'=1 \mid b=1] \\
  + \frac{1}{2} (1 - \Pr[b'=1 \mid b=0])
  \]

  \[
  = \frac{1}{2}
  \]
Security

Putting it together:

• \(\Pr[\text{(G(s))}=1:s\leftarrow\{0,1\}^\lambda] = \frac{1}{2}(1 \pm \varepsilon(\lambda))\)

• \(\Pr[\text{(x)}=1:x\leftarrow\{0,1\}^n] = \frac{1}{2}\)

• Absolute Difference: \(\frac{1}{2}\varepsilon, \Rightarrow\) Contradiction!
Security

Thm: If $G$ is a $(t+t',\varepsilon/2)$-secure PRG, then $(\text{Enc},\text{Dec})$ is has $(t,\varepsilon)$-ciphertext indistinguishability, where $t'$ is the time to:

- Flip a random bit $b$
- XOR two $n$-bit strings
Security

**Thm:** If $G$ is a $(t+\text{poly},\epsilon/2)$-secure PRG, then $(\text{Enc, Dec})$ is has $(t,\epsilon)$-ciphertext indistinguishability
An Alternate Proof: Hybrids

Idea: define sequence of “hybrid” experiments “between” $\text{IND-Exp}_0$ and $\text{IND-Exp}_1$

In each hybrid, make small change from previous hybrid

Hopefully, each small change is undetectable

Using triangle inequality, overall change from $\text{IND-Exp}_0$ and $\text{IND-Exp}_1$ is undetectable
An Alternate Proof: Hybrids

Hybrid 0: IND-Exp_0

\[
\begin{align*}
& \text{m}_0, \text{m}_1 \in \mathcal{M} \\
& \text{c} \leftarrow G(k) \oplus \text{m}_0
\end{align*}
\]
An Alternate Proof: Hybrids

Hybrid 1:

\[ m_0, m_1 \in M \]

\[ x \leftarrow \{0,1\}^n \]

\[ c \leftarrow x \oplus m_0 \]

\[ b' \]
An Alternate Proof: Hybrids

Hybrid 2:

$m_0, m_1 \in \mathcal{M}$

$x \leftarrow \{0,1\}^n$

$c \leftarrow x \oplus m_1$

$b'$
An Alternate Proof: Hybrids

Hybrid 3: \text{IND-Exp}_1

\begin{align*}
m_0, m_1 &\in M \\
k &\leftarrow K \\
c &\leftarrow G(k) \oplus m_1
\end{align*}
An Alternate Proof: Hybrids

<table>
<thead>
<tr>
<th>Pr[b'\mathbf{=}1 : \text{IND-Exp}_0] - Pr[b'\mathbf{=}1 : \text{IND-Exp}_1]</th>
</tr>
</thead>
<tbody>
<tr>
<td>=</td>
</tr>
<tr>
<td>≤</td>
</tr>
<tr>
<td>+</td>
</tr>
<tr>
<td>+</td>
</tr>
</tbody>
</table>

If |Pr[b'\mathbf{=}1:\text{IND-Exp}_0]-Pr[b'\mathbf{=}1:\text{IND-Exp}_1]| ≥ \varepsilon,

Then for some i=0,1,2,

|Pr[b'\mathbf{=}1:\text{Hyb } i]-Pr[b'\mathbf{=}1:\text{Hyb } i+1]| ≥ \varepsilon/3
An Alternate Proof: Hybrids

Suppose \( \epsilon \) distinguishes **Hybrid 0** from **Hybrid 1** with advantage \( \epsilon/3 \)

\[
\begin{align*}
\text{k} & \leftarrow K \\
\text{m}_0, \text{m}_1 \in \mathcal{M} & \quad \text{c} \leftarrow G(k) \oplus \text{m}_0 \\
\downarrow & \quad \downarrow \\
\text{b}' & \quad \text{b}'
\end{align*}
\]

\[
\begin{align*}
\text{x} & \leftarrow \{0,1\}^n \\
\text{m}_0, \text{m}_1 \in \mathcal{M} & \quad \text{c} \leftarrow x \oplus \text{m}_0 \\
\downarrow & \quad \downarrow \\
\text{b}' & \quad \text{b}'
\end{align*}
\]
An Alternate Proof: Hybrids

Suppose \( \mathcal{A} \) distinguishes Hybrid 0 from Hybrid 1 with advantage \( \varepsilon/3 \) \( \Rightarrow \) Construct

\[ m_0, m_1 \in M \]

\[ c \]

\[ b' \]

(\( \text{either } G(s) \text{ or truly random} \))
An Alternate Proof: Hybrids

Suppose $\hat{\mathcal{A}}$ distinguishes Hybrid 0 from Hybrid 1 with advantage $\varepsilon/3 \implies$ Construct

If $\mathcal{B}$ is given $G(s)$ for a random $s$, $\hat{\mathcal{A}}$ sees Hybrid 0
If $\mathcal{B}$ is given $x$ for a random $x$, $\hat{\mathcal{A}}$ sees Hybrid 1

Therefore, advantage of $\mathcal{B}$ is equal to advantage of $\hat{\mathcal{A}}$ which is at least $\varepsilon/3 \implies$ Contradiction!
An Alternate Proof: Hybrids

Suppose \( \hat{\Pi} \) distinguishes Hybrid 1 from Hybrid 2 with advantage \( \epsilon/3 \)

\[
\begin{align*}
\text{Hybrid 1:} & \quad x \leftarrow \{0,1\}^n \quad m_0, m_1 \in M \\
\text{Hybrid 2:} & \quad x \leftarrow \{0,1\}^n \quad m_0, m_1 \in M \\
\end{align*}
\]
An Alternate Proof: Hybrids

Suppose $\delta$ distinguishes Hybrid 1 from Hybrid 2
with advantage $\epsilon = \frac{1}{3}$

Impossible by OTP security
An Alternate Proof: Hybrids

Suppose distinguishes **Hybrid 2** from **Hybrid 3** with advantage $\frac{\varepsilon}{3}$

\[ x \leftarrow \{0,1\}^n \quad k \leftarrow K \]

\[ m_0, m_1 \in M \quad c \leftarrow x \oplus m_1 \]

\[ m_0, m_1 \in M \quad c \leftarrow G(k) \oplus m_1 \]

Proof essentially identical to Hybrid 0/Hybrid 1 case
How do we build PRGs?
Linear Feedback Shift Registers

In each step,
• Last bit of state is removed and outputted
• Rest of bits are shifted right
• First bit is XOR of subset of remaining bits
Linear Feedback Shift Registers

In each step,
- last bit of state is removed and outputted
- Rest of bits are shifted right
- First bit is XOR of subset of remaining bits
Linear Feedback Shift Registers

In each step,
- last bit of state is removed and outputted
- Rest of bits are shifted right
- First bit is XOR of subset of remaining bits
Linear Feedback Shift Registers

In each step,
• last bit of state is removed and outputted
• Rest of bits are shifted right
• First bit is XOR of subset of remaining bits
Linear Feedback Shift Registers

In each step,
- last bit of state is removed and outputted
- Rest of bits are shifted right
- First bit is XOR of subset of remaining bits
Linear Feedback Shift Registers

In each step,
• last bit of state is removed and outputted
• Rest of bits are shifted right
• First bit is XOR of subset of remaining bits
Linear Feedback Shift Registers

In each step,
- last bit of state is removed and outputted
- Rest of bits are shifted right
- First bit is XOR of subset of remaining bits
Linear Feedback Shift Registers

In each step,
- last bit of state is removed and outputted
- Rest of bits are shifted right
- First bit is XOR of subset of remaining bits
Linear Feedback Shift Registers

In each step,
• last bit of state is removed and outputted
• Rest of bits are shifted right
• First bit is XOR of subset of remaining bits
Linear Feedback Shift Registers

Are LFSR’s secure PRGs?
Linear Feedback Shift Registers

Are LFSR’s secure PRGs? No!

First $n$ bits of output = initial state

Write $x = x_1, \ldots, x_n, \ x'$

Initialize LFSB to have state $x_1, \ldots, x_n$

Run LFSB for $|x|$ steps, obtaining $y$

Check if $y = x$
PRGs should be Unpredictable

More generally, it should be hard, given some bits of output, to predict subsequent bits

**Definition:** \( G \) is \((t,p,\epsilon)\)-unpredictable if, for all running in time at most \( t \),

\[
\left| \Pr[G(s)_{p+1} \leftarrow (G(s)_{[1,p]})] - \frac{1}{2} \right| \leq \epsilon
\]
PRGs should be Unpredictable

More generally, it should be hard, given some bits of output, to predict subsequent bits

Theorem: $G$ is unpredictable iff it is pseudorandom
Proof

Pseudorandomness $\rightarrow$ Unpredictability

Assume towards contradiction s.t.

$$\left| \Pr[G(s)_{p+1} \leftarrow (G(s)_{[1,p]})] - \frac{1}{2} \right| > \varepsilon$$
Proof

Pseudorandomness $\rightarrow$ Unpredictability

Construct

$1 \oplus b \oplus x_{p+1}$
Proof

Pseudorandomness \( \rightarrow \) Unpredictability

Analysis:

- If \( x \) is random, \( \Pr[1 \oplus b \oplus x_{p+1} = 1] = \frac{1}{2} \)
- If \( x \) is pseudorandom,
  \[
  \Pr[1 \oplus b \oplus x_{p+1} = 1] = \Pr[G(s)_{p+1} \leftarrow (G(s)_{[1,p]})] > (\frac{1}{2} + \varepsilon) \text{ or } < (\frac{1}{2} - \varepsilon)
  \]
Proof

Unpredictability $\rightarrow$ Pseudorandomness

Assume towards contradiction $\exists s$ s.t.

$$\left| \Pr[(G(s)) = 1 : s \leftarrow \{0,1\}^n] - \Pr[(x) = 1 : x \leftarrow \{0,1\}^n] \right| > \varepsilon$$
Proof

Unpredictability $\rightarrow$ Pseudorandomness

Hybrids:

$H_i$: $x_{[1,i]} \leftarrow G(s), \ x_{[i+1,t]} \leftarrow \{0,1\}^{t-i}$

$H_0$: truly random $x$

$H_t$: pseudorandom $t$
Proof

Unpredictability $\rightarrow$ Pseudorandomness

Hybrids:

$H_i: x_{[1,i]} \leftarrow G(s), x_{[i+1,t]} \leftarrow \{0,1\}^{t-i}$

$$\left| \Pr[\hat{x}(x)=1:x \leftarrow H_s] - \Pr[\hat{x}(x)=1:x \leftarrow H_0] \right| > \varepsilon$$

Let $q_i = \Pr[\hat{x}(x)=1:x \leftarrow H_i]$
Proof

Unpredictability $\rightarrow$ Pseudorandomness

Hybrids:

$H_i: x_{[1,i]} \leftarrow G(s), x_{[i+1,t]} \leftarrow \{0,1\}^{t-i}$

$|q_t - q_0| > \varepsilon$

Let $q_i = \Pr[\text{(x) = 1: } x \leftarrow H_i]$
Proof

Unpredictability $\rightarrow$ Pseudorandomness

Hybrids:
$$ H_i : x_{[1,i]} \leftarrow G(s), \ x_{[i+1,t]} \leftarrow \{0,1\}^{t-i} $$

By triangle inequality, there must exist an $i$ s.t.
$$ |q_i - q_{i-1}| > \frac{\varepsilon}{t} $$

Can assume wlog that
$$ q_i - q_{i-1} > \frac{\varepsilon}{t} $$
Proof

Unpredictability $\rightarrow$ Pseudorandomness

Construct

\[ y = G(s)_{[1,i-1]} \]

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

\[ y' \leftarrow \{0,1\}^{t-i} \]

\[ x = y || b || y' \]

\[ b' \]

\[ 1 \oplus b \oplus b' \]
Proof

Unpredictability $\rightarrow$ Pseudorandomness

Analysis:
• If $b = G(s)_i$, then $\overline{H}_i$ sees $H_i$

  $\Rightarrow$ $\overline{H}_i$ outputs $1$ with probability $q_i$

  $\Rightarrow$ $\overline{H}_i$ outputs $b = G(s)_i$ with probability $q_i$
Proof

Unpredictability $\rightarrow$ Pseudorandomness

Analysis:
• If $b = 1 \oplus G(s)_i$, then
  Define $q_i'$ as $\Pr[\text{outputs 1}]$

  $\frac{1}{2}(q_i' + q_i) = q_{i-1} \Rightarrow q_i' = 2q_{i-1} - q_i$

  $\Rightarrow$ outputs $G(s)_{[1,i]}$ with probability

  $1 - q_i' = 1 + q_i - 2q_{i-1}$
Proof

Unpredictability $\rightarrow$ Pseudorandomness

Analysis:

- $\Pr[\text{outputs } G(s)_i]$
  
  $= \frac{1}{2} (q_i) + \frac{1}{2} (1 + q_i - 2q_{i-1})$

  $= \frac{1}{2} + q_i - q_{i-1}$

  $> \frac{1}{2} + \frac{\epsilon}{t}$
Linearity
Linearity

LFSR’s are linear:

\[
\text{state'} = (0 0 0 0 1) \oplus \text{state}
\]

\[
\text{output} = (0 0 0 0 0 1) \oplus \text{state}
\]
Linearity

LFSR’s are linear:
• Each output bit is a linear function of the initial state (that is, $G(s) = A \cdot s \pmod{2}$)

Any linear $G$ cannot be a PRG
• Can check if $x$ is in column-span of $A$ using linear algebra
Introducing Non-linearity

Non-linearity in the output:

Non-linear feedback:
LFSR period

Period = number of bits before state repeats

After one period, output sequence repeats

Therefore, should have extremely long period
• Ideally almost $2^\lambda$
• Possible to design LFSR’s with period $2^\lambda - 1$
Hardware vs Software

PRGs based on LFSR’s are very fast in hardware

Unfortunately, not easily amenable to software
RC4

Fast software based PRG

Resisted attack for several years

No longer considered secure, but still widely used
RC4

State = permutation on $[256]$ plus two integers
• Permutation stored as 256-byte array $S$

Init(16-byte $k$):
• For $i=0,\ldots,255$
  $S[i] = i$
• $j = 0$
• For $i=0,\ldots,255$
  $j = j + S[i] + k[i \text{ mod } 16] \pmod{256}$
  Swap $S[i]$ and $S[j]$
• Output $(S,0,0)$
RC4

GetBits(S,i,j):
• i++ (mod 256)
• j+= S[i] (mod 256)
• Swap S[i] and S[j]
• t = S[i] + S[j] (mod 256)
• Output (S,i,j), S[t]

New state Next output byte
Insecurity of RC4

Second byte of output is slightly biased towards 0
• $\text{Pr}[\text{second byte} = 0^8] \approx \frac{2}{256}$
• Should be $\frac{1}{256}$

Means RC4 is not secure according to our definition
• outputs 1 iff second byte is equal to $0^8$
• Advantage: $\approx \frac{1}{256}$

Not a serious attack in practice, but demonstrates some structural weakness
Insecurity of RC4

Possible to extend attack to actually recover the input $k$ in some use cases

- The seed is set to $(IV, k)$ for some initial value $IV$
- Encrypt messages as $RC4(IV,k) \oplus m$
- Also give $IV$ to attacker
- Cannot show security assuming RC4 is a PRG

Can be used to completely break WEP encryption standard
Extending the Stretch of a PRG

Suppose you have a fixed-stretch PRG $G$

- Better yet, a PRG that expands by a single bit

$G: \{0,1\}^\lambda \rightarrow \{0,1\}^{\lambda+1}$

Construct a PRG $G'$ of arbitrary output length
Extending the Stretch of a PRG
Security Proof

Assume towards contradiction ...

Define hybrids...
Security Proof

$H_0: \{0,1\}^\lambda$

\[
\begin{array}{c}
\text{seed} \\
G \\
\text{state}_1 \\
G \\
\text{state}_2 \\
G \\
\text{state}_3 \\
G \\
\cdots
\end{array}
\]
Security Proof

$H_1:\{0,1\}^\lambda \overset{\text{state}_1}{\rightarrow} G \overset{\text{state}_2}{\rightarrow} G \overset{\text{state}_3}{\rightarrow} G \cdots$
Security Proof

$H_2$: 

$\{0,1\} \rightarrow \{0,1\}^\lambda \rightarrow \text{state}_2 \rightarrow \text{state}_3 \rightarrow \ldots$
Security Proof

$H_+ : \{0,1\} \rightarrow \{0,1\} \rightarrow \{0,1\} \rightarrow \{0,1\} \rightarrow \cdots$
Security Proof

$H_0$ corresponds to pseudorandom $\mathbf{x}$

$H_t$ corresponds to truly random $\mathbf{x}$

Let $q_i = \Pr[\mathcal{H}(x)=1:x \leftarrow H_i]$

By assumption, $|q_t - q_0| > \varepsilon$

$\Rightarrow \exists i \text{ s.t. } |q_i - q_{i-1}| > \varepsilon/t$
Security Proof
Security Proof

Analysis

• If $y = G(s)$, then $\mathcal{A}$ sees $H_{i-1}$

  $\Rightarrow \Pr[\mathcal{A} \text{ outputs 1}] = q_{i-1}$

  $\Rightarrow \Pr[\mathcal{C} \text{ outputs 1}] = q_{i-1}$

• If $y$ is random, then $\mathcal{A}$ sees $H_i$

  $\Rightarrow \Pr[\mathcal{A} \text{ outputs 1}] = q_i$

  $\Rightarrow \Pr[\mathcal{C} \text{ outputs 1}] = q_i$
Summary

Stream ciphers = secure encryption for arbitrary length, number of messages
(though we did not completely prove it)

However, implementation difficulties due to having to maintaining state
Reminders

Project 1 part 1 Due Tomorrow

HW2 will be released tonight