The slides are loosely based on those of Prof. Mihir Bellare, UC San Diego.
Agenda

1. Defining PRF Security

2. Birthday Attack
Recall

<table>
<thead>
<tr>
<th>Possible Properties</th>
<th>Necessary</th>
<th>Sufficient</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security against key recovery</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Hard to find $M$ given $C \leftarrow E_K(M)$</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Want:** a single “master” property that is sufficient to ensure security of common usage of blockcipher.
An Analogy: Turing Test

What does it mean for a machine to be “intelligent”?

Possible Answers

<table>
<thead>
<tr>
<th>Possible Answers</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>It can be happy</td>
<td>But no such list is satisfactory</td>
</tr>
<tr>
<td>It recognizes pictures</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>
An Analogy: Turing Test

A

interaction

b

Man (0) or Machine (1)?
# Real versus Ideal

<table>
<thead>
<tr>
<th>Notion</th>
<th>Real object</th>
<th>Ideal object</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intelligence</td>
<td><img src="image" alt="Robot" /></td>
<td><img src="image" alt="Man" /></td>
</tr>
<tr>
<td>PRF</td>
<td>$E_K$</td>
<td>Random function</td>
</tr>
</tbody>
</table>
Informal View of PRF Security

$$E : \{0, 1\}^k \times \{0, 1\}^n \rightarrow \{0, 1\}^n$$

Sample random $f : \{0, 1\}^n \rightarrow \{0, 1\}^n$

$K \leftarrow \mathcal{K}$

Adversary doesn’t know $K$ or $f$
Defining Random Function: Lazy Sampling

**Want:** a random function $f : \{0, 1\}^n \to \{0, 1\}^m$

- Pick a fresh random answer for a new query, and remember the answer.
Defining Random Function: Lazy Sampling

**Want:** a random function $f : \{0, 1\}^n \rightarrow \{0, 1\}^m$

- Pick a fresh random answer for a new query, and remember the answer.
Defining Random Function: Lazy Sampling

**Want:** a random function \( f : \{0, 1\}^n \rightarrow \{0, 1\}^m \)

- Pick a fresh random answer for a new query, and remember the answer.
Defining Random Function: Lazy Sampling

**Want:** a random function $f : \{0, 1\}^n \rightarrow \{0, 1\}^m$

- Pick a fresh random answer for a new query, and remember the answer
Want: a random function $f : \{0, 1\}^n \rightarrow \{0, 1\}^m$
**Putting Things in Code**

**Game** Real\(_E\)

**procedure** Initialize()

\[ K \leftrightarrow K \]

**procedure** Fn\( (M) \)

return \( E_K(M) \)

---

**Game** Rand\(_E\)

string array \( T = \{\} \) // Global variable

**procedure** Fn\( (M) \)

If \( T[M] = \perp \) then \( T[M] \leftrightarrow \{0, 1\}^n \)

return \( T[M] \)

---

**Adv\(_E^{prf} (A) = Pr[\text{Real}_E^A \Rightarrow 1] - Pr[\text{Rand}_E^A \Rightarrow 1]**
Practicing PRF Attack

\[ E_K(M) = M \oplus K \]

\[ E_K(M) = \pi(M \oplus K) \]

Public permutation

\[ \pi, \pi^{-1} \text{ are public} \]
Practicing PRF Attack

\[ E_K(M) = AES_M(K) \]

Two-round Feistel
1. Defining PRF Security

2. Birthday Attack
Birthday Problem

Fact: For $q \leq \sqrt{2N}$, 

$$\frac{q(q-1)}{4N} \leq C(N, q) \leq \frac{q(q-1)}{2N}$$
Birthday Attack on PRF Security

distinct $M_1, \ldots, M_q$

$E_K$

distinct $C_1, \ldots, C_q$

distinct $M_1, \ldots, M_q$

$f$

random $C_1, \ldots, C_q$
Birthday Attack on PRF Security

\[ E : \{0, 1\}^k \times \{0, 1\}^n \rightarrow \{0, 1\}^n \]

Input $A$ requests $M_1, \ldots, M_q$ distinct

Output 1 if $C_1, \ldots, C_q$ are distinct

\[ \text{Adv}^\text{prf}_E (A) = C(2^n, q) \approx \frac{q^2}{2^n} \]

Need $2^{n/2}$ queries to break PRF security

<table>
<thead>
<tr>
<th>Blockcipher</th>
<th>$n$</th>
<th>$2^{n/2}$</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>DES, 2DES, 3DES</td>
<td>64</td>
<td>$2^{32}$</td>
<td>Insecure</td>
</tr>
<tr>
<td>AES</td>
<td>128</td>
<td>$2^{64}$</td>
<td>Secure</td>
</tr>
</tbody>
</table>