PSEUDORANDOM FUNCTION

VIET TUNG HOANG

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>It can be happy</th>
<th>But no such list is satisfactory</th>
</tr>
</thead>
<tbody>
<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="image1.png" alt="Robot" /></td>
<td><img src="image2.png" alt="Human" /></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 \( f : \{0, 1\}^n \rightarrow \{0, 1\}^m \)

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 \( f : \{0, 1\}^n \to \{0, 1\}^m \)

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

**Game** \( \text{Real}_E \)

**procedure** Initialize()

\[ K \xleftarrow{\$} \mathcal{K} \]

**procedure** \( \text{Fn}(M) \)

return \( E_K(M) \)

---

**Game** \( \text{Rand}_E \)

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

**procedure** \( \text{Fn}(M) \)

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

return \( T[M] \)

---

\[ \text{Adv}_{E}^{\text{prf}}(A) = \Pr[\text{Real}_E^A \Rightarrow 1] - \Pr[\text{Rand}_E^A \Rightarrow 1] \]
Exercise: PRF Attacks

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

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

\[ \pi, \pi^{-1} \text{ are public} \]
Easy to Break PRF Security After Key Recovery

KR attack

$K$

$E_K(M) \not\equiv C$

Yes

No

1

0

new msg, not used in KR attack
PRF Security

Key Recovery Security
Exercise: PRF Attacks

\[ E_K(M) = AES_K(M) || AES_K(M^{\overline{}}) \]
1. Defining PRF Security

2. Birthday Attack
Birthday Problem

\[ y_1, \ldots, y_q \rightarrow \{1, \ldots, N\} \]

\[ C(N, q) = \Pr[y_1, \ldots, y_q \text{ not distinct}] \]

**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

\[
\begin{array}{cc}
\text{distinct } M_1, \ldots, M_q & \text{distinct } M_1, \ldots, M_q \\
\downarrow & \downarrow \\
E_K & f \\
\downarrow & \downarrow \\
\text{distinct } C_1, \ldots, C_q & \text{random } C_1, \ldots, C_q
\end{array}
\]
Birthday Attack on PRF Security

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

distinct \( M_1, \ldots, M_q \)

\[ A \rightarrow \text{Fn} \]

Output 1 if \( C_1, \ldots, C_q \) are distinct

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

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>
Does It Matter In Practice?

Sweet32: Birthday Attacks on 64-bit Blockciphers in TLS and OpenVPN

[ Bhargavan, Leurent 16 ]

HTTPS encryption via 3DES

Recover cookie after capturing 785GB