Lecture 4 – Pseudorandom Functions

COSC-260 Codes and Ciphers

Adam O’Neill

adapted from http://cseweb.ucsd.edu/~mihir/cse107/
What is a “good” blockcipher?

We want to define a notion of a “good” blockcipher, where “good” means natural uses of the blockcipher are secure.
What is a “good” blockcipher?

We want to define a notion of a “good” blockcipher, where “good” means natural uses of the blockcipher are secure. One idea is to list requirements:
What is a “good” blockcipher?

We want to define a notion of a “good” blockcipher, where “good” means natural uses of the blockcipher are secure.

One idea is to list requirements:

• Key recovery is hard.
What is a “good” blockcipher?

We want to define a notion of a “good” blockcipher, where “good” means natural uses of the blockcipher are secure.

One idea is to list requirements:

• Key recovery is hard.
• Message recovery is hard.
Analogy to Intelligence

What if we want to define the notion of “intelligent” for a computer program?
Analogy to Intelligence

What if we want to define the notion of "intelligent" for a computer program?
Again, one idea is to list requirements:
Analogy to Intelligence

What if we want to define the notion of "intelligent" for a computer program?
Again, one idea is to list requirements:
- It can be happy.
Analogy to Intelligence

What if we want to define the notion of “intelligent” for a computer program?
Again, one idea is to list requirements:

• It can be happy.
• It can multiply numbers
Analogy to Intelligence

What if we want to define the notion of “intelligent” for a computer program?

Again, one idea is to list requirements:

• It can be happy.
• It can multiply numbers
• ... but only small numbers.
A program is “intelligent” if its input/output behavior is indistinguishable from that of a human.
The Turing Test

Game:
- Put tester in room 0 and let it interact with object behind wall
- Put tester in room 1 and let it interact with object behind wall
- Now ask tester: which room was which?

The measure of “intelligence” of $P$ is the extent to which the tester fails.
## The Analogy

<table>
<thead>
<tr>
<th>Notion</th>
<th>Real object</th>
<th>Ideal object</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intelligence</td>
<td>Program</td>
<td>Human</td>
</tr>
<tr>
<td>PRF</td>
<td>Block cipher</td>
<td>?</td>
</tr>
</tbody>
</table>

- good block cipher

*Handwritten notes:* good block cipher
Random Functions

A random function with $L$-bit outputs is implemented by the following box $\text{Fn}$, where $T$ is initially ⊥ everywhere:

\[
\begin{align*}
\text{Caller} & \quad \xrightarrow{x} \quad T[x] \\
& \quad \xleftarrow{T[x]} \\
\text{Fn} & \\
\text{If } T[x] = \bot \text{ then} \\
T[x] & \leftarrow \{0, 1\}^L \\
\text{Return } T[x]
\end{align*}
\]
Random Functions

A random function with $L$-bit outputs is implemented by the following box $F_n$, where $T$ is initially $\perp$ everywhere:

\[
F_n
\]

\[
\begin{array}{c}
\text{Caller} \\
\text{if } T[x] \neq \perp \text{ then} \\
T[x] \leftarrow \{0, 1\}^L \\
\text{Return } T[x]
\end{array}
\]

Henceforth we make a rule:

- A prf-adversary is not allowed to repeat an oracle query.

Then our game is:

\[
\begin{array}{c}
\text{Game Rand}_{\text{Range}(F)} \\
\text{procedure } F_n(x) \\
T[x] \leftarrow \text{Range}(F) \\
\text{Return } T[x]
\end{array}
\]
Function Families

A family of functions $F : \text{Keys}(F) \times \text{Dom}(F) \rightarrow \text{Range}(F)$ is a two-argument map. For $K \in \text{Keys}(F)$ we let $F_K : \text{Dom}(F) \rightarrow \text{Range}(F)$ be defined by

$$\forall x \in \text{Dom}(F) : F_K(x) = F(K, x)$$

Examples

- Any blockcipher is a family of functions with $\text{Keys}(F) = \{0, 1\}^l$
  $\text{Dom}(F) = \text{Range}(F) = \{0, 1\}^{2l}$
Let $F: \text{Keys}(F) \times \text{Dom}(F) \rightarrow \text{Range}(F)$ be a family of functions.

Associated to $F$, $A$ are the probabilities

\[
P \left[ \text{Real}_F^A \Rightarrow 1 \right] \quad \mid \quad P \left[ \text{Rand}_{\text{Range}(F)}^A \Rightarrow 1 \right]
\]

that $A$ outputs 1 in each world. The advantage of $A$ is

\[
\text{Adv}^{\text{prf}}_F(A) = P \left[ \text{Real}_F^A \Rightarrow 1 \right] - P \left[ \text{Rand}_{\text{Range}(F)}^A \Rightarrow 1 \right]
\]
Advantage Interpretation

\[ \text{Adv}^\text{prf}_F (A) = \Pr \left[ \text{Real}_F^A \Rightarrow 1 \right] - \Pr \left[ \text{Rand}^A_{\text{Range}(F)} \Rightarrow 1 \right] \]

A “large” (close to 1) advantage means
- \( A \) is doing well
- \( F \) is not secure

A “small” (close to 0 or \( \leq 0 \)) advantage means
- \( A \) is doing poorly
- \( F \) resists the attack \( A \) is mounting
PRF Security

\[ \text{Adv}_{F}^{\text{prf}}(A) = \Pr \left[ \text{Real}^A_F \Rightarrow 1 \right] - \Pr \left[ \text{Rand}^A_{\text{Range}(F)} \Rightarrow 1 \right] \]

**Security:** \( F \) is a (secure) PRF if \( \text{Adv}_{F}^{\text{prf}}(A) \) is “small” for ALL \( A \) that use “practical” amounts of resources.

**Insecurity:** \( F \) is insecure (not a PRF) if there exists \( A \) using “few” resources that achieves “high” advantage.
Examples

Suppose \( E : \{0,1\}^l \times \{0,1\}^l \rightarrow \{0,1\}^l \) is such that \( E_k(x) = k \oplus x \), \( \forall k, x \in \{0,1\}^l \).

Is \( E \) a secure PRF?

Adversary \( A \):

\( \text{No!} \)

\( Y_1 \leftarrow \text{Fn}(0^l) \) if \( Y_2 = 0^l \) then ret 1

\( Y_2 \leftarrow \text{Fn}(Y_1) \) else ret 0

\( \text{might repeat query (if } Y_1 = 0^l \text{) but otherwise} \)
To analyze $\Pr[REAL^A \Rightarrow 1]$

Note $Y_i = K \oplus 0^l = K$

$Y_i = Y_i \oplus K = K \oplus K = 0^l$

$\forall K \in \{0,1\}^l$

hence $\Pr[REAL^A \Rightarrow 1] = 1$

To analyze $\Pr[\text{RAND}_{s^0,s^1}^A \Rightarrow 1]$

( Ignore possible repeat query)

$\Pr[\gamma_2 = 0^l] = 2^{-l} \text{ hence } \Pr[\text{RAND}_{s^0,s^1}^A \Rightarrow 1] = 2^{-l}$
Birthday Attack

What's the best generic attack on a blockcipher under the prf notion?

- We've seen exhaustive key-search takes $O(2^k)$ time.

- Now we look at the birthday attack.

A q:
Query first q points to $F_n$.
If there is any collision in the outputs return $\bot$, else set $i$.
Birthday Problem

Pick $y_1, \ldots, y_q \leftarrow \{1, \ldots, N\}$ and let

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

Let

$$0.3 \cdot \frac{q(q-1)}{N} \leq C(N, q) \leq 0.5 \cdot \frac{q(q-1)}{N}$$

where the lower bound holds for $1 \leq q \leq \sqrt{2N}$. 

Fact: Then

$$0.3 \cdot \frac{q(q-1)}{N} \leq C(N, q) \leq 0.5 \cdot \frac{q(q-1)}{N}$$

where the lower bound holds for $1 \leq q \leq \sqrt{2N}$. 

Birthday bounds

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

Fact:

$$0.3 \cdot \frac{q(q-1)}{N} \leq C(N, q) \leq 0.5 \cdot \frac{q(q-1)}{N}$$

where the lower bound holds for $1 \leq q \leq \sqrt{2N}$. 

Mihir Bellare UCSD 40
Analysis

$A \cup_{\mathcal{E}} \text{pref}(A) \geq \frac{q(q-1)}{2^{e+1}}$
Conclusion: If $E : \{0, 1\}^k \times \{0, 1\}^\ell \rightarrow \{0, 1\}^\ell$ is a block cipher, there is an attack on it as a PRF that succeeds in about $2^{\ell/2}$ queries.

Depends on block length, not key length!

<table>
<thead>
<tr>
<th></th>
<th>$\ell$</th>
<th>$2^{\ell/2}$</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>
PRF-Security Implications

PRF-security can be seen as a "master property" for blockciphers that implies all other security properties we want.
PRF-Security Implications

**PRF-security** can be seen as a “master property” for blockciphers that implies all other security properties we want.
PRF-Security Implications

**PRF-security** can be seen as a “master property” for blockciphers that implies all other security properties we want.

E.g., we can show that **PRF-security** implies security against key-recovery.
Reduction Sketch

Suppose we have a KR-adversary $A$. Then we can build a PRF-adversary $B$ as follows:

Adversary $B$:

Run $A$:

When $A$ makes $F_n$ query $X$

$B$ queries $X$ to its own $F_n$ procedure and returns the result.

Until $A$ halts with output $K'$

If $K'$ is a consistent key return 1, else return 0.
Conclusion

• We believe DES, AES are “good” blockciphers in the sense that there is no significantly “better than generic” attacks under the PRF notion.
Conclusion

• We believe DES, AES are “good” blockciphers in the sense that there is no significantly “better than generic” attacks under the PRF notion.

• Generic attacks:
Conclusion

• We believe DES, AES are “good” blockciphers in the sense that there is no significantly “better than generic” attacks under the PRF notion.

• Generic attacks:
  • Exhaustive key-search.
Conclusion

• We believe DES, AES are “good” blockciphers in the sense that there is no significantly “better than generic” attacks under the PRF notion.

• Generic attacks:
  • Exhaustive key-search.
  • Birthday attack.