1. Blockcipher and Key Recovery

2. A Bird’s-Eye View of Real Blockciphers
Blockcipher

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

Key space Domain

efficiently invertible given the key
Blockcipher Usage

Random key $K$ is known to both parties, but not given to adversary $A$
Key-Recovery Attack: Scenario

$C_1 \leftarrow E_K(M_1)$  ...  $C_q \leftarrow E_K(M_q)$

Guess $K$
Modeling Key-Recovery Attack

**Game** $KR_E$

**procedure** Initialize()

\[ K \leftrightarrow \mathcal{K} \]

**procedure** $Enc(M)$

\[ \text{return } E_K(M) \]

**procedure** Finalize($K'$)

\[ \text{return } (K' = K) \]

\[ \text{Adv}_{E}^{kr}(A) = \Pr[KR_E \Rightarrow 1] \]

\[ \text{Adv}_{E}^{kr}(A) \approx 0 \text{ means } A \text{ is doing poorly} \]
Practicing Key-Recovery Attack

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

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

Public permutation

\( \pi, \pi^{-1} \) are public
1. Blockcipher and Key Recovery

2. A Bird’s-Eye View of Real Blockciphers
DES: Parameters and History

- Designed by IBM in 1974
- Used in ATM machines
- Replaced in 2001
Design of DES: Feistel Network

(One-round) Feistel($K, \cdot$)

Inverse of Feistel

Question: How to invert?
Construction of DES

56 bits

$K$

Key scheduler

$K_1$  ...  $K_{16}$

48 bits

$M$

Unkeyed processing

Process($\cdot$)

Feistel($K_1, \cdot$)

... 

Feistel($K_{16}, \cdot$)

Process$^{-1}$(.$)

$C$
Exhaustive Key Search Attack

For $K \in \mathcal{K}$ do

If $E_K(M_i) = C_i$ for every $i \in \{1, \ldots, d\}$ then return $K$

For $E : \{0, 1\}^k \times \{0, 1\}^n \rightarrow \{0, 1\}^n$, if $d > k/n$ then $\text{Adv}^\text{kr}_E(A) \approx 1$
## Exhaustive Key Search Attack on DES

<table>
<thead>
<tr>
<th>Timeline</th>
<th>Source</th>
<th>Attack time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1997</td>
<td>DESCHALL</td>
<td>96 days</td>
</tr>
<tr>
<td>1998</td>
<td>Distributed.NET</td>
<td>41 days</td>
</tr>
<tr>
<td>1998</td>
<td>EFF</td>
<td>56 hours</td>
</tr>
<tr>
<td>1998</td>
<td>Distributed.NET + EFF</td>
<td>22 hours</td>
</tr>
</tbody>
</table>
Incorrect Fix of DES: Double Encryption

112-bit key → prohibitive for exhaustive key search

But there’s a more clever attack!
Meet-in-the-Middle Attack

$M$ $\xrightarrow{E_{K_1}}$ $E_{K_2}$ $\xrightarrow{C}$

$E_{K_1}(M)$ $\xleftarrow{E_{K_2}^{-1}(C)}$
Meet-in-the-Middle Attack

Let $L_1, \ldots, L_N$ be all possible DES keys

$$N = 2^{56}$$

Find $L_i, L_j$ such that $E_{L_i}(M) = E_{L_j}^{-1}(C)$

Can use further testing with Enc to eliminate false positives

By using hashing, can find the matching in $O(N)$ time
The 3DES Constructions

3DES2

3DES3

$E_{K_1}$ $E^{-1}_{K_2}$ $E_{K_1}$

$E_{K_1}$ $E^{-1}_{K_2}$ $E_{K_3}$
Block Size Matters, Too

Birthday attack: $O(2^{n/2})$ time

Distinguish outputs from random

Practical for DES/2DES/3DES

$k$ bits

$n$ bits

$C$

$E$

$M$

$K$
State of the Art: AES

- NIST standard since 2001
- Best known key-recovery attack takes about $2^{126}$ time

$k \in \{128, 192, 256\}$
Security Against Key Recovery Is Not Enough

A Bad Example: Consider the following $E : \{0, 1\}^{128} \times \{0, 1\}^{256} \rightarrow \{0, 1\}^{256}$

$$E_K(M_1 M_2) = AES_K(M_1) || M_2$$

As secure against key recovery as AES

Send half of the message in the clear!
So What Is a Good Blockcipher?

<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.