Admin:

(Pset #1 due Friday
Email to 6857-staff@mit.edu

Guest lecturers next week:
Mon: Victor Costan - Web security
Wed: Srinivasa Devadas - Authentication & PUP's

Today:

DES (Data Encryption Standard)
AES (Advanced Encryption Standard)
Ideal block cipher defn.
Modes of operation:
ECB
CTR
CBC
CFB
IND-CCA security defn
UFE mode
Block ciphers:

\[
\begin{align*}
P & \xrightarrow{\text{key } k} \text{Enc} \xrightarrow{} C \\
\text{plaintext block} & \quad \quad \text{ciphertext block}
\end{align*}
\]

fixed-length $P, C, K$

- DES: $|P| = |C| = 64 \text{ bits}$ $|K| = 56 \text{ bits}$
- AES: $|P| = |C| = 128 \text{ bits}$ $|K| = 128, 192, 256 \text{ bits}$

Use a "mode of operation" to handle variable-length input.
"Data Encryption Standard"
Standardized in 1976. Now deprecated in favor of AES.

"Feistel structure"

16 rounds total

plaintext 64 bits

all 16 round keys derived from 64-bit encryption key (only 56 bits are really used) via "key schedule"

ciphertext

Note: Invertible for any $f$ and any key schedule.

$F$ uses 8 "S-boxes" mapping 6-bits $\rightarrow$ 4 bits nonlinearly.

Key is too short! (Breakable now quite easily by brute-force)

Subject to differential attacks:

$$ M \xrightarrow{\oplus \Delta} M \oplus \Delta $$

$$ k \xrightarrow{\text{DES}} C \xrightarrow{\oplus \delta} C \oplus \delta $$

$2^{47}$ chosen pairs (Biham/Shamir)

Subject to linear attacks:

e.g. if $M_3 \oplus M_{15} \oplus C_2 \oplus K_{14} = 0$ (eqn on bit3) with prob $p = 1/2 + \epsilon$

then need $1/\epsilon^2$ samples to break (Matsui, $2^{43}$ PT/CT pairs)
"Advanced Encryption Standard" (U.S. govt)

Replaces DES

AES "contest" 1997-1999:
15 algorithms submitted: RCE, Mars, Twofish, Rijndael, ...
Winner = Rijndael (by Joan Daemen & Vincent Rijmen, (Belgians))

Specs:
- 128-bit plaintext/ciphertext blocks
- 128, 192, or 256-bit key
- 10, 12, or 14 rounds (dep. on key length)

Byte-oriented design (some math done in Galois field $GF(2^8)$)

View input as $4 \times 4$ byte array:

```
<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

$4 \times 4 \times 8 = 128$

For version with 128-bit keys, 10 rounds:

- Derive 11 "round keys", each 128 bits ($4 \times 4 \times $ byte)
- In each round:
  1. XOR round key
  2. Substitute bytes (lookup table)
  3. Rotate rows (by different amounts)
  4. Mix each column (by linear opn)
- Output final state

See readings for details.
There are very fast implementations. Also Intel has put supporting hardware into its CPUs.

Security: Good; perhaps 10 rounds should be a bit larger...
For practical purposes, can treat AES as ideal black cipher: 

For each key, mapping $Enc(K, \cdot)$ is a random independent permutation of $\{0,1\}^{128}$ to itself.

**Modes of Operation:**

How to encrypt variable-length messages? (using AES)

- **"ECB"** = "Electronic code book"
- **"CTR"** = "Counter mode"
- **"CBC"** = "Cipher-block chaining" (CBC-MAC)
- **"CFB"** = "Cipher feedback" (others...)

---

**ECB:**

\[
\begin{array}{cccc}
M_1 & M_2 & \cdots & M_n \\
E & E & \cdots & E \\
C_1 & C_2 & \cdots & C_n \\
\end{array}
\]

$\leftarrow$ divide data into $b$-bit blocks, where $b =$ input block size

To handle data that is not a multiple of $b$ bits in length:

- Append a "1" bit (always)
- Append enough "0" bits to make length a multiple of $b$ bits.

This gives invertible (1-1) "paddling" operation. Pad before encryption; unpadd after decryption.

**ECB preserves many patterns:** repeated message blocks $\Rightarrow$ repeated ciphertext blocks

**ECB really only good for encrypting random data** (e.g., keys)
**CTR (Counter mode):**

Generate a PR (pseudorandom) sequence by encrypting $i, i+1, \ldots$

XOR with message to obtain ciphertext.

\[
\begin{array}{cccc}
   i & & i+1 & & i+2 \\
   \downarrow & & \downarrow & & \downarrow \\
   K \rightarrow E & & K \rightarrow E & & K \rightarrow E \\
   \downarrow & & \downarrow & & \downarrow \\
   X_i & & X_{i+1} & & X_{i+2} \\
   \downarrow & & \downarrow & & \downarrow \\
   M_1 & \oplus & M_2 & \oplus & M_3 \\
   \downarrow & & \downarrow & & \downarrow \\
   C_1 & & C_2 & & C_3 \\
\end{array}
\]

Initial counter value can be transmitted first:

\[i', C_1, C_2, \ldots\]

Of course, no counter value should be re-used!
CBC (Cipher-block chaining):

Choose IV ("initialization value") randomly, then use each \( C_i \) is "IV" for \( M_{i-1} \). Transmit IV with ciphertext:

\[ I V, C_1, C_2, \ldots, C_n \]

Decryption easy, and parallelizable (little error propagation).

Lookup "cipher text stealing" for cute way of handling messages that are not a multiple of 64 bits in length. This method gives ciphertext length = message length.

Last block \( C_n \) is the "CBC-MAC" (CBC Message Authentication code) for message \( M \). [A fixed IV is used here.] The MAC is a "cryptographic checksum" (more later...) (If messages have variable length then key for last block should be different.)
CFB (Cipher Feedback mode)

Similar to CBC mode, uses random IV transmitted with ciphertext.

If M is not a multiple of b bits in length, can just transmit shortened ciphertext. (No need for ciphertext stealing.)
Are these modes good ones? What do we want?

If block cipher is indistinguishable from ideal block cipher then mode provides indistinguishability based on chosen ciphertext attack (IND-CCA):

- Define as game with adversary.
- Mode is IND-CCA secure if adversary can win with probability at most $\frac{1}{2} + \varepsilon$ for "negligible" $\varepsilon$.

Let $K$ be randomly chosen key.
Let $E_K$ denote encryption (using mode) with key $K$.
Let $D_K$ denote decryption.

Phase I ("Find"):  
- Adversary given black-box access to $E_K, D_K$ (can encrypt/decrypt whatever it likes).
- Adversary outputs two messages $m_0, m_1$ of same length, plus state information $S$.

Phase II ("Guess"):  
- Examiner secretly picks $d \leftarrow_R \{0, 1\}$
  Examiner computes $y = E_K(m_d)$
- Adversary given $y, S$, access to $E_K$, and access to $D_K$ (except on $y$)
- Adversary computes for a while, then must produce bit $\hat{d}$ as its guess for $d$.
- Adversary's advantage is $|P(\hat{d} = d) - \frac{1}{2}|$

Encryption secure against CCA attack if advantage is negligible.
Fact: To be IND-CCA secure, method must be randomized!
(else Adv can encrypt $m_0, m_1$ & compare to $y$)
A randomization should not be evident to Adv (i.e. not usable for decryption)

Previous modes are not IND-CCA secure!

**ECB**: not randomized

**CTR**: starting counter value might be random, but it is transmitted in clear. In any case, it is legal for Adv to ask for decryption of prefix of $y$ (giving prefix of $md$)

**CBC**: similar to CTR: IV might be random, but it is transmitted in clear. Decryption of prefix of $y$ also works.

**CFB**: Same. IV in clear; prefix argument works.

Can one design a IND-CCA scheme?
Here is a sketch of one IND-CCA secure method, (due to Desai. UFE = "Unbalanced Feistel encryption")

\[ M = \text{long message, sequence } M_1, M_2, \ldots, M_n \text{ of } b \text{-bit blocks.} \]

\[ K = (K_1, K_2, K_3) \quad \text{Three indep. keys for block ciphers} \]

\[ r \xleftarrow{\text{R}} \{0, 1\}^b \quad \text{starting counter value} \]

\[ \text{pad } P = P_1, P_2, \ldots, P_n \text{ where } P_i = E_{K_1}(r + i) \leftarrow (\text{CTR mode}) \]

\[ \text{ciphertext } C = C_1, C_2, \ldots, C_n \text{ where } C_i = M_i \oplus P_i \]

\[ \text{CBC-MAC: } X_0 = 0^b \]

\[ X_i = E_{K_a}(X_{i-1} \oplus C_i) \quad 1 \leq i < n \]

\[ X_n = E_{K_3}(X_{n-1} \oplus C_n) \quad (\text{MAC}) \]

\[ \sigma = r \oplus X_n \quad \text{use MAC to mask } r \]

\[ \text{Output: } C_1, C_2, \ldots, C_n, \sigma \]
• Encryption with UFE can be done in single pass over data, but decryption requires two passes:
  - first to compute mac Xₙ, then to get r
  - second to decrypt C to get M

• Only designed for confidentiality (there is no way provided for receiver to tell if ciphertext has been tampered with.) (Need to use MAC on top of all of this, or some "combined mode" providing both confidentiality & integrity.)

• Note "unbalanced Feistel structure".

• Length of ciphertext \((C, r) = |M| + |r|\); expansion only as needed for randomization.
  No need for "ciphertext stealing" since we use CTR mode.