Admin:

- Pset #1 grading:
- Lecture notes for last week (#6, #7) up soon

Notes:

RSA Conference, discussion of dual-ec-drbg

Project idea:

"Security of routers"

Today:

Block ciphers:

- DES (Data Encryption Standard)
- AES (Advanced Encryption Standard)

Ideal Block Cipher

- "Modes of Operations" - ECB
  - CTR
  - CBC
  - CFB

IND-CCA security def.

UFE mode
Block ciphers:

\[ P \xrightarrow{\text{key } k} Enc \xrightarrow{} C \]

- Plaintext block
- Cipherertext block

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.
DES

"Data Encryption Standard"
Standardized in 1976. Now deprecated in favor of AES.

Feistel structure

16 rounds total

one round

\[
\begin{align*}
L_0 & \\
R_0 & \\
F & \\
k_1 & \text{"round key"}
\end{align*}
\]

plaintext 64 bits

ciphertext

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

Notes:

Invertible for any f and any key schedule.

F uses 8 "S-boxes" mapping 6-bits \( \rightarrow \) 4 bits non-linearly.

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

Subject to differential attacks:

\[
\begin{align*}
M & \xrightarrow{L} M \oplus A \\
\rightarrow & \text{DES} \\
k & \xrightarrow{L} \text{DES} \\
\rightarrow & c \oplus 8
\end{align*}
\]

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

Subject to linear attacks:

e.g. \( M_3 \oplus M_5 \oplus C_3 \oplus K_{14} = 0 \) (e.g. on bits)

with prob \( p = 1/2 + \epsilon \)

then need \( 1/\epsilon^2 \) samples to break (Matsui, \( 2^{43} \) PT/CT pairs)
AES

"Advanced Encryption Standard" (U.S. govt)

Replaces DES

AES "contest" 1997-1999:
15 algorithms submitted: Rc6, 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 x 4 byte arrays:

```
4 x 4 x 8 = 128
```

For version with 128-bit keys, 10 rounds:

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

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

Security: Good, perhaps # rounds should be a bit longer...
For practical purposes, can treat AES as ideal block cipher:

- For each key, mapping \( \text{Enc}(K, \cdot) \) is a random independent permutation of \( \{0,1\}^B \) 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:**

```
\[ M_1 \downarrow \quad \vdots \quad \downarrow M_n \]
\[ \text{E} \quad \text{E} \quad \vdots \quad \text{E} \quad \text{E} \quad \text{E} \downarrow \]
\[ C_1 \quad \vdots \quad \vdots \quad \downarrow C_n \]
```

- To handle data that is not a multiple of \( b \) bits in length:
  1. Append a “1” bit (always)
  2. Append enough “0” bits to make length a multiple of \( b \) bits.

This gives invertible \((a+1)\) “padding” operation. Pad before encryption; unpad after decryption.

ECB preserves many patterns: repeated message blocks \( \Rightarrow \) repeated ciphertext blocks

ECB really only good for encrypting random data (e.g., keys)