The Data Encryption Standard

- see Susan Landau’s paper: “Standing the test of time: the data encryption standard.”

DES

- adopted in 1977 as a standard for “unclassified” applications
- after a public solicitation from NBS (now NIST), IBM developed this cryptosystem
- initially expected to be the standard for 10-15 years; however, it remained a strong cryptosystem until the mid-90’s
The Data Encryption Standard

- a special type of iterated cipher called a Feistel Cipher

- in each round $i$, the state is split into two halves of equal length, called $L_i$ and $R_i$

- for all rounds $i \geq 1$:
  - $L_i = R_{i-1}$
  - $R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$

How to decrypt?

Knowing $L_i, R_i$, how do we find $L_{i-1}, R_{i-1}$?

We know:

$$R_{i-1} = L_i$$  \hspace{1cm} \text{easy}

$$L_{i-1} = R_i \oplus f(R_{i-1}, K_i) = R_i \oplus f(L_i, K_i)$$

Same as encryption, just left/right switched.
The Data Encryption Standard

Recall:

- $L_i = R_{i-1}$
- $R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$

DES:
- a 16-round Feistel cipher with block length 64
- a 56-bit key; each round key has 48 bits
(a different subset of key bits is used in each round – known as the “key schedule”)

See Figure 4.4 in the book or Figure 1 in Landau.

Note: DES begins with a fixed initial permutation and ends with its inverse. These permutations have no cryptographic significance, and are often ignored in the cryptanalysis.
The DES f function

\[ f: \{0, 1\}^{32} \times \{0, 1\}^{48} \rightarrow \{0, 1\}^{32} \]

How to compute \( f(A, J) \) (see Figure 4.5 in the book):

1. Expand \( A \) to a bitstring of length 48, using a fixed expansion function \( E \).

2. Compute \( E(A) \oplus J \). View \( E(A) \oplus J \) as the concatenation of eight 6-bit strings, \( B_1B_2\ldots B_8 \).

3. Apply S-box \( S_i \) to \( B_i \). DES has 8 different S-boxes. Each S-box maps 6 bits to 4 bits. Let \( C_i = S_i(B_i) \). We now have a 32-bit string \( C_1C_2\ldots C_8 \).

4. Permute \( C_1C_2\ldots C_8 \) using a fixed permutation \( P \). Then, \( f(A, J) = P(C_1C_2\ldots C_8) \).
The DES S boxes

- see page 128: eight different S-boxes

- recall:
  - how many bits in? 6
  - how many bits out? 4

- how to understand the S-box “tables”:
  - a 4x16 array
  - the 6-bit string $b_1b_2...b_6$ is mapped to the location given by row $b_1b_6$ and column $b_2b_3b_4b_5$
Modes of Operation

Electronic codebook (ECB):
- split into plaintext into blocks of 64 bits
- possible problems?

  - problems with letters/plaintexts that always start in the same way
    (e.g. e-mail messages or “Dear...”)

Cipher Block Chaining (CBC):
- use the previous block for encryption of the next block
- how?

  \[
  \begin{array}{c}
  \text{64 bit} \\
  \downarrow \\
  P_1 \\
  \leftrightarrow \\
  P_1 \oplus C_1 \\
  \downarrow \\
  C_1 \\
  \end{array}
  \]

Other modes to:
- process blocks of length <64 bits: Cipher Feedback (CFB)
- avoid error propagation: Output Feedback (OFB)
When DES was proposed as standard, there was immediate criticism.

First and foremost, it was felt that 56 bits (keyspace size $2^{56}$) was not enough to be secure.

In 1998, a $250,000 computer built by the Electronic Frontier Foundation (the “DES Cracker”) found a DES key in 56 hours, testing 88 billion keys per second. In 1999, the DES Cracker and 100,000 networked computers found a DES key in 22 hours, testing over 245 billion keys per second.

A second concern was the S-boxes. People were concerned that the S-boxes might contain hidden “trapdoors” that would allow the NSA to decrypt messages. No such trapdoor has ever been found.
Differential Cryptanalysis

Biham and Shamir discovered differential cryptanalysis for DES. Their attack needed “only” $2^{47}$ chosen plaintexts, so this attack is not practical. They also found that almost every variation on DES that they tried was weaker than original DES.

This was no accident! IBM revealed that they knew about differential cryptanalysis when they developed DES, and that they had tried to make DES secure against differential cryptanalysis. They also kept their knowledge of differential cryptanalysis a secret for almost 20 years, until it was rediscovered.
The (secret) criteria for S-box design

1. Each S-box should have 6 bits of input and 4 bits of output. (Largest possible if DES were to fit on a single chip in 1974.)

2. No output bit of an S-box should be too close to a linear function of the input bits. [Any ideas how to improve this criterion?]

3. Each “row” of an S-box should contain all possible outputs.

4. If two inputs to an S-box differ in exactly 1 bit, their outputs must differ by at least 2 bits.

5. If two inputs to an S-box differ exactly in the middle 2 bits, their outputs must differ by at least 2 bits.

6. If two inputs to an S-box differ in their first 2 bits and agree on the last 2, the two outputs must differ.

7. For any nonzero 6-bit difference between inputs, no more than 8 of the 32 pairs of inputs exhibiting this difference may result in the same output difference.
The IBM researchers had not anticipated linear cryptanalysis. In 1994, Matsui used $2^{43}$ plaintext-ciphertext pairs and 50 days to decrypt a DES-encoded message. Again, this is not really practical.

Still, linear and differential cryptanalysis are extremely important. These attacks work against any SPN-like cryptosystem. So, all these cryptosystems must be designed to be “secure” against differential and linear cryptanalysis.
Meet-in-the-Middle Attack

Double DES:
- what happens if we apply DES twice (with different keys)?
- will this mean that the key space is of size \((2^{56})^2\)?

Meet-in-the-Middle Attack (known plaintext):
- Eve intercepted \(m\) and \(E_{k_2} (E_{k_1} (m)) = c\)
- Generate all \(2^{56}\) keys \(k'\) and compute the encryption \(E_{k'}(m)\)
- Generate all \(2^{56}\) keys \(k''\) and compute ...

Note: triple DES seems to be approx. equiv. to a 112-bit key
Weak Keys

There are certain keys one should avoid when using DES. These are the “weak keys”: keys such that every subkey is the same, and the “possibly weak keys”: keys that generate only 4 different subkeys. All these keys are known and thus should be avoided.