Secret Key Cryptography
Block Cipher Scheme

Encrypt

Secret key

Decrypt

Plaintext block of length N

Cipher block of length N
Generic Block Encryption

• Convert a plaintext block into an encrypted block: one-to-one
  – Key length too short?
  – Block Length: Long enough to avoid known-plaintext attack, but not too long (performance)
    • 64 bit typical

• Output should look random
  – No correlation between plaintext and ciphertext
  – Spread bits around
    • A single input bit should have influence on all the bits of the output
    • Be able to change any one of the output with a probability of about 50%
Example of Block Encryption

Loop for n rounds

64-bit input

8 bits 8 bits 8 bits 8 bits 8 bits 8 bits 8 bits 8 bits

S_1 S_2 S_3 S_4 S_5 S_6 S_7 S_8

8 bits 8 bits 8 bits 8 bits 8 bits 8 bits 8 bits 8 bits

Divide input into eight 8-bit pieces

Eight 8-bit substitution functions derived from the key

64-bit intermediate

Permute the bits, possibly based on the key

64-bit output
Generic Block Encryption

• Two kinds of simple transformations (k-bit blocks)
  – A Substitution specifies, for each of the $2^k$ possible values of the input, the k-bit output.
  – A Permutation specifies, for each of the k input bits, the output position to which it goes.
  – Round: a combination of substitution and permutation

• Given the key, an encryption mechanism must be efficient to reverse
DES (Data Encryption Standard)

- Published in 1977, standardized in 1979
- Key: 64 bit quantity = 8-bit parity + 56-bit key
- 64-bit input block, 64-bit output block
- Efficient to implement in hardware but relatively slow if implemented in software
Basic Structure of DES

- 64-bit Input
- Permutation
- Round 1
  - 48-bit $K_1$
- Round 2
  - 48-bit $K_2$
- …
- Round 16
  - 48-bit $K_{16}$
- Swap
- Permutation
- Final Permutation

- Generate 16 48-bit per-round keys
- Initial Permutation
- 56-bit Key
- Swap left and right halves
- 64-bit Output
Permutation of the Data

- **Initial Permutation (IP)**
- View the input as $M: 8$(-byte) x 8(-bit) matrix
- Transform $M$ into $M_1$ in two steps
  - Transpose row $x$ into column $(9-x)$, $0 < x < 9$
  - Apply permutation on the rows:
    - For even column $y$, it becomes row $y/2$;
    - For odd column $y$, it becomes row $(5+y/2)$.
- **Final Permutation $FP = IP^{-1}$**
- IP and FP are not random-looking permutation
- IP and FP do essentially nothing to enhance DES’s security
Permutation of the Data

<table>
<thead>
<tr>
<th>Octet</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>E</td>
<td>F</td>
<td>G</td>
<td>H</td>
</tr>
</tbody>
</table>

input bit

output bit

1 | 2 | 3 | 4 | 5 | 6 | 7 | 8
B | D | F | H | A | C | E | G
Generating the Per-Round Keys

Initial Permutation of the 56 useful bits of the key

One round

48 bits

K_i

C_{i-1} 28 bits \rightarrow Circular Left Shift \rightarrow Permutation with Discard \rightarrow C_i 28 bits

D_{i-1} 28 bits \rightarrow Circular Left Shift \rightarrow Permutation with Discard \rightarrow D_i 28 bits

Round 1,2,9,16: single shift
Others: two bits

Also see Fig. 3-5 of the textbook
Initial Permutation of Key

\[
\begin{array}{cccccc}
1 & 2 & 3 & 4 & 5 & 6 & 7 \\
8 & 9 & 10 & 11 & 12 & 13 & 14 \\
15 & 16 & 17 & 18 & 19 & 20 & 21 \\
22 & 23 & 24 & 25 & 26 & 27 & 28 \\
29 & 30 & 31 & 32 & 33 & 34 & 35 \\
36 & 37 & 38 & 39 & 40 & 41 & 42 \\
43 & 44 & 45 & 46 & 47 & 48 & 49 \\
50 & 51 & 52 & 53 & 54 & 55 & 56 \\
57 & 58 & 59 & 60 & 61 & 62 & 63 \\
\end{array}
\]
Apply the same operations (keys in reverse order: $k_{16}, k_{15}, \ldots, k_1$)
- Input: $R_{n+1} | L_{n+1}$: swap operation
- Output: $R_n | L_n$: The swap operation at the end will produce the correct result
Expansion of R from 32 bits to 48 bits

- Taking the adjacent bits and concatenating them to the chunk
The Mangler Function

The permutation produces “spread” among the chunks/S-boxes!

- Also see Figure 3-8 of the textbook
**S-Box (Substitute and Shrink)**

- 48 bits $\Rightarrow$ 32 bits. $(8 \times 6 \Rightarrow 8 \times 4)$
- 2 bits used to select amongst 4 substitutions for the rest of the 4-bit quantity

$S_i$  

2 bits row

- $I_1$  
- $I_2$  
- $I_3$  
- $I_4$  
- $I_5$  
- $I_6$

4 bits column

- An integer between 0 and 15

$i = 1, \ldots, 8.$

- Also see Figure 3-8 of the textbook
Example: input: 100110 output: ???
8 S-Boxes

- Logic behind the selection of the S-Boxes remains unpublished secret
- Is it a good idea technically to publish it?
- For details, see p. 72 – p. 73 of the textbook
DES Standard

<table>
<thead>
<tr>
<th>Cipher Iterative Action</th>
<th>Key Generation Box</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input: 64 bits</td>
<td>Input: 56 bits</td>
</tr>
<tr>
<td>Key: 48 bits</td>
<td>Output: 48 bits</td>
</tr>
<tr>
<td>Output: 64 bits</td>
<td></td>
</tr>
</tbody>
</table>

One round (Total 16 rounds)

- Design process of DES was not more public
  - If published the design process, this will involve many cryptanalytic attacks
DES Design Controversy

- Although DES standard is public, there was considerable controversy over design
  - in choice of 56-bit key
  - and because design criteria were classified
- Subsequent events and public analysis show in fact design was appropriate
- 16 Weak and Semi-Weak Keys
  - $C_0$ and $D_0$ are: all ones, all zeros, alternating ones and zeros, alternating zeros and ones
- DES has become widely used, especially in financial applications
International Data Encryption Algorithm (IDEA)

- ETH Zurich, 1991
- Encrypt a 64-bit block of plaintext into a 64-bit block of ciphertext
- 128-bit key
- Both DES and IDEA have the property that encryption and decryption are identical except for key expansion
Basic Structure of IDEA

64-bit Input

Round 1

Round 2

......

Round 16

Round 17

64-bit Output

128-bit Key

key expansion

k₁k₂k₃k₄

k₅k₆

k₄₉k₅₀k₅₁k₅₂
IDEA Primitive Operations

• Map two 16-bit quantities into a 16-bit quantity
• Three operations:
  – Bitwise exclusive or: $\oplus$
  – Slightly modified add + mod $2^{16}$
  – Slightly modified multiply $\otimes$ mod($2^{16}$+1)
  – reversible
Key Expansion

- Generation of Key 1 through 8

- Generation of Key 9 through 16
Key Expansion

- The 128-bit key is expanded into 52 16-bit keys K1, K2, … K52
- Once the key are generated, the encryption and decryption operations are the same
- Chop off 16 bits at a time to get 8 16-bit keys
- Start at bit 25, chop (and wrap around) again to get next 8 16-bit keys
- Offsetting 25 more bits, repeat, until 52 keys are generated
One Round

- 17 rounds, even and odd
- 64 bits input are divided into 4 16-bit quantities, $X_a$, $X_b$, $X_c$, and $X_d$
- Odd round
  - Use four of $K_i$
- Even round
  - Use two of $K_i$
IDEA: Odd Round

- $X_a' = X_a \otimes K_a$
- $X_b' = X_c + K_c$
- $X_c' = X_b + K_b$
- $X_d' = X_d \otimes K_d$
- To decrypt, use the inverse (relative to the operations) of the keys
IDEA: Even Round

\[ Y_{out} = ((K_e \otimes Y_{in}) + Z_{in}) \otimes K_f \]

\[ Z_{out} = (K_e \otimes Y_{in}) + Y_{out} \]
IDEA: Even Round

- Mangler: $Y_{\text{out}}, Z_{\text{out}} = f(Y_{\text{in}}, Z_{\text{in}}, K_e, K_f)$

- First step:
  - $Y_{\text{in}} = X_a \oplus X_b$
  - $Z_{\text{in}} = X_c \oplus X_d$

- Second Step, mangler:
  - $Y_{\text{out}} = ((K_e \otimes Y_{\text{in}}) + Z_{\text{in}}) \otimes K_f$
  - $Z_{\text{out}} = (K_e \otimes Y_{\text{in}}) + Y_{\text{out}}$

- Third step:
  - $X_a' = X_a \oplus Y_{\text{out}}, X_b' = X_b \oplus Y_{\text{out}},$
  - $X_c' = X_c \oplus Z_{\text{out}}, X_d' = X_d \oplus Z_{\text{out}}$
IDEA Decryption

• Decryption
  – Use the same keys
  – Use the exact operations as encryption
    • The same code can perform either encryption or decryption given different expanded keys
Advanced Encryption Standard (AES)
Origins

- clearly a replacement for DES was needed
  - have theoretical attacks that can break it
  - DES’s key was too small
  - have demonstrated exhaustive key search attacks
- can use Triple-DES – but slow with small blocks
- US NIST issued call for ciphers in 1997
- 15 candidates accepted in Jun 98
- 5 were short-listed in Aug-99
- Algorithm Rijndael was selected as the AES in Oct-2000
- issued as FIPS PUB 197 standard in Nov-2001
AES Requirements

- private key symmetric block cipher
- 128-bit data, 128/192/256-bit keys
- stronger & faster than Triple-DES
- active life of 20-30 years (+ archival use)
- provide full specification & design details
- both C & Java implementations
- NIST have released all submissions & unclassified analyses