Content of this part

♦ Introduction to DES
♦ Overview of the DES Algorithm
♦ Internal Structure of DES
♦ Decryption
♦ Security of DES
**DES Facts**

- Data Encryption Standard (DES) encrypts blocks of size 64 bit.
- Developed by IBM based on the cipher Lucifer under influence of the National Security Agency (NSA), the design criteria for DES have not been published.
- Standardized 1977 by the National Bureau of Standards today called National Institute of Standards and Technology (NIST).
- Most popular block cipher for most of the last 30 years.
- By far best studied symmetric algorithm.
- Nowadays considered insecure due to the small key length of 56 bit.
- But: 3DES yields very secure cipher, still widely used today.
- Replaced by the Advanced Encryption Standard (AES) in 2000.

**Block Cipher Primitives: Confusion and Diffusion**

- Shannon: There are two primitive operations with which strong encryption algorithms can be built:
  1. **Confusion:** An encryption operation where the relationship between key and ciphertext is obscured. Today, a common element for achieving confusion is substitution, which is found in both AES and DES.
  2. **Diffusion:** An encryption operation where the influence of one plaintext symbol is spread over many ciphertext symbols with the goal of hiding statistical properties of the plaintext. A simple diffusion element is the bit permutation, which is frequently used within DES.

- Both operations by themselves cannot provide security. The idea is to concatenate confusion and diffusion elements to build the so called product ciphers.
**Product Ciphers**

- Most of today’s block ciphers are *product ciphers* as they consist of rounds which are applied repeatedly to the data.
- Can reach excellent diffusion: changing of one bit of plaintext results on average in the change of half the output bits.

*Example:*

\[
\begin{align*}
\chi_1 &= 0010\ 1011 \\
\chi_2 &= 0000\ 1011 \\
&\xrightarrow{\text{Block Cipher}} \\
\nu_1 &= 1011\ 1001 \\
\nu_2 &= 0110\ 1100
\end{align*}
\]

↑ single bit flip → many bit flips

---

**Overview of DES**

- Encrypts blocks of size 64 bits.
- Uses a key of size 56 bits.
- Symmetric cipher: uses same key for encryption and decryption
- Uses 16 rounds performing identical operation
- Different subkey in each round derived from main key
The DES Feistel Network (1)

- DES structure is a Feistel network
- Advantage: encryption and decryption differ only in keyschedule

- Bitwise initial permutation, then 16 rounds
  1. Plaintext is split into 32-bit halves $L_i$ and $R_i$
  2. $R_i$ is fed into $f$, the output of which is then XORed with $L_i$
  3. Left and right half are swapped

- Rounds can be expressed as:
  
  $$L_i = R_{i-1},$$
  
  $$R_i = L_{i-1} \oplus f(R_{i-1}, k_i)$$

The DES Feistel Network (2)

- $L$ and $R$ swapped again at the end of the cipher, i.e., after round 16 followed by a final permutation
- Permutations do not add security, only simplify layout

$$R_{i-1} = L_i, \quad L_{i-1} = R_{i} \oplus f(k_i, L_i)$$
Initial and Final Permutation

- Bitwise Permutations.
- Inverse operations.
- Described by tables \( IP \) and \( IP^{-1} \).

Initial Permutation

| 38 | 50 | 42 | 34 | 26 | 18 | 10 | 2 |
| 60 | 52 | 44 | 36 | 28 | 20 | 12 | 4 |
| 54 | 46 | 38 | 30 | 22 | 14 | 6 |
| 84 | 70 | 68 | 60 | 52 | 44 | 36 | 28 |
| 57 | 49 | 41 | 33 | 25 | 17 | 9 | 1 |
| 59 | 51 | 43 | 35 | 27 | 19 | 11 | 3 |
| 61 | 53 | 45 | 37 | 29 | 21 | 13 | 5 |
| 63 | 55 | 47 | 39 | 31 | 23 | 15 | 7 |

Final Permutation

| 40 | 8 | 48 | 10 | 36 | 24 | 64 | 32 |
| 39 | 7 | 47 | 15 | 55 | 23 | 63 | 31 |
| 38 | 6 | 46 | 14 | 54 | 22 | 62 | 30 |
| 87 | 5 | 45 | 13 | 53 | 21 | 61 | 29 |
| 36 | 4 | 44 | 12 | 52 | 20 | 60 | 28 |
| 55 | 3 | 43 | 11 | 51 | 19 | 59 | 27 |
| 54 | 2 | 42 | 10 | 30 | 18 | 58 | 26 |
| 53 | 1 | 41 | 9 | 40 | 17 | 57 | 35 |

The \( f \) Function

- Main operation of DES
- \( f \)-Function inputs: \( R_{i-1} \) and round key \( k_i \)
- 4 Steps:
  1. Expansion \( E \)
  2. XOR with round key
  3. S-box substitution
  4. Permutation
The Expansion Function $E$

1. Expansion $E$
   - main purpose: increases diffusion

<table>
<thead>
<tr>
<th>$E$</th>
</tr>
</thead>
<tbody>
<tr>
<td>32 1 2 3 4 5</td>
</tr>
<tr>
<td>4     5 6 7 8 9</td>
</tr>
<tr>
<td>8     9 10 11 12 13</td>
</tr>
<tr>
<td>12 13 14 15 16 17</td>
</tr>
<tr>
<td>16 17 18 19 20 21</td>
</tr>
<tr>
<td>20 21 22 23 24 25</td>
</tr>
<tr>
<td>24 25 26 27 28 29</td>
</tr>
<tr>
<td>28 29 30 31 32 1</td>
</tr>
</tbody>
</table>

Add Round Key

2. XOR Round Key
   - Bitwise XOR of the round key and the output of the expansion function $E$
   - Round keys are derived from the main key in the DES key schedule (in a few slides)
The DES S-Boxes

3. S-Box substitution
   - Eight substitution tables.
   - 6 bits of input, 4 bits of output.
   - Non-linear and resistant to differential cryptanalysis.
   - Crucial element for DES security!

The Permutation P

4. Permutation P
   - Bitwise permutation.
   - Introduces diffusion.
   - Output bits of one S-Box effect several S-Boxes in next round.
   - Diffusion by E, S-Boxes and P guarantees that after Round 5 every bit is a function of each key bit and each plaintext bit.
Key Schedule (1)

♦ Derives 16 round keys (or subkeys) $k_i$ of 48 bits each from the original 56 bit key.

♦ The input key size of the DES is 64 bit: 56 bit key and 8 bit parity:

<table>
<thead>
<tr>
<th>MSB</th>
<th>64</th>
<th>LSB</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P</td>
<td></td>
</tr>
<tr>
<td></td>
<td>...</td>
<td></td>
</tr>
<tr>
<td></td>
<td>P</td>
<td></td>
</tr>
</tbody>
</table>

$P = \text{parity bit}$

♦ Parity bits are removed in a first permuted choice $PC-1$:
  (note that the bits 8, 16, 24, 32, 48, 56 and 64 are not used at all)

<table>
<thead>
<tr>
<th>PC-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>57</td>
</tr>
<tr>
<td>49</td>
</tr>
<tr>
<td>13</td>
</tr>
<tr>
<td>32</td>
</tr>
<tr>
<td>22</td>
</tr>
<tr>
<td>10</td>
</tr>
<tr>
<td>8</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>C0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>D0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

Key Schedule (2)

♦ Split key into 28-bit halves $C_0$ and $D_0$.

♦ In rounds $i = 1, 2, 9, 16$, the two halves are each rotated left by one bit.

♦ In all other rounds the two halves are each rotated left by two bits.

♦ In each round $i$ permuted choice $PC-2$ selects a permuted subset of 48 bits of $C_i$ and $D_i$ as round key $k_i$, i.e. each $k_i$ is a permutation of $k$

♦ Note: The total number of rotations:
  $4 \times 1 + 12 \times 2 = 28 \Rightarrow D_0 = D_{16}$ and $C_0 = C_{16}$
Decryption

- In Feistel ciphers only the keyschedule has to be modified for decryption.
- Generate the same 16 round keys in reverse order. (for a detailed discussion see Ch. 3 Understanding Cryptography)
- Reversed key schedule:
  As \( D_0 = D_{16} \) and \( C_0 = C_{16} \) the first round key can be generated by applying \( PC-2 \) right after \( PC-1 \) (no rotation here).
  All other rotations of \( C \) and \( D \) can be reversed to reproduce the other round keys resulting in:
  - No rotation in round 1.
  - One bit rotation to the right in rounds 2, 9 and 16.
  - Two bit rotations to the right in all other rounds.

Properties of DES

- Complementation: \( \text{DES}(k, m) = \text{DES}(k, \overline{m}) \)
- Key must be randomly chosen
- „Weak“ keys:
  encryption=decryption
  \( \text{DES}(k, \text{DES}(k, m)) = m, \forall m \in \mathcal{M} \)
- „Semi-weak“ keys:
  \( \text{DES}(k, \text{DES}(k', m)) = m, \forall m \in \mathcal{M} \)

Group 1

<table>
<thead>
<tr>
<th>( k )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( 0x \ 0101010101010101 )</td>
</tr>
</tbody>
</table>
| \( 0x \ FEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEFEF

Adapted from Paar & Pelzl, “Understanding Cryptography,” and other sources
Security of DES

- After proposal of DES two major criticisms arose:
  1. Key space is too small (2^{56} keys)
  2. S-box design criteria have been kept secret: Are there any hidden analytical attacks (backdoors), only known to the NSA?

- Analytical Attacks: DES is highly resistant to both differential and linear cryptanalysis, which have been published years later than the DES. This may mean that IBM and NSA had been aware of these attacks for 15 years. So far there is no known analytical attack which breaks DES in realistic scenarios.

- Exhaustive key search: For a given pair of plaintext-ciphertext (x, y) test all 2^{56} keys until the condition \( \text{DES}_k^{-1}(x) = y \) is fulfilled.
  \( \Rightarrow \) Relatively easy given today's computer technology

<table>
<thead>
<tr>
<th>Year</th>
<th>Proposed/implemented DES Attack</th>
</tr>
</thead>
<tbody>
<tr>
<td>1993</td>
<td>Wiener proposes design of a very efficient key search machine: Average search requires 36h. Cost: $1M</td>
</tr>
<tr>
<td>1993</td>
<td>Matsui proposes linear cryptanalysis (2^{43} ciphertexts)</td>
</tr>
<tr>
<td>1997</td>
<td>DES Challenge I broken, 4.5 months of distributed search</td>
</tr>
<tr>
<td>1998</td>
<td>DES Challenge II-(1) broken, 39 days (distributed search)</td>
</tr>
<tr>
<td>Jul. 1998</td>
<td>DES Challenge II-(2) broken, key search machine Deep Crack: 1800 ASICs with 24 search engines each, Costs: $250,000 15 days average search time</td>
</tr>
<tr>
<td>Jan. 1999</td>
<td>DES Challenge III broken in 22h 15min (distributed search assisted by Deep Crack)</td>
</tr>
<tr>
<td>2006-2008</td>
<td>Reconfigurable key search machine COPACOBANA (Germany), uses 120 FPGAs to break DES in 6.4 days (avg.) at a cost of $10,000.</td>
</tr>
</tbody>
</table>
**Triple DES – 3DES3**

- Triple encryption using DES is often used in practice to extend the effective key length of DES to either 168 or 112 (if k3=k1).
- Became standard in 1998

\[ c = \text{DES}(k_3, \text{DES}(k_2, \text{DES}(k_1, m))) \]

- Why not use 2DES with 112-bit key?

\[ \text{DES}^{-1}(k_1, c) = \text{DES}(k_2, m) \]

- A „meet-in-the-middle“ attack is possible:
  - For all \(2^{56}\) possible values \(x\) of \(k_2\) calculate \(A_i = \text{DES}(x, m)\)
  - For all \(2^{56}\) possible values \(y\) of \(k_1\) calculate \(B_j = \text{DES}^{-1}(y, c)\)
  - Compare \(A_i\) to \(B_j\) and find \(k_1\) and \(k_2\)

- Large storage space but almost same complexity as DES

---

**Alternate Triple DES**

- Alternative version of 3DES: \(c = \text{DES}(k_1, \text{DES}^{-1}(k_2, \text{DES}(k_3, m)))\)
  - Advantage: choosing \(k_1=k_2=k_3\) performs single DES encryption.
- No practical attack known today for both versions of 3DES3 and even 3DES2.
- Used in many legacy applications, i.e., in banking systems.
- 3DES3 vs 2DES2: 3DES3 with \(k_1 \neq k_2 \neq k_3\) can be subject to „meet-in-the-middle“ attack \(\text{DES}^{-1}(k_1, c) = \text{DES}^{-1}(k_2, \text{DES}(k_3, m))\)
- Calculate and store \(2^{112}\) values of \(\text{DES}^{-1}(k_2, \text{DES}(k_3, m))\)
- 3DES2 is preferred \(\text{DES}(k_1, \text{DES}^{-1}(k_2, \text{DES}(k_3, m)))\)
- Another variation DES-X: \(k_2 \oplus \text{DES}(k, m \oplus k_1)\)
- 56-bit k and 64-bit \(k_2\) and \(k_3\)
Alternatives to DES

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Data Bits</th>
<th>key lengths</th>
<th>remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>AES / Rijndael</td>
<td>128</td>
<td>128/192/256</td>
<td>DES 'replacement', worldwide used standard</td>
</tr>
<tr>
<td>Triple DES</td>
<td>64</td>
<td>112 (effective)</td>
<td>conservative choice</td>
</tr>
<tr>
<td>Mars</td>
<td>128</td>
<td>128/192/256</td>
<td>AES finalist</td>
</tr>
<tr>
<td>RC6</td>
<td>128</td>
<td>128/192/256</td>
<td>AES finalist</td>
</tr>
<tr>
<td>Serpent</td>
<td>128</td>
<td>128/192/256</td>
<td>AES finalist</td>
</tr>
<tr>
<td>Twofish</td>
<td>128</td>
<td>128/192/256</td>
<td>AES finalist</td>
</tr>
<tr>
<td>IDEA</td>
<td>64</td>
<td>128</td>
<td>patented</td>
</tr>
</tbody>
</table>

Lightweight alternative - PRESENT

- An example of a new block cipher for power constrained devices (e.g., RFID tags)
- Data block=64, key=80,128
- 31 rounds using substitution-permutation (SP) network
- A single 4 to 4 S-Box
- Permutation:
  \[ P(i) = \begin{cases} 
  i - 16 & \text{mod } 63, \ i \in \{0, \ldots, 62\} \\
  63, \ i = 63. 
\end{cases} \]
  \(~1000\) gates

- Key schedule (64 MSBs):
  1. Rotate left 61 positions
  2. S-Box to leftmost 4 bits
  3. XOR 5-bit round-counter
Lessons Learned

♦ DES was the dominant symmetric encryption algorithm from the mid-1970s to the mid-1990s. Since 56-bit keys are no longer secure, the Advanced Encryption Standard (AES) was created.

♦ Standard DES with 56-bit key length can be broken relatively easily nowadays through an exhaustive key search.

♦ DES is quite robust against known analytical attacks: In practice it is very difficult to break the cipher with differential or linear cryptanalysis.

♦ By encrypting with DES three times in a row, triple DES (3DES) is created, against which no practical attack is currently known.

♦ The “default” symmetric cipher is nowadays often AES. In addition, the other four AES finalist ciphers all seem very secure and efficient.