Levels of Security

**Definition:** Unconditional Security
A cryptosystem is unconditionally secure if it cannot be broken even with infinite computational resources.

Q: Which actual cryptosystems are unconditionally secure?
Levels of Security

**Definition:** Computational Security
A cryptosystem is “computational secure” if best possible algorithm for breaking requires N operations, where N is very large and known.

Q: Which actual cryptosystems are “computational secure”?

---

**One-time Pad**

**Vernam Cipher**

*Gilbert Vernam, AT&T*

*Major Joseph Mauborgne*

1926

\[ c_i = m_i \oplus k_i \]

| \( m_i \) | 01110110101001010110101 |
| \( k_i \) | 11011111011101011011101 |
| \( c_i \) | 10101011011111111000011 |

All bits of the key must be chosen at random and never reused

---

**One-time Pad**

**Equivalent version**

\[ c_i = m_i + k_i \mod 26 \]

| \( m_i \) | TO BE OR NOT TO BE |
| \( k_i \) | AX TC VI URD WM OF |
| \( c_i \) | TL UG JZ HFW PK PJ |

All letters of the key must be chosen at random and never reused
Perfect Cipher

Claude Shannon
Communication Theory of Secrecy Systems, 1948

\[ \forall \quad m \in M \quad c \in C \quad P(M=m \mid C=c) = P(M = m) \]

The codebreaker can guess a message with the same probability without knowing a ciphertext as with the knowledge of the ciphertext

Is substitution cipher a perfect cipher?

\[ C = XRZ \]

\[ P(M=ADD \mid C=XRZ) = 0 \]

\[ P(M=ADD) \neq 0 \]

Is one-time pad a perfect cipher?

\[ C = XRZ \]

\[ P(M=ADD \mid C=XRZ) \neq 0 \]

\[ P(M=ADD) \neq 0 \]

M might be equal to
CAT, PET, SET, ADD, BBC, AAA, HOT, HIS, HER, BET, WAS, NOW, etc.
Shannon Product Ciphers

- Computationally secure ciphers based on the idea of diffusion and confusion
- Confusion: relationship between plaintext and ciphertext is obscured, e.g. through the use of substitutions
- Diffusion: spreading influence of one plaintext letter to many ciphertext letters, e.g. through the use of permutations

Basic operations of S-P networks

<table>
<thead>
<tr>
<th>Substitution</th>
<th>Permutation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 1 0 1 1 0</td>
<td>0 0 0 0 1 1 0</td>
</tr>
<tr>
<td>0 0 1 0 1 1 0</td>
<td>0 0 0 0 1 1 0</td>
</tr>
<tr>
<td>0 0 1 0 1 1 0</td>
<td>0 0 0 0 1 1 0</td>
</tr>
<tr>
<td>0 0 1 0 1 1 0</td>
<td>0 0 0 0 1 1 0</td>
</tr>
</tbody>
</table>

S-box  P-box
**Avalanche effect**

- Diagram showing the avalanche effect in cryptography, illustrating how changes in the plaintext lead to significant changes in the ciphertext.

**LUCIFER** by Horst Feistel and Walt Tuchman at IBM

- Diagram showing the structure of the LUCIFER algorithm, including S-boxes, permutations, and rounds.

**LUCIFER-external look**

- Diagram presenting a simplified view of the LUCIFER algorithm, highlighting the plaintext and ciphertext blocks and the key.
NBS public request for a standard cryptographic algorithm
May 15, 1973, August 27, 1974

The algorithm must be:
• secure
• public
  - completely specified
  - easy to understand
  - available to all users
• economic and efficient in hardware
• able to be validated
• exportable

DES - chronicle of events
1973 - NBS issues a public request for proposals for a standard cryptographic algorithm
1975 - first publication of the IBM's algorithm and request for comments
1976 - NBS organizes two workshops to evaluate the algorithm
1977 - official publication as FIPS PUB 46: Data Encryption Standard
1983, 1987, 1993 - recertification of the algorithm for another five years
1993 - software implementations allowed to be validated

Controversies surrounding DES

<table>
<thead>
<tr>
<th>Unknown design criteria</th>
<th>Slow in software</th>
<th>Too short key</th>
</tr>
</thead>
<tbody>
<tr>
<td>Most criteria reconstructed from cipher analysis</td>
<td>Only hardware implementations certified</td>
<td>Theoretical designs of DES breaking machines</td>
</tr>
</tbody>
</table>
**Life of DES**

- **1970**: DES developed by IBM and NSA
- **1980**: In common use for over 20 years
- **1990**: Federal and banking standard
- **2000**: Over 300 validated implementations
- **2000**: De facto world-wide standard
- **2010**: Transition to a new standard

**Most popular secret-key ciphers**

<table>
<thead>
<tr>
<th>Year</th>
<th>DES</th>
<th>Triple DES</th>
<th>AES - Rijndael</th>
</tr>
</thead>
<tbody>
<tr>
<td>1977</td>
<td>56 bit key</td>
<td>AES contest</td>
<td>112, 168 bit keys</td>
</tr>
<tr>
<td>1999</td>
<td>IDEA</td>
<td>Serpent</td>
<td>128, 192, and 256 bit keys</td>
</tr>
<tr>
<td>2000</td>
<td>RC5</td>
<td>Twofish</td>
<td></td>
</tr>
<tr>
<td>2010</td>
<td>CAST</td>
<td>Mars</td>
<td></td>
</tr>
</tbody>
</table>

**DES - external look**

- Plaintext block: 64 bits
- DES: 56 bits
- Ciphertext block: 64 bits
- Key: 56 bits
DES — high-level internal structure

Feistel Structure

Encryption

Decryption
Classical Feistel Network

plaintext = L₀R₀
for i=1 to n
{ 
    Lᵢ = Rᵢ₋₁ 
    Rᵢ = Lᵢ₋₁ ⊕ f(Rᵢ₋₁, Kᵢ) 
} 
Lₙ₊₁ = Rₙ 
Rₙ₊₁ = Lₙ 
ciphertext = Lₙ₊₁Rₙ₊₁

Mangler Function of DES, F

Figure 5.8 Calculation of F(R, Kᵢ)
Notation for Permutations

Input
\[ i_1, i_2, i_3, i_4, i_5, i_6, i_7, i_8, i_9, \ldots, i_{56}, i_{57}, i_{58}, i_{59}, i_{60}, i_{61}, i_{62}, i_{63}, i_{64} \]

\[ 58, 50, 42, 34, 26, 18, 10, 2, \ldots, 5, 63, 55, 47, 39, 31, 23, 15, 7 \]

Output
Notation for S-boxes

Input

\[ i_1 \quad i_2 \quad i_3 \quad i_4 \quad i_5 \quad i_6 \]

- \( i_1 \) determines a row number in the S-box table, 0..3
- \( i_2 \) \( i_3 \) \( i_4 \) \( i_5 \) determines a column in the S-box table, 0..15
- \( o_1 \) \( o_2 \) \( o_3 \) \( o_4 \) is a binary representation of a number from 0..15 in the given row and the given column

Output

Figure 3.6 Single Round of DES Algorithm
General design criteria of DES

1. Randomness
   Changing a single bit at the input changes on average half of the bits at the output.

2. Avalanche property
   Every output bit is a complex function of all input bits (and not just a subset of input bits).

3. Completeness property
   Every output bit is a complex function of all input bits (and not just a subset of input bits).

4. Nonlinearity
   Encryption function is non-affine for any value of the key.

5. Correlation immunity
   Output bits are statistically independent of any subset of input bits.

Completeness property

Every output bit is a complex function of all input bits (and not just a subset of input bits).

Formal requirement:

For all values of \( i \) and \( j \), \( i=1..64, j=1..64 \)
there exist inputs \( X_1 \) and \( X_2 \), such that

\[
\begin{align*}
X_1 & = x_1 \ x_2 \ x_3 \ \ldots \ x_{i-1} \ 0 \ x_{i+1} \ \ldots \ x_{63} \ x_{64} \\
X_2 & = x_1 \ x_2 \ x_3 \ \ldots \ x_{i-1} \ 1 \ x_{i+1} \ \ldots \ x_{63} \ x_{64} \\
Y_1 & = DES(X_1) \\
Y_2 & = DES(X_2)
\end{align*}
\]

Linear Transformations

Transformations that fulfill the condition:

\[
T(X_{[m \times 1]}) = Y_{[n \times 1]} = A_{[n \times m]} \cdot X_{[m \times 1]}
\]

or

\[
T(X_1 \oplus X_2) = T(X_1) \oplus T(X_2)
\]

Affine Transformations

Transformations that fulfill the condition:

\[
T(X_{[m \times 1]}) = Y_{[n \times 1]} = A_{[n \times m]} \cdot X_{[m \times 1]} \oplus B_{[n \times 1]}
\]
Linear Transformations of DES

IP, IP⁻¹, E, PC1, PC2, SHIFT

e.g.,

IP(X₁ ⊕ X₂) = IP(X₁) ⊕ IP(X₂)

Non-Linear and non-affine transformations of DES

S

There are no such matrices A[4x6] and B[4x1] that


Design of S-boxes

• 16! ≈ 2 · 10¹⁵ possibilities
• precisely defined initially unpublished criteria
• resistant against differential cryptanalysis
  (attack known to the designers and rediscovered
  in the open research in 1990 by E. Biham and A. Shamir)

Typical Flow Diagram of a Secret-Key Block Cipher

Round Key[0] → Initial transformation → Round Key[1] → Cipher Round → ... 

i<#rounds+1

#rounds times:

Round Key[0] → Initial transformation

Round Key[i]

Cipher Round

Round Key[#rounds+1] → Final transformation
Implementation of a secret-key cipher in hardware
Round keys computed on-the-fly

Implementation of a secret-key cipher
Round keys precomputed

Basic iterative architecture of secret key ciphers
## Theoretical design of the specialized machine to break DES

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Method:</td>
<td><strong>exhaustive key search attack</strong></td>
</tr>
<tr>
<td>Basic component:</td>
<td>specialized integrated circuit in CMOS technology, 75 MHz</td>
</tr>
<tr>
<td>Checks:</td>
<td>200 mln keys per second</td>
</tr>
<tr>
<td>Costs:</td>
<td>$10</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Total cost</th>
<th>Estimated time</th>
</tr>
</thead>
<tbody>
<tr>
<td>$1 mln</td>
<td>35 minutes</td>
</tr>
<tr>
<td>$100,000</td>
<td>6 hours</td>
</tr>
</tbody>
</table>

## DES breaking machine

![Diagram of DES breaking machine](image)

- **known ciphertext**
- **plaintext**
- **comparator**
- **Round key 1**
- **Round key 2**
- **Round key 16**
- **Key Scheduling Round 1**
- **Key Scheduling Round 2**
- **Key Scheduling Round 16**

## Deep Crack

*Electronic Frontier Foundation, 1998*

- **Total cost:** $220,000
- **Average time of search:** 4.5 days/key

- **1800 ASIC chips, 40 MHz clock**
Deep Crack

**Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of ASIC chips</td>
<td>1800</td>
</tr>
<tr>
<td>Clock frequency</td>
<td>40 MHz</td>
</tr>
<tr>
<td>Number of clock cycles per key</td>
<td>16</td>
</tr>
<tr>
<td>Number of search units per ASIC</td>
<td>24</td>
</tr>
<tr>
<td>Search speed</td>
<td>90 bln keys/s</td>
</tr>
<tr>
<td>Average time to recover the key</td>
<td>4.5 days</td>
</tr>
</tbody>
</table>

COPACOBANA

Cost-Optimized Parallel COde Breaker

*Ruhr University, Bochum, University of Kiel, Germany, 2006*

Cost: € 8980 (ver. 1)

- Based on Xilinx FPGAs (Field Programmable Gate Arrays)
  - ver. 1 – based on **120 Spartan 3 FPGAs**
  - ver. 2 – based on **128 Virtex 4 SX 35 FPGAs**
- Description, FAQ, and news available at [http://www.copacobana.org/](http://www.copacobana.org/)
- For ver. 1 based on Spartan FPGAs
  - Clock frequency = 136 MHz
  - Average search time for a single DES key = **6.4 days**
  - Worst case search time for a single DES key = **12.8 days**
Minimum length of the key for symmetric ciphers

I. Panel of experts, January 1996
   M. Blaze, W. Diffie, R. Rivest, B. Schneier, T. Shimomura, E. Thompson, M. Wiener

Report:
   “Minimal Key Lengths for Symmetric Ciphers to Provide Adequate Commercial Security”

II. National Academy of Sciences, National Research Council, May 1996

Report:
   “Cryptography’s Role in Securing the Information Society”
Secure key length today and in 20 years (against an intelligence agency with the budget of $300M)

<table>
<thead>
<tr>
<th>key length</th>
<th>Secure key length in 2008</th>
<th>Secure key length in 2028</th>
</tr>
</thead>
<tbody>
<tr>
<td>80 bits</td>
<td>84 bits</td>
<td>128 bits IDEA, minimum key length in AES</td>
</tr>
<tr>
<td>56 bits</td>
<td></td>
<td>120 bits DESX</td>
</tr>
<tr>
<td></td>
<td></td>
<td>112 bits Triple DES with two keys</td>
</tr>
</tbody>
</table>

Secure key length - discussion

• increasing key length in a newly developed cipher costs NOTHING
• increasing effective key length, assuming the use of an existing cipher has a limited influence on the efficiency of implementation (DESX, Triple DES)

It is economical to use THE SAME secure key length FOR ALL applications

The primary barriers blocking the use of symmetric ciphers with a secure key length have been of the political nature (e.g., export policy of USA)

Other attacks

• differential cryptanalysis
  Biham, Shamir 1991

• linear cryptanalysis
  Matsui, 1993
Differential cryptanalysis

\[
\begin{array}{cccc}
M_1 & M_1^* & M_2 & M_2^* \\
\ldots & & M_{N-1} & M_{N-1}^* \\
\end{array}
\]

\[M_i \oplus M_i^* = \text{const}\]

- access to the encryption module with the key inside
- analysis of trillions of pairs plaintext-ciphertext

Differential cryptanalysis of DES

\textit{Biham, Shamir 1991}

Requirements:

- access to the encryption module with the key inside
- time for encryption of \(2^{47} = 1.4 \cdot 10^{14}\) plaintext blocks
  \[= 1 \text{ million gigabytes of plaintext}\]

Conclusions:

- attack impossible to mount
- DES specifically designed (IBM, NSA) to be resistant against differential cryptanalysis

Linear cryptanalysis of DES

\textit{Matsui 1993}

Requirements:

- \(2^{43} = 8.8 \cdot 10^{12}\) known plaintext blocks
  \[= 70.3 \text{ terabytes of known plaintext}\]
- \(2^{43}\) operations
- probability of success 85%

Conclusions:

- attack impossible to mount in practice
What if creators of DES did not know about differential cryptanalysis...

Required number of plaintext blocks

<table>
<thead>
<tr>
<th>Modification</th>
<th>Required number of plaintext blocks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original DES</td>
<td>$2^{47} = 1$ mln GB</td>
</tr>
<tr>
<td>Identity permutation in place of P</td>
<td>$2^{19} = 4$ MB</td>
</tr>
<tr>
<td>Order of S-boxes</td>
<td>$2^{38} = 2,000$ GB</td>
</tr>
<tr>
<td>XOR replaced by addition</td>
<td>$2^{31} = 2$ GB</td>
</tr>
<tr>
<td>S-boxes random</td>
<td>$2^{31} = 16$ MB</td>
</tr>
<tr>
<td>one position changed</td>
<td>$2^{30} = 8$ GB</td>
</tr>
<tr>
<td>Expansion function E eliminated</td>
<td>$2^{29} = 64$ MB</td>
</tr>
</tbody>
</table>

Differential and linear cryptanalysis - discussion

- Attacks infeasible for correctly designed ciphers
- Perfect tool for comparing strengths of various ciphers
- Resistance against these attacks does not imply resistance against other unknown methods of attack

Triple DES EDE mode with two keys

Diffie, Hellman, 1977
**Triple DES EDE mode with three keys**

**Advantages:**
- secure key length (112 or 168 bits)
- increased compared to DES resistance to linear and differential cryptanalysis
- possibility of utilizing existing implementations of DES

**Disadvantages:**
- relatively slow, especially in software

---

**Triple DES**

**Advantages:**
- secure key length (112 or 168 bits)
- increased compared to DES resistance to linear and differential cryptanalysis
- possibility of utilizing existing implementations of DES

**Disadvantages:**
- relatively slow, especially in software

---

**DESX**

**Advantages:**
- secure key length (112 or 168 bits)
- increased compared to DES resistance to linear and differential cryptanalysis
- possibility of utilizing existing implementations of DES

**Disadvantages:**
- relatively slow, especially in software