Towards modern ciphers

Data Encryption Standard and its extensions

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

\[ c_i = m_i \oplus k_i \]

\[
\begin{array}{c}
m_i \\
k_i \\
c_i \\
\end{array}
\begin{array}{c}
011101101010101101101101 \\
110011111011011101011011 \\
1010101101111111000011 \\
\end{array}
\]

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

**One-time Pad**
**Equivalent version**

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

\[
\begin{array}{c}
m_i \\
k_i \\
c_i \\
\end{array}
\begin{array}{c}
TO BE OR NOT TO BE \\
AX TC VI URD 890 OF \\
TL UG JZ HFW PK PJ \\
\end{array}
\]

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

Claude Shannon

Communication Theory of Secrecy Systems, 1948

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

\[ m \in M \quad c \in C \]

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 1 1 1</td>
<td>0 1 0 0</td>
</tr>
<tr>
<td>0 0 0 0</td>
<td>0 1 0 0</td>
</tr>
<tr>
<td>1 1 1 1</td>
<td>1 0 1 0</td>
</tr>
<tr>
<td>1 1 1 1</td>
<td>1 0 1 0</td>
</tr>
<tr>
<td>0 0 0 0</td>
<td>0 1 0 0</td>
</tr>
</tbody>
</table>

S-box | P-box
Avalanche effect

LUCIFER

Horst Feistel, Walt Tuchman

IBM

16 rounds

LUCIFER- external look

plaintext block

128 bits

key

512 bits

ciphertext block

128 bits
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

- DES developed by IBM and NSA
- In common use for over 20 years
- Federal and banking standard
- Over 300 validated implementations
- De facto world-wide standard

Most popular secret-key ciphers

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>56 bit key</td>
<td>112, 168 bit</td>
<td>AES - Rijndael</td>
<td>AES contest</td>
<td>IDEA</td>
<td>Blowfish</td>
<td></td>
</tr>
<tr>
<td>American standards</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Other popular algorithms</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Twofish</td>
<td>Serpent</td>
<td>Mars</td>
<td>RC6</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

DES - external look

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

\[ L_{n+1} = R_n \]
\[ R_{n+1} = L_n \oplus f(R_n, K_{n+1}) \]

DES Main Loop
Feistel Structure

\[ L_{n+1} = R_n \]
\[ R_{n+1} = L_n \oplus f(R_n, K_{n+1}) \]

Feistel Structure

Encryption

Decryption
Decryption

Classical Feistel Network

\[
\text{plaintext} = L_0 R_0 \\
\text{for } i=1 \text{ to } n \\
\{ \\
L_i = R_{i-1} \\
R_i = L_{i-1} \oplus f(R_{i-1}, K_i) \\
\} \\
L_{n+1} = R_n \\
R_{n+1} = L_n \\
ciphertext = L_{n+1} R_{n+1}
\]

Mangler Function of DES, F

Figure 3.3: Calculation of F(R, K)
Notation for Permutations

Input
\[ i_1 \quad i_2 \quad i_3 \quad i_4 \quad i_5 \quad i_6 \quad i_7 \quad i_8 \quad i_9 \quad i_{10} \quad \ldots \quad i_{56} \quad i_{57} \quad i_{58} \quad i_{59} \quad i_{60} \quad i_{61} \quad i_{62} \quad i_{63} \quad i_{64} \]
\[ 58 \quad 50 \quad 42 \quad 34 \quad 26 \quad 18 \quad 10 \quad 2 \quad \ldots \quad 5 \quad 63 \quad 55 \quad 47 \quad 39 \quad 31 \quad 23 \quad 15 \quad 7 \]
\[ i_{58} \quad i_{50} \quad i_{42} \quad i_{34} \quad i_{26} \quad i_{18} \quad i_{10} \quad i_{2} \quad \ldots \quad i_{5} \quad i_{63} \quad i_{55} \quad i_{47} \quad i_{39} \quad i_{31} \quad i_{23} \quad i_{15} \quad i_{7} \]

Output
Notation for S-boxes

Input

\[ \begin{array}{c|c|c|c|c|c} i_1 & i_2 & i_3 & i_4 & i_5 & i_6 \\
\end{array} \]

- \( i_1 i_6 \) determines a row number in the S-box table, 0..3
- \( i_2 i_3 i_4 i_5 \) determine 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
General design criteria of DES

1. Randomness

2. Avalanche property
   changing a single bit at the input changes on average half of the bits
   at the output

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

\[
\begin{align*}
X_1 &= x_1 x_2 x_3 \cdots x_{i-1} 0 x_{i+1} \cdots x_{63} x_{64} \\
X_2 &= x_1 x_2 x_3 \cdots x_{i-1} 1 x_{i+1} \cdots x_{63} x_{64}
\end{align*}
\]

\[
\begin{align*}
Y_1 &= \text{DES}(X_1) == y_1 y_2 y_3 \cdots y_{j-1} y_j y_{j+1} \cdots y_{63} y_{64} \\
Y_2 &= \text{DES}(X_2) == y'_1 y'_2 y'_3 \cdots y'_{j-1} y'_j y'_{j+1} \cdots y'_{63} y'_{64}
\end{align*}
\]

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
S(X_{6x1}) = A_{4x6} ⋅ X_{6x1} ⊕ B_{4x1}

Design of S-boxes

• 16! ≈ 2 · 10¹³ 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
Implementation of a secret-key cipher in hardware
Round keys computed on-the-fly

- Input
- Key
- Encryption/decryption
- Round keys
- Output

Implementation of a secret-key cipher
Round keys precomputed

- Input
- Key
- Encryption/decryption
- Key scheduling
- Memory of round keys
- Output

Basic iterative architecture of secret key ciphers

- Key scheduling
- Round keys
- Output
- Input
Theoretical design of the specialized machine to break DES


Method: exhaustive key search attack

Basic component: specialized integrated circuit in CMOS technology, 75 MHz

Checks: 200 mln keys per second

Costs: $10

<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

known ciphertext → Encryption Round 1 → Key Scheduling Round 1 → Encryption Round 2 → Key Scheduling Round 2 → Encryption Round 16 → Key Scheduling Round 16 → known plaintext

Key counter

Round key 1

Round key 2

Round key 16

comparator

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  
- 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**
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 2009</th>
<th>Secure key length in 2030</th>
</tr>
</thead>
<tbody>
<tr>
<td>56 bits</td>
<td>DES</td>
<td>80 bits Skipjack</td>
</tr>
<tr>
<td>80 bits</td>
<td>DESX</td>
<td>120 bits DESX</td>
</tr>
<tr>
<td>112 bits</td>
<td>Triple DES with three different keys</td>
<td>128 bits IDEA, minimum key length in AES</td>
</tr>
<tr>
<td>120 bits</td>
<td>DESX</td>
<td>128 bits IDEA, minimum key length in AES</td>
</tr>
<tr>
<td>128 bits</td>
<td>IDEA, minimum key length in AES</td>
<td>99 bits Secure key length in 2030</td>
</tr>
<tr>
<td>99 bits</td>
<td>Secure key length in 2030</td>
<td>128 bits IDEA, minimum key length in AES</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

\[
\begin{align*}
M_1 & \oplus M_1^* = const \\
M_2 & \oplus M_2^* \\
\vdots & \\
M_{N,1} & \oplus M_{N,1}^* \\
M_N & \oplus M_N^*
\end{align*}
\]

\[
\begin{align*}
C_1 & \oplus C_1^* \\
C_2 & \oplus C_2^* \\
\vdots & \\
C_{N,1} & \oplus C_{N,1}^* \\
C_N & \oplus C_N^*
\end{align*}
\]

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

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
  $= 1$ million gigabytes of plaintext

Conclusions:
• attack impossible to mount
• DES specially designed (IBM, NSA) to be resistant against differential cryptanalysis

Linear cryptanalysis of DES

Matsui 1993

Requirements:
• $2^{43} = 8.8 \cdot 10^{12}$ known plaintext blocks
  $= 70.3$ 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

Original DES
$2^{47} = 1$ mln GB

Modifications:
- Identity permutation in place of P
  $2^{19} = 4$ MB
- Order of S-boxes
  $2^{38} = 2,000$ GB
- XOR replaced by addition
  $2^{31} = 2$ GB
- S-boxes random
  $2^{21} = 16$ MB
  one position changed
  $2^{23} = 8$ GB
- Expansion function E eliminated
  $2^{26} = 64$ MB
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**

**Encryption**

plaintext → E encryption → D decryption → E encryption → ciphertext

<table>
<thead>
<tr>
<th>plaintext</th>
<th>E encryption</th>
<th>D decryption</th>
<th>E encryption</th>
<th>ciphertext</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>E K1</td>
<td>D K2</td>
<td>E K1</td>
<td></td>
</tr>
</tbody>
</table>

**Decryption**

ciphertext → D decryption → E encryption → D decryption → plaintext

<table>
<thead>
<tr>
<th>ciphertext</th>
<th>D decryption</th>
<th>E encryption</th>
<th>D decryption</th>
<th>plaintext</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>D K1</td>
<td>E K2</td>
<td>D K1</td>
<td></td>
</tr>
</tbody>
</table>

**Triple DES EDE mode with three keys**

**Encryption**

plaintext → E encryption → D decryption → E encryption → ciphertext

<table>
<thead>
<tr>
<th>plaintext</th>
<th>E encryption</th>
<th>D decryption</th>
<th>E encryption</th>
<th>ciphertext</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>E K1</td>
<td>D K2</td>
<td>E K3</td>
<td></td>
</tr>
</tbody>
</table>

**Decryption**

ciphertext → D decryption → E encryption → D decryption → plaintext

<table>
<thead>
<tr>
<th>ciphertext</th>
<th>D decryption</th>
<th>E encryption</th>
<th>D decryption</th>
<th>plaintext</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>D K1</td>
<td>E K2</td>
<td>D K3</td>
<td></td>
</tr>
</tbody>
</table>
### Best Attacks Against Triple DES

- Version with three keys (168 bits of key)
  - Meet-in-the-middle attack
    - \(2^{52}\) known plaintexts
    - \(2^{113}\) steps
    - \(2^{90}\) single DES encryptions, and
    - \(2^{88}\) memory
  - **Effective key size = \(2^{112}\)**
- Version with two keys (112 bits of key)
  - **Effective key size = \(2^{80}\)**

### 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

**DESX**  
*Rivest, 1988*

- plaintext
- \(64\) bits
- \(K_a\)
- \(64\) bits
- \(K_b\)
- ciphertext
- \(64\) bits
- \(K\)
- \(56\) bits

**KEY =**  
\(K, K_a\)

**120 bits**

\(K_b = \text{hash function}(K, K_a)\)