Symmetric Key Ciphers

Debdeep Mukhopadhyay

Assistant Professor
Department of Computer Science and Engineering
Indian Institute of Technology Kharagpur
INDIA -721302

Objectives

- Definition of Symmetric Types of Symmetric Key ciphers
  - Modern Block Ciphers
- Full Size and Partial Size Key Ciphers
- Components of a Modern Block Cipher
  - PBox (Permutation Box)
  - SBox (Substitution Box)
  - Swap
  - Properties of the Exclusive OR operation
- Diffusion and Confusion
- Types of Block Ciphers: Feistel and non-Feistel ciphers
Symmetric Key Setting

Assumptions

- Eve
  - Eve is the encryption key, $K_b$ is the decryption key.
  - For symmetric key ciphers, $K_a = K_b$
  - Only Alice and Bob knows $K_a$ (or $K_b$)
  - Eve has access to $E$, $D$ and the Communication Channel but does not know the key $K_a$ (or $K_b$)

Types of symmetric key ciphers

- Block Ciphers: Symmetric key ciphers, where a block of data is encrypted
- Stream Ciphers: Symmetric key ciphers, where block size=1
Block Ciphers

Block Cipher

- A symmetric key modern cipher encrypts an n bit block of plaintext or decrypts an n bit block of ciphertext.
- Padding:
  - If the message has fewer than n bits, padding must be done to make it n bits.
  - If the message size is not a multiple of n, then it should be divided into n bit blocks and the last block should be padded.
Full Size Key Ciphers

• Transposition Ciphers:
  – Involves rearrangement of bits, without changing value.
  – Consider an n bit cipher
  – How many such rearrangements are possible?
    • \( n! \)
  – How many key bits are necessary?
    • \( \lceil \log_2 (n!) \rceil \)

Full Size Key Ciphers

• Substitution Ciphers:
  – It does not transpose bits, but substitutes values
  – Can we model this as a permutation?
    – Yes. The \( n \) bit inputs and outputs can be represented as \( 2^n \) bit sequences, with one 1 and the rest 0’s. This can be thus modeled as a transposition.
  – Thus it is a permutation of \( 2^n \) values, thus needs \( \lceil \log_2(2^n!) \rceil \) bits.
Examples

- Consider a 3-bit block ciphers. How many bits are needed for the full-size key?
  - Transposition cipher: $\text{ceil}(\log_2 6) = 3$ bits.
  - Substitution cipher:
    - There are $8! = 40,320$ possible substitutions
    - Thus there are $\text{ceil}(\log_2 (40,320)) = 16$ bits
  - Lots of unused key.

Permutation Group

- The fact that the full-size key transposition or substitution cipher is a permutation shows cascading is not of use.
- This is because permutation forms a group under the composition operation.
- Multiple applications of the ciphers has the same effect as a single application of the transformation.
Partial-Size Key Ciphers

- Actual ciphers cannot use full size keys, as the size is large.
- Block ciphers are substitution ciphers (and not transpositions). Why?
- Consider DES, with 64 bit block cipher.
  - Size of full key = ceil(log₂(2^64!)) ≈ 2^70
  - Much large compared to 56 bits which is actually used.

Is the partial-key cipher a group?

- Important, because if yes then again multiple applications of the cipher is useless.
- A partial-key cipher is a group if it is a subgroup of the corresponding full key cipher.
- It has been proved that the multi-stage DES with a 56 bit key is not a group because no subgroup with 2^{56} mappings can be created from the corresponding group with 2^{64}! mappings
Components of a Modern Block Cipher

• Most important components:
  – PBox: It is a key-less fixed transposition cipher
  – SBox: It is a key-less fixed substitution cipher
• They are used to provide:
  – **Diffusion**: it hides the relationship between the ciphertext and the plaintext
  – **Confusion**: it hides the relationship between the ciphertext and the key

Principle of Confusion and Diffusion

• The design principles of Block Cipher depends on these properties
• The S-Box is used to provide **confusion**, as it is dependent on the unknown key
• The P-Box is fixed, and there is no confusion due to it
• But it provides **diffusion**
• Properly combining these is necessary.
Diffusion (P) Boxes

- **Straight Boxes**

<table>
<thead>
<tr>
<th>Example 24x24 Box</th>
</tr>
</thead>
<tbody>
<tr>
<td>01 15 02 13 06 17 03 19 09 04 21 11</td>
</tr>
<tr>
<td>14 05 12 16 18 07 24 10 23 08 22 20</td>
</tr>
</tbody>
</table>

- **Expansion Boxes**

<table>
<thead>
<tr>
<th>Example 12x24 Box</th>
</tr>
</thead>
<tbody>
<tr>
<td>01 03 02 01 06 17 03 07 09 04 09 11</td>
</tr>
<tr>
<td>02 05 12 04 06 07 12 10 11 08 10 08</td>
</tr>
</tbody>
</table>

- **Compression Boxes**

<table>
<thead>
<tr>
<th>Example 24x12 Box</th>
</tr>
</thead>
<tbody>
<tr>
<td>01 15 02 13 06 17 03 19 09 04 21 11</td>
</tr>
</tbody>
</table>

SBox

An SBox (substitution box) is an mxn substitution box, where m and n are not necessarily same.

Each output bit is a Boolean function of the inputs.

\[
\begin{align*}
y_1 &= f_1(x_1, x_2, \ldots, x_n) \\
y_2 &= f_2(x_1, x_2, \ldots, x_n) \\
\vdots \\
y_m &= f_m(x_1, x_2, \ldots, x_n)
\end{align*}
\]
Non-linear SBox

\[ y_1 = a_{11}x_1 \oplus a_{12}x_2 \oplus \ldots \oplus a_{1n}x_n \]
\[ y_2 = a_{21}x_1 \oplus a_{22}x_2 \oplus \ldots \oplus a_{2n}x_n \]
\[ \vdots \]
\[ y_m = a_{m1}x_1 \oplus a_{m2}x_2 \oplus \ldots \oplus a_{mn}x_n \]

In a non-linear S-Box, each of the elements cannot be expressed as above.

Eg.

\[ y_1 = x_1x_3 \oplus x_2, \quad y_2 = x_1x_2 \oplus x_3 \]

Other Components

- Circular Shift:
  - It shifts each bit in an n-bit word k positions to the left.
  - The leftmost k bits become the rightmost bits.
  - Invertible Transformation

- Swap:
  - A special type of shift operation where k=n/2

- Other operations involve split and combine.
- An important component is exclusive-or operation
Properties of Exor

Ex-or is a binary operator, which results in 1 when both the inputs have a different logic. Otherwise, it computes 0.

Symbol: ⊕

Closure: Result of exoring two n bit numbers is also n bits.

Associativity: Allows to use more than one ‘⊕’s in any order:

\[ x \oplus (y \oplus z) = (x \oplus y) \oplus z \]

Commutativity: \[ x \oplus y = y \oplus x \]

Identity: The identity element is the n bit 0, represented by \( (00...0) = 0^n \)

Thus, \[ x \oplus 0^n = x \]

Inverse: Each word is the additive inverse of itself.

Thus, \[ x \oplus x = 0^n \]

Application of Ex-or

- The key is known to both the encryptor and decryptor and helps to recover the plaintext.
A product cipher made of 2 rounds

Diffusion and Confusion
Practical Ciphers

- Large data blocks
- More S-Boxes
- More rounds
- These help to improve the diffusion and confusion in the cipher.

Two classes of product ciphers

- Feistel Ciphers, example DES (Data Encryption Standard)
- Non-Feistel Ciphers (Substitution Permutation Networks), example AES (Advanced Encryption System)
Feistel Cipher

- **Feistel cipher** refers to a type of block cipher design, not a specific cipher
- Split plaintext block into left and right halves:
  \[ \text{Plaintext} = (L_0,R_0) \]
- For each round \( i=1,2,\ldots,n \), compute
  \[
  L_i = R_{i-1} \\
  R_i = L_{i-1} \oplus f(R_{i-1}, K_i)
  \]
  where \( f \) is **round function** and \( K_i \) is **subkey**
- Ciphertext = \((L_n,R_n)\)

Feistel Permutation

- Decryption: Ciphertext = \((L_n,R_n)\)
- For each round \( i=n,n-1,\ldots,1 \), compute
  \[
  R_{i-1} = L_i \\
  L_{i-1} = R_i \oplus f(R_{i-1}, K_i)
  \]
  where \( f \) is round function and \( K_i \) is subkey
- Plaintext = \((L_0,R_0)\)
- Formula “works” for any function \( F \)
- But only secure for certain functions \( F \)
Repeating/Iterating this transformation we obtain the Feistel Cipher.
Non-Feistel Ciphers

• Composed of only invertible components.
• Input to round function consists of key and the output of previous round
• These functions are obtained by the repeated application of Substitution (invertible SBoxes) and Permutation.
• Thus they are called Substitution Permutation Networks (SPN).

Further Reading

• B. A Forouzan, Cryptography & Network Security, Tata Mc Graw Hills, Chapter 5
Points to ponder!

• State true or false:
  – The following key mixing technique is linear wrt. exclusive-or:
    • \( y = (x + k) \mod 2^8 \), where \( x \) and \( k \) are 8 bit numbers, and ‘+’ denotes integer addition.
  – Having a final permutation step in an SPN (Substitution Permutation Network) cipher has no effect on the security of a block cipher.

Next Day’s Topic

• Designs of Modern Block Ciphers:
  – Data Encryption Standard (DES)
  – Advanced Encryption Standard (AES)
Data Encryption Standard

- DES developed in 1970’s
- Based on IBM Lucifer cipher
- U.S. government standard
- DES development was controversial
  - NSA was secretly involved
  - Design process not open
  - Key length was reduced
  - Subtle changes to Lucifer algorithm

DES Numerology

- DES is a Feistel cipher
- 64 bit block length
- 56 bit key length
- 16 rounds
- 48 bits of key used each round (subkey)
- Each round is simple (for a block cipher)
- Security depends primarily on “S-boxes”
- Each S-boxes maps 6 bits to 4 bits
One Round of DES

Function f

Note that the design of DES is reduced to the design of f, which works on shorter lengths

DES Expansion

- Input 32 bits
  0 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
- Output 48 bits
  31 0 1 2 3 4 3 4 5 6 7 8
  7 8 9 10 11 12 11 12 13 14 15 16
  15 16 17 18 19 20 19 20 21 22 23 24
  23 24 25 26 27 28 27 28 29 30 31 0
DES S-box (Substitution Box)

- 8 “substitution boxes” or S-boxes
- Each S-box maps 6 bits to 4 bits
- S-box number 1

<table>
<thead>
<tr>
<th>input bits (0,5)</th>
<th>input bits (1,2,3,4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>00 0000 0001 0011 0100 0110 0111 1000 1100 1101 1110 1111 1010 1011 1001 0000</td>
<td>0000 0000 0001 0011 0100 0111 1000 1100 1101 1110 1111 1010 1011 1001 0001</td>
</tr>
<tr>
<td>01 0000 0010 0011 0100 0110 0111 1000 1100 1101 1110 1111 1010 1011 1001 0100</td>
<td>0001 0010 0011 0100 0110 0111 1000 1100 1101 1110 1111 1010 1011 1001 0101</td>
</tr>
<tr>
<td>10 0100 0001 0101 0011 0111 1000 1100 1101 1110 1111 1010 1011 1001 0101 0011</td>
<td>1001 0010 1011 1111 1000 0010 0100 1001 1001 0111 0101 0001 0111 0101 0000</td>
</tr>
<tr>
<td>11 1111 1100 1000 0010 0100 1001 0001 0111 0101 1011 0011 1110 1000 0110 0110</td>
<td>1100 0000 1110 0110 0100 1101 0111 0101 0011 0101 0001 0111 0101 0000 0100</td>
</tr>
</tbody>
</table>

For other tables refer to Stinson’s Book

S-Box with Table entries in decimal

Output=13

What is the output if input is 101000?

Row=10=2 Column=0100=4
Properties of the S-Box

• There are several properties
• We highlight some:
  – The rows are permutations
  – The inputs are a non-linear combination of the inputs
  – Change one bit of the input, and half of the output bits change (Avalanche Effect)
  – Each output bit is dependent on all the input bits

DES P-box (Permutation Box)

• Input 32 bits
  0  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
• Output 32 bits
  15  6 19 20 28 11 27 16  0 14 22 25  4 17 30  9
  1  7 23 13 31 26  2  8 18 12 29  5 21 10  3 24
DES Subkey

- 56 bit DES key, 0,1,2,...,55
- Left half key bits, $L_K$
  
  49 42 35 28 21 14  7
  0 50 43 36 29 22 15
  8 1 51 44 37 30 23
  16 9 2 52 45 38 31
- Right half key bits, $R_K$
  
  55 48 41 34 27 20 13
  6 54 47 40 33 26 19
  12 5 53 46 39 32 25
  18 11 4 24 17 10  3

DES Subkey

- For rounds $i=1,2,...,n$
  
  - Let $L_K = (L_K$ circular shift left by $r_i)$
  - Let $R_K = (R_K$ circular shift left by $r_i)$
  - Left half of subkey $K_i$ is of $L_K$ bits
    
    13 16 10 23  0  4  2 27 14  5 20  9
    22 18 11  3 25  7 15  6 26 19 12  1
  - Right half of subkey $K_i$ is $R_K$ bits
    
    12 23  2  8 18 26  1 11 22 16  4 19
    15 20 10 27  5 24 17 13 21  7  0  3
**DES Subkey**

- For rounds 1, 2, 9 and 16 the shift $r_i$ is 1, and in all other rounds $r_i$ is 2
- Bits 8, 17, 21, 24 of LK omitted each round
- Bits 6, 9, 14, 25 of RK omitted each round
- **Compression permutation** yields 48 bit subkey $K_i$ from 56 bits of LK and RK
- **Key schedule** generates subkey

**DES Some Points to Ponder**

- An initial perm $P$ before round 1
- Halves are swapped after last round
- A final permutation (inverse of $P$) is applied to $(R_{16}, L_{16})$ to yield ciphertext
- *None of these serve any security purpose*
Further Reading


Next Day’s Topic

- Linear Cryptanalysis of SPN ciphers