[CS304] Introduction to Cryptography and Network Security

Course Instructor: Dr. Dibyendu Roy — Winter 2023-2024
Scribed by: Archit Verma (202151192) — Lecture (Week 3)

Initial Permutation (IP)

The initial permutation function rearranges the 64-bit block $m_1m_2\ldots m_{64}$ as follows:

$$\text{IP}(m_1m_2\ldots m_{64}) = m_{58}m_{50}m_{42}\ldots m_7$$

The permutation matrix representing the change is given by:

$$\begin{bmatrix} 58 & 50 & 42 & \ldots & 2 \\ 60 & 52 & 44 & \ldots & 50 \end{bmatrix}$$

Algorithm of F function

For each round, the F function operates on a 32-bit block R_i and a 48-bit subkey k_i, producing a 32-bit output x_{i+1}:

$$f(R_i, k_i) = P(S(E(R_i) \oplus k_i))$$

Where:

- E expands the 32-bit block R_i to 48 bits.
- S is the S-box operation.
- P is the permutation matrix.

S-box Operation

The S-box operation $S : \{0,1\}^{48} \to \{0,1\}^{32}$ is defined as follows:

$$S(X) = Y$$

$$X = B_1B_2B_3B_4B_5B_6B_7B_8$$

$$Y = (2r + b_1)$$

where B_i is a 6-bit block and r, b_1 are binary.

The S-boxes are represented by matrices:

$$\begin{bmatrix} S_1 & S_2 & S_3 & S_4 \\ S_5 & S_6 & S_7 & S_8 \end{bmatrix}$$

with each S_i having 4 rows (0 to 3) and 16 columns (0 to 15).

Key Scheduling Algorithm of DES

DES (Data Encryption Standard) is a symmetric key block cipher that operates on 64-bit blocks of plaintext, producing 64-bit ciphertext. It employs a Feistel block cipher structure. This discussion focuses on the Key Expansion Function and Key Schedule of DES.

Key Expansion Function

The Key Expansion Function in DES is responsible for generating 16 subkeys, each of 48 bits, from the initial 64-bit key. These subkeys are crucial for the encryption of plaintext during different rounds of DES.

Initial Key (64 bits)

Let the initial key be denoted as K with the following binary representation:

$$K = 0001001100110100010101110111100110011011101111001101111111110001$$

Permuted Key (56 bits)

The initial key is permuted using the permutation box $PC - 1$, reducing its size to 56 bits. The permutation involves selecting specific bits from the original key and arranging them as per the defined order.

PC-1:

57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

The permuted key, denoted as K^+, is obtained by selecting the bits according to the order defined by $PC - 1$.

$$K^+ = 11110000110011001010101011110101010101100110011110001111$$

Splitting Into Halves and Left Circular Shifts

The permuted key K^+ is then split into left (C_0) and right (D_0) halves, each of 28 bits. The halves undergo 16 rounds of cyclic left shifts, where the number of shifts is determined by the iteration number.

Left Circular Shifts:

Iteration Number	Number of Left Shifts
1	1
2	1
3	2
4	2
5	2
6	2
7	2
8	2
9	1
10	2
11	2
12	2
13	2
14	2
15	2
16	1

For each iteration, the left and right halves (C_n, D_n) are obtained from the previous (C_{n-1}, D_{n-1}) through the specified number of left shifts.

From the original pair C_0 and D_0, the subsequent pairs are calculated as follows:

$$C_0 = 1111000011001100101010101111$$

$$D_0 = 0101010101100110011110001111$$

$$C_1 = 1110000110011001010101011111$$

$$D_1 = 1010101011001110001111100011$$

$$\vdots$$

$$C_{16} = 1111000011001100101010101111$$

$$D_{16} = 0101010101100110011110001111$$

Combining and Permuting to 48 bits

Next, the pairs C_n and D_n are combined and subjected to permutation using the permutation box $PC - 2$ to reduce the number of bits to 48.

Example for the first key (K1):

$$C_1D_1 = 11100001100110010101010111111010101011001100111100011110$$

PC-2:

14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

After applying $PC - 2$, the first subkey $K1$ is obtained:

$$K1 = 000110110000001011101111111111000111000001110010$$

Similarly, all 15 keys ($K2$ to $K16$) are generated through the Key Expansion Function.

Key Schedule

The Key Schedule in DES is the collection of all 16 subkeys generated by the Key Expansion Function. These subkeys are used during different rounds of the DES encryption process.

The complete key schedule is as follows:

$$\begin{aligned}
K1 &: 000110110000001011101111111111000111000001110010 \\
K2 &: 011110011010111011011001110110111100100111100101 \\
K3 &: 010101011111110010001010010000101100111110011001 \\
&\ldots \\
K16 &: 110010110011110110001011000011100001011111110101
\end{aligned}$$

The Key Schedule is crucial for the DES algorithm's security, and the generated subkeys contribute to the encryption process in each round.

DES Encryption

The DES (Data Encryption Standard) encryption process involves transforming a 64-bit block M with a key k using a series of well-defined steps.

1. **Initial Permutation (IP):** The 64-bit block M undergoes an initial permutation, rearranging its bits based on a predefined pattern.

2. **16 Rounds:** The initial permuted block goes through 16 rounds of a Feistel network, a specific type of cryptographic structure. Each round consists of operations such as expansion, substitution (using S-boxes), permutation, and XOR with the round key.

   - **Expansion (E):** The 32-bit block R_i is expanded to 48 bits to increase the complexity of subsequent operations.
   - **S-box Operation (S):** The expanded block undergoes the S-box operation, a crucial non-linear substitution, introducing confusion and making the encryption more secure.

- **Permutation (P):** A permutation matrix is applied to the output of the S-box operation, further enhancing diffusion and making the encryption resistant to cryptanalysis.

3. **Final Permutation (FP):** After the 16 rounds, the output undergoes a final permutation, which is the inverse of the initial permutation.

4. **Output (Ciphertext C):** The final permuted block becomes the ciphertext C.

Complemented DES Encryption

Complemented DES encryption, denoted as $\overline{\text{DES}}(\overline{M}, \overline{k})$, follows the same steps as DES encryption. However, it involves using the complemented versions of the plaintext block $\overline{M}$ and the key $\overline{k}$. This complementing introduces additional complexity and contributes to the overall security of the algorithm.

Brute Force Attack and 2^{56} Permutations

The key length of DES is 56 bits, allowing for 2^{56} possible keys. A brute force attack entails trying all possible keys until the correct one is discovered. The 2^{56} permutations illustrate the key space's size, emphasizing the necessity for a sufficiently large key space to resist brute force attacks. As computational power increases, exhaustive searches become more feasible, underscoring the need for stronger encryption methods with larger key sizes.

Example Matrix for Left Circular Shift

Consider the following example matrix representing the left circular shift of a 28-bit block by one position:

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{bmatrix}$$

This left circular shift is an essential part of the key scheduling algorithm in DES, contributing to the generation of different round keys for each round of encryption. The specific number of left shifts applied during key generation adds another layer of complexity and security to the DES algorithm.

Attack Models in Cryptography

Cipher Text Only Attack

In a cipher text-only attack, the attacker has access only to the ciphertext. The goal is to recover the plaintext or deduce the secret key.

Known Plain Text Attack

In a known plain text attack, the attacker has knowledge of some plaintext and its corresponding ciphertext. The objective is to find a plaintext corresponding to different ciphertext or discover the secret key.

Chosen Plain Text Attack

The attacker selects some plaintext of their choice and obtains the corresponding ciphertext. The goal is to generate a new plaintext, ciphertext pair, or find the secret key.

Chosen Cipher Text Attack

The attacker chooses some ciphertext and is provided with the corresponding plaintext. The goal is to generate a different valid plaintext-ciphertext pair or find the secret key.

DES (Data Encryption Standard)

For DES with keys $K_1, K_2, \ldots, K_{2^{56}}$:

$$M \to \text{DES}(M, K_i) = C_1$$

$$M' \to \text{DES}(M', K_i') = C_2$$

If the attacker observes $\text{DES}(M, K_i) = C_1$ and $\text{DES}(M', K_i') = C_2$:

- If $C_2 \neq C_1$, then $K_i \neq K_{i'}$.
- If $C_2' \neq C_1'$, then $K_i' \neq K_{i'}'$.

Single encrypted DES can be broken with only 2^{43} searches using optimal algorithms, making it insecure. Multiple encryption, such as double and triple DES, is recommended.

Double Encryption

For double encryption with keys K_0, K_1:

$$P, C \to \text{Double DES}$$

Select K_i

$$\text{Enc}(P, K_i) = X_i$$

$$\text{Dec}(C, K_j) = Y_j$$

If $X_i = Y_j$, then K_i and K_j are potential keys. The total searches are $2 \times 2^{56} = 2^{57}$, rendering double DES insecure.

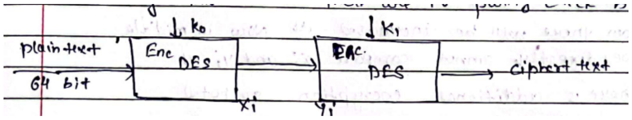

Figure 1: Double DES

Triple Encryption

In triple DES, there is an increased security layer as one encryption lies between X_i and Y_j. Search time $= 2^{56} \times 2^{56}$, providing twice the bit security, i.e., $2n$ bit security.

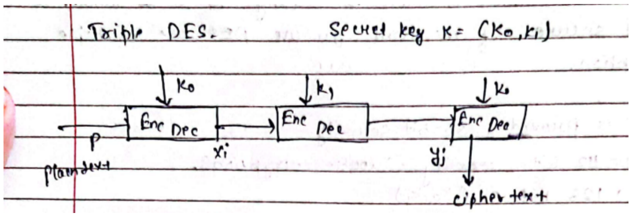

Figure 2: Triple DES

Groups and Binary Operations

A binary operation $*$ on a set S is a mapping from $S \times S$ to S. A group $(G, \cdot)$ consists of a set G with a binary operation $\cdot$ that satisfies the following axioms:

1. The group operation is associative: $(a \cdot b) \cdot c = a \cdot (b \cdot c)$.
2. There is an identity element e such that $a \cdot e = e \cdot a = a$ for all $a \in G$.
3. For each $a \in G$, there exists an element a' (inverse of a) such that $a \cdot a' = a' \cdot a = e$.

If the group is Abelian (commutative), $a \cdot b = b \cdot a$.

Groups in Different Operations

Integers Modulo n under Addition

The set of integers modulo n under addition, denoted as $\mathbb{Z}_n$, forms a group. The group operation is addition modulo n, and it satisfies the group axioms of associativity, identity element, and inverses.

Integers under Addition

The set of integers under addition, denoted as $\mathbb{Z}$, forms a group. The group operation is addition, and it satisfies the group axioms.

Matrices under Matrix Multiplication

The set of invertible matrices under matrix multiplication forms a group. The group operation is matrix multiplication, and it satisfies the group axioms of associativity, identity element, and inverses.

Integers under Subtraction

The set of integers under subtraction, denoted as $\mathbb{Z}$, does not form a group. The operation of subtraction lacks the existence of an identity element and inverses for all elements.

Integers under Division

The set of integers under division, denoted as $\mathbb{Z}$, does not form a group. The operation of division lacks the existence of an identity element and inverses for all elements.

Integers under Multiplication and Modulo

The set of integers under multiplication modulo n, denoted as $\mathbb{Z}_n^*$, forms a group. The group operation is multiplication modulo n, and it satisfies the group axioms.