[CS304] Introduction to Cryptography and Network Security

Course Instructor: Dr. Dibyendu Roy — Winter 2023-2024
Scribed by: Dipean Dasgupta (202151188) — Lecture (Week 3)

1 Data Encryption Standard(DES)

A symmetric-key block cipher developed by IBM, the Data Encryption Standard (DES) was extensively used to protect electronic data until the National Institute of Standards and Technology (NIST) formally deprecated it in 2005 because of its tiny key size. DES was a key contributor to the creation of contemporary encryption standards and is still regarded as a significant historical cryptographic algorithm even though it has been deprecated.

Block Size: **64-bit** fixed-size data blocks for operation.
Key Size: **56 bits + 8 parity bits**. total 64.
Network Structure: The network structure of DES is Feistel.A function including key mixing, substitution (S-boxes), permutation (P-boxes), and XOR operations is applied alternately to the two halves of 32 bits.
Rounds: Total **16** rounds for both encryption and decryption.

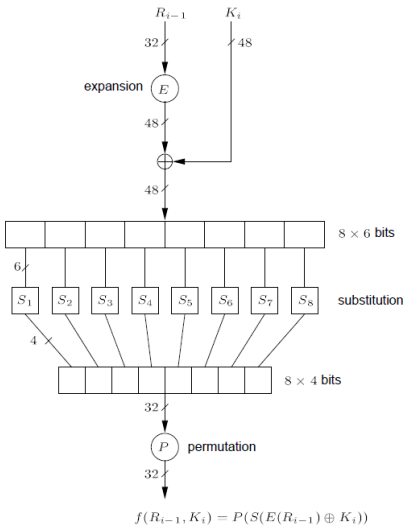

Figure:DES Encryption
Source:TutorialsPoint

ENCRYPTION:

$$Li = R^{-1};$$
$$\mathrm{R} = \mathrm{Li}^{-1} f(R_{i-1}, K_1), \text{where } f(R_{i-1}, K_1) = P(S(E(R_{i-1})K_1))$$

Encryption proceeds in 16 stages or rounds. From the input key K, sixteen 48-bit subkeys K_i are generated, one for each round. Within each round, 8 fixed, carefully selected 6-to-4 bit substitution mappings ($S-boxes$) S_i, collectively denoted S, are used. The 64-bit plaintext is divided into 32-bit halves L_o and R_o. Each round is functionally equivalent,taking 32-bit inputs L_{i-1} and R_{i-1} from the previous round and producing 32-bit outputs L_i and R_i for $1 \leq i \leq 16$.
In this case, all bits are utilized once, but some are used twice, in a fixed expansion permutation mapping(E) R_{i_1} from 32 to 48 bits. P is an additional 32-bit fixed permutation. The first round starts with an initial bit permutation (IP), the left and light halves are switched after the last round, and the resultant string is bit-permuted by the inverse of IP.

- **Initial Permutation(IP):**
  The Initial Permutation (IP) is the initial step in the encryption process in the DES (Data Encryption Standard) algorithm. Before further processing, it entails reordering the plaintext block's components. In essence, the IP operation is a fixed permutation of the plaintext's 64 bits. A specific permutation table is used to reorder the bit positions.

$$IP : \{0,1\}^{64} \rightarrow \{0,1\}^{64}$$
$$IP(m_1, m_2 \ldots m_{64}) = m_{58} m_{50} \ldots m_7$$

IP							
58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
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

IP^{-1}							
40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

Table 7.2: *DES initial permutation and inverse (IP and IP^{-1}).*

Figure: DES initial permutation and inverse (IP and IP^{-1}).
Source:HAC

- **Expansion(E)**:
  A 32-bit half-block of data (originating from the right side in DES) is expanded to 48 bits by the Expansion Permutation (E). This expansion makes use of a fixed permutation table and duplicates some of the bits. That is:

$$E : \{0,1\}^{32} \rightarrow \{0,1\}^{48}$$
$$E(x_1, x_2 \ldots x_{32}) = y_1, y_2 \ldots y_{48}$$

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

Figure: Expansion Table.
Source:HAC

- **Substitution Box(S):**
The DES algorithm becomes more secure because to the non-linearity introduced by the replacement boxes. Every S-box receives a 6-bit block as input. Eight S-boxes are used in the DES algorithm, and they are used concurrently during a particular Feistel network cycle. Every S-box produces a 4-bit block as its output. The 6-bit input value to the S-box determines the substitution, which is preset in the S-box itself.

$$E : \{0,1\}^{48} \rightarrow \{0,1\}^{32}$$
$$X \rightarrow \mathrm{B}_1 B_2 B_3 B_4 B_5 B_6 B_7 B_8 \qquad length(B_i = 6)$$
$$S_i = \{0,1\}^6 \rightarrow \{0,1\}^4 for all i = 1, 2 \ldots 8$$
$$S_i(B_i) = C_i$$
$$S(X) = S_1(B_1)S_2(B_2)S_3(B_3)S_4(B_4)S_5(B_5)S_6(B_6)S_7(B_7)S_8(B_8)$$

- **Permutation(P):**
The 64-bit block's final configuration before it is used as the output ciphertext is produced by switching the 32-bit halves from the Feistel network's final round in the Final Permutation.The Final Permutation is used to offer one last bit of mixing and rearranging before the ciphertext is generated.

$$P : \{0,1\}^{32} \rightarrow \{0,1\}^{32}$$
$$P(X_1, X_2 \ldots X_{32}) = (X_{16} X_7 X_{20} X_{21} \ldots X_{25})$$

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

Figure: Permutation Table
Source:HAC

Key Scheduling Algorithm
INPUT: 64-bit key $K = k_1k_2...k_{64}$ (including 8 odd-parity bits).
OUTPUT: Sixteen 48-bit keys K_i, $1 \leq i \leq 16$
Algorithm:
1. Define v_i, $1 < i < 16$ as follows: $v_i = 1$ for $i \in \{1, 2, 9, 16\}$; $v_i = 2$ otherwise.(These are left-shift values for 28-bit circular rotations below.)

2. TPC1(K); represents T as 28-bit halves (Co, Do). (Use PCI in image to select bits from K: Co $= k_{57}k_{49}k_{41}...k_{36}$, Do $= k_{63}k_{55}...k_4$.)

3. For i from 1 to 16, compute K_i as follows: $C_i = (C_{i-1} \lll v_i)$, $D_i = (D_{i-1} \lll v_i)$, $K_i = KPC2(C_i, D_i)$. (Use PC2 in image below to select 48 bits from the concatenation $b_1b_2...b_{64}$ of C_i and D_i: $K_i = b_{14}b_{17}...b_{63}b_2$ denotes left circular shift.)

PC1						
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
above for C_i; below for D_i						
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

PC2					
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

Figure: DES key schedule bit selections (PCl and PC2).
Source:HAC

DECRYPTION:

DES decryption consists of the encryption algorithm with the same key but a reversed key schedule, using in order K16, K15, ... K_1.This works as follows. The effect of IP^{-1} is canceled by IP in decryption, leaving (R16, L16);
Consider applying round 1 to this input. The operation on the left half yields, rather than $L_0 \oplus$ f(R_0, K_1), $now R_16 \oplus$ f(L16, K16) which, since $L_16 = R_15 and R_16 = L_15(R_15, K16), is equal to L_15(R_15, K_16) \oplus$ f(R_15, K_16) = L_15.
$Thus, round 1 decryption yields (R_15, L_15) i.e., inverting round 16$.
It should be noted that each round's cancellation happens regardless of the definition of f and the precise value of K_i

Attack Methods:

- **Brute Force attack:** Brute force attack includes exhaustive precomputation where data complexity is chosen.
  Storage Complexity= 2^{56}
  Processing Complexity = 1(table Lookup)
- **Ciphertext-only Attack:** In ciphertext-only attack, an attacker gains access to encrypted data(ciphertext) but does not have any knowledge of the associated plaintext or further details about the encryption key or technique.
  Goal: Assess the ciphertext and attempt to infer relevant details about the source plaintext or, in the absence of any further information, discover the encryption key.
- **Known Plaintext Attack:** Known Plaintext Attack(KPA) occurs when the attacker obtains access to both the original, unencrypted data known as the plaintext and the corresponding encrypted data known as the ciphertext.
  Goal: To explore how the plaintext and ciphertext relate to one another to figure out the encryption key or decrypt more ciphertexts that have the same encryption.
- **Chosen Plaintext Attack**: In this process, some of the plaintext and ciphertext text is known. based on the known information, the search for secret keys continues. Secret Key is found using the chunks of Plaintext and Cipher text.

  $$DES(M, K) = C_1 \qquad DES(M', K) = C_2$$

  $DES(M, K_i) = C$
  $if \quad C \neq C_1 \quad discard \quad K_i \quad from S$
  $if \quad C' \neq C_2 \quad discard \quad K'_i \quad from S$
  $DES(M', K'_i) = C$
  So, here data complexity is known. So in one round if not match both K and K' are discarded. so Processing complexity reduced from 2^{56} to 2^{55}.
- **Chosen Ciphertext Attack:** In Chosen Ciphertext Attack (CCA) the attacker can obtain the decryption of chosen ciphertexts. In other words, the attacker can choose arbitrary ciphertexts and figure out their corresponding plaintexts through a decryption oracle or in some other ways/methods.
  Goal:to obtain knowledge of the encryption key and use that knowledge to carry out additional malicious deeds.

Attack on DES:

Lets consider a scenario of Chosen Plaintext Attack on DES. In Brute force attack, one has to check combination of $K_1 K_2 \dots K_{2^{56}}$.

In case of Chosen Plaintext Attack(CPA):

$$1) M \rightarrow DES(M, K) = C_1$$
$$2) \overline{M} \rightarrow DES(\overline{M}, K) = C_2$$

For Attacker,

$$DES(M, K_i) = \widetilde{C}_i$$
$$if \ \widetilde{C}_i \neq C_1 \rightarrow K_i \neq K$$
$$if \ \widetilde{C}_i \neq C_2 \rightarrow \widetilde{K}_i \neq K$$
$$DES(\widetilde{M}, \widetilde{K}) = \widetilde{C}_2$$

After these 4 steps, M is recovered.

2 2-DES:

Double DES is a type of encryption where the same plain text is encrypted using two instances of DES. Different keys are used in each case to encrypt the plain text. When decrypting, both keys are necessary. The 64-bit plain text is sent to the first DES instance, where it is first converted, using the first key, into a 64-bit middle text. Next, it is sent to the second DES instance, where it is ciphered, using the second key.$2 - DES$ provides 112-bit security instead of 56-bit security provided by DES. Here, Key $K \rightarrow 128bit = (k_0, k_1)$. Among 128 bits, 16 are parity check bits.

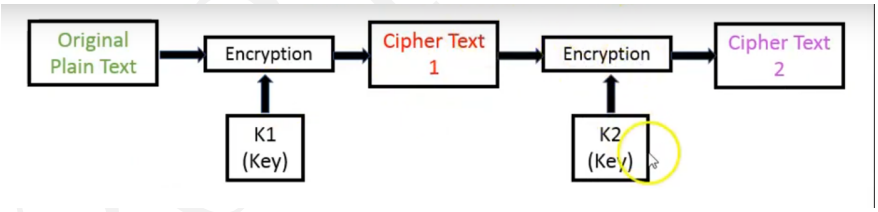

Figure: 2-DES Encryption Process
Source:GeeksforGeeks

Here, the attacker has only one valid plaintext(P) and ciphertext(C) pair.

$P, C \rightarrow DoubleDES$

Select K_i,

$$\text{ENC}_{DES}(P, K_i) = X_i \qquad Table1 : (X_i, K_i)$$
$$\text{ENC}_{DES}(C, K_j) = Y_j \qquad Table2 : (Y_j, K_j)$$

$if \quad X_i = Y_j \quad then, \quad (\text{K}_i, K_j) \rightarrow SecretKey$

3 Triple DES:

Three instances of DES are used on the same plain text in the Triple DES encryption technique. It makes use of several key-choosing techniques: in the first, every key utilised is unique; in the second, two keys are identical and one is unique; and in the third, every key is identical.

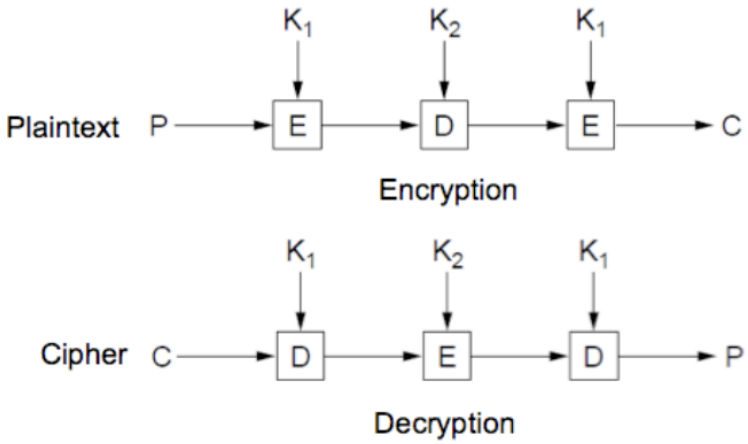

Figure: Triple-DES
Source:ResearchGate

$$C = E(K_1, D(K_2, E(K_3, \mathrm{P})))$$

- E represents the encryption function.
- D represents the decryption function.
- $K_1, K_2,$ and K_3 are three different 56-bit DES keys.
- P is the input data block.
- C is the encrypted output.

Let's break down the steps:

First Encryption (E with K_1):

$$\text{Intermediate\_1} = E(K_1, \text{Plaintext})$$

Decryption (D with K_2):

$$\text{Intermediate\_2} = D(K_2, \text{Intermediate\_1})$$

Second Encryption (E with K_3):

$$\text{Ciphertext} = E(K_3, \text{Intermediate\_2})$$

Due to its susceptibility to meet-in-the-middle attacks, Triple DES only offers a total security level of 2^{112}, as opposed to 168 bits of key. Since short blocks allow for the encryption of enormous text volumes with the same key, block collision attacks are also possible. Moreover, it is susceptible to sweet32 attacks.

4 Mathematical Recall:

Binary Operation:
A binary operation on a set is reffered as mapping from same set to same set. It is denoted by $*$. That is:

$$* = S \times S \quad S = \text{a set}$$

It plays a role which assigns to each ordered pair of elements from S to an element of S.
Properties:
$*(a, b) = c \qquad a, b, c \in S$
$*(b, a) = d \qquad d \in S$
It is not necessary that d=c.

Group:
A group $(G, *)$ consists of a set G with a binary operation $*$ on G that satisfies the following three conditions:

- The group operation is associative. That is, $a \cdot (b \cdot c) = (a \cdot b) \cdot c$ for all $a, b, c \in G$.
- There is an element $1 \in G$, called the identity element, such that $a \cdot 1 = 1 \cdot a = a$ for all $a \in G$.
- For each $a \in G$ there exists an element $a^{-1} \in G$, called the inverse of a, such that $a \cdot a^{-1} = a^{-1} \cdot a = 1$.

A group G is abelian (or commutative) if, $a \cdot b = b \cdot a$ for all $a, b \in G$.
A group G is finite if $|G|$ is finite. The number of elements in a finite group is called its order.
Examples of Group:

- (G,*) here G is invertible nxn matrix and * is ordinary matrix multiplication $\rightarrow$ a group
- ($\mathbb{Z}$,+) is a group
- ($\mathbb{Z}$,x) x=multiplication is not a group
- (Q,*) here Q is set of all rational numbers and * is standard multiplication operation $\rightarrow$ a group
- $(\mathbb{Z}_n, +_n)$ where $\mathbb{Z}_n$ refers to, let's say, $\mathbb{Z}_{26}$ – a group.
- $(\mathbb{Z}_n, *_n)$ where $*_n$ represents multiplication mod n. This is not a group.