[CS304] Introduction to Cryptography and Network Security

Course Instructor: Dr. Dibyendu Roy — Winter 2023-2024
Scribed by: Aniruddh Modi (202151022) — Lecture 4 5 (Week 3)

1 Lecture 4

1.1 Data Encryption Standard

1.1.1 Initial Permutation function and its inverse

During the encryption (and decryption), we use an initial permutation function IP to permute the bits of the data.

$$IP : \begin{pmatrix} 1 & 2 & 3 & . & . & . & 64 \\ 58 & 50 & 42 & . & . & . & 7 \end{pmatrix}$$

This is a predefined function which maps the bits of input to the above permutation.

$$IP(m_1 m_2 m_3 ... m_{64}) = m_{58} m_{50} m_{42} ... m_7$$

We also use its inverse IP^{-1} in the encryption / decryption. Calculating it is trivial.

$$IP^{-1} : \begin{pmatrix} 1 & 2 & . & . & . & 64 \\ 40 & 8 & . & . & . & 25 \end{pmatrix}$$

1.1.2 Function F of the Feistel rounds

$$F : \{0,1\}^{32} \times \{0,1\}^{48} \rightarrow \{0,1\}^{32}$$

This function F takes a 32 bit input text R_i and a 48 bit round key K_i and outputs a 32 bit result.

$$F(R_i, K_i) = X_{i+1}$$

Now, this function F, in the DES, is equal to :

$$F(R_i, K_i) = P(S(E(R_i) \oplus K_i))$$

Where,

1. E is a function which expands R_i to 48 bits so that it is compatible for the xor operation with K_i.
2. S is a function which converts a 48 bit input into a 32 bit output.
3. P is a permutation on a 32 bit integer

1.1.2.1 E-step

Expansion step (or E-step) uses a function E.

$$E : \{0,1\}^{32} \rightarrow \{0,1\}^{48}$$

The expansion by E takes place by using the following table

$$\begin{bmatrix} 32 & 1 & 2 & 3 & 4 & 5 \\ 4 & 5 & 6 & 7 & 8 & 9 \\ 8 & 9 & 10 & 11 & 12 & 13 \\ 12 & 13 & 14 & 15 & 16 & 17 \\ 16 & 17 & 18 & 19 & 20 & 21 \\ 20 & 21 & 22 & 23 & 24 & 25 \\ 24 & 25 & 26 & 27 & 28 & 29 \\ 28 & 29 & 30 & 31 & 32 & 1 \end{bmatrix}$$

Now, suppose we have some input X of 32 bits, then,

$$E(x_i x_2 x_3 \ldots. x_{32}) = x_{32} x_1 x_2 x_3 x_4 x_5 x_4 x_5 x_6 \ldots x_1$$

1.1.2.2 Substitution Box

Substitution Box (or S-box) uses a function S.

$$S : \{0,1\}^{48} \rightarrow \{0,1\}^{32}$$

S takes in input X of 48 bits and outputs some Y of 32 bits.

$$S(X) = Y$$

To do this, first divide X into 8 blocks, each of 6 bits.

$$X = B_1 B_2 \ldots B_8, \text{where } |B_i| = 6$$

For each block i, we define

$$S_i : \{0,1\}^6 \rightarrow \{0,1\}^4$$

$$S(X) : (S_1(B_1), S_2(B_2), \ldots, S_8(B_8))$$

Now, let $B_i = b_1 b_2 b_3 \ldots b_6$. We define two variables r and c as follows.

$$r = 2b_1 + b_6$$

$$c = b_2 b_3 b_4 b_5$$

Therefore, $0 \leq r \leq 3$ and $0 \leq c \leq 15$

Now, for each S_i, we have predefined tables of dimensions 4×16. So, for each B_i, we calculate the pair (r, c) and return the value corresponding to the cell (r, c).

$$S_i = (r, c)_{th} \text{ entry of the table}$$

1.1.2.3 Permutation Box

Permutation Box (or P-box) uses a function P.

$$P : \{0,1\}^{32} \to \{0,1\}^{32}$$

$$P(X_1X_2...X_{32}) = P(X_{16}X_7...X_{25})$$

1.1.3 Key scheduling algorithm

Since DES uses 16 rounds of Feistel Cipher, we need 16 keys for encryption / decryption. So, an algorithm was devised which takes in input a 64 bit key and produces 16 keys, each of 48 bits.

$$K \to K_1, K_2, K_3, ..., K_{16}$$

So, the algorithm goes like this,

1. Define some constants v_i such that $v_i = \begin{cases} 1, & i \in \{1,2,9,16\} \\ 2, & \text{otherwise} \end{cases}$
2. Discard the parity bits. $K \to \tilde{K}$
3. Perform a permutation PC1 on K. $T = PC1(\tilde{K})$ where $T : \{0,1\}^{56} \to \{0,1\}^{56}$
4. Now, divide the permuted Key T in two halves C_0 and D_0. $(C_0, D_0) = T$. $|C_0| = |D_0| = 28$
5. Then follow the following algorithm.

   for $i = 1$ to 16 **do**
   $C_i \leftarrow C_{i-1} \hookleftarrow v_i$
   $D_i \leftarrow D_{i-1} \hookleftarrow v_i$
   $K_i = PC2(C_i, D_i)$
   end for ▷ Here, $\hookleftarrow$ is the circular left shift operator.

PC1 and PC2 are some permutations. $PC1 : \{0,1\}^{56} \to \{0,1\}^{56}$ and $PC2 : \{0,1\}^{56} \to \{0,1\}^{48}$. C_0 and D_0 are first 28 and last 28 bits of the output of $PC1$ respectively. Meanwhile $PC2$ has a 8×6 table for the permutation.

1.1.4 Properties of DES

$$\overline{\text{DES}(\overline{M}, \overline{K})} = \text{DES}(M, K) \tag{1}$$

$$\overline{\text{PC1}(\overline{K})} = \text{PC1}(K) \tag{2}$$

$$\overline{\text{PC2}(\overline{K})} = \text{PC2}(K) \tag{3}$$

$$\overline{\text{IP}(\overline{M})} = \text{IP}(M) \tag{4}$$

2 Lecture 5

2.1 Attack Models

1. Ciphertext only attack :

   - Attacker only have the Cipher Text

   Goal : To get back the plaintext or recover the secret key

2. Known plaintext attack :

   - Attacker knows the plaintext and the corresponding cipher text

   Goal : Find a plaintext corresponding to different ciphertext or recover the secret key

3. Chosen plaintext attack :

   - Attacker chooses some plaintext and is allowed to get corresponding encyptions

   Goal : To generate a valid (plaintext, ciphertext) pair or recover the secret key

4. Chosen ciphertext attack :

   - Attacker chooses some ciphertexts and is allowed to get corresponding plaintexts

   ***** Useful for public key cryptography

   Goal : To generate a valid (plaintext, ciphertext) pair or recover the secret key

2.2 Chosen plaintext attack on DES

In DES, we know that the key is of 56 *bits*. Hence, a brute force attack would require an attacker to check 2^{56} keys to get the secret key.

But, with the chosen plaintext attack, we can bring down our search space to 2^{55}. We will use the following property of DES for this :

$$\overline{\mathrm{DES}(\overline{M}, \overline{K})} = \mathrm{DES}(M, K) \tag{5}$$

Let K = $\{K_1, K_2, K_3, ...K_{2^{56}}\}$ be the set of all possible keys. Let M be our chosen plaintext. Then we will generate two cipher texts from it.

- $C_1 = DES(M, K)$
- $C_2 = DES(\overline{M}, K)$

2.2.1 Algorithm

for $K_i \in K$ **do**
  $\tilde{C_i} \leftarrow DES(M, K_i)$
  if $\tilde{C_i} \neq C_1$ **then**
    Discard K_i
  end if
  if $\tilde{C_i} \neq \overline{C_2}$ **then**
    Discard $\overline{K_i}$
  end if

end for

The discard of $\overline{K_i}$ is based on following :

$$\textbf{Since, } DES(\overline{M}, K) = C_2$$
$$\implies DES(\overline{\overline{M}}, \overline{K}) = \overline{C_2}$$
$$\implies DES(M, \overline{K}) = \overline{C_2}$$

2.3 Double DES

In double DES, we use two keys K1, K2 in hope of providing enhanced security.

$$K = (K_0, K_1) \rightarrow 128 \text{ bits}$$

Using the K_1, we first perform the encryption (or decryption) using DES. Now, using the result obtained in the previous step, we perform encryption (or decryption) using K_2. This produces our desired cipher text.

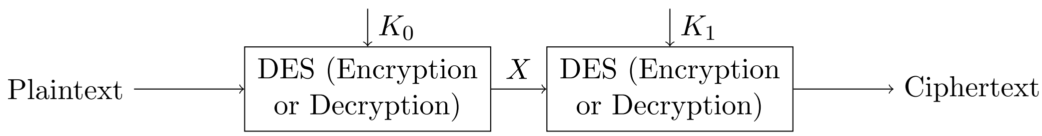

Figure 1: Double DES Encryption

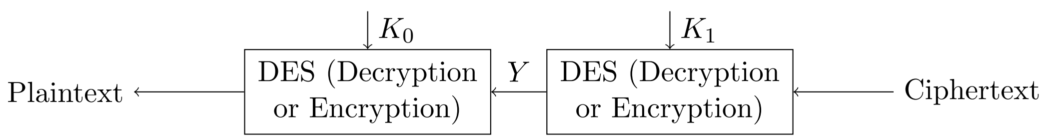

Figure 2: Double DES Decryption

$$DES(M, K_0) = C_1$$
$$DES(C_1, K_1) = C_2$$

We hoped that this provides enhanced security but this cipher fails deliberately using the **meet in the middle attack** and an attacker is able to break it in almost the same complexity as single DES.

2.3.1 Meet in the middle attack

Suppose for generating ciphertext, we are first encryption using K_0 and then decrypting using K_1. Then, to get the plaintext back, we will have to first encrypt the cipher text using K_1 and then decrypt it using K_0.

X and Y are defined in the Figure 1 and Figure 2 respectively. To break this model in less than 128 *bits* complexity, we will first generate a valid (M, C) pair. Let K = $\{K_1, K_2, K_3, ...K_{2^{56}}\}$ be the set of all possible keys. Now, we will build two tables, T_1 and T_2 using the following algorithm.

2.3.1.1 Algorithm

for $K_i \in K$ **do**
  $X_i \leftarrow DES_{Decryption}(M, K_i)$
  Store (X_i, K_i) in T_1
  $Y_i \leftarrow DES_{Encryption}(C, K_i)$
  Store (Y_i, K_i) in T_2
end for

Now, for some (i, j), if $X_i = Y_j$, then the desired pair of secret keys will be (K_i, K_j). **Hence, double DES does not provide any enhanced security.**

2.4 Triple DES

In triple DES, we use two keys K_1 and K_2 (Just like we did in double DES. Now, to deal with the *meet in the middle attack*, we will modify our model as follows :

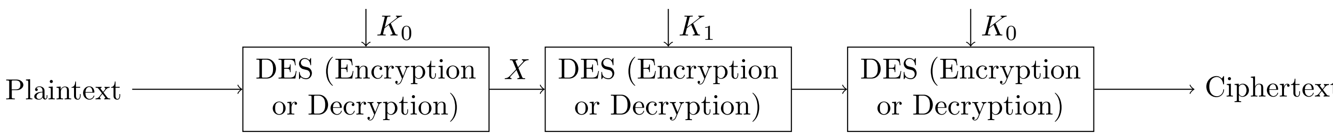

Figure 3: Triple DES Encryption

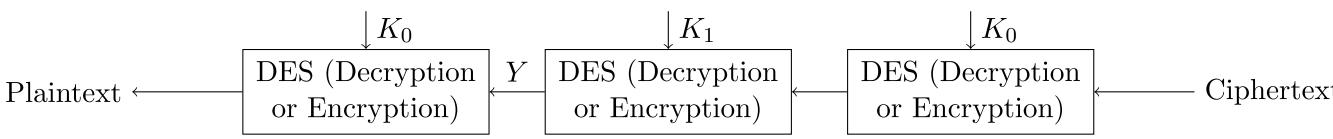

Figure 4: Triple DES Decryption

Because of the third encryption using K_0, if an attack tries to guess X and Y, then in the case of Y, the key search space will be multiplied for K_0 and K_1, that is, to get a valid Y_i, the attacker will have to go through all possible (K_{0_i}, K_{1_j}) pairs. Now, the number of such pairs are $2^{56} \times 2^{56}$. Hence, meet in the middle attack is not a good choice here.

Hence, if some algorithm provides $n - bit$ security and we desire to enhance it to $2n - bit$, then we must use a $2n - bit$ key with triple layer set up.

2.5 Maths

2.6 Binary operation

A binary operation $*$ on S is a mapping from $S \times S$ to S itself.

$$* : S \times S \rightarrow S$$

Now, suppose we have a binary operation on a and b.

$$*(a, b) = c \text{ where } a, b, c \in S$$

$$*(b, a) = d \text{ where } a, b, d \in S$$

Then, $d = c$ is not necessarily true.

2.6.1 Group

A binary operation on set G, $(G, *)$ is defined as group if it follows the following axioms :

Assoviative : $a * (b * c) = (a * b) * c$

Identity Element : There exists an element, denoted by 1 in the set G such that

$$a * 1 = 1 * a = a \ \forall a \in G$$

Inverse Element : These exist an element $a^{-1} \in G \ \forall a \in G$ such that,

$$a * a^{-1} = a^{-1} * a = 1$$

where 1 is the identity element.

Some points about groups.

- Now, for some group G, if following is true then G is **abelian** or commutative.

$$a * b = b * a \ \forall a, b \in G$$

- Also, if $|G|$ is finite, $\implies$ $(G, *)$ if also finite.

2.6.1.1 Example 1

Let * be the matrix multiplication and G be set of all *invertible* matrices, then $(G, \times)$ is a group.

- $A \times (B \times C) = (A \times B) \times C$
- $A \times I = I \times A = A$
- $A \times A^{-1} = A^{-1} \times A = I$

2.6.1.2 Example 2

Let * be the addition and Z be set of all integers, then (Z, +) is a group.

- $a + (b + c) = (a + b) + c$
- $a + 0 = 0 + a = a$
- $a + (-a) = (-a) + a = 0$

2.6.1.3 Example 3

Let * be the multiplication and Z be set of all integers, then (Z, .) is **not** a group.

- $a.(b.c) = (a.b).c$
- $a.1 = 1.a = a$
- a^{-1} does not exist for all $a \in Z$

2.6.1.4 Example 4 Let * be the subtraction and Z be set of all integers, then (Z, +) is **not** a group.

- $a - (b - c) \neq (a - b) - c$
- $a - 0 \neq 0 - a \neq a$

2.6.1.5 Example 5 Let * be the multiplication and $Q \setminus \{0\}$ be set of all rational numbers except 0, then (Q, .) is a group.

2.6.1.6 Example 5 Let * be $+_n$ (addition under modulo n) and Z_n whole numbers less than n, then $(Z_n, +_n)$ is a group.