Scribe: Cryptography and Network security (Week 6: Class 2)

Rohit - 20CS60R71

08-Oct-2020

1 Feistel permutation

Feistel cipher is a type of block cipher design. It is not a specific cipher. In this, plaintext block is split into left half (L_0) and right half (R_0).So,

$$Plaintext = (L_0, R_0)$$

Then several round of operations are applied on this splited plaintext. And finally we get output, which is the ciphertext. For each round i = 1,2,...,n, following operations computed

$$L_i = R_{i-1}$$

$$R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$$

where f is round function and K_i is subkey for round i. After nth round we obtain ciphertext (L_n, R_n).

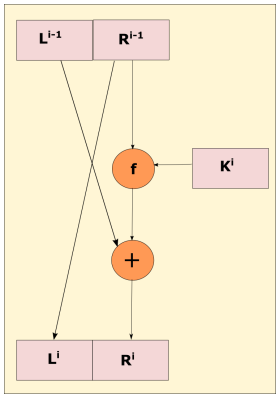

Figure 1: Feistel cipher encryption for round i. By repeating/iterating this transformation we obtain the feistel cipher.

We can observe that it is reversible, irrespective of what function we choose for f because if we know L_i then from first equation we get R_{i-1}. Then if we know k_i then value of function f can be calculated from R_{i-1} and K_i. Then,

$$L_{i-1} = R_i \oplus F(R_{i-1}, K_i)$$

Therefore, we can say that feistel cipher is invertible, irrespective of what function we choose for f. This is the basic principle of decryption. Therefore, decryption routine can be written as

$$R_{i-1} = L_i$$

$$L_{i-1} = R_i \oplus F(R_{i-1}, K_i)$$

where f is round function and K_i is subkey of round i.

2 Non-Feistel Ciphers

These types of ciphers are composed of only invertible coponents. In these, input to the round function f consist of key of that round and the output of previous round. These functions are obtained by the repeated application of substitution and permutation. Thus they are called substitution permutation networks (SPN). Here SBoxes need to be invertible because otherwise we will not be able to do decryption algorithm.

3 Data Encryption Standard (DES)

It is a very old cipher, developed in 1970's.It is based on older cipher named as Lucipher cipher designed by IBM. It is a Federal Information Processing Standard (FIPS).It is a first decryption algorithm that comes from military domain to pulic domain. It's development is very controversial. It's design process was not open, people feared of hidden trapdoors that would have allowed NSA to decrypt messages without keys. In this key length was small, equal to 56 bits and the remaining 8 bits are used for parity computation. So, People at this point, do not like it much and therefore we look for alternatives and AES is one of that choices.

4 DES Numerology

DES is an example of a Feistel cipher. It has got a 64 bit block length. The cryptography key has got a 56 bits and 8 bits of block is used for parity computation. It has 16 rounds for encryption. In each round 48 bits of key is used and this is called subkey. Every round is very simple because it has got blocks which are easy to compute and they are also similar i.e. every round has got same functionality, otherwise if it has different functionality in every round then

we have to do n implementations. All the rounds are symmetric in their construction. Primarily, there are box implementations which are called "S-boxes". Security depends primarily on these S-boxes. Each of these S-boxes are actually non-bijective because S-boxes makes our function f and for reversibility of cipher, it is not important that function f should be bijective and theresfore the constituent S-boxes inside the f function need not be bijective. Each S-boxes maps 6 bits to 4 bits.

5 Initial Permutations

DES has an initial permutation and a final permutation after 16 rounds. When we get a plaintext then a initial permutation is performed called as IP and then encryption algorithm is applied. And when we get cipher text at the end of encryption then a final permutation happen on it called as inverse permutation. These two permutations are inverses of each other and operates on 64 bits because dimension of input is 64 bits. These permuation part is also known to adversaries because our algorithm is open. They have no cryptographic significance but it makes implementation more tenable. But the designers did not disclose their purpose.

6 One round of DES

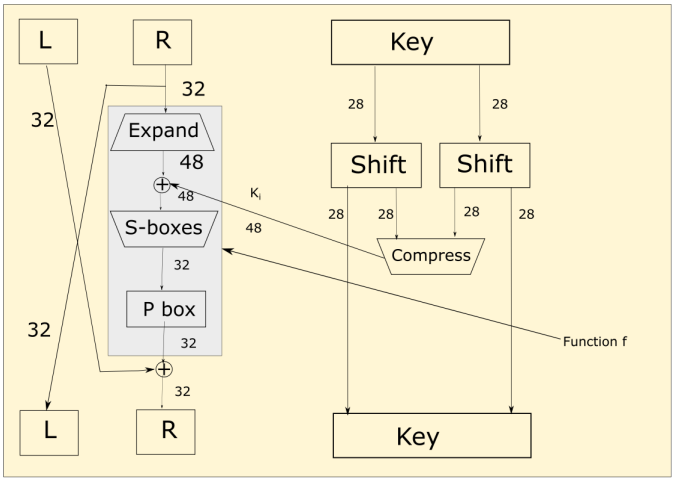

Figure 2: Illustration of one DES round

In starting, 8 bits are removed from input plaintext which are used for parity and error correction purposes. And then remaining 56 bits are splited into two equal parts. Then circular shift is done on these parts. Then a compression function is applied on each parts. It generates output of 48 bits. It compresses

56 bits to 48 bits. This is essentially the round key of 48 bits, it is applied as input to round function along with input right half. In round function, 32 bits of right half are applied to expand block, it expands these to 48 bits. Then these 48 bits are XORed with 48 bits of round key. The result is applied to S-boxes. Dimension of S-boxes is 6x4 (6 inputs, 4 outputs). So, we need 8 S-boxes. All of these S-boxes produces 4 bits, So we get total 32 bits at output. Then it is input to permutation layer, which produce 32 bits output. Then, it is XORed with 32 bits left half bits. And we get right 32 bits and left 32 bits are exactly the right 32 bits of input. So, these 64 bits are propagated to next round.

7 DES Expansion

In DES expansion, we take a input of 32 bits and got an output of 48 bits.

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

We are doing extension, So some input bits are repeating. For example, 0 comes two times (2nd and last bit).

8 DES S-box (Substitution Box)

There a 8 "Substitution boxes" or S-boxes are present in a single S-box block. Each S-box maps 6 bits to 4 bits.

input bits (0,5)

↓ input bits (1,2,3,4)

| 0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111

---

00 | 1110 0100 1101 0001 0010 1111 1011 1000 0011 1010 0110 1100 0101 1001 0000 0111

01 | 0000 1111 0111 0100 1110 0010 1101 0001 1010 0110 1100 1011 1001 0101 0011 1000

10 | 0100 0001 1110 1000 1101 0110 0010 1011 1111 1100 1001 0111 0011 1010 0101 0000

11 | 1111 1100 1000 0010 0100 1001 0001 0111 0101 1011 0011 1110 1010 0000 0110 1101

Figure 3: S-box table for mapping 6 input bits to 4 output bits

From this S-box table, we can see that the row is selected by 0th and 5th input bit and column is selected by 1st, 2nd, 3rd and 4th bits. From this we obtain the 4 output bits.

9 S-Box with table entries in decimal

S_1															
14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

Figure 4: S-box with table entries in decimal

In this table, the four bits of output of previous table are converted to decimal. So, suppose our input bit is 101000. We get the row number from 1st bit and last bit, so row index is '10' and we get column index from remaining bits, so column index is '0100'.

$$Row = 10 = 2$$

$$Column = 0100 = 4$$

So, going to row from 0 to 2 and to column from 0 to 4, we reach the output decimal number equal to 13.

$$Output = 13$$

We can observe one thing that in above table is that decimal numbers in each row are permutations of numbers from 0 to 15.

10 Properties of the S-Box

Some of the properties are:-

1. Each row of S-Box table is permutations of numbers from 0 to 15
2. The outputs are a non-linear combination of the inputs. Each of the output bit can be expressed as a boolean function of input bits. Each one has n terms in their representation. And linear combination of output bits is also non-linear in terms of input bits.
3. If we change one bit of the input, then on an average half of the output bits will change. This property is known as Avalanche Effect.
4. Each output bit is dependent on all the input bits, otherwise this will make avenues of attack.

All this properties need not only satisfied by each output bit but also need to satisfy by linear combination of all output bits.

11 Exercise

Prove that decryption in DES can be done by applying the encryption algorithm to teh ciphertext, with the key schedule reversed

Because we want to encrypt and decrypt from same DES algorithm or device, Halves are swapped after last round and before final inverse permutation. So, We get ciphertext (R_{16}, L_{16}) after complete encryption.
For Decryption we input $(R_{16}.L_{16})$ and give K_{16} as key of 1st round. then R_{15} is become equal to L_{16} and L_{15} is obtained by XOR of R_{16} with output of round function f(L_{16}, K_{16}). After performing 16 rounds we get (R_0, L_0) at output and after swapping we obtain our original plaintext. Decryption and encryption equation of DES are same, just we need to do a swap between right and left halves.
Encryption

$$L_i = R_{i-1}$$

$$R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$$

Decryption

$$R_{i-1} = L_i$$

$$L_{i-1} = R_i \oplus F(R_{i-1}, K_i)$$

12 Weak keys

DES has also got lot of bad keys which are called as weak keys. A weak key is the one which after parity drop operation, consists either of all 0's, a 1's or half 0's and half 1's. Four out of the 2^{56} keys are weak keys.

Keys before parity drop (64 bits)	Actual key (56 bits)
0101 0101 0101 0101	0000000 0000000
1F1F 1F1F 0E0E 0E0E	0000000 FFFFFFF
E0E0 E0E0 F1F1 F1F1	FFFFFFF 0000000
FEFE FEFE FEFE FEFE	FFFFFFF FFFFFFF

Figure 5: Example of weak keys

Consequences of weak keys are:-

1. The round keys created from any of these weak keys are the same. For example, for the first weak key, all the round keys are 0. And if we take the second key then it will lead to half 0s and half 1s.

2. If we encrypt a block with a weak key and subsequently encrypt the result with the same weak key, then we get the original block. Suppose if all the right keys are 0's then writting then in reverse order does not make any change. So, we get the original block by using same keys. And this not desired because we do not want the ability to encrypt will get back to original keys.

13 Semi weak keys

A semi weak key creates only two different round keys and each of them is repeated eight times. There are six key pairs that are called semi-weak keys. The round keys created from each pair are the same in different order.

1	9153E54319BD	6EAC1ABCE642
2	6EAC1ABCE642	9153E54319BD
3	6EAC1ABCE642	9153E54319BD
4	6EAC1ABCE642	9153E54319BD
5	6EAC1ABCE642	9153E54319BD
6	6EAC1ABCE642	9153E54319BD
7	6EAC1ABCE642	9153E54319BD
8	6EAC1ABCE642	9153E54319BD
9	9153E54319BD	6EAC1ABCE642
10	9153E54319BD	6EAC1ABCE642
11	9153E54319BD	6EAC1ABCE642
12	9153E54319BD	6EAC1ABCE642
13	9153E54319BD	6EAC1ABCE642
14	9153E54319BD	6EAC1ABCE642
15	9153E54319BD	6EAC1ABCE642
16	6EAC1ABCE642	9153E54319BD

Figure 6: A sample round key generation

In the ablove figure, we calculated all the 16 round keys. We can observe that there are exactly 8 same round keys in each semi-weak keys. Also, the round key in the first set is the same as the 16th key in the second set. This means that the keys are inverses of each other.

$$E_{k2}(E_{k1}(P)) = P$$