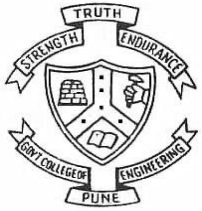
Cryptography and Network Security

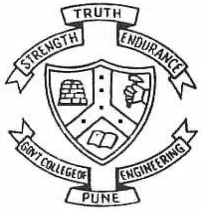
Session 14

Dr. V. K. Pachghare



Department of Computer Engineering and Information Technology
College of Engineering Pune (COEP)
Forerunners in Technical Education

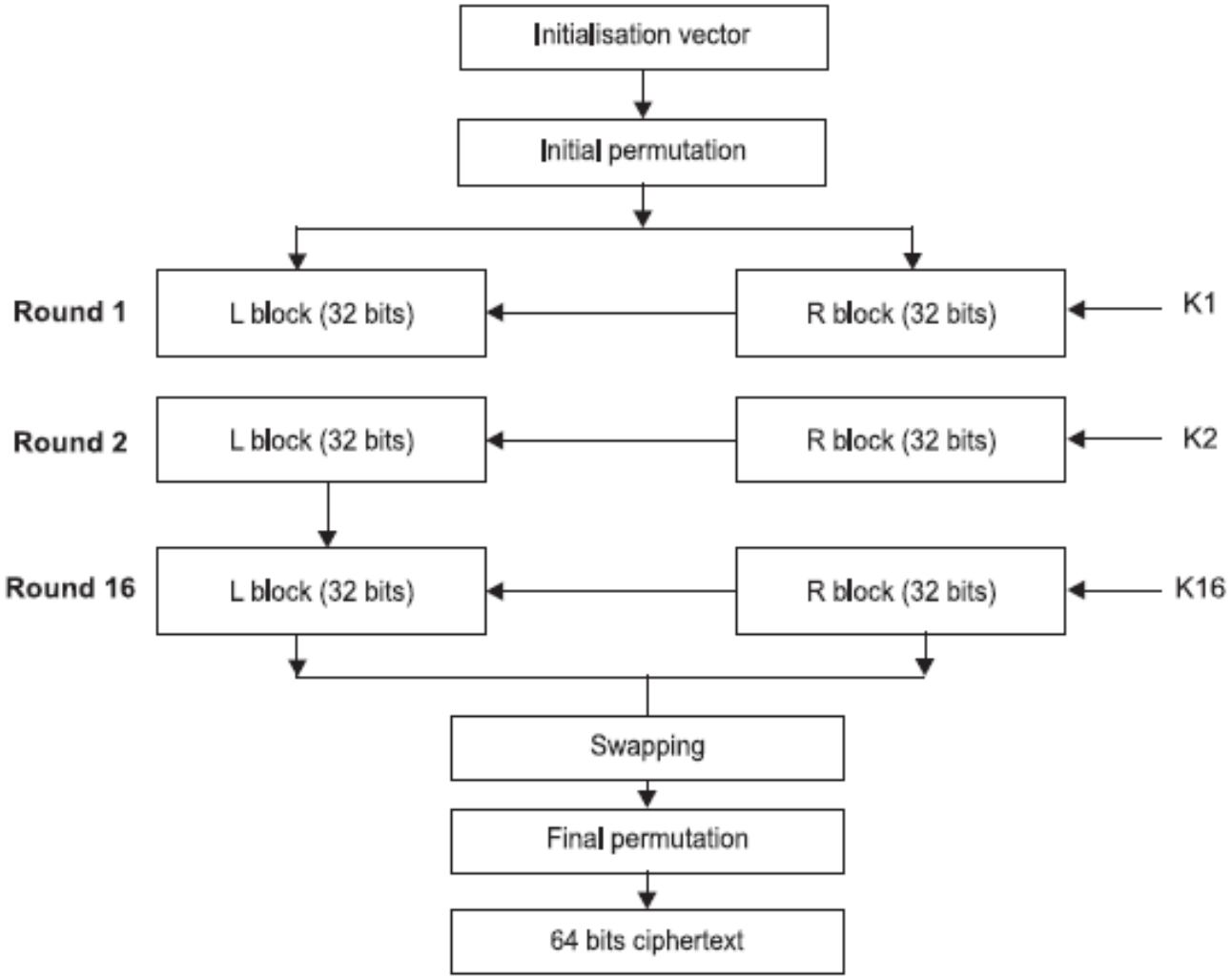
Data Encryption Standard (DES)



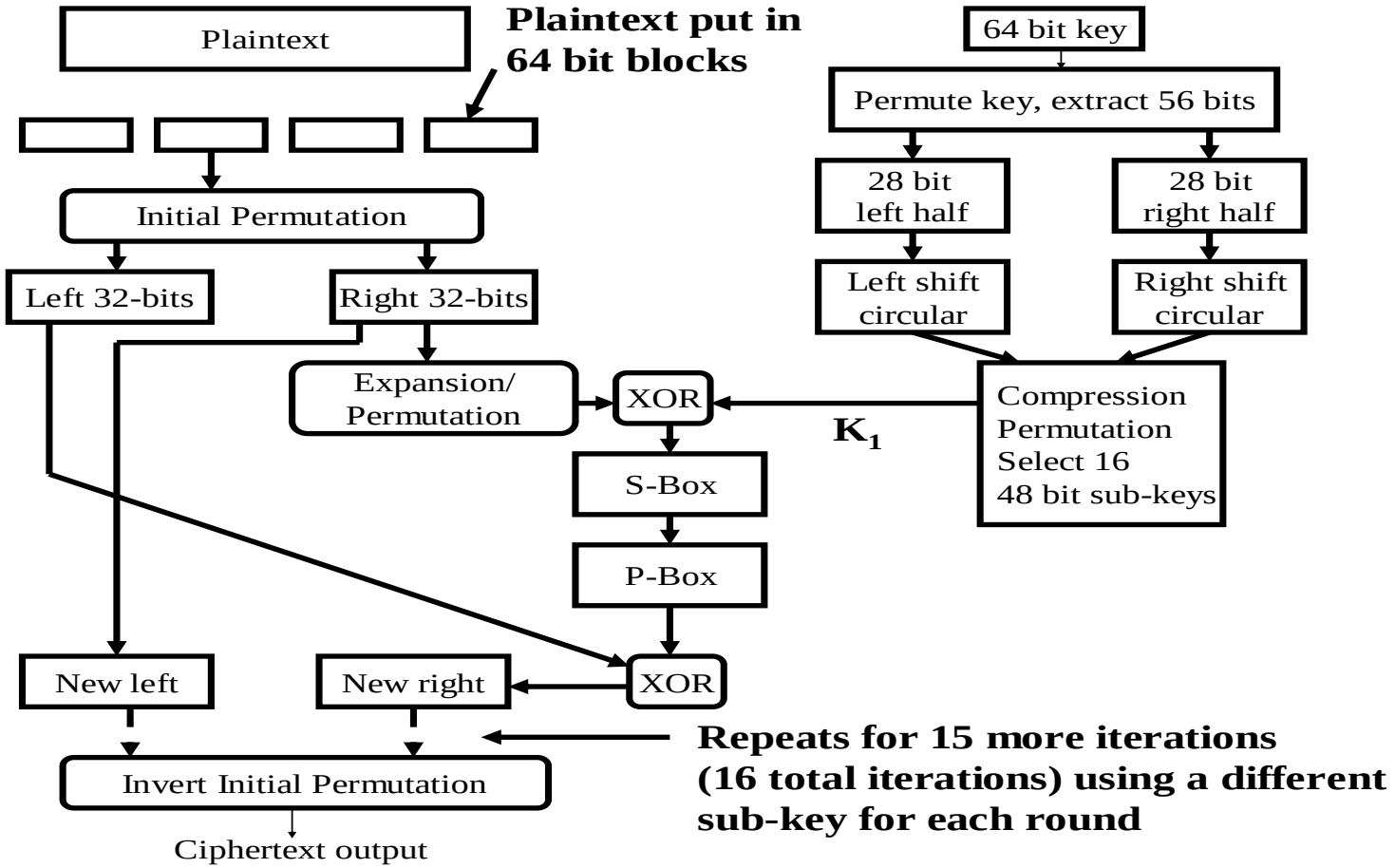
Introduction

- Block cipher.
- The design based on Feistel-structure.
- DES was published in 1977.
- Plaintext block size of 64 bits
- Key size 64 bits key (actually DES uses 56 bits out of 64 bits)
- 16-rounds

- Design of DES is based on Feistel structure
- Various permutations and S-boxes (substitution boxes) are used to provide confusion and diffusion


Initialisation vector
Initial permutation
Round 1
L block (32 bits)
R block (32 bits)
K1
Round 2
L block (32 bits)
R block (32 bits)
K2
Round 16
L block (32 bits)
R block (32 bits)
K16
Swapping
Final permutation
64 bits ciphertext

DES - More Detailed Block



Permutation Table

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

Expansion Permutation [56-bits]

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

Compression Permutation [48-bits]

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

S Box

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

Details of DES

1. SUB KEY GENERATION
2. INITIAL PERMUTATION
3. ROUND 1 TO 16
4. FINAL PERMUTATION

Sub-Key Generation

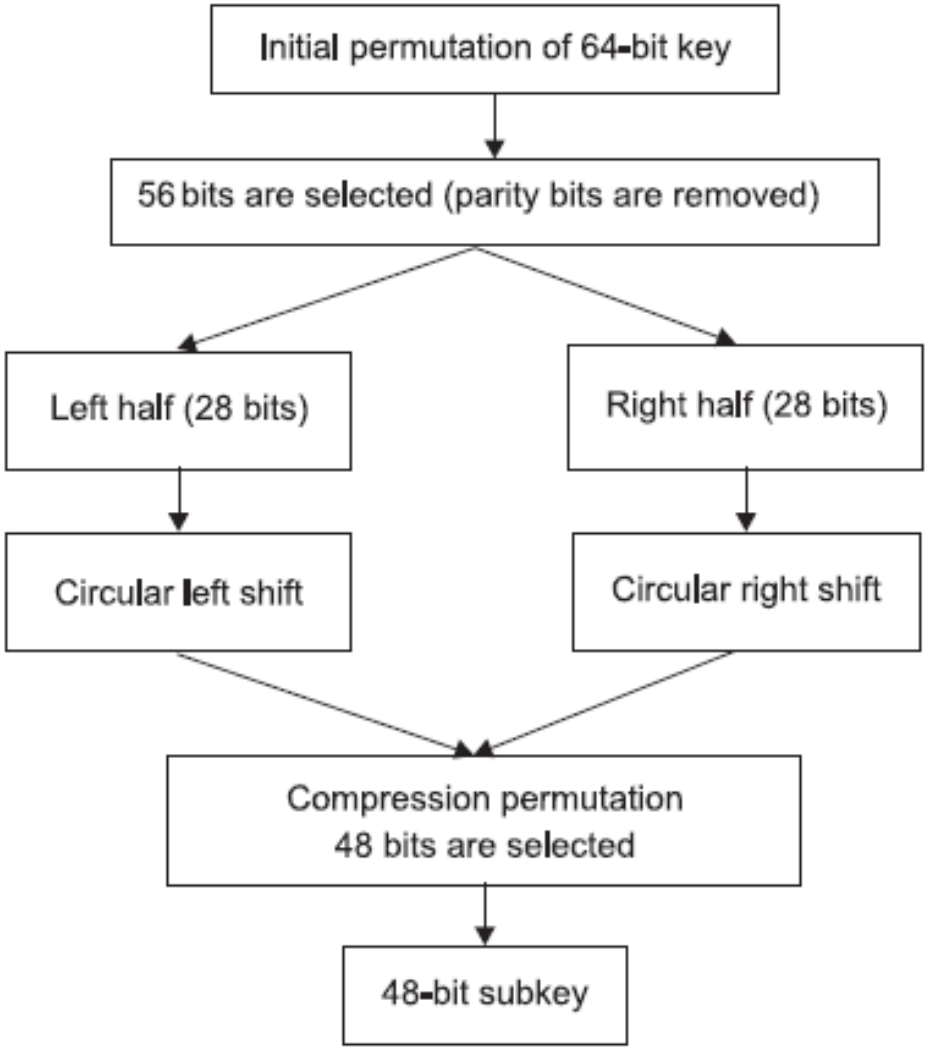


Figure 3.13 Subkey generation for DES.

Sub-Key Generation

1. 64 bit key (including parity bits) is permuted & reduced to 56 bits

Note that the parity bits are not selected.

8, 16, 24, 32, 40, 48, 56, 64 are the parity bits

All other bits are selected and permuted.

Input Key [64 bits]

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	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64

Permutation – 1 [56 bits]

From the original 64-bit key we get the 56-bit permutation.

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

2. The 56 bit result is split in two halves (left and right).

Split this key into left and right halves, where each half has 28 bits

3. The halves are circular shifted by 1 or 2 bits (round dependent)

Round	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Bits rotated	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

do a left shift and right shift for each left and right half

Remember, we are generating 16 sub-keys, one for each round and each step after 3 above is performed on each sub-key at each round

4. After the shift, 48 of the 56 bits are selected using a compression permutation (56 bit to 48 bit compression)

Compression Permutation – 2 [48 bits]

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

Subkey generation steps

- Apply initial permutation on 64-bit key
- Remove the parity bits from 64 bits
- Apply permutation on the 56-bit key.
- Split the 56-bit key into two halves 28 bits each.
- Apply circular left shift on each half.
- Concatenate the two halves.
- Apply compression permutation

INITIAL PERMUTATION [64 – BITS]

- The message is converted into binary form.
- Then each bit allocates its position starting from right (0^{th} position) to left (64^{th} position).
- Initial permutation is applied on 64-bit block of plaintext using permutation table.
- The first bit is the 58^{th} bit from the plaintext block; second bit is the 50^{th} bit and so on.
- This table performs the permutations of the plaintext bits.

Initial Permutation [64 – bits]

- Initially apply permutation on 64 bit plaintext using IP table

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

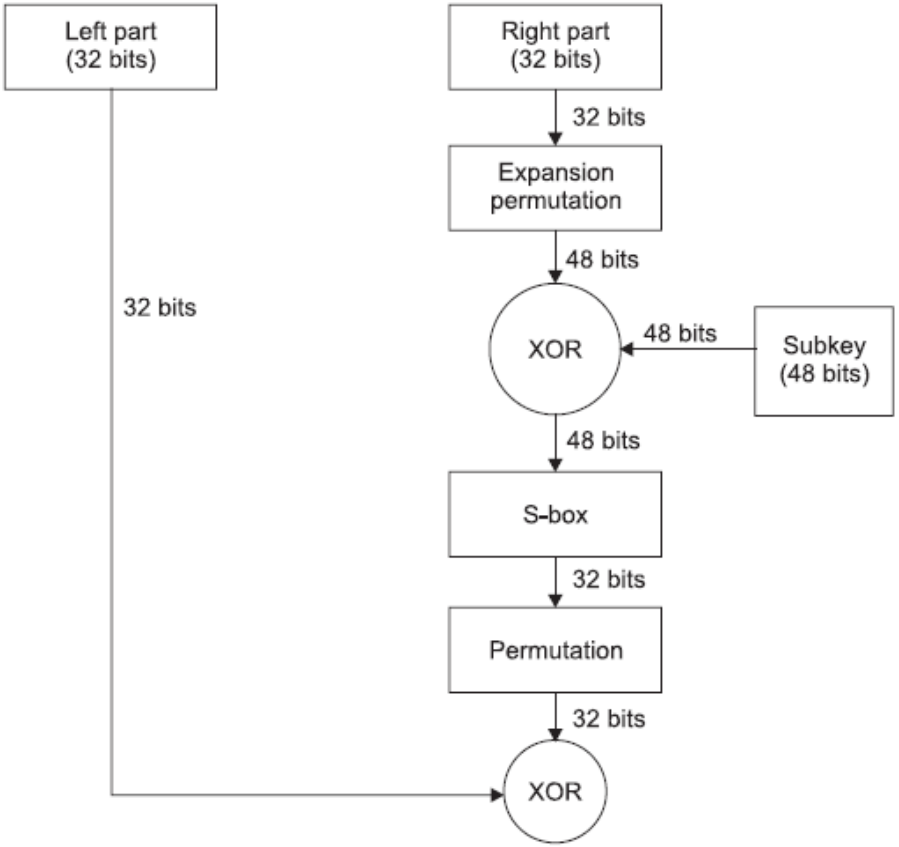
- Then this 64-bit plaintext block undergoes different operations from round 1 to 10

Round 1

Following steps are performed in each round of the DES

1. Expansion Permutation
2. XOR Operation
3. S-Box Substitution
4. P-Box Permutation
5. XOR and
6. Swapping

Single Round of DES



DES - Discussion

The permuted output is split into a 32 bit left and right half

1. Expansion Permutation

- The right part having 32 bits undergoes expansion permutation using expansion permutation table.
- This step expands the right half part to 48 bits. Therefore, some of the bits are repeated.

Expansion Permutation [48 – bits]

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

- The new bit position is as per the permutation table, i.e., first bit is the 32nd bit, and second bit is the 1st bit and so on.

2. XOR Operation

- In this step XOR operation is performed between the 48 bits of right half and the subkey. This creates key dependency
- Output is 48 bits

3. S-Box Operation

- The 48 bits of right half are divided into 8 groups.
- Each group has 6 bits.
- There are eight S-boxes, one for each group

3. S-Box Operation

- 48 bit plaintext is split into eight groups of 6 bits each.
- Then select MSB and LSB, i.e., the first and the last bit of a group.
- These two bits form a 2-bit binary number.
- The decimal equivalent of this number is a row number from the S-box.

- For example, suppose a group is **1**0000**0**
- Then MSB is **1** and LSB is **0**.
- The pair is **(10)$_2$** .
- The decimal equivalent of $(10)_2$ is $(02)_{10}$.
- So row 2 is selected.

3. S-Box Operation

- Then select the middle four bits of the same group. These four bits form a 4-bit binary number.
- The equivalent decimal number is a column number from the S-box.

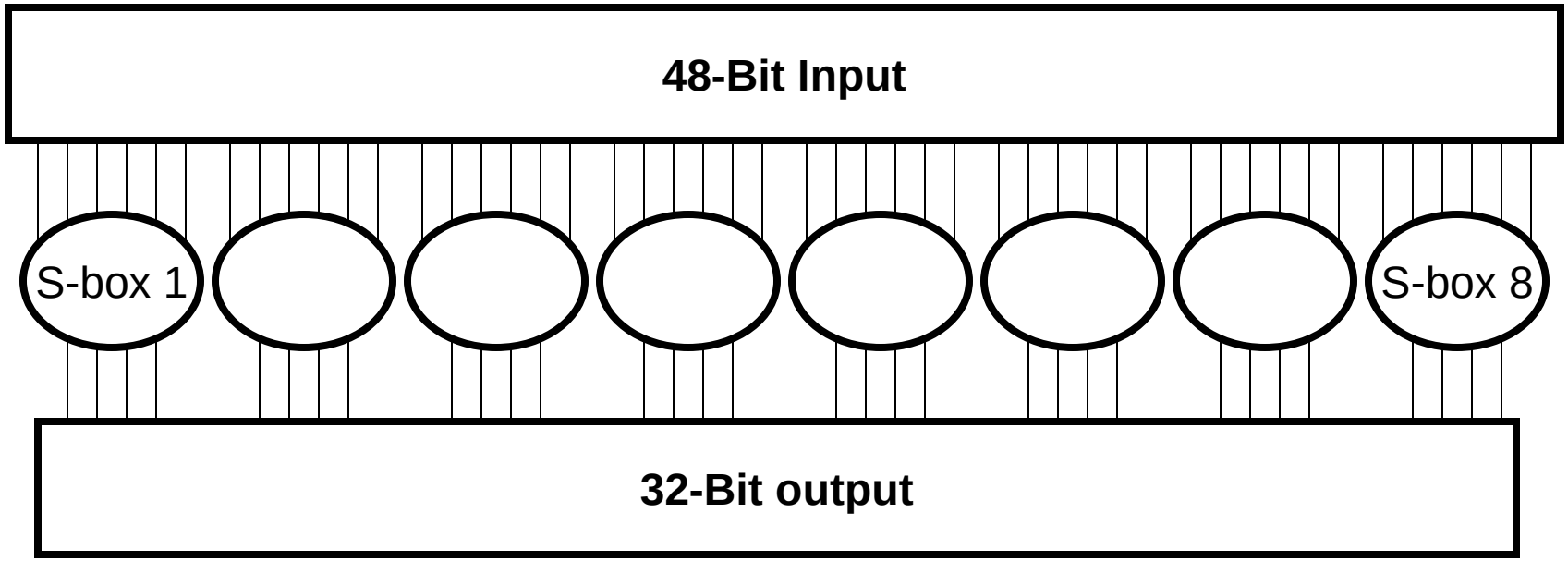
- For example, suppose a group is 100000, then middle four bits are **0000**. Then the binary number is $(0000)_2$ The decimal equivalent of $(0000)_2$ is $(00)_{10}$.
- So column 0 is selected.

S-Box

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

If we number the inputs: a_1, a_2, a_3, a_4, a_5, a_6 then a_1 and a_6 specify the row and a_2,a_3, a_4, a_5 specify the column

DES - Encryption - S-boxes



- Each S box provides 4 bit output.
- There are total 8 S boxes one for each group
- In total, the S-Boxes provide 4-bits/box x 8 boxes = 32-bit output.
- So Input to this step is 48 bits and output is 32 bits
- The S-Boxes provide the confusion function.

4. P-Box Permutation

- The 32 bit output of the S-box is sent to the P-Box for permutation
- To provide additional diffusion. In the P-Box, the 32-bit input is mapped to a 32-bit output - every bit is used, and no bit is used twice
- This is a total of 32 bits

5. XOR

- The output of the P-Box is then XORed with the left half output of the original permutation
- Output of XOR is 32 bits

6. Swapping

- The result of XOR is used as a new right half for the next round and the old right half part is used as a new left half for the next round.
- The 64 bit output is now used as input for round 2.
- This continues for 16 total rounds with each round using a different sub-key.

Decryption

- DES uses exactly the same functions to decrypt as encrypt, but the sub-keys must be used in reverse order for decryption (i.e., encryption uses K_1, K_2, ….K_{15}, so decryption uses K_{16}, K_{15}, ….K_1).

Weak Keys in Symmetric Algorithms

- In any encryption algorithm we can find weak keys
- These are keys that yield encryptions that are relatively easy to break using cryptanalysis
- Simple weak keys are those with a high degree of internal redundancy or symmetry

- For example, any key composed of:
  - all zeros,
  - all ones,
  - alternating ones and zeros, and
  - alternating zeroes and ones.

Weak Keys in DES

There are 16 DES keys that are considered weak. The probability of generating these is small ($16/2^{56}$).

Suspect keys are those for which the initial permutation of the 2-28 bit quantities C_0, D_0 produce one of the four values:

All zeroes, alternating ones and zeroes, alternating zeroes and ones, and all ones. This produces 16 weak keys as follows:

	C_0	C_0	C_0	C_0
	0,0..	1,0..	0,1..	1,1..
D_0	0,0..	0,0..	0,0..	0,0..
D_0	1,0..	1,0..	1,0..	1,0..
D_0	0,1..	0,1..	0,1..	0,1..
D_0	1,1..	1,1..	1,1..	1,1..

- The 4 keys for which C_0 and D_0 are all zeroes or ones are weak keys since they are their own inverses. This means that the key used in every round will be the same (shift produces no change). This is also true of keys where each half is all 1's or 0's.
- The remaining 12 keys are considered semi-weak since each is the inverse of one of the others.
- Relatively easy to test for these keys after generation.
- Also a good idea to avoid keys with values < 1000, since an adversary may start searching the key space from the bottom.
- This can be also be avoided by a simple test.

Lab Assignment 4

Implement Simplified DES algorithm for encryption and decryption of any message.

Count the time required for encryption/decryption of messages of different length.

Note: You have to use assignments 3 for various operations in S-DES

Last date for submission: 25 Sept 2020

Questions?