USN										

CMRIT CMR INSTITUTE OF TECHNOLOGY, BENGALURU. ACCREDITED WITH A+ GRADE BY NAAC

INTERNAL ASSESSMENT TEST – II

Sub:	CRYPTOGRAPHY							Code:	18EC744
Date:	2/ 12 / 2022	Duration:	90 mins	Max Marks:	50	Sem:	VII	Branch:	ECE

Answer any 5 full questions

		Marks	**CO**	**RBT**
1	Explain in detail Feistel cipher. Demonstrate with a neat diagram Feistel Encryption and Decryption.	[10]	CO1	L1
2	For the group G = ⟨Z11 +,×⟩ find the additive and multiplicative inverses. Prepare a table of discrete logarithms for a base value of 2.	[10]	CO1	L3
3	With a block diagram, explain DES encryption and decryption algorithm.	[10]	CO1	L3
4.	Using the properties of discrete logarithms, show how to solve the following congruences: a. $3\cdot5^x \equiv 6(\text{mod}23)$ b. $3\cdot5^x \equiv 9(\text{mod}23)$ c. $2\cdot5^x \equiv 6(\text{mod}23)$	[10]	CO3	L3
5.	Explain the AES encryption and decryption with a neat block diagram.	[10]	CO2	L1
6.	Find the solutions to each of the following linear equations a. $3x \equiv 4(\text{mod}5)$ b. $8x \equiv 6(\text{mod}5)$ c. $2x \equiv 7(\text{mod}23)$	[10]	CO4	L4
7.	State and prove (i) Fermat's theorem (ii) Euler's theorem. Give one example for the above theorems.	[10]	CO4	L1

IAT-2 Scheme of solutions

Q. no.	**Questions**	**Marks**
1.	Explain in detail Feistel cipher. Demonstrate with a neat diagram Feistel Encryption and Decryption Fiestel structure. Encryption and decryption. **FEISTEL CIPHER STRUCTURE:** 1. The inputs to the encryption algorithm are a plaintext block of length 2w bits and a key K. 2. The plaintext block is divided into two halves, $L0$ and $R0$. 3. The two halves of the data pass through n rounds of processing and then combine to produce the ciphertext block. 4. Each round i has as inputs Li-1 and Ri-1 derived from the previous round, as well as a subkey Ki derived from the overall K. The subkeys Ki are different from K and from each other.	10M

5. 16 rounds are used, although any number of rounds could be implemented. All rounds have the same structure.

6. A **substitution** is performed on the left half of the data. This is done by applying a *round function* F to the right half of the data and then taking the exclusive-OR of the output of that function and the left half of the data.

7. The round function F has the same general structure for each round. The round function F is represented as $(REi, Ki+1)$

8. Following this substitution, a **permutation** is performed that consists of the interchange of the two halves of the data.

9. Feistel network depends on the choice of the following parameters and design features:

a) **Block size:** larger block sizes mean greater security, but it reduces encryption/decryption speed for a given algorithm. The greater security is achieved by greater diffusion. Traditionally, a block size of 64 bits has been considered a reasonable tradeoff and was nearly universal in block cipher design. However, the new AES uses a 128-bit block size.

b) **Key size:** Larger key size means greater security but may decrease encryption decryption speed. The greater security is achieved by greater resistance to brute-force attacks and greater confusion. Key sizes of 64 bits or less are now widely considered being inadequate and 128 bits has become a common size.

c) **Number of rounds:** The essence of the Feistel cipher is that a single round offers inadequate security but that multiple rounds offer increasing security. A typical size is 16 rounds.

d) **Subkey generation algorithm:** Greater complexity in this algorithm should lead to greater difficulty of cryptanalysis.

e) **Round function F:** Again, greater complexity generally means greater resistance to cryptanalysis.

10. There are two other considerations in the design of a Feistel cipher:

a) **Fast software encryption/decryption:** Encryption is embedded in applications hence the speed of execution of the algorithm becomes a concern.

b) **Ease of analysis:** Although we would like to make our algorithm as difficult as possible to cryptanalyze, there is great benefit in making the algorithm easy to analyze. That is, if the algorithm can be concisely and clearly explained, it is easier to analyze that algorithm for cryptanalytic vulnerabilities and therefore develop a higher level of assurance as to its strength. DES, for example, does not have an easily analyzed
functionality.

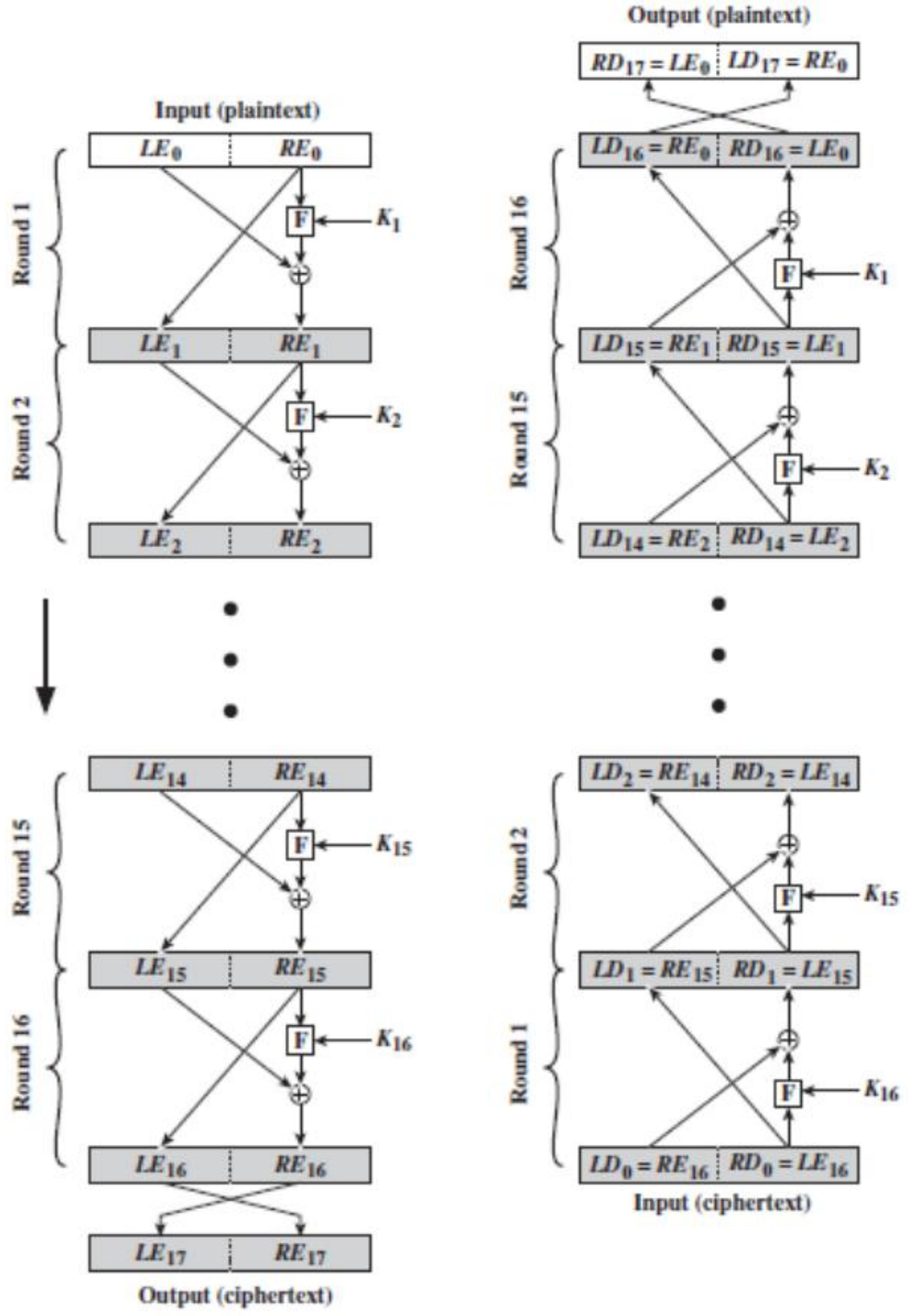


Figure: Feistel Encryption and Decryption (16 rounds)

11. **Feistel Decryption Algorithm:**

a) Decryption with a Feistel cipher is same as the encryption process.

b) In decryption the ciphertext is used as input to the algorithm, and the subkeys Ki are used in reverse order.

c) That is, Kn is used in the first round, Kn-1 in the second round, and so on, until K is used in the last round. It is an advantage because no need to implement two different algorithms; one for encryption and one for decryption.

d) For clarity, the notation $LEi\ and\ REi$ is used for data traveling through the encryption algorithm and $LDi\ and\ RDi$ for data traveling through the decryption algorithm.

e) The diagram indicates that, at every round, the intermediate value of the decryption process is equal to the corresponding value of the encryption process with the two halves of the value swapped. i.e. $RE_i \| LE_i = LD_{16-i} \| RD_{16-i}$

f) Example: (for better clarity)

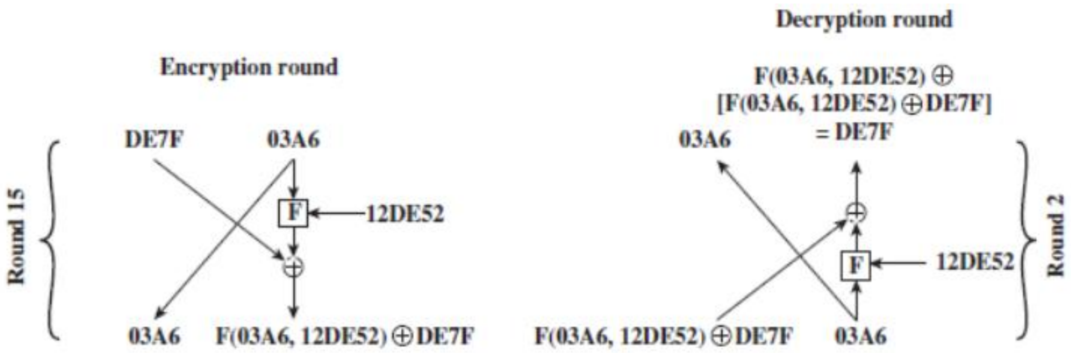


Figure: Feistel Example

2. For the group $G = \langle Z11 +, \times \rangle$ find the additive and multiplicative inverses. Prepare a table of discrete logarithms for base value of 2. 10M

Ans: For a given No, $p=11$, we define the finite field of order 11, GF(11), as the set Z_{11} of integers {0,1,2,3,4,5,6,7,8,9,10}

+	0	1	2	3	4	5	6	7	8	9	10
0	0	1	2	3	4	5	6	7	8	9	10
1	1	2	3	4	5	6	7	8	9	10	0
2	2	3	4	5	6	7	8	9	10	0	1
3	3	4	5	6	7	8	9	10	0	1	2
4	4	5	6	7	8	9	10	0	1	2	3
5	5	6	7	8	9	10	0	1	2	3	4
6	6	7	8	9	10	0	1	2	3	4	5
7	7	8	9	10	0	1	2	3	4	5	6
8	8	9	10	0	1	2	3	4	5	6	7
9	9	10	0	1	2	3	4	5	6	7	8
10	10	0	1	2	3	4	5	6	7	8	9

Additive & multiplicative inverses

×	0	1	2	3	4	5	6	7	8	9	10
0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7	8	9	10
2	0	2	4	6	8	10	1	3	5	7	9
3	0	3	6	9	1	4	7	10	2	5	8
4	0	4	8	1	5	9	2	6	10	3	7
5	0	5	10	4	9	3	8	2	7	1	6
6	0	6	1	7	2	8	3	9	4	10	5
7	0	7	3	10	6	2	9	5	1	8	4
8	0	8	5	2	10	7	4	1	9	6	3
9	0	9	7	5	3	1	10	8	6	4	2
10	0	10	9	8	7	6	5	4	3	2	1

w	-w	w^{-1}
0	0	-
1	10	1
2	9	6
3	8	4
4	7	3
5	6	9
6	5	2
7	4	8
8	3	7
9	2	5
10	1	10

a	1	2	3	4	5	6	7	8	9	10
b=$\text{dlog}_{2,11}$(a) a ≡2^{b}(mod11)	10	1	8	2	4	9	7	3	6	5

3. Block Diagram of DES encryption and decryption. 10M

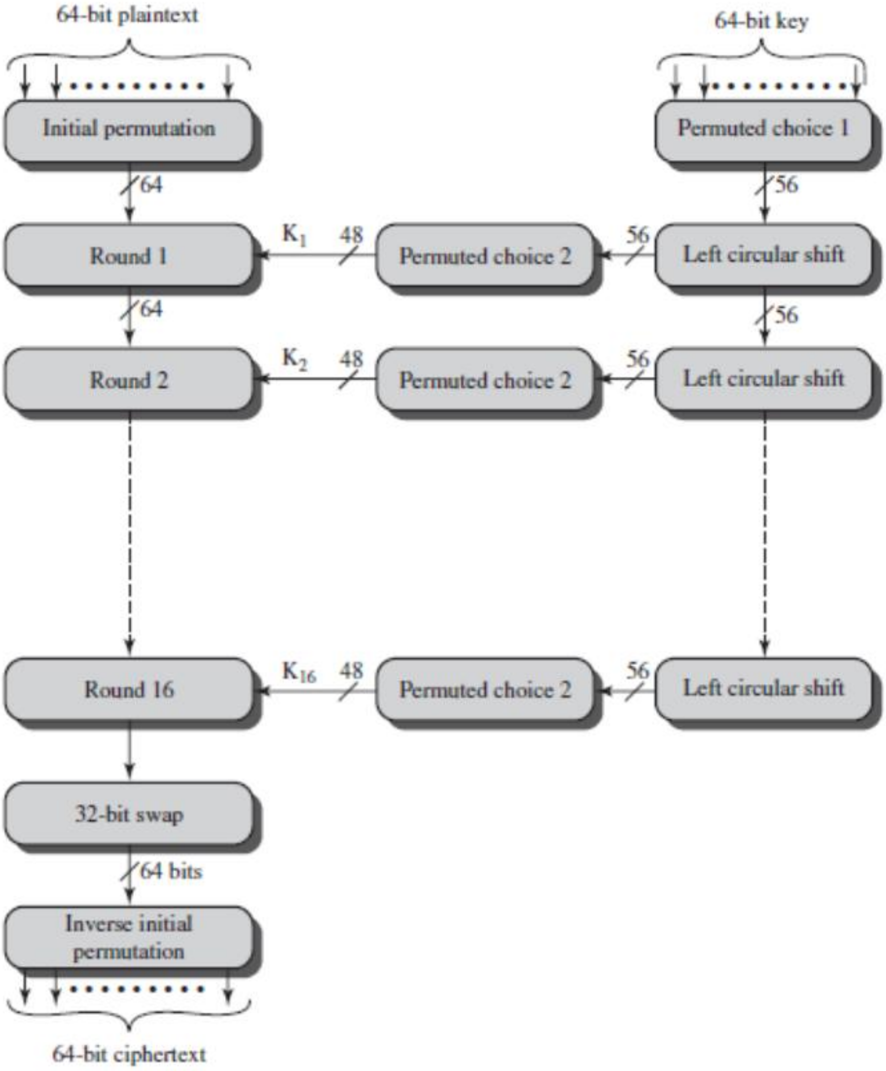


Figure: General Depiction of DES Encryption Algorithm

64 bit key is used but every 8th bit is the parity bit hence it is taken as 56 bit key. Initially the key is passed through the permutation function. For each 16 round, a sub key Ki is produced by the combination of left circular shift and permutation. The same permutation function is used in eachround.

The plain text are processed through these phases

a) Initial Permutation

b) 16 rounds of same function

c) Swap

d) Final Permutation

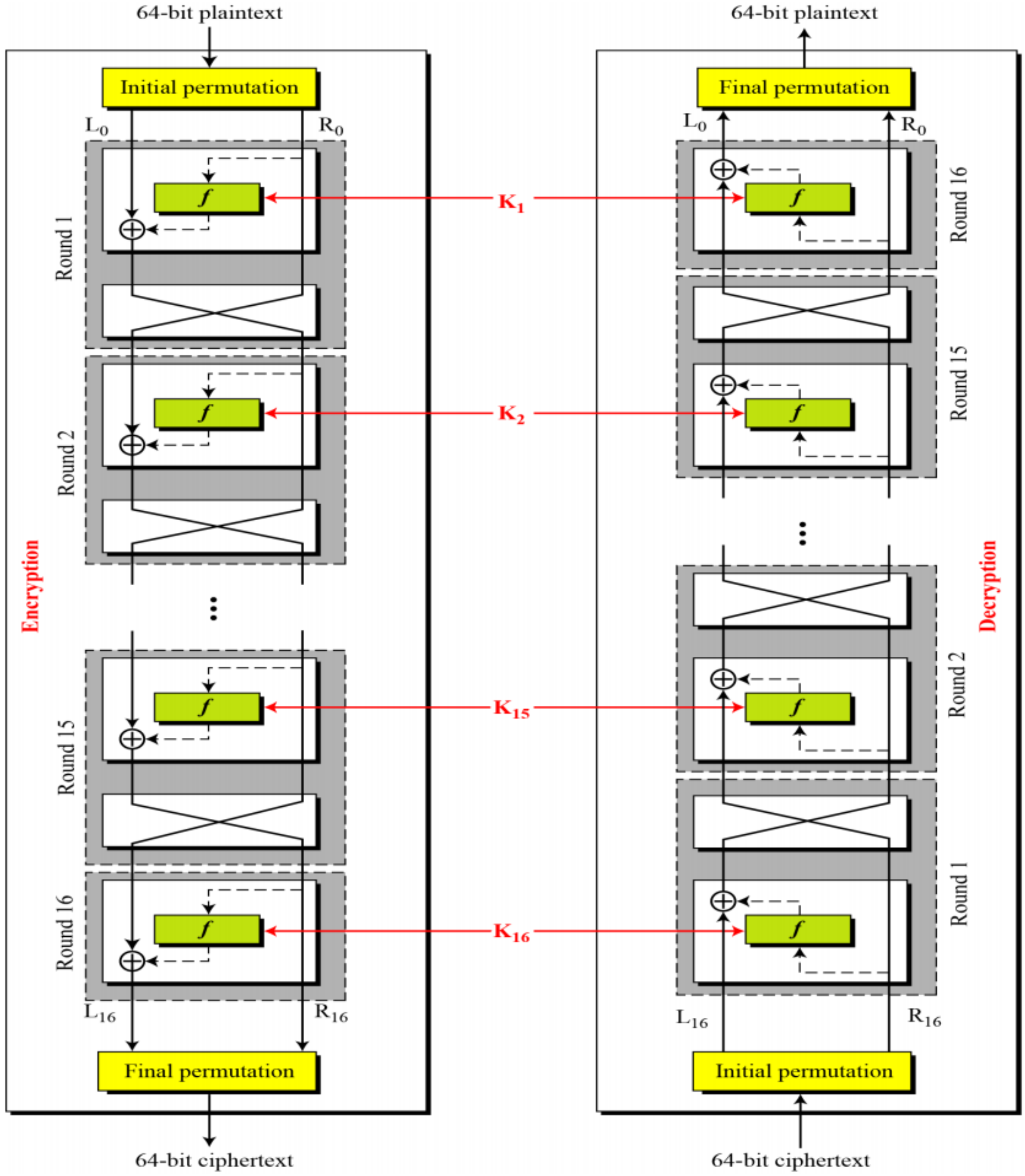
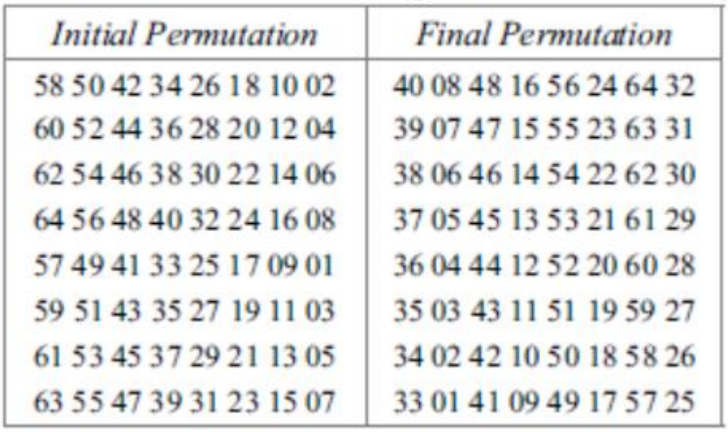


Figure: DES Encryption and Decryption

Initial Permutation and Final Permutation:

The input is 64 bit. These inputs are permuted according to a predefined rule. The permutation table contains a permutation of the number from 1 to 64. These permutation table and inverse permutation table can be designed such that the original bits can be restored.



Initial Permutation	*Final Permutation*
58 50 42 34 26 18 10 02	40 08 48 16 56 24 64 32
60 52 44 36 28 20 12 04	39 07 47 15 55 23 63 31
62 54 46 38 30 22 14 06	38 06 46 14 54 22 62 30
64 56 48 40 32 24 16 08	37 05 45 13 53 21 61 29
57 49 41 33 25 17 09 01	36 04 44 12 52 20 60 28
59 51 43 35 27 19 11 03	35 03 43 11 51 19 59 27
61 53 45 37 29 21 13 05	34 02 42 10 50 18 58 26
63 55 47 39 31 23 15 07	33 01 41 09 49 17 57 25

DES Encryption:

a) In DES Encryption, there are two inputs to the encryption function:

i. the plaintext to be encrypted

ii. Key

b) In this case, the plaintext must be 64 bits in length and the key is 56 bits in length.

c) The processing of the plaintext proceeds in three phases.

i. First, the 64-bit plaintext passes through an initial permutation (IP) that rearranges the bits to produce the permuted input.

ii. This is followed by a phase consisting of sixteen rounds of the same function, which involves both permutation and substitution functions.

iii. The left and right halves of the output are swapped to produce the preoutput.

iv. Finally, the pre-output is passed through a permutation [IP−1] that is the inverse of the initial permutation function, to produce the 64-bit ciphertext.

d) With the exception of the initial and final permutations, DES has the exact structure of a Feistel cipher.

Key Generation:

a) In DES, 56-bit key is used.

b) Initially, the key is passed through a permutation function.

a) Then, for each of the sixteen rounds, a *subkey* (Ki) is produced by the combination of a left circular shift and a permutation.

b) The permutation function is the same for each round, but a different subkey is produced because of the repeated shifts of the key bits.

DES Decryption:

a) As with any Feistel cipher, decryption uses the same algorithm as encryption, except that the application of the subkeys is reversed.

b) Additionally, the initial and final permutations are reversed.

4. Use of discrete logarithms to solve the congruence: 10M

a	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
$\log_{5,23}$ (a)	22	2	16	4	1	18	19	6	10	3	9	20	14	21	17	8	7	12	15	5	13	11

(a) $\mathbf{3 \cdot 5^x \equiv 6(mod23)}$

Multiply both sides by 8

$5^x \equiv 48(mod23) \equiv 2\ mod\ 23$ By above lookup table *x*=2.

(b) $\mathbf{3 \cdot 5^x \equiv 9(mod23)}$

Multiply both sides by 8

$5^x \equiv 72(mod23) \equiv 3\ mod\ 23$ By above lookup table *x*=16.

(c) $\mathbf{2 \cdot 5^x \equiv 6(mod23)}$

Multiply both sides by 12

$5^x \equiv 72(mod23) \equiv 3\ mod\ 23$ By above lookup table *x*=16.

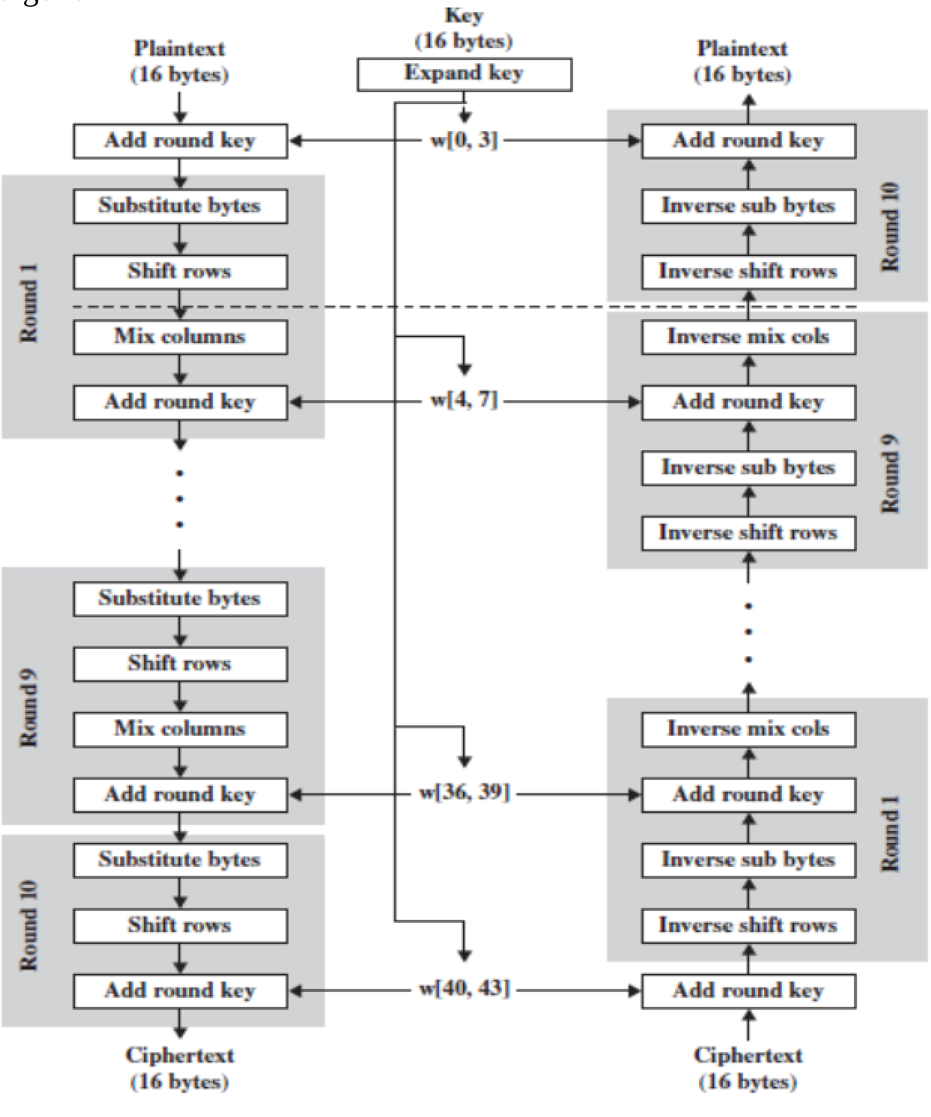
5. AES encryption and decryption with block diagram 10M

AES doesn’t use the Feistel structure. Feistel structure, half of the data block is used to modify the other half of the data block and then the halves are swapped. AES instead processes the entire data block as a single matrix during each round using substitutions and permutation. The key that is provided as input is expanded into an array of forty-four 32-bit words, w[i]. Four different stages are used, one of permutation and three of substitution:

a) Substitute bytes: Uses an S-box to perform a byte-by-byte substitution of the block

b) ShiftRows: A simple permutation
c) MixColumns: A substitution that makes use of arithmetic over GF(28)
d) AddRoundKey: A simple bitwise XOR of the current block with a portion of the expanded key

The cipher begins with an AddRoundKey stage, followed by nine rounds that each includes all four stages, followed by a tenth round of three stages. AddRoundKey stage makes use of the key.The cipher begins and ends with an AddRoundKey stage. Each stage is easily reversible. For the Substitute Byte, ShiftRows, and MixColumns stages, an inverse function is used in the decryption algorithm. For the AddRoundKey stage, the inverse is achieved by XORing the same round key to the block, using the result that $A \oplus B \oplus B = A$. In AES, the decryption algorithm is not identical to the encryption algorithm.



(a) Encryption (b) Decryption

Figure: AES Encryption and Decryption

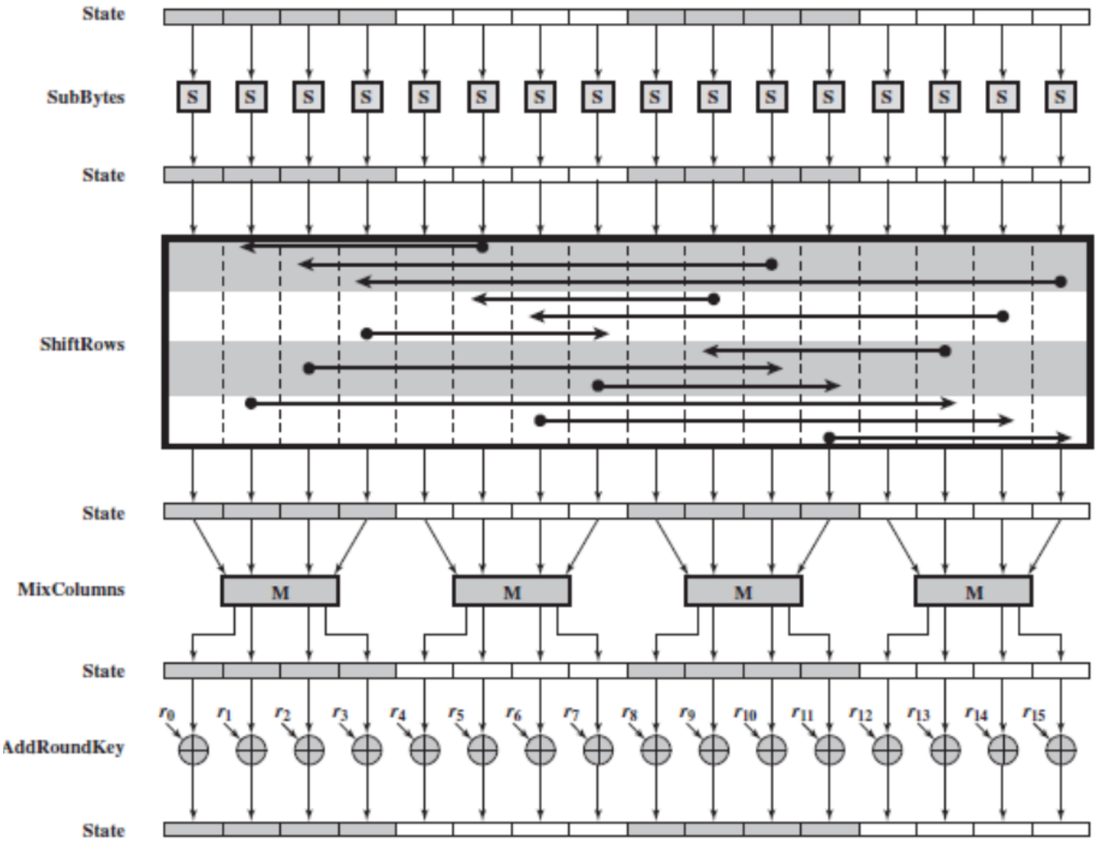

State
SubBytes
S
State
ShiftRows
State
MixColumns
M
State
AddRoundKey
r_0 r_1 r_2 r_3 r_4 r_5 r_6 r_7 r_8 r_9 r_{10} r_{11} r_{12} r_{13} r_{14} r_{15}
State

Figure: AES Encryption Round

Figure: AES Encryption Round

As all stages are reversible, it is easy to perform decryption to recover the plain text. Encryption and decryption going in opposite vertical directions. The first N - 1 rounds consist of four distinct transformation functions:

- Sub Bytes,
- Shift Rows,
- Mix Columns,
- Add RoundKey

The final round contains only three transformations those are SubBytes, ShiftRows and AddRoundKey, and there is an initial single transformation (AddRoundKey) before the first round, which can be considered Round 0.

6. Find the solutions to each of the following linear equations — 10M

a.$\boldsymbol{3x \equiv 4(mod5)}$

7 is the multiplicative inverse of 3 mod 5 so multiplying both sides by 7

$\therefore x \equiv 28(mod5) \equiv 3(mod5)$

b. $\boldsymbol{8x \equiv 6(mod5)}$

2 is the multiplicative inverse of 8 mod 5 so multiplying both sides by 2

$\therefore x \equiv 12(mod5) \equiv 2(mod5)$

c.$\boldsymbol{2x \equiv 7(mod23)}$

12 is the multiplicative inverse of 2 mod 23 so multiplying both sides by 12

$\therefore x \equiv 84(mod23) \equiv 15(mod\ 23)$

7. State and prove (i) Fermat's theorem (ii) Euler's theorem. — 10M

Ans: Fermat's Theorem: $a^{p-1} \equiv 1$ (mod p)

- Consider the set of positive integers less than *p*: {1, 2, ………, *p* - 1}
- Let us multiply each element by *a* (mod *p)*

- Get the set X = {a (mod p), 2a (mod p), …….,(p - 1)a (mod p)}
- None of the elements of X is equal to zero because p does not divide a. Furthermore, the (p - 1) elements of X are all positive integers with no two elements equal.
- We can conclude the X consists of the set of integers {1, 2, ………, p - 1} in some order
- $a \times 2a \times \ldots\ldots \times (p - 1)a \equiv [(1 \times 2 \times \ldots\ldots \times (p - 1)](\text{mod } p)$
- $a^{p-1}(p - 1)! \equiv (p - 1)!\ (\text{mod } p)$
- $a^{p-1} \equiv 1(\text{mod } p)$, ∵(p - 1)! is relatively prime to p

Euler's theorem: $a^{\phi(n)} \equiv 1\ (\text{mod } n)$

Proof if n is prime, because in that case, $\phi(n) = (n - 1)$ and Fermat's theorem holds. However, it also holds for any integer n. Recall that f (n) is the number of positive integers less than n that are relatively prime to n.

Consider the set of such integers, labeled as $R = \{x_1, x_2, \ldots, x_{\phi(n)}\}$

That is, each element x_i of R is a unique positive integer less than n with $\gcd(x_i, n) = 1$.

Now multiply each element by a, modulo n:

$S = \{(ax_1 \text{ mod } n), (ax_2 \text{ mod } n), \ldots\ldots, (ax_{\phi(n)} \text{ mod } n)\}$

Because a is relatively prime to n and xi is relatively prime to n, ax_i must also be relatively prime to n. Thus, all the members of S are integers that are less than n and that are relatively prime to n.

2. There are no duplicates in S. If ax_i mod n = ax_jmod n, then $x_i = x_j$. Therefore,

$$\prod_{i=1}^{\emptyset(n)}(ax_i \text{ mod } n) = \prod_{i=1}^{\emptyset(n)} x_i \quad \prod_{i=1}^{\emptyset(n)} ax_i \equiv \prod_{i=1}^{\emptyset(n)}(x_i mod\ n)$$

$$a^{\emptyset(n)} \prod_{i=1}^{\emptyset(n)} x_i \equiv \prod_{i=1}^{\emptyset(n)}(x_i mod\ n) \quad \boxed{\therefore a^{\emptyset(n)} \equiv 1 mod\ n}$$

e. g, $a = 7$, $p = 19$

- $7^2 = 49 \equiv 11$ (mod 19)
- $7^4 \equiv 121 \equiv 7$ (mod 19)
- $7^8 \equiv 49 \equiv 11$ (mod 19)
- $7^{16} \equiv 121 \equiv 7$ (mod 19)
- **$a^{p-1} = 7^{18} = 7^{16} \times 7^2 \equiv 7 \times 11 \equiv 1$ (mod 19)**