CNS QP CAT 1 ANSWERS

1. B) Explain in detail about various transposition ciphers used in cryptography.

Ans.1.B)

Transposition Techniques are based on the permutation of the plain-text instead of substitution.

1) Rail-Fence Technique

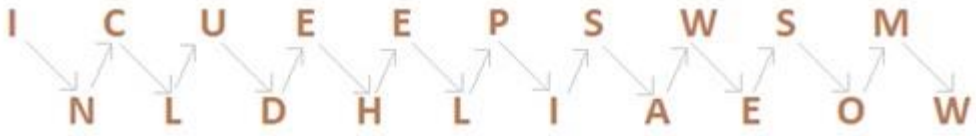
This technique is a type of Transposition technique and does is write the plain text as a sequence of diagonals and changing the order according to each row.

It uses a simple algorithm,

1. Writing down the plaintext message into a sequence of diagonals.
2. Row-wise writing the plain-text written from above step.

Example,

Let's say, we take an example of “INCLUDEHELP IS AWESOME”.



So the Cipher-text are, **ICUEEPSWSMNLDHLIAEOW**.

First, we write the message in a zigzag manner then read it out direct row-wise to change it to cipher-text.

Now as we can see, Rail-Fence Technique is very to break by any cryptanalyst.

2) Columnar Transition Technique

A. Basic Technique

It is a slight variation to the Rail-fence technique, let's see its algorithm:

1. In a rectangle of pre-defined size, write the plain-text message row by row.
2. Read the plain message in random order in a column-wise fashion. It can be any order such as 2, 1, 3 etc.
3. Thus Cipher-text is obtained.

B. Columnar Technique with multiple rounds

In this method, we again change the chipper text we received from a Basic technique that is in round 1 and again follows the same procedure for the cipher-text from round 1.

Algorithm:

1. In a rectangle of pre-defined size, write the plain-text message row by row.
2. Read the plain message in random order in a column-wise fashion. It can be any order such as 2, 1, 3 etc.
3. Thus, Cipher-text of round 1 is obtained.
4. Repeat from step 1 to 3.

These multi-round columnar techniques are harder to crack as compared to methods seen earlier.

1. C) Demonstrate the working of encryption and decryption procedure in Hill Cipher with respect to following parameters:

Plain Text : ACOLLEGE

Key :

7	8
19	3

Ans.1.C)

2. B) Explain in detail about Playfair Cipher and then apply it to encrypt with respect to:

Plain Text : CHANDRAYAAN

Key:

T	**M**	**P**	**O**	**S**
Z	**V**	**W**	**X**	**Y**
E	**Q**	**C**	**U**	**R**
F	**N**	**A**	**B**	**D**
L	**G**	**H**	**I/J**	**K**

Ans.2.B)

The Playfair cipher was the first practical digraph substitution cipher. The scheme was invented in 1854 by Charles Wheatstone but was named after Lord Playfair who promoted the use of the cipher. In playfair cipher unlike traditional cipher we encrypt a pair of alphabets(digraphs) instead of a single alphabet.

It was used for tactical purposes by British forces in the Second Boer War and in World War I and for the same purpose by the Australians during World War II. This was because Playfair is reasonably fast to use and requires no special equipment.

The Playfair Cipher Encryption Algorithm:

The Algorithm consists of 2 steps:

1. **Generate the key Square(5×5):**
   - The key square is a 5×5 grid of alphabets that acts as the key for encrypting the plaintext. Each of the 25 alphabets must be unique and one letter of the alphabet (usually J) is omitted from the table (as the table can hold only 25 alphabets). If the plaintext contains J, then it is replaced by I.
   - The initial alphabets in the key square are the unique alphabets of the key in the order in which they appear followed by the remaining letters of the alphabet in order.
2. **Algorithm to encrypt the plain text:** The plaintext is split into pairs of two letters (digraphs). If there is an odd number of letters, a Z is added to the last letter.

Plain Text : CHANDRAYAAN

Key:

T	M	P	O	S
Z	V	W	X	Y
E	Q	C	U	R
F	N	A	B	D
L	G	H	I/J	K

CH	AN	DR	AY	AX	AN
AP	BA	KD	DW	BW	BA

Cipher Text : APBAKDDWBWB

2. **C) Apply Extended Euclid algorithm to compute GCO (99,78).**

   Show all the computations.

Ans.2.C)

GCD (99 , 78) using Extended Euclid's Algorithm

Q	A	B	R
1	99	78	21
3	78	21	15
1	21	15	6
2	15	6	3
1	6	3	0
	3	0	

GCD (99 , 78) = 3

Computations

- a = b * q + r
- 99 = 78 * 1 +21
- 78 = 21 * 3 +15
- 21 = 15 * 1 +6
- 15 = 6 * 2 +3
- 6 = 3 * 2 +0

3. B) Differentiate stream ciphers and block ciphers.

Ans.3.B)

S.NO	Block Cipher	Stream Cipher
1.	Block Cipher Converts the plain text into cipher text by taking plain text's block at a time.	Stream Cipher Converts the plain text into cipher text by taking 1 byte of plain text at a time.
2.	Block cipher uses either 64 bits or more than 64 bits.	While stream cipher uses 8 bits.
3.	The complexity of block cipher is simple.	While stream cipher is more complex.
4.	Block cipher Uses confusion as well as diffusion.	While stream cipher uses only confusion.
5.	In block cipher, reverse encrypted text is hard.	While in-stream cipher, reverse encrypted text is easy.
6.	The algorithm modes which are used in block cipher are ECB (Electronic Code Book) and CBC (Cipher Block Chaining).	The algorithm modes which are used in stream cipher are CFB (Cipher Feedback) and OFB (Output Feedback).
7.	Block cipher works on transposition techniques like rail-fence technique, columnar transposition technique, etc.	While stream cipher works on substitution techniques like Caesar cipher, polygram substitution cipher, etc.
8.	Block cipher is slow as compared to a stream cipher.	While stream cipher is fast in comparison to block cipher.
9.	Suitable for applications that require strong encryption, such as file storage and internet communications	Suitable for applications that require strong encryption, such as file storage and internet communications
10.	More secure than stream ciphers when the same key is used multiple times	Less secure than block ciphers when the same key is used multiple times
11.	key length is Typically 128 or 256 bits	key length is Typically 128 or 256 bits
12.	Operates on fixed-length blocks of data	Encrypts data one bit or byte at a time

3. C) What are the block cipher modes of operation of DES? Explain in detail.

Ans.3.C)

Encryption algorithms are divided into two categories based on the input type, as a block cipher and stream cipher.

Block cipher is an encryption algorithm that takes a fixed size of input say b bits and produces a ciphertext of b bits again. If the input is larger than b bits it can be divided further.

For different applications and uses, there are several modes of operations for a block cipher.

i. Electronic Code Book (ECB) –

Electronic code book is the easiest block cipher mode of functioning. It is easier because of direct encryption of each block of input plaintext and output is in form of blocks of encrypted ciphertext. Generally, if a message is larger than b bits in size, it can be broken down into a bunch of blocks and the procedure is repeated.

ii. Cipher Block Chaining –

Cipher block chaining or CBC is an advancement made on ECB since ECB compromises some security requirements. In CBC, the previous cipher block is given as input to the next encryption algorithm after XOR with the original plaintext block. In a nutshell here, a cipher block is produced by encrypting an XOR output of the previous cipher block and present plaintext block.

iii. Cipher Feedback Mode (CFB) –

In this mode the cipher is given as feedback to the next block of encryption with some new specifications: first, an initial vector IV is used for first encryption and output bits are divided as a set of *s* and *b-s* bits. The left-hand side *s* bits are selected along with plaintext bits to which an XOR operation is applied. The result is given as input to a shift register having b-s bits to lhs,s bits to rhs and the process continues. Both of them use encryption algorithms.

iv. Output Feedback Mode –

The output feedback mode follows nearly the same process as the Cipher Feedback mode except that it sends the encrypted output as feedback instead of the actual cipher which is XOR output. In this output feedback mode, all bits of the block are sent instead of sending selected *s* bits. The Output Feedback mode of block cipher holds great resistance towards bit transmission errors. It also decreases the dependency or relationship of the cipher on the plaintext.

v. Counter Mode –

The Counter Mode or CTR is a simple counter-based block cipher implementation. Every time a counter-initiated value is encrypted and given as input to XOR with plaintext which results in ciphertext block. The CTR mode is independent of feedback use and thus can be implemented in parallel.

4. B) Explain Key Calculation Procedure in Simplified DES algorithm.

Ans. 4.B)

The Simplified Data Encryption Standard (S-DES) is a simplified version of the original Data Encryption Standard (DES) algorithm. It uses a shorter key length and fewer rounds for simplicity.

The key calculation procedure in S-DES involves generating two subkeys from aninitial 10-bit key. Here's an explanation of the key calculation procedure in S- DES:

Initial 10-Bit Key (K): The S-DES algorithm starts with an initial 10-bit key, represented as K. This key is provided as input to the encryption and decryptionprocesses.

Permutation P10: The first step in key calculation is to permute the 10-bit key using a fixed permutation called P10. P10 rearranges the bits in the following way:

3 5 2 7 4 10 1 9 8 6

The bits of the initial key are rearranged according to this permutation. The resulting 10-bit value is divided into two 5-bit halves, often denoted as left and right halves (L0 and R0).

Key Generation Rounds:

Round 1:

Left Circular Shift (LS-1):Both L0 and R0 are separately subjected to a left circularshift by one bit. This means that the leftmost bit is moved to the rightmost position, and the other bits are shifted one position to the left.

L0: 3 5 2 7 4 10 1 9 8 6

R0: 5 2 7 4 10 1 9 8 6 3

Permutation P8: After the left circular shift, both L0 and R0 are subjected to another fixed permutation called P8. P8 selects and permutes specific bits from the 10-bit halves to generate the first subkey, often denoted as K1.

P8: 6 3 7 4 8 5 10 9

The bits selected by P8 from both L0 and R0 are combined to form K1, which is a 8-bit subkey.

Round 2:

Left Circular Shift (LS-2): Both L0 and R0 (after the first round) are separately subjected to a left circular shift by two bits this time.

L1: 2 7 4 10 1 9 8 6 3 5

R1: 7 4 10 1 9 8 6 3 5 2

Permutation P8: After the left circular shift, both L1 and R1 are subjected to the P8 permutation again to generate the second subkey, often denoted as K2.

P8: 6 3 7 4 8 5 10 9

The bits selected by P8 from both L1 and R1 are combined to form K2, which is another 8-bit subkey.

At the end of the key calculation procedure, you have generated two 8-bit subkeys, K1 and K2, from the original 10-bit key K. These subkeys are used in the S-DES encryption and decryption processes.

In S-DES, these subkeys are used in a Feistel network structure to perform the initial and final permutations, as well as the rounds of substitution and permutation. This process helps encrypt and decrypt the plaintext.

4. C) Explain in detail about encryption procedure in IDEA algorithm.

Ans. 4.C)

The International Data Encryption Algorithm (IDEA) is a symmetric-key block cipher that operates on 64-bit blocks of data and uses a 128-bit key for encryption and decryption. IDEA is a well-regarded encryption algorithm known for its security and efficiency. The encryption procedure in IDEA in detail:

IDEA Encryption Procedure:

IDEA operates on 64-bit blocks of plaintext and uses a 128-bit key. Here's a step-by-step explanation of the encryption process:

1. Key Expansion: The 128-bit encryption key is expanded into 52 round subkeys. These round subkeys will be used in each of the 8.5 rounds of the encryption process. Each subkey is 16 bits in length.

2. Initial Permutation: The 64-bit block of plaintext is subjected to an initial permutation. This permutation rearranges the bits in the block according to a fixed pattern.

3. Rounds: IDEA consists of 8.5 rounds (16 rounds divided by 2, where 0.5 rounds are applied to the middle of the data block). Each round consists of thefollowing steps:

a. **Subkey Mixing:** The 64-bit data block is divided into four 16-bit blocks (X1,X2, X3, X4). Each of these blocks is then mixed with a 16-bit round subkey.

b. **Substitution (S-Box):** Each of the four 16-bit blocks is passed through a substitution (S-box) step. IDEA uses eight 16x16 S-boxes in this step.

c. **Permutation (P-Box):** After the substitution, each of the four 16-bit blocks goes through a permutation (P-box) step, which shuffles the bits within the blocks.

d. **Linear Transformation:** The outputs of the S-boxes and P-boxes are combined in a linear transformation step, which involves bitwise XOR and modulo addition.

4. Final Permutation: After all rounds are completed, the resulting data block is subjected to a final permutation, which is the inverse of the initial permutation.

5. Output: The final 64-bit block, after the final permutation, is the ciphertext.

It's important to note that IDEA is a symmetric-key encryption algorithm, which means the same key is used for both encryption and decryption. To decrypt data encrypted with IDEA, you would perform the inverse of the encryption process using the same key and round subkeys.

5.B Apply the Chinese Remainder Theorem to solve following congruent equations.

X ≡ (mod 3) X ≡ (mod 5) X ≡ (mod 7)

Ans.5.B)

To solve the system of congruent equations using the Chinese Remainder Theorem (CRT), follow these steps:

1. Write down the given congruences:

--> X ≡ 1 (mod 3)

--> X ≡ 2 (mod 5)

--> X ≡ 3 (mod 7)

2. Calculate the products of the moduli:

Find N, which is the product of all the moduli (3, 5, and 7):

N = 3 * 5 * 7 = 105

3. Find the individual remainders when N is divided by each modulus:

--> N_1 = N/3 = 35

--> N_2 = N/5 = 21

--> N_3 = N/7 = 15

4. Calculate the modular inverses:

For each modulus, find the modular inverse of N_i modulo the corresponding modulus. In this case, the modular inverses are as follows:

--> $y_1 \equiv 35^{-1}$ (mod 3) (Find the modular inverse of 35 modulo 3)

--> $y_2 \equiv 21^{-1}$ (mod 5) (Find the modular inverse of 21 modulo 5)

--> $y_3 \equiv 15^{-1}$ (mod 7) (Find the modular inverse of 15 modulo 7)

The modular inverses are:

--> $y_1 \equiv 2 \pmod 3$

--> $y_2 \equiv 1 \pmod 5$

--> $y_3 \equiv 1 \pmod 7$

5. Compute the final solution:

Use the Chinese Remainder Theorem formula to find X:

$X = (a_1 * N_1 * y_1) + (a_2 * N_2 * y_2) + (a_3 * N_3 * y_3) \pmod N$

where a_1, a_2, a_3 are the remainders when dividing X by the respective moduli:

--> $a_1 = 1$ (from the first congruence)

--> $a_2 = 2$ (from the second congruence)

--> $a_3 = 3$ (from the third congruence)

Now, plug these values into the formula:

$X = (1 * 35 * 2) + (2 * 21 * 1) + (3 * 15 * 1) \pmod{105}$

$X = 70 + 42 + 45 \pmod{105}$

$X = 157 \pmod{105}$

Finally, reduce X modulo 105 to get the smallest non-negative solution:

$X \equiv 52 \pmod{105}$

So, the solution to the system of congruent equations is $X \equiv 52 \pmod{105}$.

6. B) In public key system using RSA you Intercept the cipher text C 10 sent to the user whose public key is e = 5, n = 35, what is the plaintext M?

Ans.6.B)

To demonstrate the RSA decryption algorithm, you'll need the ciphertext (C) andthe public key (e, n). The decryption process involves using the private key (d, n), where d is the private exponent. In order to decrypt, you need to calculate the private key (d) first.

Here are the parameters given:

- Ciphertext (C) = 10

- Public Key (e, n) = (5, 35)

RSA Decryption Steps:

1. Calculate the private key (d):
To calculate the private key (d), you need to find the modular multiplicative inverse of the public exponent (e) modulo (n). In other words, you need to find a value for d such that (d * e) ≡ 1 (mod n). You can use the Extended Euclidean Algorithm to find d.

First, find the modular multiplicative inverse of 5 modulo 35:Using the

Extended Euclidean Algorithm:

35 = 5 * 7 + 0

Since the GCD(5, 35) is 5, there is no modular multiplicative inverse because 5 does not have an inverse modulo 35. This means that the given public key is notvalid, and you cannot decrypt the ciphertext with the provided parameters. RSA encryption and decryption require a valid public-private key pair where the chosen values of e, d, and n satisfy certain conditions, including having amodular multiplicative inverse for e modulo (n).

Therefore, if the public key is not valid, it is not possible to perform RSAdecryption.