1

a) The keystream in stream ciphers is produced by a truly random process or by a pseudo-random number generator. This means that if you generate a long enough keystream, it will ultimately repeat because there are only a finite number of potential states in every finite system and the keystream creation process is predictable. Repetition results from the pigeonhole principle, which says that if there are more "pigeons" than "pigeonholes" then at least one pigeon must occupy the same slot as another.

b. The stream cipher, the repetition of keystream indicates the beginning of the encryption process. Repeating this can result in a number of security flaws. An attacker can infer that the plaintext as the parts of it is mostly repeated which makes it easy to know the plain text. This can help with a variety of cryptographic attacks, like ciphertext-only or known-plaintext assaults, in which the attacker uses the ciphertext's recurring patterns to know the encryption key or the plaintext. Repetition essentially makes the cipher less secure by giving attackers more information that they can use to crack the encryption.

2.

a) From the given question we know that turdy already knows the Plaintext which will be P and the Ciphertext C was also known by turdy as it is sent to Bob by Alice. So, by XORing the plaintext with keystream K we get the ciphertext C which would be C= P⊕K .Turdy can easily determine the keystream by XORing the ciphertext and plaintext by which he can get the keystream that is used on the plaintext.

b) From the already answered above part-a we can say that turdy knows the keystream which makes it easy to change the plaintext P1 as turdy wants. Now, Using the same process as part-a we get new ciphertext C1 by XORing the and K which would be C1= P1⊕K.Turdy can send the C1 ciphered text she wants to Bob in place of the original cipher text from that Bob only gets the P1 plaintext which is the changed text that turdy wants bob to receive. The bob gets the text by using P1⊕K=(P1⊕K) ⊕K=P1⊕0 = P1.So, it proves that Bob got the plaintext which is different from the original text.

3.

a) The X register only works when the clocking bit is 1. The Clocking bit is calculated as the majority bit for some internal bits. As The majority bit is changed frequently the X register works frequently. Hence, we can say on average it steps for every other step.

b) The Y register works similarly to the X register. So, The Y register steps in for every other step on average based on the majority bits like the X register.

c) The Z register works similarly to the X and Y registers. So, The Z register steps in for every other step on average.But the process how it works is different from the other registers.

d) The average on which all 3 registers going to work simultaneously would depend on the product of probabilities of the registers working which would be 1/8.If the registers step in for every half the time.

e) The probability on which two registers are working simultaneously would depend on all the combinations that occur and the product of those probabilities that is 3\*1/8=3/8 which is the average of times that the two registers work simultaneously.

f) The occurrence of one register stepping into work would depend on the product of the number of registers and the number of times it occurs that is ½ from which we get the 3\*1/2= 3/2 which would be the average of a single register stepping in.

g) At least one register does step in at least half the time the probability would be 1-3/2= ½. So, from the probability we can say that on average it takes ½ for no register to step in.

4.  
a) The symmetric structure that is employed in block cipher creation is the Feistel Cipher. It splits the block into two and does several rounds of operations on each half, processing one half as an input to a function that depends on the other half as well as a subkey. This block's other half and the function's output are then XORed, and the two halves are switched. For a certain number of rounds, this procedure is repeated, using a different subkey each time.

b) Yes, the Feistel Cipher is the DES. Employing a 56-bit key, it runs on 64-bit plaintext blocks. It is comprised of sixteen rounds of Feistel operations, permutations, substitutions, and XOR operations depending on subkeys that are subtracted from the primary key.

c) No. Unfortunately, the Advanced Encryption Standard is not a Feistel cipher. The substitution-permutation network structure is used by AES to operate on data blocks. Depending on the encryption key, it performs several round-by-round substitution and permutation operations on the minor part of the block.

d) As it resembles Feistel structures but is different from the traditional definition of a Feistel Cipher, TEA is regarded as "almost" a Feistel Cipher. TEA employs a 128-bit key and runs on 64-bit blocks. It is played across several rounds, with each round applying a straightforward procedure that combines the subkey and both block halves. Conversely, TEA does not have the key-dependent round function that makes Feistel Ciphers unique. Therefore, even if TEA's structure is comparable to that of Feistel Ciphers, it is different from the classic Feistel design.

5.

Create two tables: one containing all potential values for E(P, K1) and another containing all possible values for D(C, K2). This calls for recalculating and storing the outcomes of applying the keys K1 and K2 to encrypt and decrypt every conceivable pair of plaintexts and ciphertext. Encrypt the known plaintext P using every key that is possible. K1, producing a collection of ciphertexts E(P, K1). Using every key that can be generated (K2), decrypt the known ciphertext C to get a collection of plaintexts D(C, K2). If a match is discovered between E (P, K1) and D(C, K2)for any key pairings (K1, K2 ), it suggests a possible contender for the right keys. This match implies that the ciphertext's decryption under K2 and the plaintext's encryption under K1 are identical. Check to see if the pairs correspond to the original ciphertext C. In case the match is verified, the right keys are K1 and K2. By using the meet-in-the-middle attack the complexity is reduced from to which makes it easier as this takes place in between the process of decryption and encryption.

6

a) The attacking approach needs to be modified in a known plaintext assault on double DES when the attacker may access pairings of plaintext and ciphertext but cannot select the plaintext. The attacker would create a table of values for all possible keys (K2) used in the second encryption, matched with their corresponding ciphertexts, rather than a table of ciphertext values for all possible plaintexts encrypted with all possible first keys (K1). The attacker would then use every second key (K2) that could be generated to decrypt the known ciphertexts during the reverse search and compare the intermediate values that were acquired to those in the table. The attacker can deduce that the matching first key (K1) used in the encryption matches the known plaintext-ciphertext pair if a match is found.

b) The work factor in the meet-in-the-middle double DES attack in plaintext is , which is the same as in the exhaustive search for a 56-bit key. This is because, to retrieve the intermediate values and compare them with the values in the table for each of the potential second keys (K2), the attacker must perform a total of operations to decrypt the known ciphertexts.

7

a) To ensure that the ciphertexts are unique, an IV does not have to be random; rather, it should be distinct for every encryption operation using the same key. In situations when an attacker may have some knowledge of the plaintext or the ciphertext's structure, randomness in IVs gives an extra degree of protection.

b)

-Cons:

1. When IVs are chosen sequentially, there's a chance that the same key will be used in several encryption sessions, which could result in repeated IVs. This may result in patterns in the ciphertext, which could expose it to assaults using cryptography techniques like frequency analysis.

2. Sequentially generated IVs can bring deterministic behavior into the encryption process, hence compromising the cipher's security. Cryptanalysis may be aided if an attacker can anticipate the IV utilized in a specific encryption process.

3. Vulnerabilities like IV reuse attacks, in which an attacker takes advantage of ciphertext patterns to obtain information about the plaintext or the encryption key, may arise if the same IV is used for numerous encryption operations using the same key.

-Pros:

1. Sequentially generated IVs may be simpler to administer and put into practice in some systems than random IVs, particularly in situations where randomization is hard to attain or preserve.

2. Sequentially generated IVs may be desirable in some cases where deterministic behavior is needed or acceptable, such as in specific streaming applications or protocols, as they guarantee consistency and reproducibility of encryption outcomes.

These are the pros and cons that happen if IVs are selected in sequence instead of being generated randomly as there are a lot of pros while using the sequential, but it compromises security purposes. So, using the random IVs will give more stability in security.

8.

The formula even after the modification still remains the counter-mode encryption and the keystream is generated by encryption of the key K. So, the security when the modified formula If the function E(K, W + i) acts as a pseudo-random function (PRF), then the modified counter mode is secure. Because each keystream block is derived from a unique combination of the encryption key K and the counter value i incremented by a W, each keystream block is derived from a unique combination of the encryption key K and the concatenated with the counter value, just like in the standard counter mode. As long as the PRF stays secure and produces unexpected outputs for a range of inputs, the updated counter mode will give a security level that is comparable to that of the conventional counter mode. If the function E(K, W + i) does not display attributes typical of a PRF, the modified counter mode's security is compromised. An attacker could utilize any weaknesses in the way K and W + i cooperate or in the way E operates to carry out attacks like plaintext recovery, key recovery, or ciphertext alteration. Should the function E exhibit patterns or K and W + i fail to produce sufficient entropy, biases or correlations could exist in the keystream, which could lead to security issues. So,It completely depends on the E(K, W + i) being PRF or not.

9.

a) The corresponding encryption rule would be the reverse of the process which would be the below decryption:

P0 = IV ⊕ D( C 0 , K),

P1 = C0 ⊕ D(C1 , K)

P2 = C1 ⊕ D(C2 , K)

b)

Potential Leakage of Information: In this, there is no diffusion of information throughout the process of encryption as there is in the CBC as it depends on plaintext and ciphertext of the precious block. In this, only the plaintext is dependent on the previous block of plaintext rather than the ciphertext of the previous block. This all leads to a leak of information and also causes the plaintext to be manipulated.

Lack of Chaining: In this encryption, there are no chaining cipher text blocks. So, every ciphertext is derived without any feedback which causes it to be vulnerable to attacks like ciphertext manipulation and block recording.

Hence, The probability of information getting leaked and getting compromised is very insecure compared to the CBC mode because above disadvantages and especially in the scenarios where there is absolute control over the plaintext and ciphertext.

10.

a) One security issue with CBC mode is that, in order to enable diffusion, the ciphertext from the previous block is XORed with the plaintext before encryption. Every message encrypted with the same key will result in the same beginning ciphertext block if the same IV is used for successive encryption operations using the same key. Even if the plaintexts are distinct, this produces a pattern in the ciphertext. By using this pattern, an attacker might deduce details about the plaintexts and possibly launch attacks like padding oracle attacks or chosen plaintext assaults.

b) CTR mode security issue: The IV is normally employed as a counter value in CTR (Counter) mode to create a stream of key-dependent pseudorandom bits, which are XORed with the plaintext to create the ciphertext. The same keystream is generated for every encryption if the same IV is utilized for numerous encryption operations using the same key. An attacker can XOR ciphertexts together to cancel out the keystream and reveal the XOR of the plaintexts when the same keystream is used for many communications. This might result in other cryptographic attacks or the recovery of plaintext.

c) CTR mode is regarded as more secure when comparing the security of CBC and CTR modes with the same IV. Even though the IV is not random, it serves as a counter in CTR mode to guarantee that every encryption produces a distinct keystream. Keystream reuse attacks are thus avoided. As opposed to this, the CBC mode XORs the IV with the plaintext, resulting in a chaining effect that makes the ciphertext predictable and open to attack. For all modes, it is nevertheless advised to use a distinct and random IV for every encryption operation to guarantee maximum security.

11.

For all the rounds we know that Li = Ri-1

And Ri = Li-1 ⊕F( Ri-1,Ki)

Round - 1

L1 = R0 = 0101

F(R0,K1) = R0 ⊕ K1 = 0101 ⊕ 1011 = 1110

R1 = L0 ⊕F( R0,K1)= 1011⊕1110=0101

Round - 2

L2 = R1 = 0101

F(R1,K2) = R1 ⊕ K2 = 0101 ⊕ 0100 = 0001

R2 = L1 ⊕F( R1,K2)= 0101⊕0001=0100

Round - 3

L3 = R2 = 0100

F(R2,K3) = R2⊕ K3 = 0100 ⊕ 0101 = 0001

R3 = L2 ⊕F( R2,K3)= 0101⊕0001=0100

Round - 4

L4 = R3 = 0100

F(R3,K4) = R3⊕ K4 = 0100 ⊕ 1010 = 1110

R4 = L3 ⊕F( R3,K4)= 0100⊕1110=1010

Therefore, The Feistel cipher is 10100100.