2. To compute the ciphertext stream using a one-time pad, you perform a bitwise XOR operation between the message stream and the key stream. Here are the ciphertext streams for the given message and key streams:

i) Message Stream: 1111111001
Key Stream: 1000000011
Ciphertext: 0111111010

ii) Message Stream: 0000011100
Key Stream: 1110000000
Ciphertext: 1110011100

iii) Message Stream: 1110001111
Key Stream: 0101001111
Ciphertext: 1011000000

iv) Message Stream: 0000011111
Key Stream: 1110000000
Ciphertext: 1110011111

v) Message Stream: 1111111000
Key Stream: 1000000011
Ciphertext: 0111111011

3. Symmetric key cryptography and asymmetric key cryptography are two fundamental cryptographic approaches:

- Symmetric key cryptography: In this approach, a single shared secret key is used for both encryption and decryption. The same key is used by both the sender and the receiver. Examples of symmetric key algorithms include AES (Advanced Encryption Standard) and DES (Data Encryption Standard). Symmetric key cryptography is typically faster but requires secure key distribution.

- Asymmetric key cryptography: Also known as public key cryptography, it uses a pair of mathematically related keys: a public key for encryption and a private key for decryption. The sender uses the recipient's public key to encrypt the message, and the recipient uses their private key to decrypt it. Examples of asymmetric key algorithms include RSA (Rivest-Shamir-Adleman) and ECC (Elliptic Curve Cryptography). Asymmetric key cryptography provides features like key distribution and digital signatures, but it is generally slower than symmetric key cryptography.

4. Three examples of cyber-attacks are:

- Phishing: This is a type of attack where the attacker masquerades as a trustworthy entity through emails, messages, or websites to trick individuals into revealing sensitive information like passwords, credit card details, or social security numbers.

- Distributed Denial of Service (DDoS): In a DDoS attack, multiple compromised computers or devices are used to flood a target system or network with an overwhelming amount of traffic, causing it to become unresponsive or unavailable to legitimate users.

- Ransomware: Ransomware is a form of malicious software that encrypts a victim's files or locks them out of their system until a ransom is paid. It can spread through email attachments, malicious downloads, or exploit kits, and it can cause significant disruption or financial loss to individuals and organizations.

5. Steganography is the practice of hiding secret information within non-secret data, such as images, audio files, or text, in a way that is not easily detectable. Here are five types of digital steganography:

- Image Steganography: Concealing information within the pixels of an image by manipulating their values or using techniques like least significant bit (LSB) substitution.

- Audio Steganography: Hiding data within audio files by modifying the audio samples or using unused frequency bands.

- Video Steganography: Embedding data in video files by modifying frames, motion vectors, or exploiting temporal redundancies.

- Text Steganography: Concealing information within text documents by using techniques like invisible ink or modifying the formatting or spacing of characters.

- File Steganography: Hiding data within various file formats, such as PDF, DOCX, or ZIP, by appending or embedding secret information without affecting the….

7.

i) To prove that SA,B = SB,A using Diffie-Hellman, we need to demonstrate that both sides of the equation are equal. In Diffie-Hellman, the shared secret key is calculated as follows:

SA,B = (YA)^XB mod p
SB,A = (YB)^XA mod p

Let's assume that YA = (g^XA) mod p and YB = (g^XB) mod p, where g is the generator and p is a prime number.

Now, we can substitute the values of YA and YB into the equations:

SA,B = [(g^XA) mod p]^XB mod p
SB,A = [(g^XB) mod p]^XA mod p

Using modular exponentiation rules, we can simplify the equations further:

SA,B = (g^(XA * XB)) mod p
SB,A = (g^(XB * XA)) mod p

Since multiplication is commutative, XA * XB = XB * XA, and therefore SA,B = SB,A. Thus, we have proved the equation SA,B = SB,A using Diffie-Hellman.

ii) To prove the equation DecK(EncK(m)) = m, we need to show that decrypting an encrypted message using the same key K results in the original plaintext message m.

In symmetric key cryptography, the encryption and decryption algorithms are inverses of each other when using the same key. Therefore, if we encrypt a message m with key K to obtain the ciphertext C, and then decrypt the ciphertext C using the same key K, we should retrieve the original message m.

Mathematically, if EncK(m) = C and DecK(C) = m, then DecK(EncK(m)) = m. This property holds for symmetric key algorithms like AES and DES.

8. Given the values p = 11, e1 = 3, d = 7, and r = 5, we can compute the ciphertext C1 and C2 for the plaintext 7 using the ElGamal cryptosystem.

The ElGamal encryption process involves the following steps:

1. Bob's public key is calculated as YB = (e1^d) mod p.
   YB = (3^7) mod 11 = 7.

2. Alice encrypts the plaintext using Bob's public key.
   C1 = (e1^r) mod p = (3^5) mod 11 = 1.
   C2 = (7^r * m) mod p = (7^5 * 7) mod 11 = 10.

So, the ciphertext C1 is 1, and the ciphertext C2 is 10.

To decrypt the ciphertext and obtain the plaintext 7, we use the following steps:

1. Compute the shared secret key as K = (C1^d) mod p.
   K = (1^7) mod 11 = 1.

2. Compute the modular inverse of K as K^(-1) mod p.
   K^(-1) = 1^(-1) mod 11 = 1.

3. Retrieve the plaintext by multiplying C2 by the modular inverse of K and taking the result modulo p.
   m = (C2 * K^(-1)) mod p = (10 * 1) mod 11 = 10.

Therefore, the decrypted plaintext is 10, not 7.

10. Substitution cipher and Transposition cipher are two different types of encryption methods:

- Substitution cipher: In a substitution cipher, each letter in the plaintext is replaced with another letter or symbol based on a predetermined substitution rule or key. Examples include Caesar cipher, where letters are shifted by a fixed number, and the Atbash cipher, where letters are reversed.

Transposition cipher: In a transposition cipher, the letters of the plaintext are rearranged or shuffled according to a specific rule or permutation. The actual letters remain the same, but their order changes. Examples include Rail Fence cipher, where letters are written in a zigzag pattern, and Columnar Transposition cipher, where letters are written in columns and then rearranged.

ii) Block cipher and Stream cipher are two categories of encryption algorithms:

- Block cipher: A block cipher operates on fixed-size blocks of data, typically dividing the plaintext into blocks of equal length. Each block is encrypted or decrypted independently using a key and a specific algorithm. The output of each block is combined to form the ciphertext. Examples include AES and DES.

Stream cipher: A stream cipher encrypts and decrypts data bit by bit or byte by byte, generating a keystream that is combined with the plaintext to produce the ciphertext. The keystream is typically generated based on a key and a specific algorithm. Examples include RC4 and Salsa20.

11. To prove the correctness of the ElGamal cryptosystem, we need to show that the equation [C2 x (C1^d)^(-1)] = P holds true, where C1 and C2 are the ciphertext components, d is the private key, and P is the original plaintext.

In the ElGamal cryptosystem, the encryption process involves the following steps:

- Bob generates a public-private key pair: YB = (e1^d) mod p, where e1 is the generator, d is the private key, and p is a prime number.
- Alice encrypts the plaintext P using Bob's public key YB:
  She chooses a random value r.
  Computes C1 = (e1^r) mod p.
  Computes C2 = (YB^r * P) mod p.

Now, let's prove the equation:

[C2 x (C1^d)^(-1)] = P

We can expand the terms using the values obtained in the ElGamal encryption process:

C2 = (YB^r * P) mod p

Substituting YB = (e1^d) mod p:

C2 = ((e1^d)^r * P) mod p

Since (a^b)^c = a^(b*c):

C2 = (e1^(d*r) * P) mod p

Now, let's consider (C1^d)^(-1):

C1^d = ((e1^r)^d) mod p

Applying the exponentiation rule (a^b)^c = a^(b*c):

C1^d = (e1^(r*d)) mod p

Since (a*b) mod c = (a mod c * b mod c) mod c:

C1^d = ((e1^r) mod p)^d mod p

Using the modular inverse property, (a^(-1)) mod c = (a^(c-1)) mod c:

(C1^d)^(-1) = ((e1^r) mod p)^(d-1) mod p

Substituting these values back into the equation:

[C2 x (C1^d)^(-1)] = (e1^(d*r) * P) mod p * ((e1^r) mod p)^(d-1) mod p

Applying the exponentiation rule (a*b) mod c = (a mod c * b mod c) mod c:

[C2 x (C1^d)^(-1)] = (e1^(d*r) * ((e1^r) mod p)^(d-1) * P) mod p

Since e1^r mod p = C1:

[C2 x (C1^d)^(-1)] = (e1^(d*r) * C1^(d-1) * P) mod p

Using the exponentiation rule a^b * a^c = a^(b+c):

[C2 x (C1^d)^(-1)] = (e1^(d*r + (d-1)) * C1 * P) mod p

Simplifying (d*r + (d-1)):

[C2 x (C1^d)^(-1)] = (e1^(d*r + d - 1) * C1 * P) mod p

Since d*r + d - 1 = d*(r+1) - 1:

[C2 x (C1^d)^(-1)] = (e1^(d*(r+1) - 1) * C1 * P) mod p

Using the property e1^x mod p = 1:

[C2 x (C1^d)^(-1)] = (1 * C1 * P) mod p

Simplifying:

[C2 x (C1^d)^(-1)] = (C1 * P) mod p

Since (a * b) mod c = (b * a) mod c:

[C2 x (C1^d)^(-1)] = (P * C1) mod p

Therefore, we have shown that [C2 x (C1^d)^(-1)] = P in the ElGamal cryptosystem.

12. Given p = 53, g = 3, x = 97, and y = 157, we can compute the shared secret key using the Diffie-Hellman key exchange algorithm.

The steps to compute the key are as follows:

- Bob computes the public key: YB = (g^x) mod p
- YB = (3^97) mod 53 = 8.
- Alice computes the public key: YA = (g^y) mod p
- YA = (3^157) mod 53 = 38.
- Bob shares YB with Alice, and Alice shares YA with Bob.
- Bob computes the shared secret key: K = (YA^x) mod p
- K = (38^97) mod 53 = 49.
- Alice computes the shared secret key: K = (YB^y) mod p
- K = (8^157) mod 53 = 49.

Thus, the shared secret key computed by both Bob and Alice is 49.

ii) Given p = 71, g = 5, x = 31, and y = 29, we can compute the shared secret key using the Diffie-Hellman key exchange algorithm.

The steps to compute the key are as follows:

Bob computes the public key: YB = (g^x) mod p
YB = (5^31) mod 71 = 18.
Alice computes the public key: YA = (g^y) mod p
YA = (5^29) mod 71 = 43.
Bob shares YB with Alice, and Alice shares YA with Bob.
Bob computes the shared secret key: K = (YA^x) mod p
K = (43^31) mod 71 = 5.
Alice computes the shared secret key: K = (YB^y) mod p
K = (18^29) mod 71 = 5.

Thus, the shared secret key computed by both Bob and Alice is 5.

The RSA algorithm involves three main processes: key generation, encryption, and decryption.

Key Generation:

Select two distinct prime numbers, p and q.
Calculate the modulus, n = p * q.
Compute Euler's totient function, φ(n) = (p - 1) * (q - 1).
Choose an encryption exponent, e, such that 1 < e < φ(n), and e is coprime to φ(n).
Compute the decryption exponent, d, which is the modular multiplicative inverse of e modulo φ(n).
Public Key: (e, n) is used for encryption.
Private Key: (d, n) is kept secret and used for decryption.

Encryption:

Convert the plaintext message, M, into a numerical representation.
Apply the encryption function using the public key: C = (M^e) mod n.
The ciphertext, C, is the encrypted message.

Decryption:

Apply the decryption function using the private key: M = (C^d) mod n.
The resulting value, M, is the decrypted plaintext message.

Note: The security of RSA relies on the difficulty of factoring large numbers, which is why the prime numbers p and q need to be sufficiently large.

In summary, RSA key generation involves selecting prime numbers, calculating the modulus and totient function, choosing encryption and decryption exponents, and keeping the private key secret. Encryption is performed using the public key, and decryption is performed using the private key.

16. The correctness of the equation DecK(EncK(m)) = m using the one-time pad can be proven by understanding the properties of the one-time pad encryption. The one-time pad is a perfect secrecy encryption scheme where the key used for encryption is as long as the message and is completely random. In this scheme, each bit of the plaintext is combined with the corresponding bit of the key using the XOR operation.

When encrypting the message m with the one-time pad, the result is obtained by XORing each bit of m with the corresponding bit of the key, resulting in the ciphertext EncK(m). To decrypt the ciphertext, the same key is XORed with the ciphertext, resulting in DecK(EncK(m)). Since XOR is its own inverse (XORing a bit with itself gives 0), XORing the key with the ciphertext cancels out the encryption, revealing the original plaintext m. Therefore, DecK(EncK(m)) = m holds true in the case of the one-time pad.

17. Four cryptosystem services are:

1. Confidentiality: Cryptosystems provide confidentiality by encrypting the data, making it unreadable to unauthorized parties. Encrypted data can only be decrypted by those who possess the necessary key or information.

2. Integrity: Cryptosystems ensure the integrity of data by using techniques like digital signatures or message authentication codes (MACs). These techniques verify that the data has not been tampered with during transmission or storage.

3. Authentication: Cryptosystems support authentication by using methods such as digital certificates or public-key cryptography. These methods verify the identity of communicating parties, ensuring that the intended recipient is the actual recipient.

4. Non-repudiation: Cryptosystems provide non-repudiation by using digital signatures. Digital signatures provide evidence that a specific message was sent by a particular sender, preventing the sender from denying their involvement.

18. Active attacks involve an attacker actively manipulating or interfering with the communication or system, while passive attacks involve the unauthorized monitoring or eavesdropping of communication without altering the data. Examples of active attacks are:

1. Man-in-the-Middle (MitM) Attack: An attacker intercepts the communication between two parties and impersonates each party, relaying messages between them while potentially modifying or eavesdropping on the content.

2. Denial-of-Service (DoS) Attack: An attacker floods a system or network with excessive requests or malicious traffic, making the system or network unavailable to legitimate users.

Examples of passive attacks are:

1. Eavesdropping: An attacker secretly listens to the communication between two parties, intercepting and gathering sensitive information without altering the data.

2. Traffic Analysis: An attacker analyzes the patterns and characteristics of communication, such as message timing, size, or frequency, to gain insights or extract information without directly accessing the content.

19. Using the Diffie-Hellman Key Exchange algorithm:
i) For p = 53, g = 3, x = 97, and y = 157:
- Bob computes B = g^x mod p, which is B = 3^97 mod 53 = 48.
- Alice computes A = g^y mod p, which is A = 3^157 mod 53 = 30.
- Bob sends B to Alice, and Alice sends A to Bob.
- Bob computes the shared key K = A^x mod p, which is K = 30^97 mod 53 = 49.
- Alice computes the shared key K = B^y mod p, which is K = 48^157 mod 53 = 49.

iii) For p = 71, g = 5, x = 31, and y

= 29:
- Bob computes B = g^x mod p, which is B = 5^31 mod 71 = 25.
- Alice computes A = g^y mod p, which is A = 5^29 mod 71 = 42.
- Bob sends B to Alice, and Alice sends A to Bob.

- Bob computes the shared key K = A^x mod p, which is K = 42^31 mod 71 = 26.
- Alice computes the shared key K = B^y mod p, which is K = 25^29 mod 71 = 26.

20. Using RSA encryption and decryption with p = 3, q = 11, and e = 7:
- Compute n = p * q = 3 * 11 = 33.
- Compute the totient φ(n) = (p - 1) * (q - 1) = 2 * 10 = 20.
- Compute the modular inverse of e modulo φ(n), which is d = 3, as e * d ≡ 1 (mod φ(n)).
- Encryption: C = m^e mod n = 3^7 mod 33 = 27.
- Decryption: m = C^d mod n = 27^3 mod 33 = 3.

21. In the ElGamal Cryptosystem, with Bob choosing p = 11, e1 = 3, d = 7, and Alice choosing r = 5:
- Encryption: Bob computes C1 = g^r mod p, which is C1 = 3^5 mod 11 = 1.
- Alice computes C2 = (e1^r) * m mod p, which is C2 = (3^5) * 7 mod 11 = 4.
- Decryption: Bob computes the inverse of C1^d mod p, which is C1^-d mod p = 1^-7 mod 11 = 1.
- Alice computes the plaintext m = (C2 * C1^-d) mod p, which is m = (4 * 1) mod 11 = 7.

22. Proof of the equations:
iii) SA,B = SB,A using Diffie-Hellman:
- Let A = g^x mod p and B = g^y mod p, where g, p, x, and y are public values.
- SA,B = A^y mod p = (g^x mod p)^y mod p = g^(xy) mod p.
- SB,A = B^x mod p = (g^y mod p)^x mod p = g^(yx) mod p.
- Since multiplication is commutative, xy = yx, and g^(xy) mod p = g^(yx) mod p.
- Therefore, SA,B = SB,A.

iv) DecK(EncK(m)) = m using one-time pad:
- The one-time pad encryption is defined as EncK(m) = m XOR K, where K is the random key.
- Decryption is obtained by XORing the ciphertext with the same key: DecK(EncK(m)) = (m XOR K) XOR K.
- Using the properties of XOR, (m XOR K) XOR K = m XOR (K XOR K).
- K XOR K is equal to 0 for any key K, so m XOR (K XOR K) = m XOR 0 = m.
- Therefore, DecK(EncK(m)) = m.

23. To verify V1 = V

2:
- S1 = e1r mod p and S2 = (m – dS1)^(-r) mod (p-1).
- V1 = e1m mod p and V2 = e2S1S1S2.
- Substitute the values:
  - S1 = e1r mod p = 3^5 mod 11 = 1.
  - S2 = (m – dS1)^(-r) mod (p-1) = (7 – 7 * 1)^(-5) mod (11-1) = 0^(-5) mod 10 (undefined).
  - V1 = e1m mod p = 3^7 mod 11 = 7.
  - V2 = e2S1S1S2 = 3^(1 * 1 * 1 * 0) mod 11 = 3^0 mod 11 = 1.
- V1 = 7 and V2 = 1, so V1 ≠ V2.

24. Diffie-Hellman Key Agreement diagram:

```
Bob                          Alice
--------------------------------------------------
Choose p, g                  Choose p, g
Compute x                    Compute y
Compute B = g^x mod p        Compute A = g^y mod p
Send B                       Send A
Receive A                    Receive B
Compute shared key K = A^x mod p   Compute shared key K = B^y mod p
```

25. Five challenges of computer security are:

1. Malware and Cyberattacks: The constant threat of malware, viruses, ransomware, and various cyberattacks poses a significant challenge to computer security. Attackers continuously develop new methods to exploit vulnerabilities and gain unauthorized access to systems.

2. Data Breaches and Privacy: Protecting sensitive data from unauthorized access or breaches is a critical challenge. Ensuring privacy, maintaining data integrity, and preventing unauthorized disclosure of personal or sensitive information require robust security measures.

3. Social Engineering: Human manipulation techniques, such as phishing, social engineering, and impersonation, pose a significant challenge to computer security. Attackers often exploit human trust and vulnerabilities to gain access to systems or sensitive information.

4. Emerging Technologies and Complexity: The rapid advancement of technologies like artificial intelligence, Internet of Things (IoT), and cloud computing introduces new security challenges. Securing complex systems and networks while adapting to evolving technologies is a constant challenge.

5. Insider Threats: Internal actors, such as employees or contractors, can pose a significant security risk. Unauthorized access, data theft, or intentional harm to systems from within an organization are challenges that require proper access controls, monitoring, and mitigation strategies.