1. **Explain the relationship between Security Services and mechanisms in detail.**

In the field of information security, security services and mechanisms are two interrelated concepts that work together to provide a secure environment for data communication.

Security services refer to the protection that is provided to data in a communication system. The five primary security services are confidentiality, integrity, availability, authentication, and non-repudiation.

1. Confidentiality: This service ensures that data is protected from unauthorized disclosure. This can be achieved through encryption, where the data is transformed into a coded message that can only be read by those who have the decryption key.
2. Integrity: This service ensures that data is protected from unauthorized modification. Data integrity can be ensured by using cryptographic hash functions that calculate a unique message digest for the data, which can be used to verify the integrity of the data.
3. Availability: This service ensures that data is available to authorized users when they need it. Availability can be ensured by implementing backup and redundancy systems to prevent downtime or loss of data.
4. Authentication: This service ensures that the identity of the users and systems in a communication system is verified. Authentication can be achieved through passwords, digital certificates, and biometric authentication.
5. Non-repudiation: This service ensures that the originator of a message cannot deny sending it and the recipient cannot deny receiving it. Non-repudiation can be achieved through the use of digital signatures.

Security mechanisms are the tools and techniques used to implement security services. Some of the commonly used security mechanisms include:

1. Encryption: Encryption is a technique that converts plain text into cipher text to protect the confidentiality of data.
2. Digital signatures: Digital signatures use cryptographic techniques to ensure the authenticity and integrity of data.
3. Firewalls: Firewalls are security devices that monitor and control network traffic to prevent unauthorized access.
4. Intrusion Detection Systems (IDS): IDS are used to monitor network traffic for signs of unauthorized access or malicious activity.
5. Access controls: Access controls are used to restrict access to data and systems based on user credentials, roles, and permissions.

In summary, security services provide the goals of security, while security mechanisms provide the means of achieving those goals. Effective security implementation requires careful selection and integration of both security services and mechanisms to ensure the confidentiality, integrity, and availability of data.

1. **Explain the ECB and CBC, OFB modes of block cipher.**

[**https://www.geeksforgeeks.org/block-cipher-modes-of-operation/**](https://www.geeksforgeeks.org/block-cipher-modes-of-operation/)

ECB, CBC, and OFB are three common modes of operation for block ciphers, which are cryptographic algorithms that process fixed-length blocks of data at a time. Each of these modes offers a unique way of using a block cipher to provide confidentiality and integrity to data.

Electronic Codebook (ECB) mode:

ECB mode is the simplest and most straightforward mode of operation. In ECB mode, each block of plaintext is encrypted independently using the block cipher, with no dependence on the previous or subsequent blocks. This means that identical plaintext blocks will always produce identical ciphertext blocks, which can lead to security vulnerabilities. ECB mode is suitable for encrypting small amounts of data or when there is a requirement to encrypt individual blocks independently.

Cipher Block Chaining (CBC) mode:

In CBC mode, each plaintext block is XORed with the previous ciphertext block before being encrypted with the block cipher. This means that the ciphertext for each block depends on the previous block, which makes it more resistant to certain types of attacks than ECB mode. CBC mode also requires an initialization vector (IV) to XOR with the first plaintext block. CBC mode is commonly used for secure data transmission over a network or storage of data in a database.

Output Feedback (OFB) mode:

OFB mode is a stream cipher mode, meaning it converts the block cipher into a stream cipher by generating a pseudo random stream of bits, which is then XORed with the plaintext to generate the ciphertext. The pseudorandom stream is generated by repeatedly encrypting an initialization vector (IV) with the block cipher. OFB mode can be used for encryption and authentication of data and is also suitable for encrypting data in a streaming mode.

In summary, the ECB mode encrypts each block independently, while CBC mode encrypts each block based on the previous block's ciphertext. The OFB mode generates a pseudorandom stream of bits by repeatedly encrypting an initialization vector. Each mode has its advantages and disadvantages, and the choice of mode depends on the specific security requirements and constraints of the application.

1. **Define non-repudiation and authentication. Show with example how it can be achieved.**

Non-repudiation and authentication are two essential concepts in information security that are used to establish trust and ensure the integrity and authenticity of digital communications.

**Non-Repudiation:**

Non-repudiation is the ability to prove that a sender sent a message and that the message was not altered or denied by the sender after transmission. Non-repudiation is important to prevent someone from denying that they sent a message or claiming that the message was modified during transmission. Non-repudiation can be achieved through the use of digital signatures, which provide cryptographic proof of the origin and integrity of a message.

For example, consider a scenario where Alice wants to send a message to Bob, and both parties want to ensure that the message cannot be repudiated. To achieve non-repudiation, Alice can use a digital signature to sign the message. A digital signature is a cryptographic technique that uses a private key to create a unique hash value that can only be created by the owner of the private key. When Bob receives the message, he can verify the digital signature using Alice's public key, which can confirm that the message was not modified and that Alice is the sender.

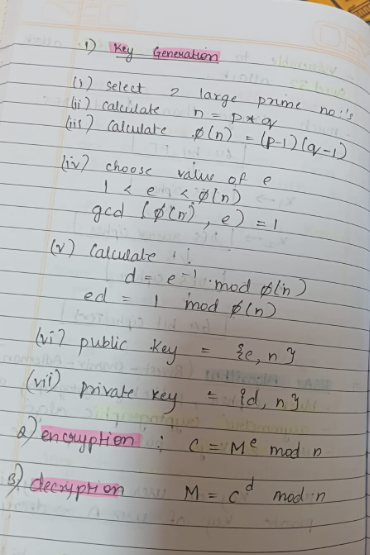
**Authentication**:

Authentication is the process of verifying the identity of a user, system, or device. Authentication is important to ensure that only authorized users or systems can access sensitive information or resources. Authentication can be achieved through the use of passwords, tokens, biometrics, or digital certificates.

For example, consider a scenario where Alice wants to access a secure system that requires authentication. To authenticate herself, Alice can provide a username and password, which are verified by the system's authentication server. If the credentials are correct, Alice is granted access to the system. Another example of authentication is the use of digital certificates, which are used to verify the identity of a website or server. When a user accesses a secure website, the website's digital certificate is presented to the user's browser, which can verify the authenticity of the website and establish a secure connection.

In summary, non-repudiation is the ability to prove that a message was sent by a specific sender and was not altered or denied, while authentication is the process of verifying the identity of a user, system, or device. Both concepts are essential for ensuring the integrity and security of digital communications.

1. **Elaborate the steps of key generation using the RSA algorithm. In the RSA system the public key (E,N) of user A is defined as (7,187). Calculate phi(N) and private Key ‘D’. What is the cipher text for M=10 using the public key.**

****

The RSA Algorithm is a popular Public-Key encryption Technique.

It is based on exponentiation in a finite field over integers module a prime

The integers used by this method is very large make it difficult to solve (by an attacker w/o the key)

There are two set of keys in this method: a private key and a public key.

Initially we begin with finding two prime numbers p and q.

Care must be taken that these two prime numbers must be sufficiently large.

A guess-and-check method to find these numbers will be nice.

Next we need two exponents e (encryption key)and d(decryption key);

Of these both we guess the value of e randomly such that :

1<e<Φ(n)andGCD(e,Φ(n))=1…………{Note:Φ(n)=(p−1)∗(q−1)}

Next, to find the d value (decryption key) , we use the formula: e.d=1 mod Φ(n) and 0≤d≤n

This is solved by using Extended Euclid’s Method shown in Q-1.

Now, we publish the public key which can be read by anyone. The public key is made up of PU= {e,n} . Any third party can access this public-key and encrypt the message.

You CANNOT decrypt a message using the public key.

The Private key, on the other had is kept secret. The private key is given as PR={d,n}

RSA (Rivest-Shamir-Adleman) is a widely used public-key cryptography algorithm that can be used for encryption, digital signature, and key exchange. Now let's apply these steps to the given values:

Choose two large prime numbers p and q: Let p = 11 and q = 17.

Calculate the modulus n = p \* q = 11 \* 17 = 187.

Calculate Euler's totient function φ(n) = (p-1) \* (q-1) = 10 \* 16 = 160.

Choose the public exponent e = 7. Since gcd(7, 160) = 1, e is a valid choice for the public key.

Calculate the modular multiplicative inverse d of e modulo φ(n) using the extended Euclidean algorithm:

φ(n) = 160, e = 7

160 = 22 \* 7 + 6

7 = 1 \* 6 + 1

6 = 6 \* 1 + 0

Since gcd(e, φ(n)) = 1, the modular inverse of e modulo φ(n) exists.

1 = 7 \* 23 - 160 \* 1

d = 23

The private key is (23, 187).

To encrypt a message M using the public key (e, n), we use the formula C ≡ M^e (mod n), where C is the ciphertext.

For M = 10, using the given public key (e, n) = (7, 187):

C ≡ 10^7 (mod 187)

C ≡ 47829690 (mod 187)

C ≡ 59

Therefore, the ciphertext for M = 10 using the given public key is 59.

1. **Discuss DES with reference to following points.**

* **Block size and key size.**
* **Need of expansion permutation.**
* **Role of S-box.**
* **Weak keys and semi-weak keys.**

DES (Data Encryption Standard) is a symmetric block cipher that was widely used in the 1970s and 1980s. It uses a 64-bit block size and a 56-bit key size, which is considered relatively small by modern standards.

**Block size and key size:**

The block size of DES is 64 bits, which means that it can encrypt messages up to 64 bits in length. However, if a message is longer than 64 bits, it must be broken up into multiple 64-bit blocks and encrypted separately. The key size of DES is 56 bits, which means that there are 2^56 possible keys. However, due to weaknesses in the algorithm, the effective key size is only 48 bits.

**Need for expansion permutation:**

The expansion permutation is used to expand the 32-bit input of the Feistel function to a 48-bit value, which is then XORed with the round key. The purpose of the expansion permutation is to provide diffusion by increasing the number of bits that are affected by each round of the Feistel function. This helps to make the algorithm more resistant to cryptanalysis.

**Role of S-boxes:**

The S-boxes (Substitution boxes) are a key component of the DES algorithm. They take a 6-bit input and produce a 4-bit output, based on a fixed lookup table. The S-boxes provide confusion by making it difficult to determine the relationship between the input and output of the Feistel function. This helps to prevent attacks such as differential cryptanalysis.

**Weak keys and semi-weak keys:**

DES has a number of weak keys and semi-weak keys that can make the algorithm more vulnerable to attack. A weak key is a key that produces the same output for all 64-bit plaintext blocks, while a semi-weak key is a key that produces a limited set of output values. These keys can make it easier for an attacker to recover the plaintext or key.

In summary, DES is a symmetric block cipher with a 64-bit block size and a 56-bit key size. It uses an expansion permutation to increase the number of bits affected by each round of the Feistel function and S-boxes to provide confusion. However, DES has weaknesses such as a small key size and weak/semi-weak keys that can make it vulnerable to attack. As a result, DES is no longer considered secure for use in modern applications and has been replaced by stronger encryption algorithms such as AES.

1. **Possible attacks on DES.**

DES (Data Encryption Standard) is a widely known and studied encryption algorithm that has been in use since the 1970s. Over the years, several attacks on DES have been developed, some of which have demonstrated the weaknesses of the algorithm. Here are some of the possible attacks on DES:

**Brute-force attack**: Since DES uses a relatively small key size of 56 bits, it is vulnerable to brute-force attacks. This involves trying every possible key until the correct one is found. While this may seem like an overwhelming task, advances in computing power have made it feasible to perform a brute-force attack on DES within a reasonable time frame.

**Cryptanalysis attack:** This attack involves analyzing the encryption algorithm to find vulnerabilities that can be exploited to decrypt the message. Cryptanalysis attacks can be targeted at the algorithm itself or at the implementation of the algorithm.

**Side-channel attacks**: These are attacks that exploit weaknesses in the implementation of the algorithm, rather than weaknesses in the algorithm itself. For example, an attacker could measure the power consumption of a device while it is running the DES algorithm to gain information about the key.

**Meet-in-the-middle attack:** In this attack, an attacker intercepts an encrypted message and applies a brute-force attack by trying all possible keys to decrypt the message. This process is done for each key in the first round of encryption, and then the results are encrypted again with all possible keys for the second round. The attacker then compares the results of both brute-force attacks to find the matching key pair.

In summary, DES has several vulnerabilities and is susceptible to a range of attacks, including brute-force attacks, differential cryptanalysis, linear cryptanalysis, side-channel attacks, and meet-in-the-middle attacks. While DES was a groundbreaking algorithm in its time, it is no longer considered secure and has been largely replaced by stronger encryption algorithms such as AES.

1. **DES function.**

The Data Encryption Standard (DES) is a symmetric key encryption algorithm that encrypts data in 64-bit blocks using a 56-bit key. The DES algorithm is built around a Feistel network structure, which consists of a series of rounds that apply a substitution-permutation network to the input block of data.

**Here are the steps involved in the DES function:**

**Key generation**: The 56-bit key is transformed into 16 subkeys, each consisting of 48 bits. These subkeys are generated using a process called key schedule, which involves permutation and shifting the bits of the original key.

**Initial permutation:** The 64-bit input block is permuted using a fixed permutation table. The purpose of this permutation is to distribute the input bits across the entire block, so that each bit affects multiple output bits.

**Splitting**: The permuted input block is split into two 32-bit halves, which are denoted as L0 and R0.

**Round function**: The DES algorithm uses a Feistel network structure, which applies a round function to the right half of the block (R0) in each round. The round function consists of the following steps:

a. **Expansion**: The 32-bit input is expanded to 48 bits using an expansion permutation.

b. **Key mixing**: The 48-bit result is XORed with a 48-bit subkey.

c. **Substitution**: The result of the key mixing is split into eight 6-bit chunks, which are each substituted using a different S-box. The S-boxes provide non-linear diffusion by mapping a 6-bit input to a 4-bit output.

d. **Permutation**: The result of the S-boxes is permuted using a fixed permutation table.

**Round repetition**: The Feistel network repeats the round function 16 times, with the subkeys applied in reverse order for each round. The left half (L0) is swapped with the right half (R0) after each round.

**Final permutation**: After 16 rounds, the left and right halves of the block are combined and permuted using a fixed permutation table. This produces the final encrypted output.

In summary, the DES function uses a Feistel network structure to apply a substitution-permutation network to a 64-bit input block. The algorithm applies a round function to the right half of the block in each round, using a series of 48-bit subkeys that are generated from the original 56-bit key. The DES algorithm has been widely studied and is now considered insecure for modern applications due to its small key size and susceptibility to certain attacks.

1. **Significance of extra swap between left and right half blocks.**

In the Data Encryption Standard (DES) algorithm, a crucial step in each round of the Feistel network structure involves swapping the left and right halves of the input block. This extra swap is significant because it ensures that the left half of the block is used as the input to the round function in the next round.

To understand why this is important, consider the basic structure of a Feistel network. In a Feistel network, the input block is split into two halves, which are denoted as L and R. The output of each round is a new pair of L and R values, which are then used as the input to the next round. The important property of a Feistel network is that the same round function is applied to the R value in each round, while the L value is passed through unchanged.

If the left and right halves of the block were not swapped at the end of each round in DES, then the right half of the output from the previous round would be used as the input to the round function in the next round. This would mean that the right half of the block would be subject to the same round function in every round, while the left half would be unchanged. This would not satisfy the basic structure of a Feistel network, where the same function is applied to both halves of the block.

By swapping the left and right halves at the end of each round in DES, the Feistel network structure is preserved. The left half of the output from the previous round becomes the right half of the input to the current round, and vice versa. This ensures that both halves of the block are subject to the same round function, but with different subkeys, in each round.

In summary, the extra swap between the left and right halves of the block in DES is a crucial step in the Feistel network structure. It ensures that both halves of the block are subject to the same round function, while preserving the basic structure of a Feistel network.

**9. Feistel structure and its significance.**

The Feistel structure is a type of block cipher design that uses a symmetric key to encrypt and decrypt data. The structure was first introduced by Horst Feistel in the early 1970s and has since become a widely used technique for building secure and efficient encryption algorithms.

The Feistel structure consists of a series of rounds, each of which applies a substitution-permutation network to a portion of the input block. The input block is typically split into two halves, denoted as L and R, and each round applies the same function to the R half while leaving the L half unchanged. The output of each round is then used as the input to the next round, with the L and R halves swapped before the next round begins. The final output of the algorithm is the result of the last round.

The significance of the Feistel structure lies in its ability to provide both diffusion and confusion, two key properties of secure encryption algorithms. Diffusion refers to the property of an encryption algorithm that ensures that small changes in the input data result in large changes in the output data. Confusion refers to the property that ensures that the relationship between the input and output data is complex and non-linear, making it difficult to infer the original input from the encrypted output.

The Feistel structure achieves these properties through the use of the same function in each round, applied to one half of the input block while leaving the other half unchanged. The function typically involves a combination of substitution and permutation operations, which provide both diffusion and confusion by introducing non-linearity and ensuring that changes to one half of the block affect both halves in a complex way.

Another significant advantage of the Feistel structure is its flexibility. The structure allows for the use of a variety of different functions and permutations, which can be tailored to the specific needs of the application. This flexibility has led to the development of many different encryption algorithms based on the Feistel structure, including DES, Blowfish, and Twofish, among others.

In summary, the Feistel structure is a block cipher design that uses a series of rounds to apply a substitution-permutation network to the input data. The structure provides both diffusion and confusion, two key properties of secure encryption algorithms, and is flexible enough to support a wide variety of different encryption algorithms.

**9. What are the properties of a hash function? Explain role of hash function in security.**

Cryptographic hash function takes a message of arbitrary length and creates a message digest of fixed length.

The cryptographic hash function h(x) criteria are as follows:

Compression: For any size of input x, the output length of y = h(x) is small. Hash functions produce a fixed size output regardless of the length of the input.

Efficiency: It must be easy to compute h(x) for any input x. the computational effort required to compute h(x) will certainly grow with the length of x, but it should not grow too fast.

One-way: Given any value y, it is computationally infeasible to find a value x such that h(x)=y.

It is difficult to invert the hash.

Weak collision resistance: Given x and h(x) it is infeasible to find y with y≠x

such that h(y)=h(x).

Strong collision resistance: It is infeasible to find y with x≠y

such that h(x)=h(y).

The role of hash functions in security is crucial. They are used in a wide variety of applications to ensure data integrity, authenticity, and confidentiality. Some of the main applications of hash functions in security include:

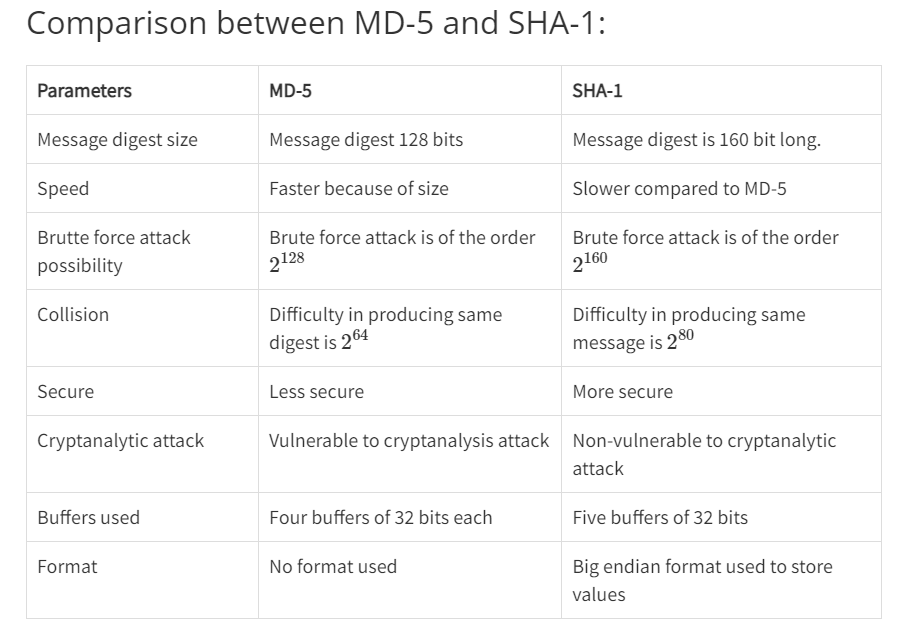
Message authentication: Hash functions are commonly used in message authentication codes (MACs) to provide message integrity and authenticity. A MAC is a cryptographic checksum that is computed using a secret key and a message. The receiver can verify the authenticity and integrity of the message by recomputing the MAC and comparing it to the received MAC.

Digital signatures: Hash functions are used in digital signature algorithms to ensure the integrity of a signed message. The sender computes a hash of the message and then signs the hash using their private key. The receiver can verify the signature by computing the hash of the message and verifying it against the signed hash using the sender's public key.

Password storage: Hash functions are used to store passwords securely by hashing them before storing them in a database. This ensures that even if the database is compromised, an attacker cannot easily recover the passwords.

File and data integrity: Hash functions are used to verify the integrity of files and data by computing a hash of the file or data and comparing it to the expected hash. If the hashes match, the file or data is considered to be intact and unmodified.

In summary, hash functions are an essential tool in cryptography and security. They provide a secure and efficient means of ensuring data integrity, authenticity, and confidentiality in a wide range of applications.



1. **What are traditional ciphers? Discuss any one substitution cipher and transposition cipher with example. List their merits and demerits.**

Traditional ciphers refer to classical encryption techniques that were developed before the advent of modern cryptography. These ciphers rely on the mathematical transformation of plaintext into ciphertext. There are two main types of traditional ciphers: substitution ciphers and transposition ciphers.

Substitution Cipher:

A substitution cipher replaces each letter in the plaintext with a different letter in the ciphertext. One example of a substitution cipher is the Caesar cipher. In the Caesar cipher, each letter is shifted by a fixed number of positions in the alphabet. For example, if the shift is three, then the letter A would be replaced with D, B would be replaced with E, and so on. The following is an example of the Caesar cipher with a shift of 3:

Plaintext: ATTACKATDAWN

Ciphertext: DWWDFNDWGZDQ

Merits of Substitution Cipher:

Easy to understand and implement

Can be used as a building block for more complex ciphers

Can provide a basic level of security against casual eavesdropping

Demerits of Substitution Cipher:

Vulnerable to frequency analysis attacks

Limited key space makes it easy to brute-force

Not secure against modern cryptanalytic techniques

Transposition Cipher:

A transposition cipher rearranges the letters of the plaintext without changing them. One example of a transposition cipher is the Rail Fence cipher. In the Rail Fence cipher, the plaintext is written in a zigzag pattern across a number of rows, and then read off row by row. For example, if the plaintext is ATTACKATDAWN and there are three rows, the following is the ciphertext:

A . . . T . . . W . .

. T . A . K . A . N .

. . D . . C . . A . .

Ciphertext: ATWTAKANDCDA

Merits of Transposition Cipher:

Difficult to perform frequency analysis attacks

Can be used in conjunction with substitution ciphers to provide additional security

Can be used to encrypt messages of any length

Demerits of Transposition Cipher:

Vulnerable to brute-force attacks if the key space is small

May reveal patterns in the plaintext if the key is not chosen carefully

Not secure against modern cryptanalytic techniques

In summary, substitution and transposition ciphers are traditional ciphers that rely on simple mathematical transformations of plaintext into ciphertext. While they may provide some level of security against casual eavesdropping, they are not secure against modern cryptanalytic techniques and should not be used for sensitive data.

1. **A secure e-voting system is to be designed. Discuss the security goals that must be met and enlist mechanisms for the same**

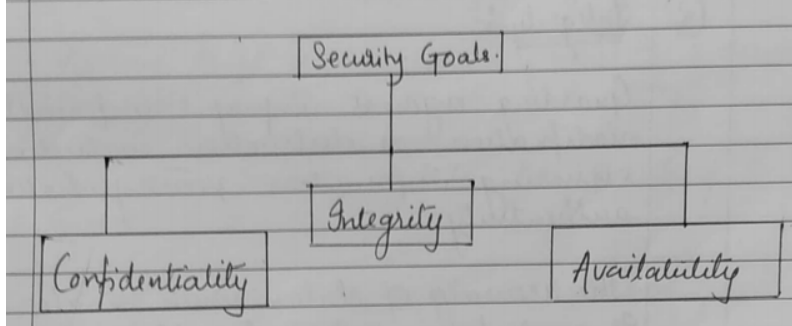
Security goals for a secure e-voting system & mechanism for the same

The three Security Goals are

i. Confidentiality

ii. Integrity

iii. Availability



1. Authentication: The system must authenticate the voters to ensure that only authorized voters can cast their votes. Authentication can be achieved using various mechanisms such as biometric authentication, smart card authentication, or username-password authentication.

2. Confidentiality: The system must ensure that the voter's choices are kept confidential to prevent vote buying, coercion, or any other form of manipulation. This can be achieved by using encryption techniques to protect the voter's choices.

3. Integrity: The system must ensure that the voter's choices are recorded accurately and cannot be tampered with. This can be achieved by using digital signatures to protect the integrity of the votes.

4. Non-repudiation: The system must ensure that the voter cannot deny having cast a vote. This can be achieved by using digital signatures to provide a trail of the voter's actions.

5. Availability: The system must ensure that the voting process is available to all authorized voters throughout the voting period. This can be achieved by using redundant servers, load balancing techniques, and other mechanisms to ensure high availability.

Mechanisms

Encipherment :- Hiding or covering the vote casted by the voter. that is, it provides confidentiality.

Data integrity:- A start checkvalue must be specified, one vote should be casted by a single voter & it should not be altered for that some checkvalue to be given to the voter that his vote is sealed & no modification must be done

Notarization :- It means a third party to control communication between two entities.

4.Access Control :- Use methods to prove that a voter has access right to the data response ovened by him.

Example: His voter id, voter card no etc.

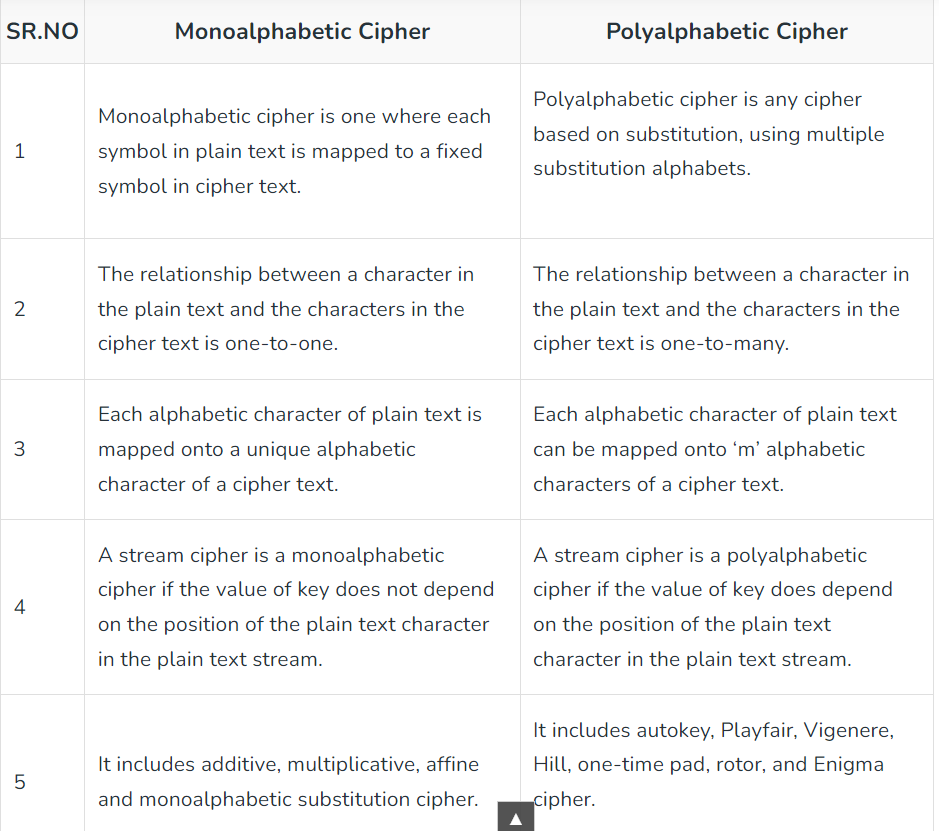
1. **With the help of suitable examples, compare and contrast monoalphabetic and polyalphabetic ciphers?**

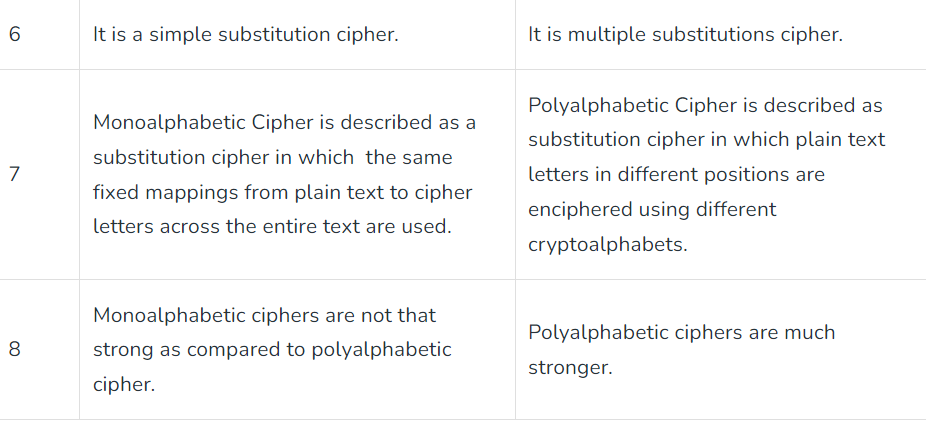
1. Monoalphabetic Cipher :

A monoalphabetic cipher is any cipher in which the letters of the plain text are mapped to cipher text letters based on a single alphabetic key. Examples of monoalphabetic ciphers would include the Caesar-shift cipher, where each letter is shifted based on a numeric key, and the atbash cipher, where each letter is mapped to the letter symmetric to it about the center of the alphabet.

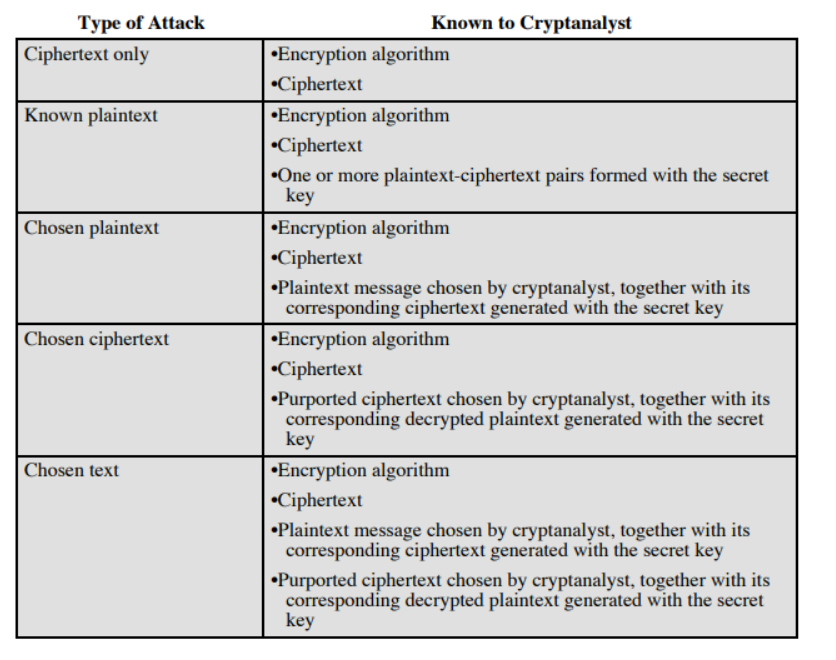
2. Polyalphabetic Cipher :

A polyalphabetic cipher is any cipher based on substitution, using multiple substitution alphabets. The Vigenère cipher is probably the best-known example of a polyalphabetic cipher, though it is a simplified special case.





1. **List and explain various types of attacks on encrypted messages**

****

1. **Explain with an example keyed and keyless transposition cipher.**

Transposition Cipher:

A transposition cipher does not substitute one symbol for another (as in substitution cipher), but changes the location of these symbols.

It reorders (jumbles) the given plain-text to give the cipher-text.

They are of two types: Keyed and Keyless Transposition Cipher.

Keyless Transposition Cipher:

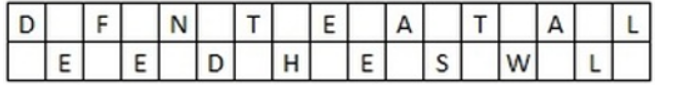
In this cipher technique, the message is converted to ciphertext by either of two permutation techniques:

a. Text is written into a table column-by-column and is then transmitted row-by-row.

b. Text is written into a table row-by-row and is then transmitted column-by-column

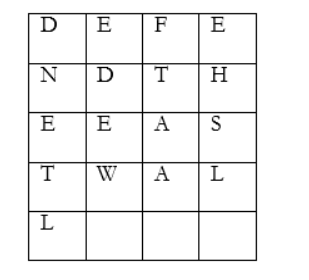
The first method (a) is also popularly known as Rail-fence cipher

E.g. We need to send the message “DEFENDTHEEASTWALL”. Arranging into tables we get :



Now, the message is sent row-by-row. So Ciphertext is “DFNTEATALEEDHESWL”(Note: the no. of rows is 2 by default, unless specified)

Similarly for the (b) method, we can arrange the same above message into tables with four columns.



The Data is then transmitted column-by-column as “DNETLEDEWFTAAEHSL”

Keyed Transposition cipher:

In this approach, rather than permuting all the symbols together, we divide the entire plaintext into blocks of predetermined size and then permute each block independently.

Suppose A wants to send a message to B “WE HAVE AN ATTACK”. Both A and B agreed to had previously agreed oved the blocks size as 5. So the blocks would be as:



The last character X is a bogus character so as to complete the block size of 5.

A and B is using the following key for encryption and decryption:



1. Explain man in the middle attack on Diffie Hellman. Explain how to overcome the same.

A Man-in-the-middle (MITM) attack is a type of cyber attack where an attacker intercepts the communication between two parties and alters or eavesdrops on the messages being exchanged. The attacker can modify or manipulate the communication between the two parties to gain unauthorized access or information.

In the case of Diffie-Hellman key exchange, the MITM attack occurs when the attacker intercepts the public keys exchanged between the two parties and replaces them with their own public keys. The attacker then performs a key exchange with each of the two parties, establishing two separate secret keys - one with each party. The attacker then forwards the messages between the two parties, decrypting and re-encrypting them as necessary using the two separate secret keys. This way, the attacker can eavesdrop on the conversation between the two parties, and even modify the messages as needed.

To overcome the MITM attack on Diffie-Hellman, several methods can be used:

Authentication: One way to prevent MITM attack is to use authentication mechanisms to ensure that the public keys exchanged between the two parties belong to the intended recipients.

Digital signatures: Digital signatures can be used to verify the authenticity of the public keys exchanged between the two parties.

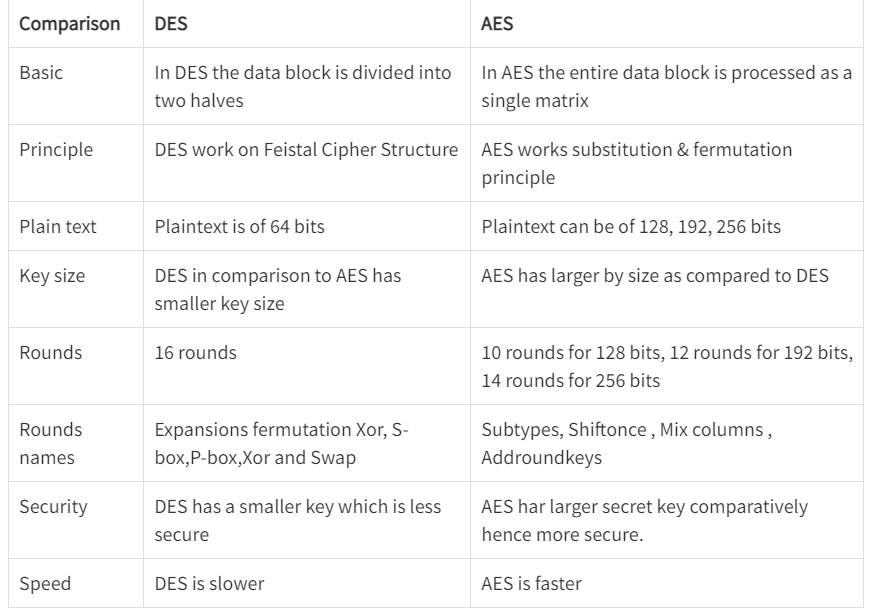
Public key infrastructure (PKI): A PKI can be used to issue digital certificates that are used to verify the identity of the parties involved in the key exchange. This can prevent MITM attacks by ensuring that the public keys exchanged between the parties are authentic.

Certificate pinning: Certificate pinning is a method where the client verifies the server's certificate against a pre-defined fingerprint or public key. This ensures that the server's certificate is not replaced by an attacker's certificate.

Perfect Forward Secrecy (PFS): PFS is a mechanism that ensures that even if an attacker intercepts the communication and gains access to the secret key, the attacker cannot decrypt past or future messages. This is achieved by generating a new secret key for each session, ensuring that even if one session is compromised, the other sessions remain secure.

In summary, the MITM attack on Diffie-Hellman can be prevented by using authentication, digital signatures, PKI, certificate pinning, and PFS. By implementing these security measures, the parties involved in the key exchange can be assured that their communication remains secure and confidential.

1. **Compare AES and DES. Which one is bit oriented? Which one is byte oriented?**

****

1. Discuss in detail block cipher modes of operation.

<https://www.geeksforgeeks.org/block-cipher-modes-of-operation/>

1. **Short note on: 3DES,blowfish.**

**Triple DES with 2-keys**

It uses three stages of DES for encryption and decryption. The 1st, 3rd stage use K1

key and 2nd stage use K2

key.

To make triple DES compatible with single DES, the middle stage uses decryption in the encryption side and encryption in the decryption side.

It’s much stronger than double DES.

The function follows an encrypt-decrypt-encrypt (EDE) sequence.

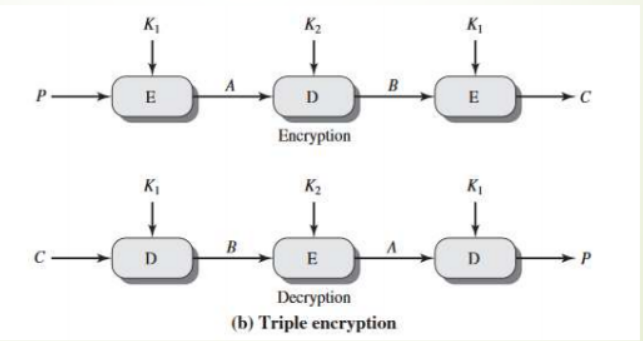
C=E(K1,D(K2,E(K1,P)))

P=D(K1,E(K2,D(K1,C)))

By the use of triple DES with 2-key encryption, it raises the cost of the meet-in-the-middle attack to 2112

It has the drawback of requiring a key length of 56×3=168

bits which may be somewhat unwieldy.



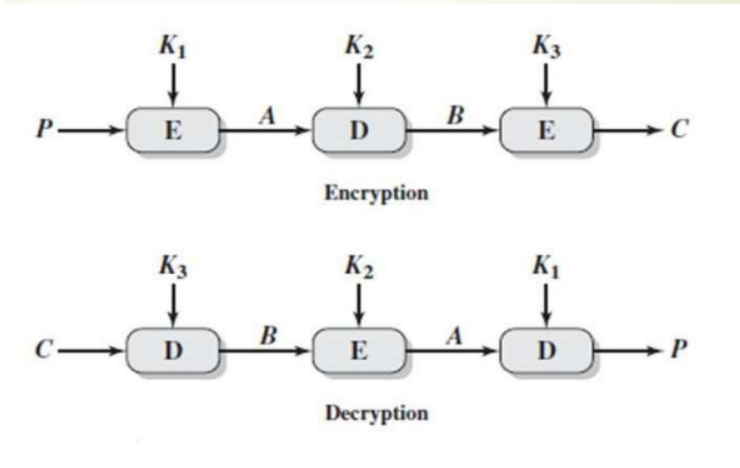
Triple DES with 3-key

Although the attacks just described appear impractical, anyone using two key 3DES may feel some concern. Thus, many researchers now feel that 3-key 3DES is the preferred alternative.

It uses three stages of DES for encryption and decryption with three different keys. 3-key 3DES has an effective key length of 168 bits and is defined as,

C=E(K3,D(K2,E(K1,P)))

P=D(K1,E(K2,D(K3,C)))



Blowfish:

Blowfish is a symmetric block cipher developed by Bruce Schneier in 1993. It is a fast and efficient cipher that can be implemented in software, making it suitable for use in applications where hardware acceleration is not available. Blowfish uses a variable-length key, ranging from 32 bits to 448 bits, making it more flexible than other ciphers with fixed key sizes.

Blowfish encrypts data in 64-bit blocks using a Feistel network, a technique that splits the block into two halves and applies a series of substitutions and permutations. The key schedule, which generates the round keys used in encryption and decryption, is also based on the key itself, making it difficult for attackers to guess the key.

Blowfish has been widely adopted due to its fast encryption and decryption speeds, making it suitable for use in software applications. However, Blowfish is not as widely used as other modern ciphers, such as AES, due to concerns about its security and the lack of formal analysis.

In summary, both 3DES and Blowfish are strong encryption algorithms, but they have different strengths and weaknesses. 3DES is slower but more secure than Blowfish, while Blowfish is faster and more flexible but may not provide the same level of security as other modern ciphers. The choice of cipher depends on the specific requirements of the application and the level of security needed.

1. What is the drawback of Double DES algorithm? How is it overcome by Triple DES?

The main drawback of Double Data Encryption Standard (DES) algorithm is its vulnerability to meet-in-the-middle attacks. In this attack, an attacker intercepts an encrypted message and applies a brute-force attack by trying all possible keys to decrypt the message. This process is done for each key in the first round of encryption, and then the results are encrypted again with all possible keys for the second round. The attacker then compares the results of both brute-force attacks to find the matching key pair. Triple DES (3DES) was introduced to overcome this vulnerability by increasing the key size and the number of encryption rounds. 3DES applies three rounds of encryption using three different keys, providing a much higher level of security than DES. In 3DES, the encryption and decryption process is as follows: 1. Encrypt with key 1 2. Decrypt with key 2 3. Encrypt with key 3 This process provides the equivalent of a 168-bit key length, which makes it much more difficult to crack through brute-force attacks. Additionally, 3DES is backward compatible with DES, which means it can be used to encrypt messages that are intended for DES, but with much stronger security. Overall, the use of 3DES provides a higher level of security than Double DES, while still maintaining backward compatibility with legacy systems that use DES

1. Explain avalanche effect and completeness effect.

Avalanche effect and completeness effect are two important characteristics of a cryptographic algorithm that determine the level of security provided by the algorithm.

Avalanche effect refers to the property of a cryptographic algorithm where a small change in the input or key results in a significant change in the output. In other words, a slight change in the plaintext or the key should cause the ciphertext to change drastically. This property ensures that the slightest modification in the input or key cannot be ignored, and the output is unpredictable. For example, if two plaintexts differ by only one bit, the corresponding ciphertexts should differ significantly.

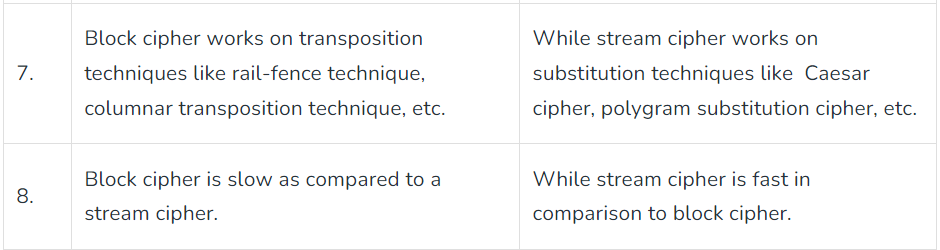
Completeness effect refers to the property of a cryptographic algorithm where each bit of the output depends on every bit of the input and key. In other words, the output should be a complete function of the input and the key, and no bit of the output can be predicted without knowing every bit of the input and the key. This property ensures that the entire input and key are taken into account when generating the output, making it difficult for an attacker to predict the output without knowing the entire input and key.

The avalanche effect and completeness effect are essential for ensuring the security of a cryptographic algorithm. If an algorithm does not exhibit these properties, it may be vulnerable to attacks such as differential cryptanalysis and linear cryptanalysis, which can compromise the confidentiality and integrity of the encrypted data.

Overall, the avalanche effect and completeness effect are important properties of a cryptographic algorithm that ensure that the slightest change in the input or key results in significant changes in the output, and the entire input and key are taken into account when generating the output. These properties ensure the security of the algorithm and make it difficult for attackers to compromise the encrypted data.

1. **Compare and contrast Block and stream cipher**

|  | **Block Cipher** | **Stream Cipher** |
| --- | --- | --- |
| Operation mode | Processes data in fixed-size blocks | Processes data one bit or byte at a time |
| Encryption speed | Slower due to processing data in fixed blocks | Faster due to processing data one bit or byte at a time |
| Key length | Longer key length required to maintain security | Shorter key length can provide sufficient security |
| Security | Considered more secure due to the complexity of the algorithm | Can be less secure than block ciphers due to their simplicity |
| Memory usage | Requires more memory to process fixed-size blocks | Requires less memory as it processes one bit or byte at a time |

****

1. **Key generation in Blowfish.**
2. **Briefly define the idea behind RSA and also explain.** 
   1. **What is one way function in this system?**
   2. **What is the trap door in this?**
   3. **Give public and private keys?**
   4. **Describe security in this system**

RSA (Rivest-Shamir-Adleman) is a widely used public-key cryptographic algorithm. It was introduced in 1977 by Ron Rivest, Adi Shamir, and Leonard Adleman. The idea behind RSA is based on the mathematical problem of factoring large numbers.

In RSA, two large prime numbers p and q are selected, and their product N = p \* q is calculated. N is a part of the public key, along with an integer e, which is selected such that it is coprime with (p-1) \* (q-1). The private key consists of the two prime numbers p and q and an integer d, which is the modular multiplicative inverse of e modulo (p-1) \* (q-1). In other words, (e \* d) % ((p-1) \* (q-1)) = 1.

To encrypt a message M using RSA, the sender computes C = M^e mod N and sends C to the receiver. To decrypt the ciphertext C, the receiver uses their private key by computing M = C^d mod N.

One-way function is a key concept in RSA. It refers to a function that is easy to compute in one direction but difficult to invert in the other direction. In RSA, the function used for encryption is a one-way function, making it easy for the sender to compute C from M, but difficult for an attacker to compute M from C without knowing the private key.

The trapdoor in RSA refers to the mathematical property that makes it easy to compute the private key from the public key, but difficult to compute the public key from the private key. This property is based on the fact that it is easy to compute the modular multiplicative inverse of e modulo (p-1) \* (q-1) if p and q are kept secret, but difficult to compute p and q from N.

The security of RSA depends on the difficulty of factoring large numbers. If an attacker can factor N into p and q, they can compute the private key and decrypt the ciphertext. Therefore, the security of RSA is based on the assumption that factoring large numbers is a computationally difficult problem.

In summary, RSA is a widely used public-key cryptographic algorithm based on the mathematical problem of factoring large numbers. It uses a one-way function for encryption, and the private key is calculated using the trapdoor property. The security of RSA depends on the difficulty of factoring large numbers, and the public and private keys are used for encryption and decryption, respectively.

1. Compare and contrast HMAC and CMAC.

| **Feature** | **HMAC** | **CMAC** |
| --- | --- | --- |
| Full form | Hash-based Message Authentication Code | Cipher-based Message Authentication Code |
| Type | Hash function-based MAC | Block cipher-based MAC |
| Input | Variable-length message | Fixed-length message |
| Key length | Any length | Same as block size |
| Hash function | Any secure hash function | None, uses block cipher |
| Security | Provides high security | Provides higher security than HMAC |
| Performance | Slower compared to CMAC | Faster compared to HMAC |
| Use cases | Used in IPsec, SSL/TLS, SSH, and other protocols | Used in AES-based encryption systems, disk encryption, and other applications |

1. SHA provides better security than MD. Justify.

The Secure Hash Algorithm (SHA) and Message Digest (MD) are two families of cryptographic hash functions that generate a fixed-size output from an input message. While both SHA and MD are widely used for data integrity, message authentication, and digital signatures, SHA is generally considered more secure than MD. Here's why:

Output Size: SHA has larger output sizes than MD, making it harder for attackers to find collisions or other vulnerabilities. SHA-256, for example, produces a 256-bit output, while MD5 produces only a 128-bit output.

Algorithmic Design: SHA is based on the Merkle-Damgård construction, which is more resistant to collision attacks than the MD algorithm. The MD algorithm, particularly MD5, has been shown to have vulnerabilities such as collisions and pre-image attacks.

NIST Certification: SHA is a family of hash functions certified by the National Institute of Standards and Technology (NIST) for use in cryptographic applications. This certification ensures that the algorithms have undergone rigorous testing and meet high standards for security.

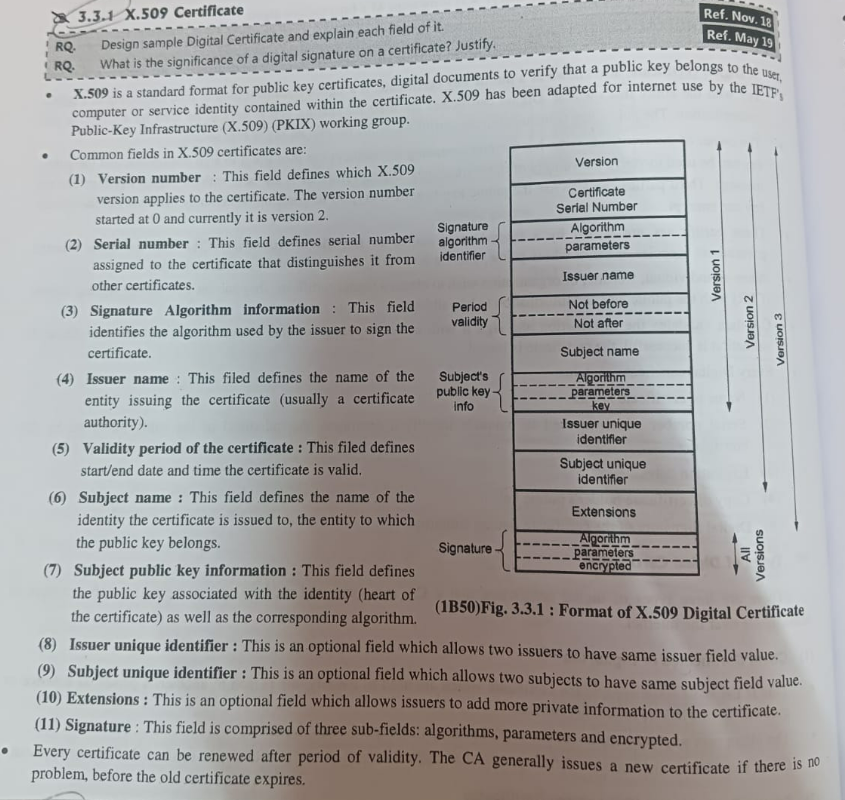
Resistance to Attacks: SHA is more resistant to various attacks, such as length extension attacks, birthday attacks, and chosen prefix attacks, compared to MD.

Popularity and Support: SHA is more widely used and supported by modern cryptographic libraries and applications, making it easier to implement and maintain secure systems.

In summary, SHA provides better security than MD because of its larger output size, algorithmic design, NIST certification, resistance to attacks, and popularity and support. Therefore, it is recommended to use SHA for new applications that require secure hash functions.

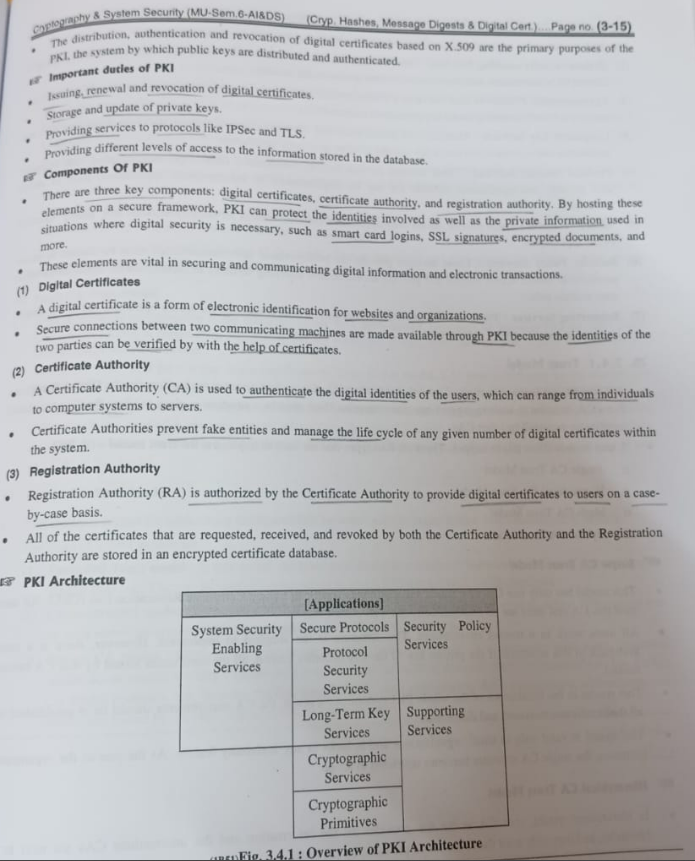
MD5 generates a message digest of 128-bits, while SHA1 generates a message digest of 160-bit hash value. Hence, SHA1 is a relatively complex algorithm and provides better security than MD5

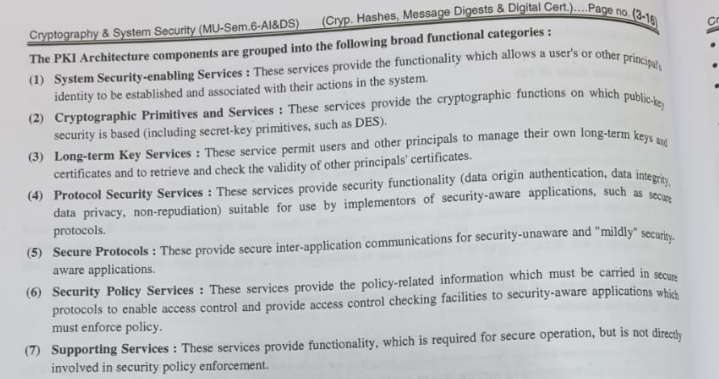
1. Design sample digital certificates and explain each field of it.



1. What is PKI? Explain different PKI architectures in detail.

It is the framework of encryption and cybersecurity that protects communication between the server and the client.





1. **What are the requirements of cryptographic hash functions? Compare MD5 and SHA hash functions. State real world applications of hash functions.**

The requirements of a cryptographic hash function are as follows:

Pre-image resistance: It should be computationally infeasible to find a message that produces a given hash value.

Second pre-image resistance: Given a message, it should be computationally infeasible to find a second message that produces the same hash value.

Collision resistance: It should be computationally infeasible to find two different messages that produce the same hash value.

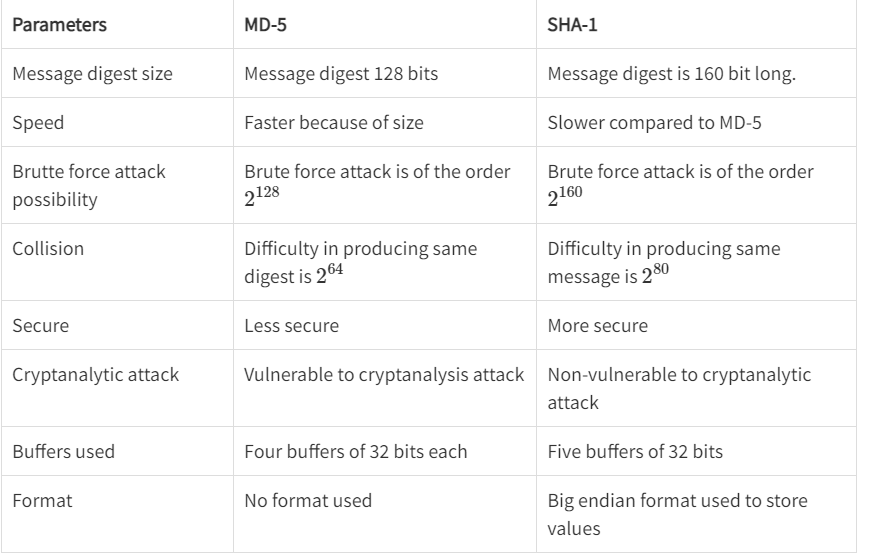
Determinism: Given the same input, it should always produce the same output.

Efficiency: It should be computationally efficient to compute the hash value for a given input.

Avalanche effect: A small change in the input should produce a significant change in the output.

Uniformity: The hash function should produce an output that is uniformly distributed across the entire range of possible hash values.

Non-reversibility: Given the hash value, it should be computationally infeasible to recover the original input.

****

Hash functions have a wide range of applications in various fields, including:

Digital signatures: Hash functions are used to ensure the integrity and authenticity of digital signatures. A hash function is applied to the message, and the resulting hash value is signed with a private key.

Password storage: Hash functions are used to store passwords securely. When a user creates a password, the hash function is applied to the password, and the resulting hash value is stored in the database instead of the plaintext password.

Message authentication codes: Hash functions are used to generate message authentication codes (MACs) that ensure the authenticity and integrity of messages transmitted over insecure channels.

Data integrity: Hash functions are used to ensure the integrity of data by generating a fixed-size hash value that uniquely identifies the data. This is used in file sharing protocols to verify that the downloaded file is the same as the original file.

Blockchain technology: Hash functions are used in blockchain technology to ensure the integrity and immutability of transactions. Each block in the blockchain contains a hash value of the previous block, making it impossible to modify a block without modifying the entire blockchain.

Anti-virus software: Hash functions are used to generate hash values for known viruses and malware, allowing anti-virus software to quickly identify and quarantine infected files.

Passwordless authentication: Hash functions are used to generate one-time passwords (OTPs) for passwordless authentication systems. A hash function is applied to a secret key and a timestamp to generate a unique OTP for each authentication attempt.

Overall, hash functions play a crucial role in ensuring the security, integrity, and authenticity of data in various real-world applications.

1. **Explain the operation of a secure hash algorithm on a 512 bit block.**

Secure Hash Algorithm (SHA) is a family of cryptographic hash functions that operate on a message input and produce a fixed-size output, typically 160 or 256 bits. Here is an overview of how SHA-512, which operates on a 512-bit block, works:

Initialization: SHA-512 starts by initializing eight 64-bit words, known as the "message schedule," and eight 64-bit words, known as the "hash values." The message schedule is initialized with constants and the hash values are initialized with the first 64 bits of the fractional parts of the square roots of the first eight prime numbers.

Padding: The input message is padded to a multiple of 1024 bits, with the padding consisting of a 1-bit followed by zeros and a 128-bit block at the end containing the length of the original message.

Processing: The padded message is processed in 1024-bit blocks. Each block is split into 16 64-bit words, and then 80 additional 64-bit words are generated using a series of operations that involve shifting, XORing, ANDing, and rotating the words.

Compression: The 80 words are then compressed using a series of rounds that involve combining the words in various ways and applying non-linear functions known as "round functions." Each round produces a new set of 80 words, which are used in the next round until all rounds have been completed.

Output: After processing all blocks, the final hash value is produced by concatenating the eight 64-bit hash values.

The resulting hash value is a fixed-size output that uniquely identifies the input message. It is important to note that even a small change in the input message will result in a completely different hash value, due to the avalanche effect and other cryptographic properties of the hash function. This makes hash functions like SHA-512 useful for data integrity, authentication, and other security applications.

1. **How does a digital certificate help to validate the authenticity of a user? Explain X.509 certificate format**

A digital certificate is a digital document that contains identifying information about a user or organization, including their name, public key, and digital signature. It is issued by a trusted third-party called a Certificate Authority (CA), who verifies the identity of the certificate holder before issuing the certificate.

When a user wants to authenticate themselves to another party, such as a website or server, they can present their digital certificate as proof of their identity. The other party can then use the public key contained in the certificate to encrypt a challenge message, which only the holder of the private key associated with the public key can decrypt and respond to. If the response is correct, the other party can be confident that the user is who they claim to be, because only someone in possession of the private key associated with the public key in the certificate could have generated the correct response.

In this way, a digital certificate helps to validate the authenticity of a user by providing a trusted means of verifying their identity. It also provides assurance that communications with the user are secure and cannot be intercepted or modified by an attacker

.

1. **What is Authentication? Explain Needham Schroeder Authentication protocol with suitable diagram.**

The term Needham–Schroeder protocol can refer to one of the two key transport protocols intended for use over an insecure network, both proposed by Roger Needham and Michael Schroeder. These are:

i. The Needham–Schroeder Symmetric Key Protocol is based on a symmetric encryption algorithm. It forms the basis for the Kerberos protocol. This protocol aims to establish a session key between two parties on a network, typically to protect further communication.

ii. The Needham–Schroeder Public-Key Protocol, based on public-key cryptography. This protocol is intended to provide mutual authentication between two parties communicating on a network, but in its proposed form is insecure.

The symmetric Protocol

Here, Alice (A) initiates the communication to Bob (B). S is a server trusted by both parties. In the communication:

i. A and B are identities of Alice and Bob respectively

ii. KAS is a symmetric key known only to A and S

iii. KBS is a symmetric key known only to B and S

iv. NA and NB are nonces generated by A and B respectively

v. KAB is a symmetric, generated key, which will be the session key of the session between A and B

The protocol can be specified as follows in security protocol notation:

A → S : A , B , NA

Alice sends a message to the server identifying herself and Bob, telling the server she wants to communicate with Bob.

S → A : { NA,KAB,B,KAB,AKBS}KAS

The server generates KAB

and sends back to Alice a copy encrypted under KBS

for Alice to forward to Bob and also a copy for Alice. Since Alice may be requesting keys for several different people, the nonce assures Alice that the message is fresh and that the server is replying to that particular message and the inclusion of Bob's name tells Alice who she is to share this key with.

A → B : {KAB,A}KBS

Alice forwards the key to Bob who can decrypt it with the key he shares with the server, thus authenticating the data.

B−→A{NB}KAB

Bob sends Alice a nonce encrypted under KAB to show that he has the key.

A → B : {NB–1}KAB

Alice performs a simple operation on the nonce, re-encrypts it and sends it back verifying that she is still alive and that she holds the key.

Attacks on the protocol

The protocol is vulnerable to a replay attack (as identified by Denning and Sacco). If an attacker uses an older, compromised value for KAB, he can then replay the message

{KAB,A}KBS

to Bob, who will accept it, being unable to tell that the key is not fresh.

Fixing the attack

This flaw is fixed in the Kerberos protocol by the inclusion of a timestamp. It can also be fixed with the use of nonces as described below.At the beginning of the protocol:

A → B : A

Alice sends to Bob a request.

B → A : {A, N′B

} KBS

Bob responds with a nonce encrypted under his key with the Server.

A → S : A,B,NA,{A,N′B}KBS

Alice sends a message to the server identifying herself and Bob, telling the server she wants to communicate with Bob.

S → A : {NA,KAB,B,KAB,A,N′BKBS}KAS

Note the inclusion of the nonce.

The protocol then continues as described through the final three steps as described in the original protocol above. Note that N’B is a different nonce from NB. The inclusion of this new nonce prevents the replaying of a compromised version of {KAB,A}KBS since such a message would need to be of the form {KAB,A,N′B}KBS

which the attacker can't forge since she does not have KBS

.

The Public-Key protocol

This assumes the use of a public-key encryption algorithm.

Here, Alice (A) and Bob (B) use a trusted server (S) to distribute public keys on request. These keys are:

i. KPA and KSA

, respectively public and private halves of an encryption key- pair belonging to A (S stands for "secret key" here)

ii. KPB andKSB

, similar belonging to B

iii. KPS

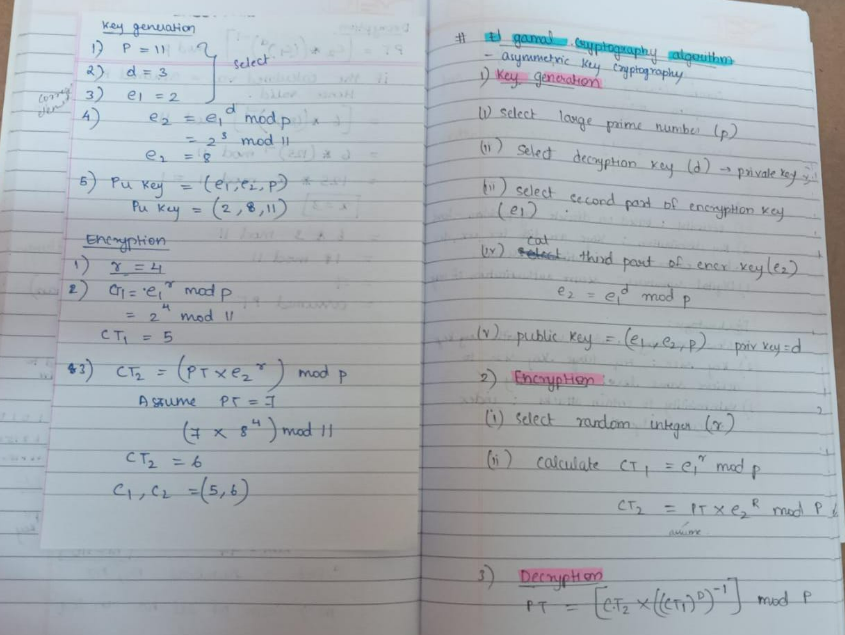
and KSS

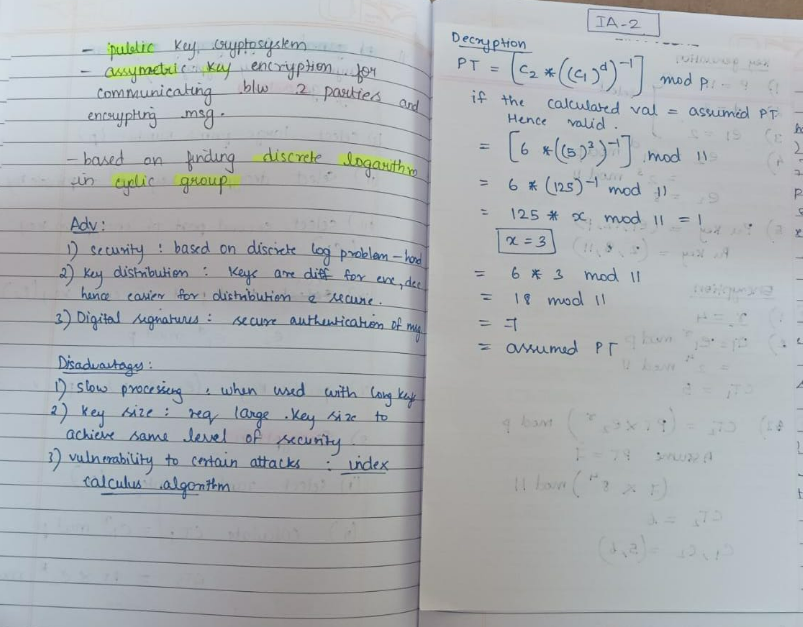
, similar belonging to S. (Note this has the property that KSS

is used to encrypt and KPS

to decrypt).

1. Two users wish to establish a secure communication channel and exchange a session key after mutual authentication. Show how this can be done with the help of a KDC.
2. **Short note on: El-Gamal Algorithm,**





1. **Why are digital certificates and signatures required? What is the role of digital signatures in digital certificates? Explain any one digital signature algorithm.**

Digital certificates and signatures are required to ensure the authenticity, integrity, and confidentiality of digital information and communications. In the digital world, it is essential to have a reliable way to identify users, verify their identities, and protect the data they transmit.

Digital certificates are used to verify the identity of the user or organization that holds the certificate. They contain identifying information, such as the user's name and public key, and are issued by trusted third-party Certificate Authorities (CAs). When a user presents their digital certificate to another party, that party can verify the certificate's authenticity and trust that the user is who they claim to be.

Digital signatures, on the other hand, are used to ensure the integrity of digital information. A digital signature is a mathematical scheme for verifying the authenticity of digital messages or documents. It is generated using a private key that is only known to the signer and can be verified using the signer's public key. By adding a digital signature to a document or message, the signer can prove that they created or approved the content, and that the content has not been tampered with since the signature was applied.

The role of digital signatures in digital certificates is to provide a means of verifying the authenticity of the information contained in the certificate itself. The digital signature on the certificate is generated using the private key of the issuing CA and can be verified using the CA's public key. By verifying the signature on the certificate, users can trust that the certificate is authentic and has not been tampered with since it was issued by the CA. This allows users to trust the identity of the certificate holder and to rely on the information contained in the certificate for secure communications and transactions.

1. **What is the need for message authentication? List various techniques used for message authentication. Explain any one of them.**

The need for message authentication arises from the fact that messages transmitted over a communication channel can be modified, replayed, or replaced by an attacker, leading to security threats such as unauthorized access, data tampering, and impersonation. Message authentication techniques aim to provide the recipient with a way to verify the integrity and authenticity of the message and ensure that it has not been modified or tampered with during transmission.

Various techniques are used for message authentication, including:

Message authentication codes (MACs): These are algorithms that generate a fixed-length tag or checksum that is appended to the message to verify its authenticity. The tag is generated using a secret key that is known only to the sender and receiver.

Hash functions: These are one-way functions that generate a fixed-length digest or hash value of the message. The digest is transmitted along with the message, and the receiver can compute the hash of the received message and compare it with the transmitted digest to verify the message's integrity.

Digital signatures: These are cryptographic schemes that use public key cryptography to provide authentication, integrity, and non-repudiation of the message. A digital signature is generated by the sender using their private key and can be verified using the sender's public key.

One commonly used message authentication technique is the HMAC (Hash-based Message Authentication Code). The HMAC algorithm uses a secret key and a cryptographic hash function to generate a message authentication code that can be used to verify the message's authenticity. The HMAC algorithm works as follows:

The message is hashed using a cryptographic hash function, such as SHA-256, to generate a hash value.

The hash value is then combined with a secret key using a key-based function to produce the HMAC.

The HMAC is appended to the message and transmitted to the receiver.

The receiver can then use the same secret key and hash function to compute the HMAC for the received message and compare it with the transmitted HMAC to verify the message's authenticity.

The HMAC algorithm provides several advantages, including resistance to known attacks, computational efficiency, and flexibility in the choice of hash function. It is widely used in network protocols, such as SSL/TLS, IPsec, and SSH, to provide message authentication and integrity.

1. **What is a nonce in the key distribution scenario? Explain the key distribution scenario if A wishes to establish logical connection with B. A and B both have a master key which they share with itself and key distribution center.**

A nonce is a random or semi-random number that is generated for a specific use. It is related to cryptographic communication and information technology (IT). The term stands for "number used once" or "number once" and is commonly referred to as a cryptographic nonce.

1. **What is the significance of dual signature?**

Dual signature is a type of digital signature scheme that involves the use of two or more signatures from different authorities or signers to authenticate a document or transaction. The significance of dual signature is that it provides an additional layer of security and trust in situations where a single signature may not be sufficient or when multiple parties need to authorize a transaction.

In dual signature schemes, two or more signers create separate digital signatures using their private keys, which are then combined or concatenated to create a single signature for the document or transaction. This combined signature can be verified using the corresponding public keys of the signers.

The use of dual signature has several advantages, including:

Increased security: Dual signature provides additional security and protection against unauthorized or fraudulent transactions. It ensures that both signers have authorized the transaction and reduces the risk of one signer acting alone.

Increased trust: The use of dual signature can increase trust and confidence in transactions, particularly in situations where there is a high risk of fraud or where large amounts of money or sensitive information are involved.

Legal compliance: Dual signature may be required by law or regulatory requirements in certain industries or jurisdictions.

Dual signature schemes are commonly used in financial transactions, such as wire transfers, loan agreements, and contracts, as well as in other industries that require high levels of security and trust.

1. **Explain handshake protocol in SSL. What is the need of SSL?**

The SSL handshake is the initial process that occurs when a client, such as a web browser, connects to a web server over SSL/TLS. The SSL handshake involves a series of steps to establish a secure and encrypted connection between the client and server.

Here is a brief overview of the steps involved in the SSL handshake:

ClientHello: The SSL handshake begins when the client sends a ClientHello message to the server. The message includes information about the SSL/TLS version supported by the client, a random number, and a list of supported cipher suites.

ServerHello: The server responds with a ServerHello message, which includes the SSL/TLS version that will be used for the connection, a random number, and the chosen cipher suite.

SSL/TLS Certificate: The server sends its SSL/TLS certificate to the client, which includes the server's public key and other identifying information. The certificate is verified to ensure that it is issued by a trusted certificate authority (CA).

ClientKeyExchange: The client sends a message to the server containing a pre-master secret key, which is encrypted using the server's public key from the SSL/TLS certificate.

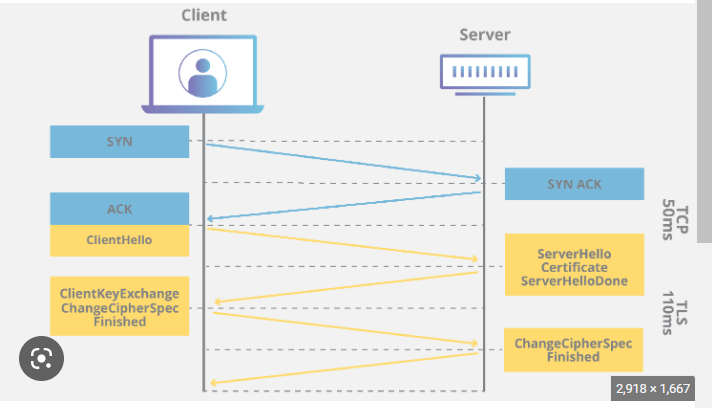
ServerKeyExchange: Depending on the chosen cipher suite, the server may send a ServerKeyExchange message to the client, which includes additional cryptographic parameters required for the key exchange.

CertificateRequest: If the server requires the client to authenticate itself, it sends a CertificateRequest message, requesting the client's SSL/TLS certificate.

ServerHelloDone: The server sends a ServerHelloDone message to indicate that the SSL/TLS handshake is complete.

CertificateVerify: If the client has provided an SSL/TLS certificate, it sends a CertificateVerify message to prove the authenticity of the certificate.

Finished: Finally, the client and server exchange Finished messages to verify that the SSL/TLS connection has been established successfully and that the shared secret key is correct.



Once the SSL handshake is complete, the client and server can begin transmitting encrypted data over the SSL/TLS connection. The SSL handshake is a critical part of the SSL/TLS protocol, as it establishes the foundation for secure and encrypted communication between the client and server.

The need for SSL (Secure Sockets Layer) arises from the need for secure communication over the internet. SSL is a protocol that provides secure communication between two parties by encrypting the data being transmitted. Here are some reasons why SSL is necessary:

Confidentiality: SSL provides confidentiality by encrypting the data being transmitted between two parties, making it unreadable to anyone who intercepts the data.

Integrity: SSL ensures that the data being transmitted is not tampered with during transmission, by using digital signatures and message authentication codes.

Authentication: SSL provides authentication by verifying the identity of the parties involved in the communication. This is done using digital certificates and a public key infrastructure.

Trust: SSL creates a level of trust between the parties involved in the communication by providing assurance that the data being transmitted is secure and the parties involved are who they claim to be.

1. **What are different types of firewall?**

A firewall is a network security system that monitors and controls incoming and outgoing network traffic based on a set of predefined security rules. The primary function of a firewall is to act as a barrier between a private internal network and the public Internet, preventing unauthorised access to the internal network and protecting it from potential attacks.

There are several types of firewalls, including:

Packet filtering firewalls: These firewalls inspect packets of data as they pass through a network, and allow or block traffic based on a set of predefined rules. Packet filtering firewalls are the simplest type of firewall and are often used in routers and switches.

Stateful inspection firewalls: These firewalls not only inspect packets of data, but also keep track of the state of network connections. This allows them to identify and block unauthorised traffic that may be disguised as legitimate traffic.

Application-level firewalls: These firewalls operate at the application layer of the network stack, and can inspect the contents of network traffic to ensure that only valid traffic is allowed through. Application-level firewalls are often used to protect web servers and other application servers.

Next-generation firewalls: These firewalls combine the features of other firewall types, including packet filtering, stateful inspection, and application-level filtering, with additional capabilities such as intrusion detection and prevention, deep packet inspection, and user identification.

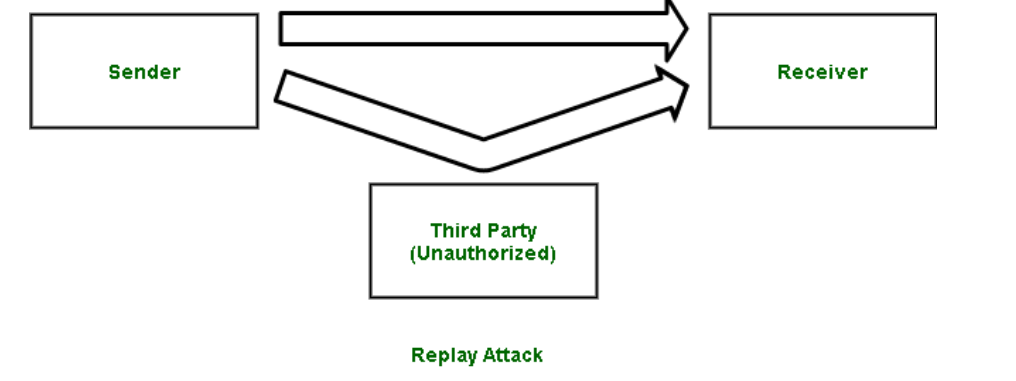
Host-based firewalls: These firewalls are software-based and run on individual computers or devices. They can be configured to allow or block specific types of network traffic, and are often used to protect individual computers or servers.

Firewalls can be hardware-based or software-based, and can be deployed at different points in a network, including at the network perimeter, on individual devices, and in the cloud. The choice of firewall type and deployment depends on the specific security needs of the network and the applications it supports.

1. **Give examples of replay attacks. List three general approaches for dealing with replay attacks.**

Replay Attack is a type of security attack to the data sent over a network.

In this attack, the hacker or any person with unauthorized access, captures the traffic and sends communication to its original destination, acting as the original sender. The receiver feels that it is an authenticated message but it is actually the message sent by the attacker. The main feature of the Replay Attack is that the client would receive the message twice, hence the name, Replay Attack.



Prevention from Replay Attack :

Timestamp method –

Prevention from such attackers is possible, if timestamp is used along with the data. Supposedly, the timestamp on a data is more than a certain limit, it can be discarded, and sender can be asked to send the data again.

Session key method –

Another way of prevention, is by using session key. This key can be used only once (by sender and receiver) per transaction, and cannot be reused.

Nonces: Nonces are unique numbers used only once in a given context. They can be included in messages to ensure that each message is unique and has not been replayed. If a message is received with a nonce that has already been used, it can be rejected as a replay attack.

Sequence numbers: Sequence numbers can be used to ensure that messages are delivered only once and in the correct order. If a message is received with a sequence number that is lower than the last message received, it can be rejected as a replay attack.

1. **How can we achieve web security? Explain with examples.**

Updated Software: You need to always update your software. Hackers may be aware of vulnerabilities in certain software, which are sometimes caused by bugs and can be used to damage your computer system and steal personal data. Older versions of software can become a gateway for hackers to enter your network. Software makers soon become aware of these vulnerabilities and will fix vulnerable or exposed areas. That’s why It is mandatory to keep your software updated, It plays an important role in keeping your personal data secure.

Beware of SQL Injection: SQL Injection is an attempt to manipulate your data or your database by inserting a rough code into your query. For e.g. somebody can send a query to your website and this query can be a rough code while it gets executed it can be used to manipulate your database such as change tables, modify or delete data or it can retrieve important information also so, one should be aware of the SQL injection attack.

Cross-Site Scripting (XSS): XSS allows the attackers to insert client-side script into web pages. E.g. Submission of forms. It is a term used to describe a class of attacks that allow an attacker to inject client-side scripts into other users’ browsers through a website. As the injected code enters the browser from the site, the code is reliable and can do things like sending the user’s site authorization cookie to the attacker.

Error Messages: You need to be very careful about error messages which are generated to give the information to the users while users access the website and some error messages are generated due to one or another reason and you should be very careful while providing the information to the users. For e.g. login attempt – If the user fails to login the error message should not let the user know which field is incorrect: Username or Password.

Data Validation: Data validation is the proper testing of any input supplied by the user or application. It prevents improperly created data from entering the information system. Validation of data should be performed on both server-side and client-side. If we perform data validation on both sides that will give us the authentication. Data validation should occur when data is received from an outside party, especially if the data is from untrusted sources.

Password: Password provides the first line of defense against unauthorised access to your device and personal information. It is necessary to use a strong password. Hackers in many cases use sophisticated software that uses brute force to crack passwords. Passwords must be complex to protect against brute force. It is good to enforce password requirements such as a minimum of eight characters long including uppercase letters, lowercase letters, special characters, and numerals.

1. **Consider a scenario where an intruder wants to access some valuable info from an ongoing communication. What security services should be implemented and which mechanism can be used to achieve those security services.**
2. **What are the different threats to emails? Give an algorithm to secure emails being sent from user A to user B.**

There are several types of email attacks that are commonly used by cybercriminals to target individuals and organisations. Here are some of the most common types:

Phishing: Phishing attacks involve sending fake emails that appear to be from a legitimate source, such as a bank or an online retailer, in order to trick the recipient into providing sensitive information, such as passwords, credit card numbers, or social security numbers. These emails often contain links to fake websites that look like the real thing, but are designed to steal the victim's information.

Spear phishing: Spear phishing attacks are more targeted than regular phishing attacks, and are designed to trick a specific individual or organization into providing sensitive information. These attacks often use personal information or details about the victim's job or organization to make the email appear more legitimate.

Malware: Malware attacks involve sending emails that contain malicious attachments or links to infected websites. When the recipient clicks on the link or opens the attachment, their computer or network can become infected with viruses, spyware, or other types of malware.

Man-in-the-middle (MITM) attacks: MITM attacks involve intercepting email traffic and redirecting it to a malicious server, where the attacker can intercept and read the messages. This allows the attacker to steal sensitive information, such as login credentials or financial data.

Email spoofing: Email spoofing involves creating fake emails that appear to be from a legitimate source, such as a bank or an online retailer. These emails often contain links to fake websites or malware, and are designed to trick the recipient into providing sensitive information.

Business Email Compromise (BEC): BEC attacks involve impersonating an executive or employee of a company and requesting sensitive information or funds from other employees or vendors. These attacks can be very convincing, and often involve extensive research and social engineering.

To protect against email attacks, individuals and organisations can take several measures, such as:

Using anti-spam and anti-malware software to filter out suspicious emails.

Educating employees about how to recognize and report suspicious emails.

Enforcing strong password policies and using two-factor authentication.

Verifying the authenticity of emails before clicking on links or opening attachments.

Encrypting sensitive email traffic to prevent interception or snooping.

1. **Short note on ARP Spoofing,Session Hijacking ,eavesdropping,.**
2. **A user wishes to do online transactions with Amazon.com Discuss a protocol which can be used to set up a secure communication channel and provide server side and client side authentication. Show the steps involved in the handshake process.**
3. **Explain briefly: Session hijacking, Salami attack, SQL injection.**

Session hijacking is a type of attack that targets web applications by stealing the session token of an authenticated user and using it to impersonate the user on the website. The attacker can then carry out actions on behalf of the user, such as changing the user's password, transferring funds, or accessing sensitive information.

Here's how session hijacking works:

Suppose a user logs into a web application, such as an online banking website, and receives a session token from the server.

The session token is usually stored in a cookie or a URL parameter, and is used by the server to authenticate the user's subsequent requests.

Meanwhile, an attacker intercepts the session token, either by eavesdropping on the user's network traffic or by stealing the token from the user's browser.

The attacker then uses the session token to impersonate the user on the web application, without needing to know the user's login credentials.

To prevent session hijacking attacks, web developers can implement a number of security measures, such as using HTTPS to encrypt network traffic, using secure cookies that are not vulnerable to cookie stealing, and using session timeouts that invalidate inactive sessions after a period of time.

Session management is also an important aspect of web application security, as it involves managing the user's session state and ensuring that the session data is secure and protected. This includes measures such as using secure session storage mechanisms, implementing proper access controls, and regularly monitoring and auditing session activity.

Users can also protect themselves against session hijacking by using strong and unique passwords, logging out of web applications after use, and using a virtual private network (VPN) or other secure network to access web applications.

1. **Consider a voter data management system in an E-voting system with sensitive and non-sensitive attributes. Show with sample queries how attacks(Direct, Inference) are possible on such data sets. Suggest 2 different ways to mitigate the problem. (5,6)**
2. **List various Software vulnerabilities. How vulnerabilities are exploited to launch an attack.**

There are various types of software vulnerabilities, some of the common ones are:

Buffer overflow: A buffer overflow occurs when a program tries to store more data in a buffer than it was designed to hold. Attackers can exploit this vulnerability by inserting malicious code into the extra data, which can cause the program to crash or execute the attacker's code.

Cross-site scripting (XSS): XSS attacks occur when an attacker injects malicious code into a website, which can then be executed by unsuspecting users who visit the site. This type of attack can be used to steal user data, such as passwords or credit card numbers.

SQL injection: SQL injection attacks occur when an attacker inserts malicious code into a website's SQL database, which can allow the attacker to access sensitive data or modify the database.

Remote code execution: Remote code execution (RCE) vulnerabilities allow attackers to execute code on a victim's computer or server, which can give them access to sensitive data or control of the system.

Denial of Service (DoS): A DoS attack occurs when an attacker floods a website or server with traffic, which can cause it to crash or become unavailable to legitimate users.

These vulnerabilities can be exploited by attackers using various techniques such as:

Exploit kits: Exploit kits are pre-packaged software tools that attackers can use to exploit vulnerabilities in a target system. These kits are often sold on the dark web and can be used by attackers with little technical knowledge.

Phishing: Phishing attacks use social engineering techniques to trick users into clicking on a malicious link or opening a malicious attachment. Once the user does so, the attacker can exploit vulnerabilities on the user's computer or server.

Malware: Malware is malicious software that is designed to exploit vulnerabilities on a victim's computer or server. Once installed, the malware can be used to steal data or launch further attacks.

To prevent these attacks, software developers must follow secure coding practices and regularly update their software to patch vulnerabilities. Users can also protect themselves by keeping their software up to date and using antivirus software to detect and block malware. Additionally, network security measures such as firewalls and intrusion detection systems can help prevent attacks by detecting and blocking suspicious activity.

1. **What are the various ways for memory and address protection in Operating Systems?**

The Fence method is a memory protection technique that involves placing a "fence" around the memory allocated to a process. The fence consists of a special value or pattern that is placed at the beginning and end of each memory allocation.

When the process accesses the memory, the operating system checks the fence values to ensure that the process is not accessing memory beyond its allocated region. If the process attempts to access memory beyond its allocated region, the fence value will not match, and the operating system will terminate the process.

The Fence method is often used in embedded systems and real-time operating systems, where memory protection is critical. It is relatively simple to implement and can detect most types of memory access violations, including buffer overflows and underflows.

However, the Fence method has some limitations. It cannot prevent all types of memory-related vulnerabilities, and it can be circumvented by attackers who are able to modify the fence values in memory. Therefore, it is usually used in combination with other memory protection techniques, such as ASLR and virtual memory, for more comprehensive protection.

The relocation method is a technique used for address protection and memory protection in operating systems. The relocation method involves using a base address register (BAR) to relocate a program's memory address space.

When a program is loaded into memory, the operating system allocates a certain memory address space to it. The program is compiled with a starting address in mind, but that address is only valid if the program is loaded into memory at that address. However, due to the dynamic nature of the operating system, it is not possible to guarantee that the program will always be loaded into memory at the same address.

This is where the relocation method comes in. The program's starting address is defined as a relative offset from a base address. The base address register (BAR) contains the actual starting address where the program is loaded in memory. When the program is executed, the processor adds the relative offset to the base address in the BAR, resulting in the actual memory address where the instruction is located.

The relocation method provides address protection, as it allows the program to be loaded into any available memory address space, while still ensuring that the program is executed correctly. It also provides memory protection, as the program can only access memory locations that are within its allocated address space.

Overall, the relocation method is a powerful technique for ensuring both address protection and memory protection in operating systems.

The Bound Registers method is a technique used for memory protection in operating systems. It involves using special registers called Bound Registers to define the upper and lower bounds of a program's memory access.

When a program is loaded into memory, the Bound Registers are set to the upper and lower limits of the program's memory access. These registers are then checked every time the program attempts to access memory. If the program attempts to access memory outside the bounds defined by the Bound Registers, an exception is raised, and the program is terminated.

The Bound Registers method provides an effective way to prevent buffer overflow attacks, as it ensures that a program can only access memory within its allocated memory space. It also provides protection against malicious code injection, as the program cannot execute code outside of its allocated memory space.

However, the Bound Registers method has some limitations. It does not protect against attacks that exploit vulnerabilities in the program itself, such as code injection attacks that modify the program's instructions to bypass the Bound Registers. It is also limited to protecting memory access within a single process, and it does not provide protection for inter-process communication or shared memory.

Overall, the Bound Registers method is a useful technique for memory protection in operating systems, but it should be used in conjunction with other techniques for comprehensive memory protection.

The Segmentation method is a technique used for memory protection in operating systems. It involves dividing the memory space of a process into segments, with each segment representing a logical unit of the process's memory usage.

Each segment is assigned a specific base address and a limit, which defines the range of memory locations that the segment can access. The processor checks these limits each time a memory access instruction is executed. If the instruction attempts to access memory outside the limits of the segment, an exception is raised, and the process is terminated.

The Segmentation method provides a flexible way to manage a process's memory usage, as it allows the memory space to be divided into segments based on the process's needs. It also provides a way to protect against buffer overflow attacks, as each segment has its own limit that prevents the process from accessing memory outside of its allocated range.

However, the Segmentation method has some limitations. It does not provide protection against attacks that exploit vulnerabilities within the process itself, such as code injection attacks that modify the process's instructions to bypass the segment limits. It can also be difficult to manage, as the number and size of segments required by a process can vary depending on its behavior.

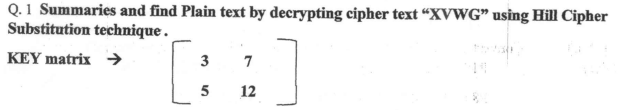
The Paging method is a technique used for memory protection and virtual memory management in operating systems. It involves dividing the physical memory of a system into fixed-size blocks called pages, and dividing the logical memory space of a process into the same size blocks called page frames.

Each page frame is mapped to a physical page in memory through a page table, which contains information about the page's location and permissions. When a process attempts to access a memory location, the processor translates the logical address to a physical address by looking up the page table.

The Paging method provides several benefits, including memory protection and virtual memory management. Memory protection is achieved by assigning each page a set of permissions, such as read, write, or execute, which the operating system enforces through the page table. This ensures that a process cannot access memory outside of its allocated page frames.

Virtual memory management is achieved by allowing the operating system to move pages in and out of physical memory as needed, known as swapping or paging. This allows the system to manage memory more efficiently, as it can allocate memory on demand and free up memory when it is not needed. It also allows multiple processes to share the same physical memory without interfering with each other.

However, the Paging method has some limitations. It can incur overhead due to the need to constantly translate logical addresses to physical addresses through the page table. It can also lead to performance issues if the system is constantly swapping pages in and out of memory. Additionally, it can be vulnerable to certain types of attacks, such as side-channel attacks that exploit patterns in the page table to infer information about the process's memory usage.

1. **Short note on: Phishing attack, Cross-site scripting attack.**
2. If A and B wish to use RSA to communicate securely. A chooses public key (e,n) as (7,247) and B chooses public key(e,n) as (5,221)
   1. Calculate A’s Private key.
   2. Calculate B’s Private key.
   3. What will be the cipher text sent by A to B, if A wishes to send M=5 to B.
   4. Show the message signing and verification using RSA digital signature.
3. If a generator g=2 and n or p=11, using diffie hellman, solve:
   1. Show that 2 is the primitive root of 11.
   2. If A has public key 9, what is private key?
   3. If B has public key 3, what is the private key?
   4. Calculate shared secret key
4. Encrypt “academic committee will meet today” using Playfair cipher with keyword “ROYAL ENFIELD”.
5. 
6. Use hill cipher to encrypt the text ‘short’. The key to be used is ‘hill’.
7. Elaborate the steps of key generation using the RSA algorithm. In the RSA system, the public key (E,N) of user A is defined as (7,187). Calculate phi(N) and private key ‘D’. What is the cipher text for M=10 using the public key.
8. Encrypt “This is the final exam” with the Playfair cipher using the key “Guidance”. Explain the steps involved.
9. Given modulus n=91 and public key, e=5, find the values of p,q, phi(n), and d using RSA. Encrypt M=25. Also perform decryption.
10. Users A and B use the Diffie Hellman technique with a common prime 71 and primitive root 7. Show that 7 is the primitive root of 71. If user A has private key x=5, what is A’s Public key R1? If user B has private key y=12,what is B’s public key R2? What is a shared secret key?
11. Encrypt the plaintext message “SECURITY” using affine cipher with key pair(3,7). Decrypt to get back the original plaintext.
12. Use the playfair cipher with the keyword: “HEALTH” to encipher the message “Life is full of Surprises”.
13. Encrypt the given message using Autokey Cipher, key=7 and the message is: “The house is being sold tonight”.