What is digital signature ? briefly state the steps in producing digital signatures?

A **digital signature** is a cryptographic technique used to verify the authenticity and integrity of a digital message or document. It ensures that the message has not been altered and that it comes from a legitimate sender.

Steps in Producing a Digital Signature:

1. **Message Hashing**: The sender creates a hash (a unique, fixed-size string of characters) of the original message using a hashing algorithm like SHA-256.
2. **Hash Encryption**: The hash is then encrypted using the sender's private key. This encrypted hash is the actual digital signature.
3. **Signature Attachment**: The digital signature is attached to the original message and sent to the recipient.
4. **Signature Verification**: Upon receiving the message, the recipient decrypts the signature using the sender's public key, revealing the original hash.
5. **Hash Comparison**: The recipient hashes the received message and compares it to the decrypted hash. If they match, the message is verified as authentic and unaltered.

Calculating Resources:

Online Matrix Multiplication : Matrix Calculator

Online Inverse Matrix : Inverse Matrix Calculator

Online Modulo Converter : Modulo Calculator

Learning Resources :

Playfair Cipher with Examples - GeeksforGeeks

ElGamal Encryption Algorithm - GeeksforGeeks

Nonrepudiation means a user cannot deny (repudiate) having performed a transaction. It combines authentication and integrity: nonrepudiation authenticates the identity of a user who performs a transaction and ensures the integrity of that transaction.

Open Design Principle

The Open Design Principle argues that the security of a system should not depend on the secrecy of its design or implementation. Instead, it should rely on well-established, robust, and transparent security mechanisms. Even if an attacker knows all the details of the system's design, they should not be able to compromise its security if the design is sound.

Security by Obscurity

Security by obscurity is a principle where the security of a system relies on keeping the internal workings, configurations, or vulnerabilities of the system secret. In this approach, if attackers don't know how the system operates, it is believed that they will be less able to exploit it.

OTP is a type of encryption that is theoretically unbreakable when used correctly. It includes a random key name pad which is same size of the actual message . this pad is used only once to encrypt or decrypt a message

A dictionary attack is a type of brute-force attack used to crack passwords or decrypt encrypted data by systematically trying every word in a predefined list, called a "dictionary."

A **Message Authentication Code (MAC)** is a cryptographic technique used to ensure both the **integrity** and **authenticity** of a message. It is a small piece of data, typically a hash value, that is computed from the original message using a secret key. The MAC is sent along with the message, and the recipient uses the same key to recompute the MAC to verify the message has not been altered.

How MAC Works:

1. **Message and Secret Key**: A MAC function takes two inputs: the original message (or data) and a secret key that is shared between the sender and the receiver.
2. **Generating the MAC**: The sender runs the message and the secret key through the MAC algorithm, which produces a fixed-length output called the MAC tag. This tag is sent along with the original message to the receiver.
3. **Verifying the MAC**: Upon receiving the message and the MAC tag, the receiver runs the same MAC algorithm using the received message and the secret key to compute a new MAC tag. The receiver compares this newly computed MAC with the one received.
4. **Integrity and Authenticity Check**:
    - If the two MACs match, the receiver can trust that the message was not altered during transmission (integrity).
    - Additionally, since only the sender and receiver know the secret key, a matching MAC indicates that the message is from an authenticated source (authenticity).

Pretexting:

Description: In pretexting, an attacker creates a fake scenario (pretext) to obtain information from a victim. This often involves impersonating someone in a position of authority or trust to gain sensitive details.
Example: An attacker poses as an IT support technician and calls an employee, claiming they need the employee's login credentials to fix a system issue.

Quid Pro Quo:

Description: The attacker offers something in return for information or access. This often involves impersonating technical support or offering fake assistance.
Example: A scammer pretends to be tech support and offers to fix a computer problem in exchange for the user's login credentials or remote access to the system.

The **Diffie-Hellman key exchange** is a cryptographic protocol that allows two parties to establish a shared secret key over an insecure communication channel. This shared key can then be used for encrypting and decrypting messages between the parties, ensuring secure communication.

How the Protocol Works

1. **Public Parameters:**
    - Both parties agree on two public parameters:
        - A large prime number ppp (modulus).
        - A generator ggg (a number less than ppp, with special mathematical properties).
2. **Private Keys:**
    - Each party selects a private key:
        - Alice selects a private key aaa.
        - Bob selects a private key bbb.
    - These private keys are kept secret.
3. **Public Keys:**
    - Each party calculates their public key using the generator ggg raised to the power of their private key, modulo ppp:
        - Alice computes A=gamod pA = g^a \mod pA=gamodp.
        - Bob computes B=gbmod pB = g^b \mod pB=gbmodp.
    - They then exchange their public keys AAA and BBB over the insecure channel.
4. **Shared Secret:**
    - Both parties compute the shared secret key using the other party's public key and their own private key:
        - Alice computes s=Bamod ps = B^a \mod ps=Bamodp.

        - Bob computes s=Abmod ps = A^b \mod ps=Abmodp.
    - Since B=gbB = g^bB=gb and A=gaA = g^aA=ga, both computations yield the same result:
        - s=gabmod ps = g^{ab} \mod ps=gabmodp.

The shared secret sss is now known to both Alice and Bob, but an eavesdropper who intercepts the public keys AAA and BBB would not be able to compute sss without knowing the private keys aaa or bbb, due to the difficulty of solving the **discrete logarithm problem**.

$\vec{C} = K \cdot \vec{x}$

$\vec{x} = K^{-1} \cdot \vec{C}$

Cipher Text : TQX

$$K = \begin{bmatrix} 6 & 24 & 1 \\ 13 & 16 & 10 \\ 20 & 17 & 15 \end{bmatrix}$$

$$\vec{C} = \begin{bmatrix} 19 \\ 16 \\ 23 \end{bmatrix}$$

Text, $\therefore \vec{x} = K^{-1} \cdot \vec{C}$

$$= \begin{bmatrix} 0.16 & -0.78 & 0.51 \\ 0.01 & 0.16 & -0.11 \\ -0.224 & 0.86 & -0.49 \end{bmatrix} \begin{bmatrix} 19 \\ 16 \\ 23 \end{bmatrix}$$

$$= \begin{pmatrix} 2.254 \\ 0.3 \\ -1.8 \end{pmatrix} \simeq \begin{bmatrix} 2 \\ 0 \\ -2 \end{bmatrix} \text{ mod } 26$$

$$= \begin{bmatrix} 2 \\ 0 \\ 24 \end{bmatrix}$$

$\therefore \vec{x} = CAY$