
Mastering Cryptography:
Foundations & Applications
by HIMANSHU GOHITE

Instructor

Himanshu Gohite

Security Enthusiast

Brief Intro:

- **CompTIA Security+**
- **NPTEL Cryptography and Network Security**
- **MP FITT IITD Blockchain Builder Certificate**

Course Structure

- **Day 1: Introduction to Cryptography**
- **Overview of Cryptography**
  - What is Cryptography? (History and Evolution)
  - Importance in Cybersecurity
  - Types of Ciphers
- **Classical Ciphers**
  - Basic Cryptographic Algorithms (Caesar Cipher, Playfair Cipher)
  - Cryptanalysis: Breaking Simple Ciphers
- **Modern Ciphers**
  - Types of Ciphers
  - Symmetric Cipher

- **Day 2: Cryptography and Data Security**
- **Overview of Data Security**
  - What is Data Security?

- **States of Data**
  - Data at Rest,
  - Data in Transition
  - Data in Use

- **Hashing**
  - What is Hashing
  - Types
  - Applications

by HIMANSHU GOHITE

- **Day 3: Understanding Symmetric Key Cryptography**
- **Data Encryption Standard (DES)**
  - Working
  - Applications and Vulnerabilities
- **Advanced Encryption Standard**
  - Working
  - Applications and Vulnerabilities
- **Block Cipher Modes of Operations**
  - What are modes of operations
  - Types, Applications, Pros & Cons
- **Message Authentication Code (MAC)**

by HIMANSHU GOHITE

- **Day 4: Understanding Asymmetric Key Cryptography**
- Overview of Asymmetric Cryptography
  - Working & Applications
- Diffie-Hellman Key Exchange
  - Working
  - Application and Vulnerabilities
- RSA Encryption
  - Working
  - Application and Vulnerabilities

- Digital Signatures
  - Working
  - Application

by HIMANSHU GOHITE

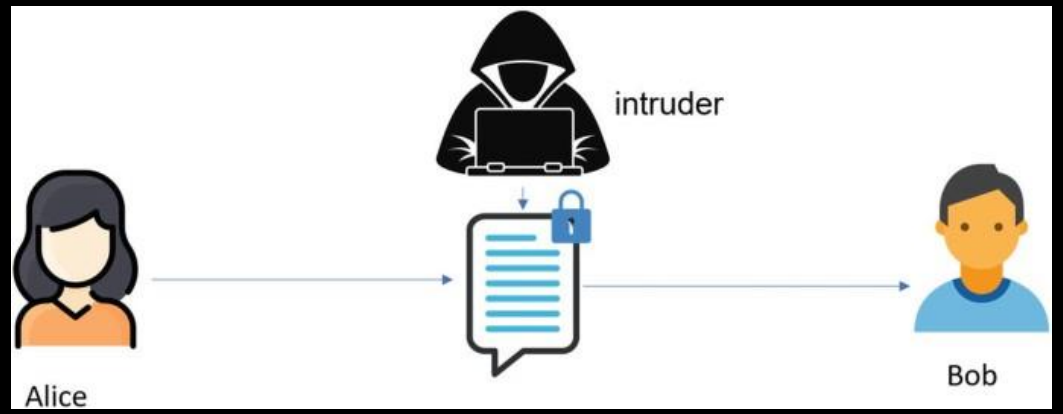
- **Day 5: Understanding Obfuscation**
- **Overview of Obfuscation**
  - Definition and Techniques
- **Steganography**
  - Working
  - Application and Vulnerabilities
- **Tokenization**
  - Working
  - Application and Vulnerabilities
- **Data Masking**
  - Working
  - Application

Day 1 : Agenda

- **Day 1: Introduction to Cryptography**
- **Overview of Cryptography**
  - What is Cryptography? (History and Evolution)
  - Importance in Cybersecurity
  - Types of Ciphers
- **Classical Ciphers**
  - Basic Cryptographic Algorithms (Caesar Cipher, Playfair Cipher)
  - Vulnerabilities of classical Ciphers
- **Modern Ciphers**
  - Types of Ciphers
  - Symmetric Cipher

Overview of Cryptography

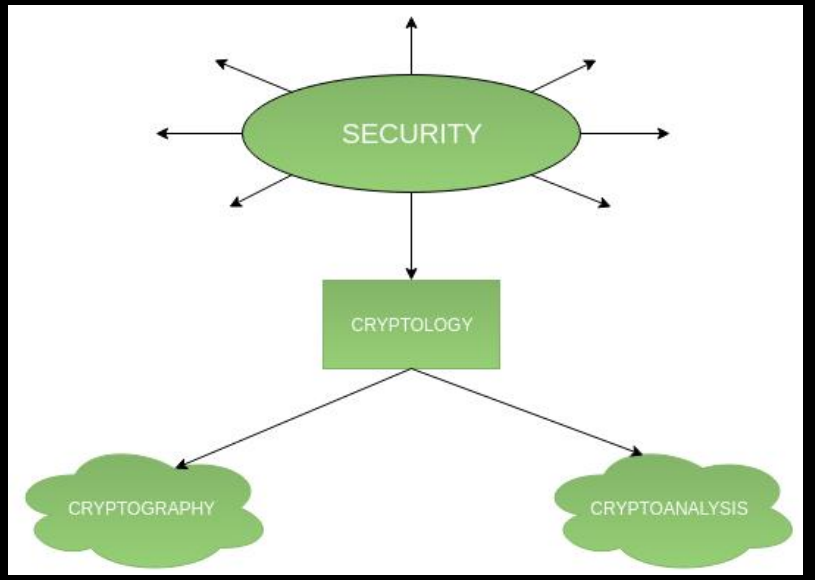
- Cryptography, or cryptology (from Ancient Greek : kryptós "hidden, secret" and graphein, "to write", "study").



- The main objective of cryptography is to enable two people to communicate over an insecure channel/medium in securely such that any third person can not understand what is being said.

Cryptology: How Cryptography and Cryptanalysis Work Together

- Cryptology = Cryptgraphy + Cryptanalysis
- **Cryptology**: The science of secure communication, encompassing both the creation and breaking of cryptographic systems.
- **Cryptography**: The practice of designing techniques and algorithms to secure data and communications.
- **Cryptanalysis**: The study and techniques used to break or analyze cryptographic systems without access to the secret key.



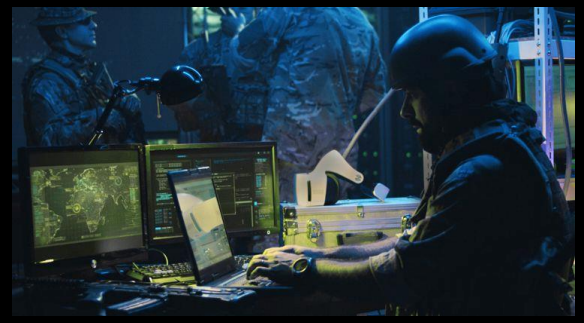
by HIMANSHU GOHITE

Functions

- There are 5 main functions of cryptography:-
- **Confidentiality**: Ensures that only the intended recipient can read the message.
- **Integrity**: Assures the recipient that the message hasn't been altered during transmission.
- **Authentication**: Proves the identity of the sender or receiver.
- **Non-repudiation**: Prevents the sender from denying that they sent the message.
- **Key Exchange**: Securely shares cryptographic keys between the sender and receiver.

Applications

- Practical applications of cryptography include :
  - **Electronic commerce**
  - **Chip-based payment cards**
  - **Digital currencies**
  - **Computer passwords**
  - **Military communications**



by HIMANSHU GOHITE

History

- The oldest known cryptography dates back to around 1900 BCE in ancient Egypt, where hieroglyphs used a substitution cipher. Similarly, Mesopotamian cuneiform tablets from around 1500 BCE encrypted trade secrets, illustrating that cryptography has been used for millennia to protect information.

by HIMANSHU GOHITE

Cryptography and Ciphers

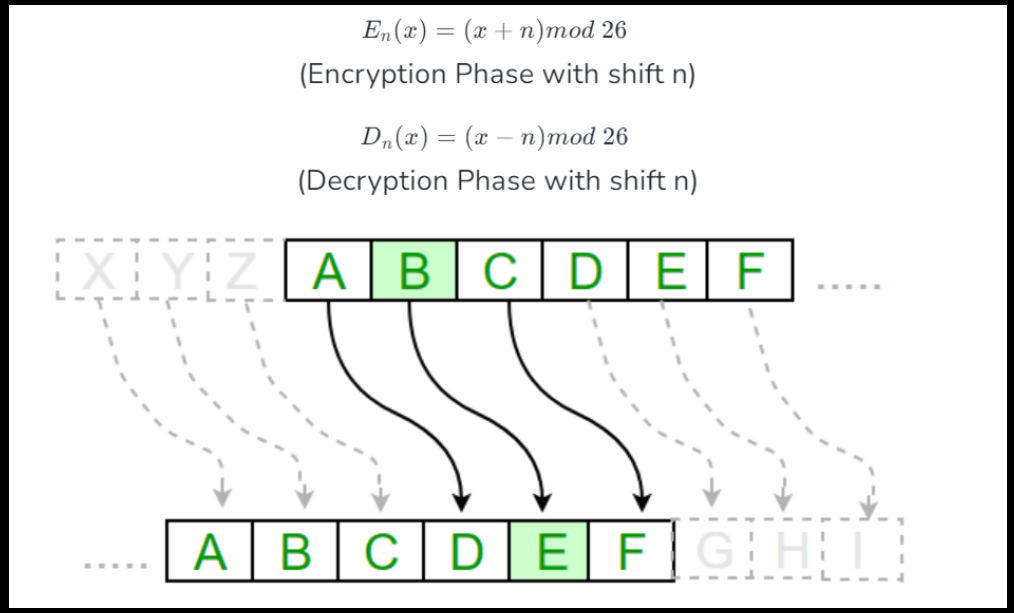
- Ciphers are the algorithms used for this transformation. Historically, ciphers have evolved from simple, manual techniques to complex, computational methods, and are classified into two main categories: Classical and Modern.
- Classical ciphers focus on manual techniques like substitution and transposition
- Modern ciphers employ more advanced, computational methods like symmetric and asymmetric encryption.

by HIMANSHU GOHITE

Classical Cipher : Ceaser Cipher

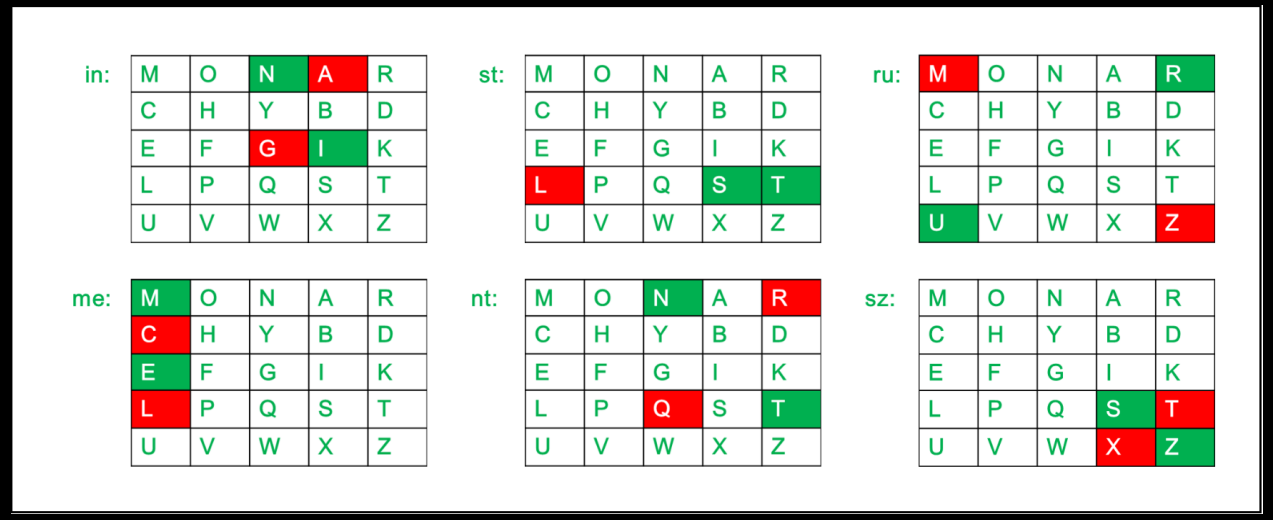
- The Caesar cipher is a simple encryption technique that was used by Julius Caesar to send secret messages to his allies. It works by shifting the letters in the plaintext message by a certain number of positions, known as the “shift” or “key”.

- It is a monoalphabetic substitution symmetric encryption based cipher with fixed key.



Classical Cipher : Playfair cipher

- The Playfair cipher was the first practical digraph substitution cipher. The scheme was invented in 1854 by Charles Wheatstone but was named after Lord Playfair who promoted the use of the cipher.
- It is a polyalphabetic substitution symmetric encryption based cipher with fixed key.



by HIMANSHU GOHITE

- **Playfair Cipher Encryption Algorithm**
- **Key Square**: A 5×5 grid of unique letters is generated based on the key. The letter "J" is omitted, and replaced with "I" if it appears in the plaintext.
- **Plaintext Preparation**: Split plaintext into digraphs (pairs of two letters). If there's an odd number of letters, add a bogus letter (e.g., 'Z' or 'X'). Insert a bogus letter if identical letters appear together.
- **Encryption Rules**:
  - **Same Column**: Replace each letter with the one below it (wrap around to the top if needed).
  - **Same Row**: Replace each letter with the one to the right (wrap around to the left if needed).
  - **Different Row & Column**: Form a rectangle and replace each letter with the one on the opposite corner.
- Example:
  - Key: “monarchy”
  - Plaintext: "instruments"
  - Prepared Text: "in st ru me nt sz"
  - Encrypted Text: "gatlmzclrqtx"

by HIMANSHU GOHITE

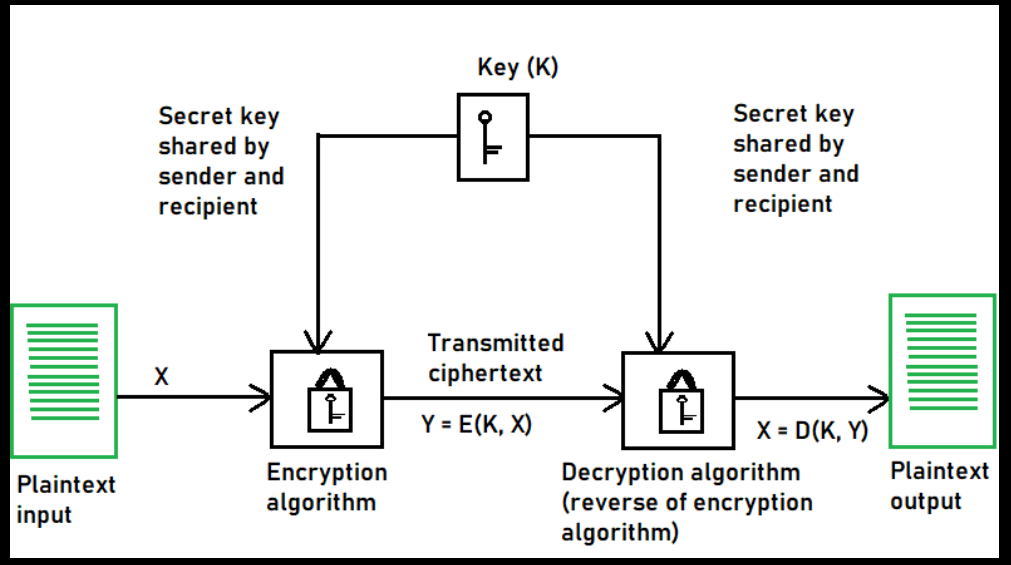
Vulnerabilities in Classical Ciphers

- Frequency Analysis: Letter frequencies are preserved, making patterns easy to exploit
- Repetitive Structures: Repeating key sequences can reveal patterns in the ciphertext.
- Small Keyspace: Limited key options make brute-force attacks feasible.
- Lack of Diffusion: Plaintext structure isn't sufficiently spread across the Ciphertext.
- No Authentication: Ciphers don't verify message integrity, allowing tampering or forgery

by HIMANSHU GOHITE

Modern Cipher : Symmetric Key Cryptography

- Symmetrical Key Cryptography also known as conventional or single-key or secret key encryption was the primary method of encryption before the introduction of public key cryptography in the 1970s. In symmetric-key algorithms, the same keys are used for data encryption and decryption.



What It Is:

- A cryptographic method where the same key is used for both encryption and decryption of data.
- Both the sender and receiver share a single secret key.

How It Works:

- The sender encrypts the plaintext using the secret key.
- The receiver uses the same key to decrypt the ciphertext back into plaintext.
- The key must be kept secure, as anyone with the key can decrypt the message.

Types:

- Block Ciphers: Encrypts fixed-size blocks of data (e.g., AES, DES).
- Stream Ciphers: Encrypts data one bit or byte at a time (e.g., RC4, ChaCha20).

by HIMANSHU GOHITE

Applications

- File Encryption: Protects sensitive files on a disk from unauthorized access.
- Disk Encryption: Secures entire disks or partitions to protect data at rest.
- TLS/SSL: Secures data transmitted over networks using encryption.
- VPNs: Encrypts data between users and VPN servers for secure communication.
- Email Encryption: Secures email content to prevent unauthorized access.

Day-2 : Agenda

- **Day 2: Cryptography and Data Security**
- **Overview of Data Security**
  - What is Data Security?

- **States of Data**
  - Data at Rest,
  - Data in Transition
  - Data in Use

- **Hashing**
  - What is Hashing
  - Types
  - Applications

Overview of Data Security

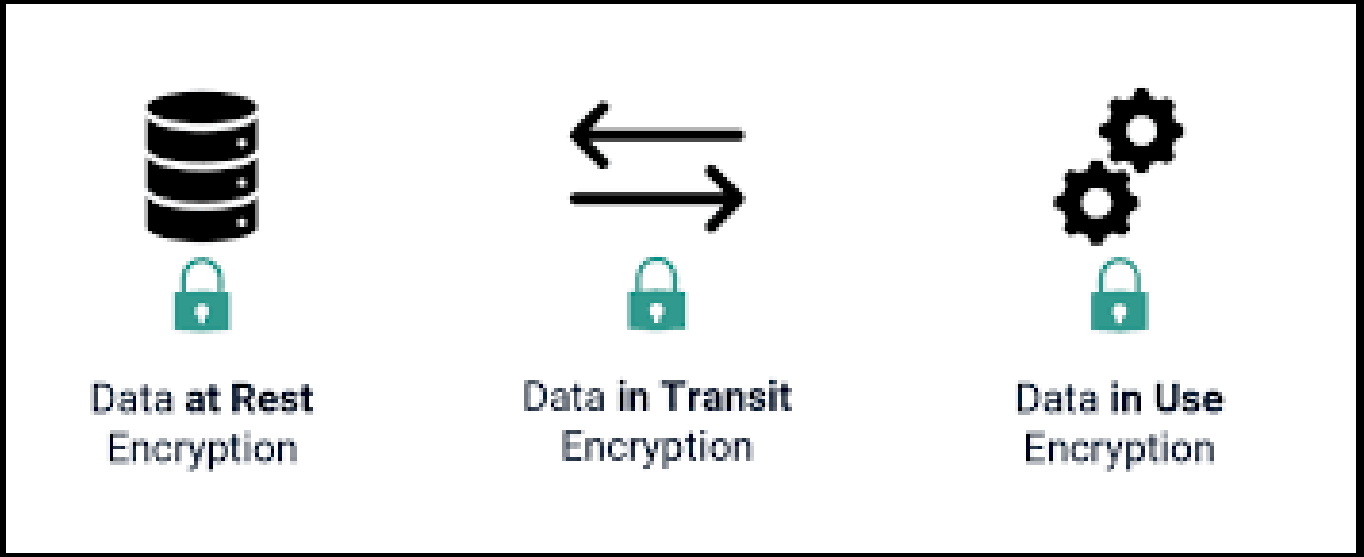
- Data Security is the practice of protecting digital information from unauthorized access, corruption, or theft throughout its lifecycle. It involves implementing various tools and techniques to ensure the confidentiality, integrity, and availability (CIA) of data, making sure it remains safe in various environments and use cases.

Data Security Components	Data Security Threats
Confidentiality : Access to Authorised	Data Exposure : Unauthorised Access
Integrity : Authorized Modification	Alteration : Unauthorised Modification
Availability : Timely Access	Denial of Service : No Timely Access

by HIMANSHU GOHITE

States of Data

- Data exists in three different states, and each requires specific security measures:
  - Data in Rest
  - Data in Transit
  - Data in Use



Data at Rest

- Definition: Data stored on physical or virtual storage devices (e.g., hard drives, SSDs, databases, cloud storage).
- Threats:
  - Unauthorized access (e.g., via stolen credentials or physical theft).
  - Insider threats (e.g., employees with malicious intent).
  - Ransomware attacks that encrypt or delete stored data.
- Security Measures:
  - Encryption: Protects data with encryption keys, ensuring it is unreadable without authorization.
  - Access Control: Ensures only authorized personnel can access the data.
  - Physical Security: Securing data storage devices in locked areas or using tamper-proof systems.
  - Backups: Regular backups to prevent data loss from attacks or hardware failure

Data in Transit

- Definition: Data being transmitted over a network (e.g., emails, file transfers, API communications).
- Threats:
  - Man-in-the-Middle (MitM) attacks intercepting and modifying the data during transmission.
  - Eavesdropping (unauthorized listening to or reading data).
  - Packet sniffing by attackers monitoring network traffic.
- Security Measures:
  - Encryption: Use of SSL/TLS to encrypt data during transmission, preventing unauthorized interception.
  - VPNs (Virtual Private Networks): Encrypts traffic to protect data being sent over unsecured networks.
  - Secure Protocols: Using secure communication protocols like HTTPS, FTPS, or SSH.
  - Firewalls: Protect networks from unauthorized access and monitor incoming and outgoing traffic

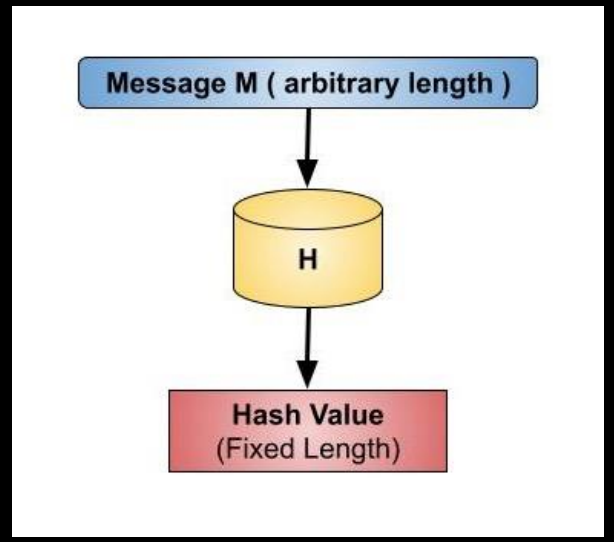
by HIMANSHU GOHITE

Data in Use

- Definition: Data currently being processed, accessed, or modified by applications or users (e.g., when you're working on a document).
- Threats:
  - Memory-based attacks (e.g., malicious software accessing data in memory).
  - Side-channel attacks that monitor system activity to infer data usage patterns.
  - Malware targeting running processes to extract sensitive data.
- Security Measures:
  - Secure Enclaves: Isolate data during processing (used in technologies like Intel SGX).
  - Access Control: Ensuring only authorized users and processes can access or modify the data.
  - Antivirus and Anti-malware: Protecting systems from malicious software that targets active data.

Hashing

- A one-way cryptographic function is a mathematical process that converts data (such as a file or password) into a fixed-length output, known as a hash. This function is designed to be irreversible, meaning it is computationally infeasible to retrieve the original input from the hash. It ensures data integrity by verifying that the data has not been altered.



- Example:
  - MD5 (Message Digest Algorithm 5): Produces a 128-bit hash; considered broken due to vulnerability to collisions.
  - SHA-1 (Secure Hash Algorithm 1): Generates a 160-bit hash; no longer secure due to discovered vulnerabilities.
  - SHA-256: Part of the SHA-2 family, produces a 256-bit hash; widely used in security protocols.
  - SHA-3: The latest member of the Secure Hash Algorithm family, with configurable output sizes (e.g., 256-bit, 512-bit).
  - RIPEMD-160: Produces a 160-bit hash; used in some cryptographic systems as an alternative to SHA-1.

by HIMANSHU GOHITE

Properties:

- Properties:
  - Deterministic: A given input will always produce the same hash output.
  - Fixed Output Size: Regardless of the input size, the hash output has a fixed length (e.g., 256 bits for SHA-256).
  - Preimage Resistance: It is computationally infeasible to reverse a hash and retrieve the original input.
  - Collision Resistance: No two different inputs should produce the same hash output.
  - Avalanche Effect: A small change in the input results in a drastically different hash output.
  - Efficiency: Hash functions should be fast to compute for any given input.

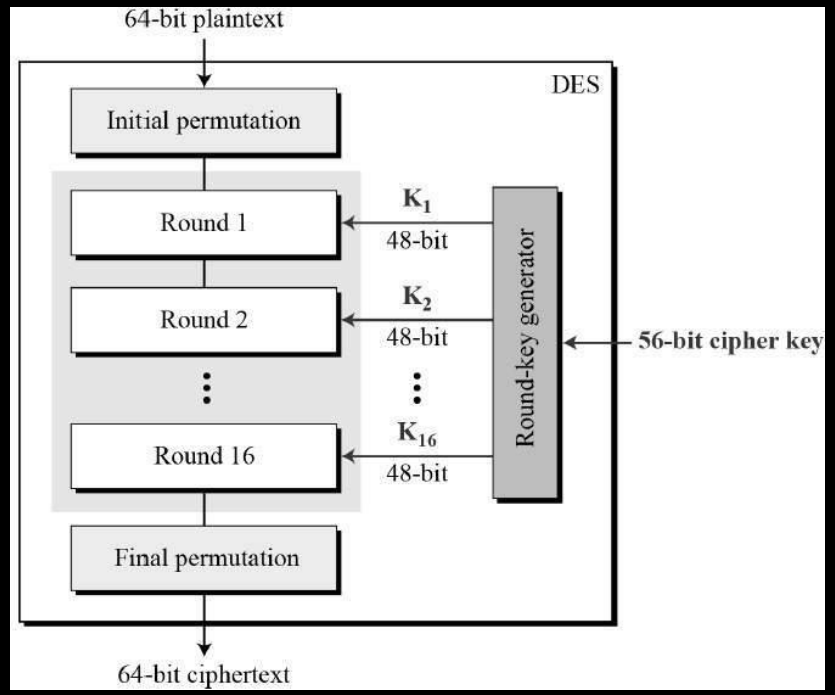
Day-3 : Agenda

- **Day 3: Understanding Symmetric Key Cryptography**
- **Data Encryption Standard (DES)**
  - Working
  - Applications and Vulnerabilities
- **Advanced Encryption Standard**
  - Working
  - Applications and Vulnerabilities
- **Block Cipher Modes of Operations**
  - What are modes of operations
  - Types, Applications, Pros & Cons
- **Message Authentication Code (MAC)**

Data Encryption Standard

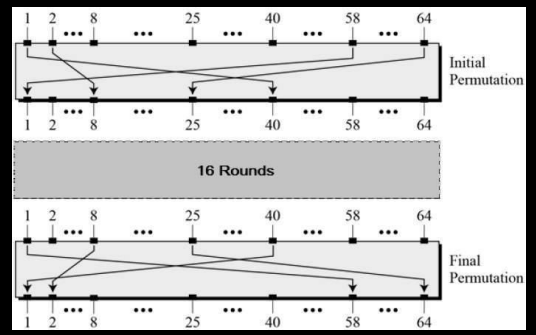
- The Data Encryption Standard (DES) is a symmetric-key block cipher published by the National Institute of Standards and Technology (NIST).

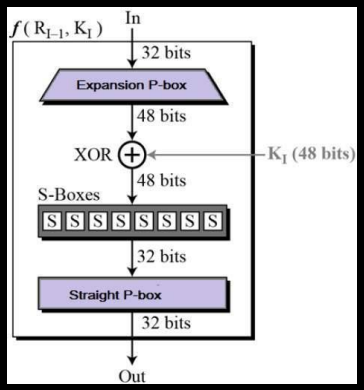
- DES is an implementation of a Feistel Cipher. It uses 16 round Feistel structure. The block size is 64-bit. Though, key length is 64-bit, DES has an effective key length of 56 bits, since 8 of the 64 bits of the key are not used by the encryption algorithm (function as check bits only).

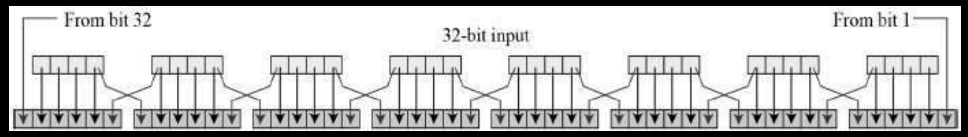


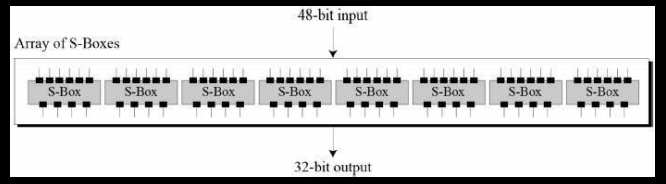
Working

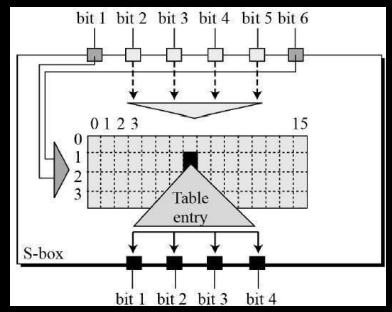
- **1. Initial and Final Permutation (IP & FP):**
  - **Initial Permutation (IP)**: Rearranges the 64-bit plaintext to spread the bits for better diffusion.
  - **Final Permutation (FP)**: Inverses the initial permutation after the 16 encryption rounds, generating the ciphertext.
- **2. Round Function:**
  - **Expansion (E)**: Expands the 32-bit right half (R) to 48 bits.
  - **XOR with Key**: The expanded R is XORed with a 48-bit round key.
  - **Substitution (S-boxes)**: 48 bits are divided into 6-bit blocks, each substituted using 8 S-boxes to produce a 32-bit output.
  - **Permutation (P)**: The 32-bit output is permuted to further shuffle the bits.
- **3. Key Generation:**
  - The 64-bit key is reduced to 56 bits, split into two halves.
  - In each of the 16 rounds, the halves are shifted and permuted to create a unique 48-bit round key.

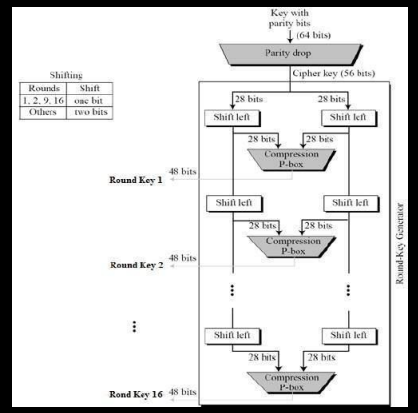












by HIMANSHU GOHITE

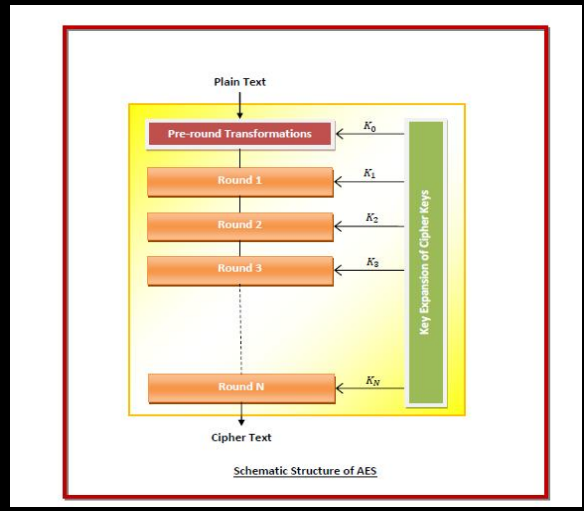
Applications and Vulnerabilities

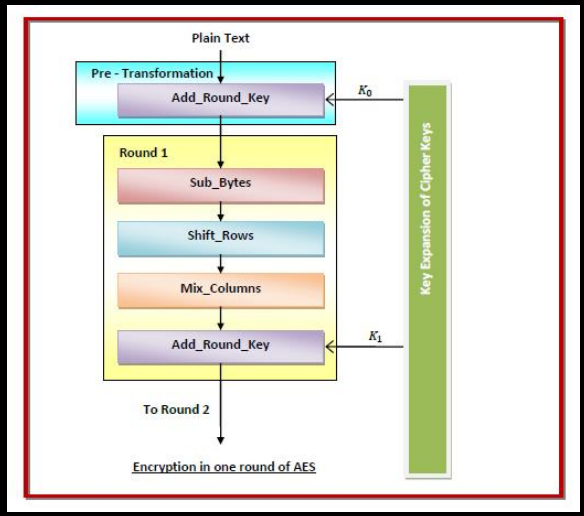
- **Applications of DES:**
  1. **File Encryption**: Used to secure sensitive files on local systems.
  2. **Network Communication**: Historically used to encrypt data in secure communications like VPNs.
  3. **ATM Transactions**: Ensured secure transmission of PINs and transaction data.
  4. **Smart Cards**: Implemented for securing data in various card-based systems.
  5. **Wireless Communication**: Applied in securing early wireless protocols.
- **Vulnerabilities of DES:**
  1. **Short Key Length**: The 56-bit key is too small by modern standards, making it vulnerable to **brute-force attacks** (attempting all possible keys).
  2. **Cryptanalysis Attacks**: **Linear and differential cryptanalysis** techniques can exploit weaknesses in DES, reducing the effort needed to break it.
  3. **Meet-in-the-Middle Attack**: DES is vulnerable when used in **Double-DES** (encrypting twice with different keys), allowing attackers to find keys faster than brute-forcing.
  4. **Key Exhaustion**: Due to its limited key space, DES can be cracked using modern hardware in a relatively short time.

by HIMANSHU GOHITE

Advanced Encryption Standard

- The Advanced Encryption Standard, or AES, is an encryption standard established in 2001 by the National Institute of Standards and Technology (NIST) of USA.
- Features:
  - AES is a subset of Rijndael block cipher.
  - It is a successor of Data Encryption Standard (DES) and is stronger and faster than DES.
  - It is a symmetric key symmetric block cipher.
  - It operates on 128-bit (16 bytes) data.
  - The cipher key may be of 128, 192 or 256 bits.
  - All computations are performed on bytes rather than bits.
  - AES gives full specification and design details.
  - It can be implemented using languages C and Java for software protection.



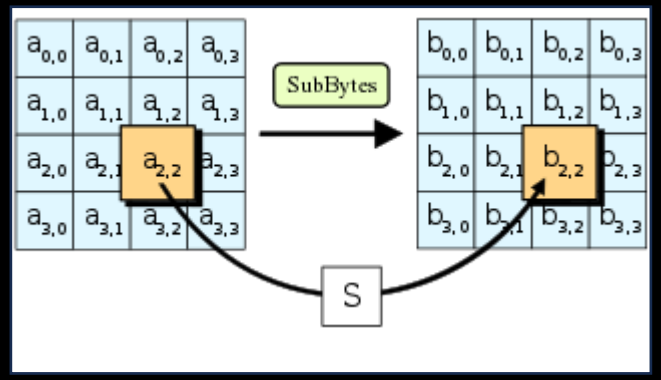


by HIMANSHU GOHITE

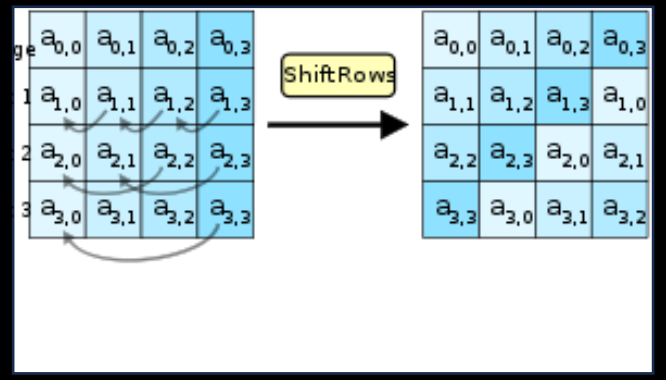
Working

- **Key Expansion** – The round keys are calculated from the cipher key using Rijndael's block cipher schedule.
- **Pre–Transformation** – This comprises of only 1 process namely Add_Round_Key. Here, XOR operation is performed on each data byte with a byte of the round key.
- **Round 1 to Round (N-1)** – Four sub-processes are performed here.
  - Sub_Bytes – Non-linear substitution is performed on each byte whereby the byte is replaced with another byte as per a lookup table.
  - Shift_Rows – Transposition is performed wherein a certain number of cyclical shifting of the last three rows is done.
  - Mix_Columns – Mixing of rows and columns in a pre-defined manner is performed.
  - Add_Round_Key – XOR operation is performed on each byte with a byte of the round key.
- **Round N** – The final round comprises of three sub-processes, namely –
  - Sub_Bytes
  - Shift_Rows
  - Add_Round_Key

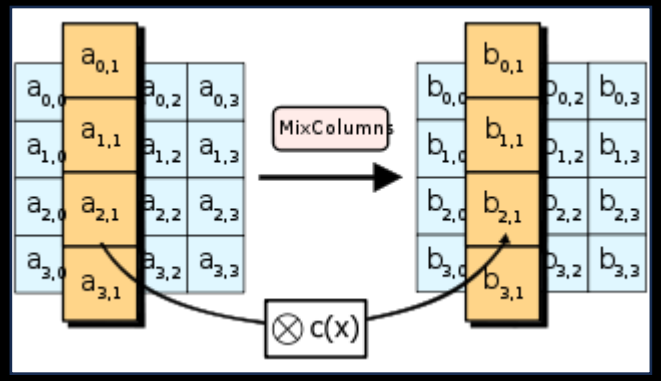
1) Sub-Bytes



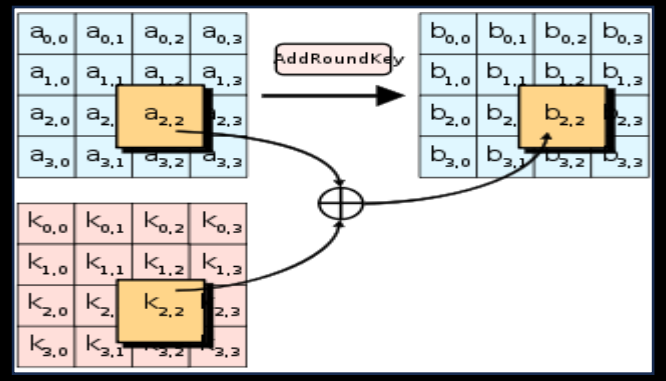
2) Shift-Rows



3) Mix-Columns



4) Add Round Keys



by HIMANSHU GOHITE

Applications and Vulnerabilities

- **Applications of AES (Advanced Encryption Standard):**
  1. **Secure Communications**: Used in protocols like SSL/TLS for encrypting web traffic (e.g., HTTPS).
  2. **File and Disk Encryption**: Employed in systems like BitLocker, VeraCrypt, and file encryption tools.
  3. **Wireless Security**: Used in Wi-Fi encryption protocols like WPA2 and WPA3.
  4. **VPNs**: Provides encryption for secure tunneling in VPNs.

- Vulnerabilities of AES:
  - **Weak Key Scheduling**: Poor key management practices can weaken AES encryption, making systems vulnerable to key-recovery attacks.
  - **Fault Attacks**: Introducing intentional faults during encryption can expose key information.
  - **Brute-Force**: While AES is highly secure, a brute-force attack is theoretically possible but infeasible with current technology due to its large key sizes (128, 192, or 256 bits).

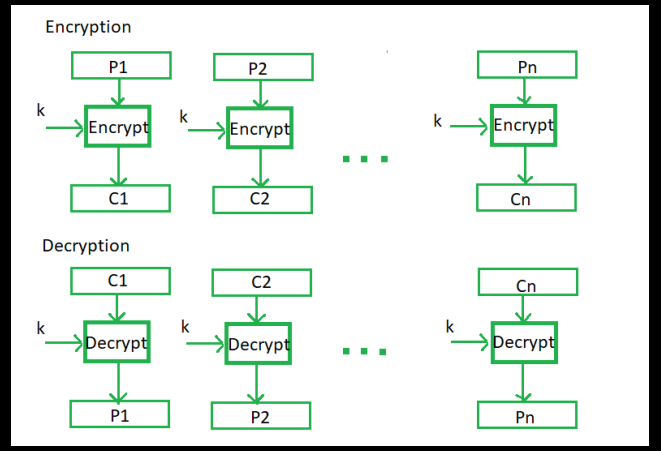
by HIMANSHU GOHITE

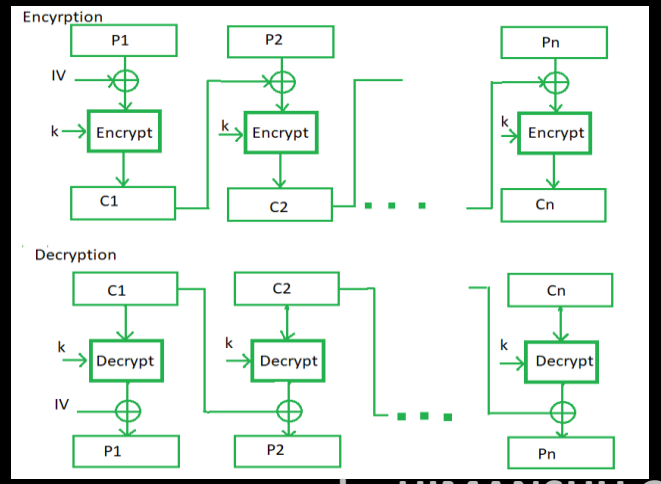
Modes Of Operations

- **Modes of operation** for a block cipher define how to repeatedly apply the cipher to securely encrypt data larger than a single block (usually 64 or 128 bits). Block ciphers process fixed-size blocks of data, and modes of operation manage how blocks are encrypted/decrypted in a series.
- Common Modes of Operations are :
  - ECB (Electronic Codebook Mode)
  - CBC (Cipher Block Chaining Mode)
  - CFB (Cipher Feedback Mode)
  - OFB (Output Feedback Mode)
  - CTR (Counter Mode)

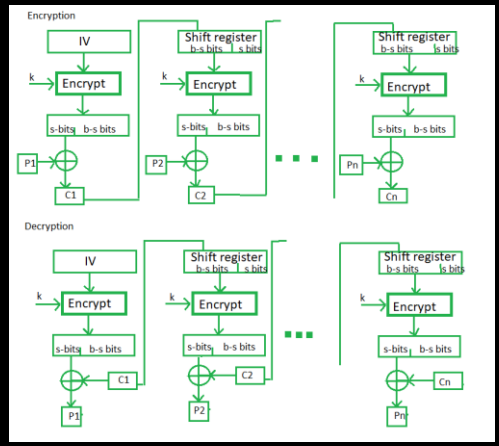
Applications, Pros & Cons

- ECB (Electronic Codebook Mode):
  - How it works: Each block is encrypted independently.
  - Pros: Simple and fast.
  - Cons: Identical plaintext blocks result in identical ciphertext blocks, revealing patterns and making it insecure for most uses.
  - Use Case: Rarely used, only for small, non-sensitive data.
- CBC (Cipher Block Chaining Mode):
  - How it works: Each plaintext block is XORed with the previous ciphertext block before encryption. The first block uses an Initialization Vector (IV).
  - Pros: Hides patterns in the plaintext.
  - Cons: Slower due to dependency on previous blocks; requires IV.
  - Use Case: Secure file encryption, VPNs, TLS.

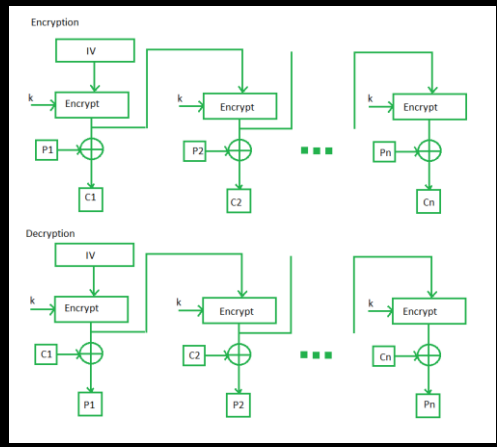




- CFB (Cipher Feedback Mode):
  - How it works: Converts a block cipher into a stream cipher. Ciphertext from the previous block is fed back into the encryption process.
  - Pros: Allows encryption of data smaller than the block size.
  - Cons: Errors propagate through the chain.
  - Use Case: Real-time data encryption like streaming.

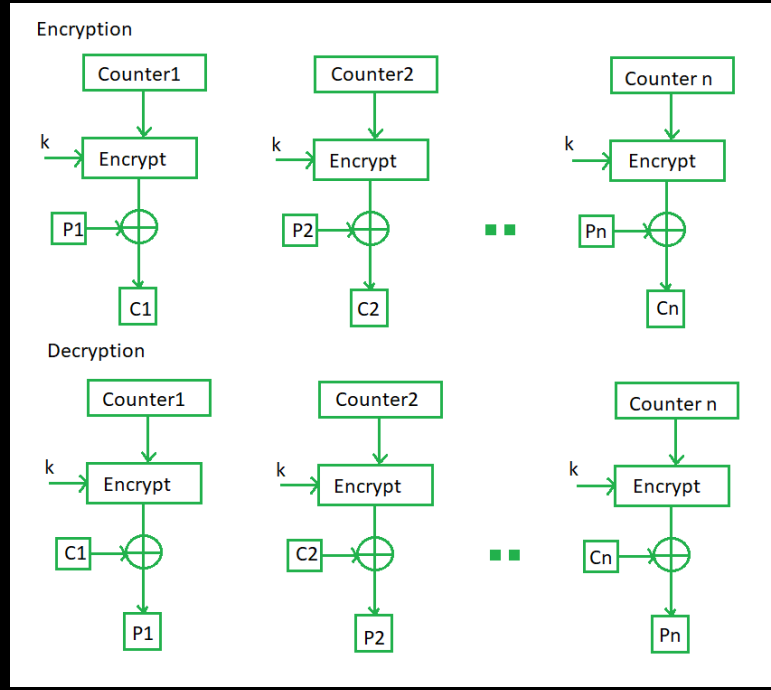


- OFB (Output Feedback Mode):
  - How it works: Similar to CFB, but the key stream is independent of both plaintext and ciphertext. The IV is encrypted repeatedly to generate a key stream.
  - Pros: No error propagation; useful for streaming data.
  - Cons: Requires a unique IV for each encryption session.
  - Use Case: Secure data transmission, satellite communication.



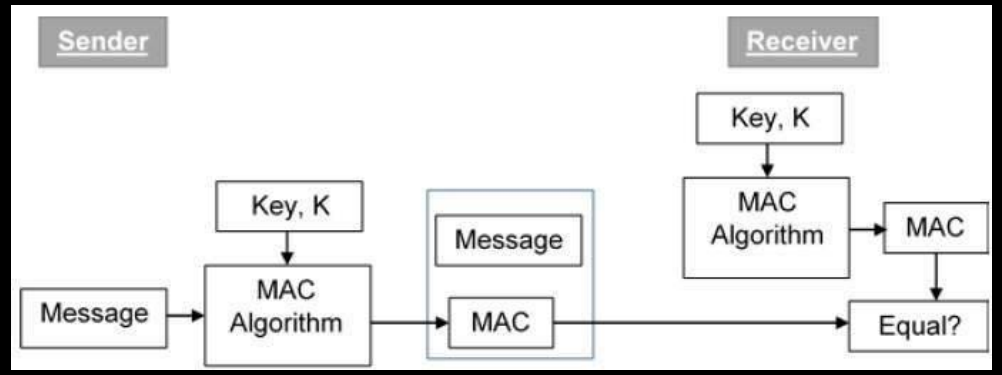
by HIMANSHU GOHITE

- **CTR (Counter Mode)**:
- **How it works**: A counter is used as an input to the block cipher for each block, incrementing for each subsequent block. This counter is XORed with the plaintext to produce ciphertext.
- **Pros**: Parallelizable, fast, no error propagation.
- **Cons**: Counter must be unique for each encryption session.



Message Authentication Code (MAC)

- MAC algorithm is a symmetric key cryptographic technique to provide message authentication. For establishing MAC process, the sender and receiver share a symmetric key K.
- Essentially, a MAC is an encrypted checksum generated on the underlying message that is sent along with a message to ensure message authentication
- Applications of MAC (Message Authentication Code):
  1. **Data Integrity**: Ensures that data has not been tampered with during transmission or storage.
  2. **Authentication**: Verifies the authenticity of the message, ensuring it was sent by a legitimate sender.
  3. **Network Security Protocols**: Used in protocols like TLS, IPsec, and SSH to provide data integrity and authentication.
- Limitations:
  - Symmetric Key Requirement: Both the sender and receiver must share the same secret key, which can complicate key management in large systems.
  - No Non-repudiation: MAC does not provide non-repudiation, meaning the sender can deny having sent the message since both parties share the same key.
  - Vulnerability to Key Disclosure: If the shared key is compromised, an attacker can forge MACs and bypass integrity checks.



by HIMANSHU GOHITE

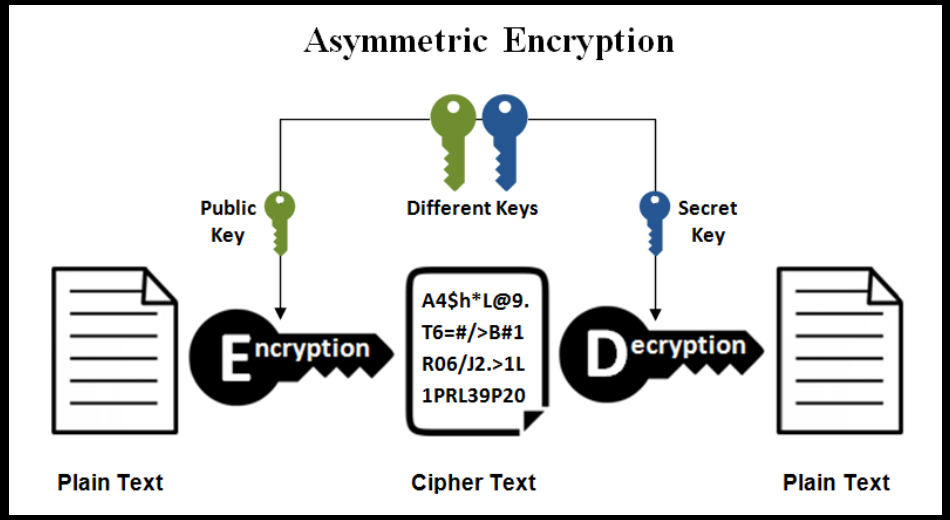
Day-4 : Agenda

- **Day 4: Understanding Asymmetric Key Cryptography**
- Overview of Asymmetric Cryptography
  - Working & Applications
- Diffie-Hellman Key Exchange
  - Working
  - Application and Vulnerabilities
- RSA Encryption
  - Working
  - Application and Vulnerabilities

- Digital Signatures
  - Working
  - Application

Overview of Asymmetric Key Cryptography

- Asymmetric encryption, also known as public-key cryptography, is a type of encryption that uses a pair of keys to encrypt and decrypt data. The pair of keys includes a public key, which can be shared with anyone, and a private key, which is kept secret by the owner.
- Public Key: A key that can be distributed openly to anyone. Used for encrypting messages or verifying digital signatures.
  - Example: If Alice wants to send a secure message to Bob, she encrypts it using Bob's public key.
- Private Key: A key that is kept secret by the owner. Used for decrypting messages or creating digital signatures.
  - Example: Bob decrypts Alice's message with his private key, ensuring only he can read it.



by HIMANSHU GOHITE

Advantages and Applications

- Advantages:
  - No Key Sharing Risk: Public keys can be shared openly, and the private key remains secure.
  - Provides Authentication: Digital signatures ensure the message comes from the intended sender.
  - Supports Non-Repudiation: The sender cannot deny sending the message if it's digitally signed.
  - Key Distribution Simplified: There’s no need for a secure channel to distribute the public key.
  - Scalability: More scalable than symmetric key cryptography, especially in large networks.
- Applications:
  - SSL/TLS (HTTPS): Secures web traffic by encrypting communications between browsers and servers.
  - Digital Signatures: Used in email encryption (e.g., PGP) and software distribution to verify authenticity.
  - Secure Email (PGP, S/MIME): Ensures email confidentiality, integrity, and authentication.
  - Cryptocurrencies: Used for wallet addresses and digital signatures in Bitcoin, Ethereum, etc.
  - Authentication: Public key infrastructure (PKI) for verifying the identity of users, devices, and systems.

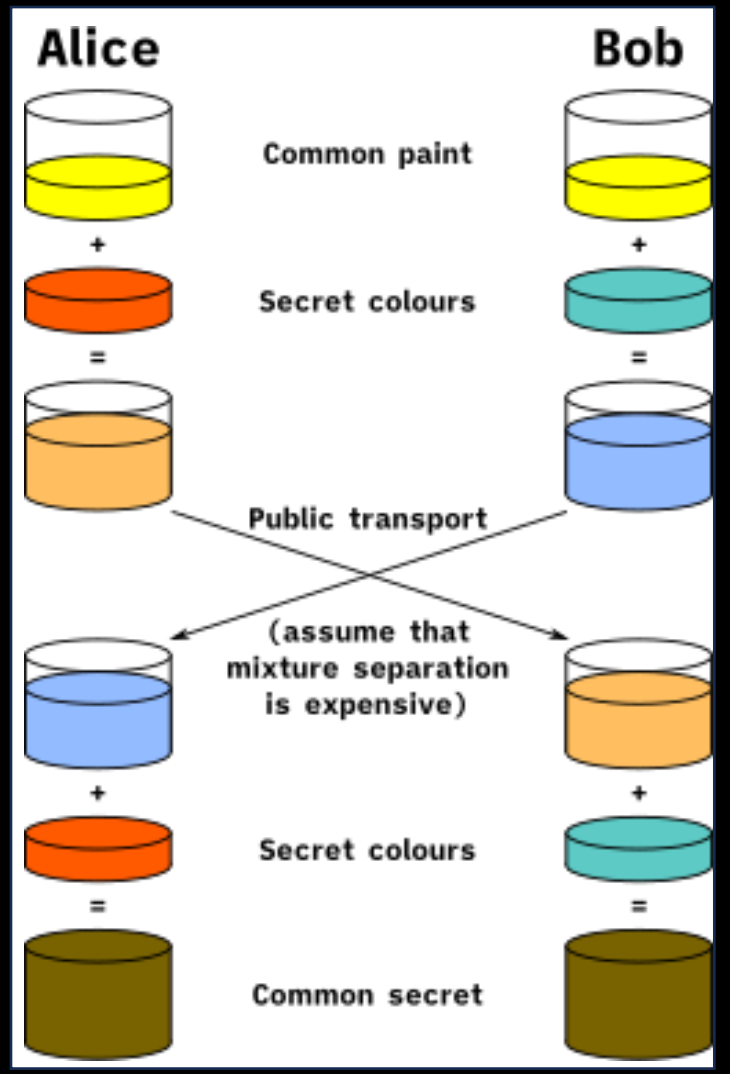
Diffie-Hellman-Merkel Key Exchange

- Diffie–Hellman (DH) key exchange is a mathematical method of securely generating a symmetric cryptographic key over a public channel and was one of the first public-key protocols as conceived by Ralph Merkle and named after Whitfield Diffie and Martin Hellman. DH is one of the earliest practical examples of public key exchange implemented within the field of cryptography. Published in 1976 by Diffie and Hellman, this is the earliest publicly known work that proposed the idea of a private key and a corresponding public key.

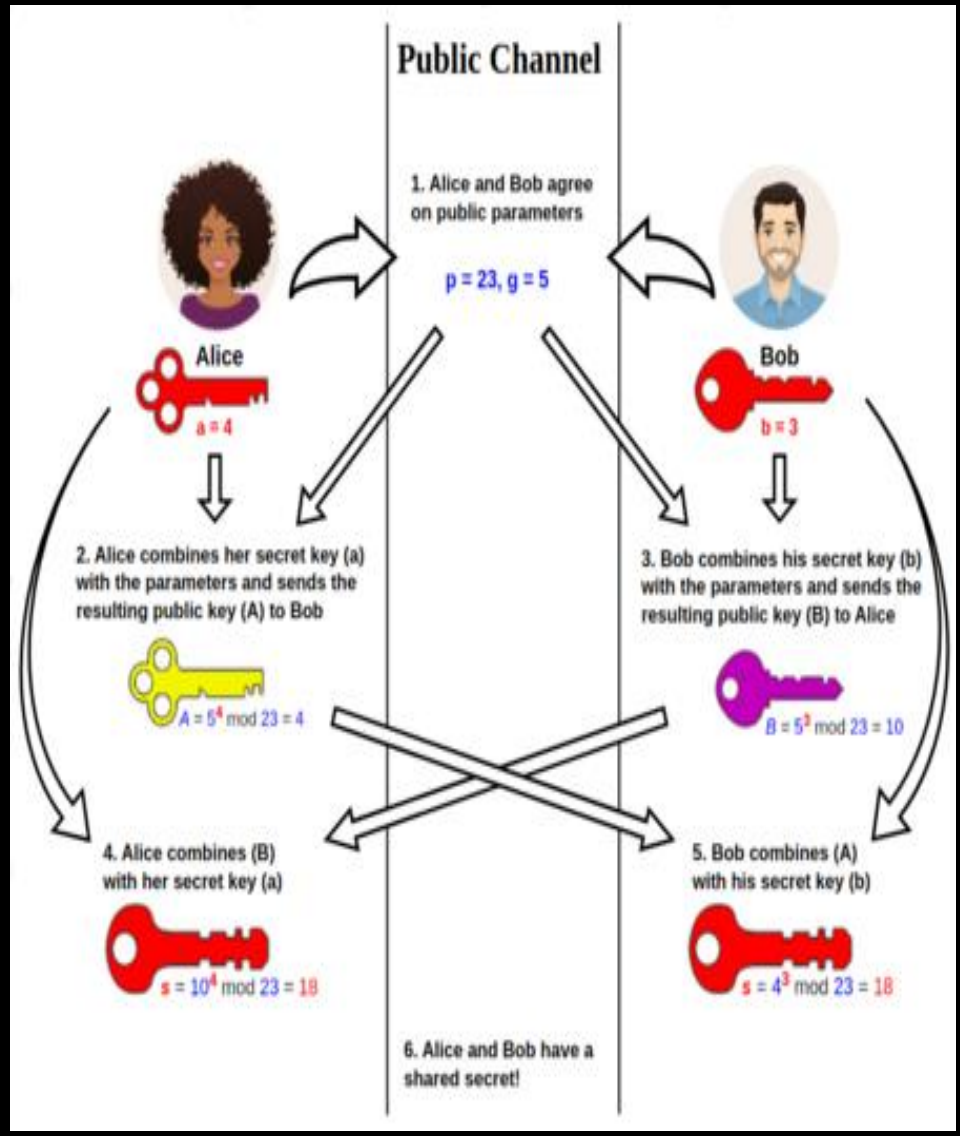
by HIMANSHU GOHITE

Working

- The process begins by having the two parties, Alice and Bob, publicly agree on an arbitrary starting color that does not need to be kept secret. In this example, the color is yellow.
- Each person also selects a secret color that they keep to themselves – in this case, red and cyan.
- The crucial part of the process is that Alice and Bob each mix their own secret color together with their mutually shared color, resulting in orange-tan and light-blue mixtures respectively, and then publicly exchange the two mixed colors.
- Finally, each of them mixes the color they received from the partner with their own private color. The result is a final color mixture (yellow-brown in this case) that is identical to their partner's final color mixture.



1. Alice and Bob publicly agree to use a modulus $p = 23$ (prime) and base $g = 5$ (which is a primitive root modulo 23).
2. Alice chooses a secret integer $a = 4$, then sends Bob $A = g^a \bmod p$
   - $A = 5^4 \bmod 23 = 4$ (in this example both A and a have the same value 4, but this is usually not the case)
3. Bob chooses a secret integer $b = 3$, then sends Alice $B = g^b \bmod p$
   - $B = 5^3 \bmod 23 = 10$
4. Alice computes $s = B^a \bmod p$
   - $s = 10^4 \bmod 23 = 18$
5. Bob computes $s = A^b \bmod p$
   - $s = 4^3 \bmod 23 = 18$
6. Alice and Bob now share a secret (the number 18).



by HIMANSHU GOHITE

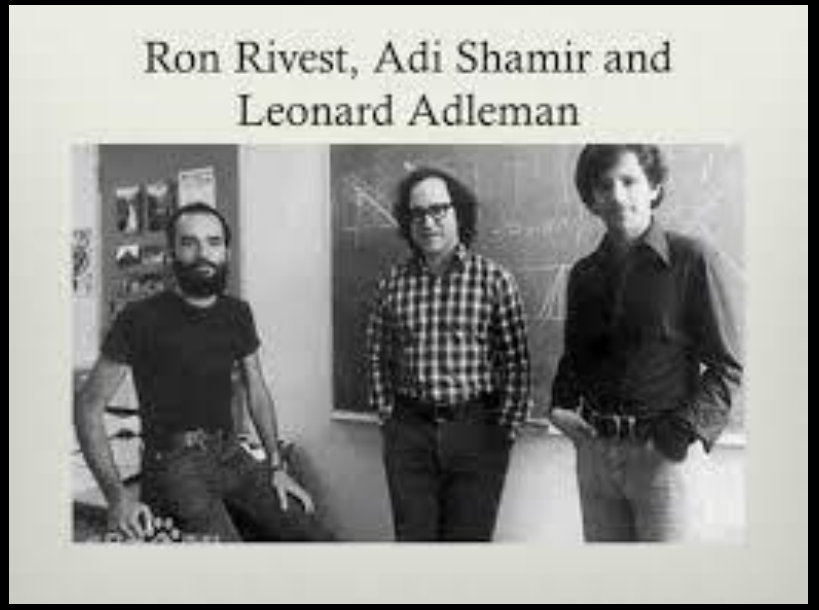
Applications and Vulnerabilities

- Applications of Diffie-Hellman (DH):
  - Secure Key Exchange: Used to generate a shared secret key over an insecure channel for symmetric encryption.
  - VPNs: Part of IPsec protocol to securely exchange encryption keys between VPN clients and servers.
  - SSH: Utilized during the handshake process to securely exchange encryption keys.
  - WPA3: Enhances Wi-Fi security by using DH for secure encryption and authentication.
  - Messaging Apps: Employed in end-to-end encryption (e.g., Signal, WhatsApp) to establish secure communication keys.
- Vulnerabilities of Diffie-Hellman:
  - Man-in-the-Middle Attack: Vulnerable without authentication, allowing attackers to intercept and alter communications.
  - Small Subgroup Attacks: Exploits weak parameters to gain information about the secret key.
  - Quantum Computing Threat: Future quantum computers could break DH by solving the discrete logarithm problem.

by HIMANSHU GOHITE

RSA Cryptosystem

- RSA (Rivest–Shamir–Adleman) is a public-key cryptosystem, one of the oldest widely used for secure data transmission. The initialism "RSA" comes from the surnames of Ron Rivest, Adi Shamir and Leonard Adleman, who publicly described the algorithm in 1977



by HIMANSHU GOHITE

Working

- **Key Generation**:
  - Two large prime numbers, p and q, are chosen.
  - Calculate n=p×q (this is the modulus used in both keys).
  - Compute ϕ(n)=(p−1)×(q−1) (this is the Euler's totient function).
  - Choose a public exponent e, typically 65537, which is coprime with ϕ(n). (Choose "e" such that 1<e < ϕ (n))
  - Calculate the private key exponent d such that (d × e) mod ϕ(n)=1.
  - The **public key** consists of (n, e).
  - The **private key** consists of (n, d).
- Encryption:
  - The sender encrypts the message M using the recipient's public key (n,e) by computing:
  - C= M^e mod n
  - C is the ciphertext sent to the recipient.
- Decryption:
  - The recipient decrypts the ciphertext C using their private key (n,d) by computing:
  - M=C^d mod n
  - The decrypted message M is recovered.

- **Key Generation**
  - Choose two prime numbers: p=and q=7.
  - Calculate n: n= p×q = 3×7=21.
    n is used as the modulus in both the public and private keys.
  - Calculate ϕ(n**)** (Euler's totient function):
    ϕ(n)= (p−1)×(q−1) => (3−1)×(7−1) => 2×6=12
  - Choose public exponent e:
    e must be coprime with ϕ(n). Let's choose e= 5 (since 5 and 12 are co-prime).
  - Calculate private key exponent d:
    d is the multiplicative inverse of e mod ϕ(n), meaning (d × e)mod ϕ(n)=1.

    Using trial and error (or a modular inverse algorithm), we find d=5, because

    (5 × 5) mod 12 => 25 mod 12 = 1.
- Thus, the **public key** is (n = 21, e = 5), and the **private key** is (n=21,d=5).
- **Encryption**:
  - Suppose the message M=4 (we're using small numbers here for simplicity).
  - To encrypt M using the public key (n=21,e=5), we compute:
  - C=M^emod n= 4^5 mod 21
  - Calculating 4^5= 1024, then 1024 mod 21=10.
  - So, the ciphertext C=10.
- **Decryption**:
  - To decrypt the ciphertext C=10 using the private key (n=21,d=5),
  - we compute:M= C^d mod n= 10^5 mod 21
  - Calculating 10^5=100000, then 100000 mod 21=4.
  - So, the original message M=4 is recovered.

by HIMANSHU GOHITE

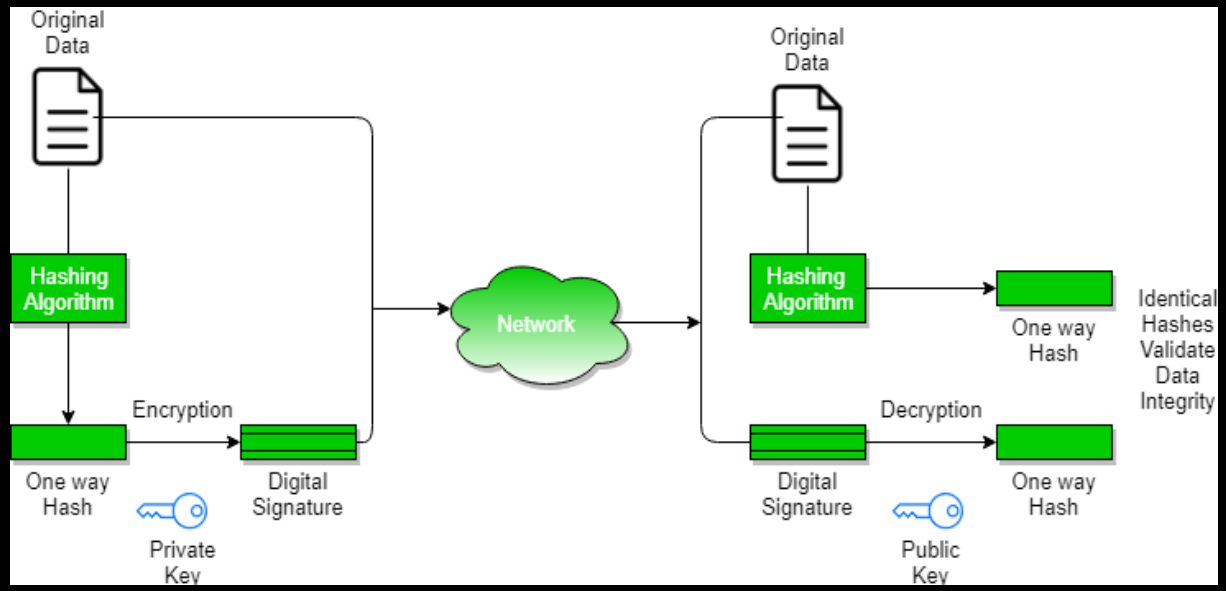
Applications and Vulnerabilities

- Applications of RSA:
  - Secure Web Communication: RSA is used in SSL/TLS for establishing secure connections in web browsers.
  - Digital Signatures: RSA provides authentication and integrity by enabling users to sign documents and verify signatures.
  - Email Encryption: RSA is used in email encryption systems like PGP to ensure confidentiality.
  - VPNs: RSA helps securely exchange keys in VPN tunnels to encrypt data between clients and servers.
  - Software Licensing: RSA is used to verify software licenses and prevent piracy.
- Vulnerabilities of RSA:
  - Factoring Attack: If large prime numbers are not used, attackers can factor the modulus nn to retrieve the private key.
  - Chosen Ciphertext Attack: Malicious users can manipulate ciphertexts and gain information about the private key.
  - Quantum Threat: Future quantum computers may be able to break RSA by efficiently factoring large numbers.
  - Weak Key Generation: Using small or improperly generated primes makes RSA susceptible to cryptographic attacks.

by HIMANSHU GOHITE

Digital Signatures

- A Digital Signature is a cryptographic method used to verify the authenticity and integrity of digital messages or documents. It ensures that the message has not been altered and confirms the sender's identity.



- How it Works:
  - The sender hashes the original message to create a message digest.
  - The sender then encrypts this message digest with their private key to create the digital signature.
  - The recipient uses the sender's public key to decrypt the signature, revealing the message digest.
  - The recipient hashes the original message again and compares it with the decrypted message digest.
  - If both match, the signature is valid, confirming integrity and authenticity.
- Key Features of Digital Signatures:
  - **Authenticity**: Ensures the message is from the claimed sender.
  - **Integrity**: Guarantees that the message has not been tampered with during transmission.
  - **Non-repudiation**: The sender cannot deny having signed the message, as only their private key could have created the signature.
  - **Confidentiality** (optional): Often combined with encryption to keep the message confidential along with verifying authenticity.

Applications

1. **Email Security**: Digital signatures are used to authenticate and ensure the integrity of email messages, preventing tampering and verifying the sender's identity (e.g., in S/MIME or PGP).
2. **Software Distribution**: Software vendors use digital signatures to verify the authenticity of software updates and installations, ensuring the code has not been altered by malicious actors.
3. **Online Transactions**: In e-commerce and banking, digital signatures provide security for transactions, verifying that they are initiated by legitimate users.
4. **Document Signing**: Digital signatures are used to sign contracts, legal documents, and certificates electronically, ensuring that the documents are legitimate and have not been altered.
5. **Blockchain**: Digital signatures secure blockchain transactions by ensuring the authenticity and integrity of each transaction in cryptocurrencies like Bitcoin and Ethereum.
6. **Digital Certificates**: In SSL/TLS protocols, digital signatures are part of digital certificates, which secure web traffic by validating server identity and establishing encrypted communication.

Day-5 : Agenda

- **Day 5: Understanding Obfuscation**
- **Overview of Obfuscation**
  - Definition and Techniques
- **Steganography**
  - Working
  - Application and Vulnerabilities
- **Tokenization**
  - Working
  - Application and Vulnerabilities
- **Data Masking**
  - Working
  - Application

Overview of Obfuscation

Obfuscation is the act of deliberately making something unclear, confusing, or difficult to understand. It is often used to conceal, distort, or obscure information, ideas, or intentions to mislead or deceive.

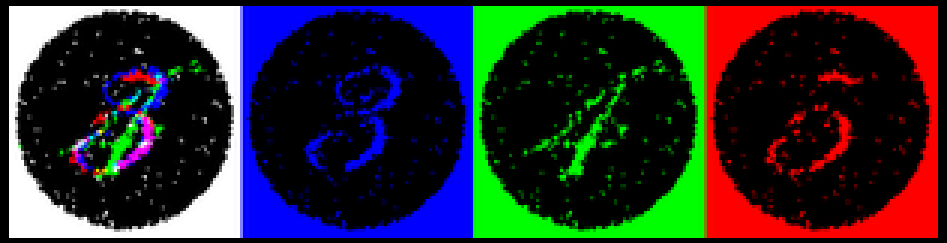
There are 3 main techniques of Obfuscation:

1. Steganography
2. Tokenization
3. Data Masking

by HIMANSHU GOHITE

Steganography

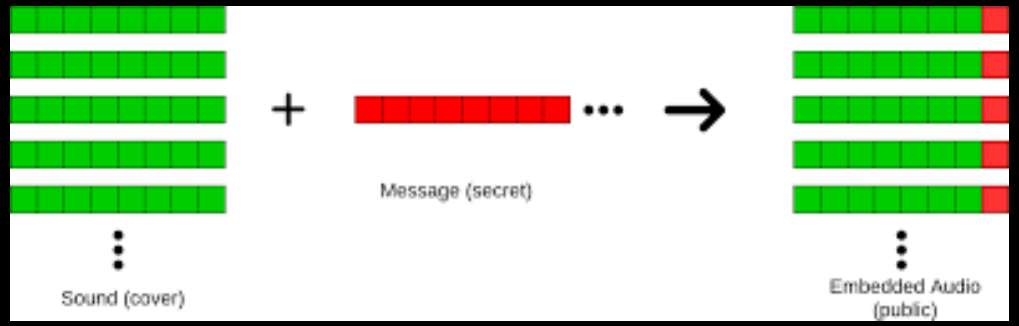
- **Steganography** is the practice of concealing a secret message within another non-secret object, such as an image, audio file, video, or text, in a way that hides its existence. Unlike encryption, which makes the message unreadable, steganography focuses on ensuring the message remains undetectable.
- **Image Steganography**:
- Secret data is embedded in digital images, often by modifying the least significant bits (LSB) of pixel values so that the changes are imperceptible.
- Example: Hiding a message inside an image file (e.g., PNG, BMP, or JPEG).



The same image viewed by white, blue, green, and red lights reveals different hidden numbers.

by HIMANSHU GOHITE

- Audio Steganography:
  - Conceals data within audio files by altering sound signals in a way that the changes are inaudible.
  - Example: Hiding data in the background noise of an audio file (e.g., MP3, WAV).



- Video Steganography:
  - Embeds secret data in video files by modifying individual frames or the audio track.
  - Example: Hiding information in video formats like MP4 or AVI.

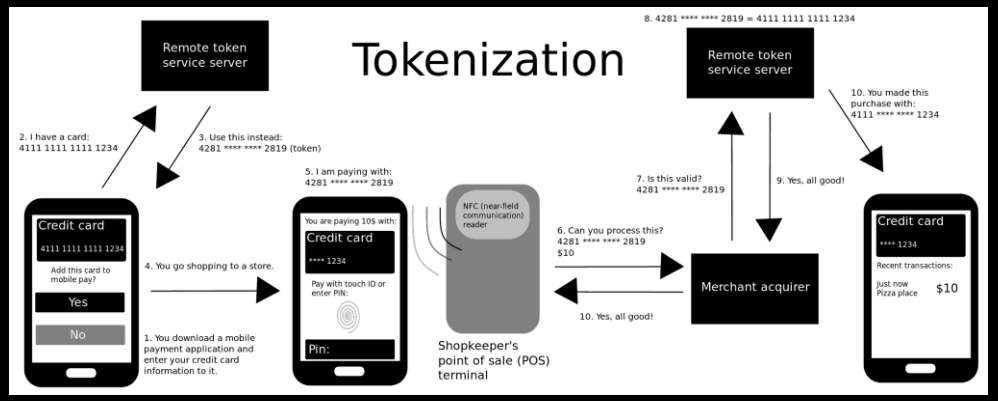
Tokenization

- **Tokenization** is a process where sensitive data is replaced with a non-sensitive equivalent called a "token." The token itself has no meaningful value or exploitable information, but it acts as a reference to the original data stored securely in a tokenization system. It is commonly used to protect sensitive information like credit card numbers, personal identification numbers (PINs), or Social Security numbers (SSNs).
- **How Tokenization Works:**

1. **Sensitive Data Collection**: The sensitive data (e.g., a credit card number) is collected by the system.
2. **Token Generation**: A tokenization system generates a unique token, which is a randomized or structured value that represents the sensitive data.
3. **Mapping Stored**: The original data is securely stored in a tokenization vault, and the generated token is mapped to the original data.
4. **Token Use**: The token is used in place of the sensitive data in systems or transactions, ensuring the actual sensitive data is not exposed.
5. **Detokenization**: When needed (e.g., for authorized users), the token can be converted back into the original data through detokenization by accessing the tokenization vault.

by HIMANSHU GOHITE

Applications



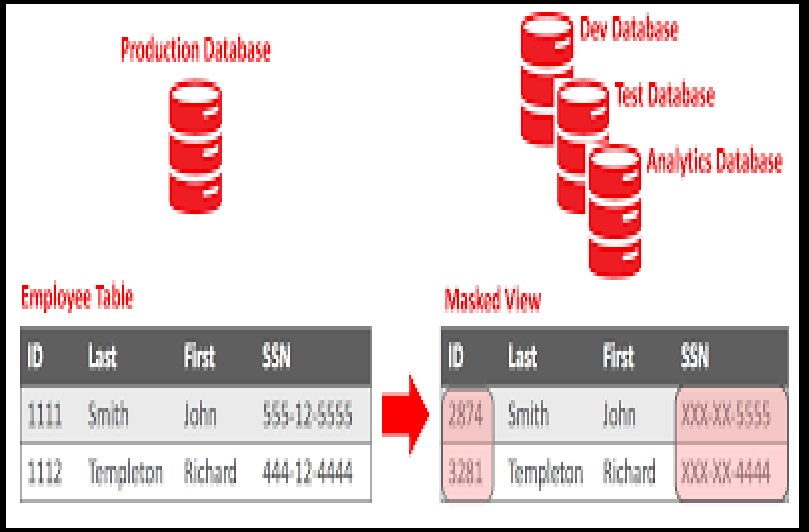
- **E-Commerce:**
  - Online merchants use tokenization to protect customer payment details during checkout processes, ensuring that sensitive data is never stored or transmitted in an unprotected form.
- **Payment Processing**:
  - Tokenization is widely used in **credit card payment systems**, such as in **Apple Pay** or **Google Pay**, to replace credit card numbers with tokens, protecting card data from breaches.
- **Healthcare**:
  - In the healthcare industry, tokenization helps secure sensitive **patient data** such as medical records or insurance information, ensuring compliance with regulations like HIPAA.
- **Data Privacy**:
  - Tokenization is used in systems handling **personally identifiable information (PII)** to protect customer or user data (e.g., SSNs, addresses) while maintaining system functionality.

Data Masking

- Masking is another obfuscation technique used to protect sensitive data by partially or fully concealing it with characters, symbols, or other data.
- **How Data Masking Works:**
  1. **Sensitive Data Identification**: Identify the sensitive data that needs to be masked, such as personal identifiable information (PII), payment details, or health records.
  2. **Data Transformation**: The sensitive data is replaced with masked values. These new values maintain the same format and structure as the original data, ensuring that the dataset remains usable for testing, analysis, or development.
  3. **Data Access**: Only the masked version of the data is made available to non-production environments. The original data is stored securely and only accessible to authorized personnel.
  4. **Irreversible Masking**: Unlike tokenization, data masking is usually irreversible. Once data is masked, the original data cannot be retrieved from the masked version.

Applications

- Software Development and Testing:
  - Use Case: Data masking is used to provide developers and testers with realistic datasets for software testing without exposing actual sensitive data, such as personal customer details.
- Data Analytics and Reporting:
  - Use Case: Data masking allows data analysts to work with realistic data while maintaining privacy compliance. The masked data maintains the statistical value needed for analysis but hides sensitive information.
- Healthcare Data Protection:
  - Use Case: In the healthcare industry, data masking is used to comply with regulations like HIPAA by anonymizing sensitive patient information in test environments or for research purposes.
- Cloud Data Security:
  - Use Case: As companies move data to the cloud, data masking ensures that sensitive data is protected during migrations and cloud usage, keeping compliance with privacy laws.



by HIMANSHU GOHITE

THANKYOU

References

1. https://www.geeksforgeeks.org/cryptography-and-its-types/
2. https://www.redhat.com/en/blog/brief-history-cryptography
3. https://www.geeksforgeeks.org/caesar-cipher-in-cryptography/
4. https://www.geeksforgeeks.org/playfair-cipher-with-examples/
5. https://www.fortinet.com/resources/cyberglossary/data-security#:~:text=Data%20security%20is%20the%20process,and%20organizations'%20policies%20and%20procedures.
6. https://www.tutorialspoint.com/cryptography/cryptography_hash_functions.htm
7. https://www.tutorialspoint.com/cryptography/data_encryption_standard.htm
8. https://www.geeksforgeeks.org/advanced-encryption-standard-aes/
9. https://www.geeksforgeeks.org/block-cipher-modes-of-operation/
10. https://www.tutorialspoint.com/cryptography/message_authentication.htm
11. https://www.geeksforgeeks.org/what-is-asymmetric-encryption/
12. https://en.wikipedia.org/wiki/Diffie%E2%80%93Hellman_key_exchange