dINFORMATIONAL SECURITY

CONTENTS

Security Attack:

A security attack specifically targets the security mechanisms, confidentiality, integrity, or availability of data, systems, or networks. It aims to breach defenses, gain unauthorized access, steal sensitive information, disrupt operations, or cause other detrimental effects. Security attacks can take many forms and may exploit vulnerabilities in technology, exploit human weaknesses, or combine both approaches.

1. Active Attacks

Definition: Active attacks involve actions that modify or disrupt the normal operation of a system or network. These attacks attempt to alter data, introduce malicious code, or interrupt communication.

Examples: Man-in-the-Middle (MITM) attacks, Denial of Service (DoS) attacks, Distributed Denial of Service (DDoS) attacks, session hijacking, and packet injection.

Methods: In MITM attacks, an attacker intercepts and possibly alters communication between two parties without their knowledge. DoS and DDoS attacks flood a network or server with traffic, rendering it inaccessible to legitimate users.

2. Passive Attacks

Definition: Passive attacks involve monitoring and eavesdropping on data transmissions without altering or disrupting them. The goal is typically to obtain sensitive information without the target's knowledge.

Examples: Packet sniffing, wiretapping, and traffic analysis.

Methods: Packet sniffers capture and analyze network traffic, allowing attackers to view transmitted data, including usernames, passwords, and other confidential information. Traffic analysis involves analyzing patterns in network traffic to infer information about the communication.

3. Brute Force Attacks

Definition: Brute force attacks involve attempting all possible combinations of characters or encryption keys until the correct one is found. These attacks are straightforward but can be time-consuming and resource-intensive.

Examples: Password cracking, cryptographic key brute forcing.

Methods: In password cracking, an attacker tries numerous combinations of characters until the correct password is discovered. Similarly, cryptographic key brute forcing involves trying all possible encryption keys until the correct one is found, allowing unauthorized access to encrypted data.

4. Cryptanalysis:

Definition: Cryptanalysis involves analyzing cryptographic systems to uncover weaknesses or vulnerabilities that can be exploited to decrypt encrypted data without knowing the correct key.

Examples: Frequency analysis, chosen plaintext attacks, known plaintext attacks.

Methods: Frequency analysis exploits patterns in the frequency of letters or symbols in a ciphertext to deduce the encryption key. Chosen plaintext and known plaintext attacks involve obtaining ciphertexts corresponding to specific plaintexts to analyze and deduce information about the encryption key.

5. Insider Attacks

Definition: Insider attacks occur when individuals within an organization misuse their privileges or access to compromise security. These individuals may be employees, contractors, or other trusted entities with legitimate access to systems or data.

Examples: Unauthorized data access, data theft, sabotage, espionage.

Methods: Insiders may abuse their access privileges to steal sensitive information, intentionally introduce malware or vulnerabilities, or disrupt system operations. They can exploit their knowledge of the organization's infrastructure and security measures to evade detection and carry out malicious activities.

Security Services

Security services in the context of cybersecurity are designed to protect the confidentiality, integrity, and availability (CIA) of data and resources. Additionally, another set of services, known as AAA, focuses on authentication, authorization, and accounting. Let's delve into each of these service categories:

CIA Triad:

- **Confidentiality:** Confidentiality ensures that sensitive information is only accessible to authorized individuals or systems. It involves measures such as encryption, access controls, and data masking to prevent unauthorized disclosure.
- **Integrity:** Integrity ensures that data remains accurate, consistent, and trustworthy throughout its lifecycle. It involves mechanisms to detect and prevent unauthorized alterations or modifications to data, such as checksums, digital signatures, and integrity checks.
- **Availability:** Availability ensures that data and resources are accessible and usable when needed by authorized users. It involves measures to prevent and mitigate disruptions, downtime, or denial of service attacks, such as redundancy, failover, and disaster recovery planning.

AAA Services:

- **Authentication:** Authentication verifies the identity of users or systems attempting to access resources. It ensures that users are who they claim to be by validating credentials such as usernames, passwords, biometrics, or digital certificates.
- **Authorization:** Authorization determines what actions or resources users are allowed to access once they have been authenticated. It involves enforcing policies and permissions to control access rights based on roles, privileges, or other attributes.
- **Accounting (or Auditing):** Accounting tracks and logs activities related to system access and resource usage. It involves recording information such as login attempts, resource access, changes to configurations, and other security-relevant events for monitoring, analysis, and forensic purposes.

Security Terminology

Understanding security terminologies is crucial in comprehending and addressing cybersecurity threats effectively. Here are explanations of four key terms: adversary, vulnerability, threat, and attack:

1. Adversary

Definition: An adversary, also known as an attacker or threat actor, refers to an individual, group, organization, or automated system that poses a threat to the security of an entity's assets, such as data, systems, or networks.

Characteristics: Adversaries may have various motives, including financial gain, espionage, activism, sabotage, or personal gratification. They exploit vulnerabilities and employ tactics, techniques, and procedures (TTPs) to achieve their objectives.

Examples: Hackers, cybercriminals, state-sponsored actors, insiders, and malware are common types of adversaries in cybersecurity.

2. Vulnerability:

Definition: A vulnerability is a weakness or flaw in a system, application, network, or process that could be exploited by adversaries to compromise the security or integrity of assets.

Characteristics: Vulnerabilities can arise due to programming errors, misconfigurations, design flaws, or inadequate security controls. They may exist in software, hardware, firmware, or human behavior.

Examples: Buffer overflow, SQL injection, misconfigured permissions, unpatched software, weak passwords, and social engineering are examples of vulnerabilities

3. Threat:

Definition: A threat is any potential danger or circumstance that can exploit vulnerabilities, leading to harm, damage, or disruption to an organization's assets or operations.

Characteristics: Threats can be categorized based on their nature, origin, or impact. They may include natural events, human actions, technological failures, or malicious activities.

Examples: Malware infections, phishing attacks, insider threats, data breaches, natural disasters, and hardware failures are examples of threats that organizations may face.

4. Attack:

Definition: An attack is a deliberate and malicious action taken by an adversary to exploit vulnerabilities and compromise the security of an entity's assets. It involves unauthorized access, manipulation, or destruction of data or resources.

Characteristics: Attacks may target confidentiality, integrity, availability, or other security attributes of information systems. They can be executed through various methods and techniques, ranging from software exploits to social engineering tactics.

Examples: Denial of Service (DoS) attacks, ransomware infections, password cracking, privilege escalation, and phishing scams are common examples of cyber attacks.

Assumption and Trust:

Assumption: Assumptions are the foundational beliefs or conditions upon which security policies, mechanisms, and decisions are based. They often involve expectations about the behavior of systems, users, or entities within a particular context.

Trust: Trust refers to the confidence or reliance placed on systems, components, entities, or individuals to behave as expected and to fulfill their intended functions securely and reliably.

Relation: Assumptions influence the level of trust placed in various components of a system or network. Trust decisions are made based on the alignment of observed behaviors with expected assumptions.

Security Policy and Mechanism:

Security Policy: A security policy is a formal statement or set of rules that define the requirements, constraints, and responsibilities related to protecting an organization's assets, ensuring compliance with regulations, and mitigating risks.

Security Mechanism: Security mechanisms are technical or procedural controls implemented to enforce security policies, protect assets, and mitigate threats. They include encryption, access controls, authentication methods, intrusion detection systems, firewalls, and security protocols

Threat Analysis:

Threat: A threat is any potential danger or circumstance that can exploit vulnerabilities and cause harm or disruption to an organization's assets, operations, or objectives.

Threat Analysis: Threat analysis involves identifying, assessing, and prioritizing threats based on their likelihood and potential impact. It aims to understand the nature, capabilities, motivations, and tactics of adversaries, as well as the vulnerabilities they may exploit.

Attacker Modeling:

Attacker: An attacker, also known as a threat actor or adversary, is an individual, group, organization, or automated system that poses a threat to the security of an entity's assets.

Modeling: Attacker modeling involves creating abstract representations or profiles of potential adversaries, including their motivations, capabilities, resources, and behaviors. It helps in understanding the tactics, techniques, and procedures (TTPs) that adversaries may employ to achieve their objectives.

Purpose: Attacker modeling assists in designing effective security measures, controls, and response strategies by anticipating potential threats and adversaries' likely actions.

Authentication Mechanisms

Password-Based Authentication:

Definition: Password-based authentication is a traditional method where users prove their identity by providing a combination of characters (i.e., a password) known only to them.

Process:

1. **User Input:** The user enters their username and password into the authentication interface.
2. **Comparison:** The system verifies the entered password against the stored password associated with the provided username.
3. **Authentication:** If the entered password matches the stored password, the user is authenticated and granted access.

Advantages:

- Simplicity: Easy for users to understand and use.
- Wide Adoption: Ubiquitous across various systems and applications.

Challenges:

- Password Security: Weak passwords, reuse, and storage vulnerabilities can compromise security.
- Credential Theft: Passwords can be stolen through various means, such as phishing or data breaches.
- User Compliance: Users may struggle to create and remember complex passwords, leading to insecure practices.

Biometric Authentication:

Definition: Biometric authentication uses unique physical or behavioral characteristics of individuals to verify their identity.

Examples of Biometrics: Fingerprints, facial recognition, iris scans, voice recognition, palm prints, and behavioral biometrics (e.g., typing patterns, gait analysis).

Process:

1. **Biometric Capture:** The user's biometric data is captured using specialized sensors or devices.
2. **Comparison:** The captured biometric data is compared with previously stored biometric templates.
3. **Authentication:** If the captured biometric data matches the stored template(s) within an acceptable threshold, the user is authenticated and granted access.

Advantages:

- Strong Security: Biometric characteristics are unique and difficult to replicate, enhancing security.
- Convenience: Users don't need to remember passwords, making authentication more convenient.
- Resistance to Theft: Biometric traits are difficult to steal or share compared to passwords.

Challenges:

- Privacy Concerns: Biometric data is sensitive and requires careful handling to protect privacy.
- Accuracy and Reliability: Biometric systems may produce false positives or false negatives, impacting user experience.
- Deployment Costs: Biometric authentication may require specialized hardware and infrastructure, increasing implementation costs.

Challenge-Response Authentication

Definition: Challenge-Response Authentication is a method where the authenticating party (like a server) challenges the user (client) to provide specific information or perform a task that only the legitimate user should be able to complete.

Example: When logging into your online banking account, the bank's server sends you a challenge, like asking for a unique code generated on your mobile app. You provide the correct response (the code), proving you're the legitimate user.

Process:

1. The server sends a challenge to the client, such as a unique code or encrypted message.
2. The client generates a response based on the challenge using a secret key or algorithm.
3. The server verifies the response provided by the client. If it matches the expected value, authentication is successful.

Advantages:

- Enhanced Security: Difficult for attackers to intercept or guess the response without knowing the secret key or algorithm.
- Dynamic Authentication: Challenges and responses can change each time, adding an extra layer of security.

Challenges:

- Implementation Complexity: Requires additional computation and infrastructure to handle challenges and responses.
- Potential for Replay Attacks: If an attacker intercepts a challenge-response pair, they may be able to replay it to gain unauthorized access.

One-Way Authentication

Definition: One-Way Authentication, also known as unilateral authentication, involves only one party proving its identity to another party without the need for mutual verification.

Example: When you log into a website using a username and password, you're proving your identity to the website's server. The server verifies your credentials but doesn't need to prove its identity to you.

Process:

- The client (user) proves its identity to the server without requiring the server to prove its identity in return.

Advantages:

- Simplified Process: Reduces complexity by focusing on one party authenticating the other.
- Lower Overhead: Eliminates the need for mutual verification, reducing computational and communication overhead.

Challenges:

- Vulnerable to Man-in-the-Middle Attacks: Without mutual verification, attackers can intercept communication and impersonate one of the parties.
- Limited Trust Assurance: The party being authenticated (e.g., the client) must trust that the other party (e.g., the server) is legitimate without verifying it.

Mutual Authentication

Definition: Mutual Authentication, also known as two-way authentication, requires both parties involved in a communication or transaction to authenticate each other, establishing trust and ensuring the identities of both parties are verified.

Example: When you connect to a secure website (HTTPS), your browser verifies the server's identity using its digital certificate, and the server verifies your browser's identity. This ensures that both parties are who they claim to be.

Process:

- Both the client (e.g., your web browser) and the server (e.g., the website) exchange digital certificates or credentials to verify each other's identity.
- After successful verification, both parties can trust each other and proceed with the communication or transaction securely.

Advantages:

- Enhanced Security: Provides assurance that both parties are legitimate, reducing the risk of man-in-the-middle attacks.
- Trust Establishment: Establishes trust between communicating parties, ensuring that sensitive information can be exchanged securely.

Challenges:

- Complexity: Requires additional configuration and management to implement mutual authentication, increasing system complexity.
- Performance Overhead: The exchange of certificates and verification processes can introduce latency, affecting system performance.

Multifactor Authentication (MFA)

Definition: Multifactor Authentication (MFA) is a method that requires users to provide multiple forms of identification or authentication factors to access a system, application, or resource.

Example: When logging into your email account with MFA enabled, you may need to enter your password (something you know) and then verify your identity using a one-time code sent to your smartphone (something you have).

Process:

- Users are required to provide two or more authentication factors, typically from the following categories:
  - Something the user knows (e.g., password, PIN)
  - Something the user has (e.g., smartphone, security token)
  - Something the user is (e.g., biometric traits like fingerprints, facial recognition)
- After successfully presenting multiple factors, the user is authenticated and granted access.

Advantages:

- Enhanced Security: Requires attackers to overcome multiple barriers to gain unauthorized access, reducing the risk of compromised accounts.
- Versatility: Allows organizations to choose authentication factors based on their security needs and user preferences.

Challenges:

- User Experience: Users may find MFA more cumbersome than single-factor authentication, potentially impacting adoption rates.
- Implementation Complexity: Setting up and managing MFA systems requires additional resources and expertise, increasing deployment costs.

Cryptography

Cryptography is the practice and study of techniques for securing communication and data against adversaries. It involves the use of mathematical algorithms and principles to transform plaintext (readable data) into ciphertext (encoded data), making it unintelligible to unauthorized users. Cryptography plays a crucial role in ensuring the confidentiality, integrity, and authenticity of information in various contexts, including digital communication, e-commerce, data storage, and authentication.

5. Plaintext:

Definition: Plaintext refers to the original, readable, and unencrypted form of data or information.

Characteristics: Plaintext can include text, numbers, symbols, or any other type of data that is understandable to humans.

Example: In the context of messaging, plaintext is the message as it's typed or written before any encryption is applied. For example, "Hello, how are you?" is plaintext.

Usage: Plaintext is what users typically interact with and understand. However, it's vulnerable to interception or eavesdropping during transmission, which is why encryption is used to protect sensitive information.

6. Cyphertext:

Definition: Ciphertext is the encrypted and unreadable form of data or information resulting from applying encryption algorithms to plaintext.

Characteristics: Ciphertext appears as a random or scrambled sequence of characters that is unintelligible without the proper decryption key.

Example: Using encryption algorithms like AES or RSA, plaintext messages are transformed into ciphertext, such as "L#8dkf$%J2P@3e!".

Usage: Ciphertext is used to protect the confidentiality of sensitive information during transmission or storage. Only authorized parties with the decryption key can transform ciphertext back into plaintext.

7. Encryption:

Definition: Encryption is the process of transforming plaintext (readable data) into ciphertext (encoded data) using an encryption algorithm and a secret key. It ensures the confidentiality and security of sensitive information by making it unreadable to unauthorized users.

Process:

Encryption: Plaintext is input into an encryption algorithm along with a secret key, resulting in ciphertext.

Example: Using an encryption algorithm like AES (Advanced Encryption Standard) with a secret key, the plaintext "Hello, how are you?" might be transformed into ciphertext like "5K0x3lNzPf#2&1@!".

Purpose: Encryption protects data during transmission or storage, preventing unauthorized access and ensuring privacy and security.

8. Decryption:

Definition: Decryption is the process of reversing encryption, transforming ciphertext back into plaintext using a decryption algorithm and the appropriate key.

Process:

Decryption: Ciphertext is input into a decryption algorithm along with the correct key, resulting in plaintext.

Example: Using the same decryption algorithm and key used for encryption, the ciphertext "5K0x3lNzPf#2&1@!" would be transformed back into the original plaintext "Hello, how are you?".

Purpose: Decryption allows authorized users to access and interpret encrypted data, restoring it to its original readable form.

9. Symmetric Encryption:

Definition: Symmetric encryption, also known as secret-key or single-key encryption, uses a single secret key for both encryption and decryption processes.

Characteristics:

Same Key: The same secret key is used for both encryption and decryption.

Efficiency: Symmetric encryption algorithms are typically faster and more efficient than asymmetric encryption.

Example: AES (Advanced Encryption Standard), DES (Data Encryption Standard), and 3DES (Triple DES) are examples of symmetric encryption algorithms.

Usage: Symmetric encryption is used for securing data transmission, such as encrypting files, messages, or network traffic.

10. Asymmetric Encryption:

Definition: Asymmetric encryption, also known as public-key encryption, uses a pair of keys: a public key for encryption and a private key for decryption.

Characteristics:

Key Pairs: A public-private key pair is used for encryption and decryption.

Security: The public key can be shared openly, while the private key remains secret.

Example: RSA (Rivest-Shamir-Adleman) and ECC (Elliptic Curve Cryptography) are examples of asymmetric encryption algorithms.

Usage: Asymmetric encryption is used for key exchange, digital signatures, secure communication, and authentication in various cryptographic protocols like SSL/TLS, SSH, and PGP

11. Block Cypher:

Definition: Block ciphers are symmetric encryption algorithms that process fixed-size blocks of data at a time. Each block of plaintext is encrypted into a corresponding block of ciphertext using a secret key.

Characteristics:

Fixed Block Size: Block ciphers operate on fixed-size blocks of data, typically 64 or 128 bits.

Key-dependent: Encryption and decryption depend on a secret key shared between the sender and recipient.

Block Modes: Various block cipher modes of operation, such as Electronic Codebook (ECB), Cipher Block Chaining (CBC), and Counter (CTR), determine how blocks are encrypted and combined.

Example: AES (Advanced Encryption Standard) is a widely used block cipher algorithm that operates on 128-bit blocks and supports key sizes of 128, 192, or 256 bits.

Usage: Block ciphers are commonly used for encrypting data at rest, such as files, disk partitions, and databases

12. Stream Cypher:

Definition: Stream ciphers are symmetric encryption algorithms that encrypt data one bit or byte at a time, usually by generating a pseudorandom keystream based on a secret key.

Characteristics:

13. Keystream Generation: Stream ciphers generate a keystream of pseudorandom bits or bytes based on the secret key and an initialization vector (IV).
14. Bit-by-Bit Encryption: Each plaintext bit or byte is combined with a corresponding keystream element to produce ciphertext.
15. Synchronization: Synchronization between sender and receiver is critical for stream ciphers to ensure that both parties generate the same keystream.

Example: RC4 (Rivest Cipher 4) is a well-known stream cipher historically used in protocols like WEP (Wireless Equivalent Privacy) and SSL/TLS (Secure Sockets Layer/Transport Layer Security).

Usage: Stream ciphers are suitable for encrypting real-time data streams, such as network traffic, audio, and video transmissions.

9. Historical Cipher Techniques:

Substitution Ciphers: Substitution ciphers replace plaintext characters with ciphertext characters based on a fixed substitution table or rule. Examples include Caesar cipher, Atbash cipher, and ROT13.

Transposition Ciphers: Transposition ciphers rearrange the order of characters or blocks of plaintext to produce ciphertext. Examples include Rail Fence cipher and Columnar Transposition cipher.

Polyalphabetic Ciphers: Polyalphabetic ciphers use multiple alphabets or cipher alphabets to encrypt plaintext characters, providing stronger security than simple substitution ciphers. Examples include Vigenère cipher and Playfair cipher.

Mechanical Ciphers: Historical cipher techniques also include mechanicaldevices such as the Enigma machine, used by the German military during World War II for encryption and decryption of messages.

Usage: Historical cipher techniques were used for centuries before the advent of modern cryptographic algorithms. While many are now obsolete due to vulnerabilities, they remain of historical and educational interest.

Cryptographic Algorithm:

A cryptographic algorithm is a set of mathematical procedures and rules used to encrypt and decrypt data securely. These algorithms form the foundation of cryptographic systems, ensuring the confidentiality, integrity, and authenticity of information. Cryptographic algorithms can be categorized based on their function, such as encryption, hashing, or digital signatures.

Operation OF AES:

The Advanced Encryption Standard (AES) is a symmetric encryption algorithm widely adopted for securing sensitive data. It operates on fixed-size blocks of data, typically 128 bits, using a key of 128, 192, or 256 bits. Here's a detailed overview of the operation of AES:

1. **Key Expansion:**
   - AES starts with a single secret key, which is expanded into a key schedule containing multiple round keys. The number of round keys depends on the key size: 10 rounds for a 128-bit key, 12 rounds for a 192-bit key, and 14 rounds for a 256-bit key.
   - Each round key is derived from the original key through a series of transformations, including key expansion, substitution, and mixing operations.
2. **Initial Round:**
   - The plaintext is divided into a block of 128 bits.
   - The initial round of AES consists of adding the round key to the plaintext block. Each byte of the plaintext block is XORed with a corresponding byte of the round key.
3. **Main Rounds (Encryption):**
   - AES consists of multiple rounds (10, 12, or 14 rounds depending on the key size) of transformations, where each round operates on the state formed by the previous round's output.
   - Each round consists of four main operations: SubBytes, ShiftRows, MixColumns, and AddRoundKey.
     - **SubBytes:** Each byte of the state is replaced with a corresponding byte from the S-box, a predefined lookup table.
     - **ShiftRows:** The nh bytes in each row of the state are shifted cyclically to the left. The first row remains unchanged, the second row is shifted one position to the left, the third row is shifted two positions, and the fourth row is shifted three positions.
     - **MixColumns:** Each column of the state is mixed using a matrix multiplication operation with a fixed matrix known as the MixColumns matrix.
     - **AddRoundKey:** The round key derived from the key schedule is XORed with the state.
   - These operations are repeated for the specified number of rounds, except for the final round.
4. **Final Round (Encryption):**

   - The final round is similar to the main rounds but excludes the MixColumns operation.
   - It consists of SubBytes, ShiftRows, and AddRoundKey operations applied to the state.
5. **Decryption:**
   - AES decryption involves the same operations as encryption, but in the reverse order.
   - Each decryption round consists of operations similar to encryption but with inverses of the SubBytes, ShiftRows, and MixColumns transformations.
   - The decryption key schedule is derived from the encryption key schedule.
6. **Output:**
   - After all rounds (including the initial and final rounds) are completed, the resulting state represents the ciphertext (for encryption) or plaintext (for decryption).

AES provides a high level of security and efficiency, making it suitable for various applications, including securing communications, protecting data at rest, and ensuring the integrity of digital content. Its well-defined structure and standardized operations contribute to its widespread adoption and interoperability across different systems and platforms.

AES VARIANTS

There are several variants or modes of operation for the Advanced Encryption Standard (AES), each offering different characteristics and suitability for various cryptographic applications. Here are some of the commonly used AES variants:

1. **Electronic Codebook (ECB):**
   - ECB is the simplest mode of operation, where each block of plaintext is encrypted independently with the same key.
   - However, ECB does not provide confidentiality for identical blocks of plaintext, making it vulnerable to pattern analysis attacks.
   - Due to its lack of security for certain types of data, ECB is not recommended for general use.
2. **Cipher Block Chaining (CBC):**
   - CBC is a widely used mode of operation that provides confidentiality and protection against certain types of attacks.
   - In CBC mode, each plaintext block is XORed with the previous ciphertext block before encryption, creating a dependency chain.
   - CBC requires an initialization vector (IV) to ensure randomness and prevent patterns in the ciphertext.
   - While CBC offers stronger security than ECB, it may be susceptible to padding oracle attacks and requires careful IV management.
3. **Cipher Feedback (CFB):**

- CFB mode converts a block cipher into a self-synchronizing stream cipher, where plaintext is encrypted in units smaller than the block size.
- CFB allows for variable-length plaintext input and is resistant to error propagation.
- However, CFB mode may suffer from a performance overhead due to its self-synchronizing nature.

4. **Output Feedback (OFB):**
    - OFB mode transforms a block cipher into a synchronous stream cipher by encrypting a fixed-size feedback register rather than plaintext.
    - OFB does not require padding and allows for parallel encryption and decryption.
    - However, OFB may be vulnerable to bit-flipping attacks if the feedback register is reused.
5. **Counter (CTR):**
    - CTR mode turns a block cipher into a stream cipher by encrypting a counter value to produce a keystream.
    - CTR mode offers parallel encryption and decryption, making it suitable for high-performance applications.
    - CTR mode is resistant to padding oracle attacks and does not require padding.
    - Additionally, CTR mode allows for random access to encrypted data without decrypting the entire ciphertext.
6. **Galois/Counter Mode (GCM):**
    - GCM is an authenticated encryption mode that combines CTR mode with polynomial hashing for authentication.
    - GCM provides confidentiality, integrity, and authenticity in a single mode of operation.
    - It is widely used in network security protocols like TLS and IPsec due to its efficiency and security properties.

Each AES variant offers different trade-offs in terms of security, performance, and functionality. The choice of mode depends on the specific requirements and constraints of the cryptographic application, including security goals, data transmission characteristics, and computational resources.