Content from different reference book suggested by Prof. Prakash Sir and mentioned in CNS Syllabus file, Presentations, Notes provided in 2023

Cheat sheet summarizing the information you provided on "Types of Attacks," "Security Services," "Security Mechanisms," "Security Attacks," "An Information Security Model: C.I.A. Triangle," and "Confusion and Diffusion":

Types of Attacks

1. Fabrication
   - Description: Creation of false or illegitimate information within a system.
   - Examples: SQL Injection, Route Injection, Email Spoofing.
   - Mitigation: Use Authentication, Authorization, Firewalls, Digital Signatures.

2. Interception
   - Description: Unauthorized access to confidential information.
   - Examples: Eavesdropping, Wiretapping, Packet Sniffing.
   - Mitigation: Encryption (SSL, VPN), Traffic Padding.

3. Interruption
   - Description: Targeting availability of network services.
   - Examples: DoS attacks, Cutting communication lines.
   - Mitigation: Firewalls, Backups, Replication.

4. Modification
   - Description: Compromising integrity through changes, insertions, or deletions.
   - Examples: Modifying messages, Changing data files.
   - Mitigation: Intrusion Detection Systems (IDS), Encryption, Checksums.

Security Services

1. Confidentiality
   - Objective: Limit information access to authorized parties.
   - Examples: Data disclosure prevention.

2. Authentication
   - Objective: Verify message origin for identity assurance.

3. Integrity
   - Objective: Prevent unauthorized modification.
   - Actions: Writing, changing, deleting data.

4. Non-repudiation
   - Objective: Prevent message denial by sender or receiver.

5. Access Control
   - Objective: Regulate resource access.

6. Availability
   - Objective: Ensure resource access when needed.

Security Mechanisms

1. Encipherment
   - Purpose: Protect confidentiality.
   - Description: Transform data into unreadable format.
   - Use Cases: Secure data transmission, data at rest.

2. Digital Signature
   - Purpose: Ensure authenticity and integrity.
   - Description: Verify origin and integrity of digital documents.
   - Use Cases: Authenticating emails, documents, transactions.

3. Access Control
   - Purpose: Control system access.
   - Description: Restrict and regulate resource access.

Security Attacks

1. Interruption
   - Description: Destroy or render assets unavailable.
   - Examples: Destruction of hardware, cutting communication lines.

2. Interception
   - Description: Unauthorized access to system assets.
   - Examples: Wiretapping, unauthorized data access.

3. Modification
   - Description: Unauthorized tampering with assets.
   - Examples: Changing data, altering program code.

4. Fabrication
   - Description: Insert counterfeit objects or data.
   - Examples: Inserting false messages, adding fake records.

An Information Security Model: C.I.A. Triangle

Confidentiality: Limit access to authorized parties.
   - Measures: Classification, secure storage, education.

Integrity: Ensure data remains whole and uncorrupted.
   - Threats: Corruption during storage, transmission.
   - Measures: Digital signatures, hashing, code review.

Availability: Ensure access to assets when needed.
- Measures: RAID, redundant components, server clusters.

Confusion and Diffusion

Confusion:
- Focus: Complex relationship between key and cipher text.
- Goal: Obscure the relationship between CT and PT.
- Properties: No info from CT about PT, key, etc.

Diffusion:
- Focus: Spread changes in PT to CT.
- Goal: Small PT change causes significant CT change.
- Properties: Single bit change in PT cascades in CT.

Stream Cipher vs. Block Cipher:
- Stream Cipher: Bit by bit, faster, e.g., Vernam Cipher.
- Block Cipher: Fixed-length blocks, slower, e.g., AES.

Substitution vs. Transposition:
- Substitution: Change character identity.
- Transposition: Change position, not identity.
- Substitution Forms: Monoalphabetic, polyalphabetic.
- Transposition Forms: Keyless, keyed.

Substitution vs. Transposition Characteristics:
- Vulnerability to frequency analysis.
- Difficulty of implementation.
- Impact on plaintext characteristics.

Feel free to use this cheat sheet as a reference for information security concepts and techniques.
Here is a cheat sheet summarizing the Classical Encryption Techniques you provided:

Caesar Cipher
- Description: A substitution cipher that shifts each letter in the plaintext by a fixed number of positions down or up the alphabet.
- Encryption: Replace each letter with the letter shifted by the key. Wrap around the alphabet if needed.
- Decryption: Reverse the process by shifting each letter back by the key.
- Example: "HELLO" with key=3 becomes "KHOOL."

Monoalphabetic Cipher
- Description: A substitution cipher where each letter in the plaintext is replaced by a fixed corresponding letter.
- Encryption: Use a fixed mapping between plaintext and ciphertext alphabets.
- Decryption: Reverse the mapping to obtain the original plaintext.

- Example: "HELLO" with mapping becomes "ROVVY."

Polyalphabetic Cipher (Vigenère Cipher)
- Description: A polyalphabetic substitution cipher that uses a keyword to determine shifting alphabets.
- Encryption: Shift each letter in the plaintext by the corresponding letter in the keyword.
- Decryption: Reverse the process by shifting each letter back using the keyword.
- Example: "HELLO" with keyword "KEY" becomes "OIEQG."

Vernam Cipher (One-Time Pad)
- Description: Uses a random key as long as the plaintext, combined using XOR to produce ciphertext.
- Encryption: XOR each character of plaintext with the corresponding character in the key.
- Decryption: XOR the ciphertext with the same key to recover the plaintext.
- Example: "HELLO" with key "WORLD" becomes "XHP0V."

Playfair Cipher
- Description: Substitutes digraphs of letters based on a key matrix.
- Encryption: Use a key matrix to replace each pair of letters in the plaintext.
- Decryption: Reverse the substitution using the same key matrix.
- Example: "HELLO WORLD" with a key becomes "GMLY."

Hill Cipher
- Description: Encrypts plaintext using matrix multiplication with a key matrix.
- Encryption: Divide the plaintext into blocks, multiply by the key matrix, and reduce modulo the alphabet size.
- Decryption: Multiply by the inverse of the key matrix to recover the plaintext.
- Example: "HELLO" with a 2x2 key matrix becomes "GMLY."

Rail Fence Cipher (Zigzag Cipher)
- Description: Rearranges characters in a zigzag pattern based on a key.
- Encryption: Write the message diagonally in rows, then read columns.
- Decryption: Write the ciphertext in the zigzag pattern and read the original message.
- Example: "HELLO WORLD" with a key of 3 becomes "HORDELWLLO."

Simple Columnar Transposition
- Description: Rearranges characters based on a key by writing them in a grid.
- Encryption: Write the message in columns according to the key, then read the columns.
- Decryption: Create a grid with the key, write ciphertext in columns, then read rows to get plaintext.

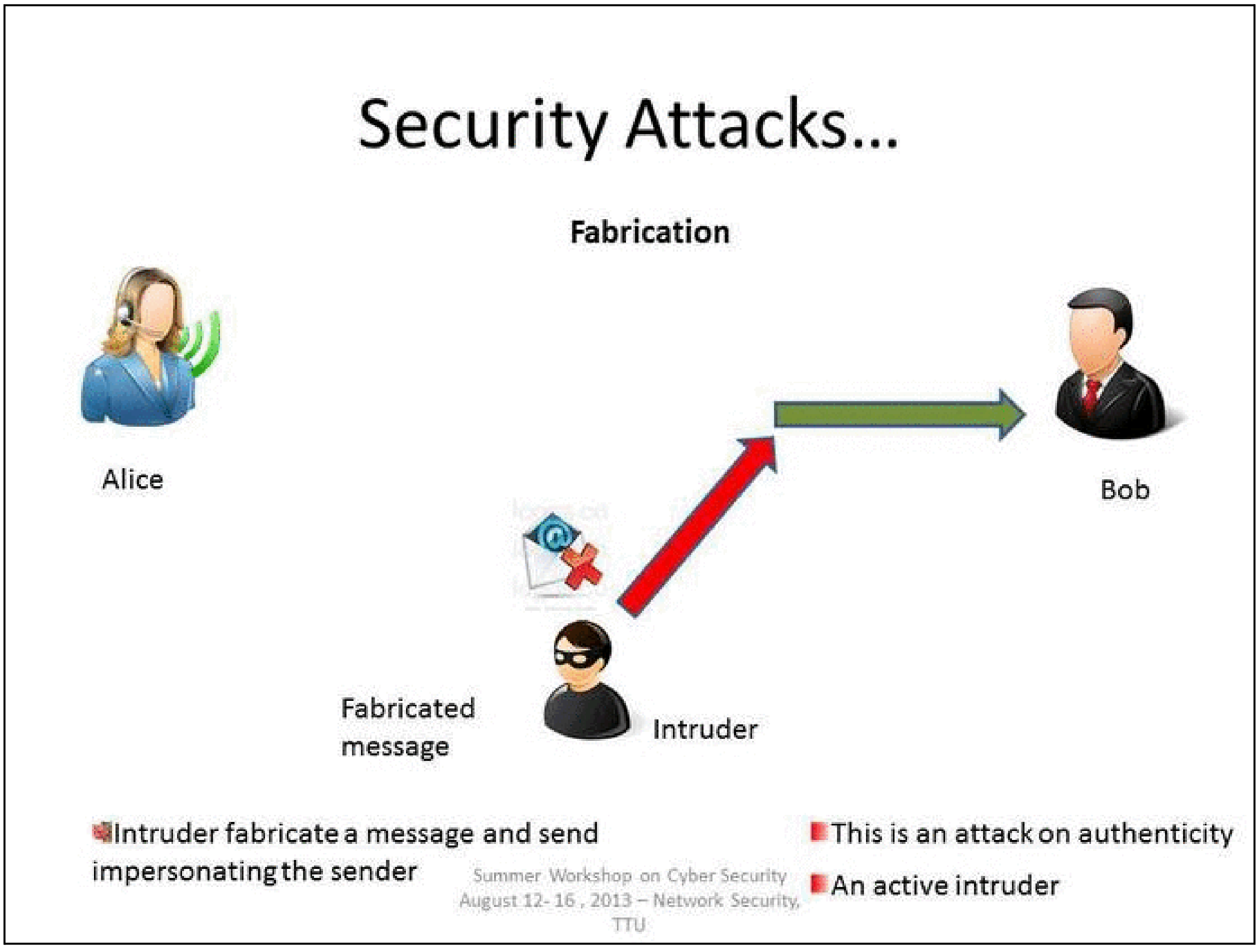
These classical encryption techniques serve as historical examples of basic encryption concepts and have varying levels of security. Modern encryption methods are used for secure communications due to their increased complexity and resistance to attacks.

Types of Attacks

In the context of Information Security, attacks can be broadly categorized into four main categories:

1. Fabrication

- Fabrication attacks involve the creation of false or illegitimate information, processes, communications, or other data within a system.
- Attackers may insert fabricated data alongside genuine data to compromise a system's integrity.
- Examples of Fabrication attacks include SQL Injection, Route Injection, User/Credential Counterfeiting, Log/Audit Trail Falsification, and Email Spoofing.
- Mitigation measures: Use Authentication and Authorization mechanisms, deploy Firewalls, and employ Digital Signatures to ensure the authenticity of digital messages or documents.



Security Attacks…

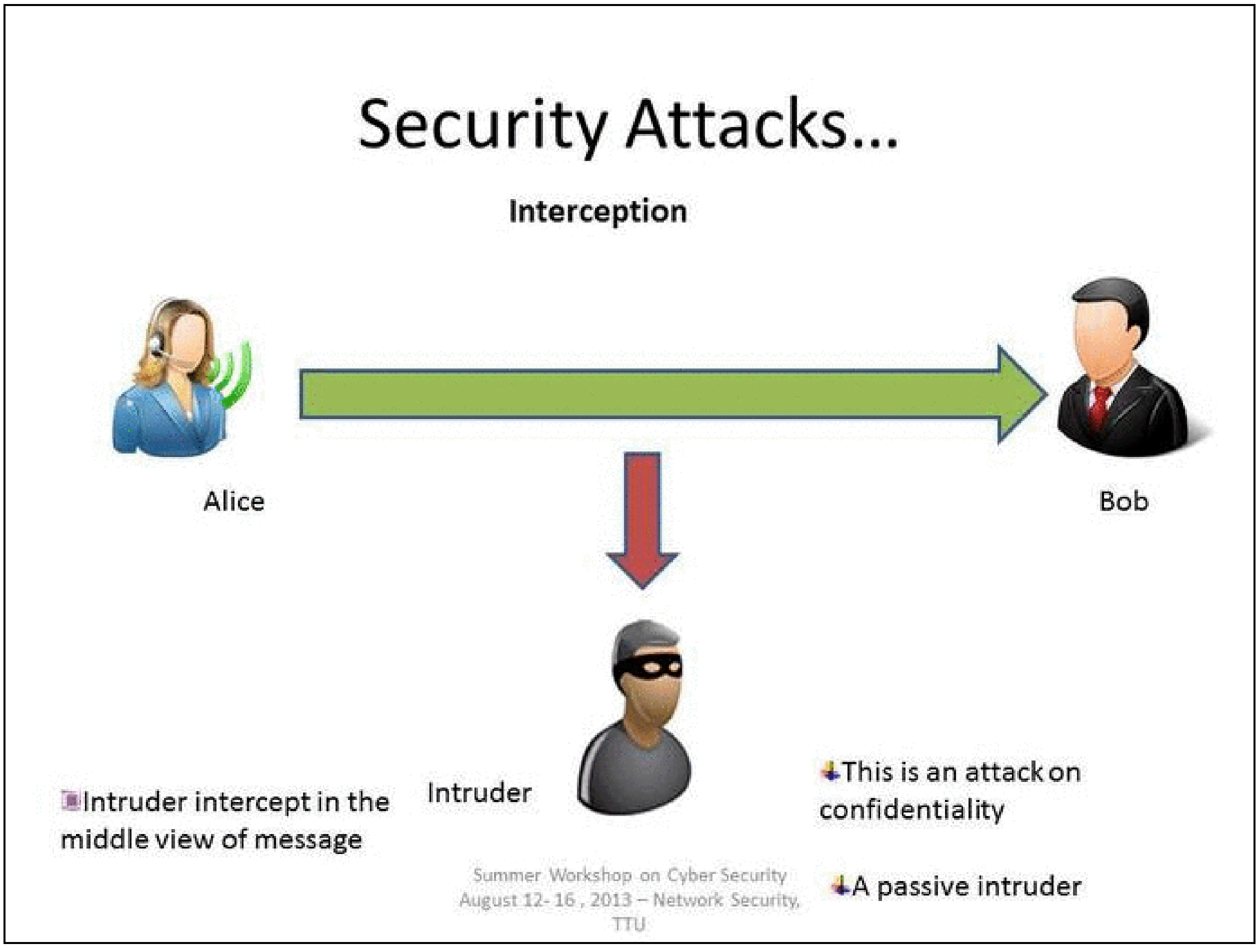
Fabrication

- Intruder fabricate a message and send impersonating the sender
- This is an attack on authenticity
- An active intruder

Summer Workshop on Cyber Security August 12- 16 , 2013 – Network Security, TTU

2. Interception

- Interception attacks occur when unauthorized individuals gain access to confidential or private information, compromising the confidentiality aspect of the CIA Triad (Confidentiality, Integrity, Availability).
- Examples of Interception attacks include eavesdropping on communication, wiretapping telecommunications networks, illicit copying of files or programs, obtaining copies of messages for later replay, and packet sniffing and keylogging to capture data from computer systems or networks.
- Mitigation measures: Use Encryption technologies such as SSL, VPN, 3DES, BPI+ to encrypt information flow, and employ Traffic Padding to create continuous ciphered text, making it difficult for attackers to distinguish between data flow and noise.



Security Attacks…

Interception

- Intruder intercept in the middle view of message

- This is an attack on confidentiality
- A passive intruder

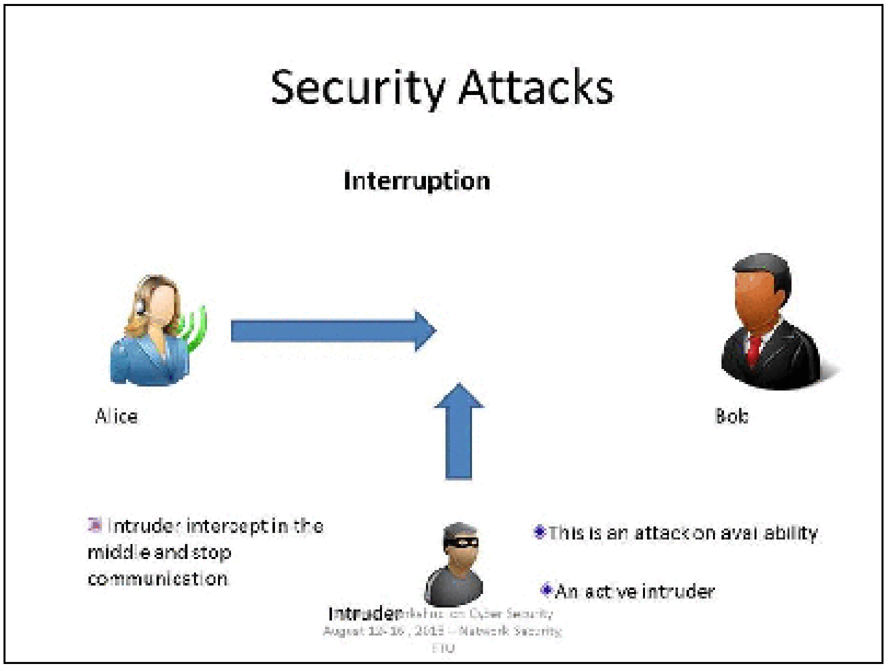
Summer Workshop on Cyber Security
August 12- 16 , 2013 – Network Security,
TTU

3. Interruption

  - Interruption attacks target the availability of network services by making them degraded or unavailable for legitimate use.

  - Examples of Interruption attacks include overloading a server host, cutting communication lines, blocking access to a service by overloading an intermediate network or network device, redirecting requests to invalid destinations, and theft or destruction of software or hardware.

  - Mitigation measures: Implement Firewalls with the capability to differentiate good traffic from DoS attack traffic, maintain backups of system configuration data, and consider replication.



Security Attacks

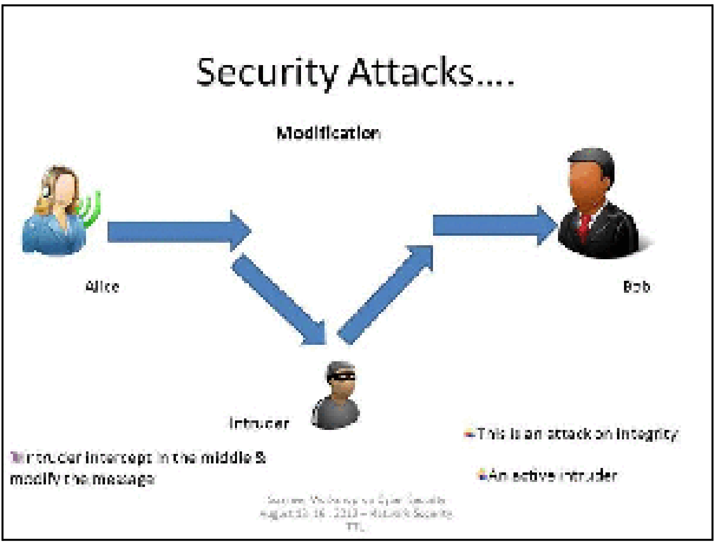
Interruption

- Intruder intercept in the middle and stop communication

- This is an attack on availability
- An active intruder

rkshop on Cyber Security
August 12-16 , 2013 – Network Security,
TTU

4. Modification
- Modification attacks compromise the integrity of information. There are three types of modifications: Change, Insertion, and Deletion.
  - Change: Modifying existing information.
  - Insertion: Adding information that did not previously exist.
  - Deletion: Removing existing information.
- Examples of Modification attacks include modifying the contents of messages in the network, changing information stored in data files, altering programs to perform differently, and reconfiguring system hardware or network topologies.
- Mitigation measures: Implement intrusion detection systems (IDS) to detect attack signatures, use encryption mechanisms, employ traffic padding, maintain backups, and use messaging techniques like checksums, sequence numbers, digests, and authentication codes.



Security Attacks….

- Intruder intercept in the middle & modify the message
- This is an attack on Integrity
- An active intruder

These four categories encompass various attack methods that target different aspects of information security. Implementing appropriate countermeasures and security measures is crucial to protect against these attacks and ensure the confidentiality, integrity, and availability of information and systems.

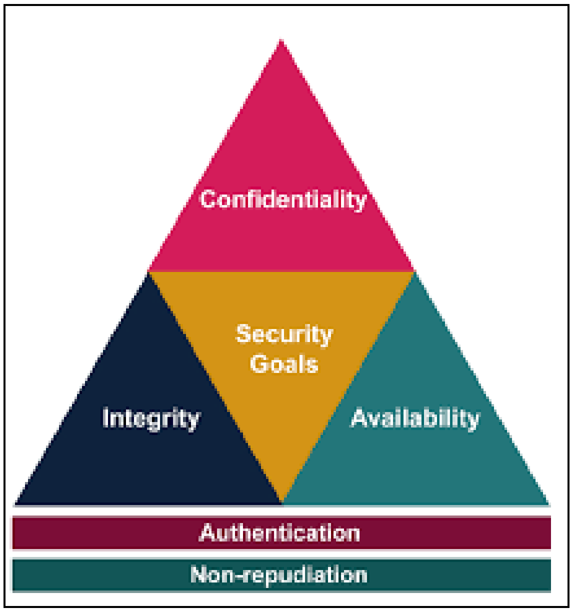
Security Services

Security services in the realm of information security are essential for safeguarding computer systems and transmitted information. These services can be classified into several categories:

1. Confidentiality
- Objective: Ensure that information in a computer system and transmitted data are only accessible to authorized parties for reading.
- Examples: Printing, displaying, and various forms of disclosure.

2. Authentication
- Objective: Verify the correct origin of a message or electronic document, providing assurance that the identity is genuine and not falsified.

3. Integrity
- Objective: Guarantee that only authorized entities have the capability to modify computer system assets and transmitted information.
- Actions: Includes writing, changing status, deleting, creating, and preventing unauthorized actions such as delaying or replaying transmitted messages.



4. Non-repudiation
- Objective: Ensure that neither the sender nor the receiver of a message can deny the transmission.

5. Access Control
- Objective: Regulate access to information resources, granting access only to authorized users or systems.

6. Availability
- Objective: Ensure that computer system assets are accessible to authorized parties whenever they are needed.

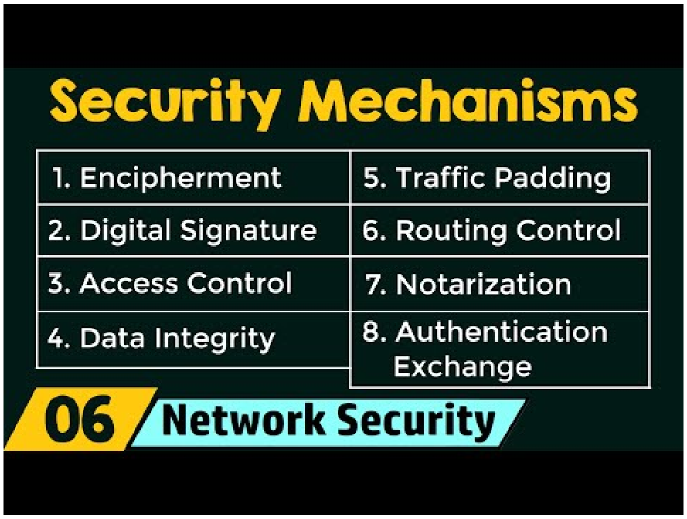
These security services collectively form the foundation of a comprehensive information security strategy. They address various aspects of security, such as preventing unauthorized access, detecting and preventing data tampering, and ensuring the confidentiality and availability of data. Effective implementation of these services is critical for protecting sensitive information and maintaining the integrity of computer systems.

Security Mechanisms

Security mechanisms play a crucial role in safeguarding information and computer systems. One of the most specific and widely-used security mechanisms is cryptographic techniques. Here are some essential security mechanisms:

1. Encipherment
- Purpose: To protect the confidentiality of information.
- Description: Encipherment, also known as encryption, involves transforming information into an unreadable format using encryption algorithms and keys. Only authorized parties with the correct decryption key can decipher and access the original information.
- Use Cases: Securely transmitting sensitive data over networks, protecting stored data on devices, securing communications, and safeguarding data at rest.

2. Digital Signature
- Purpose: To ensure the authenticity and integrity of digital documents or messages.
- Description: Digital signatures use cryptographic techniques to verify the origin and integrity of a digital document or message. The sender signs the document with a private key, and the recipient can verify the signature using the sender's public key.
- Use Cases: Verifying the authenticity of emails, documents, software updates, and online transactions.



Security Mechanisms

1. Encipherment
2. Digital Signature
3. Access Control
4. Data Integrity
5. Traffic Padding
6. Routing Control
7. Notarization
8. Authentication Exchange

06 Network Security

3. Access Control
- Purpose: To control and manage access to computer systems and resources.
- Description: Access control mechanisms restrict and regulate who can access specific resources or perform certain actions within a system. Access control lists, permissions, and authentication processes are common methods to enforce access control.
- Use Cases: Limiting user access to sensitive data, protecting critical system functions, and ensuring that only authorized users can perform specific tasks.

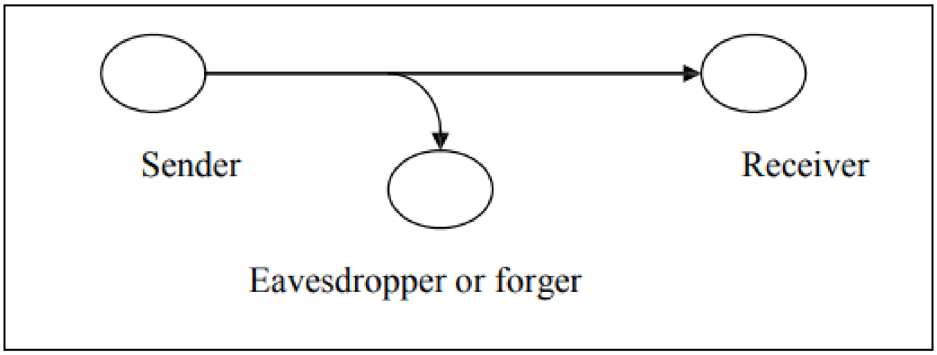
These security mechanisms are fundamental in ensuring the confidentiality, authenticity, integrity, and availability of information and systems. They are essential components of a robust security strategy, helping organizations protect their assets and sensitive data from unauthorized access and threats.

Security Attacks

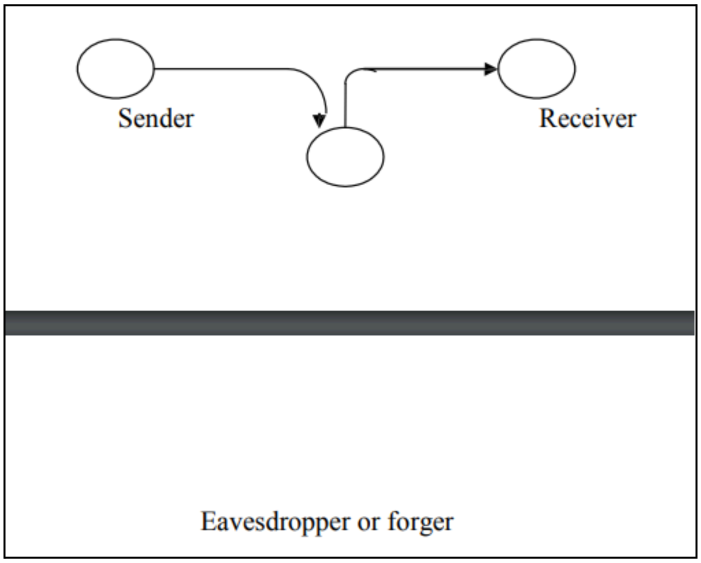
Security attacks are malicious actions or events that compromise the security of computer systems and data. These attacks can be classified into four general categories:

1. Interruption
   - Description: In an interruption attack, an asset of the system is destroyed, becomes unavailable, or is rendered unusable. This type of attack primarily targets the availability of the system.
   - Examples: Destruction of hardware components, cutting communication lines, or disabling file management systems.

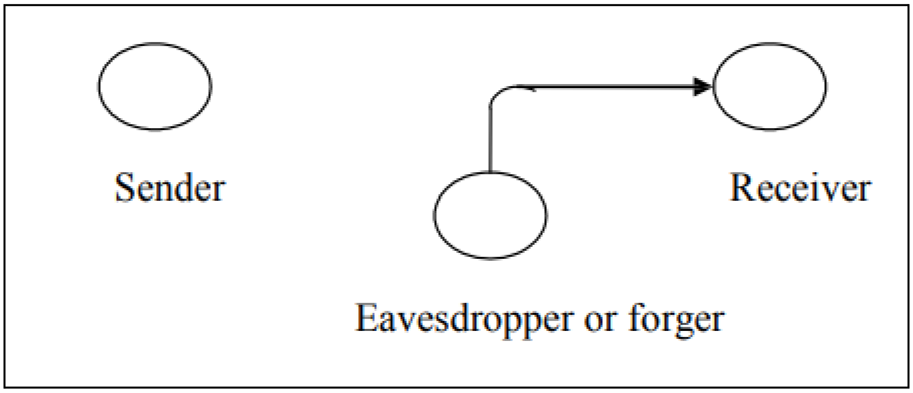
2. Interception
   - Description: Interception attacks occur when an unauthorized party gains access to a system asset, which is an attack on the confidentiality of the system. Unauthorized parties can include individuals, programs, or computers.
   - Examples: Wiretapping to capture data in a network, illicit copying of files, or unauthorized access to sensitive information.



3. Modification
   - Description: Modification attacks involve unauthorized parties not only gaining access to an asset but also tampering with it. This type of attack primarily targets the integrity of the system.
   - Examples: Changing values in data files, altering program code, or modifying the contents of messages being transmitted in a network.

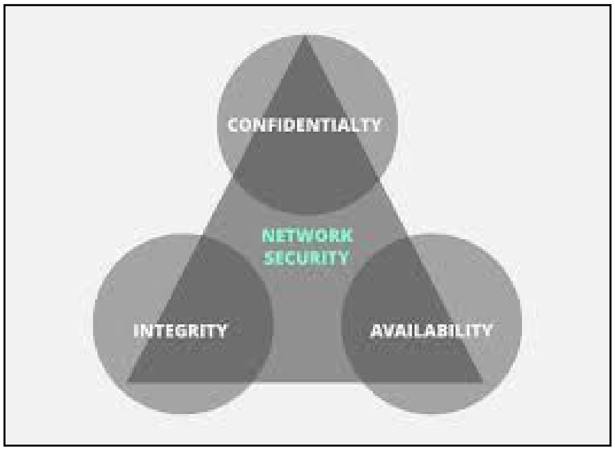


4. Fabrication
 - Description: Fabrication attacks occur when unauthorized parties insert counterfeit objects or data into the system, undermining the authenticity of the system. This can lead to the inclusion of false information.
 - Examples: Inserting spurious messages in a network, adding fictitious records to a database or file, or creating counterfeit objects within the system.



These four categories encompass various attack methods that can compromise the security of computer systems and data. Understanding these attack types is essential for developing effective security strategies and implementing appropriate countermeasures to protect against them.

An Information Security Model: C.I.A. Triangle



Confidentiality:
- Definition: The characteristic of information whereby only those with sufficient privileges may access certain information.
- Measures to protect confidentiality:
  - Information classification
  - Secure document storage
  - Application of general security policies
  - Education of information custodians and end-users

Integrity:
- Definition: The quality or state of being whole, complete, and uncorrupted.
- Threats to information integrity: Exposure to corruption, damage, destruction, or other disruption of its authentic state.
- Corruption can occur during compilation, storage, or transmission.
- Measures to protect integrity:
  - Digital signatures
  - Hashing
  - Code review to detect covert channels

Availability:
- Definition: The characteristic of information that enables user access to information in a required format, without interference or obstruction.
- Availability includes authorized user access.
- Ensures that assets are accessible to authorized parties at appropriate times.
- Measures to ensure availability:

  - RAID (Redundant Array of Independent Disks)
  - Redundant components (power supply, fan)
  - Server clusters

Computer Security Goals (CIA):

1. Confidentiality:
   - Ensures authorized access to computer-related assets.
   - Control measures include encryption, access control lists, and physical security.

2. Integrity:
   - Ensures that assets can be modified only by authorized parties or in authorized ways.
   - Control measures include digital signatures, hashing, and code review to detect covert channels.

3. Availability:
   - Ensures that assets are accessible to authorized parties at appropriate times.
   - Applies to both data and services (information and information processing).
   - Control measures include RAID, redundant components (power supply, fan), and server clusters.

Additional Notes:

- Availability means that information is present in a usable form and has enough capacity to meet the service's needs.
- Security measures should be comprehensive, addressing all aspects of the C.I.A. triangle.
- Denial of service is the opposite of availability, where authorized parties are intentionally obstructed from accessing assets.

This model forms the foundation of information security, ensuring the protection, reliability, and accessibility of critical data and resources.

Confusion and Diffusion

Confusion:

- Focus: Making the relationship between the encryption key and the cipher text as complex as possible.
- Goal: Obscure the relationship between Cipher text (CT) and plain text (PT).
- Properties:
  - Given CT, no information should be discernible about PT, key, encryption algorithm, etc.
  - Example technique: Substitution, where characters or bits are replaced with other characters or bits according to a predefined scheme.

Diffusion:

- Focus: Making each plain text bit affect as many cipher text bits as possible.
- Goal: Ensure that a small change in PT results in a significant effect on CT.
- Properties:
  - A single bit change in PT should cause a cascade of changes in CT.
  - Example technique: Transposition or permutation, where the positions of characters or bits are rearranged according to a predefined rule.

Stream Cipher:

- Encryption Process: Each plain text digit is encrypted one at a time with the corresponding digit of the key stream to produce a digit of the cipher text stream.
- Operation: Bit by bit or stream by stream.
- Example:
  - Plain text: 101010110110011
  - Key generator: Generates a key stream like 0011100011001010
  - Encryption: PT XOR Key Stream = CT
  - Result: 1001001110101110

Block Cipher:
- Encryption Process: A block cipher is a deterministic algorithm that operates on fixed-length groups of bits called blocks.
- Block Size: Blocks have a fixed length, often 64 bits or 128 bits.
- Example:
  - Plain text: 101010110110011
  - Key generator: Generates a key like 0011100011001010
  - Encryption: PT XOR Key = CT
  - Result: 1001001110101110

Differences:

Stream Cipher:
- Length: Operates bit by bit or byte by byte.
- Design: Typically complex to ensure confusion.
- Principle: Primarily focuses on confusion.
- Speed: Faster encryption.
- Encryption Modes: Common modes include CFB (Cipher Feedback) and OFB (Output Feedback).
- Decryption: Often involves XOR operations.
- Example: Vernam Cipher is a classic stream cipher.

Block Cipher:
- Length: Operates on fixed-length blocks, such as 64 bits or 128 bits.
- Design: Generally simpler than stream ciphers, combining confusion and diffusion principles.
- Principle: Combines both confusion and diffusion.
- Speed: Slower encryption due to block processing.
- Encryption Modes: Common modes include ECB (Electronic Codebook) and CBC (Cipher Block Chaining).
- Decryption: Typically a reverse of the encryption process.
- Example: DES, AES (Advanced Encryption Standard) are block ciphers widely used in cryptography.

Substitution Technique	Transposition Technique
It replaces the plaintext characters with other numbers, characters, and symbols.	It scrambles the character's position in the plaintext.
The character's identity is changed, while its position does not change.	The character's identity is changed instead of its identity.

It utilizes the monoalphabetic, polyalphabetic substitution cipher, and Playfair cipher.	It utilizes the keyed and keyless transpositional ciphers.
The low-frequency letter may easily identify the plaintext.	The keys close to the right key lead to the discovery of the plaintext.
Caesar Cipher,monoalphabetic cipher, and polyalphabetic cipher.	Reil Fence Cipher, columnar transposition cipher, and route cipher.
Substitution Cipher’s forms are: Mono alphabetic substitution cipher and poly alphabetic substitution cipher.	Transposition Cipher’s forms are: Key-less transposition cipher and keyed transposition cipher.
In substitution Cipher Technique, character’s identity is changed while its position remains unchanged.	While in transposition Cipher Technique, The position of the character is changed but character’s identity is not changed.
Involves replacing plaintext letters or groups of letters with ciphertext letters or groups of letters according to a specific algorithm or key.	Involves rearranging the order of the plaintext letters or groups of letters according to a specific algorithm or key.
Relatively easy to understand and implement, making it suitable for simple applications.	Can be more difficult to implement and understand, but can be more secure than substitution ciphers for certain applications.
The frequency distribution of the plaintext letters is typically obscured, but patterns can still be detected with statistical analysis.	The frequency distribution of the plaintext letters remains the same, but the order is scrambled, making it difficult to detect patterns with statistical analysis.
Vulnerable to frequency analysis attacks, where the most commonly used letters or letter combinations in the language can be identified and used to deduce the key.	Less vulnerable to frequency analysis attacks, but still susceptible to attacks such as brute force and known plaintext attacks.

Classical encryption techniques

Caesar Cipher technique
https://www.geeksforgeeks.org/caesar-cipher-in-cryptography/

The Caesar cipher, also known as the Caesar shift or Caesar's code, is one of the simplest and oldest encryption techniques. It's a type of substitution cipher where each letter in the plaintext is shifted a certain number of places down or up the alphabet. This shift is determined by a key, which is usually a single integer. The Caesar cipher is named after Julius Caesar, who is said to have used it to communicate confidential information.

Here's how the Caesar cipher works:

1. Choose a key: The key is an integer value that represents the number of positions each letter in the plaintext should be shifted in the alphabet. For example, if the key is 3, each letter will be shifted 3 positions down the alphabet.

2. Encrypting: To encrypt a message, take each letter in the plaintext and replace it with the letter that is located a fixed number of positions down the alphabet, according to the key. If the shift goes beyond the end of the alphabet, wrap around to the beginning.

3. Decrypting: To decrypt the message, you need to reverse the process. You take each letter in the ciphertext and shift it back the same number of positions to reveal the original plaintext.

Let's look at an example:

Encryption (Key = 3):

Plaintext: "HELLO"
Key: 3

1. H -> K (shifted 3 positions to the right)
2. E -> H
3. L -> O
4. L -> O
5. O -> L

So, "HELLO" encrypted with a Caesar cipher using a key of 3 becomes "KHOOL."

Decryption (Key = 3):

Ciphertext: "KHOOL"
Key: 3

1. K -> H (shifted 3 positions to the left)
2. H -> E
3. O -> L
4. O -> L
5. L -> O

The ciphertext "KHOOL" decrypted with a Caesar cipher using a key of 3 becomes "HELLO," which is the original plaintext.

The Caesar cipher is a straightforward and easily breakable encryption method, especially with modern computers and encryption analysis techniques. It's not suitable for secure communications today, but it serves as a historical and educational example of basic encryption concepts. To enhance its security, more advanced encryption techniques, such as the Vigenère cipher or modern symmetric-key algorithms like AES, are used.

Monoalphabhetic Cipher

https://www.geeksforgeeks.org/difference-between-monoalphabetic-cipher-and-polyalphabetic-cipher/

A monoalphabetic cipher is a type of substitution cipher where each letter in the plaintext is replaced by a fixed corresponding letter in the ciphertext. In other words, each letter in the alphabet is mapped to a unique letter in the ciphertext. This means that if you know the key or the mapping between the plaintext and ciphertext letters, you can easily encrypt and decrypt messages.

Here's how a monoalphabetic cipher works:

1. Key Generation: The key for a monoalphabetic cipher is the mapping between the letters in the plaintext alphabet and the letters in the ciphertext alphabet. This mapping remains constant for the entire message.

2. Encryption: To encrypt a message using a monoalphabetic cipher, you simply substitute each letter in the plaintext with its corresponding letter from the key. For example, if the key mapping is as follows:

   Plaintext alphabet: A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
   Ciphertext alphabet: M N O P Q R S T U V W X Y Z A B C D E F G H I J K L

   Then, if you want to encrypt the word "HELLO," it would become "ROVVY" in the ciphertext.

3. Decryption: To decrypt a message encrypted with a monoalphabetic cipher, you simply reverse the process. You use the same key to map each letter in the ciphertext back to its corresponding letter in the plaintext.

   Using the same key as above, if you receive the ciphertext "ROVVY," you would decrypt it to "HELLO."

Example:

Let's say we want to encrypt the message "OPENAI" using a monoalphabetic cipher with the key mapping provided earlier:

Plaintext alphabet: A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Ciphertext alphabet: M N O P Q R S T U V W X Y Z A B C D E F G H I J K L

To encrypt "OPENAI":

1. O becomes R
2. P becomes S
3. E becomes Y
4. N becomes A
5. A becomes M
6. I becomes R

So, "OPENAI" is encrypted to "RSYAMR" using this monoalphabetic cipher.

To decrypt "RSYAMR" back to "OPENAI," you would simply use the reverse key mapping, where R maps back to O, S maps back to P, Y maps back to E, and so on.

Monoalphabetic ciphers are straightforward and easy to understand, but they are not secure for modern encryption purposes because they are vulnerable to frequency analysis attacks. These attacks rely on the fact that certain letters and letter combinations occur more frequently in English text, which can be used to guess the key and decrypt the message. Therefore, monoalphabetic ciphers are not suitable for secure communication.

Polyalphabetic Cipher

https://www.geeksforgeeks.org/vigenere-cipher/

A polyalphabetic cipher is a type of cryptographic algorithm used for encrypting and decrypting messages. Unlike simpler ciphers like the Caesar cipher, which shift characters by a fixed amount, polyalphabetic ciphers use multiple substitution alphabets to encode the plaintext. This makes them more complex and resistant to frequency analysis attacks, where an attacker tries to deduce the plaintext by analyzing the frequency of characters in the ciphertext.

The basic idea behind a polyalphabetic cipher is to use a keyword or a repeating pattern to determine which alphabet to use for each character in the plaintext. Each alphabet is essentially a shifted version of the regular alphabet, and the shift depends on the position of the character in the keyword or pattern.

Here's a step-by-step explanation of how a polyalphabetic cipher works, along with an example:

1. Choose a Keyword: Start by choosing a keyword or a repeating pattern of characters. This keyword will determine the shifting of alphabets. For this example, let's use the keyword "KEY."

2. Create Alphabets: Create a series of shifted alphabets based on the keyword. Each alphabet is obtained by shifting the regular alphabet by the corresponding letter in the keyword.

  - Regular Alphabet: A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
  - Alphabet for 'K': K L M N O P Q R S T U V W X Y Z A B C D E F G H I J
  - Alphabet for 'E': E F G H I J K L M N O P Q R S T U V W X Y Z A B C D
  - Alphabet for 'Y': Y Z A B C D E F G H I J K L M N O P Q R S T U V W X

3. Encrypt: To encrypt a message, substitute each letter in the plaintext with the corresponding letter in the shifted alphabet. Repeat this process for each character in the message.

Example:
Let's encrypt the message "HELLO" using the keyword "KEY."

- H is in the regular alphabet, so it becomes "K" in the ciphertext.
- E is in the "E" alphabet, so it remains "E" in the ciphertext.
- L is in the "Y" alphabet, so it becomes "B" in the ciphertext.
- L is in the "E" alphabet, so it becomes "D" in the ciphertext.
- O is in the "K" alphabet, so it becomes "V" in the ciphertext.

So, "HELLO" is encrypted as "KEDBV."

4. Decrypt: To decrypt a message, you reverse the process. You use the keyword to determine the appropriate alphabet for each character in the ciphertext and substitute it with the corresponding letter in the regular alphabet.

Polyalphabetic ciphers can be quite secure if used properly, especially with longer and more complex keywords. However, they require a shared secret (the keyword) between the sender and receiver and can be vulnerable to attacks if the keyword is too short or predictable.

Vigenère cipher

https://www.geeksforgeeks.org/vigenere-cipher/

The Vigenère cipher is a classical symmetric-key encryption algorithm that was developed in the 16th century by Blaise de Vigenère. It is a polyalphabetic substitution cipher, which means that it uses multiple substitution alphabets to encrypt plaintext characters, making it more secure than simple Caesar ciphers.

Here's how the Vigenère cipher works:

1. Key Generation:
   - Choose a secret keyword or keyphrase, which is a sequence of letters (e.g., "KEY").
   - Repeat the keyword or keyphrase as many times as necessary to match the length of the plaintext message you want to encrypt. If the keyword is shorter than the plaintext, repeat it until it matches or exceeds the length of the plaintext. For example, if the keyword is "KEY" and the plaintext is "HELLO," repeat "KEY" to get "KEYKEYKEYK."

2. Encryption:
   - Assign numerical values to the letters of the alphabet, typically using A=0, B=1, C=2, ..., Z=25.
   - Break the plaintext into individual letters and the key into corresponding letters as well. For example:
     - Plaintext: H E L L O
     - Key: K E Y K E

   - For each letter in the plaintext, add its value to the corresponding letter in the key. Use modular arithmetic to ensure the result falls within the range of 0 to 25 (the size of the English alphabet). If the result is greater than 25, subtract 26. This step is performed by shifting letters in a cyclic manner.
     - Plaintext: H E L L O
     - Key: K E Y K E
     - Encrypted: O I E Q G

3. Decryption:
   - To decrypt the ciphertext, you need to know the key. Subtract the numerical value of the key from each corresponding letter in the ciphertext, again using modular arithmetic to ensure the result is within the range of 0 to 25.
     - Ciphertext: O I E Q G
     - Key: K E Y K E
     - Decrypted: H E L L O

Example:

Let's encrypt the plaintext "HELLO" using the keyword "KEY."

1. Key Generation:
   - Keyword: KEY
   - Repeated to match the length of the plaintext: KEYKEYKEYK

2. Encryption:
   - Assign numerical values (A=0, B=1, ..., Z=25).
     - Plaintext: H E L L O
     - Key: K E Y K E

   - Perform the Vigenère encryption by adding the values of the corresponding letters in the plaintext and key:
     - Encrypted: O I E Q G

3. Decryption:
   - To decrypt the ciphertext, subtract the key values from the ciphertext values:
     - Ciphertext: O I E Q G
     - Key: K E Y K E

     - Decrypted: H E L L O

The Vigenère cipher is more secure than simple substitution ciphers like the Caesar cipher because it uses a variable substitution alphabet for each letter in the plaintext, making it more challenging to decipher without the key. However, it is still vulnerable to frequency analysis and other cryptographic attacks, especially if the key is short or easily guessable. To enhance security, longer and more random keys are recommended.

Vernam Cipher

https://www.geeksforgeeks.org/vernam-cipher-in-cryptography/

The Vernam Cipher, also known as the one-time pad, is a symmetric-key encryption algorithm that provides perfect secrecy when used correctly. It was developed by Gilbert Vernam in 1917 and is considered one of the most secure encryption methods ever devised. The key idea behind the Vernam Cipher is to use a random key that is as long as the plaintext message, and this key is combined with the plaintext to produce the ciphertext.

Here's how the Vernam Cipher works:

1. Key Generation:
   - Generate a random key that is exactly as long as the plaintext message. The key should consist of random characters, typically letters or numbers. Each character in the key must be chosen independently and uniformly at random.
   - This key is used only once and should never be reused for any other message. Hence, the term "one-time pad."

2. Encryption:
   - To encrypt a message, the plaintext is combined (usually by XOR operation) with the random key character by character. XOR (exclusive or) is a bitwise operation that returns 1 if the bits being compared are different and 0 if they are the same.
   - The result of the XOR operation between each plaintext character and the corresponding key character becomes the ciphertext character.
   - The ciphertext is essentially a random jumble of characters that provide no information about the original plaintext.

Mathematically, the encryption process can be represented as:
   - Ciphertext (C[i]) = Plaintext (P[i]) XOR Key (K[i]) for each character i in the message.

3. Decryption:
   - To decrypt the ciphertext, the recipient uses the same one-time pad (key) that was used for encryption.
   - The recipient XORs the ciphertext with the key character by character to retrieve the original plaintext.
   - Mathematically, the decryption process can be represented as:
     - Plaintext (P[i]) = Ciphertext (C[i]) XOR Key (K[i]) for each character i in the ciphertext.

The strength of the Vernam Cipher lies in the fact that without knowing the exact key used for encryption, it is impossible to decrypt the ciphertext. Even with advanced computational methods, breaking the Vernam Cipher requires as much effort as trying all possible keys, making it practically unbreakable if the key is truly random, used only once, and kept secret.

Here's a detailed example of the Vernam Cipher:

Suppose we want to encrypt the message "HELLO" using a randomly generated one-time pad key "WORLD." The key and plaintext are as follows:

Plaintext (P):   H   E   L   L   O
Key (K):   W   O   R   L   D

Now, we perform the XOR operation character by character:

- H XOR W = X
- E XOR O = H
- L XOR R = P
- L XOR L = 0
- O XOR D = V

So, the ciphertext (C) is "XHP0V."

To decrypt the ciphertext, you would perform the same XOR operation with the same key:

- X XOR W = H
- H XOR O = E
- P XOR R = L
- 0 XOR L = L

- V XOR D = O

You've successfully decrypted the ciphertext back to "HELLO."

Remember that the key "WORLD" should be kept absolutely secret and only used once. If it's reused or falls into the hands of an attacker, the security of the Vernam Cipher is compromised.

Playfair Cipher

https://www.geeksforgeeks.org/playfair-cipher-with-examples/

The Playfair cipher is a symmetric key substitution cipher that was invented in 1854 by Charles Wheatstone but is named after its promoter Lyon Playfair. It is a relatively simple but effective method for encrypting text. The key idea behind the Playfair cipher is to break the plaintext into pairs of two letters (digraphs) and substitute them with different pairs of letters based on a key matrix. Unlike simple substitution ciphers, the Playfair cipher provides a more complex encryption process that makes it harder to decipher.

Here's how the Playfair cipher works:

1. Key Matrix Setup:
   - Start with a 5x5 matrix (5 rows and 5 columns) called the Playfair matrix.
   - Fill the matrix with a keyword (usually without repeating letters) followed by the remaining letters of the alphabet (excluding 'J' since it's treated as 'I' in the Playfair cipher).

   Example:
   ```
   Keyword: KEYWORD
   Playfair Matrix:
   K E Y W O
   R D A B C
   F G H I L
   M N P Q S
   T U V X Z
   ```

2. Text Preparation:
   - Remove any spaces and punctuation from the plaintext.
   - Replace any occurrences of 'J' with 'I'.
   - Break the plaintext into pairs of two letters (digraphs).

   Example:
   ```
   Plaintext: HELLO WORLD
   Processed: HE LX LO WO RL D
   ```

3. Encryption:

- For each digraph in the plaintext, apply the following rules:
  - If both letters in the digraph are in the same row of the matrix, replace each letter with the letter to its right (looping back to the beginning of the row if needed).
  - If both letters are in the same column, replace each letter with the letter below it (looping back to the top of the column if needed).
  - If the letters are in different rows and columns, form a rectangle using the two letters and replace them with the letters at the corners of the rectangle.
  - If a digraph contains a repeated letter (e.g., "LL" in "HELLO"), insert an 'X' between them and then apply the above rules.

Example:

```
Processed: HE LX LO WO RL D
Encrypted: GGATJKFG
```

4. Decryption:
   - To decrypt, use the same Playfair matrix and apply the reverse of the encryption rules.
   - If the plaintext has any 'X' characters between repeated letters, remove them during decryption.

Let's decrypt the example ciphertext "GGATJKFG" using the same Playfair matrix:

- Processed: GG AT JK FG
- Decrypted: HELLO WORLD

So, the original plaintext "HELLO WORLD" is successfully decrypted from the ciphertext "GGATJKFG" using the Playfair cipher.

The Playfair cipher is relatively secure against simple frequency analysis attacks and provides a reasonable level of security for its simplicity. However, it can be vulnerable to more advanced cryptanalysis methods, especially if the key is short or weakly chosen.

Hill Cipher

https://www.geeksforgeeks.org/hill-cipher/

The Hill cipher is a classical symmetric key encryption algorithm that operates on blocks of plaintext, typically in the form of letters, and transforms them into ciphertext using matrix multiplication. It was developed by Lester S. Hill in 1929 and is a substitution cipher, but it differs from other substitution ciphers like the Caesar cipher or Atbash cipher because it encrypts multiple letters at a time.

Here's how the Hill cipher works:

1. Key Generation:
   - The first step in using the Hill cipher is to generate a key matrix. This key matrix is usually a square matrix, and its size (dimension) determines how many letters will be encrypted at a time. Common dimensions are 2x2, 3x3, and 4x4.

- The key matrix should be invertible, meaning its determinant must be relatively prime to the alphabet size (usually 26 for English letters). This ensures that a unique decryption key exists.
- For example, let's use a 2x2 key matrix:

```
K = | 6  24 |
    | 13  16 |
```

2. Encryption:

- Divide the plaintext into blocks, each containing the same number of letters as the dimension of the key matrix. If the last block doesn't have enough letters, pad it with some character (e.g., 'X').
- Convert each block of plaintext letters into numbers, typically using a letter-to-number mapping (e.g., A=0, B=1, ..., Z=25).
- For each block of letters, represented as a column vector, perform matrix multiplication with the key matrix.
- Take the result modulo the alphabet size (26 in English) to obtain the ciphertext values.
- Convert the ciphertext values back into letters using the reverse letter-to-number mapping.

Let's encrypt the plaintext "HELLO" using the key matrix mentioned earlier:

- Convert "HELLO" to numerical form: [7, 4, 11, 11, 14]
- Split into 2x2 blocks: [7, 4] and [11, 11]
- Perform matrix multiplication:

```
| 6  24 |   | 7  11 |   | 226  262 |
| 13 16 |  | 4  11 | = | 217  252 |
```

- Take modulo 26 for each element: [226 % 26, 262 % 26] = [6, 12] and [217 % 26, 252 % 26] = [11, 24]
- Convert back to letters: "GM" and "LY"

So, "HELLO" is encrypted to "GMLY" using the Hill cipher with the given 2x2 key matrix.

3. Decryption:

- To decrypt the ciphertext, you need the inverse of the key matrix. This requires a key matrix that is invertible, as mentioned earlier.
- Perform the same matrix multiplication with the ciphertext and the inverse key matrix to obtain the plaintext in numerical form.
- Convert the numerical values back to letters using the reverse letter-to-number mapping.

In this case, if we have the inverse key matrix, we can decrypt "GMLY" back to "HELLO."

Hill cipher is more secure than simple substitution ciphers like the Caesar cipher but less secure than modern encryption techniques like AES. Its security relies on the size of the key matrix and its ability to resist cryptanalysis techniques like matrix inversion, which becomes harder as the dimension of the matrix increases.

Rail Fence Cipher

https://www.geeksforgeeks.org/rail-fence-cipher-encryption-decryption/

The Rail Fence Cipher, also known as the Zigzag Cipher, is a simple transposition cipher that rearranges the letters of a message to obscure its content. It gets its name from the way the letters are arranged in a zigzag pattern, resembling a fence made of rails. This cipher is not very secure and is mainly used for educational purposes or as a fun puzzle.

Here's how the Rail Fence Cipher works:

1. Key and Setup:
   - The key for the Rail Fence Cipher is the number of rails or rows you want to use to encrypt the message. Typically, you choose a positive integer for the key.
   - You write the message in a zigzag pattern along the number of rails specified by the key.

2. Encryption:
   - Start by writing the message from left to right along the top rail.
   - Then, move diagonally down to the next rail and continue writing the message from left to right.
   - Repeat this process until you reach the bottom rail.
   - Once you reach the bottom rail, reverse the direction and start moving diagonally up to the second-to-last rail, and continue writing the message from left to right.
   - Keep repeating the process until you have filled all the rails.

3. Concatenate Rows:
   - After writing the message along the zigzag pattern, read the letters row by row from top to bottom to form the encrypted message.

Let's illustrate this with an example. We'll encrypt the message "HELLO WORLD" with a key of 3:

1. Write the message in a zigzag pattern along three rails:

```
H . . O . . R . . D
. E . L . W . L .
. . L . . O . . .
```

2. Concatenate the rows to get the encrypted message:

```
HORD ELWL LO
```

So, "HELLO WORLD" encrypted with a key of 3 using the Rail Fence Cipher becomes "HORDELWLLO."

To decrypt a message encrypted with the Rail Fence Cipher, you would follow a similar process. You'd know the key and write the encrypted message in the zigzag pattern with the appropriate number of rails, then read the original message from left to right along the rails.

---

simple columnar transposition technique

https://www.geeksforgeeks.org/columnar-transposition-cipher/

Simple columnar transposition is a classical encryption technique that involves rearranging the characters of a plaintext message into a grid or matrix based on a specific key. This rearrangement makes it more difficult for an unauthorized party to read the message without knowledge of the key. To decrypt the message, the recipient needs to know the key and the method used for the transposition.

Here's how the simple columnar transposition technique works, along with a detailed example:

Encryption Process:

1. Choose a Key: The key is a word or phrase used to determine the order in which the columns are arranged. It can be a keyword or any word or phrase of your choice. For this example, let's use the key "SECRET."

2. Create the Grid: Write down the key at the top of the grid in alphabetical order, removing any duplicate letters. In this case, it becomes "CERST."

```
Key: SECRET
Grid: C E R S T
```

3. Write the Message: Write the plaintext message horizontally into the grid, row by row. Start from the leftmost column and continue to the right until you reach the end of the row, then move to the next row. If the message length does not fill the entire grid, you can pad the remaining cells with placeholder characters (e.g., "X" or "Z").

   Let's say our plaintext message is: "HELLO, WORLD!"

```
Key: SECRET
Grid: C E R S T
Message: H E L L O ,   W O R L D !
```

4. Read the Columns: To create the ciphertext, read the columns of the grid from left to right, starting with the column associated with the first letter of the key ("C" in this case), followed by "E," "R," "S," and "T."

```
Ciphertext: LRO LDWO,HEL!
```

The spaces and punctuation are preserved, but the letters are rearranged based on the key.

Decryption Process:

1. Recreate the Grid: Write down the key in alphabetical order at the top of a new grid.

```
Key: SECRET
Grid: C E R S T
```

2. Determine the Number of Rows: Calculate the number of rows needed based on the length of the ciphertext and the number of columns in the grid. In this case, we have 15 characters in the ciphertext and 5 columns in the grid, so we need 3 rows.

```
Key: SECRET
Grid: C E R S T
```

3. Write the Ciphertext: Write the ciphertext into the grid column by column, starting with the first column associated with the first letter of the key ("C" in this case), followed by "E," "R," "S," and "T."

```
Ciphertext: LRO LDWO,HEL!
Key: SECRET
Grid: C E R S T
Ciphertext: L R O   L D W O , H E L !
```

4. Read the Rows: To obtain the original plaintext message, read the rows of the grid from top to bottom, combining the characters.

```
Plaintext: HELLO, WORLD!
```

The spaces and punctuation are preserved, and the message is reconstructed in its original form.

That's the simple columnar transposition technique. It relies on a key to rearrange the message's characters and then reverses the process during decryption to retrieve the original message.