Cryptography, Network & Software Security

Jaspreet Singh

Project Engineer
E-Security
CDAC, Hyderabad

S2

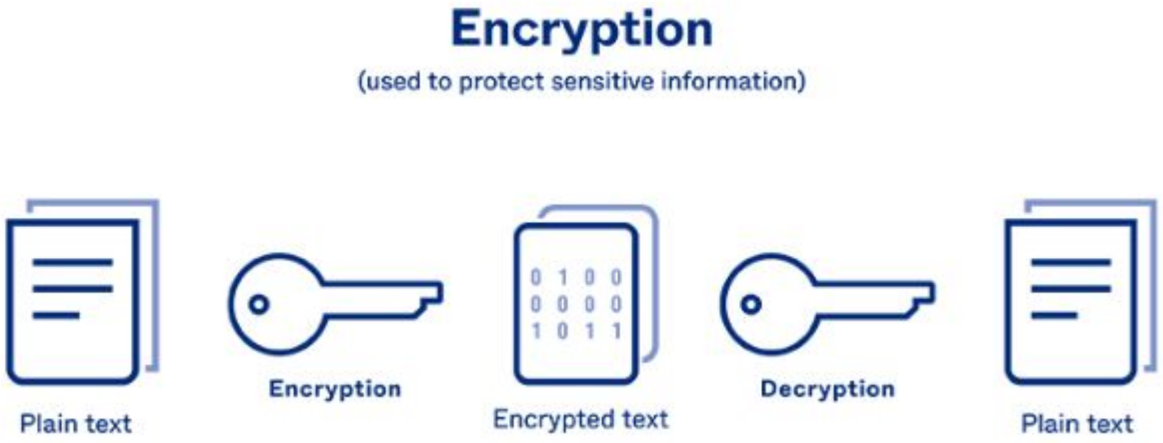
Introduction

Classical Encryption

- Classical encryption involves methods of encoding messages so that only authorized parties can understand the information.

- Encryption is crucial for maintaining the confidentiality and integrity of data, preventing unauthorized access and ensuring secure communication.

- Examples of early encryption include the use of hieroglyphs in Egypt and the Spartan Scytale.



Encryption

(used to protect sensitive information)

Plain text

Encryption

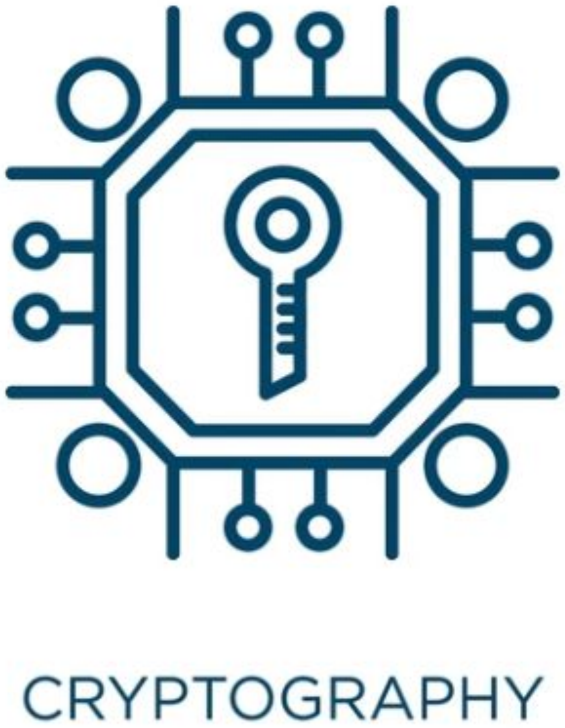
Encrypted text

Decryption

Plain text

Overview

- Cryptography, Cryptanalysis & Brute Force Attacks.
- Substitution & Transposition Techniques.
- Cryptographically strong random numbers/APIs.
  Steganography.
- Symmetric and Asymmetric Key
  Cryptography with OpenSSL



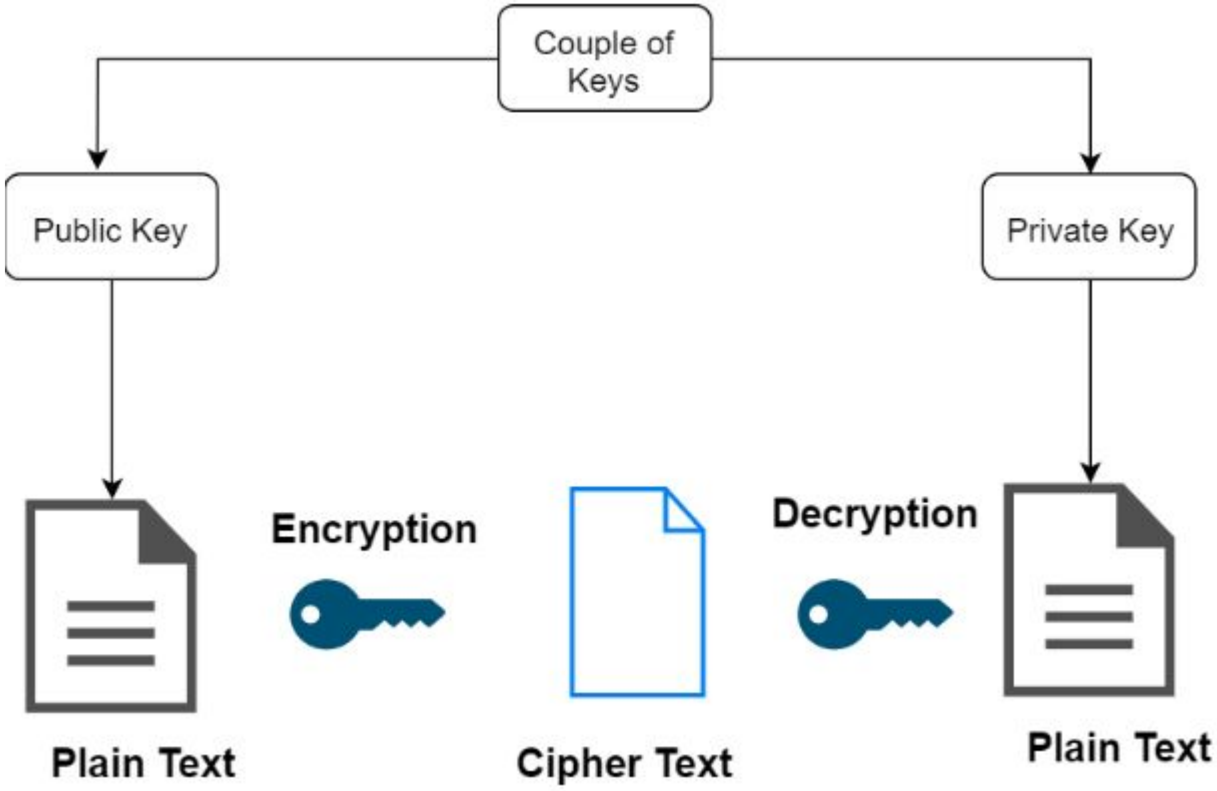
Cryptography

- ❖ Cryptography is the art of writing or solving codes and encompasses techniques for secure communication in the presence of adversaries.

Types:

- **Symmetric Cryptography**: Uses the same key for encryption and decryption.
- **Asymmetric Cryptography**: Uses a pair of keys (public and private) for encryption and decryption.

Cryptography

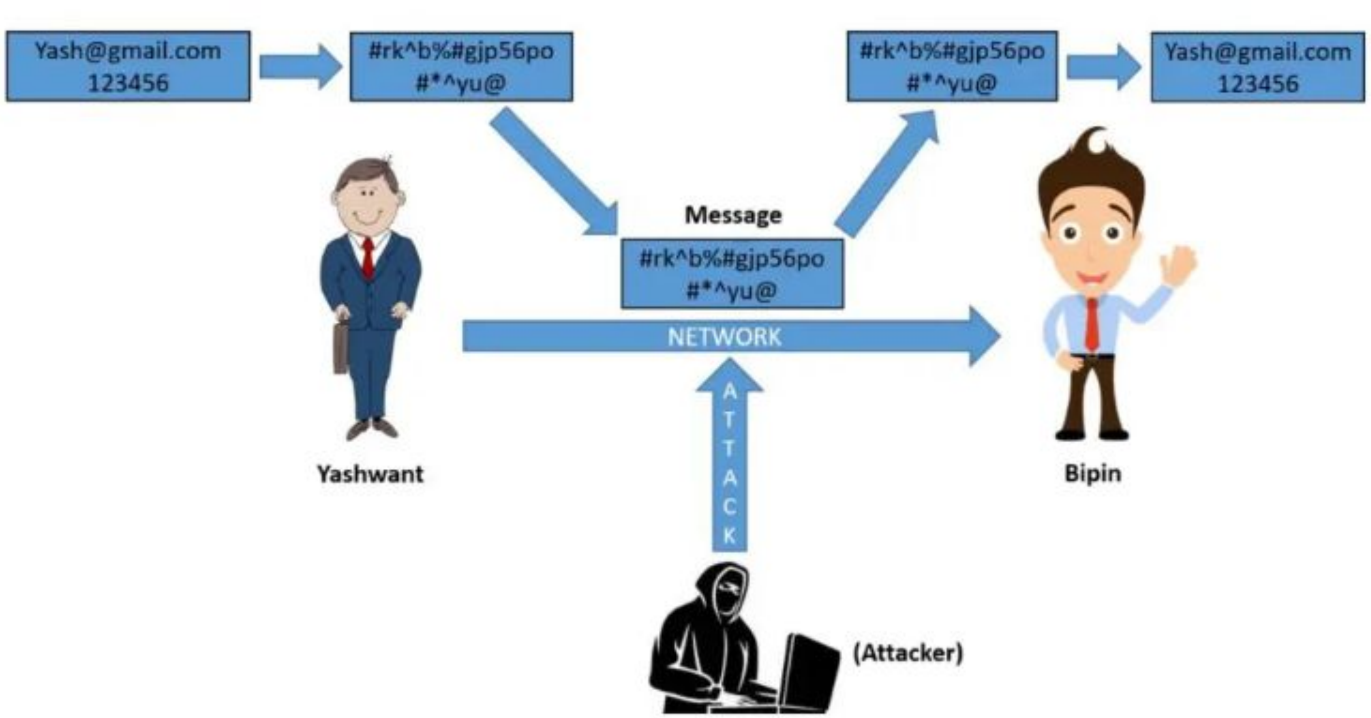


Basic Terms:

- **Plaintext**: The original readable message.
- **Ciphertext**: The encrypted message.
- **Key**: The secret used to transform plaintext into ciphertext and vice vers

Cryptanalysis

- Cryptanalysis is the study of methods for obtaining the meaning of encrypted information without access to the secret key.



Cryptanalysis

Purpose: To discover weaknesses in cryptographic algorithms and protocols.

Types:

- **Frequency Analysis**: Analyzing the frequency of letters or groups of letters.
- **Pattern Analysis**: Looking for patterns or repetitions in the ciphertext.

Brute Force Attacks

- A brute force attack attempts to find a password or key by systematically checking all possible combinations until the correct one is found.

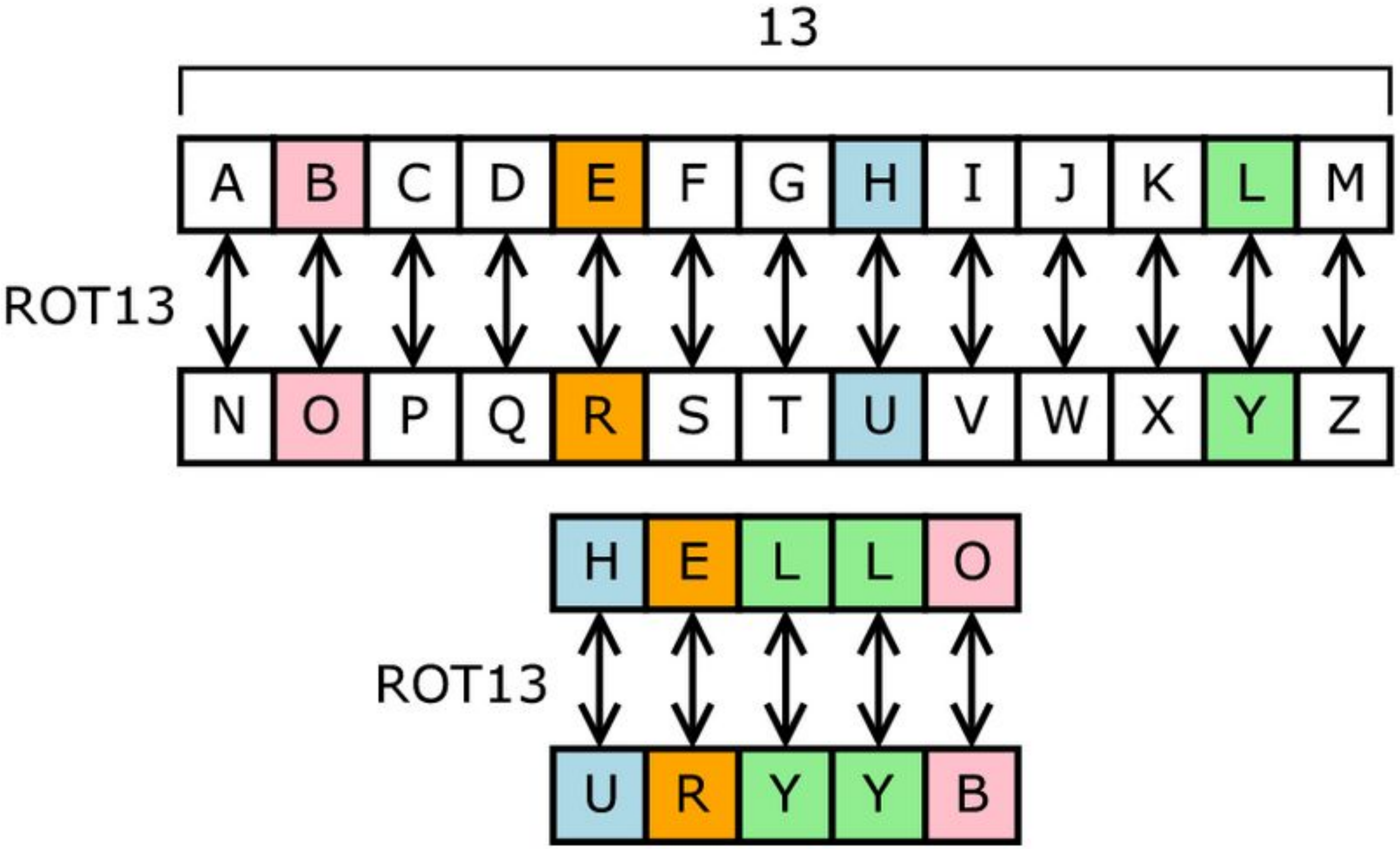
Characteristics: Time-consuming and computationally intensive.

Protection:

- Use of strong, complex passwords.
- Increasing key length.
- Implementing account lockout mechanisms after a number of failed attempts

Substitution Techniques

- Substitution techniques encode a message by replacing elements of the plaintext with corresponding elements of the ciphertext.



Substitution Techniques

- **Examples**:
  - **Caesar Cipher**: Shifts each letter in the plaintext by a fixed number of places.
  - **Monoalphabetic Cipher**: Uses a fixed substitution over the entire message.

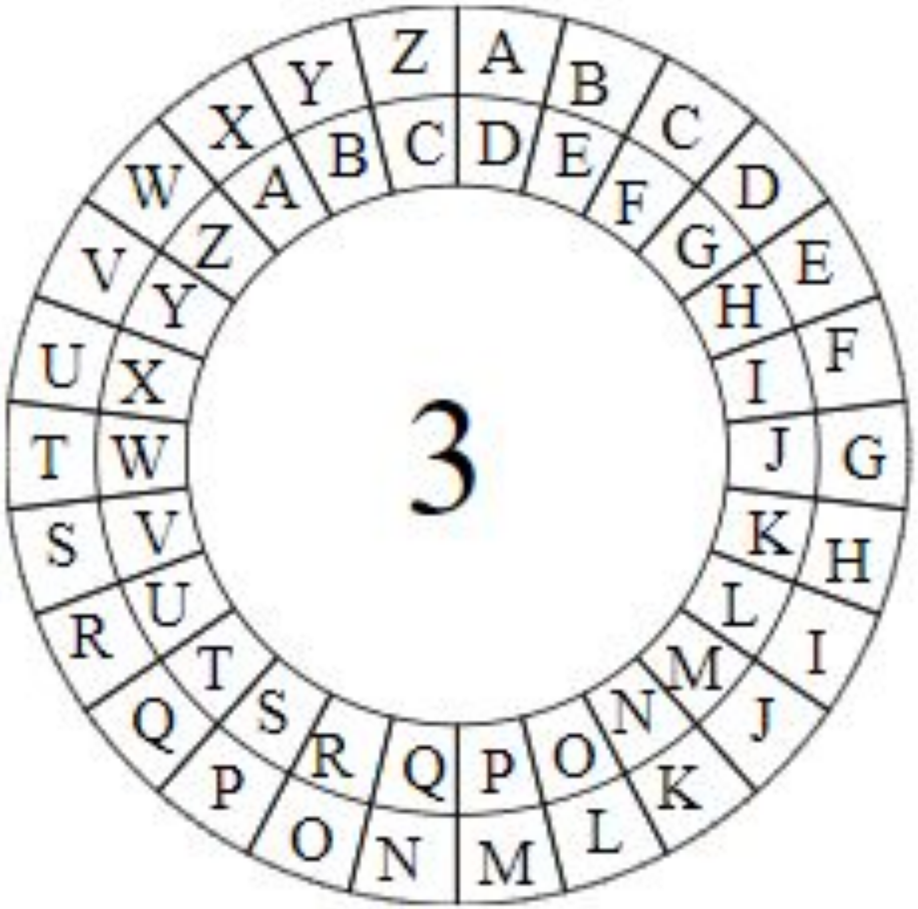
- **Strengths & Weaknesses**:
  - **Strengths**: Simple and easy to implement.
  - **Weaknesses**: Vulnerable to frequency analysis and other cryptanalysis methods.

Caesar Cipher

Explanation: Each letter in the plaintext is shifted a certain number of places down or up the alphabet.

Example:

- Plaintext: "HELLO"
- Shift: 3
- Ciphertext: "KHOOR"

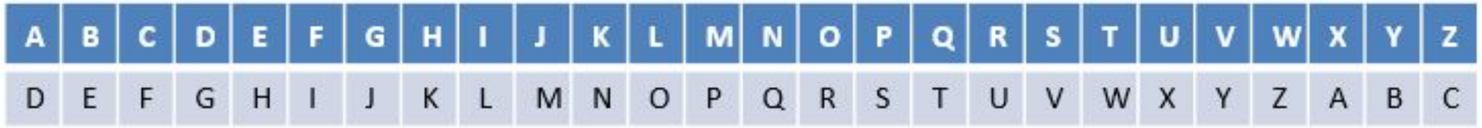


A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
D E F G H I J K L M N O P Q R S T U V W X Y Z A B C
3

Cryptanalysis: The Caesar Cipher is easy to break using frequency analysis since there are only 25 possible shifts.

Caesar Cipher

- Plain: this is crypto algo class
- Cipher: wklv lv fubswr dojr fodvv

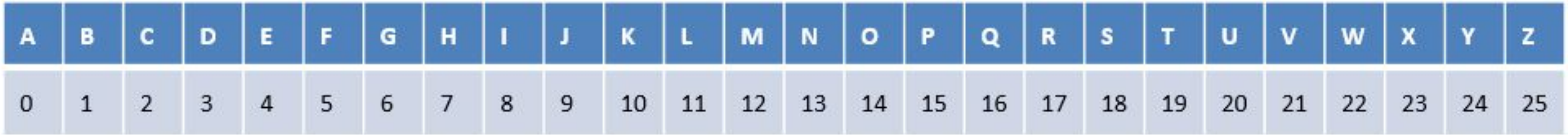


• Plain:	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
• Cipher:	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

c=cipher, p=plain text, k=key, d=decrypted text, E = encrytion, D = decryption

$c = E(k, p) = (p + k) \bmod 26$

$p = D(k, c) = (c\ k) \bmod 26$



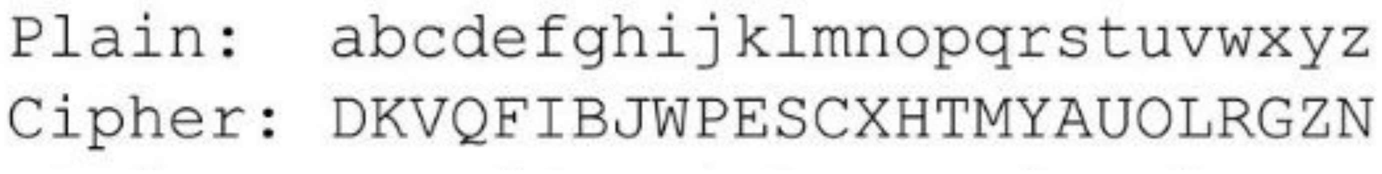
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Monoalphabetic Cipher

❖ Each letter of the plaintext is mapped to a corresponding letter of ciphertext using a single substitution alphabet.

Example:

- Plaintext: "hello"
- Ciphertext: "JFSSH"

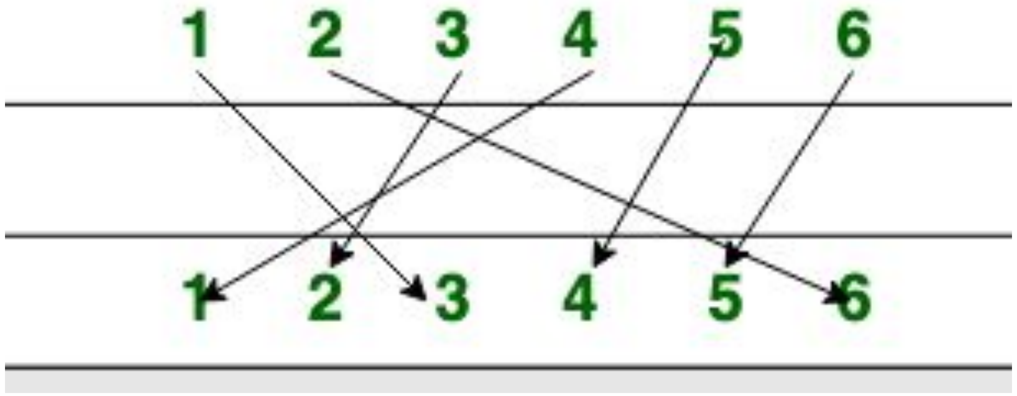


Plain: abcdefghijklmnopqrstuvwxyz
Cipher: DKVQFIBJWPESCXHTMYAUOLRGZN

Cryptanalysis: More complex than the Caesar Cipher but still vulnerable to frequency analysis due to the fixed nature of the substitution.

Transposition Techniques

- Transposition techniques encode a message by rearranging the characters of the plaintext according to a specific system.



Transposition Techniques

Examples:

- **Rail Fence Cipher**: Writes the message in a zigzag pattern and then reads off each line.
- **Columnar Transposition**: Writes the plaintext in a grid and reads the columns in a specified order.

Strengths & Weaknesses:

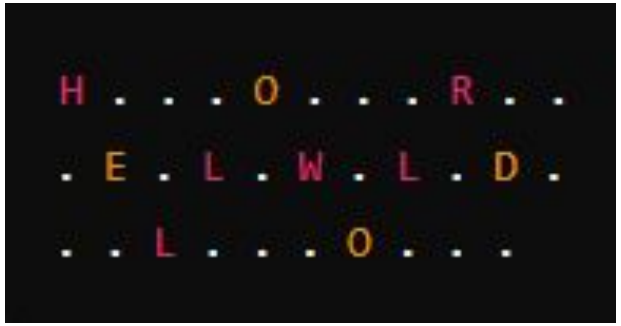
- **Strengths**: Transposition doesn't change the frequency of individual elements.
- **Weaknesses**: Still vulnerable to pattern recognition and known-plaintext attacks

Rail Fence Cipher

- ❖ The plaintext is written in a zigzag pattern down and up across multiple "rails" (lines), and then read line by line.

Example:

- Plaintext: *"HELLO WORLD"*
- Zigzag pattern on 3 rails
- Ciphertext: *"HOR ELWLD LO"*



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

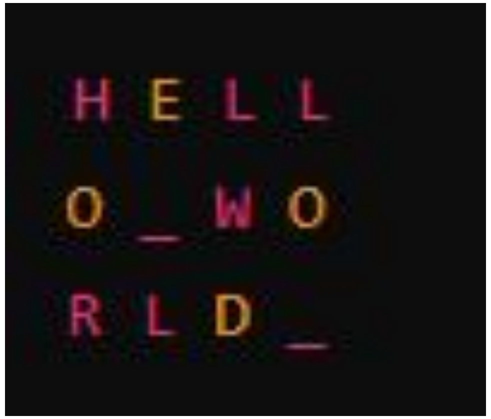
Cryptanalysis: Often easily broken by visually inspecting the ciphertext and testing different numbers of rails.

Columnar Transposition

❖ The plaintext is written into a rectangle grid of fixed width and read off column by column in a specified order.

Example:

- Plaintext: "HELLO WORLD"
- Grid (width 4)
- Reading columns: "HOR E_L LWD LO_"



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

Cryptanalysis: Requires trying different column permutations and widths to decipher.

Generating Strong Random Numbers

Methods:

- Hardware random number generators.
- Cryptographic libraries and functions.

Python:

```
import secrets
random_number = secrets.token_hex(16)
```

Java:

```
SecureRandom random = new SecureRandom();
byte[] values = new byte[16];
random.nextBytes(values);
```

Best Practices:

- Use high-entropy sources.
- Regularly reseed the random number generator.
- Avoid predictable patterns.

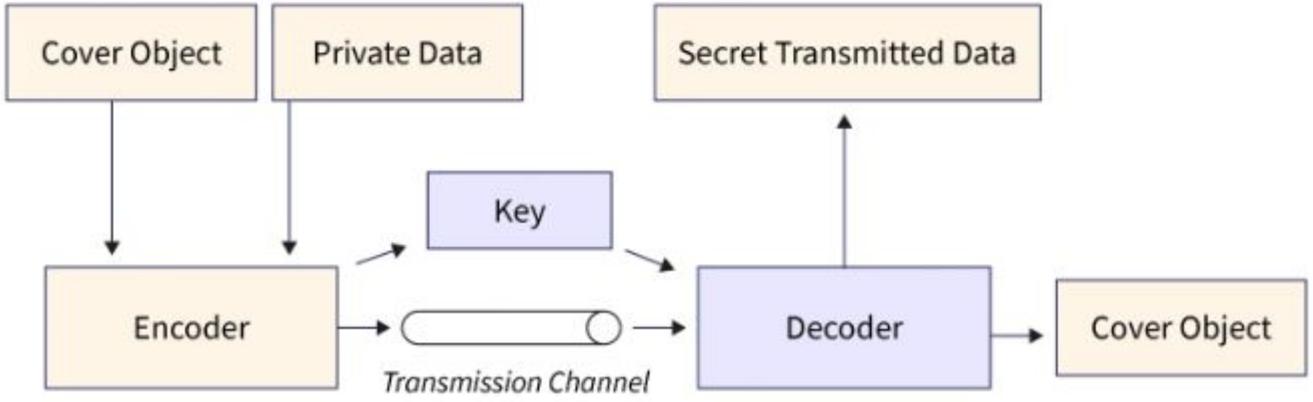
Steganography

- ❖ The practice of concealing messages or information within other non-secret text or data.

History: Examples from ancient Greece, where messages were hidden in wax tablets or within the physical structure of objects.

Modern Use: Digital steganography involves embedding data in multimedia files:

images, audio, and video



Techniques of Steganography

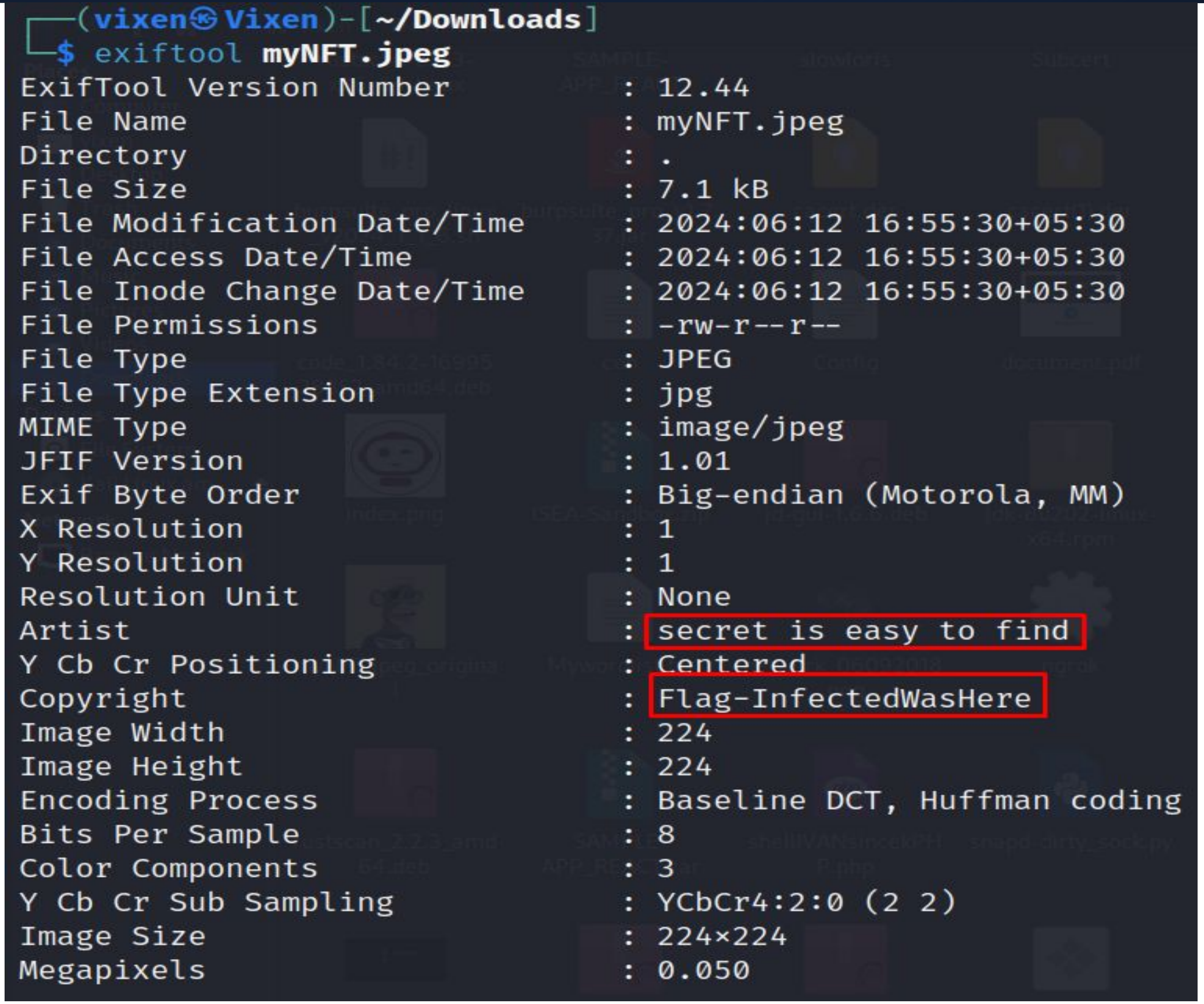
- **Image Steganography**: Uses the Least Significant Bit (LSB) method to embed information within the pixel values.

- **Audio Steganography**: Modifies sound waves to hide data within audio files.

- **Video Steganography**: Embeds data within the frames of a video file, often using the LSB method or other encoding techniques.

Example of Steganography

Image Steganography



```
┌──(vixen㉿Vixen)-[~/Downloads]
└─$ exiftool myNFT.jpeg
ExifTool Version Number         : 12.44
File Name                       : myNFT.jpeg
Directory                       : .
File Size                       : 7.1 kB
File Modification Date/Time     : 2024:06:12 16:55:30+05:30
File Access Date/Time           : 2024:06:12 16:55:30+05:30
File Inode Change Date/Time     : 2024:06:12 16:55:30+05:30
File Permissions                : -rw-r--r--
File Type                       : JPEG
File Type Extension             : jpg
MIME Type                       : image/jpeg
JFIF Version                    : 1.01
Exif Byte Order                 : Big-endian (Motorola, MM)
X Resolution                    : 1
Y Resolution                    : 1
Resolution Unit                 : None
Artist                          : secret is easy to find
Y Cb Cr Positioning             : Centered
Copyright                       : Flag-InfectedWasHere
Image Width                     : 224
Image Height                    : 224
Encoding Process                : Baseline DCT, Huffman coding
Bits Per Sample                 : 8
Color Components                : 3
Y Cb Cr Sub Sampling            : YCbCr4:2:0 (2 2)
Image Size                      : 224x224
Megapixels                      : 0.050
```

Detecting Steganography

Techniques:

- Statistical analysis to detect anomalies in the file structure.
- Visual or auditory inspection for irregularities.
- Using specialized steganalysis software.
-

Tools: Examples of tools used for detecting steganography:

- **StegExpose**: A tool for detecting LSB steganography in images.
- **Xiao Steganography**: Software for detecting hidden data in various file formats.
-

Challenges: Advanced steganography methods can be very difficult to detect, requiring sophisticated analysis and tools.

Detecting Steganography

Techniques:

- Statistical analysis to detect anomalies in the file structure.
- Visual or auditory inspection for irregularities.
- Using specialized steganalysis software.
-

Tools: Examples of tools used for detecting steganography:

- **StegExpose**: A tool for detecting LSB steganography in images.
- **Xiao Steganography**: Software for detecting hidden data in various file formats.
-

Challenges: Advanced steganography methods can be very difficult to detect, requiring sophisticated analysis and tools.

Conclusion

Summary: Recap of key points:

- Classical encryption techniques and their historical context.
- The roles of cryptography and cryptanalysis.
- Brute force attacks and how to protect against them.
- Substitution and transposition techniques.
- Importance of cryptographically strong random numbers.
- Steganography methods and detection techniques.

Q & A

Reference

- Fortinet
  - https://www.fortinet.com/resources/cyberglossary/what-is-cryptography
- Cybersecurity & Infrastructure Security Agency (CISA)
  - https://www.cisa.gov/
- Techtarget
  - https://www.techtarget.com/searchsecurity/
- Exiftool for steganography
  - https://www.geeksforgeeks.org/installing-and-using-exiftool-on-linux/

Thank you