[Hello Security]

WGUiSW

0x02 About insider threats

- Crypto, Stego, DLP

Krystian Bajno, 2024

_baycode.eu

[Table of Contents]

[Links are clickable]

[Links]

https://gchq.github.io/CyberChef/

https://quipqiup.com/

https://www.cisa.gov/topics/physical-security/insider-threat-mitigation

https://github.com/HavocFramework/Havoc

https://github.com/krystianbajno/articles/tree/main/0x04%20Lateral%20Movement/src

WGUiSW _baycode.eu

0x03 Whoami

Krystian Bajno

Cyber Security Specialist
Penetration Tester

Full-Stack Software Engineer
Backend, Frontend, Mobile

AKADEMIA WIT

Comp. Sci. I – Cloud Computing Technology

Comp. Sci. II - Cloud Computing Architecture and Security

0x04 Insider Threats

Define, Detect, Identify, Assess, Manage

https://www.cisa.gov/topics/physical-security/insider-threat-mitigation

👉 *Employees*

- **Employee errors** due to training issues (accidental data deletion, misconfigurations, sending information to wrong recipients)
- **Negligence**, careless actions
- **HR issues** - layoff employee revenge
- **Covert intelligence** - intentional internal espionage
- **Intentional sabotage**

👉 *Data leakage*

- **Unintentional or intentional disclosure of sensitive information** to unauthorized parties, both internally, and externally.
- **Whistleblowing**

👉 *Bribery*

Bribed by external actors into giving access to information or systems, as well as to perform a certain action.

👉 *Abuse of Privileges*

Misuse of **administrative privileges** or excessive access rights for personal gain or malicious activities.

👉 *Social Engineering*

Attempts **to manipulate or deceive** employees in order to divulge sensitive information, or perform actions compromising security.

👉 *Shadow IT*

- Usage **of out of control file transfer services** (eg. WeTransfer), **not approved cloud** services, Facebook groups, **unencrypted messaging** apps for work.
- Usage of **not inventoried personal devices for work** purposes can introduce security risks, especially if not properly secured and managed.
- Any **unapproved technology** can lead to data breaches and maintenance costs.
- Lack of **oversight / detection / visibility**

👉 *Third party vendors*

Inadequate **oversight** and security measures concerning **third-party vendors and contractors** who have access to an organization's systems and data.

0x05 Advanced Persistent Threats

There is a ghost on the wire

🍒 *Key attributes*

1. ***Nation state-sponsored*** *actors*
2. ***Well-resourced***
3. *Often backed by* ***intelligence agencies***
4. ***Sophisticated***
5. ***Advanced hacking groups***
6. ***Advanced techniques, tools***
7. ***Persistent, undetected*** *for extended period of time*
8. ***Prolonged and targeted*** *cyber attacks*

LAPSUS$

Cyber criminal group known for **bribing** in order to perform **SIM swaps** and steal MFA codes, attacking organizations globally.

NoName057(16)

Self-proclaimed pro-Russian hacktivist group that is known for its **distributed denial-of-service** campaigns targeting entities in Europe, North America, and Japan.

CL0P / FANCYCAT

Known for running multiple criminal activities, including **distributing cyber attacks**, **operating a high-risk exchanger**, **laundering money from dark web operations,** and high-profile cyber attacks such as **Cl0p and Petya ransomware**

APT28 FancyBear/BlueDelta

Russian state-sponsored APT, known for **spear-phishing** campaigns and **zero-day** exploits **(GRU)**

APT37, APT38, Lazarus Group

North Korean state-sponsored APT known for **ransomware** operations, **phishing**, and **zero-day supply chain exploits** usage **(RGB)**

APT39, APT35, APT34

Iranian state-sponsored APT **(MOIS)**

APT10

Chinese state-sponsored APT **(Tiajnjin SSB)**

Equation Group

USA state-sponsored APT, known for **Stuxnet** and **EternalBlue** exploits, as well as **advanced use of cryptography (NSA)**

Unit 8200

Israeli state-sponsored APT **(IDF, Aman)**

0x06 Bad USB attacks

Let's do something practical – a live showcase

A **rubber ducky** is a USB device emulating HID devices, such as a keyboard.

When plugged in, it executes the previously defined commands of an attacker via keyboard input at high speed.

1. **Plug the device in**
2. **C2 agent gets downloaded and executed from attacker's machine** – the payload resides in ***RESOURCE*** part of binary, and was encrypted using **XOR bitwise encryption**. The agent is being decrypted, and then injected into explorer.exe.
3. **Microsoft Defender is unable to detect the payload**

https://github.com/HavocFramework/Havoc
https://github.com/krystianbajno/articles/tree/main/0x04%20Lateral%20Movement/src

0x07 Cryptography – Caesar Cipher

First rule of crypto – don't make your own crypto

- Also known as **ROT13**
- Not secure
- Serves as a substitution cipher example
- Was developed in Ancient Rome

Input	ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz
Output	NOPQRSTUVWXYZABCDEFGHIJKLMnopqrstuvwxyzabcdefghijklm

A + K mod 26
E(x), D(x) = A + 13 mod 26
I am really hard to decrypt -> V nz ernyyl uneq gb qrpelcg

https://quipqiup.com/
Solve cryptograms quickly

https://gchq.github.io/CyberChef/
A very good tool for everything crypto

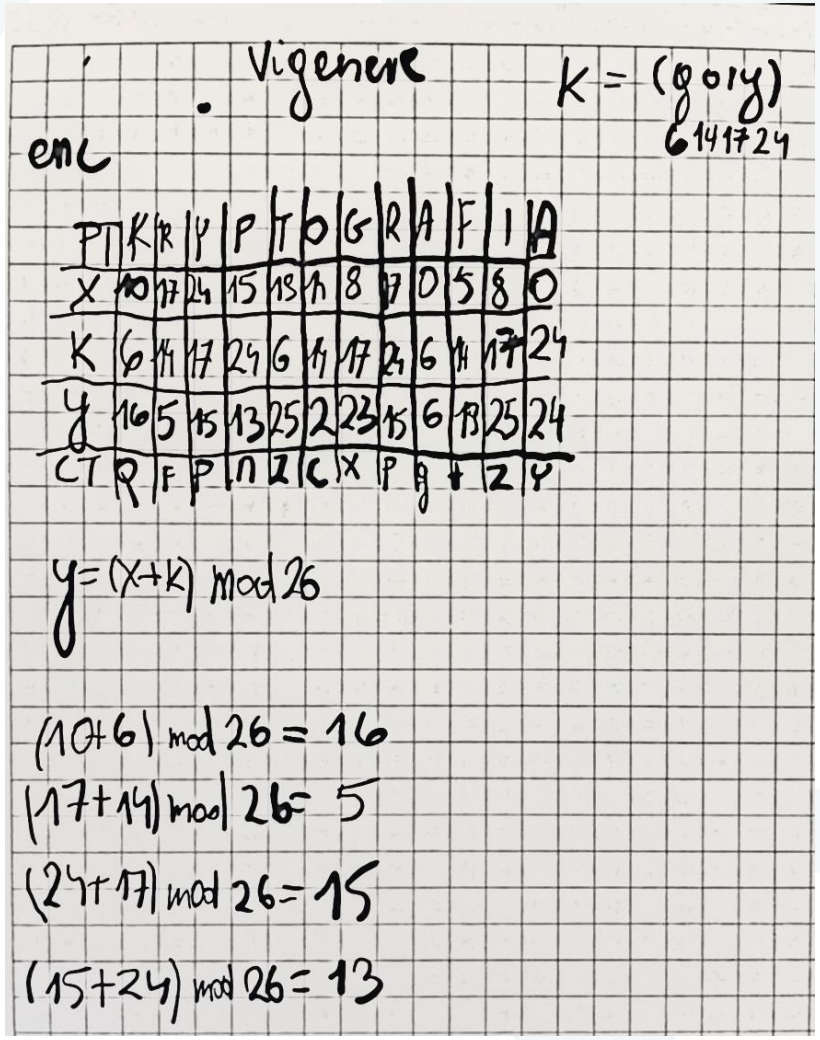
0x08 Cryptography – Vigenère Cipher

Vigenère cipher is a type of substitution cipher similar to Caesar cipher, but for each letter, there is a seperate key character.

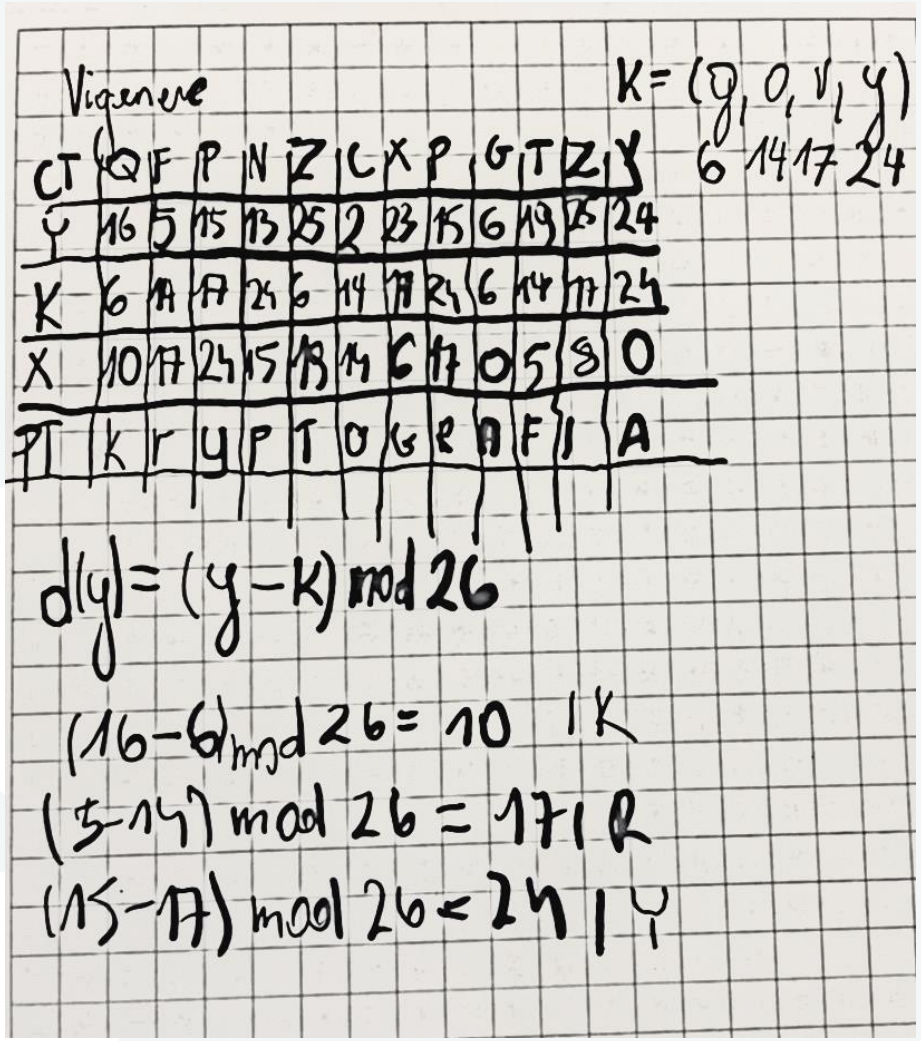
It can be **an unbreakable cipher**, given the key is at least the length of plaintext, the key is completely random, and is never reused.

(One time pad)

Encryption



Decryption



0x09 Cryptography – Affine Cipher

Affine cipher is a type of monoalphabetic substitution cipher, where each letter is encrypted using a simple mathematical function.

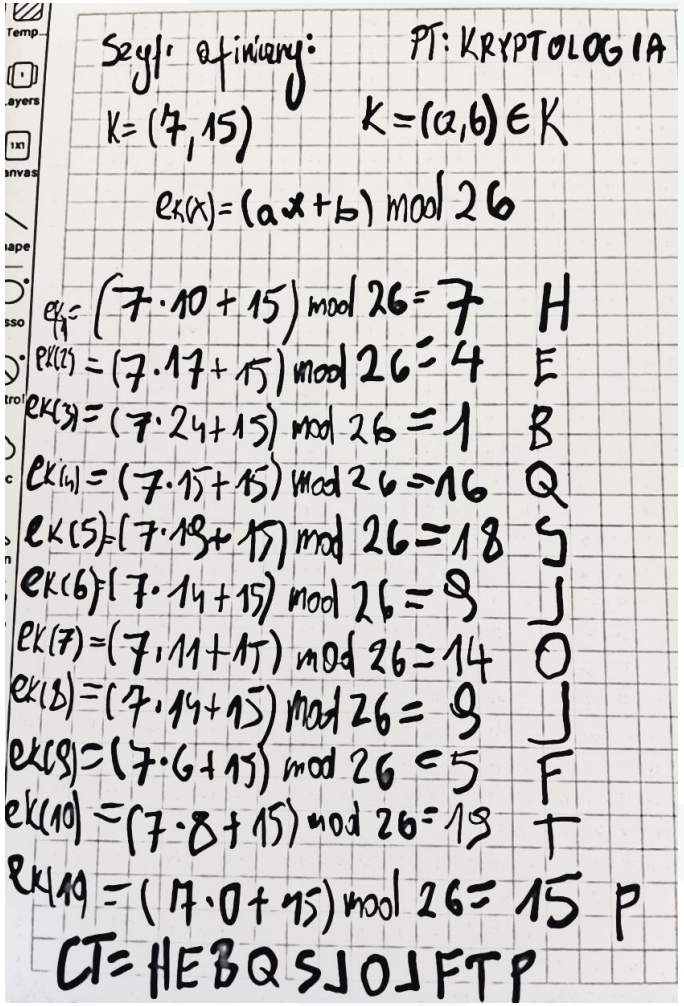
There are a total of 312 possible keys for 26 character alphabet. This number comes from the fact, that there are 12 numbers that are coprime with 26 that are less than 26.

Therefore, this cipher is not sufficient for use, and is easy to brute force.
This cipher is also susceptible to known plaintext attacks, when part of the plaintext is known.

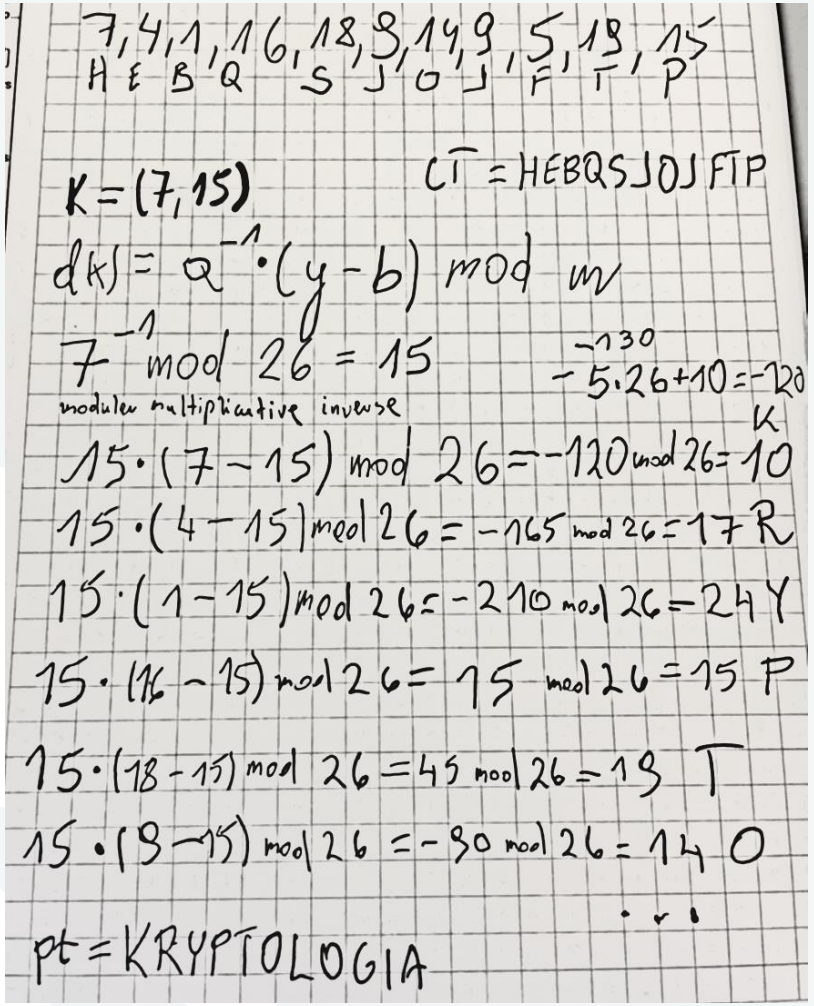
K = (a,b)

GCD(a, m) must be equal to 1, where m is number of characters in alphabet (eg. 26) and *a* is part of key.

Encryption



Decryption



0x0A Cryptography – Hill Cipher

Hill cipher is a polygraphic substitution cipher (more than one letter at once) based on linear algebra. The key is stored in a matrix, and letters are stored in vectors.

P(X) = X*K mod 26

D(X) = X * (K^-1 mod 26) mod 26

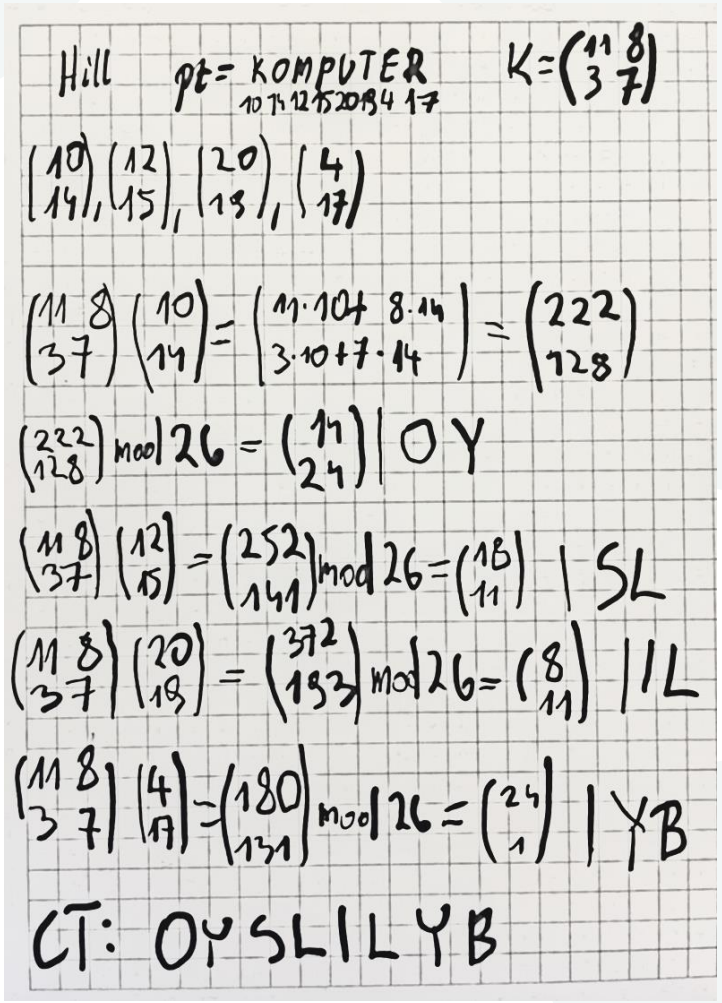
Where the numbers of rows in square K matrix is the length of X vectors.

As the dimension increases, the cipher rapidly becomes infeasible for a human to operate by hand, yet Hill cipher is susceptible to known plaintext attacks.

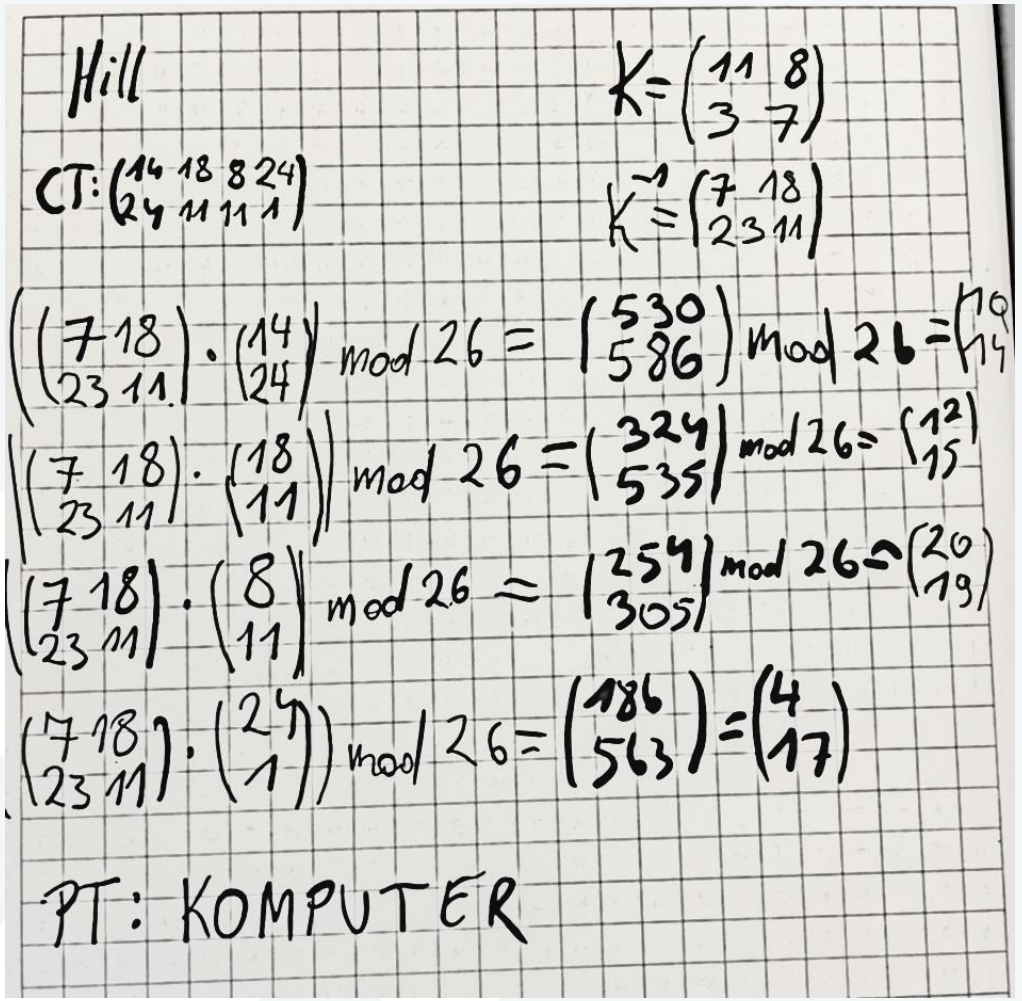
Hill cipher serves as example of using Matrix multiplication.

- **AES** MixColumns step is a matrix multiplication.
- **Twofish** is a combination of non-linear substitution-boxes with a matrix multiplication.

Encryption



Decryption



0x0B Cryptography – Rijndael Block Cipher

Advanced Encryption Standard

AES is a block cipher with **fixed block size of 128 bits**, and a **key size of 128, 192, or 256 bits**. It operates on **4x4 column-major order array of 16 bytes**, termed as state. **The block size is always 16 bytes**.

The key size specifies number of transformation rounds:

10 rounds for 128-bit keys – AES128

12 rounds for 192-bit keys – AES192

14 rounds for 256-bit keys – AES256.

The cipher needs an **IV initialization vector with a size of 16 bytes (the first block)** in order to randomize the encryption and produce distinct ciphertext even if the same plaintext is encrypted multiple times. **The intialization vector then becomes part of the ciphertext as first 16 bytes.**

It is often used in malware in order to encrypt the payload and hide from AV engines. The software that encrypts the payload is called a „Crypter".

```
import crypto from 'crypto';

const message = "Hello world"

// Generate a random symmetric key
const symmetricKey = crypto.randomBytes(32); // 256 bits for AES

// Encrypt the message with the symmetric key
const iv = crypto.randomBytes(16); // Initialization vector for AES-256-CBC
const cipher = crypto.createCipheriv('aes-256-cbc', symmetricKey, iv);
const encryptedMessage = Buffer.concat([cipher.update(message, 'utf-8'), cipher.final()]);

console.log(encryptedMessage)
```

Encryption

```
// Extract the IV from the encrypted message (first 16 bytes)
const iv = encryptedMessageBuffer.slice(0, 16);

console.log('IV:', iv.toString('hex'));

// Decrypt the message using the symmetric key and IV
const decipher = crypto.createDecipheriv('aes-256-cbc', symmetricKey, iv);

// Update the decipher with the rest of the encrypted message
const decryptedMessage = Buffer.concat([decipher.update(encryptedMessageBuffer.slice(16)), decipher.final()]);

console.log('Decrypted Message:', decryptedMessage.toString('utf-8'));
```

Decryption

It operates in several modes, the one most often used is **CBC**, which encrypts each block with the previous block. **ECB** on the other hand, encrypts each block independently, and is considered less secure.

0x0C Cryptography – bitwise XOR

XOR cipher is a type of bitwise additive cipher, and operates according to the following table:

A	B	Q
0	0	0
0	1	1
1	0	1
1	1	0

Key = „unbreakable" = 01110101 01101110 01100010 01110010 01100101 01100001 01101011 01100001 01100010 01101100 01100101
Text = „crypt" = 01100011 01110010 01111001 01110000 01110100

Cipher = 01100011 01110010 01111001 01110000 01110100 ^ 01110101 01101110 01100010 01110010 01100101 = **00010110 00011100 00011011 00000010 00010001**

XOR operator is extremely common as a component in more complex ciphers.

It is often used in malware in order to encrypt the payload and hide from AV engines. The software that encrypts the payload is called a „Crypter".

It can be an unbreakable cipher, given the key is at least the length of plaintext, the key is completely random, and is never reused. (One Time Pad)

```python
# XOR encryption key (replace with your own)
key_hex = "4f130123a70d83b551efed9191e71a30ef5ed5dc660c5cbe

# convert the key to bytes
key = bytes.fromhex(key_hex)

# XOR encryption
def xor(data, key):
    encrypted_data = bytearray(len(data))
    for i in range(len(data)):
        encrypted_data[i] = data[i] ^ key[i % len(key)]

    return bytes(encrypted_data)
```

Encryption

```cpp
// XOR decryption function
void xorDecrypt(char* data, size_t dataSize, const std::vector<uint8_t>& key) {
    for (size_t i = 0; i < dataSize; i++) {
        data[i] ^= key[i % key.size()];
    }
}
```

Decryption

0x0D Steganography

Make it disappear

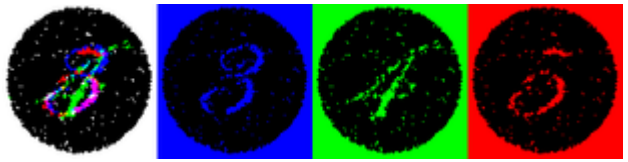
Steganography involves hiding the message in a carrier – such as image, audio file, or other medium.

For example – hiding the message in an audio spectrogram, or on the least significant bit of the bytes of an image. It is possible to use **storage/volumes** or **time** (or to be more precise – delay between messages) as another channel to store data in.

The **combination of steganography and cryptography** effectively hides information from prying eyes and ensures the privacy of information.

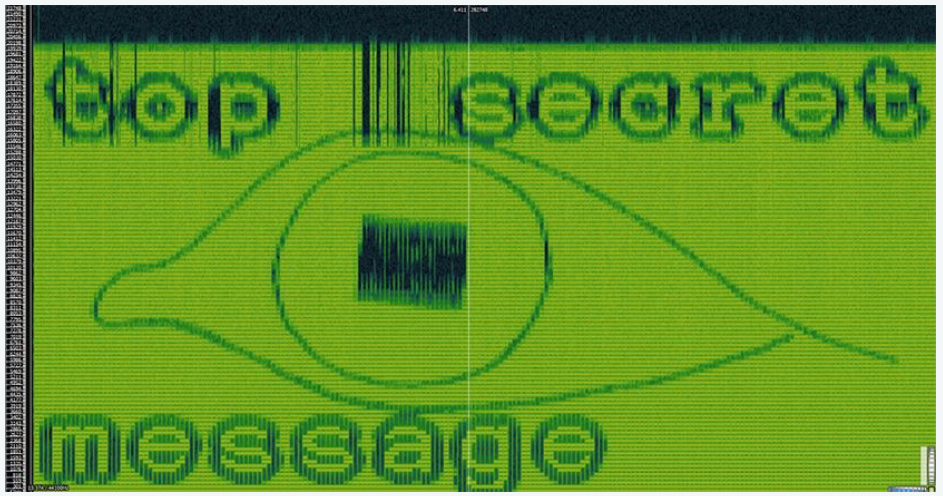
Printers employ steganography in order to save model, serial numbers, and timestamps on each printout for traceability reasons.

Source: https://en.wikipedia.org/wiki/Steganography



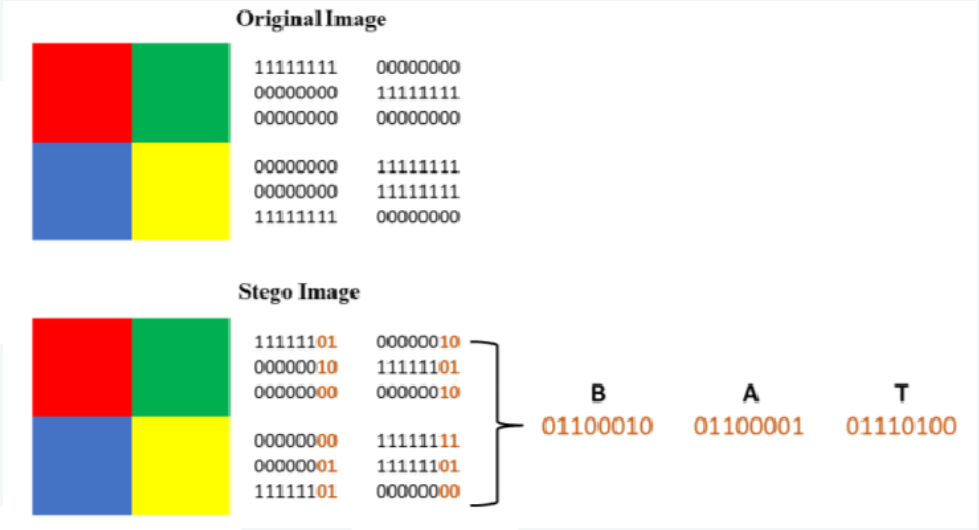
Colour channels steganography

Source: https://solusipse.net/blog/post/basic-methods-of-audio-steganography-spectrograms/



Spectrogram

Source: https://www.researchgate.net/figure/Least-Significant-Bit-Steganography_fig1_366169511



Least Significant Bit

0x0E Internal threat live crypto-example

Let's do something practical – a live showcase

1. **Create** a key for XOR based one-time pad encryption
2. **Encrypt** the data using one-time pad
3. **Write** the data into a .png image using Least Significant Bit steganography
4. **Attempt to bypass** Data Loss Prevention system and extract sensitive data

0x0F Falling into a rabbit hole

Win a bunny

1. **What** were the famous last words of Julius Caesar?
2. **Who** disclosed NSA highly classified information in 2013?
3. **What** is the name of the system disclosed?
4. **Which** Tech/EV automobile company employees leaked PII data to foreign media outlet in 2023? (If you don't know, guess)
5. **K=F F=X C=Q L=D** EUM CBGHT DWNPY KNF XBZIJ NSMW EUM QORA LNV

https://quipqiup.com/

0x10 Q&A

Ask me anything you want

0x11 Thank you

Presentation in PDF format is available on **https://news.baycode.eu**

*Meet me again at **https://wguisw.org***