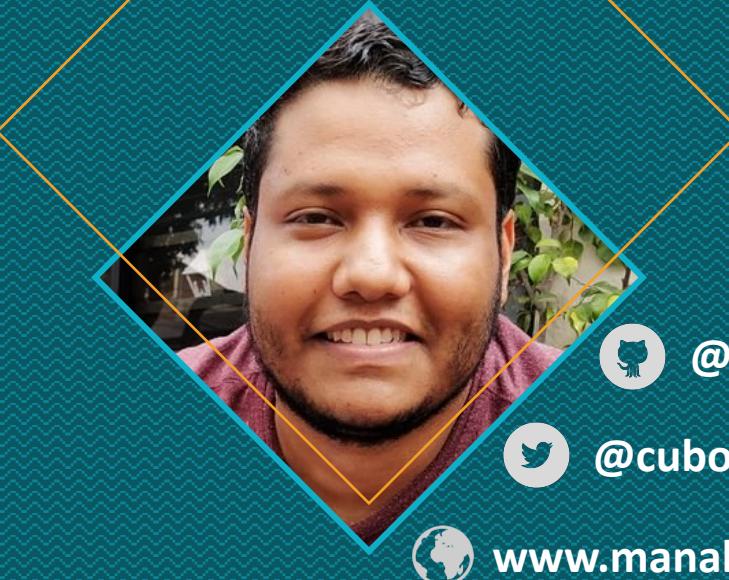


STEGANOGRAPHY

HIDING DATA WITHIN DATA



@cubosensei



@cubosensei

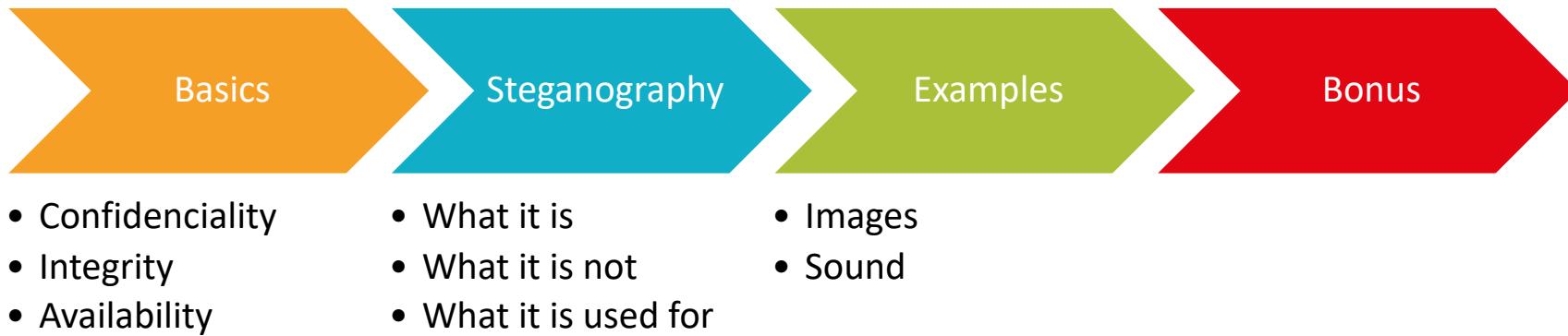


www.manalco.co

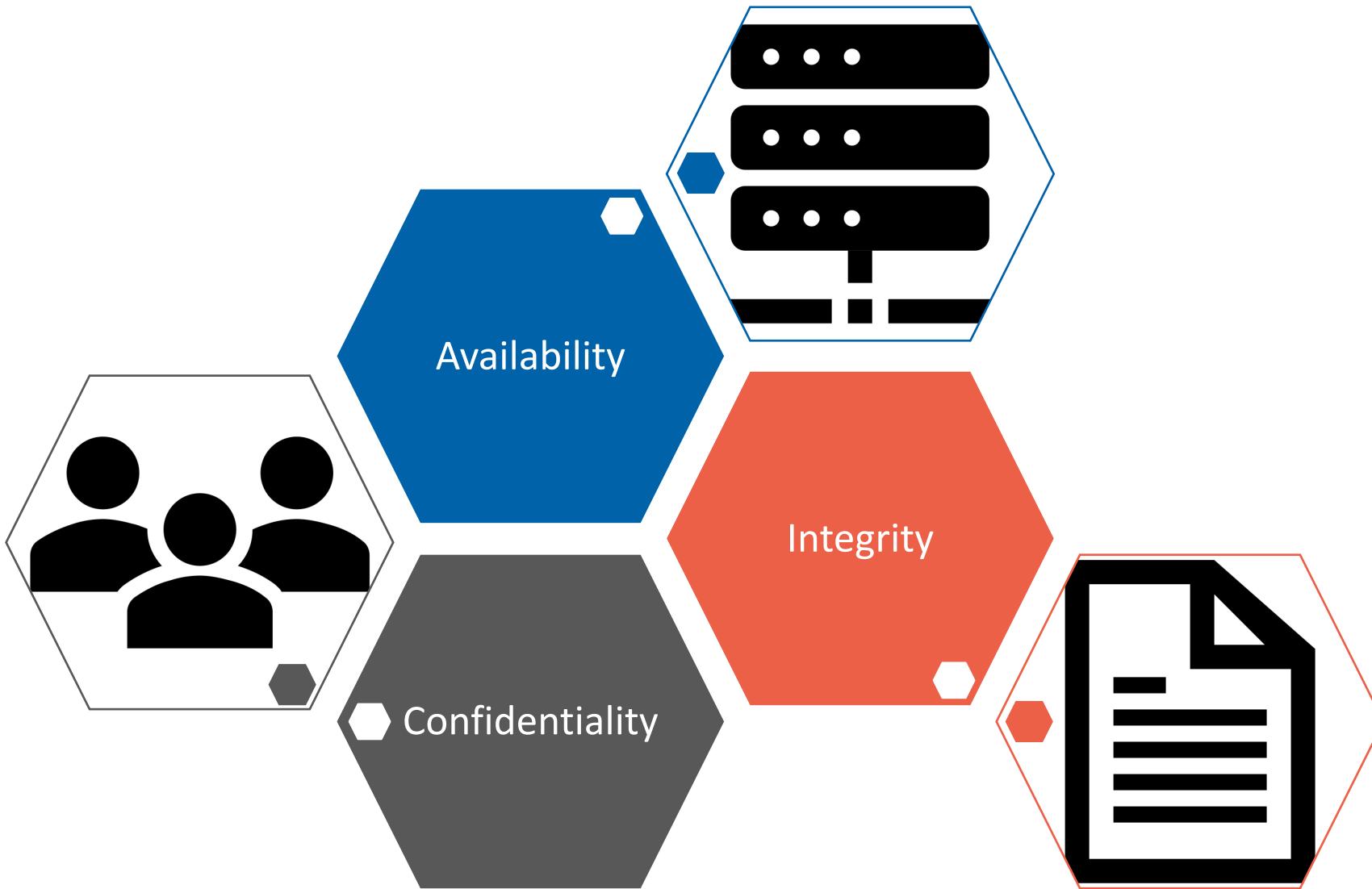
MANUEL ALVARADO

Web Developer, Systems Engineer, MSc. Information Security

Agenda



Basics



CIA Triad: Availability



- Hardware/Service oriented.
- If the resource is not available when needed it is a problem.

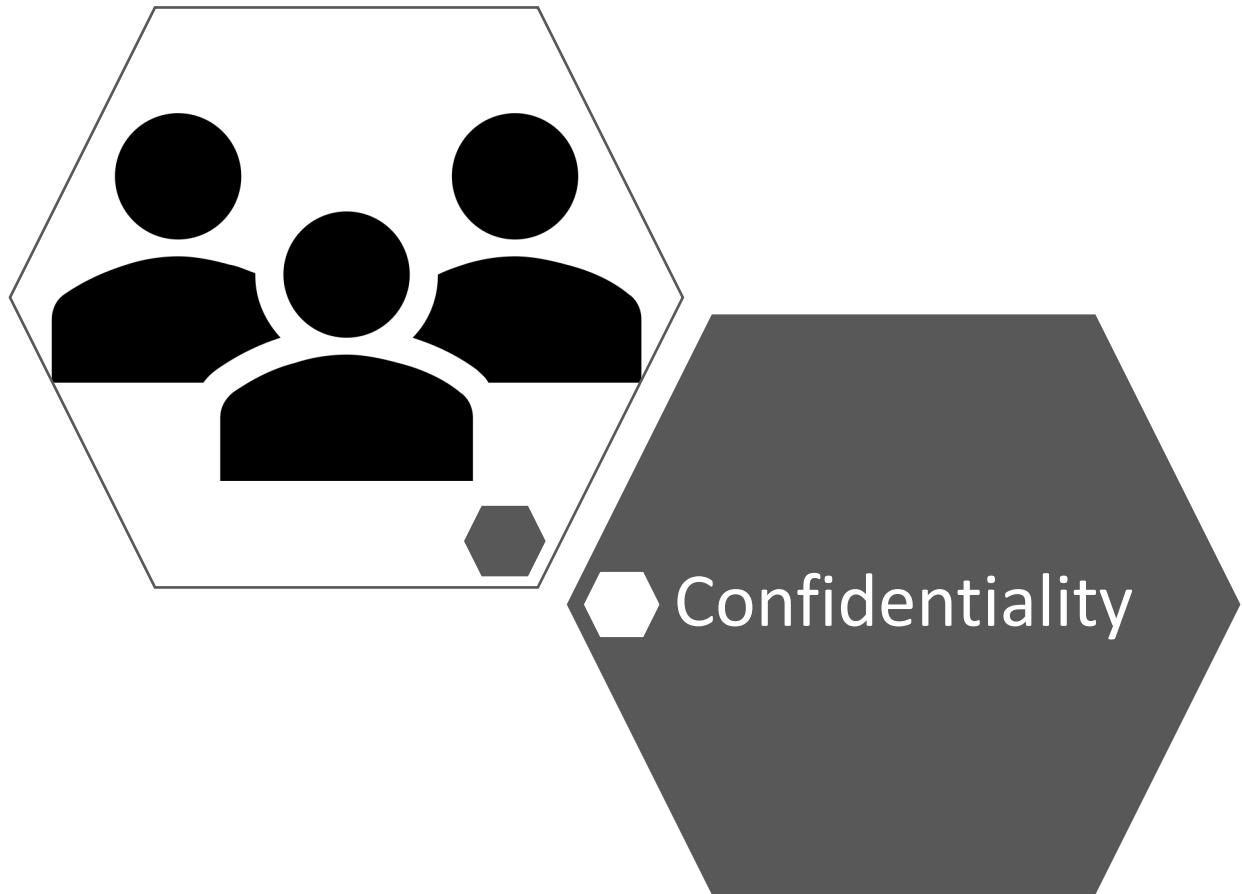
CIA Triad: Integrity



Only authorized users are able to _____ the asset.

- Update/Modify
 - Only in determined ways.

CIA Triad: Confidentiality



Only authorized users are able to _____ the asset.

- Access
- Read
- Print
- Even know of its existence

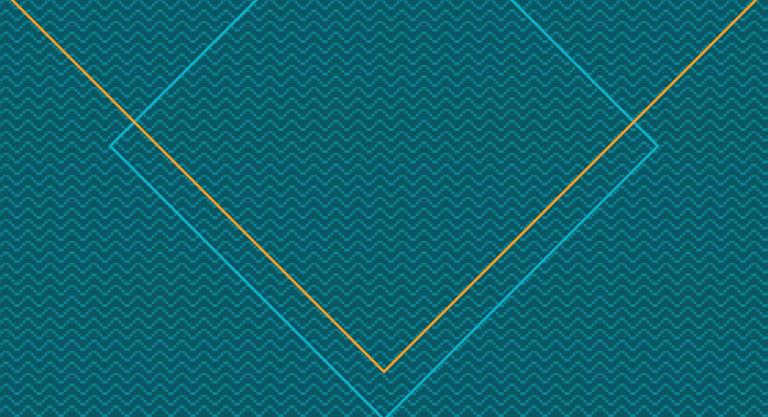
Cryptography

Iijv: Isbiik

KEY

Cryptography

- Secure communications.
- Information confidentiality.
- Enigma Machine.



STEGANOGRAPHY

(NOW FOR REAL)

What it is



PRACTICE

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

-Oxford Dictionary

ART

"Steganography is the art of hiding information within a disclosed element."

-?

What it is NOT



METHOD

"A particular procedure for accomplishing or approaching something, especially a systematic or established one."

-Oxford Dictionary

TECHNIQUE

"A way of carrying out a particular task, especially the execution or performance of an artistic work or a scientific procedure."

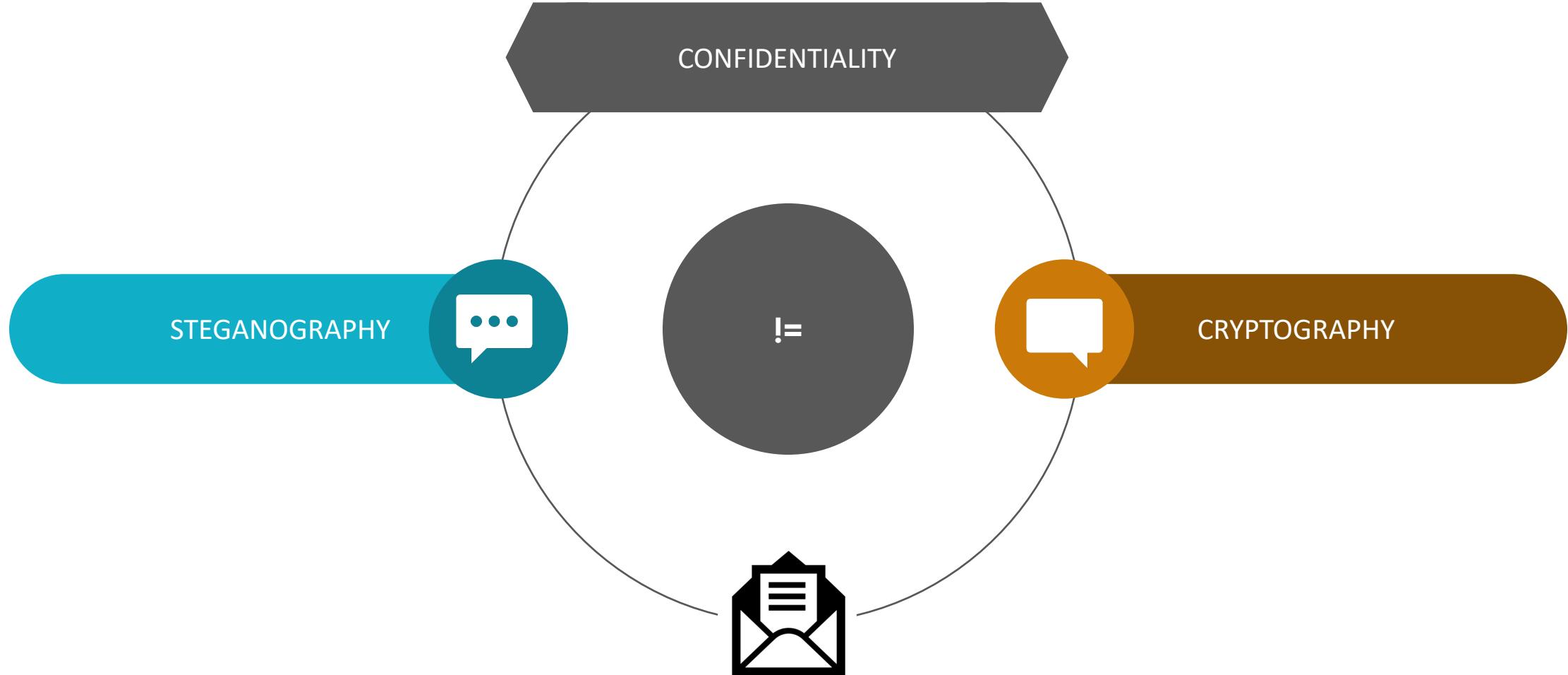
-Oxford Dictionary

CRYPTOGRAPHY

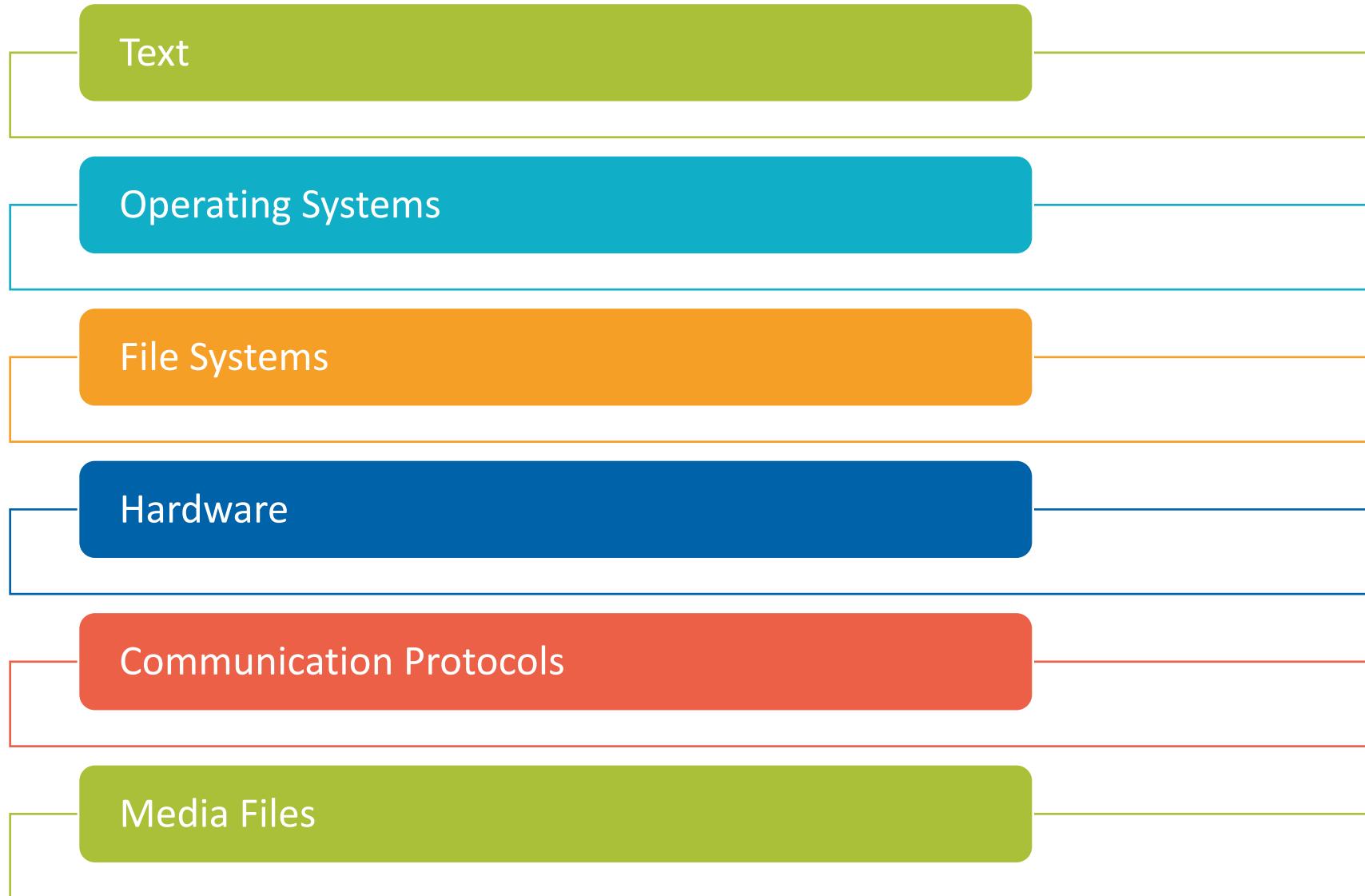
"Is the practice and study of techniques for secure communication."

-Wikipedia

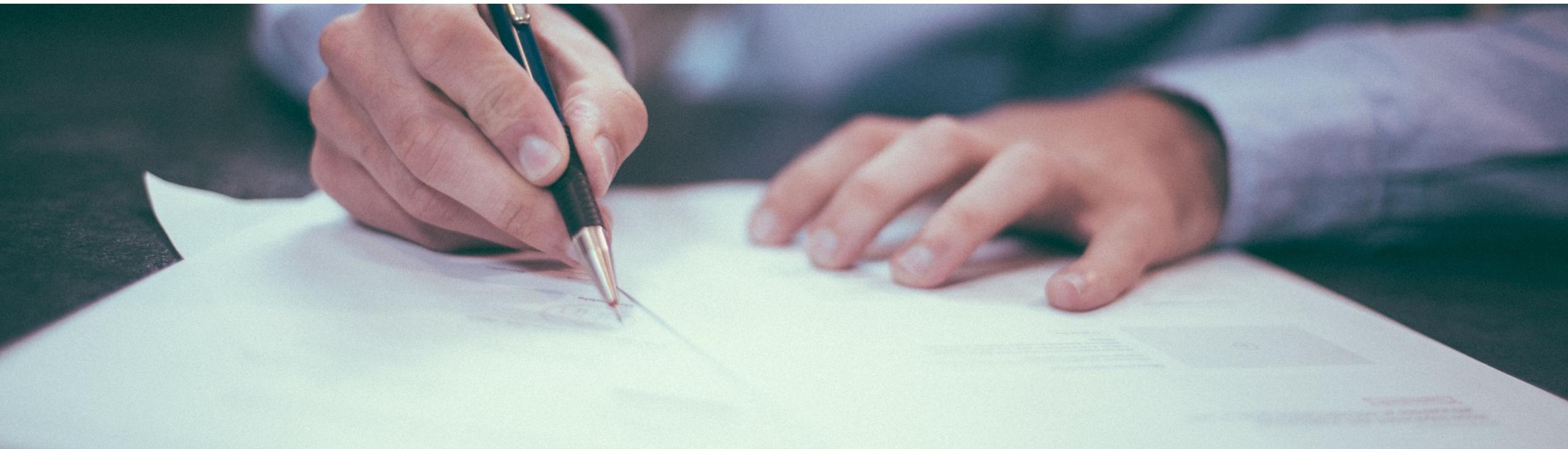
What it is NOT



Where is it used



What it is for



ACTUAL MESSAGE FROM GERMAN SPY IN WWII

Apparently neutral's protest is thoroughly discounted and ignored. Isman hard hit. Blockade issue affects pretext for embargo on by products, ejecting suets and vegetable oils.

Steganography

What it is for



ACTUAL MESSAGE FROM GERMAN SPY IN WWII

Apparently neutral's protest is thoroughly discounted and ignored. Isman hard hit. Blockade issue affects pretext for embargo on by products, ejecting suets and vegetable oils.

TAKING THE SECOND LETTER FROM EACH WORD

Pershing sails from NY June 1.

Hiding Data



STEGANOGRAPHY WITH IMAGE FILES

An image is just a set of pixels.

- Each pixel represents RGB with 3 bytes.
 - (0,0,0) is black and (255,255,255) is white.
- So it makes it possible to hide the info in the **Least Significant Bit (LSB)** and the variation may be imperceptible.

STEGANOGRAPHY WITH AUDIO FILES

MIDI files have 16 channels (0-15) for tracks. Usually a file does not use all the channels, so it makes it possible to code a message in an empty channel.

Examples

Hiding Data: Least Significant Bit

ORIGINAL

Having two pixels from an RGB image



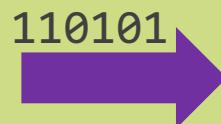
R=132, G=134, B=137
(10000100, 10000110, 10001001)



R=141, G=121, B=101
(10001101, 01111001, 01100101)

MODIFIED

It is possible to hide the message 110101 in the least significant bit (red)



R=133, G=135, B=136
(10000101, 10000111, 10001000)



R=141, G=120, B=101
(10001101, 01111000, 01100101)

Hiding Data: Inside Image Files



IMAGE WITHOUT MESSAGE

The original file



IMAGE WITH MESSAGE

This image has the message:

HELLO WORLD

Examples



Examples

Hiding Data: Inside Image Files The Easy Way

```
from stegano import lsb  
  
secret = lsb.hide("~/mario1.png", "HELLO WORLD")  
secret.save("~/mario2.png")
```

ENCODE

```
from stegano import lsb  
  
clear_message = lsb.reveal("~/mario2.png")  
print(clear_message)
```

DECODE

Examples

Hiding Data: Inside Image Files



```
# pip3 install opencv-python

import sys
import cv2

clear_text = sys.argv[2]+"-msg-"
message = ''.join([ format(ord(i), "08b") for i in clear_text ])

image = cv2.imread(sys.argv[1])

data_index = 0
data_len = len(message)
```

ENCODE



```
for row in image:
    for pixel in row:
        r = format(pixel[0], "08b")
        g = format(pixel[1], "08b")
        b = format(pixel[2], "08b")

        if data_index < data_len:
            pixel[0] = int(r[:-1] + message[data_index], 2)
            data_index += 1
        if data_index < data_len:
            pixel[1] = int(g[:-1] + message[data_index], 2)
            data_index += 1
        if data_index < data_len:
            pixel[2] = int(b[:-1] + message[data_index], 2)
            data_index += 1
        if data_index >= data_len:
            break

cv2.imwrite("output.PNG", image)
print(">>> Image encoded.")
```

Examples

Hiding Data: Inside Image Files

```
● ● ●

import sys
import cv2

image = cv2.imread(sys.argv[1])
binary_data = ""
for row in image:
    for pixel in row:
        r = format(pixel[0], "08b")
        g = format(pixel[1], "08b")
        b = format(pixel[2], "08b")
        binary_data += r[-1]
        binary_data += g[-1]
        binary_data += b[-1]

all_bytes = [ binary_data[i: i+8] for i in range(0, len(binary_data), 8) ]
decoded_data = ""
for byte in all_bytes:
    decoded_data += chr(int(byte, 2))
    if decoded_data[-5:] == "-msg-":
        break
print("">>>> Decoded data:")
print(decoded_data[:-5])
```

DECODE

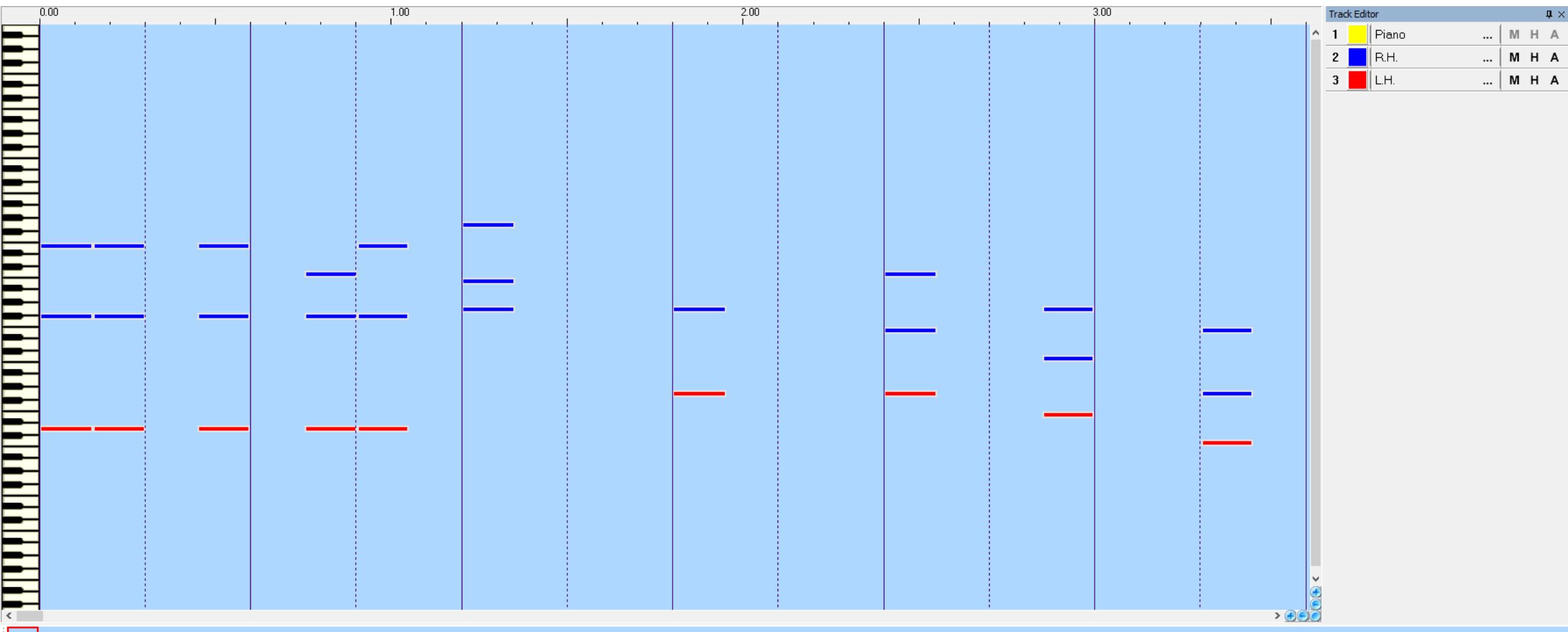
Examples

Hiding Data: Inside Image Files

LIVE DEMO

Examples

Hiding Data: Inside Audio Files



Examples

Hiding Data: Inside Audio Files

```
import sys
import midi

file_name = sys.argv[1]
message = sys.argv[2]
midi_file = midi.read_midifile(file_name)
track = midi.Track()
channels = {}
channel = 0
```

```
#Look for a free channel
for i in midi_file:
    for j in i:
        try:
            j.channel
        except:
            pass
        else:
            if channels.has_key(j.channel):
                channels[j.channel] += 1
            else:
                channels[j.channel] = 1

    for c in range(0,15):
        if not channels.has_key(c):
            channel = c
            break
```

ENCODE

Examples

Hiding Data: Inside Audio Files

```
message = message.upper()
message = "-msg-"+message

#Encode the message in the selected channel
for character in list(message):
    note = midi.NoteOnEvent(tick=0, channel=channel, data=[(ord(character)), 0])
    track.append(note)

#write the new MIDI file
midi_file.append(track)
file_name = file_name.replace('.mid', '_encoded.mid')
midi.write_midifile(file_name, midi_file)

print("Message has been coded into : "+file_name+" in channel "+str(channel))
```

ENCODE

Examples

Hiding Data: Inside Audio Files

```
● ● ●  
  
import sys  
import midi  
  
midi_file = midi.read_midifile(sys.argv[1])  
message = ""  
channels = {}  
channel = -1
```

DECODE

Examples

Hiding Data: Inside Audio Files

```
● ● ●  
  
#Look for used channels  
for i in midi_file:  
    for j in i:  
        try:  
            j.channel  
        except:  
            pass  
        else:  
            if channels.has_key(j.channel):  
                channels[j.channel] += 1  
            else:  
                channels[j.channel] = 1  
  
for c in channels:  
    initializer = ""  
    for track in midi_file:  
        for event in track:  
            if isinstance(event, midi.NoteOnEvent) and event.channel == c and len(initializer) < 5:  
                initializer += chr(event.data[0])  
            else:  
                if initializer == "-msg-":  
                    channel = c  
                    break  
            if channel >= 0:  
                break  
    if channel >= 0:  
        break
```

DECODE

Examples

Hiding Data: Inside Audio Files



```
for track in midi_file:
    for event in track:
        if isinstance(event, midi.NoteOnEvent) and event.channel == channel:
            message += chr(event.data[0])

message = message.replace('-msg-', '')

print("Message is: '" + message + "'")
```

DECODE

Examples

Hiding Data: Inside Audio Files

Bonus



AVERAGE PLAYER VISUALIZATION



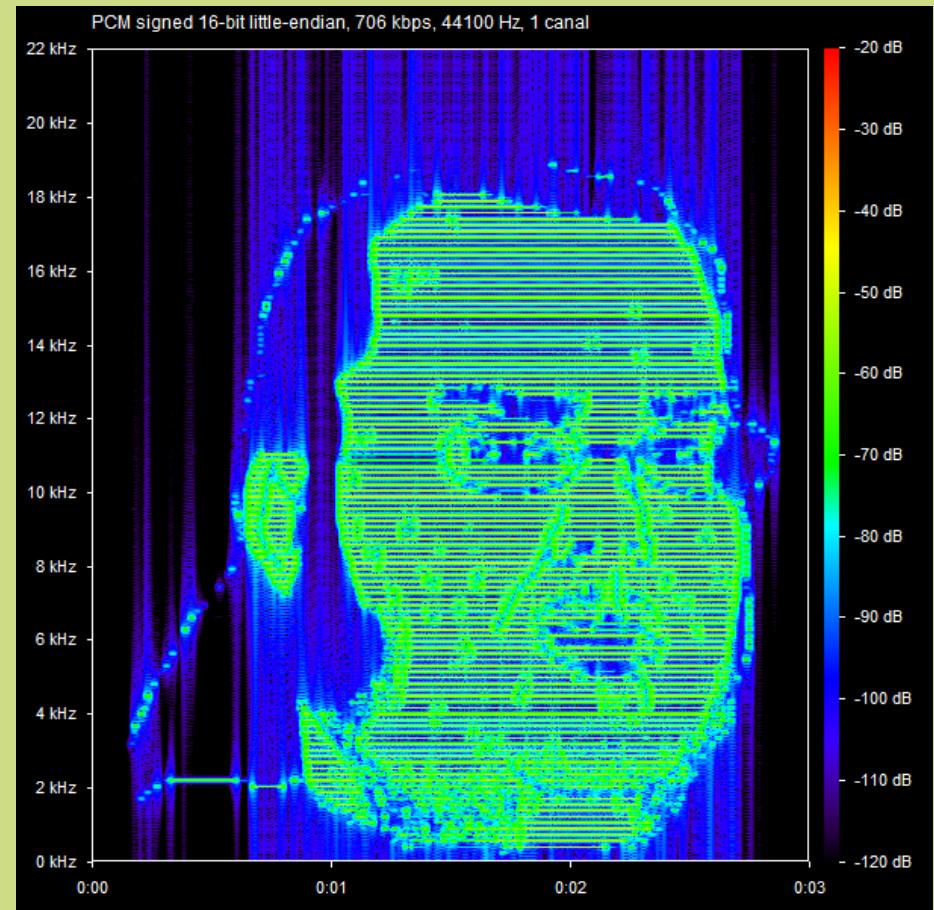
BONUS

Hiding Data: Inside Audio Files

Bonus



AVERAGE PLAYER VISUALIZATION



ACOUSTIC SPECTRUM

BONUS

Bonus: Challenge



https://dominicbreuker.com/post/stego_book_of_secrets/

BONUS



Thank You

Manuel Alvarado
www.manalco.co