Lab2

Student name: داود محمد نحاس

Student ID: 2240495

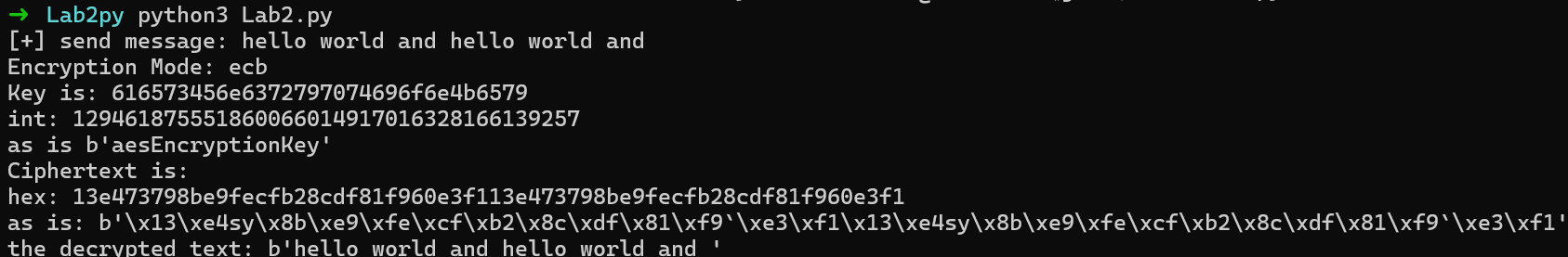
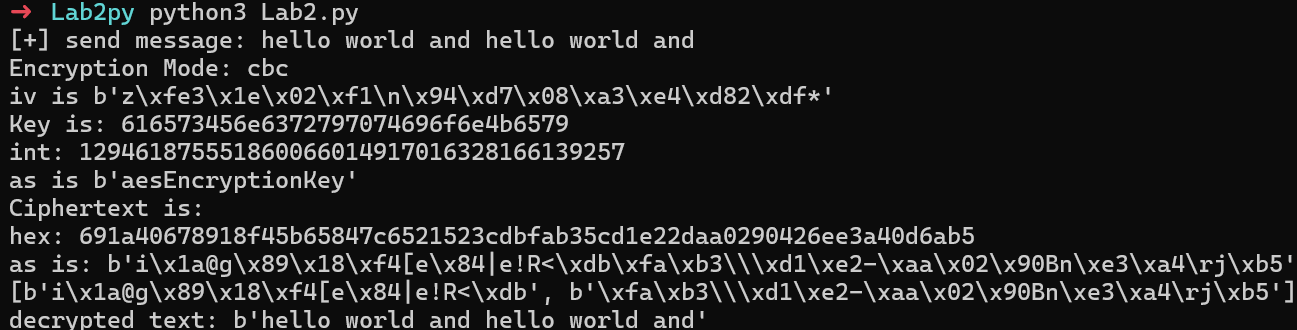
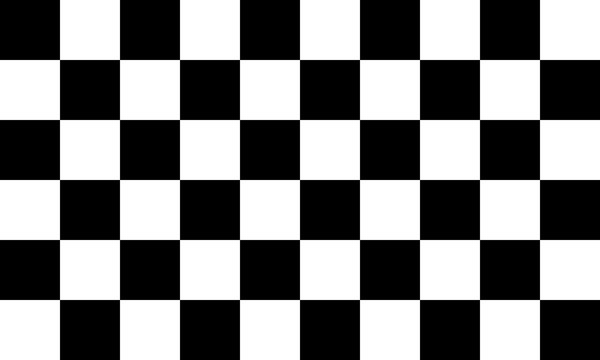
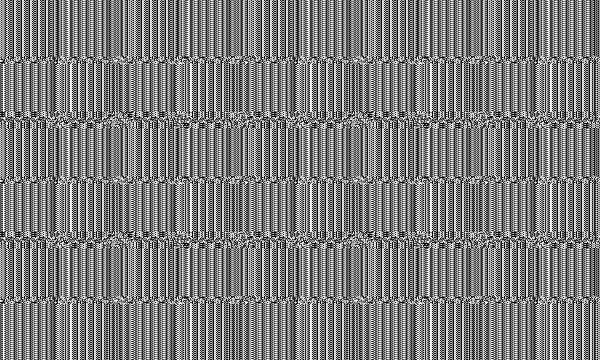
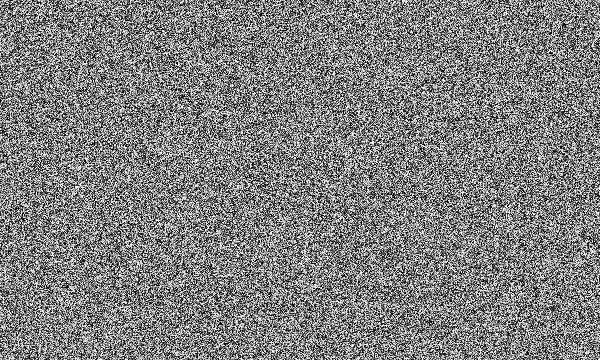
Step1:

1. Overview of AES and block cipher modes:
   1. The AES (Advanced Encryption Standard) algorithm uses the blocks method. It uses 128 bit of block size for the text and 128,192 or 256 bit key length
   2. The modes to be discussed are ECB, CBC, CFB, OFB, CTR and GCM. As of ECB is the basic implementation of AES so it will only have the basic security of AES. But when we talk about CBC we move to a new security level where the security is now depend on another factor which's the Initial Vactor. And for the CFB is also using IV but instead of encrypting the text with the AES/ECB it encrypt the key starting with IV. Where OFB is more like CFB but instead of taking the cipher text as a feed back for the next cycle it takes the encrypted key. And for the CTR it takes IV/Nonce/Serial number as increment and encrypting it and making each block encrypted with encrypted key, adding to it an authentication feature we can get the GCM
2. Why ECB Mode is Insecure?
   1. The ECB mode use basic AES encryption where it only depends on the key only and with no second factor like the IV making it have a higher possibility to find more information from the key and allowing the attacker to lower the value of the key space
   2. along the known-plaintext attack and chosen-plaintext we can find the pattern recognition attack where the attacker can exploit repeated patterns in the plaintext due, also the reply attack where the attacker can reply encrypted blocks to produce the same output
3. How CBC Mode Mitigates ECB Weaknesses?
   1. The ECB mode makes the ciphertext depend on the key and IV and the previous ciphertext resulting in diffusion that was absent in EBC and resulting in making the attacks (that was mentioned previously) Harder to and close to impossible if not is to success
4. Conclusion
   1. The risks of ECB mode is the result of the absent of the second factor IV and chain method used in CBC making the ciphertext and the plaintext a block to block mapped encryption, the use of the other modes provides what was lacked in EBC mode and making the dependency of the encryption on 3 factors and introduce diffusion in ciphertext (not in encrypting)

Step2:

1. the code can't fit in the document as is
2. The input Is 'hello world and hello world and '

Step3&4:

1. Ecb:
2. Cbc:
3. As we can see in the picture if we give an input of 16 characters repeated we can find a pattern In the output
4. Encrypt the images:
   1. 
   2. 
   3. 

رسم توضيحي 1 ECB above, CBC Down

* 1. As we can see there's a pattern when using the ECB mode and doesn't really show on CBC

import numpy as np

import matplotlib.pyplot as plt

def visualize\_encrypted\_image(filename, width, height):

with open(filename, 'rb') as f:

data = f.read()

# Convert binary data to an array of bytes

array = np.frombuffer(data, dtype=np.uint8)

# Resize the array to the desired dimensions (width, height)

# Pad or truncate if necessary

if array.size < width \* height:

array = np.pad(array, (0, width \* height - array.size), 'constant')

else:

array = array[:width \* height]

# Reshape the array to match the image dimensions

image = array.reshape((height, width))

# Display the image

plt.imshow(image, cmap='gray', vmin=0, vmax=255)

plt.axis('off')

plt.title(filename)

plt.show()

# Set the width and height according to your original image dimensions

visualize\_encrypted\_image('cbc\_enc\_img.jpg', 256, 256) # Example dimensions for ECB

visualize\_encrypted\_image('img.jpg', 256, 256) # Example dimensions for CBC

Step5:

1. The weaknesses reside in being dependability of the blocks on their each other sequentially making it a no use of improvement
2. The suggestion I made in this report is using LFSR generator and set the key as its seed, and making an array of size of numbers of the blocks and Xor'ing it with the block of its order, this make no need to use Initial vector in the algorithm and only depend on the key
   1. A screenshot of a computer

      Description automatically generated
   2. A black and white rectangle with white text

      Description automatically generated
3. The code can't fit in this document so it will be impeded with the report
4. The Native CBC depend on a key and an Initial vector, where the new CBCd uses only one key and generate a list of numbers that get Xor'ed with the blocks