import sys

import cv2

import numpy as np

from Cryptodome.Cipher import AES

from Cryptodome.Util.Padding import pad, unpad

from Cryptodome.Random import get\_random\_bytes

mode = AES.MODE\_ECB

if mode != AES.MODE\_CBC and mode != AES.MODE\_ECB:

print('Only CBC and ECB mode supported...')

sys.exit()

keySize = 32

ivSize = AES.block\_size if mode == AES.MODE\_CBC else 0

imageOrig = cv2.imread("secret.jpg")

rowOrig, columnOrig, depthOrig = imageOrig.shape

cv2.imshow("Original image", imageOrig)

keyPress = cv2.waitKey(20)

imageOrigBytes = imageOrig.tobytes()

key = get\_random\_bytes(keySize)

iv = get\_random\_bytes(ivSize)

cipher = AES.new(key, AES.MODE\_CBC, iv) if mode == AES.MODE\_CBC else AES.new(key, AES.MODE\_ECB)

imageOrigBytesPadded = pad(imageOrigBytes, AES.block\_size)

ciphertext = cipher.encrypt(imageOrigBytesPadded)

paddedSize = len(imageOrigBytesPadded) - len(imageOrigBytes)

void = columnOrig \* depthOrig - ivSize - paddedSize

ivCiphertextVoid = iv + ciphertext + bytes(void)

imageEncrypted = np.frombuffer(ivCiphertextVoid, dtype = imageOrig.dtype).reshape(rowOrig + 1, columnOrig, depthOrig)

cv2.imshow("Encrypted image", imageEncrypted)

keyPress = cv2.waitKey(20)

imageOrigBytes = imageOrig.tobytes()

rowEncrypted, columnOrig, depthOrig = imageEncrypted.shape

rowOrig = rowEncrypted - 1

encryptedBytes = imageEncrypted.tobytes()

iv = encryptedBytes[:ivSize]

imageOrigBytesSize = rowOrig \* columnOrig \* depthOrig

paddedSize = (imageOrigBytesSize // AES.block\_size + 1) \* AES.block\_size - imageOrigBytesSize

encrypted = encryptedBytes[ivSize : ivSize + imageOrigBytesSize + paddedSize]

cipher = AES.new(key, AES.MODE\_CBC, iv) if mode == AES.MODE\_CBC else AES.new(key, AES.MODE\_ECB)

decryptedImageBytesPadded = cipher.decrypt(encrypted)

decryptedImageBytes = unpad(decryptedImageBytesPadded, AES.block\_size)

decryptedImage = np.frombuffer(decryptedImageBytes, imageEncrypted.dtype).reshape(rowOrig, columnOrig, depthOrig)

cv2.imshow("Decrypted Image", decryptedImage)

keyPress = cv2.waitKey(20)

cv2.destroyAllWindows()