ADVANCED DATABASE MANAGEMENT SYSTEM

REPORT: Image Encryption And Decryption

FROM:

GROUP 12

ANIMESH KUMAR	**USN: 01JST18IS003**
HARSHIT KUMAR JAIN	**USN: 01JST18IS016**
SUMEDHA	**USN: 01JST18IS051**
MUHAMMAD DANISH KIDWAI	**USN: 01JST18IS026**
SHREYAS TIWARI	**USN: 01JST18IS046**

DEPT. OF INFORMATION SCIENCE AND ENGINEERING (IS&E)

IV SEMESTER

JSSSTU, MYSURU

TO:

PROF. MEETHA GURU

DEPT. OF INFORMATION SCIENCE AND ENGINEERING (IS&E)

JSSSTU, MYSURU

Introduction

In recent years, with the rapid development of computer technology, digital image processing technology has also rapidly developed and penetrated into all aspects of life, such as remote sensing,industrial detection, medicine, meteorology, communication, investigation, intelligent robots, etc.Therefore, image information has attracted widespread attention. Image data security is very important,especially in the special military, commercial and medical fields. Image encryption has become one of the ways to protect digital image transmission. However, the image data has the characteristics of large amounts of data, strong correlation and high redundancy, which lead to low encryption efficiency and low security, so the traditional encryption algorithms, such as Data Encryption Standard (DES)and Advanced Encryption Standard (AES), cannot meet the needs of image encryption . Chaos has the characteristics of high sensitivity to the initial conditions and system parameters, no periodicity,pseudo randomness, ergodicity and chaotic sequences can be generated and regenerated accurately,so it is especially suitable for image encryption. Therefore, many image encryption algorithms have been put forward using chaotic systems. In 1998, the American scholar Fridrich put forward the classical substitution-diffusion architecture for image encryption . This structure subsequently has drawn world-wide concern, and nowadays, most of the image encryption schemes based on chaos adopt this structure and achieve satisfactory encryption effects, such as pixel-level scrambling approaches ,enhanced diffusion

schemes , improved hyper-chaotic sequences , linear hyperbolic chaotic system , and bit-level confusion methods . However, only using a low dimensional chaotic system to encrypt images cannot guarantee enough security. Some works on cryptanalysis show that many chaos-based encryption schemes were insecure, and the main reason is that the encryption key has nothing to do with the plaintext. For example, an image encryption algorithm with only one round diffusion operation is proposed in . The algorithm has the advantages of easy implementation,low complexity and high sensitivity to ciphertext and plaintext, butDiab et al. cryptanalyze this algorithm and broke the algorithm with only one chosen plaintext.Akhavan et al. cryptanalyze an image encryption algorithm based on DNA encoding and the curve cryptography and found that the algorithm cannot resist chosen plaintext attacks. Using a skew-tent chaotic map, Zhang proposed a novel image encryption method, which adopted a ciphertext feedback mechanism to resist chosen plaintext attacks, but Zhu et al. cracked the algorithm by applying a chosen plaintext combined with chosen ciphertext attack. Various plaintext-related key stream generation mechanisms have been proposed to improve the ability to resist chosen plaintext attacks . In most of these algorithms, the SHA-256 hash value of image is used as the external key of the encryption system, so that the encryption keys of different images are different, so as to achieve the effect of "one time pad".Taking the scheme in [28] as an example, firstly, the initial values and parameters of the two-dimensionalLogistic chaotic map are calculated from the SHA 256 hash of the original image and given values.Secondly, the initial values and system parameters of the chaotic system are updated by using theHamming distance of the original image. So the generated random sequence is related to the plaintext image. This encryption method has the advantages of high sensitivity to plaintext and strong attack against plaintext. However, the decryption end needs not only the initial key which is not related to the plaintext, but also the key related to the plaintext. Therefore, decrypting different ciphertext requires different plaintext-related keys, which essentially makes the system work in OTP fashion and greatly increases the complexity for applications.

Literature Survey

The research on the design of secure image encryption tends to focus on transferring images into chaotic maps. Chaos theory,which essentially emerged from mathematics and physics, deals with the behaviour of certain nonlinear dynamic systems that exhibit a phenomenon under certain conditions known as chaos which adopt the Shannon requirement on diffusion and confusion. Due to its attractive features such as its sensitivity to initial conditions and the random-like our spreading behaviour, chaotic maps are employed for various applications of data protection[9]. In the realm of 2D data, Shih outlines the following method, called Toral automorphism map, in order to spread the neighbouring pixels into largely dispersed locations. The Transfor-mation is represented through the following formula: the new location coordinates. We refer to the determinant here as 'det'. Applying Eq.(1)to the sample image Lena, after exactly 17 iterations, termed as the stable orbit, the chaotic map converged into the original image. This discrete-time dynamical system (DTDS) is also the basic framework used. Regarding this method, it is important to note:

(A) Since the algorithm uses a determinant in its process, the input matrix can only be square. This constraint was high-lighted. A work around this problem might be in applying the algorithm on square blocks of a given image repetitively. However, that would generate noticeable peculiar periodic square patterns given the nature of the process and of course this is not an interesting fact as it con-flicts with the aim of generating chaotic maps.

(B) As far as the security systems are concerned, the conver-gence of the translated pixels into their initial locations,i.e., image exact reconstruction after some iteration, is also not an appealing factor. This is an observed phenomenon in a variety of chaotic based algorithms. Given one of the iterations is used, if an attacker gains knowledge of the algo-rithm and obtains the parameter "l", which is actually not difficult to crack

using a brute force attack, he/she will be able to invest some time to add more iterations that will reveal the original image.

Starting from different initial conditions, each chaotic map in the CMLs, after a small number of iterations, yields a different value from the initial conditions, and hence the image becomes indistinguishable because of an exponential divergence of chaotic trajectories. They introduced seven steps for encrypting images and seven steps for decryption. Moreover, four parameters were used of which two were regulated. Their settings can have a tremendous effect on the chaotic map quality. Therefore, the recei-ver must know the decryption algorithm and the parameters which act as secret keys.

The algorithm is well formulated and adequately presented; it yields good results for RGB images as proclaimed by the authors.It was noticed that they used a rounding operator which was ap-plied recursively along the different iterations. The major con-cern would be in recovering the exact intensity values of the input image as the recovered image shown in their work might be just an approximation because of the aforementioned opera-tor. This is important, especially in the application of steganogra-phy where the objective is to recover the exact embedded filerather than its approximation. The raised point was remarked independently in Ref.[18]where they stated that a sensitive generator, i.e., a generator with a rounding operator, can produce two different binary sequences (after some iterations) for the same initial values and parameters if generated on two different machines which round off fractions after unmatched decimal places. However, a desired algorithm must be efficient, repeat-able and portable (i.e., works in the same way in different soft-ware/hardware environments). As a result, such a chaotic encryption system is not invertible under.

This proposal exploits the strength of a 1D hash algorithm,namely SHA-2 and extends it to handle 2D data such as images “Secure one-way hash functions are recurring tools in cryptosys-tems just like the symmetric block ciphers. They are highly flexibleprimitives that can be used to obtain privacy, integrity and authen-ticity”.

SHA-2 hash standard underlies four secure hash algorithmsSHA-224, SHA-256, SHA-384, and SHA-512. These algorithms are the result of a continuous development of

SHA-1. SHA algo-rithms are used to provide a condensed fixed length representa-tion known as message digest of an input message. The length of the unique digest ranges from 160 to 512 bits depending on the used algorithm[35]. The security of SHA-256, SHA-384, andSHA-512 matches the security of AES with complexity of the bestattack.

Challenges Faced

Despite being computationally demanding, cryptographic techniques have facilitated solutions to a variety of security issues in image and video processing. Even though the different applications come with different challenges, it is clear that the signal processing community will have to face three main challenges in future work. First, the utility and limitations of cryptography are not very well known to the community, which hampers the widespread consideration of cryptographic solutions for security problems in image and video processing. Second, cryptographic operations are often computationally expensive. Efficient usage of cryptographic protocols is therefore imperative. And third, certain cryptographic techniques cause ciphertext expansion of two orders of magnitude, such as public-key encryption of image pixels. Efficient data packing strategies and operations are then needed.

Problem Definition

To provide security to Image data, we will implement encryption techniques and provide security .The user can provide a password that would in turn be used to generate a key. The key is generated using a hash algorithm. The key can be used to further encrypt the Image file. The encrypted file can then be shared among users, however without the key or password along with the script the user cannot decrypt the content of the Image file.

Objectives

The objective of any image encryption method is to obtain a top quality hidden image in order to keep information secret.In many cases, security and privacy risks may impede the adoption of new image and video processing services. For this reason, the use of cryptographic techniques in image and video processing applications is becoming increasingly common. The cryptographic techniques used in these applications can be classified along two dimensions. The first dimension indicates whether the attacker model involves an outsider or one of the parties involved in the processing. For instance, in paid-for video services, the content must be protected against outsiders (non paying viewers) with a cryptographic end-to-end access control solution. But at the same time the content must be protected against the paying customer to avoid them from redistributing the content illegally. Similarly, in biometric systems the threat exists that sensitive biometric information (fingerprints, faces) are obtained by outsiders, or that the party storing and processing the biometric templates misuses them for its own purposes.

The second dimension of classification is the kind of and the way whereby cryptographic techniques are applied. In some image and video processing applications, common cryptographic primitives are used as “add-ons”, such that the original (non-secured) image and video processing operations are hardly or not at all affected. In other applications the cryptographic primitives are deeply embedded into the processing algorithms (e.g. the processing algorithms may work directly on the encrypted data), and completely new solutions may emerge.

Methodology

The images used will have their bytes extracted; the bytes values are transposed as MR values and further encrypted to obtain cipher text. The ciphering of the images for this

work will be done only by using the byte values of the images. In this method, the RGB values of the image are not changed. Also there is no need for RGB expansion at the end of the encryption and decryption process. The numerical values of the MR are displaced from their respective positions and encrypted in order to obtain the cipher text. Therefore there is no change in the total size of the image during the encryption and decryption process. The characteristic of the image remains unchanged during the encryption process. The processing model of image encryption and decryption process

A cryptographic hash (sometimes called 'digest') is a kind of 'signature' for a text or a data file. SHA-512 generates an almost-unique 256-bit (32-byte) signature for a text.

A hash is not 'encryption' – it cannot be decrypted back to the original text (it is a 'one-way' cryptographic function, and is a fixed size for any size of source text). This makes it suitable when it is appropriate to compare 'hashed' versions of texts, as opposed to decrypting the text to obtain the original version.

Such applications include hash tables, integrity verification, challenge handshake authentication, digital signatures, etc.

- '*challenge handshake authentication*' (or 'challenge hash authentication') avoids transmitting passwords in 'clear' – a client can send the hash of a password over the internet for validation by a server without risk of the original password being intercepted
- *anti-tamper* – link a hash of a message to the original, and the recipient can re-hash the message and compare it to the supplied hash: if they match, the message is unchanged; this can also be used to confirm no data-loss in transmission
- *Digital signatures* are rather more involved, but in essence, you can sign the hash of a document by encrypting it with your private key, producing a digital signature for the document. Anyone else can then check that you authenticated the text by decrypting the signature with your public key to obtain the original hash again, and comparing it with their hash of the text.

Here we have implemented the the encryption algorithm using python:

Help • encrypt.py - Image-Locker-master - Visual Studio Code

encrypt.py •

encrypt.py > ...

```python
from image_encryptor import core
from image_encryptor import DEBUG

from argparse import ArgumentParser
from getpass import getpass
import sys

def getArguments():
    parser = ArgumentParser(description="List of commands for Image Encryptor")
    parser.add_argument("-d", "--dir", type=str,
                        help="The directory path for group of images.")
    parser.add_argument("-f", "--file", type=str,
                        help="The path of the file to encrpt.")

    args = parser.parse_args()
    return parser, args


def main():
    parser, args = getArguments()

    if DEBUG:
        print(args)

    if(len(sys.argv) == 1):
        parser.print_help()
        sys.exit(0)

    password = getpass("Enter the password for the operation\nPassword:  ")
    if args.dir:
        core.encryptDirectory(args.dir, password)
    elif args.file:
        core.encryptSingleImage(args.file, password)
    else:
        print("Use -h flag to get help on how to use.")
if __name__ == "__main__":
    main()
```

Here is our sample image coffee_icon which we will encrypt:

Here is our decrypt algorithm which is used to decrypt the image with the password which we have entered while executing encryption.py:

encrypt.py ● decrypt.py ×

decrypt.py > ...

```python
from image_encryptor import core
from image_encryptor import DEBUG

from argparse import ArgumentParser
from getpass import getpass
import sys


def getArguments():
    parser = ArgumentParser(description="List of commands for Image Decryptor")
    parser.add_argument("-d", "--dir", type=str,
                        help="The directory path for group of images.")
    parser.add_argument("-f", "--file", type=str,
                        help="The path of the file to encrpt.")

    args = parser.parse_args()
    return parser, args


def main():
    parser, args = getArguments()

    if(len(sys.argv) == 1):
        parser.print_help()
        sys.exit(0)

    password = getpass("Enter the password for the operation\nPassword:  ")
    if args.dir:
        core.decrptDirectory(args.dir, password)
    elif args.file:
        core.decryptSingleImage(args.file, password)
    else:
        print("Use -h flag to get help on how to use.")


if __name__ == "__main__":
    main()
```

Results:

Executing encryption.py:

encrypt.py

```
from image_encryptor import core
from image_encryptor import DEBUG

from argparse import ArgumentParser
from getpass import getpass
import sys

def getArguments():
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE powershell

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\danis\Desktop\Image-Locker-master> python encrypt.py -f coffee_icon.png
Enter the password for the operation
Password:
Output file at location :  coffee_icon.png.txt
PS C:\Users\danis\Desktop\Image-Locker-master>
```

When prompted for password, we enter the password which creates the encrypted txt file as shown:

encrypt.py coffee_icon.png.txt

coffee_icon.png.txt

```
gAAAAABgzcea6VfHXxjUgna30nizd-wdykpekUV3QNyfGqF9CZauZvqthtRdwplJPr9jrAZQENeti-ZFYwk3rGtaEd5JJE4SBIc2gAZ65ZxjD5PkEF3TSNRgqSNFsgwT
yBRugGCmz9QRB9GDtX9uzqbMyp9bCLw6TgOA1IAxe9fhDiQtajyMePW_IEcTri9rf-4OMDobGrzvUdEjQncyuoVN0ltXOe1bPaEYZ4I4iI2CfyfDeiKS21LEejuJZ2D3
RnvsbX7pXNkv5Tq55lJspTOg5aHl-snC7vIoB-dAnwTLiJHRxKMGCu_A0DJSC_01jVNYucy1wgUJe_dM8KLfkR113BtfA1GpXGfmino8DuH96JZ4zdBSlpz0NZnwPQAf
ZCce9gGekVBjzzO6SLyj01UWD7ua6uRfxg-j94H9UYoHMh6VgkA8AuTqNSlyMYwSdqFWpKYLKzwUUdyk8y-tCKDa86_VVohDkIbvBaMjNhErye7vduVnzVqicdjMcLrL
pHUCfNt22hVQlPeyg5PGF0N0H_JTwqg-xyCwV4lDjJfNJz6Re7aZN5d-X4LBmi4RCmV_u5LmsKh1JwJmBH95um5YBXV8WYrydYnj6MvbxiQrPgftjHnXsmcXRPLxDRHn
Gpx5hnuZ7YG-jUk2gvXL7mIhsNF5KHDA4BGg_gmZMTf2Yn2EKiuyerCy7BwupyJZ9L1sznyljihjosTpA1HFka2vXMR1t8R2uuzlIxEpfwZuUvoPheSO_9_tTuhaGDxG
G161Zsf6AKYDTOuqmp65inC-8XeVB1OwBxelPQXQECZ8s52vV-5g2x8-X6bAzYQq4UTDziJILjuABE3y7cj4KNXU69F0h4nB1ekRIp_y4aUN7nPQb3H1eRKEDDNkE6da
2nYqi4Mas-7O-96pThv4syvvS2aHhyvD2-M5QNR7TRMU5M0_ZPF6j6I3y85DGolLNGriFD6l0TMa4oLd6cXWLRyJGiN3fMzeNuM2nJoXoOLMjN07LdAoTKhfCMZn4spg
KykSB7sDqNMukWVh_SAOBlRd4-rBeWO20bEwj23nLqbHFIl5Oh_kePDT7iwx89wmmisyHvgegD-dk44PuUyaOTox4AbzZEcUwIRZo2Uk4VDABGnRiwmgOwSTCKQ4Nf4N
73cNsbFS78SIsMDoyOrfk4fPvtWN0D5grkNXcrCy8JAOKoBD6gVBk2HMN3Nu2XGe_XYtlIE46f88nxq-LJydRqyflRGD-sRwsOv2x-XCCAQTgrHxqMxO8P_YV3WByEnx
tQpcVqi0mmdnNFZ8MPsnlxRDXDR3F9rmrD21NuTXZ7tB786QV1o7OjzTo5DIhmo7KWkvWxE_95RYiAvA_t5l5kXghsxm4_5ygrL12ceAFbKK6KZAuIxyoUaAQAK5mLRu
FyVAI0w6HDzdzU-Zjp3fki0MT9nbBXVeM0eu42CT3qd6iUJnotianc1Y14kU_9a3hvFfukfbzNLfBCHSS9pqq7V--kXsEEgWuFdsK__JT3w2nHzPvjy8siDkbBh9ldKQ
duuLHmiB-r1E3LLtd59JzZt6IYJjTQj7sRyAE_mWnRC5c5JHiBls-B2aB_fyC0nHS-JdZ4bG9XqOg6w3KE0Y6FG-FwSNos4PTJRr_GgxNmWCM5A8Inv4sgEYaSU9orPk
zwBuVxtb9iV6xoWUwqWjskk5ocdxInv3WbDNf7uCHvnejHGFlIU3NZjR3YBZHl_PmrxVaElZGWqGYj5b_Wqm1k0yLZwMRG36Km-I7Wi_Ml0R5O6dD2eMXijIgGwbZzzR
RpJqo4NPEka0yf_3y34LybjOvJMab1t5SnywmUuc1PCh_Rp49kGANX9YyrocPnRhydXtE8Sugq77zDmxMejuwmicqFkpo5Okhl_nmsYCtZPdYfe3AEj4sCVkT_HwFxta
RAVhVJWwzjavE2YKUHz2TQuYG-IS9M2oKgS7_aXSWwwJVbf31OWgeIu-mYwB1ZcRMnnhmMchToibtdx3a3q6Alv3Ha70bjHIS6STK2L_gRAOKWfPeNaYmOTdL1Fz6I1G
y09DlKjopUVLXZoZf8u552krYfNdMQ2igyfXBJNdELGw3E8lPbMmLZvIBqSaE-XnhAbP53L6yqkWohVdSvgNFc2HXKlMsZj7XHZRJc1oXp_knsneVRNCJkqgrd8SThpD
EvGAD6Hf0aoUipeQvMjoYHatx1C3xyq4mbByBOuCbsCFscQWQRt_mllSJ65wiufQH2nYkEKwcVJuFkCpdACwctDPMUfLPmsY6kjIIH3nsMam2gmnH6dew98nDX8xSmzX
h-WO-HB8EkR-7zvCJUARRdZluAkytRDtx_RYR9zSUxUbk5tydH3fXAwWg6U_lYvv-0_BzXSIEX2YM_1wKhLbMIhESeITlZgpUocxOjH3n7MuRgIbRTRT8-_KMJuPHNr0
m0mKd-TJ9HiVdutTdyKV3sF5_53TOZzYj_GUF8W7m_B9xsBGuUjHoF8O-TkB6EPKyfuJglgjNis_isVEBrSP6_1JgnaXqohjHp6GpqTaRRLPMBC2k3IHLfuF1Jdn2qld
JPr4QUCCS9B_ly7PVblkfyvwWKLacnubMuUpGbNCTslpk_TIUUhASYNz9vwiVzQwQD2cgdYthIl9KhsftuVoUz4MoOWXriEz0TDoIBn0jyMLng64dEmb7HCUpcFrZCld
ieTTNtBDKmIogI8icsQburLMqrdPeRayOasOASY3XkdEoiDFcBqaNuHKw9QHi-WoPPcgy7O2bkjJy1zpx45c4LH7DoX8ihtwOlpjebMmQ2LAdRHSxbcBGpdWpemeqzmX
UpVrhCiekT67_wgYZX4xvXlkANYZh1KZWd5UDaIODpaPntD31KvdxH-rcaEeZaA4TrCqy_x5drumep1zVi1fjA4skLF9mfXIwXt_0QaspnoIGJSpmEqkEelUbIRqzE7
qAV7gx7GvnkrQixap36Y_58cjGW6ArR8cGs85onm7YuECs_nKS3sjIVEoeGsjUBW5SYsgvhNqBQLtUyXNhisnZEEkXk5Us1tgt0uSUk_z-pPLB6pbkF8tCSv_UTuY1ia
kITyK4oXi4xfA0WmLkazaZNRwGEzwvIoPKpHi8B5XV4DLSYxCsMqqM7tp8t8cCzkNR76F24wGa20_QX7hFNU0eS6CJ4xSId57mWwtL5XC5Sav3duBryd4WeK5gswukUV
h4XLhP0JiC7uwaf3Hw3cIm-twMLJlmSCmg8FqljcLgCa23v1nXwQm2ef8xrIpsXAUPoLQ3A7vovrn2Hq6g4TzWnjfgdmvE7gs2GW8Zl0cBZMc2tkTr5zqKJKD9DjNNkY
9AfA_3YGFskx4ylDGdyQvOraGjEIK18MiEb2IPDnLqulE7lxvVSqc3KLTYl5dfwHoOTM6XvsZRUwErUHBEFzoHo2qSRfyGrEedy8mFMuutEUD-n8y_Ol0xgf9SrXk_tK
09ltsmmkH6xabKutsk6_UCbNXOCluZK3Dd_5j11amCDmpawHhLZ_BrFkkWAx5DAYFAabfxya8a0vEvhtB3O_sP8qShT663S5kNF4etaG7jKpAvkbH6ren0AiVltymgAR
QvFlAeSVFYXFfZ-IDm-49kggcr50KzCPFL3X7_Guj_mUOJFoEC0HFLF-eskIVNwqMNUPu41wDlDDE-DXAcVZutgmpuKqGZ8P0Gz6NfcXAKRNxYF5hlPVvcg0Uk6F3Fa0
_2fjlU_2ktzJaST_V4w61EGI9R6gA4eEjyp4p8GxDhZ-KoVscAiIhwwQ1_CXxNlOJ_zLc4Z2GRUKkD0mZdMRQ5qdFmeqjArdXyBK1l2fZjmQTE89EHD7xfigQM00h1EV
NbUKhzEtspKex35G1R2tpGNTBp07cuN0wW1N3nrMTH0tFPhc5Hso8uBLp2PPtyB6P5qd--YTrmvo2VMLJo9VE-jZ3l-hLI3bpOLTGTevcUHzmJowRrTPuvvZol0k_4WS
m6iepBiso8_VmcsrGF9f2QRKHtrPSN5O6cSTwqEWE8vGJVFDLVNz7almjAYsFDDV_hA_naeMPOzORl-rkzBfVk-1ek-lzlVQ1t4es-xAGvYZg7zZYlwod7T5Yn-p8lXW
um0n4sskArhNmxYzCd3gmI-MwCEK3q6g58BZqddTubUcYbCQDabiXONu4rCEIsoxnW-iv2XD2E-RyDyqNLg2Yxha5WbciFsczH8kzWy-H5fk8IDpOLuklkf_G5bweFc0
1NAAdzW8IHLTjN41JedtVyzVeRa5A1JdfSlPjmbFnR5fIGWh9cAlktuXt_E-VflKdiGSYrKesryBeY79vMoZSenpbXLX8l0jkcN8ArFVOLag1d6N8P1Goa7aPDgd50iW
rJ7O-ZPJXd8j9G-b0zX6-iOKK3sVepVJF09YH4MgrmgMetAZ2qcBxLx15gNvdx48K-vHzdgH3uqw2azpZeHpmmxUOJuyVlHU18YnRVGXxAG2MucqZPECVzYkOsBQaWuY
PiYW5LVGvIwvG05gEVEkRvYGODHtLdp5nc_o7YBnrjzSHtIIzMHA1V-5rPGpxp3HoknsaqmoDyQg-7X-Py_QVDsfsRjP64D6gW2JrVABCSME9OHemSEGo0uolDWa4v-P
I_s9AQMw_Sx2rkT9Zut3dIBy-OAdlTpNrusdTnVTLOR-BvodYo8tTKz_J1nVyW1kIf_QJed81BbU5MX8A7TdolWTrCmz2qx8_CX_pFBl1GitUusXCYMaIpxfrMdbj4nn
SYIa87xiQwtoq7scCreOnk0kk6e38RhLgShtZMveN1JiRnvHRwMxi0G42wff1oNKsN_-OiiAINagE4EA1P5WQiTz-S2g-fxPPnni-JxR6UtaE_61dSOLxaX_VvprSAw1
```

Now we run decrypt.py:

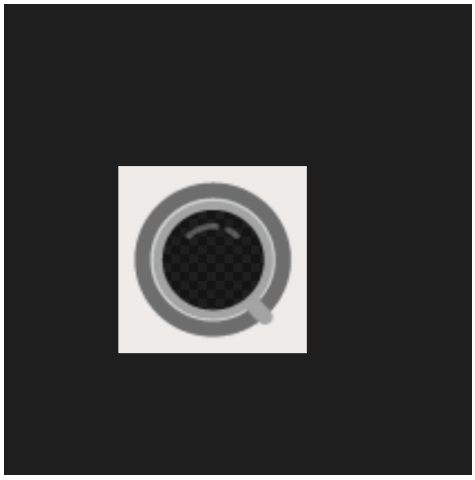
```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE                    powershell

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\danis\Desktop\Image-Locker-master> python decrypt.py -f coffee_icon.png.txt
Enter the password for the operation
Password:
Output file at location :  coffee_icon.png
PS C:\Users\danis\Desktop\Image-Locker-master>
```

When asked for password, we enter the password which decrypts the image and stores it it in our project folder:



Decrypted image.

Conclusion

A image compression and encryption based algorithm for high speed and secure transmission of digital image while passing through the web. So the proposed work is highly achieved as needed for speed and security .This proposed work is to work with color images without losing color and detailed information in image ,this work is very useful to the modern digital world. Experimental results ,that the proposed scheme is effective, secure and robust to compress, encrypt and decompress-decrypt images.

Future Work

In this project, we have implemented encryption and decryption of an image using SHA 512. Further, we are planning on creating a fully functioning website with a user-friendly UI, in which users can encrypt and decrypt whatever image they want with our website using this code in the backend.

References

[1] B. Preneel, V. Rijmen and A. Bosselears, Recent developments in the design of conventional cryptographic algorithm, LNCS, Springer-Verlag, Vol. 1528, pp. 105-130, 1998.

[2] T. Habutsu, Y. Nishio, I. Sasase and S. Mori, A secret key cryptosystem by iterating a chaotic map, Eurocrypt'91, Brighton, U.K., pp. 127-140, Apr. 1991.

[3] G. Jakimoski and L. Kocarev, Chaos and cryptography: Block Encryption ciphers based on chaotic maps, IEEE Trans. Circuits Sys. I, Fundam.

Theory Appl., vol. 48, no. 2, pp. 163-169, Feb. 2001.

[4] T. Stojanovski and L. Kocarev, Chaos-based random number generators-PartI: Analysis, IEEE Trans. Circuits Sys. I, Fundam. Theory Appl, vol. 48, no. 3, pp. 281-288, Mar. 2001.

[5] Y. Wang, X. Liao, D. Xiao and K. W. Wong, "One-way hash function construction based on 2D coupled map lattices," Information Sciences, vol. 178, no. 5, pp. 1391-1406, Mar. 2008

[6] L. S. Chen and G. X. Zheng, Multimedia Security Handbook, CRC Press, 2005, p.133.

[7] Fridrich, Symmetric ciphers based on two dimensional chaotic maps,Int. J. Bifurcat Chaos, vol. 8, pp. 1259-84, 1998

[8] J. Cheng and J. I. Guo, A New Chaotic Key-Based Design for Image Encryption and Decryption, Proceedings IEEE International Conference on Circuits and Systems, vol. 4, pp. 49-52, May. 2000.

[9] A.J.Menezes ,P.C.Van Oorschot, and S.Vanstone , "Handbook of Applied cryptography", CRC Press, Boca Raton,Florida, USA,1997

[10] Gopinath Ganapathy, and K.Mani , " Add-On Security Model for public key Cryptosystem Based on Magic Square Implementation", ISBN 978-988-17012-6-8, Proceedings of the world congress on Engineering and Computer Science 2009 Vol I, San Francisco, USA .