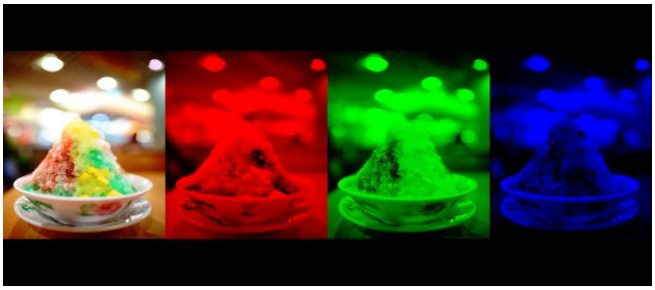
Image Encryption Using AES

Overview

Image Encryption is very important as digital images play an important role in multimedia technology. Image encryption maintains the security and privacy of the user by preventing unauthorised access to the image contents.

A digital image is a representation of a real image as a set of numbers, each of those numbers is associated with a small area called pixel, a pixel is composed of one or several cannels depending on the type of the image. For example a black and white image has a single channel which represents the intensity of the light. RGB images are usually used in computer scanners and monitors and the consist of three channels red green and blue respectively, CMYK (cyan, magenta, yellow, black) is a four channels image used in colour prints.



From The left : the original image, red channel, green channel and blue channel

There are also non optical images such as ultrasound or seismometer in which the intensity of the sound or earth vibrations is recorded as an image.

The colour depth is characterised by the bits and number of channels for each pixel, for example an RGB image has 8 bits per channel per pixel, that would result is 24 bits per pixel and 16777216 possible colours. Image resolution is characterised by its PPI (pixels per inch), higher resolution results in more pixel information and create a higher quality image.

Medical ultrasonic image

Images tends to generate large files and are usually compressed to make the files smaller, compression algorithms may take advantage of the fact that many nearby pixels in the image have similar colour, image compression can be lossy or lossless.

Image encryption

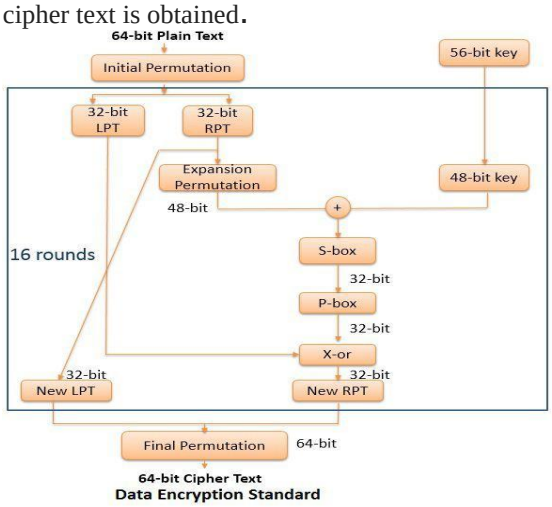
With the increase in demand for image encryption, many scholars have proposed many different encryption techniques.

Images are different from text. Although we may use the traditional cryptosystems to encrypt images directly, it is not a good idea for two reasons. One is that the image size is almost always much greater than that of text. Therefore, the traditional cryptosystems need much time to directly encrypt the image data. The other problem is that the decrypted text must be equal to the original text. However, this requirement is not necessary for image data. Due to the characteristic of human perception, a decrypted image containing small distortion is usually acceptable.

Algoithms

Data Encryption Standard (DES) is a **symmetric key block cipher** that was adopted by national institute of standard and technology in the year 1977, DES is based on the **Feistel structure** where the plaintext is divided into two halves; DES takes input as 64 bit plain text and 56 bit key to produce 64 bit cipher text

In the figure below you can see the encryption of plaintext using DES. Initially, the 64-bit plaintext undergoes initial permutation which rearranges the bits to get 64-bit permuted input. Now this 64 bit permuted input is divided into two halves i.e. 32-bit left portion and 32-bit right portion. Both this portion undergoes sixteen rounds where each round follows the same functions. After completion of sixteen rounds, final permutation is done, and the 64-bit cipher text is obtained.



Each round contains following functions:

- **Expansion Permutation**: Here the 32-bit right portion is expanded to form 48-bit right portion.

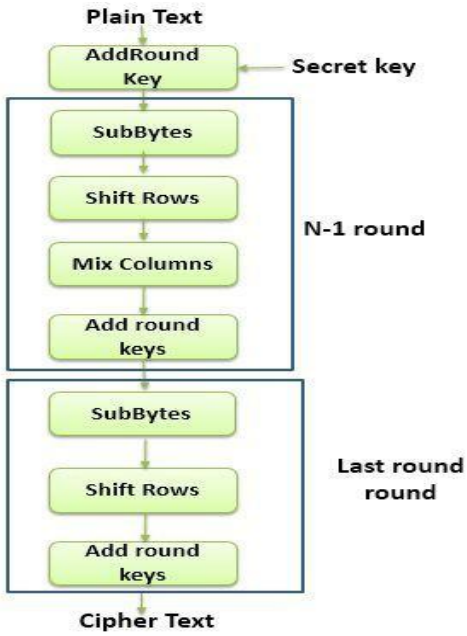
- **XOR**: The 48-bit right portion is XOR with 48-bit sub key obtained from the 56-bit key, which results in the 48-bit output.

- **S-box**: The 48-bit output obtained by XOR step is reduced to 32-bit again.

- **P-box**: Here the 32-bit result obtained from S-box is again permuted, which result in 32-bit permuted output

By the early 2000s **DES was considered no longer secure** because it cloud be broken with the state of the art computer in 22 hours due to the short key, however the fix for this problem is not only to increase the length of the key but to find and algorithm that can run with the new key in a reasonable time.

Advanced Encryption Standard (AES) is also **a symmetric key block cipher**. AES was published in 2001 by the National Institute of Standards and Technology. AES was introduced to **replace** DES as DES uses very small cipher key and the algorithm was quite slower.



Advanced Encryption Standard

AES may use 128-bits, 192-bits or 256-bits key length, AES algorithm takes 128-bit plaintext and 128-bit secret key which together forms a 128-bit block which is depicted as 4X 4 square matrix. This 4 X 4 square matrix undergoes an initial transformation. This step is followed by the 10 rounds. Among which 9 round contain following stages:

- **Sub bytes**: It uses S-box by which it performs byte by byte substitution of the entire block (matrix).

- **Shift Rows**: Rows of the matrix are shifted.

- **Mix Columns**: Columns are of the matrix are shuffled from right to left.

- **Add round keys**: Here**,** the XOR of the current block and the expanded key is performed.

- And the last 10th round involves Sub bytes, Shift Rows, and Add round keys stages only and provides 16 bytes (128-bit) cipher text

S. No	Algorithm	Key Size	File Size	Average Encryption Time (ms)	Average Decryption Time (ms)
1	AES	256	2048	302	299
2	DES	56	2048	305	316
3	RC4	64	2048	319	292
4	RSA	1024	2048	497	493
5	MD5	128	2048	475	461
6	NTRU	80	2048	331	305

The above table shows that in comparison to other encryption techniques, AES enjoys relatively higher performance.

Scope of this work

To implement the AES algorithm for image encryption/ decryption .

Methodology

The algorithm described in the previous section was implemented using python 3.8, the substation and the reverse substation LUTs are Numpy arrays, and Images are handled using Python Image Library PIL, the performance of the code is enhanced using the Numba JIT, after Image is loaded image is splatted into RGB channels each of those channels are encrypted separately, the 3 channels are then combined to form the final encrypted image. Key is always 128-bit if the user entered more than that the first 128-bits only will be considered

Tkinter :

The tkinter package (“Tk interface”) is the standard Python interface to the Tcl/Tk GUI toolkit. Both Tk and tkinter are available on most Unix platforms, including macOS, as well as on Windows systems.

The GUI of this project was designed using tkiner library, the purpose of the GUI is to provide simple interface with the user, the GUI goes through finite state machine sequence.

In case user enters a wrong path to the image or image file is corrupted, user is redirected gain to the

Numpy :

is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

Numby is used to curry out almost all the logic in the project, row shift , column mix and xor operation with the key is all done using numpy, when it comes to the substitution instate of going through elements one by one the operation was vectorized to minimize substation time.

Numba :

Numba translates Python functions to optimized machine code at runtime using the industry-standard LLVM compiler library. Numba-compiled numerical algorithms in Python can approach the speeds of C or FORTRAN

Results:

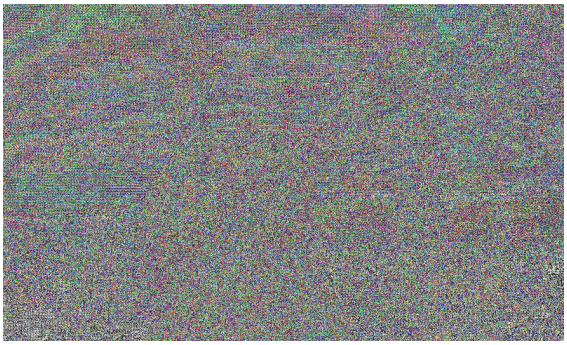
Image encryption was achieved with key length of 16 bytes (128 bits), images below shows some examples of encryption and decryption.

Encryption :

Input

Output

Decryption:



Input

Output

Significant increase in performance was observed when including the numba Just In Time compiler (JIT) , average execution time without numba is 16.4 seconds for encryption or decryption after numba execution time falls to 0.588 seconds , for RGB input of size 1242x375 pixels,about 2789%faster.