VAMOS Encryption and Decryption Algorithm

Vaishnavi Sawant
B.Tech. Student
Dept. of Computer Engineering
Dwarkadas J. Sanghvi College
of Engineering, Mumbai,
Maharashtra, India
vvaishsawant@gmail.com

Ahmed Solkar
B.Tech. Student
Dept. of Computer Engineering
Dwarkadas J. Sanghvi College
of Engineering, Mumbai,
Maharashtra, India
ahmedsolkar@gmail.com

Dr. Ramchandra Mangrulkar
Professor
Dept. of Computer Engineering
Dwarkadas J. Sanghvi College
of Engineering, Mumbai,
Maharashtra, India
ramchandra.mangrulkar@djsce.ac.in

Abstract—**Internet (commonly called as Net) is a global network of billions of interconnected systems and other electronic devices. Internet makes it possible to transfer any form of data and facilitates data communication. This is done simply by connecting computer to the Internet. As Internet is public the data being transferred is vulnerable to a lot of data theft attacks such as packet sniffing, packet spoofing etc., it is essential to protect the private data from such kind of attacks. When the network is not secure hackers can exploit the connection, giving them access to sensitive information and can possibly tamper the data.**

In today's world data is very crucial for businesses and individuals, hence it must be protected from all kinds of violations. Data must be secured from unauthorized access and manipulation. It is challenging to physically secure the network so data encryption can provide significant amount of security. Encryption is a technique through which the data is transformed in some incomprehensible form which cannot be interpreted by any attacker. After successful transmission of data, the receiver can restore the data received into its original form by applying appropriate decryption technique.

In this paper, we have presented a symmetric data encryption algorithm developed in Python to encrypt images, making the transmission secure.

Index Terms—**Cryptography, Encryption, Decryption, Internet, Python**

I. Introduction

The technique through which information is transmitted in a secured way and communication is conducted via some programs and procedures so that only the intended entity understands the actual data that is shared is known as Cryptography. This process thwarts unauthorized accessibility for the information. During data transmission across the network, Cryptography aims to fulfil four different objectives:

- **Confidentiality:** This ensures that only the intended receiver is able to decrypt the received message and access the original data.
- **Non-repudiation:** Non-repudiation ensures that the sender of the message cannot deny their participation in the communication by sending or creating the message.
- **Integrity:** Integrity focuses on ensuring that the information is not modified or compromised in any way while in storage or transit.
- **Authenticity:** Authenticity ensures that the sender and receiver can verify each other's identities and check whether the information came from legitimate source.

Encryption is a technique that is used to conceal the data that is to be transmitted. This is done by applying different algorithms which consist of numerous mathematical and logical computations. The resultant data is called as 'Cipher Text'. The original data commonly known as 'Plain Text' along with some 'Secret Key' is supplied to an algorithm which does the job of encrypting the data. At the receiver's side this 'Cipher Text' is reverted back to its original form by applying the appropriate decryption algorithm and the 'Secret Key'. Encryption has two types:

Symmetric Encryption: When only one 'Secret Key' is used for encrypting and decrypting the data it is known as Symmetric Encryption. The same Key is distributed among all the communicating parties in the network. For e.g. DES Algorithm, AES algorithm etc.

Asymmetric Encryption: In this two keys are used for the process of encryption and decryption. For Encryption the algorithm uses the 'Public Key' of the intended receiver which is shared publicly, while for Decryption it uses the 'Private Key' of the receiver which is kept private. For e.g RSA Algorithm, Elliptical Curve Cryptography etc.

II. Methodology used

A. Overview

The presented algorithm is based on symmetric encryption technique and incorporates the following computations: Circular Left Shift: Shifts the bits of the first operand by number of bits specified by the right operand such that the bits which fall off at one end are appended to the other.

Ex-OR: It is a logical operation also known as exclusive or which takes two Boolean operands and returns true if, and only if, the operands are different.

Mix Columns: It is matrix multiplication similar to AES . It ha s a predefined 'Multiplication Matrix' which is multiplied with Input 4x4 matrix. The results of these multiplications are XORed together to produce only 4 result bytes for the next state. Therefore it contains 4 bytes input, 16 multiplications 12 XORs and 4 bytes output. This is a complex computation and hence it is simplified by performing it over a Galois Field. Two tables named ‘E’ and ‘L’ are made for this and the result of the multiplication is simply the result of a lookup of the L table, followed by the addition of the results, followed by a lookup to the E table.

Shift Rows: The entire row of the input matrix is relocated to some other index in the same matrix based on some criteria.

S-BOX: Substitution Box is a 16x16 invertible matrix similar to AES encryption technique. It transforms the 8 but input data into 8 bit secret data using a precomputed Look Up table. This table provides confusion in the Cipher Text by substituting the some in place of original values. the design of S-BOX is used to protect the message and also achieve a high throughput , high energy efficiency and occupy less area.

B. Flowchart

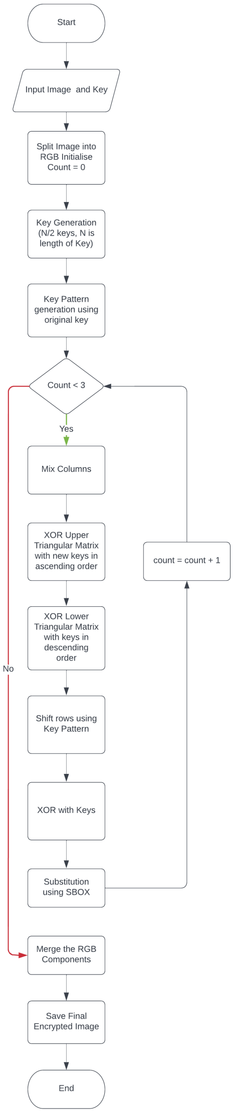

Start
Input Image and Key
Split Image into RGB Initialise Count = 0
Key Generation (N/2 keys, N is length of Key)
Key Pattern generation using original key
Count < 3
Yes
No
Mix Columns
XOR Upper Triangular Matrix with new keys in ascending order
XOR Lower Triangular Matrix with keys in descending order
Shift rows using Key Pattern
XOR with Keys
Substitution using SBOX
count = count + 1
Merge the RGB Components
Save Final Encrypted Image
End

Fig. 1: Encryption Flowchart.

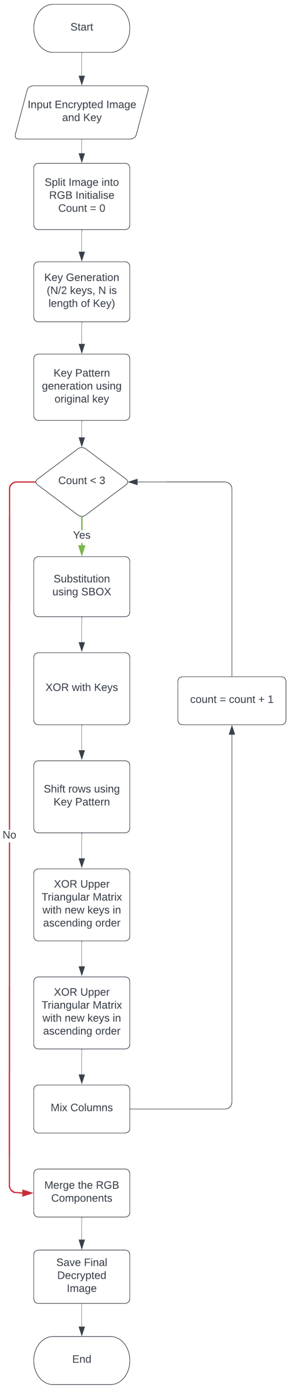

Start
Input Encrypted Image and Key
Split Image into RGB Initialise Count = 0
Key Generation (N/2 keys, N is length of Key)
Key Pattern generation using original key
Count < 3
Yes
No
Substitution using SBOX
XOR with Keys
count = count + 1
Shift rows using Key Pattern
XOR Upper Triangular Matrix with new keys in ascending order
XOR Upper Triangular Matrix with new keys in ascending order
Mix Columns
Merge the RGB Components
Save Final Decrypted Image
End

Fig. 2: Decryption Flowchart.

C. Algorithm

Step 1: Take Image to be transmitted and Key used for Encryption from the User.
Step 2: Split the Input Image into RGB matrices.
Step 3: Key Generation
3.1: Left Shift by 2
3.2: Swap the first half with the second half
3.3: Ex-OR with opposite index elements.
3.4: Apply P8 combination to generate final n/2 keys.
Step 4: Generate Key Pattern.
Step 5: Perform Mix Columns for every 4x4 matrix.
Step 6: Ex-OR Upper Triangular Matrix of the obtained result with the keys in ascending order.
Step 7: Ex-OR Lower Triangular Matrix of the obtained result with the keys in descending order.
Step 8: Shift rows of the resultant matrix according to the pattern generated in Step 4.
Step 9: Ex-OR each element with generated keys.
Step 10: Perform S-BOX substitution.
Step 11: Merge the RGB matrices and save the encrypted image.

III. Working

The presented algorithm is based on Symmetric Key Cryptography which means that it uses only one key for encryption as well as for decryption, which is taken as input from the user along with the image to be encrypted. The algorithm begins with splitting the image in 3 matrices each containing the Red, Green and Blue values of each pixel of the image respectively. All the computations included in the algorithm are applied to each of these matrices separately and the resultant matrices are merged to form the final encrypted image. Same procedure is followed while Decryption.

After the image is split the next step is Key Generation. In this step, n/2 keys are derived from the original key by performing a series of computations. The characters of the input text Key are firstly converted into their equivalent binary value and Left Shift by 2 operation is applied to all these values individually followed by swapping of their equal length sections. The resultant values are then XOR ed with their negative index values thus deriving the n/2 keys from length n Key. Each of these n/2 keys are permuted by application of p8 table and the final form of n/2 keys is achieved.

Along with this a Key Pattern based on the ASCII values of the characters of the input Key is extracted. This pattern is applied further while Shift Rows transformation.

Further to provide diffusion in the data image matrix we apply MixColumns transformation similar to AES. This is done for interbyte transformation that changes the bits inside a byte. This transformation changes the content of every byte by taking four bytes at a time and combining them to create new four bytes. It uses a polynomial function which takes four bytes of one input column and outputs four new bytes which replace the original ones.

For applying MixColumns to our input matrix we first split the original matrix into number of 4x4 matrices as MixColumns transformation works on 4x4 matrix. These 4x4 matrices are given as input to the MixColumns one by one. MixColumns transformation uses a predefined 'Multiplication Matrix' for its computation. This multiplication is performed one column at a time for each 4x4 input matrix. Eventually each value in the input column is multiplied with every value of the 'Multiplication Matrix' resulting in total 16 multiplications for a single column. The resultant values of the multiplication are XOR ed together to produce 4 bytes which replace the original column. Hence for each column there are 4 bytes input, 16 multiplications 12 XOR's giving a 4 bytes output. This multiplication is performed one row at a time for each value of the input column.

The Multiplication Matrix for Encryption is given as:

TABLE I: Multiplication Matrix

2	3	1	1
1	2	3	1
1	1	2	3
3	1	1	2

If the 4x4 input matrix is given as:

TABLE II: Input Matrix

B1	B5	B9	B13
B2	B6	B10	B14
B3	B7	B11	B15
B4	B8	B12	B16

Then the resultant value of B1 is calculated by multiplying the 4 values of the input column with the first row values of the 'Multiplication Matrix' and then every value is Ex-OR ed with the other.

B1=(B1*2) xor (B2*3) xor (B3*1) xor (B4*1)

Similarly the resultant value of B2 is calculated by multiplying the 4 values of the input column with the second row values of the 'Multiplication Matrix' and then every value is Ex-OR ed with the other.

B2=(B1*1) xor (B2*2) xor (B3*3) xor (B4*1)

However this computation is complex and takes up a lot of processing time hence an alternative known as 'Galois Field Multiplication' is implemented in the presented algorithm. In this technique lookup tables named 'L table' and 'E table' are defined for ease of multiplication. The result of multiplication is derived by referring the L table followed by the addition (regular mathematical addition) of these results and then referring the 'E table'. [1]

For example:
Consider Input = D4 BF 5D 30
Output (0) = (D4 *2) xor (BF * 3) xor (5D * 1) xor (30 *1)
= E (L (D4) + L (02)) xor E(L (BF) + L (03)) xor 5D xor 30
= E (41 + 19) xor E (9D + 01) xor 5D xor 30
= E (5A) xor E (9E) xor 5D xor 30
= B3 xor DA xor 5D xor 30
= 04

Following the application of MixColumns transformation on all the 4x4 matrices generated from the original matrix , all these resultant matrices are merged into a single matrix

After the application of MixColumns transformation we perform row- wise Ex-Or operation on the partially encrypted matrix which is divided into two parts (i.e. upper-triangular and lower-triangular). In upper triangular matrix the keys are Ex-OR ed with the values present in the row in an ascending order. While in lower triangular matrix the keys are Ex-OR ed with the values present in the row in a descending order.

The prior step is then followed by Shift Rows operation. In this the rows of resultant matrix are shuffled according to the Key Pattern generated previously.

After the shuffling, all elements of the matrix are Ex-OR ed with the keys generated in a sequential way such that each element is Ex-OR ed with one particular key.

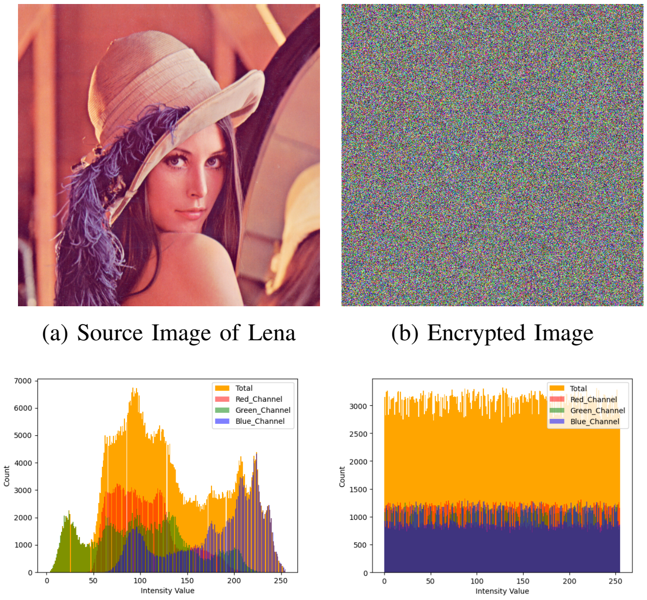
Finally the elements of the matrix undergo substitution using a precomputed Look-Up Table named S-BOX (Substitution Bytes). S-BOX is a 16x16 matrix through which each element (byte) of the matrix is substituted with some other byte by referring the pre-computed S-BOX table.

IV. Histogram Analysis

Histogram of an image is used to show the distribution of intensity values of the pixels of that image. For most Images, when we plot the histogram the intensity values are not uniformly distributed. Although, when the same Image is encrypted using certain Encryption algorithm it should have a histogram which is uniformly distributed or near equal intensity distribution in order to defend against various statistical attacks.

Now, we use standard Lena Image as our Source Image Fig 3(a). When performing Histogram analysis of the source Image, we can see it has a highly variable intensity distribution. Upon encrypting using the proposed Encryption Algorithm, the encrypted image and its following Histogram Analysis is shown in Fig 3(b) and Fig 3(c) respectively.

It can be observed that the encrypted Image is almost uniformly Distributed. The Original Lena Image has highly varied intensity distribution ranging from 1000 to 6,800. The intensity distribution of Encrypted Image ranges from 2800-3100. This reveals that the intensity distribution of the encrypted image is much more uniform than the Original Image. Thus, the proposed algorithm brings a good amount of uniformity in terms of intensity distribution. This observation implies that the proposed algorithm will provide a good level of defense and security against different statistical attacks.



(a) Source Image of Lena (b) Encrypted Image

(c) Histogram of Original Image (d) Histogram of Encrypted Image

Fig. 3: Histogram Analysis

V. Conclusions

In this paper we have presented an image encryption algorithm which is based on Symmetric Key Cryptography. In order to enhance the security and confidentiality of the information new keys are derived from the Secret Key shared between the communicating parties and these new keys are used in computations instead of the original key hence reducing the influence of Man-In-Middle attack. To enhance the final encrypted image making it difficult to decipher, the presented algorithm splits the image pixels into 3 matrices each containing Red, Green and Blue values of the pixels respectively and applies all the computations to each of these matrices separately. These computations consist of low-level operations such as bit level Ex-OR, addition, multiplication, various permutation, substitution and shifting operations. The algorithm consists 2 modules first is of Key Generation which derives novel keys to be used in the algorithm, it also generates a pattern from the Key based on some criteria which is used later in the application of algorithm. Second module consists of all the computations to be applied on the input matrices. These computations are composed of following major steps 1) Mix-Columns Transformation, 2) Upper and Lower Triangular Ex-Or, 3)Shift Rows Transformation, 4) Element-wise Ex-Or and 5) Sub-Bytes Transformation. The resultant matrices are merged into one in order to produce the final encrypted image. This image along with the Shared Secret Key is transmitted to the recipient who decrypts the image. The decryption process is the total reverse of the encryption process. At the recipient The Key Generation module is executed first generating the keys similar to the encryption process and all the other computations are applied in a reverse order to get the Original image. The presented algorithm provides a strong encrypted format of the image which can also be decrypted without any data loss and errors. Hence it provides a secure and efficient means for transfer of images in real time.

REFERENCES

[1] https://www.infosecwriters.com/textresources/pdf/AESbyExample.pdf