Image Encryption And Decryption Using AES Algorithm.

BACHELOR OF COMPUTER ENGINEERING

By

(Arpit Chauhan , Moodle:20102085)

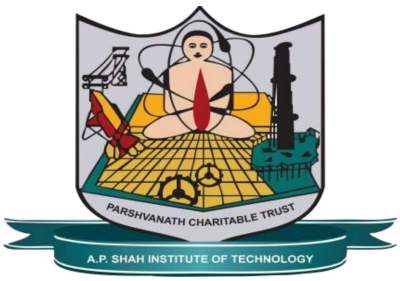
(Kirti Dubey , Moodle: 20102204)

(Pratik Chaudhari , Moodle :20102205)

(Ranjan Shettigar, Moodle:20102177)

Guide:

(Prof. Vishakha Kiran Chaudhari)



Department of Computer Engineering

A. P. SHAH INSTITUTE OF TECHNOLOGY,

THANE (2021-2022)

A. P. SHAH INSTITUTE OF TECHNOLOGY

Project Report Approval for SE

This Mini project report entitled ***(Image Encryption & Decryption Using AES Algorithm)*** by ***(Arpit Chauhan, Kirti Dubey, Pratik Chaudhari, Ranjan Shettigar)*** is approved for the degree of ***Bachelor of Engineering*** in ***Computer Engineering***, ***2021-22***.

Examiner Name Signature 1._____

2._____

Date:

Place:

Declaration

We declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Arpit Chauhan ----------

(signature)

Kirti Dubey ----------

(signature)

Pratik Chaudhari ----------

(signature)

Ranjan Shettigar ----------

(signature)

Abstract

With the fast evolution of digital data exchange, security information becomes much more important in data storage and transmission. Due to the increasing use of images in industrial processes, it is essential to protect confidential image data from unauthorized access. In this paper, we analyse the Advanced Encryption Standard (AES), and we add a keystream generator to AES to ensure improving the encryption performance; mainly for images characterised by reduced entropy. The implementation of both techniques has been realized for experimental purposes. Detailed results in terms of security analysis and implementation are given. Comparative study with traditional encryption algorithms is shown the superiority of the modified algorithm.

CONTENTS

LIST OF FIGURES

1.Introduction:

In today's time, technology has become the major source for consuming data be it for education, entertainment, analysis, news, sharing of information, etc. With the help of the internet, all the information is shared across the world in very less or almost no time. This has made the consumption of information really convenient. But, as this information is digital and uses mainly the internet for its transportation the loopholes in the communication network may subject to mutation of the piece of information. Data security has become a major concern nowadays as the information transferred is easily accessible and can be modified before it reaches its destination. In order to tackle this problem cryptography is used to ensure the security of data. With the help of various algorithms, data can be encrypted and can be transported through the internet without any concern of it getting manipulated. Among many algorithms, Advanced Encryption Standard (AES) is one most the widely used algorithm because of its strong and fast encryption. In the following project, we had used this algorithm for encryption and decryption on image.

2.Literature Survey:

[1] Integrating this architecture with MPEG2 is also proposed. The basic idea behind achieving this is using value transformation, positional permutation and using them simultaneously to produce combining effect. Value transformation works by using binary sequence generated by a chaotic system in which the grey level of pixel is assigned according to the predetermined keys. Using these a completely distorted image is obtained. Both the encryption and decryption units of the program will be using same hardware and will be highly efficient.

The cost of the encryption part is really low as the program is really efficient. A proper procedure must be followed for reconstructing the encrypted and compressed image otherwise this will result in distortion of the image.

[2] In this paper, an algorithm for image encryption and decryption is proposed along with its very large scale integration and keeping three main features in mind:

1) High security.
2) Low complexity.
3) No distortion in image quality.

[3] Advanced Encryption Standard(AES), also called Rijndael encryption. It is the block encryption standard which set by the United States Federal Government lately. AES is used instead of DES. After several rounds of screening that AES was widely used. On November 26, 2011 the NIST released the latest encryption standard screening after five years, and took effect in May 2002.After four years of precipitation and the test, AES became one of the most popular symmetric encryption algorithm.

3.Problem Statement

Since, with the effect of pandemic information technology has seen a tremendous demand in almost every sector. In education, entertainment or corporate sector.

The internet is acting as a functioning party here for transfer of information The risk of data getting corrupted or misused has also has also increased. In order to tackle this misuse of data and protect the confidential information various encryption algorithms are used among them many are based on symmetric key which are less secure and other algorithm are fish algorithm or chaotic key algorithm. These algorithms are not secure and information stored in them can easily be decrypted. Also, these are used mainly to encrypt text based data not any digital image data which is also one their limitations.

In order to protect the data in an image from getting manipulated we had used one of the strong, efficient and reliable algorithms called Advanced Encryption Standard (AES) algorithm. It works on block cipher technique. It supports data length of 128, 192 and 256 bits. It consists of multiple rounds for processing different keys ranging from 10 to 14 rounds. All the information is transferred into matrix form. The digital image is first converted into textual data and the AES encryption is applied to it. It is asymmetric key encryption algorithm. With the help of this digital images can be shared without the risk of them getting manipulated.

4. Objectives & Scope

Objectives

The main objective of our project is to provide security of the digital image by encrypting it by block cipher approach using dynamic key encryption and decryption. We can share the encrypted string and the key to the user who had already registered. The Registered user can easily access the data needed for the decryption. After encryption and decryption there should be no reduction in image quality.

(1) Can resist all known attacks.

(2) Fast and coding compaction.

(3) Simple in design.

Around this kind of design thought, the data packet is prominent in AES algorithm and the key length is variable. The iteration round of number is controlled by the key and the block length.

Scope

Currently, this system works on offline and the image is stored on the local hard drive also, the encryption key is of 128-bit which performs 10-round encryption.

The data in which the image is getting converted along with key is stored in the local database. The key and the string can be shared with the users registered on the portal with just a simple click. The person receiving the image will get a pop-up in his mail box along with the credentials of the image. Using the portal encrypted images can be shared effectively. The program also displays the size of the encrypted and decrypted image to the respective users.

5. Experimental Setup:

5.1 Hardware Requirements:

RAM : 4GB or Above

Processor : Intel i3 Gen 5 and above/ M1 and above

Ryzen 3 Gen 2 and above.

5.2 Software Requirements:

Operating System : Windows 7 or Above

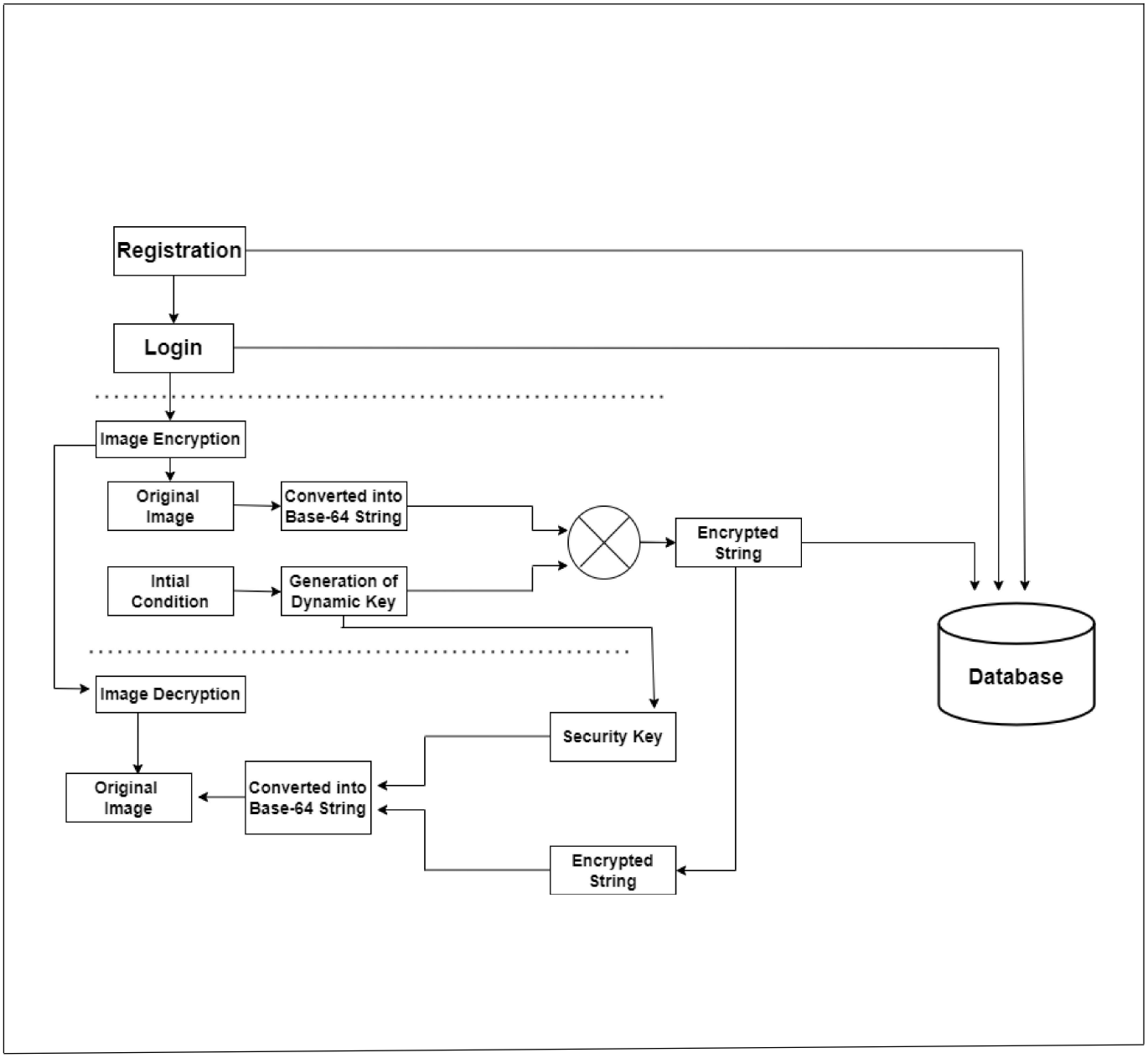
Storage : 64gb or above

Development Kit : Java JDK

Software Used : Netbeans IDE 12.5, MySQL Workbench

6. System Design:

6.1 Block Diagram:



Fig(1.1)

6.2 Algorithm/process:

1. It works on substitution and permutation.

2. A single key is expanded to be used in multiple rounds.

3. AES works on byte data, unlike DES which works on bit data. i.e. for 128 bit it works on 16-byte data.

4. No. of rounds is dependent on the key length.

128-bit key length = 10 rounds

192-bit key length = 12 rounds

256-bit key length = 14 rounds

5. The data is stored in the form of 4*4 matrix it is known as state array.

Each round takes state array as an input and gives an state array as an output.

6. Each cell represents a byte, and each byte consists of 4 words.

7. a. In each round the text to be encrypted is passed through and XOR function along with the round key.

b. Which then leads to byte substitution.

c. After that row shifting is performed on the matrix.

d. Again colomuns are mixed, which leads to a complete mess of the data.

e. After that round key is added. However, in the last round mix column step is skipped. The output of the last round is ciphertext.

7. Implementation

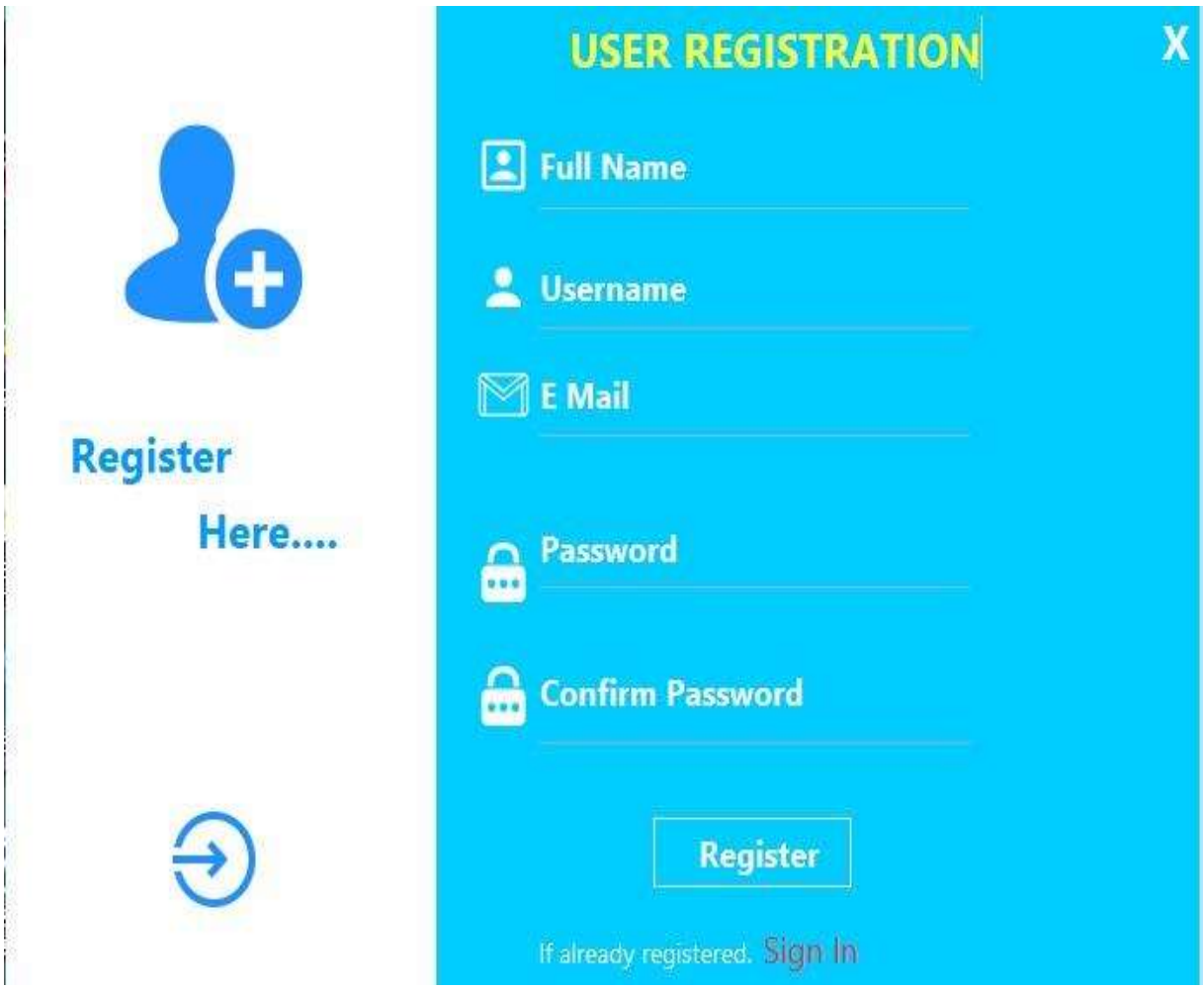
7.1 Code Snapshots

```
private String ConvertImageIconToBase64String(ImageIcon ii) {
    // Create a buffered image of the size of the original image icon
    BufferedImage image = new BufferedImage(ii.getIconWidth(),
    ii.getIconHeight(), BufferedImage.TYPE_INT_RGB);

    // Create a graphics object to draw the image
    Graphics g = image.createGraphics();

    // Paint the icon on to the buffered image
    ii.paintIcon(null, g, 0, 0);

    g.dispose();

    // Convert the buffered image into a byte array
    ByteArrayOutputStream b = new ByteArrayOutputStream();
    try {
    ImageIO.write(image, "jpg", b);
    } catch (Exception ex) {
    // Handle the exception
    }
    byte[] imageInByte = b.toByteArray();

    double bytes = imageInByte.length;
    System.out.println("File Size after: " + String.format("%.2f", bytes/1024) + "kb");

    // Return the Base64 encoded String
    return new String(Base64.getEncoder().encode(imageInByte));
```

```
public static void aesEncryption(String base64String){
    String encKey = getAlphaNumericString(16);
    textSecretKey.setText(encKey);
     try {
         String encryptedString = encrypt(base64String,base64Encode(encKey));
         //System.out.println("encryptedString :: "+encryptedString);
         Encrypt.setText(encryptedString);
     } catch (Exception ex) {
         Logger.getLogger(loginFrame.class.getName()).log(Level.SEVERE, null, ex);
     }

 }

 public static String encrypt(String Data, String secret) throws Exception {
  Key key = generateKey(secret);
  Cipher c = Cipher.getInstance("AES");
  c.init(Cipher.ENCRYPT_MODE, key);
  byte[] encVal = c.doFinal(Data.getBytes());
  String encryptedValue = Base64.getEncoder().encodeToString(encVal);
  return encryptedValue;
}
```

```
private void jButtonDecryptActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

    String encryptedString =jTextFieldEncryptedString.getText();
    String secretKey = textSecretKey.getText();

    if(secretKey.equalsIgnoreCase("") || secretKey.length()!=16){
        JOptionPane.showMessageDialog(null, "Please enter valid secret key..");
    }

    String decryptedBase64String = decrypt(encryptedString,base64Encode(secretKey));

    //jTextFieldEncryptedString.setText(decryptedBase64String);

    imageBytes = Base64.getDecoder().decode(decryptedBase64String);
    try {
        BufferedImage image = ImageIO.read(new ByteArrayInputStream(imageBytes));
          jLabelImage.setIcon(new ImageIcon(image));

        ImageIcon ii = new ImageIcon(image);
          Resize image to fit jlabel
        Image image1 = ii.getImage().getScaledInstance(jLabelImage.getWidth(), jLabelImage.getHeight(), Image.SCALE_DEFAULT);

        jLabelImage.setIcon(new ImageIcon(image1));

        double bytes = imageBytes.length;
        String fileSizeInKB = String.format("%.2f", bytes/1024) + "KB";
        fileSize.setText(fileSizeInKB);
        System.out.println("File Size : " + fileSizeInKB);
```

7.2 Project Snapshots:


Register
Here....
USER REGISTRATION
X
Full Name
Username
E Mail
Password
Confirm Password
Register
If already registered. Sign In

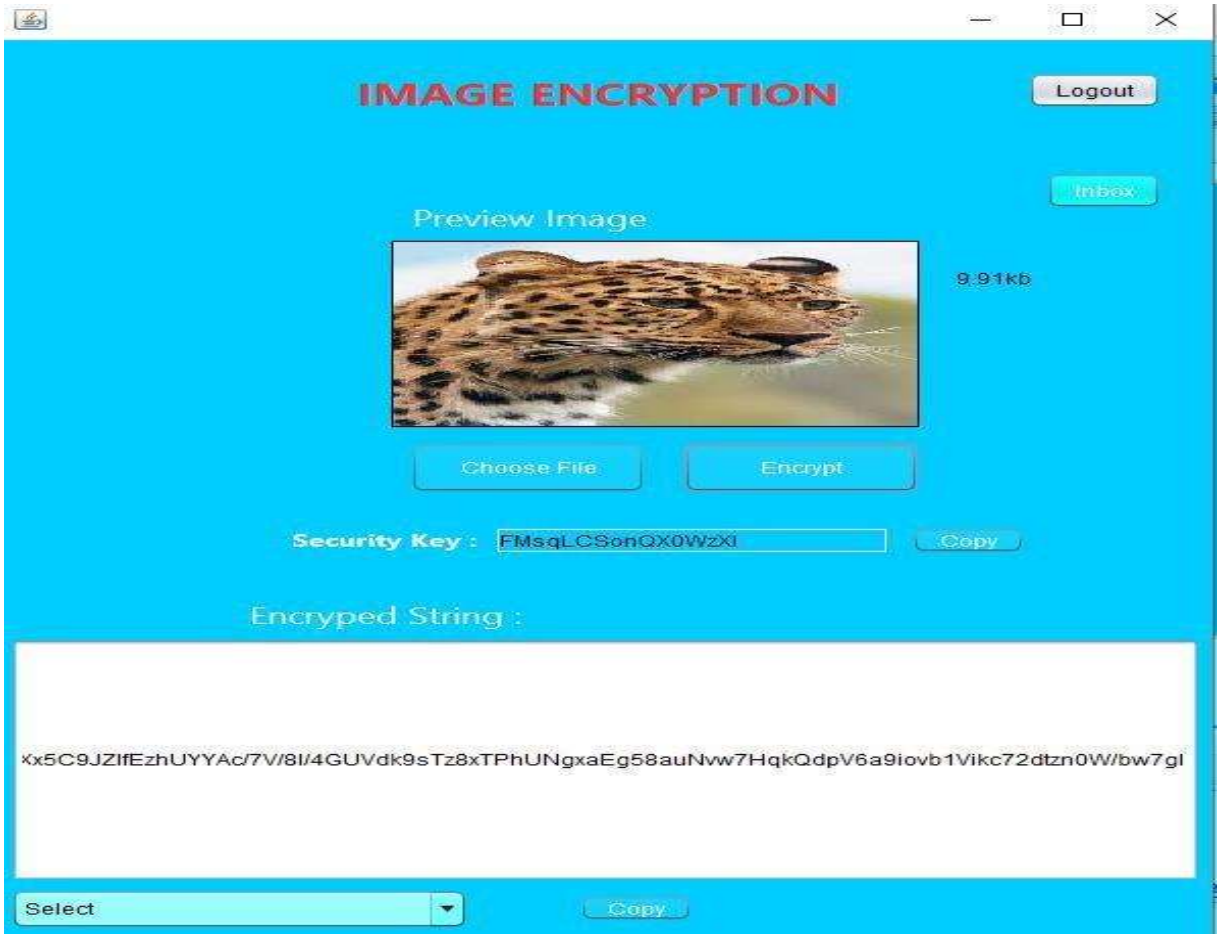


IMAGE ENCRYPTION
Logout
Inbox
Preview Image
9.91kb
Choose File
Encrypt
Security Key :
FMsqLCSonQX0WzXl
Copy
Encryped String :
xx5C9JZlfEzhUYYAc/7V/8l/4GUVdk9sTz8xTPhUNgxaEg58auNvw7HqkQdpV6a9iovb1Vikc72dtzn0W/bw7gl
Select
Copy


IMAGE DECRYPTION
Logout
Encrypted String
UCbEbQH+uCzdJHQhd6faB6XjeryZflWCF0PrRiP2aod
Security Key:
QOhdishrlnFkLJ6r
Decrypt
Download
Decrypted Image
9.91KB
Encrypt

7. Result and future scope:

The size of decrypted image is same as the encrypted image indicating no loss in the image quality. This is valid for all types of files jpeg, png etc. The encryption of the image takes around 1.5 seconds and decryption takes around 1.8 seconds. This indicates the consistency, speed and reliability of AES algorithm. The digital image is getting converted to text file and data is encrypted successfully using AES.

Currently, this system works offline and the image is stored on the local hard drive also, the encryption key is of 128-bit which performs 10-round encryption. In the future, we will scale this model as a web-based application and store the data on online servers and the user credentials will be stored. We will also implement this algorithm to encrypt documents with 192-bit and 256bit key encryption will be added. We will also add another strong algorithm 2D Cellular Automata which is more powerful, advanced and secure as compared to AES.

8. Conclusion

Thus we have made an image encryption and decryption application based on AES algorithm by converting it into text file and applying 10-key round encryption on it. The program is the connected to a database to store the user's credentials and also the encrypted string. By using this process security of the data is ensured. As the algorithm has powerful numerical calculation function, especially for arrays and matrix calculations, and the infrastructure of the AES algorithm uses the matrix as the basic unit. So to implement the image encryption based on AES algorithm in the MATLAB environment is easy. Due to the AES algorithm is easy to implement in software and hardware, it has laid a good foundation for subsequent image encryption in the transmission encryption on software and hardware. So we have reason to believe that use this method to encrypt the image will have a very good prospect in the future.

References:

[1] Chen, Chao-Shen, and Rong-Jian Chen. "Image encryption and decryption using SCAN methodology." 2006 Seventh International Conference on Parallel and Distributed Computing."

[2] M.Zeghid, M.Machhout, L.Khriji, A.Baganne and R.Tourki, "A Modified AES based Algorithm for Image Encryption."

[3] QiZhang, Qunding, Electronic Engineering College,Heilongjiang University, Harbin, China. "Digital Image Encryption Based On Advanced Encryption Standard Algorithm."