A Novel Flipped Bit Method for Image Encryption

REPORT SUBMITTED FOR THE PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF BACHELOR OF TECHNOLOGY IN
Information Technology

Submitted by

ANANYA ROY (IT2014/010)

ABIRA KUNDU (IT2014/015)

PRASHANT UPADHYAY (IT2014/048)

Under the Guidance of
PROF. (DR.) SIDDHARTHA BHATTACHARYYA
Dean (Research & Development/Academic Affairs),
RCC Institute of Information Technology, Kolkata

RCC Institute of Information Technology
Canal South Road, Beliaghata, Kolkata – 700 015
[Affiliated to West Bengal University of Technology]

Acknowledgement

We would like to express our sincere gratitude to Prof. (Dr.) Siddhartha Bhattacharyya of the department of Computer Application, whose role as project guide was invaluable for the project. We are extremely thankful for the keen interest he took in advising us, for the books and reference materials provided for the moral support extended to us.

We would like to express our special gratitude and thanks to Dr. Indrajit Pan for giving us such attention and time.

Last but not the least we convey our gratitude to all the teachers for providing us the technical skills that will always remain as our asset and to all non-teaching staff for the gracious hospitality they offered us.

Place: RCCIIT, Kolkata

Date: 25.04.18

(Ananya Roy)

(Abira Kundu)

(Prashant Upadhyay)

Department of Information Technology
RCCIIT, Beliaghata,
Kolkata – 700 015,
West Bengal, India

Approval

This is to certify that the project report entitled “A Novel Flipped Bit Method for Image Encryption” prepared under my supervision by Ananya Roy (IT2014/010), Abira Kundu, (IT2014/015), Prashant Upadhyay (IT2014/048) be accepted in partial fulfillment for the degree of Bachelor of Technology in Information Technology.

It is to be understood that by this approval, the undersigned does not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn thereof, but approves the report only for the purpose for which it has been submitted.

……..

Prof. (Dr.) Siddhartha Bhattacharyya
Dean (Research & Development/Academic Affairs),
RCC Institute of Information Technology, Kolkata

……..

Abhijit Das
Associate Professor & Head of the Department
Information Technology, RCCIIT, Kolkata

Certificate of Acceptance

The report of the Project titled "A Novel Flipped Bit Method for Image Encryption" by Ananya Roy (IT2014/010), Abira Kundu, (IT2014/015), Prashant Upadhyay (IT2014/048) 8th Semester of 2018 has been prepared under our supervision for the partial fulfillment of the requirements for B Tech (IT) degree in Maulana Abul Kalam Azad University of Technology.

Name of the Examiner	Signature with Date
1. …….	…….
2. ……..	……..
3. ……	……
4. …….	……

Table of Contents

1. Introduction

When Alpha wants to send any data to Beta he replaces every A with E, every B with F, and so on. So only the person who knows the 'shift by 4' method can only decode the message and this is what we called encryption and decryption.
Now data that can be read and understood without any special measures are plaintext and disguising plaintext by any method is known as encryption. By encrypting the data we can ensure that the information is hidden and is safe and can only be opened or decrypted those who know the way of decryption.
Now in this era everyone is on the digital world and exchange data on the same platform but the data they are sharing is safe or not is unknown to the user. So the data which gets stored in the cloud for sharing must be encrypted and should be secured from the external users who want to decrypt it without the correct combination. So we have Cloud computing which will store the data and Fog computing which is to process the data between end users and the Cloud computing.

Fog Computing is a paradigm which extends Cloud computing and other services to the edge of the network. Fog provides data, compute, storage, and application services to end-users just like cloud. Fog Computing enables a new breed of services at the edge to deliver a wide variety of applications for Internet of Things (IoT) devices. It has been introduced as a technology to bridge the gap between remote data centers and Internet of Things (IoT) devices. Enabling a wide range of benefits, including enhanced security, decreased bandwidth, and reduced latency, fog is an appropriate paradigm for many IoT services.
Fog computing that acts as an intermediate layer in between Cloud data centers and IoT devices/sensors. It offers compute, networking and storage facilities so that Cloud-based services can be extended closer to the IoT devices/sensors. CISCO first introduced the concept of fog in 2012 to address the challenges of IoT applications in conventional Cloud computing.

2. Literature Survey

In today's world, almost every organization are using cloud computing technology to protect their data and using cloud resources when necessary. The existing mechanism only works with security of data. It neither detects any invalid access nor prevents any valid distribution of data. The proposed mechanism facilitates security features to data and thereby allows for detection of invalid access. If the attacker is a malicious insider, data theft attacks are amplified. This is considered to be as one of the top threats to cloud computing by the Cloud Security Alliance. Fog Computing provides security to the data stored in the cloud. If any unauthorized user tries to access the data in the cloud, then the security will track the user and will map all the data concerned with the user. We use this technology to launch disinformation attacks against malicious insiders, preventing them from distinguishing the real sensitive customer data from fake worthless data.

The existing system is less secure and can be easily hacked by any hacking professionals. Anyone who has got unauthorized access to cloud can search for files and data. The system is unable to identify whether the user is legitimate or not. If the person is illegitimate then also this system sends the original information. Encryption is provided to existing system but cloud and data is not secure by only encryption.

[2]We propose a completely different approach to securing the cloud using decoy information technology. Here we consider Fog Computing as a paradigm through which we can provide local access to the user and with the help of decoy technology, we provide security for user data and prevent insider theft attacks.

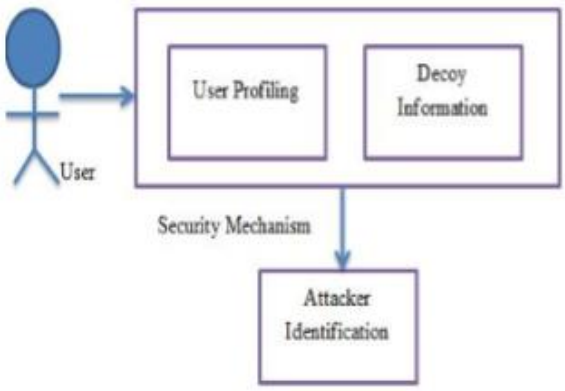

Fig 1. Security Mechanism

The proposed security mechanism makes use of two concepts known as user behavior profiling and decoys. Users who access cloud to view their own data and also perform data dynamics are expected to have some specific patterns of usage. Such users are known as normal users. This normal behavior of users is profiled in the first phase. Then decoy information is kept in file system. This decoy system consists of bogus files like social security number, credit card details as file names on it, provided by the service

provider. When an unauthorized system enters into the data and downloads the files an alert will be generated, and the system will be notified with the attack. The insider theft attackers generally do not have the behavior of normal user. For this reason they are attracted to use decoy information. As the decoy information is not the real data there is no problem when hacker uses it or steals it.

2.1 User Behavior Profiling:

[1]User behavior profiling deals with the behavior of the user. They monitor data access in the cloud and detect abnormal data access pattern. It is a well known technique that can be applied to check how, when and how much a client access their data in the cloud. Such normal user behavior can be continuously checked to determine whether abnormal access to a user's data is experience. That means, when any person get access in the cloud, our system start detecting behavior of that person on the basis of following characteristics:

1. Login Time
2. Session Time
3. Upload Count
4. Download Count
5. How many files he will read and how often.

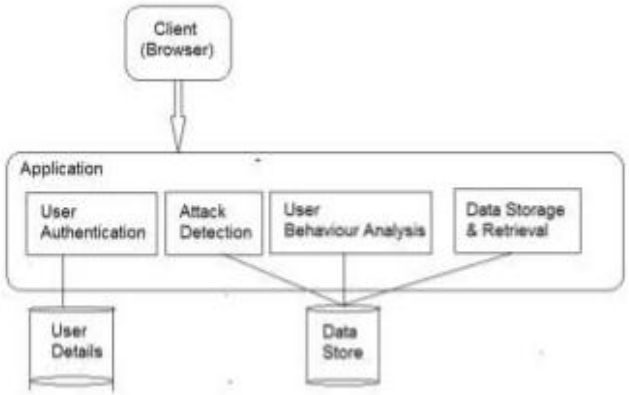

Fig 2: User Behavior Profiling

User Behavior Profiling Algorithm

1. Identify operation is executed first.
2. Tracking of user behavior profile with parameters like username, login password specified, user key specified during document access, type of document for access.
3. During login, login password is tracked.
4. During document access, the user key specified is tracked along with the type of operation (valid or invalid).
5. Profile is classified as valid or invalid using the following mathematical operation: P(IV)=count(invalid operations of each type)/count(operations of each type). If the value P (IV) is above a threshold parameter then the profile is categorized as invalid and the user is redirected to the decoy module.

The system compares all above new data set with predefined data sets which we store in the database and identify that person is authorized person or not and according to that system will send the data.

2.2 Decoy Information Technology:

[2]Decoy information, such as decoy documents, are delivered to the attacker when an unauthorized access is detected. The file sent to the attacker is in encrypted format. The decoy technology is very useful as it deceives malicious insiders. When the decoy technology is used along with user profiles, it is possible to know the suspected behavior of users and that way it is possible to prevent insider data theft attacks. The decoy system serves two purposes:
(A) Validating whether data access is authorized when abnormal information access is detected.
(B) Confusing the attacker with bogus information.
We assume that the combination of these two security features will provide better levels of security for the Cloud. These concepts are applied to detect illegitimate data access to data stored on a local file system by the attackers who impersonate legitimate users after stealing their credentials.

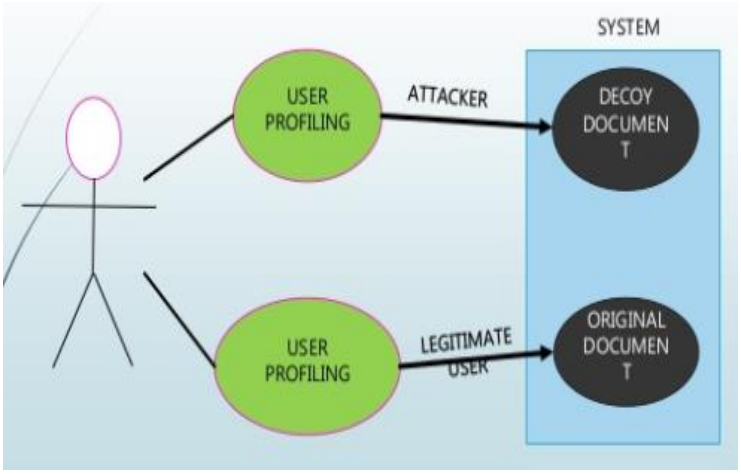

Fig 3: Decoy System

There are still some problems with the existing method leading to hacking and accessing of data in fog. This motivated us to think for encryption of data in the fog level. Hence, in this way we are trying to achieve more security by introducing encryption to the data by using the Advanced Encryption Standard algorithm technique.

DATA ENCRYPTION STANDARD (DES) was once the most widely used encryption standard. It uses symmetric key algorithm for data encryption. It has 56 bits of key size and block size is 64 bits. It was considered as the basic of advanced modern cryptography system. Although DES was not the most secured algorithm because of its key size which can be brute forced easily. The techniques that can break a key faster than brute force are Differential cryptanalysis, Linear cryptanalysis and Improved Davies Attack.

TRIPLE DES When DES failed, 3DEF came to existence which is the predecessor of the DES algorithm. It is named as Triple Data Encryption Standard where 3 instances of DES are cascaded. 3DEF made no change in the algorithm except the increase in key size where it can have variable key size like 56,112 or 168 bits of key size and block size remains the same as 64 bits. 3DEF was said to be 2 ½ times more secured as DES algorithm. Still vulnerability to security attacks was there and as it was designed for hardware implementation, it did not function properly in software applications.

ADVANCED ENCYPTION STANDARD To overcome the above problems, AES came to existence. AES is the successor of DES which uses standard symmetric key encryption for many of the US federal organizations. It was considered to be more effective, more advanced and secure standard. It is found at least six times faster than 3DES. AES accepts block size of 128 bits and key size of 128,192,256. Many organizations tried to break the key but it was unbreakable. On comparing all the above encryption techniques AES would be better and more secured type.

There are chances of leakage of data from cloud to fog or vice versa, hence by adding encryption to already encrypted data makes nearly impossible to access the data. By this method the data could overcome the man in the middle attack where an attacker will be continuously trying to enter into the data and this might be a threat to the sensitive data.

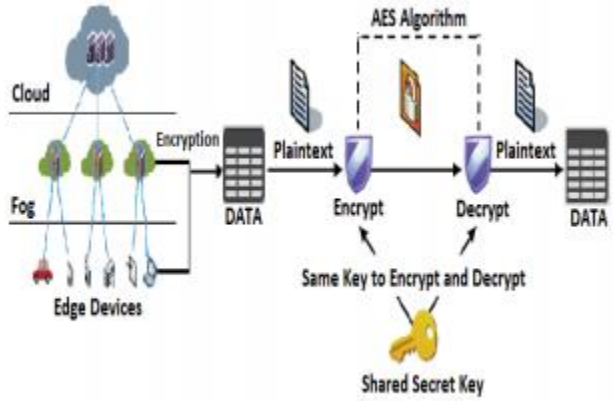

Fig 4: System Architechture

As in fig 5, the technique of encryption has been chosen to apply over the data in fog for more security. The method is all about using AES encryption algorithm and applying it over the selected datasets through deploying it in a mobile edge device and thereby collecting the performance metrics over three datasets and evaluating the best and worst cases in all the aspects of datasets. Different datasets having different sizes of data and of text, strings, and images are selected for testing the encryption method. On evaluating various factors like encryption, decryption time, utilization of memory, response time for each data set, best and worst possible cases are monitored.

3. Problem Definition

1. Encryption and Decryption: We will be taking an image and we break it into a matrix (decimal value) and then binarised that value, reversed it into 8 bit framework. Then we have taken out the decimal value form it and pushed it inside the same matrix column. Same procedure follows for the decryption part.

2. Proposed methodology in Encryption and Decryption: As we know, there are two theorems – Again we have tried to implement RSA algorithm which is used for encryption and decryption but as the block size in image each of red, green and blue is of 8 bytes, it was difficult to implement RSA Algorithm on this.
Further, we have taken each pixels of RGB (red, green, blue), we swapped the 0 bits with 1, and 1 bits with 0.Then we reversed the whole block for R,G and B and that made our encryption a bit more stronger than what we have proposed in the previous encryption and decryption method.

4. SRS (Software Requirement Specification)

Software required: Python, PyCharm

Package Used: NumPy, Pil.

5. Future Scope

Fog is considered as a nontrivial extension of the cloud. Hence security and privacy challenges will continue to persist. Fog has its distinct characteristics such as mobility support, computational power, connectivity to serve various security purposes. Fog computing could provide not only additional computational resources, but also an upgraded level of security that will help in minimizing the attacks in IoT environments.

The main problem in cloud is security. Cloud provides pay per use concept and every organization from small scale to large scale wants to store their data into the cloud. With the increase of number of devices connected to the cloud, information storing and retrieval process becomes more complicated. Man in the middle is one type of attack in cloud. Cloud can't differentiate between the attacker and the user. Twitter is one of the example of data theft from the cloud**.** Hence developing a better cloud is not enough because there will always be chances of continuous attacks in the cloud and leakage or loss of data. To overcome this issue fog-computing is evolved. In fog-computing process, application comes to the data, not the data to the application. So fog can be considered as an extension of cloud computing but not a replacement of it and also a more secured form of data storage.

Both Cloud and Fog provide data, computation, storage and application services to end-users. However, Fog can be distinguished from Cloud by its proximity to end-users, the dense geographical distribution and its support for mobility. We adopt a simple three level hierarchy as in Figure.

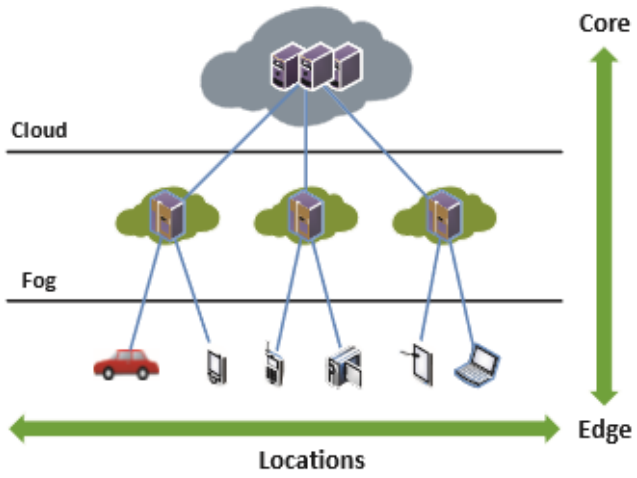

Fig 5: Fog between edge and Cloud

Fog computing is extremely virtual platform offering network services, storage, computational services to the cloud servers and the end user devices. The main reason for adopting fog computing is for the applications having latency issues. When the Internet of Things is implemented billions of devices will be added into the network. The cloud computing will not be able to provide mobility support, location awareness and low latency. Fog computing promises to overcome all the problems mentioned above.

6. Proposed Method

In our project we have done a work to illustrate a bit-reversal mechanism for encrypting the digital images in cloud system. An images has three formats like RGB, gray-scale and black & white. Our proposed method generates the indexed color-map of the selected image. Index matrix of the indexed image, which has a mapped representation of pixel values to color-map values, is then encrypted using the system key. Then the index image matrix is divided into some segments or blocks. Then the bit-reversal permutation method is applied to the each of the blocks for generating the cipher block. Thus the color-map matrix is remaining same. After that the encrypted indexed image is again re-permuted to generate the original form. This encryption mechanism deals with different types of digital images towards safe storage in cloud. Experimental results show that the method is quite simple yet effective for security of user data.

Bit-reversal Permutation

Bit-reversal permutation concept belongs to applied mathematics. A sequence of k different items is permutated and rotated using this mechanism. Number of items (k) are

usually represented in power of 2 (where k = 2n). If n = 2, then k = 4 and the numbers will be 00, 01, 10, 11 After applying bit-reversal technique, the above series will become 00, 10, 01, 11. Here each number will be read in reverse, which means earlier any number 01 will be converted to 10. Similarly if any number is 001 then its bit-reversal form will be 100. Bit-reversal mechanism follows involution, which means if bit-reversal technique is re-applied on the data then it will produce back the old record.

Design Background

Digital image has different modes like RGB (Red, Green and Blue) or true-color (though a true-color image has a little bit difference from RGB), gray-scale or gray-level and black & white etc. Each of the modes contains different color pallets. The RGB or true-color mode shows highest degree of color variation. The information storage of RGB or true-color images requires three-dimensional matrix structure. In image processing applications, the operations on gray-scale images are very common because it is less complex in computational requirements and structural relation. It can hold the major properties of an image which is impossible in black & white image. Conversion of gray-level image to RGB is difficult and it may lose the major properties of a RGB image. The encrypted digital image needs to maintain the quality and information of the original image after its decryption. In our work, for application of encryption mechanism, a concept of indexed-color map image is applied. Indexed-color approach helps to store the digital RGB image. To store pixel wise information on index value of color map mechanism, an index is created. Here a separate color map or palette is created to color information. Index matrix and color-map, are two dimensional. All modes of digital images can be decomposed in indexed image format, which develops an indexed matrix and a color-map matrix. These two matrices hold the complete information of the image. To reconstruct the main image, these two matrices can be combined together. Proposed work performs the encryption and decryption operation on the indexed matrix.

Proposed Algorithm

Encryption

[3]On index matrix of an indexed image bit-reversal encryption approach is applied. During upload a digital image is converted into an indexed image. This process divides the image information into two matrices i.e. index matrix and color-map matrix. Index matrix is then processed for encryption method. Index matrix contains decimal values which indicate the index location of color-map matrix for each pixel of the image. At first the checking of the dimension (i.e. span of row and column) of index matrix is done. Each pixel of the image incoming in decimal are then converted to its equivalent binary representation. Considering 512 different color entries for color-map, is represented in 10-bit binarized representation.

Once the binary equivalent of each pixel index is generated, then it is processed through bit-reversal modification. After bit-reversal an encrypted index matrix is generated.

PSEUDO CODE:

1. Take an image as input.
2. Convert it into matrix.
3. Loop from 0 to Size of array.
4. Loop from 0 to Size of array(row).
5. Loop from 0 to Size of array(row a[i]).
6. Binarise the decimal value.
7. Reverse the binarised value.
8. Convert it from binary to decimal.
9. Push back to the same row, column.
10. Convert the matrix to image, save it into .png format.

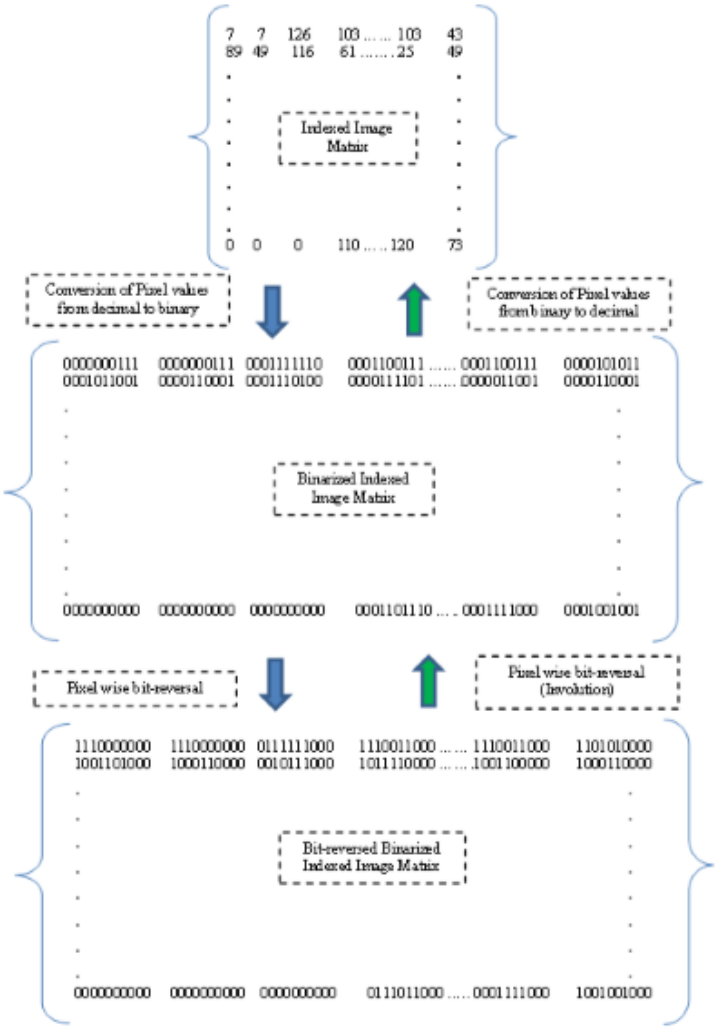

Fig 6: Flow of Encryption and Decryption

Decryption

To get back the earlier state and to generate the main matrix the bit-reversal technique is applied again on the encrypted index matrix. Pixel wise binary values are converted to decimal equivalent after applying bit-reversal on the encrypted matrix. It generates the original index matrix of the corresponding image.

PSEUDO CODE:

1. Take an encrypted image as input.
2. Convert it into matrix.
3. Loop from 0 to Size of array.
4. Loop from 0 to Size of array(row).
5. Loop from 0 to Size of array(row a[i]).
6. Binarise the decimal value.
7. Reverse the binarised value.
8. Convert it from binary to decimal.
9. Push back to the same row, column.
10. Convert the matrix to image, save it into .png format.

7. Result

Fig 7: Sample Result after Bit Reversal Encryption

The above bit-reversible approach for digital image encryption is implemented on different types of images. Applying on on index matrix of an image, it is observed that the encrypted index image doesn't contain any potential information whereas the decryption process returns the original image without any distortion. Operational complexity of this present process is very less, though the actual complexity of the algorithm is not yet derived.Also,the visual standard of encrypted image was not good in the result. Anyone could decode it easily. So we tried RSA algorithm as our next approach.

8. RSA Algorithm

RSA algorithm is an asymmetric cryptography algorithm. The term asymmetric refers to working on two different keys i.e. Public Key and Private Key**.** As the name suggests, Public Key is given to everyone and Private key is kept private.
RSA encryption is a public-key encryption technology developed by RSA Data Security. RSA encryption algorithm uses prime factorization as the trap door for encryption. Therefore, it takes a huge amount of time and processing power to deduce the RSA key. RSA is the standard encryption method for important data, especially data that's transmitted over the Internet.

The basic idea behind RSA is the fact that it is difficult to factorize a large integer. The public key consists of two numbers where one number is multiplication of two large prime numbers. And private key is also derived from the same two prime numbers. The private key can be compromised if somebody can factorize the large number. Therefore encryption strength totally lies on the key size and if we double or triple the key size, the strength of encryption increases exponentially. RSA keys are generally 1024 or 2048 bits long, but experts believe that 1024 bit keys could be broken in the near future. But till now it seems to be an infeasible task.
So we tried to implement the encryption using the RSA algorithm but the block size of RSA algorithm is too high (1024-4096) whereas that of an image is of 8 bytes so it’s not possible to implement encryption with RSA.

9. Bit Flip Effect

Now, our next approach is simple and related to what we have done in our first part. We will take the image and convert each block of RED, GREEN & BLUE into binary value.
Now we have a 8 bit binary value each of RED, GREEN & BLUE block then we will exchange the ZERO value with ONE and vice-versa and after completing the exchange we will reverse the total block, and we will perform this on every block of RED, GREEN & BLUE. In this way bits are continuously flipped. At last we will convert the total value into decimal and store it.

Algorithm for ENCRYPTION

Step 1: Take the image as input
Step 2: Convert the image into a three dimensional array.
Step 3: convert each decimal value to binary value.
Step 4: make the binary value of size of 8 bit by appending extra zeros in front.
Step 5: convert the binary value to an array.
Step 6: swap the zero value with one and vice versa in the given array.
Step 7: reverse the array after swap.

Step 8: convert the reversed value and store it over the place from where the value was been extracted.

Algorithm for DECRYPTION

Step 1: Take the image as input
Step 2: Convert the image into a three dimensional array.
Step 3: convert each decimal value to binary value.
Step 4: make the binary value of size of 8 bit by appending extra zeros in front.
Step 5: convert the binary value to an array.
Step 6: reverse the array.
Step 7: swap the zero value with one and vice versa in the given array.
Step 8: convert the value and store it over the place from where the value was been extracted.

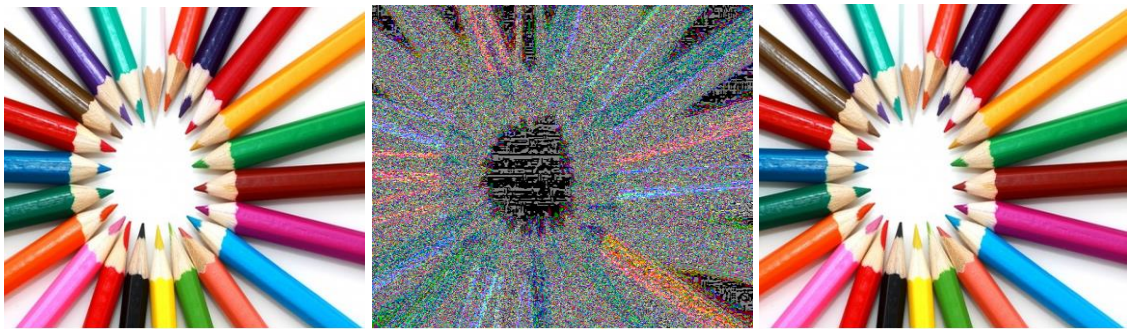

Fig 8: Sample Result after Bit Flipped Method

10. Conclusion

In the bit reversal encryption method, we tested upon its robustness against diverse statistical and differential attacks and found it vulnerable. That's why our final proposed method Bit Flipped was applied. We have analyzed its strength against different modes of statistical and differential attacks. Also some hybrid intelligent techniques are appended with it to convert the present work into a key based encryption and decryption process.
Though, quality of encryption through metrics analysis has not been tested by this method. Last but not the least, simulation of these techniques in Fog simulator has not been done, which could be one of the major future scope of our project.

11. References

1.Fog Computing: Securing the cloud and preventing insider attacks in the cloud. Aatish B. Shah, Jai Kannan, Deep Utkal Shah, Prof. S. B. Ware , Prof. R. S. Badodekar Department of Information Technology, Sinhgad Institute of Technology, Lonavala

2. FOG Computing: Preventing Insider Data Theft Attacks in Cloud Using User Behavior Profiling and Decoy Information Technology. Gayatri Kalaskar, Purva Ratkanthwar, Prachi Jagadale, Bhagyashri Jagadale. Department of Computer Enginneering, I2IT college Pune.

3. A Study on Cloud and Fog Computing Security Issues and Solutions. Archana Lisbon A, Department of Computer Science Christ University, Bengaluru, INDIA, Kavitha R, Assistant Professor, Department of Computer Science Christ University, Bengaluru, INDIA

4. Security in Fog Computing through Encryption. Akhilesh Vishwanath, Department of Computer Science, Kennesaw State University, Kennesaw, Georgia, USA Ramya Peruri, Department of Computer Science, Kennesaw State University, Kennesaw, Georgia, USA Jing (Selena) He, Department of Computer Science, Kennesaw State University, Kennesaw, Georgia, USA

5. Bit-reversal Encryption towards Secured Storage of Digital Image in Cloud Deployment

6. Wang, Q., Wang, C., Li, J., Ren, K., Lou, W.: Enabling Public Verifiability and Data Dynamics for Storage Security in Cloud Computing. In: Proceedings of Computer Security, pp. 355 – 370. (2009).

7. Fog Computing for the Internet of Things: Security and Privacy Issues
Arwa Alrawais, Abdulrahman Alhothaily, Chunqiang Hu, and Xiuzhen Cheng George Washington University

12. Appendix

CODE FOR ENCRYPTION

```
import numpy as np
from PIL import Image
img = Image.open('2.jpg', 'r')
arr = np.array(img)
z=arr
#print(arr)
for i in range(len(z)):
```

```
    for j in range(len(z[i])):
        for k in range(len(z[i][j])):
            a = z[i][j][k]
            p = int(bin(a)[2:])
            p = list(str('%08d' % p))
            for l in range(len(p)):
                if p[l] == '1':
                    p[l] = 0
                else:
                    p[l] = 1
            p = p[::-1]
            z[i][j][k] =
(p[0]*(2**7))+(p[1]*(2**6))+(p[2]*(2**5))+(p[3]*(2**4))+(p[4]*(2**3))+(p[5]*(2**2))
+(p[6]*(2**1))+(p[7]*(2**0))

im = Image.fromarray(arr)
im.save("test.bmp")
print("DONE")
```

CODE FOR DECRYPTION

```
import numpy as np
from PIL import Image
img = Image.open('test.bmp', 'r')
arr = np.array(img)
z=arr
#print(arr)
for i in range(len(z)):
    for j in range(len(z[i])):
        for k in range(len(z[i][j])):
            a = z[i][j][k]
            p = int(bin(a)[2:])
            p = list(str('%08d' % p))
    p = p[::-1]
            for l in range(len(p)):
                if p[l] == '1':
                    p[l] = 0
                else:
                    p[l] = 1
            z[i][j][k]=(p[0]*(2**7))+(p[1]*(2**6))+(p[2]*(2**5))+(p[3]*(2**4))+(p[4]*(2**
            3))+(p[5]*(2**2))+(p[6]*(2**1))+(p[7]*(2**0))

im = Image.fromarray(arr)
im.save("orignal.bmp")
print("DONE")
```