# Image Cryptography

Analysis of Image Cryptosystem

Farah Sultana
fsultan001@citymail.cuny.edu
City College of New York
USA

Mohammad Islam
mislam046@citymail.cuny.edu
City College of New York
USA

Rezwana Kabita
rkabita000@citymail.cuny.edu
City College of New York
USA

Nour Elabbasy
nelabba000@citymail.cuny.edu
City College of New York
USA

## ABSTRACT

Digital communication has become broader by the fast development of Internet technology. The demand for secure communication over the Internet is not merely a tool or service of business, but something more than that. It is an insurance of protection against both legal obligations and illegal activities. Widely used secure communication protocols in the financial industry and e-commerce, rely on secure key exchange and chaos-based encryption algorithms such as RSA and AES to safely send the picture over the Internet. In this paper, our motive was to analyze on two image encryption methods RSA and AES algorithm which have been used over the years to protect confidential data. We would research on the proposed algorithms and at the end, we will analyze on the performance to determine the efficiency of the algorithm in ciphering the image. The algorithms were implemented using Python.

## KEYWORDS

RSA, Image encryption, Image decryption, chaos based RSA, AES

**ACM Reference Format:**
Farah Sultana, Mohammad Islam, Rezwana Kabita, and Nour Elabbasy. . Image Cryptography: Analysis of Image Cryptosystem. In . ACM, New York, NY, USA, 8 pages.

## 1 INTRODUCTION

In the era of the digital environment, the majority of the people are using the internet to make media communications and picture transmission through networks. Keeping prying eyes off of confidential data such as protected health information for the healthcare industry, credit cardholder data for the e-commerce and retail industry and governmental and military documents is vital. To make the data secure, encrypted and confidential we have to ensure the information is not getting disclosed to unauthorized channels, and are only made available to authorized channels as per requirement. Many algorithms have been developed over the years for “public key” cryptography so far amongst which the most widely used one was the RSA algorithm. In this paper, we will explain how the traditional RSA works, discuss RSA’s performance relative to the chaos based RSA and then explain why the AES algorithm is more efficient to encrypt and decrypt multimedia content.

## 2 BACKGROUND

**RSA:** The RSA encryption system is known as Rivest–Shamir–Adleman (RSA) cryptosystem developed by three MIT professors . Rivest, A. Shamir, and L. Adleman. I. RSA supports an asymmetric encryption scheme in which you can use one key to encrypt the message and a different key to decrypt a message. In asymmetric cryptosystems, two various keys are fundamental: the public and private keys. The Image is encrypted by the sender and sends it to the receiver who decrypts the image with his private key. RSA is public key cryptography algorithms that are used for encryption and decryption data.

**AES:** The AES algorithm was first developed in 1998 by two Belgian cryptographers called Vincent Rijmen and Joan Daemen. AES algorithm is a symmetric block cypher that uses the same keys for encryption and decryption. This algorithm is used because it is easy to implement, has high security, and it has a fast encryption and decryption time. Additionally, AES does not require a huge amount of memory like other encryption algorithms such as DES. The three sizes for the AES encryption keys are 128, 192, and 256 bits. A 256 bit key is the strongest and most secure that it is commonly referred to as military grade encryption. However, the reason why it is not used for everything is that it is draining. For example, if there was an application that used that key then the battery would be drained faster in comparison to one that used the 192 bit key. AES has become the encryption standard world wide from Wi-Fi to government agencies such as the National Security Agency due to the many advantages that we mentioned.

## 3 OBJECTIVES

The goal of this research paper is to encrypt and decrypt images of different sizes using the RSA, chaos based RSA and AES algorithm. We will analyze the efficiency of encryption and decryption. In addition to the efficiency, we will analyze if there is any data lost in the decrypted images i.e if the decrypted grey or color image comes out to be exactly as the corresponding original image.


*CSc 48600, Fall 2021, City College of New York*

## 3.1 Farah's Objectives

My research will take into account the encryption of multimedia contents using RSA-original and RSA-chaos based and analyze the efficiency of the encryption process. I have carried out several experiments on different sized images that include true color, high and/or low pixel components. The algorithms are implemented by using the python program. First the keys are generated and then the pixels of the images were taken into account while encrypting the images using the proposed algorithms. Another aim of mine is to understand the AES cryptography concepts and compare it to RSA and chaos based RSA algorithms to demonstrate the efficiency in performance and also via visual inspection.

## 3.2 Mohammad's Objectives

My research was focused on the process of optimizing chaos based RSA algorithm by manipulating certain mathematical properties to achieve better RSA performance. I have analyzed the encryption procedure set by Farah and my aim was to decrypt the images accurately . To achieve the best understanding I have performed several testing on various images.I have extracted the pixels information in decryption process and :acquired the target sample image. Finally I have researched on AES concepts and understand how it works and why it performs faster than the first two algorithms.

## 3.3 Nour's Objectives

One of my objectives for this project revolved around researching what AES is in detail. I read many articles about why it is commonly used now, how the encryption and decryption process works, along with what were it's disadvantages and advantages. This information helped me with my second objective which was to analyze the results for the AES algorithm given different size images and also comparing it to the RSA and the RSA-chaos based algorithms to see which preformed the best.

## 3.4 Rezwana's Objectives

My individual objective for this report will be to analyze, and explain the AES algorithm for image cryptography. I will also implement the AES algorithm and verify the results.
I have implemented the AES algorithm using Python Crypto libraries to encrypt and decrypt images. I ran the script for several different sized images.Then, I took an average of encryption and decryption time of the script for same images. After gathering the average time for different sized images I also generated graphs like time vs size of the images and rgb histograms of before and after (using Microsoft Excel and Python Matplotlib) to see the overall performance of AES algorithm compared to other algorithm( such as RSA).

# 4 EVIDENCE

## 4.1 Evidence obtained by Farah

#### 4.1.1 *How cryptography is related in complexity theory:*

NP's relevance to cryptography can be considered while studying the cryptographic algorithms. In the context of hacking, reversing a cryptographic algorithm is hard whereas the generation of ciphertext is easy to compute. So the argument lies amongst the problems in NP-Complete If we can prove that the subset of problems P and the subset of problems NP are one and the same then P=NP. Although it seems obvious that a brute force attack against an encryption algorithm is much harder than encrypting a block of plain Image. If there exists a solution that can efficiently reverse a cryptographic algorithm in polynomial time, then the cyber Security would break right in front of us. Complexity theory provides a methodology for analyzing the computational complexity of different cryptographic algorithms. It compares cryptographic algorithms and techniques and determines their security.

#### 4.1.2 *Math behind traditional RSA.*

RSA gets it's security from the difficulty of factoring large numbers.It requires finding two very large integers with a high likelihood of being co-prime.RSA keys are not only asymmetric because one encrypts and the other decrypts, they are also asymmetric because you can derive an RSA public key from the private key, but not the other way around.

- ( **S = Sender**;**R= Receiver**): **R** will choose two random large prime numbers, p and q and compute the product $N = pq$. and then **R** we will generate a random number which is relatively prime with $\phi(N) = .(p-1)(q-1)$
- :Let the number be called as encryption(e). We will choose (e) s.t , $1 < e < \phi(N)$
- : **R** will calculate the modular inverse of e. The calculated inverse will be called as decryption(d). $d = e^{-1} mod((p-1)(q-1))$
- : **R** will make the key (N,k) Public.
- : **S** will then encode the image m as $C := m^k(mod(n))$ and send it to **R**
- : Finally, **R** will decode the image by computing $C^e (= m^{ke}) \equiv m$

Note that d and N are also relatively prime.The numbers e and N are the Public key and the number d is the private key. Let's take an example,
Let's say we want to encrypt an image, m First we have to divide it into numerical blocks let's say $m_1$ which is smaller than n and the encrypted Image(C) which will also be made of similarly sized image blocks, $C_i$. The encryption formula is simply, $C_i = m_i^e mod(N)$
To decrypt the image, we will take each encrypted block $(C_i)$ and compute, $m_i = C_i^d mod(N)$ . Since

$$C_i^d = (m_i^e)^d \tag{1}$$
$$= m_i^{ed} \tag{2}$$
$$= m_i^{k(p-1)(q-1)+1} \tag{3}$$
$$= m_i m_i^{k(p-1)(q-1)} \tag{4}$$
$$= m_i * 1 \tag{5}$$
$$= m_i all(mod(n)) \tag{6}$$

let's look at the proposed architecture of the RSA algorithm

#### 4.1.3 *RSA implementation:*

We have digitized some test images and using PYTHON we obtained a matrix which we have used in the encryption algorithm of RSA cryptosystem with two large

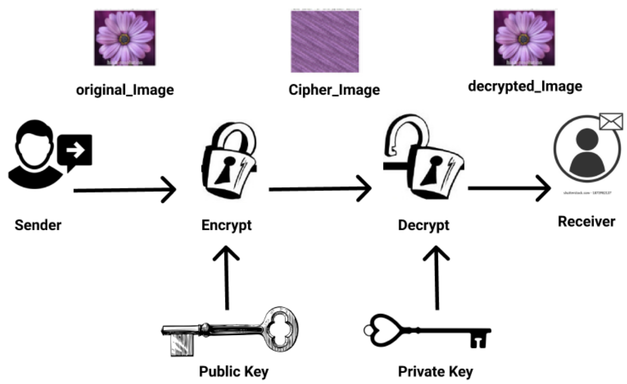


**Figure 1: shows the Image encryption and decryption by RSA**

prime numbers . The result shows that the original images can be encrypted and the decrypted image comes exactly as the original image without any noise even though it was a fairly slow process. One of the important notes is the image should be with the same dimension (n x n image) and the dimension we have tested on in the process is 250 by 250.
Here is the pseudo code flow of the traditional RSA image encryption and decryption algorithm :

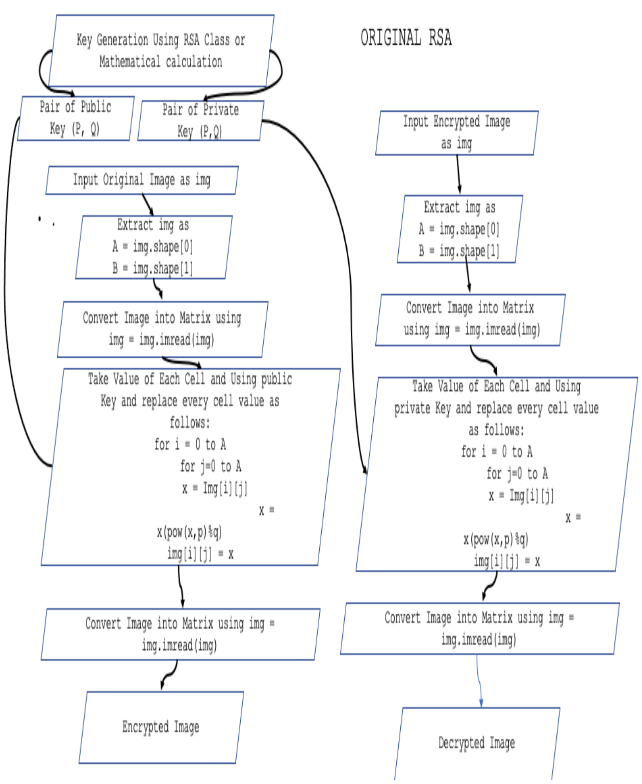


**Figure 2: shows the pseudo code of the encryption and decryption of Images using traditional RSA**

As the RSA works with two types of keys, private and public, so the very first step is to generate the keys. The public key is given to all to encrypt text or image, and the private key is a secure key that is used to decrypt the encrypted text or image. To create keys, we used the RSA class, which has some inbuild function call to export key. First, the private key is exported, and then, based on the private key, the public key is exported.In the encryption part, first, the input image is converted into a matrix form and replaced with the value of each cell of the matrix using the power function with the help of the public, and again the new matrix is converted to an image, and that is the encrypted image.A similar, but process is used to decrypt an image. First, the encrypted image is converted into a matrix form, then the value of each cell has been replaced with the new value using the power function with the help of the private key. Then the new matrix is converted into an image, and that is the encrypted image, same as the original image.

*4.1.4* ***Encryption tables***. Image Encryption in RSA depends on looping through the bits of an image and then bitwise XOR them with the bit and the key. While the decryption relies on reversing the XOR operation. To avoid data loss we will be saving the encrypted image with an enc in front of the name and the decrypted image with a dec in front of the name.
Proposed traditional RSA is tested and implemented using the following values in the tables:

**Table 1: traditional RSA encryption**

| Data Sample | Original Size(KB) | Encryption Time |
|---|---|---|
| originalImage.png | 121.85 | 3.192 |
| flower.png | 67.19 | 3.09 |
| dog.png | 96.46 | 3.872 |
| scene.png | 86.567 | 4.214 |
| amazon.png | 24.874 | 2.748 |
| docs.png | 49.518 | 4.052 |
| $our_{c}ampus.png$ | 146.61 | 4.028 |

**Table 2: Chaos based RSA encryption**

| Data Sample | Original Size(KB) | Encryption Time |
|---|---|---|
| originalImage.png | 121.85 | 1.625 |
| flower.png | 67.19 | 2.377 |
| dog.png | 96.46 | 2.341 |
| scene.png | 86.567 | 2.347 |
| amazon.png | 24.874 | 2.375 |
| docs.png | 49.518 | 2.363 |
| $our_{c}ampus.png$ | 146.61 | 2.377 |

## 4.2 Evidence obtained by Mohammad

*4.2.1* ***Chaos-based RSA over regular RSA***. In the regular RSA system, the RSA algorithm is used to create public and private keys, and text or images are encrypted with the public key. But chaos-based algorithm creates a new initial key based on the ciphertext information, and this key helps to establish a hyperchaotic system equation that calculates the keystream. For encrypting the image, permutation and defusion operations are used, and for decryption, the reverse process is implemented. If we combine the RSA algorithm and chaotic fractional system to encrypt, which employed a fast algorithm and enhance security.

*4.2.2* ***Math behind Chaos-based RSA.*** In this section we have explored an enhanced approach dependent on the RSA asymmetric system which is known as chaos based encryption where the original input images will be encrypted by using the chaos based RSA algorithm. At last the original images are retrieved back from the encrypted image by using the key that is specified during the encryption process for the decryption of the original images. Arnold's transformation is used where an image is converted that is randomized using actual arrangement of pixels. Although, after much iteration, we would get the original image back.Let I be an input image and the size is denoted by N,

$$\left[\begin{cases}A_{m+1}\\ B_{m+1}\end{cases}\right] =$$

$$C\left[\begin{cases}A_m\\ B_m\end{cases}\right](mod(N)) = \left[\begin{cases}1 & i\\ & \\ j & ij+1\end{cases}\right]\left[\begin{cases}A_m\\ B_m\end{cases}\right](mod(N))$$

here i and j represents positive integers and $(A_m, B_m)$ expresses the position of samples in the $NXN$ data like image,hence
$(a_m.b_m) \in \{0; 1; 2, ......N-1\}$
and the $(a_{m+1}, b_{m+1}$ indicates the co ordinates of the image.

*4.2.3* ***Chaos based RSA implementation:*** The chaos based pseudo code is shown below:

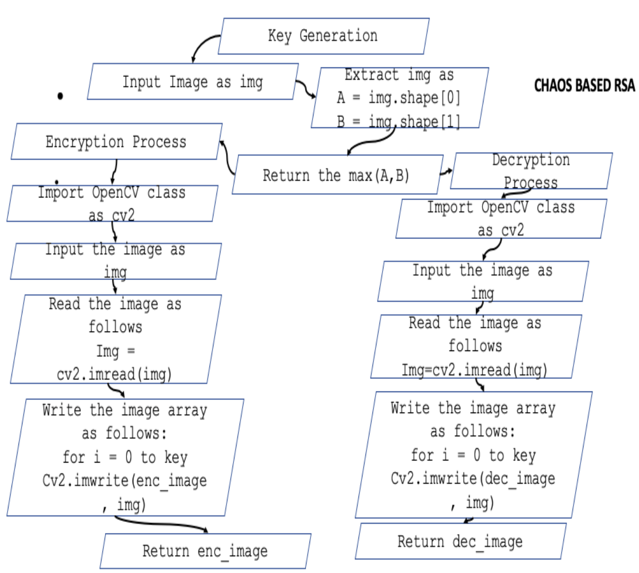


**Figure 3: shows the pseudo code of the encryption and decryption of Images using chaos based RSA traditional RSA**

The chaos-based algorithm uses one single key to encrypt and decrypt the image. The key would be the maximum size of the image of the x-axis or y-axis. In the encryption part, first, the image is read in a matrix form using an inbuilt function from the OpenCV class, and the image is written in the encrypted form using another inbuilt function.In the decryption part, a similar technique is used. First, the encrypted image is read in a matrix form using an inbuild function. Then, the image is written using another inbuild function. And the written image is the decrypted image, the same as the original image.

*4.2.4* ***Decryption tables.*** In the tables 3 and 4 decryption differences of images are provided:

**Table 3: traditional RSA decryption**

| Data Sample | Encrypted Size(KB) | Decryption Time |
|---|---|---|
| originalImage.png | 163 | 13.378 |
| flower.png | 68 | 13.457 |
| dog.png | 167 | 10.461 |
| scene.png | 147 | 12.452 |
| amazon.png | 29 | 12.785 |
| docs.png | 55 | 12.344 |
| $our_campus.png$ | 157 | 14.283 |

**Table 4: Chaos based RSA decryption**

| Data Sample | Encrypted Size(KB) | Decryption Time |
|---|---|---|
| originalImage.png | 182 | 2.057 |
| flower.png | 84 | 2.920 |
| dog.png | 187 | 3.338 |
| scene.png | 181 | 3.546 |
| amazon.png | 45 | 3.257 |
| docs.png | 54 | 3.439 |
| $our_campus.png$ | 170 | 3.643 |

## 4.3 Evidence obtained by Rezwana Kabita

*4.3.1* ***Image Encryption using AES Algorithm.*** AES algorithm is a symmetrical block cipher algorithm which uses the same keys for encryption and decryption of images. AES is a block cipher. It encrypts images in blocks of bits instead of encrypting images bit by bit. So, the first step in encrypting images by AES is dividing the data in 128 blocks and then converts them to cipher images using keys of 128, 192 or 256 bits respectively. AES uses a substitution-permutation with multiple rounds to produce cipher images. The Algorithm works in four steps. In Figure 4 we can see all the steps.

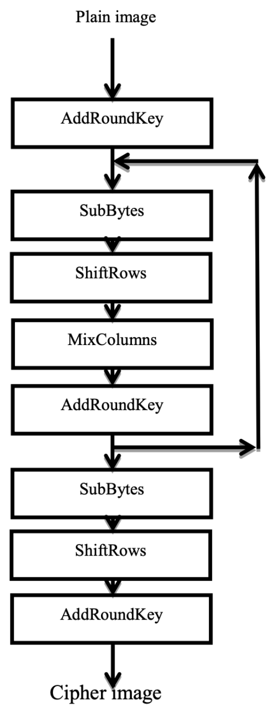


**Figure 4: Steps of Image Encryption using AES Algorithm**

**SubBytes:** At first, The AES algorithm substitutes every byte of the image with a code from a pre-calculated substitution table called S-box.
**ShiftRow:** Then, the AES algorithm shifts all the rows except the first one. For example- the 2nd row shifted by 1 rows to the left, The 3rd row gets shifted by 2 rows to the left and so on.
**MixColumns:** Next comes the columns mix. In this step, The algorithm multiplies each column by a predefined matrix, and then give back a whole new block of image.
**AddRoundKey:** At last, a round key is added to the last resulted data from previous step and then a bit wise XOR operation be performed.
This 4 steps usually performed 10, 12 or 14 times depending the number of keys selected.After all those steps the image becomes very hard to break. We have implemented the AES algorithm using Python Crypto libraries and tried to run different sized images, take the average of the time for different image sizes and compare the results.

**Table 5: Average AES encryption and decryption time**

| Image Size (KB) | Encryption Time (sec) | Decryption Time (sec) |
|---|---|---|
| 8 kb | 0.0031 | 0.00081 |
| 80 kb | 0.005 | 0.001 |
| 1900 kb | 0.013 | 0.006 |
| 2300 kb | 0.016 | 0.02 |
| 3700 kb | 0.0207 | 0.025 |

In table 5, I have listed the average encryption and decryption time for different sized images. As we can see the time of encryption and decryption usually increases with image size.

## 4.4 Evidence obtained by Nour

```
//Encryption
 Input:Image file (gif/bmp/jpg)
Output:Cipher text(Numeric Value)
Method:
Step1: Read Image File
Step2:Covert Image file into
        Sequence of bytes Array
Step3: For i= 0 to Barray.length
         //Bytearray
         Begin
              Flag=0;
               If Barray[i] <0 then
                  Begin
                     pos=-Barray[i];
                     Flag=1;
                 end
               else
                   pos=Barray[i];
          Rarray[i]=MRarray[pos];
 /*MagicRectagle array and  Result
array*/
 Step 4:  Encrypt using Algorithm
          If Flag=1 then
            Cipher[i]=-Cipher[i];
              //Cipher array
          End
 Step 5: Produce Cipher Text
```

**Figure 5: AES Encryption Pseudo Code**

#### 4.4.1 *AES Encryption Code.*

The input for this algorithm is an image of any size and the output is the cipher text, or the numeric value. The first step is to read the image file that was uploaded by the user. Then the image file is converted into a sequence of bytes and put into an array. After that a for loop is created to iterate over the array and there is an if statement inside the loop. The if statement checks if the byte is less than 0 then it moves down the array and sets the flag equal to 1 otherwise it stays at the same position. Then new array that was created from the for loop is assigned to another array called Rarray. Second to last step is the actual encryption part where if the flag is equal to 1 then it adds it to the cipher array. Lastly, the cipher text is outputted for the user.

#### 4.4.2 *AES Decryption Code.*

The input for this algorithm is the cipher text and the output would be the image file that was originally inputted by the user. First the cipher text that was uploaded by the user is read. Then a for loop is created in order to iterative over the cipher text array and there is an if statement inside the loop. The if statement checks if the cipher text is less than 0 and if it is then it moved down the array and sets the flag to 1. After that the decryption takes place by having an if statement that says if the the flag is 1 then it move along the array that was created in the step before. Second to last step is converting the byte array into the original image. Lastly, the original image is outputted to the user.

#### 4.4.3 *Image decryption using AES algorithm.*

The decryption part of this algorithm is essentially the reverse of the encryption.

```
//Decryption
Input:Cipher Text (Numeric Value)
Output: Image file (gif/bmp/jpg )
Method:
 Step1: Read Cipher text
 Step2: For i= 0 to Cipher.length
//cipher array
            Begin
               Flag=0;
               If Cipher[i] <0 then
               Begin
                  Cipher[i]=-Cipher[i];
                   Flag=1;
               End
 Step 3:  Decrypt using Algorithm
           Pos =Marray[i];
           //MagicRectaglearray
            If   Flag=1 then
                  Barray[i]=-Pos
 Step 4: Convert Byte Array into
           Image
 Step 5: Produce original Image
```

**Figure 6: AES Decryption Pseudo Code**

AES is implemented using the method of symmetric cryptography. In other words, the same key is used for both data encryption and decryption. Many other algorithms use asymmetric encryption which means that both a public and private key is required. Each step is already explained in the previous section that talks about encryption.
1. Add the round key
2. Mix the columns
3. Shift the rows
4. Byte Substitution
Figure 6 is a visual representation of the steps it take to decrypt an image using AES.

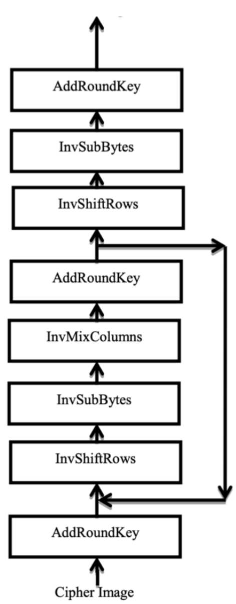


**Figure 7: AES Image Decryption Steps**

# 5 RESULTS

## 5.1 Results Analysis obtained by Farah and Mohammad

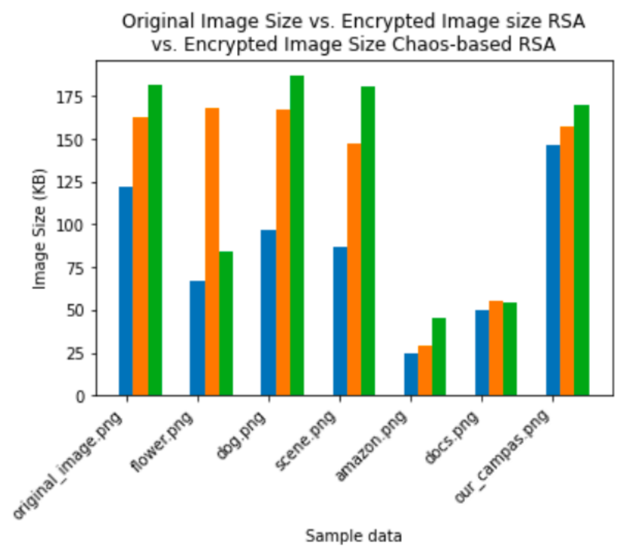


**Figure 8: shows the original image vs encrypted image vs chaos based encrypted image**

The following graph shows the comparison between the original image size and encrypted image size using the regular RSA algorithm and the Chaos-based RSA algorithm. When the image is encrypted, the size of the image gets larger. And overall, the Chaos-based RSA takes more space than the regular RSA.

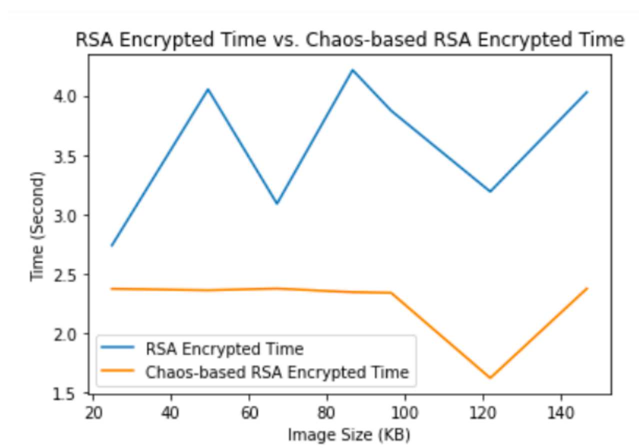


**Figure 9: shows the Encryption Time comparison**

The above graph shows the encrypted time of the regular RSA method and the Chaos-based RSA method. For this, experiment, we used the same images with the same size (250 x 250) and dimension (2D). And we got the result that the Chaos-based RSA algorithm takes much less time than regular RSA. Than means that Chaos-based RSA is faster than the regular RSA.

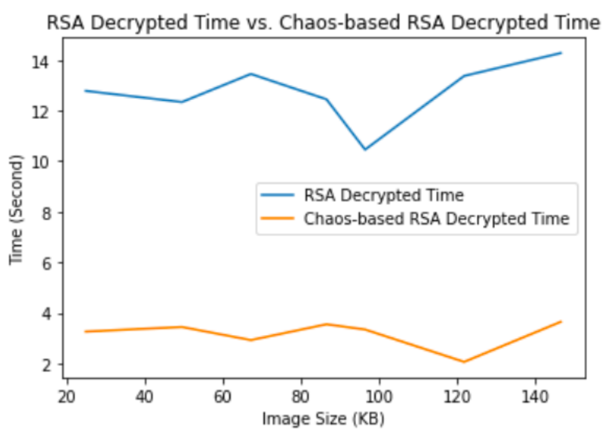


**Figure 10: shows the decrypted time comparison**

The graph shows the decrypted time of the regular RSA method and the Chaos-based RSA method. For this, experiment, we used the same encrypted images with the same size (250 x 250) and dimension (2D). And we got the result that the Chaos-based RSA algorithm takes much less time than regular RSA. That means the Chaos-based RSA algorithm is faster in decryption.

## 5.2 Results Analysis obtained by Nour and Rezwana

When we tested our AES model we decided to use different size images in order to get more comprehensive results. In Figure 11 we will see how the time changes with the size of the images.

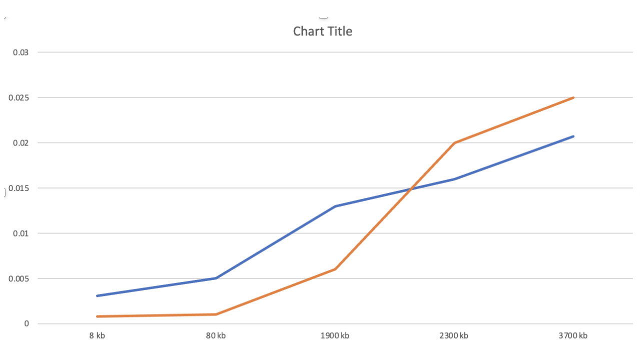


**Figure 11: Time vs size using AES algorithm**

In some scenarios we can see that, decryption takes less time than encryption. That's because in the first step of decryption by AES algorithm we have to calculate the inverse of the XOR operation. The inverse of a XOR value is the XOR value itself. Which makes it easier decryption of an image or data. Which leads to a shorter time for decryption.

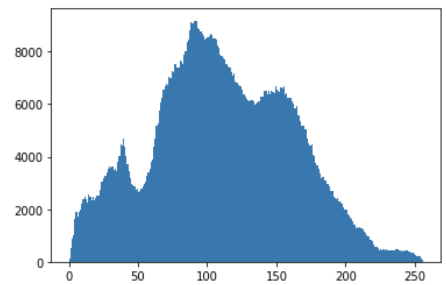

**Figure 12: rgb Histogram AES algorithm**

Figure 12 is a graph histogram of the original image, we created this graph before performing AES algorithm.

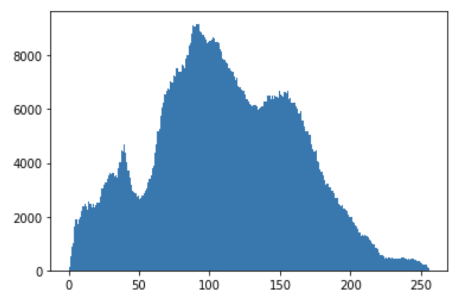

**Figure 13: rgb Histogram AES algorithm**

**Table 6: Image size before and after performing AES algorithm**

| Image Size before Encryption | Image Size after Encryption |
|---|---|
| 8 kb | 8 kb |
| 80 kb | 79.8 kb |
| 1900 kb | 1899.1 kb |
| 2300 kb | 2299 kb |
| 3700 kb | 3699.5 kb |

In Figure 13, We can see the rgb histogram graph after we get back the decrypted images. We can see that it gives us a almost same graph.
In table 6 we can see the actual size of the returned image after decryption. We can see that we lost few bytes of data while decrypting the image. This could mean that we loose any very little to no data while performing the encryption and decryption using AES. it is very safe to use AES algroithm for encryption and decryption if we want to perform a fast, safe,and secure encryption

### 5.3 Results Discussion

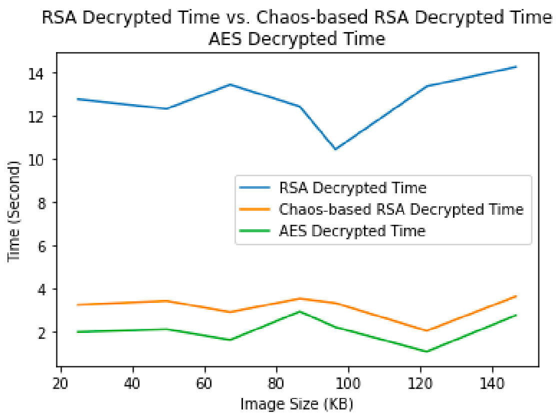


**Figure 14: shows the Encryption Time comparison**

Now that we have seen the results analyzed by our team members. We can move on to discuss the combined results. After the analysis of different dimensions of our three algorithms, we plotted a graph of decryption time, taken from the algorithms. The above graph shows that traditional RSA algorithm takes more time than the Chaos-based algorithm. According to the graph, AES is the faster, and more robust algorithm in terms of decryption. On the other hand, AES uses the same key for encryption and decryption, therefore, and the key is known to the sender and receiver. Bur in terms of RSA, two keys, public and private are used, that is why it holds more security than the AES algorithm. Overall, although the RSA algorithm is more secure, AES is faster.

## 6 CONCLUSION

Digital images play an important role in multimedia technology. Therefore it is necessary to incur the integrity and confidentiality of the digital image that is being transmitted. With time, the data transfer over the internet is increasing as well as cyber-attacks, and therefore, data security becomes more concerned for sensitive data. Encryption, decryption, and secure data transfer come with a matter of cost. So, we need to research more about the different algorithms to find the optimal solution based on the cost and security in the cryptographic sector.Digital images are comparatively less sensitive than data because the changes made in the pixels don't drastically change the entire image, but it is more prone to attackers. The algorithms discussed in the report have some considerable weaknesses. We have performed the algorithms on several images. The traditional RSA provides slower encryption-decryption time compared to Chaos based RSA and AES from which we can demonstrate the security that RSA displays. The results of the chaos based RSA clarify that decrypted images are close to the ideal, and hence, the proposed algorithm is secure against the entropy attacks.On the other hand the AES has the fasting execution time and has shown efficiency which makes the encrypted images by the algorithm resistant to the attacks. Hence We can come to a solution that whenever the security is more concerned like a military operation, critical research, autonomous weapon, etc. RSA is the best option to use. On the other hand, if time or space are concerned, we can use AES algorithm as this is faster enough.

## 7 REFERENCES


1.Chaos Based Image Encryption - Researchgate.net. https://www.researchgate.net/publication/322473608_Chaos_Based_Image_Encryption.

2.*Moghaddam, F.F., Alrshdan, M., amp; Karim, O.(n.d.).(PDF)a* hybrid encryption algorithm based on RSA Small-e ... Retrieved December 23, 2021, from https://www.researchgate.net/publication/253328671 _A_Hybrid_Encryption_Algorithm_Based_on_RSA_Small-e_and_Efficient-RSA_for_Cloud_Computing_Environments

3. Ghoradkar, S., amp; Shinde, A. (n.d.). Download limit exceeded eseerx.ist.psu.edu. Retrieved December 23, 2021, from https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.740.4766amp;rep=rep1amp;type=pdf

4.*Ghosh, A.(n.d.).(PDF)comparisonofencryptionalgorithms : AES,* blowfish ... Retrieved December 23, 2021, from https://www.rese-archgate.net/publication/342764235_Comparison_of_Encrypti-on_Algorithms_AES_Blowfish_and_Twofish_for_Security_of_Wireless_Networks

5. Suresh, G. B., amp; Mathivanan, V. (n.d.). Chaos based image encryption-researchgate.net. Retrieved December 23, 2021, from https://www.researchgate.net/publication/322473608_Chaos_Based_Image_Encryption/fulltext/5ff3a69c92851c13feeb3ce8/Chaos-Based-Image-Encryption.pdf

6. Ratna, A., Surya, F., Husna, D., amp; Purnama, I. (n.d.). Chaos -based image encryption using Arnold's Cat Map ... Retri -eved December 23, 2021, from https://www.researchgate.ne -t/publication/348730660_Chaos_Based_Image_Encryption_Using_Arnold's_Cat_Map_Confusion_and_Henon_Map_Diff -usion

7.Aumasson, J.-P. (2018). Chapter 1 and 10. In Serious cryptogr -aphy: A practical introduction to modern encryption. ess -ay, No Starch Press.

8.Ye, G., Faculty of Mathematics and Computer Science, Jiao, K., Wu, H., Pan, C., Huang, X., Corresponding author.Faculty of Mathematics and Computer Science, Alvarez, Farah, B., Gan, Ghebleh, Gong, Hua, Landir, Li, Liu, Lu, Luo, Mus -anna, … Chen, B. (n.d.). An asymmetric image encryption algorithm based on a fractional-order chaotic system and the RSA public-key cryptosystem. International Journal of Bifurcation and Chaos. Retrieved December 23, 2021, from https://www.worldscientific.com/doi/abs/10.1142/S0218127420502338