Enhancing Image Encryption Time using Hash Algorithms with Well-known Standards

Mohammed Omari*, Mohammed Echerif Abbou and Omar Guerrout
LDDI Laboratory, Mathematics and Computer Science Department
University of Adrar
Adrar 01000, Algeria
*Corresponding author's email: omarialgeria [AT] gmail.com

Abstract—**The security of images nowadays is one of the vital issues of the exponentially data growing space in the Internet. To do so, images are usually encrypted in order to hide the visual information when transmitted over an unsecure channel. The latter can be subject to many attacks and thus a reliable as well as an efficient mechanism has to be deployed for such task. For the sake of protecting data while being transmitted, multiple encryption techniques have been implemented some of them are secure but time consuming, while some others are efficient and fast. In our report, we focus on image encryption, and we introduce multiple techniques based on AES, hash functions, and block selection variations. Experiments showed very promising results compared to the original AES in terms of encryption/decryption time while maintaining almost the same level of security.**

Keyword: Image Encryption; AES; Hash Algorithms

I. INTRODUCTION

Due to the quick evolution of connection technologies, data such as images must be secured when transferred from one site to another. One famous way to perform security is through encryption which means changing information or data from its original form to another so it cannot be recognized or understood. One of the types of data that requires to be secured while travelling through the Internet is an image where it is used almost everywhere. Image encryption can be implemented using a variety of encryption methods such as Rivest-Shamir-Adleman (RSA), advanced encryption standard (AES) and data encryption standard (DES). Yet, some of these techniques may not meet the security requirements or are very time consuming. The latter is considered a very critical factor in the multimedia domain where images are videos are required to be downloaded and played as fast as possible.

Our objective is this work study is to propose a hybrid model between secured encryption techniques and fast security mechanisms like hash functions in order to enhance the encryption/decryption time for images while maintaining high level of security.

This paper is organized over four sections as follows. In the second Section, we briefly present general information about three related works. Our proposed methods are presented in Section Three. The fourth Section presents experimental results and a comparative study of some well-known encryption techniques as well as our proposed methods. Section five is the conclusion.

II. RELATED WORKS

Encryption in general is the process of encoding data and securing it from unauthorized access, more specifically in an encryption scheme the data or the message referred to as Plain-text will be encrypted using an encryption algorithm referred to as Cipher, to produce the encrypted message /data referred to as cipher-text that can be read only by decrypting it, to further secure the data most encryption ciphers use a key that can be generated by an algorithm, this key will allow the recipient to decrypt the encrypted data.

Image encryption on the other hand is not much different from normal data encryption since an image is just a series of bytes, it can be implemented by altered methods of encryption such as RSA, DES, and AES.

A. Image Encryption Method based on Using Least Square Error Techniques at the Decryption Stage

This algorithm [1] which was proposed by Mahmood Al-khassaweneh uses random vectors to convert every column from the original image A into their corresponding columns in the encrypted image B.

If the vectors X(i) represents a randomly generated normal noise, the encrypted image, B, will be a noisy image that has no visual correlation with the original image.

Moreover, the encrypted image, B, does not represent a shuffled version of A, therefore, the image B by itself tells no information about the original image. Thus, in order to recover the original image from B, the original vectors X(i) should be known.

The solution of such a system represents the first row of the original image A, similar steps can be applied to find the rest of rows for 2, 3 ,…..M. Thus, it turns out that the decrypted image is the solution of a set of linear equations that best fits the encrypted image and the encrypting vectors in the least square sense. The system can be solved using numerical analysis techniques such as Gaussian elimination method or iterative methods for linear equations.

B. Value Transformation and Random Permutation-Based Colored Image Encryption Technique

This technique [2] proposed by Mahmood Al-Khassaweneh and Shefa Tawalbeh is based on transformation of pixels values and the shuffling of their locations, technique consists of two stages; transformation of pixels values and shuffling pixels locations. The encryption key is generated during shuffling stage. This generated key is used to shuffle the columns of the image at the first step of shuffling stage and to shuffle the rows of the resulted image at the second stage. The encryption key was used in the decryption stage to get the original image. In decryption stage, the bits of each pixel are rearranged to get the original value of each pixel.

1) The transformation stage

In this stage, the values of pixels are transformed to new values. The transformation is based on masking specific bits in every pixel. The masking key is embedded with the transformed pixel. Mask operation is performed by performing logical AND operation between each pixel value and 85, such that every other bit of the original pixel remains unchanged, i.e., {b0, b2, b4, b6} remains unchanged. Similarly, another logical AND operation is performed between 170 and each pixel value such that bits {b1, b3, b5, b7} remains unchanged. The resultant pixel bits are rearranged to produce the new transformed pixel value. The shuffling stage

This stage aims to change the pixels lactations to get new shuffled image. It performs shuffling operation using random permutation. This stage consists of two steps; the first step shuffles the columns of the transformed image and the second step shuffles the rows of the already shuffled image. The encryption key is generated randomly and it is used in both steps as encryption key.

In the proposed method, the permutation operation is performed based on random line permutation. An encryption key is generated randomly which will be used to shuffle the columns and the rows of the transformed image respectively.

Suppose we have an image with size equals to N × N, where N is the number of rows or columns of the image. The key is drawn randomly with values between 1 and N. The columns and the rows of the image are reordered according to the key.

2) The decryption phase

The proposed method needs two stages to decrypt the image. The first stage, reshuffled the image to get the transformed image, while, the second stage uses the transformation arithmetic to get the original pixels values of the original image.

C. Preserving the confidentiality of digital images using a chaotic encryption scheme

In this technique [4] proposed by Jolfaei et. al., the cipher utilizes a discretized Chebyshev map and a unit step function to improve the security of the HC image encryption scheme. To elaborate the steps of the encryption algorithm, the Chebyshev map and the discretization process are firstly detailed as follows.

1) Chebyshev map

A Chebyshev map is a typical invertible iterated map that generates orthogonal real-valued sequences. Due to nonlinear properties, chaotic Chebyshev mapping has been proposed as a method of generating pseudorandom sequences.

2) Discretization of the Chebyshev map

Data discretization is a frequently used technique in computer science, statistics and cryptographic applications. Cryptographic algorithms are mathematical structures that are fundamentally discrete rather than continuous. Therefore, the raw real-valued sequences generated by chaotic systems could not be directly used in the cryptographic algorithms. To this end, the discretization process makes the real-valued sequences workable in a finite precision domain. A unit step function, can be utilized to obtain binary sequences from a chaotic real-valued sequence generated by a Chebyshev map. The unit step function is defined as:

$$\eta_c(x) = \begin{cases} 0 & x < c, \\ 1 & x \geq c. \end{cases}$$

The threshold c is a variable equal to the median value of the chaotic real-valued sequence. For any real valued input, this function generates a binary sequence namely, a Chebyshev binary sequence.

III. POPOSED METHODS

In order to enhance the encryption/decryption runtime, we propose the use of AES encryption on some parts of the image while using other simple encryptions with others.

A. Modulo-skipping method

In this method, an image is divided into blocks, where the first block is summed up modulo certain base (S), and the result sets the skipping gap between the first block and the next block to be encrypted. All blocks between the first one and the offset are not encrypted. Figure 1 describes the flowchart of the current method.

B. Modulo-XOR method

In this method, as the previous technique, an image is also divided into blocks, where the first block is summed up modulo certain base (S), and the result sets the skipping gap between the first block and the next block to be encrypted with AES. However, the blocks between the first one and the offset are encrypted using xor operation with CBC mode to generate stream of keys. Figure 2 describes the flowchart of the current method.

C. Modulo-SHA256 and Modulo-MD5 methods

In this method, as the previous technique, an image is also divided into blocks, where the first block is summed up modulo certain base (S), and the result sets the skipping gap between the first block and the next block to be encrypted with AES. However, the blocks between the first one and the offset are encrypted using xor operation with CBC mode to generate

stream of keys. Figure 3 describes the flowchart of the current method.

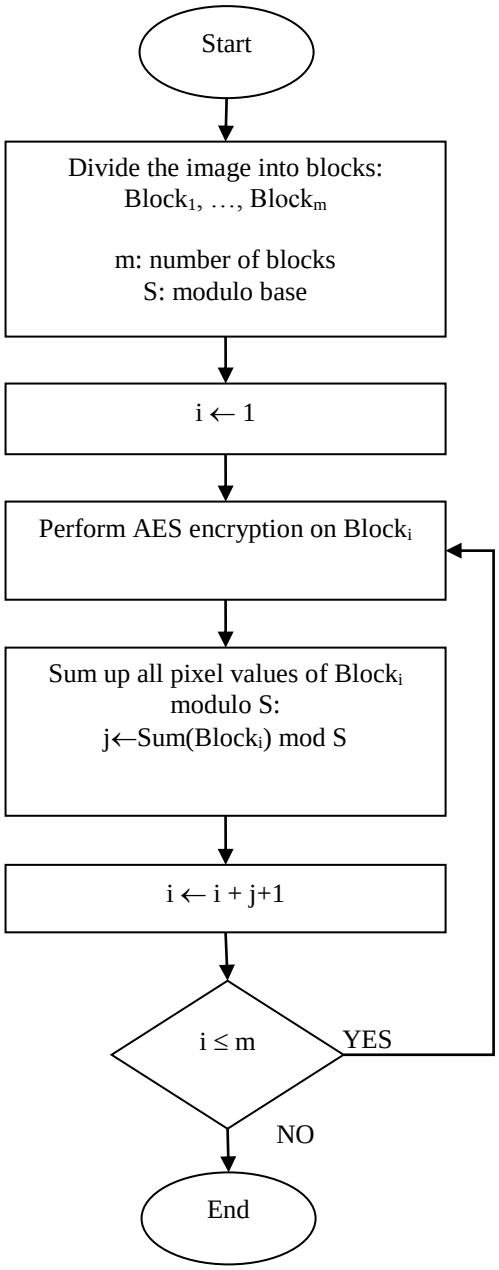


Figure 1. Flowchart of the Modulo-skipping technique.

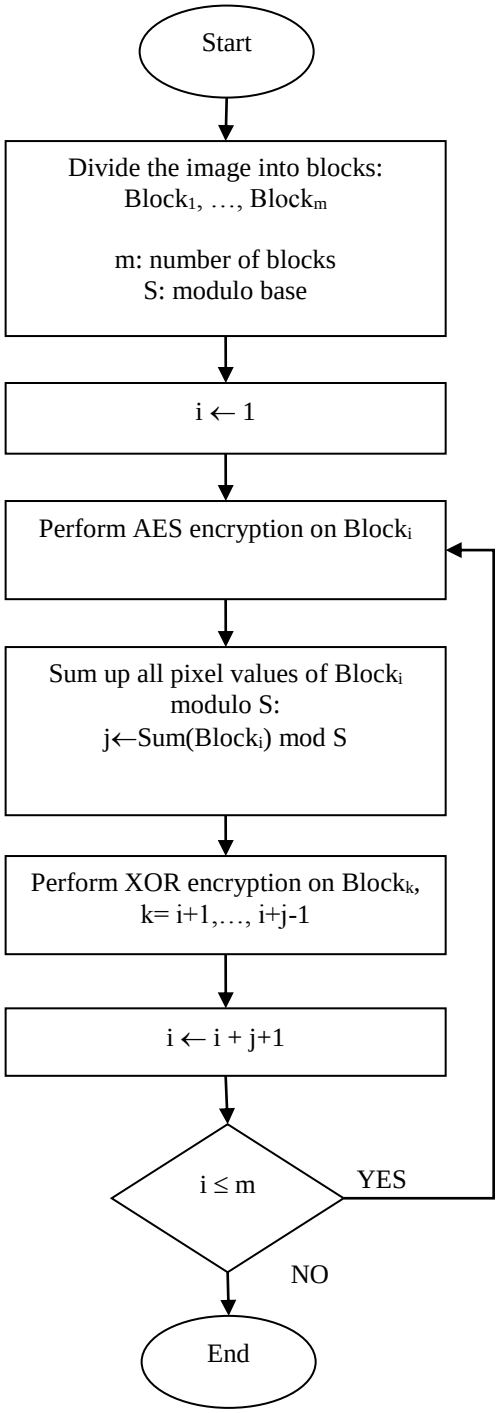


Figure 2. Flowchart of the Modulo-skipping technique.

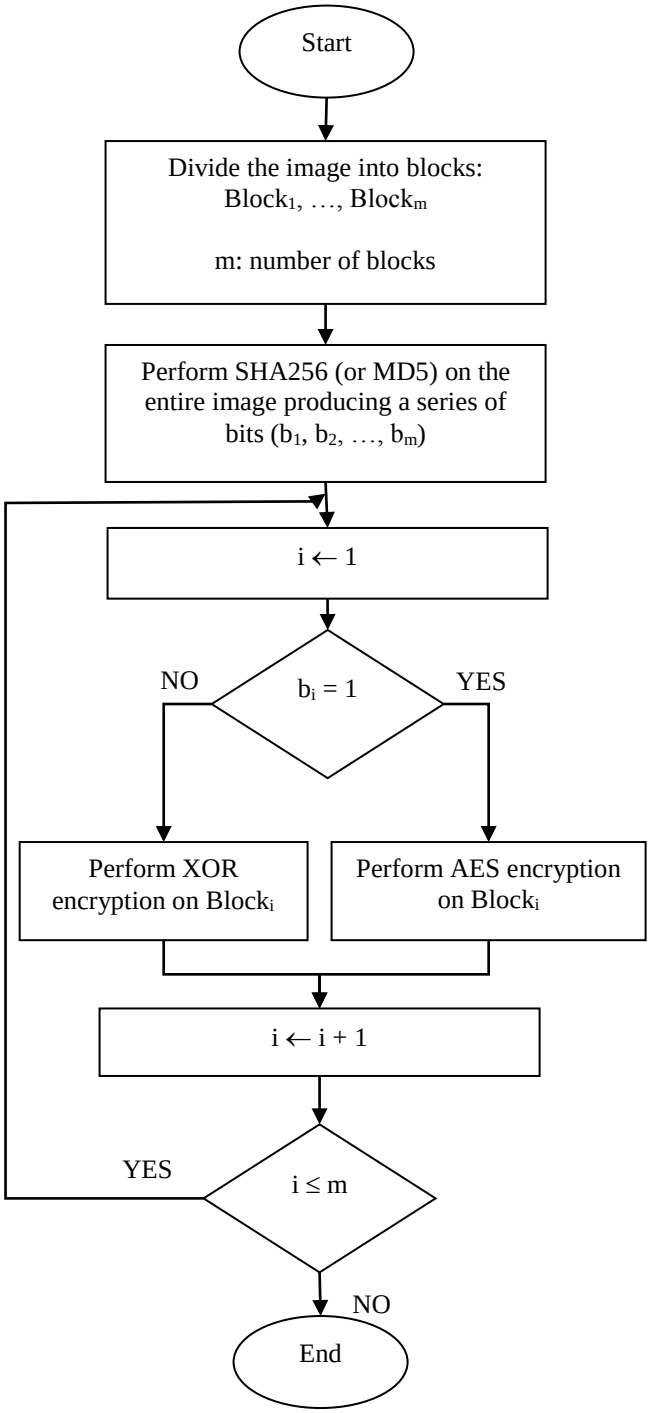


Figure 3. Flowchart of the SHA256 (or MD5) technique.

IV. Experiment, Results and Analysis

A. *Used software and hardware:*

For the implementation side of this project we used Microsoft Visual studio 2017 community edition for writing the encryption algorithms in conjunction with MATLAB r2017a for testing the results.

Microsoft visual studio is a cross platform development environment, that uses C, C+, C++, C# as its main programming languages.

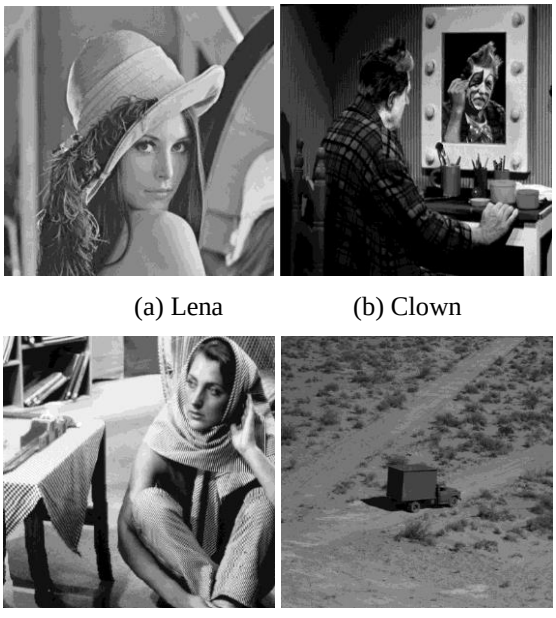
MATLAB is a development platform that uses a matrix-based programming language, which is best suited for image processing.

For the test-bench we used an MSI GE60 laptop that has the following specs:

Core i7 4710hq 2.5GH, 16gb of ram,1000gb of storage

B. *Test images*

To test our technique we used four reference images Lena, Barbra, Truck, Clown, all of them are 512*512 in size 8-bit grayscale, bitmap(bmp) images (Figure 4) [3]. The reason behind choosing these samples is that they contains a nice mixture of detail, flat regions, shading, and texture that do a good job of testing various image processing algorithms [1].



(a) Lena (b) Clown

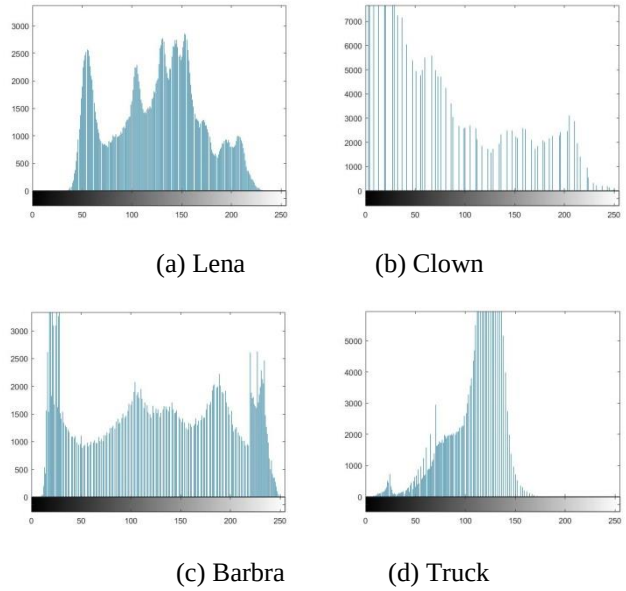
(c) Barbra (d) Truck

Figure 4. Test Images

C. *Analysis of the original images*

The histogram of an image refers to a graph of the pixel intensity values (Figure 5). The histogram is a graph showing the number of pixels in an image at different intensity values found in the image. In an 8-bit grayscale image, there are 256 different possible intensities, and so the histogram will display 256 numbers showing the distribution of pixels amongst those

grayscale values. For a good encryption, the distribution of gray scales in the encrypted image should be uniform.



(a) Lena (b) Clown

(c) Barbra (d) Truck

Figure 5. Histograms of the original images.

In an image data, each pixel is highly correlated with its adjacent pixels. An ideal encryption algorithm should produce cipher-images with no such correlation in the adjacent pixels, meaning each pair of adjacent pixels should not have same or similar values [1].

We use a scatter plot for this analysis it gives us a visual representation on this correlation values which will help us analyze the images. In all correlation analysis we have performed, from each image we extracted pixels of different values P(x,y) = 0..255, and compared it horizontally with the adjacent pixel P(x+1,y) and vertically with P(x,y+1).

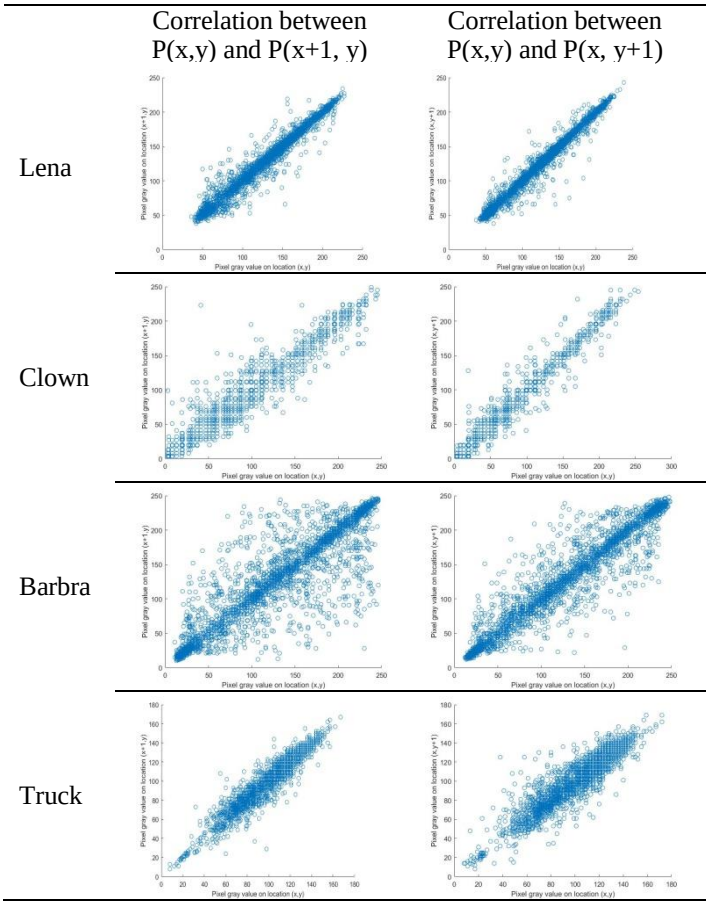
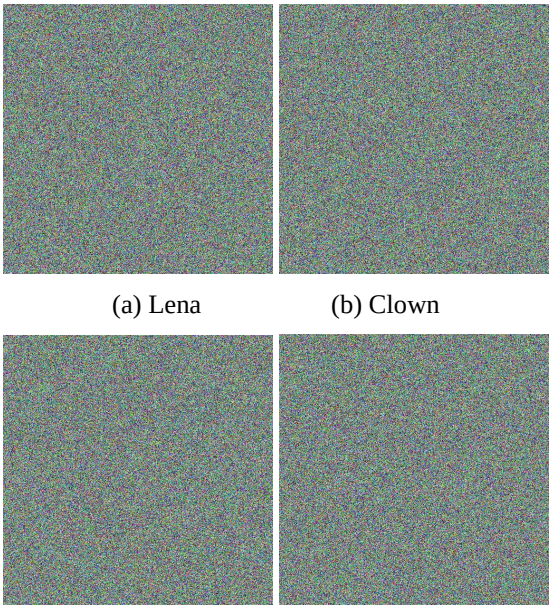


Figure 6. Correlation analysis of the plain test images..

Here we tested the pixel correlation of plain images: Lena, Clown, Barbra, and Truck. Figure 6 showed that neighboring pixels in the plain-image are correlated too much: As we see most of the values are positioned diagonally meaning that each two adjacent pixels both vertically and horizontally have similar values.

D. Results of the AES encryption technique

For comparison purposes, the first encryption we tested is AES encryption technique. Figure 7 shows the encrypted image results:

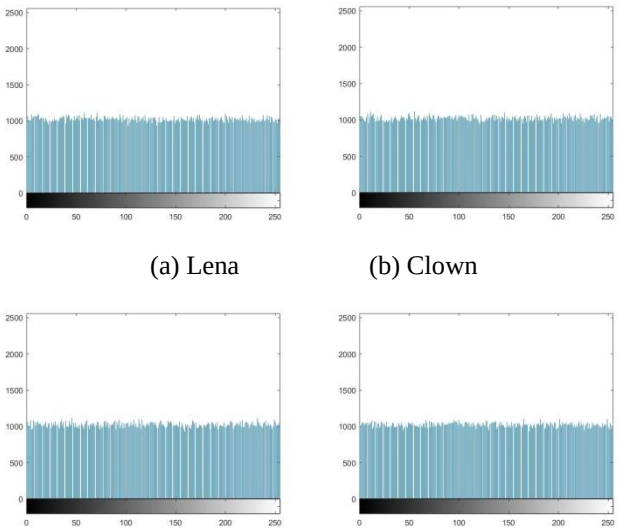


(a) Lena (b) Clown

(c) Barbra (d) Truck

Figure 7. Image Results of AES encryption.

As we can see there is no resemblance between the original images and the encrypted ones, as expected from AES which represent the standard for our comparison.



(a) Lena (b) Clown

(c) Barbra (d) Truck

Figure 8. Histograms of the AES encrypted images.

By encrypting the images using AES, Figure 8 shows that the distribution of gray scales in the encrypted image is perfectly uniform resulting in an even histogram for each image. This result was expected since we are encrypting the entire image and not just a part of it.

TABLE I. PERFORMANCE RESULTS OF AES TECHNIQUE

Image	Encryption Ratio (%)	Encryption Duration (s)	Decryption Duration (s)
Lena	100	70,13	70,03
Clown	100	69,89	70,07
Barbra	100	70,19	69,67
Truck	100	71,39	72,63

The performance results of the AES technique as shown above are very similar across the test images with the average encryption/decryption being 70 seconds, which is understandable since we are encrypting the entire image.

Next we test the correlation of the AES encrypted versions of the images. The results are shown in Figure 9.

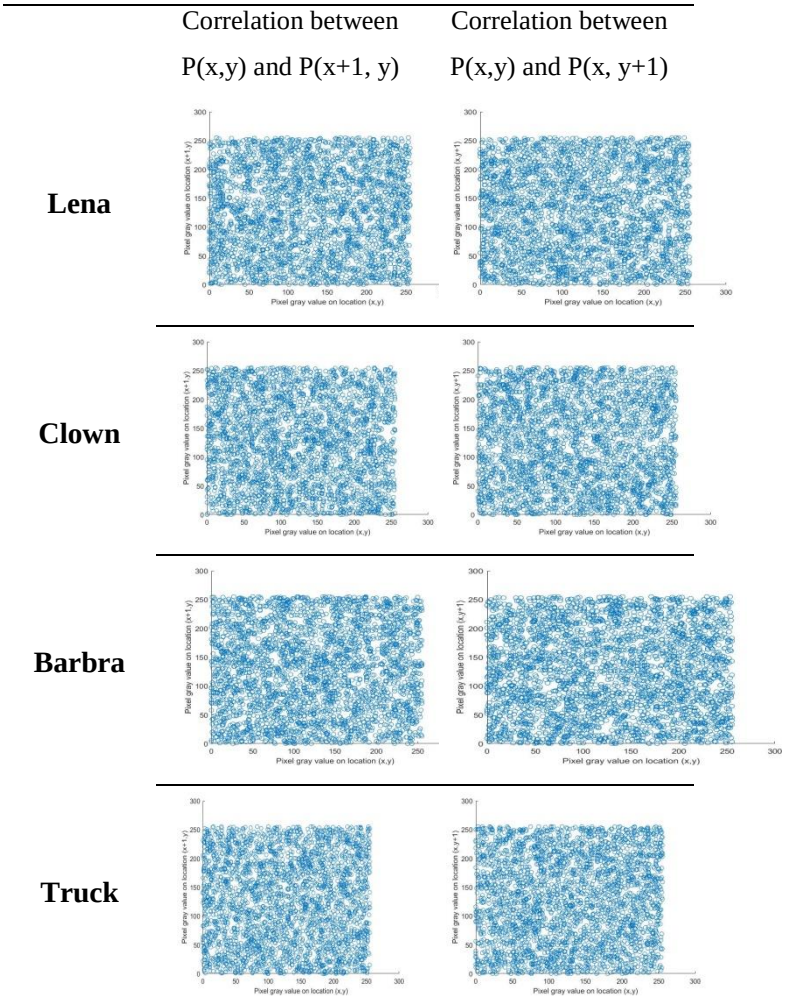


Figure 9. Correlation analysis of the encrypted images using AES.

We tested the pixel correlation of the AES encrypted versions of Lena, Clown, Barbra and Truck. Figure 9 showed

no significant correlation among neighboring pixels in the encrypted images. In fact, the values were distributed evenly, not showing any distribution pattern.

E. Results of the Modulo-Skipping Technique

For this technique instead of encrypting the entire image we only encrypt selected blocks resulting from a Modulo hash function.

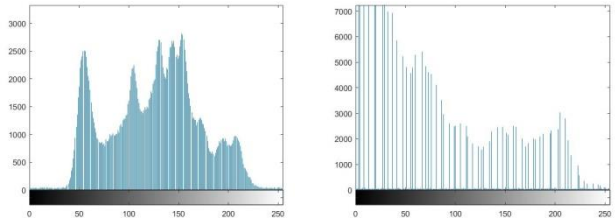
We select the value of the first pixel (v) of a block and apply the Modulo operation it with different values (S) ranging from 2 to 255. We use the result as an indicator on how many blocks we are going to skip, and therefore selecting the best base (S) for the Modulo operation. In general, setting a low (S) value means that more data are to be encrypted. On the other hand a high value of S results in less encrypted data. In our experimentations, we tested the encryption with three S values: 2, 5, and 55.

(a) Lena (b) Clown

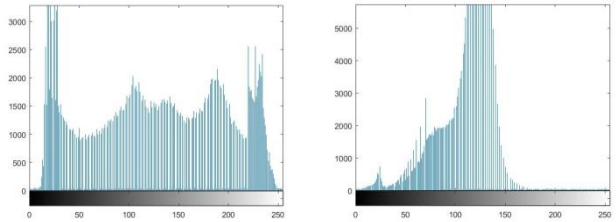
(c) Barbra (d) Truck

Figure 10. Image results of Modulo-Skipping technique, S=55.

Figure 10 shows that the resulting images are still recognizable with only a slight change in certain areas due to the high modulo base causing most parts of the image to stay unencrypted.



(a) Lena (b) Clown



(c) Barbra (d) Truck

Figure 11. Histograms of image results of Modulo-Skipping technique, S=55.

Figure 11 shows that the histograms are barely different than those of the original images. Most blocks of the images were not encrypted because of the high modulo base we selected.

We tested the pixel correlation of the encrypted images using the modulo-skipping technique with S=55. Results in Figure 12 show a little change in the correlation plot compared to the plain ones. Most of the plot values were positioned diagonally indicating a high correlation between the adjacent pixels in both the horizontal and the vertical axes. This indicates that a minor change happened on the images but they are still recognizable.

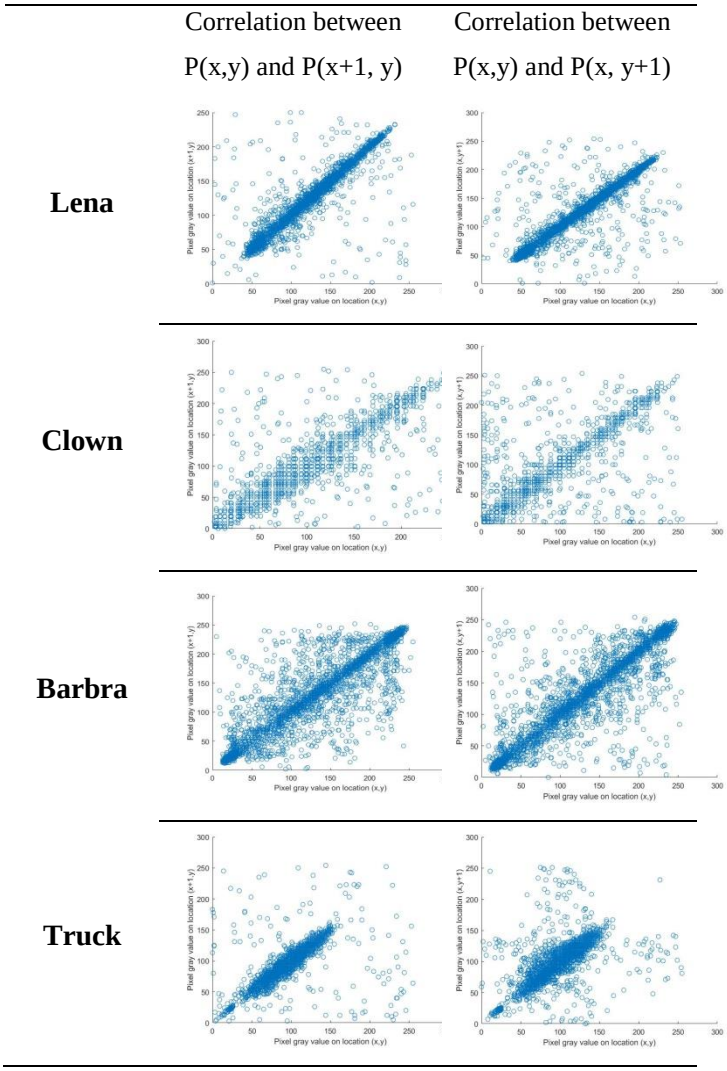


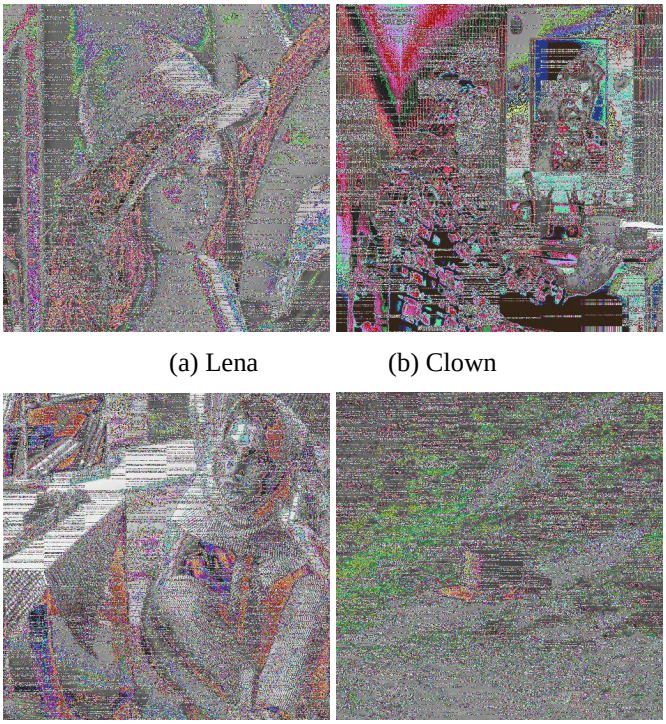
Figure 12. Correlation analysis of the Modulo-Skipping technique, value S=55.

TABLE II. PERFORMANCE RESULTS OF MODULO-SKIPPING TECHNIQUE, S=55

Image	Encryption ratio(%)	Improvement compared to AES	Encryption Duration (s)	Decryption Duration (s)
Lena	3.67%	96,3%	2,66	2,57
Clown	5.17%	94,5%	4,19	3,59
Barbra	3.72%	96,1%	2,91	2,65
Truck	4.45%	94,9%	4,03	3,11

The performance results in Table 3.2 showed a significant improvement in encryption /decryption duration with around 95% less time compared to the AES technique. However this improvement is not justified by the security standards based on the corresponding histograms and correlation plots.

The next step was to lower the modulo base to let more data to be encrypted. Figure 13 shows the encryption results with S=5:

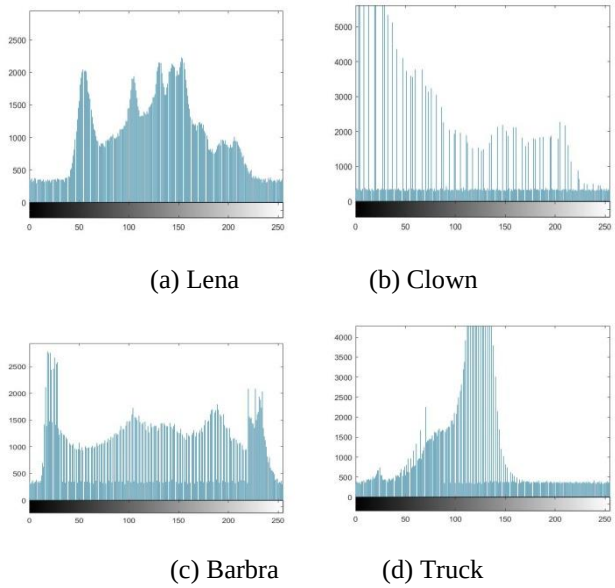


(a) Lena (b) Clown

(c) Barbra (d) Truck

Figure 13. Image results of Modulo-Skipping technique, S=5.

Figure 13 shows that the change is much noticeable: Truck is less visually recognizable compared to Lena, Clown and Barbra.



(a) Lena (b) Clown

(c) Barbra (d) Truck

Figure 14. Histograms of image results of Modulo-Skipping technique, S=5.

In Figure 14, we noticed the change on the values showing a much better distribution of gray levels causing the histograms to become more uniformed than the previous ones.

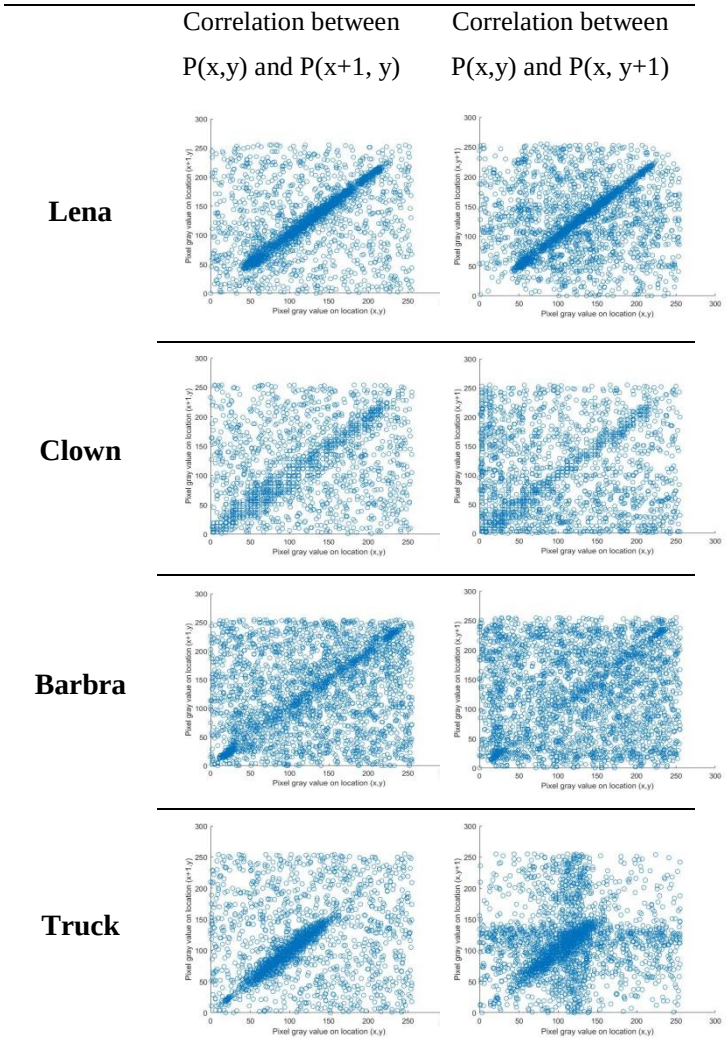


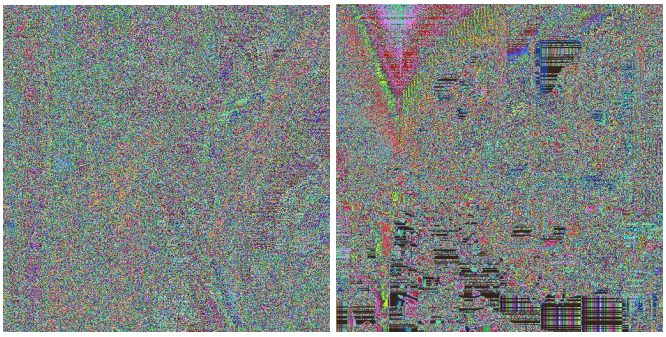
Figure 15. Correlation analysis of the Modulo-Skipping technique, value S=5.

In Figure 15, we can see a change in the distribution of the values across all the test images. However, there is still focused areas which are positioned diagonally. This indicates that there is still a correlation between adjacent pixels in both the vertical and horizontal directions.

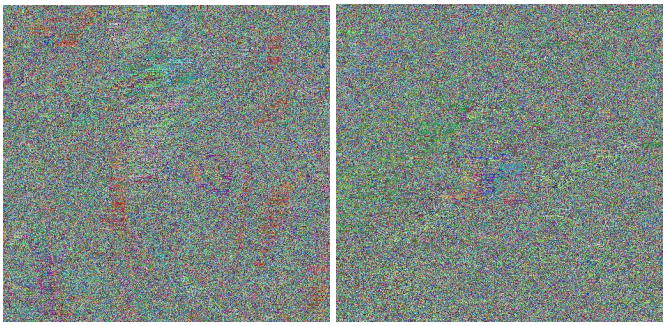
TABLE III. PERFORMANCE RESULTS OF MODULO-SKIPPING TECHNIQUE, S=5.

Image	Encryption Ratio (%)	Time Improvement compared to AES	Encryption Duration (s)	Decryption Duration (s)
Lena	33.56%	65,8%	24,84	23,94
Clown	32.68%	65.4%	25,19	23,27
Barbra	33.51%	65,3%	24,91	23,64
Truck	35.89%	64,5%	26,03	25,09

The next step was to lower the modulo base to let more data to be encrypted. The next figure shows the encryption results with S=2:



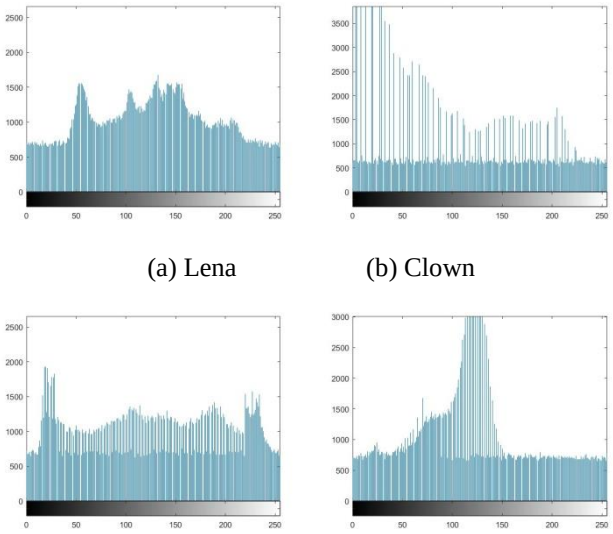
(a) Lena (b) Clown



(c) Barbra (d) Truck

Figure 16. Image results of Modulo-Skipping technique, S=2.

Figure 16 shows that the visual recognition test is passed with all four images not being recognized.



(a) Lena (b) Clown

(c) Barbra (d) Truck

Figure 17. Histograms of image results of Modulo-Skipping technique, S=2.

Figure 17 shows that the histograms are fairly uniform, with most the values being evenly distributed, especially with Lena and Barbra, with the exception of some values that are still spiking due not being encrypted.

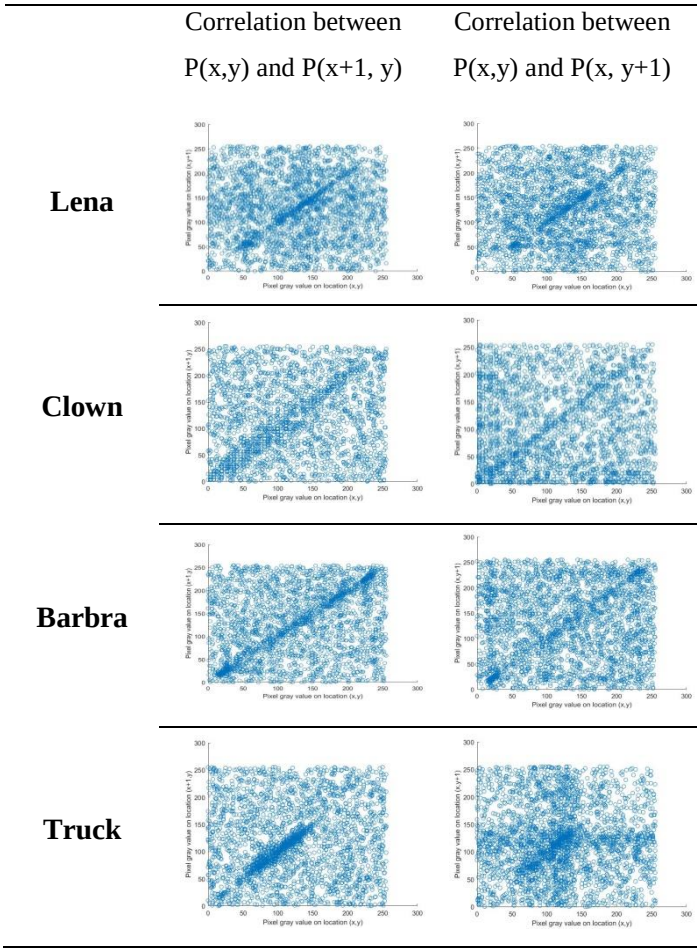


Figure 18. Correlation analysis of the modulo-skipping technique, value S=2.

Figure 18 shows more significant change on the distribution, having less focused areas indicating a much lower correlation values in both the horizontal and the vertical tests. However, we can still find some focused areas indicating the existence of high correlation between their corresponding pixels.

TABLE IV. PERFORMANCE RESULTS OF MODULO-SKIPPING TECHNIQUE, S=2.

Image	**Encryption ratio (%)**	**Time Improvement compared to AES**	**Encryption Duration (s)**	**Decryption Duration (s)**
Lena	67.07%	32,8%	47,26	46,94
Clown	61.52%	33.4%	44,11	42,84
Barbra	61.51%	34,3%	43,40	42,84
Truck	67.50%	32,1%	48,15	47,67

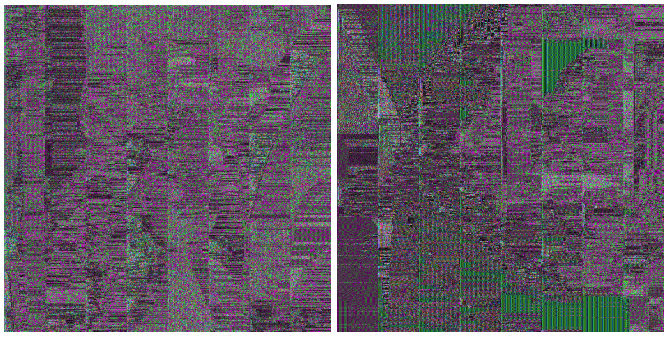
The modulo-skipping technique showed a promising result by reducing encryption/decryption duration by around 35% (with S=2) resulting in a faster encryption/decryption compared to AES technique. In terms of visual recognition, the images were not recognizable and the histograms showed acceptable distributions when selecting a low modulo base (below 5). The correlation analysis showed that even with low modulo bases there was still a noticeable correlation among adjacent pixels.

We conclude that with this technique, the encryption/ decryption duration was improved, but at the cost of losing security in encryption.

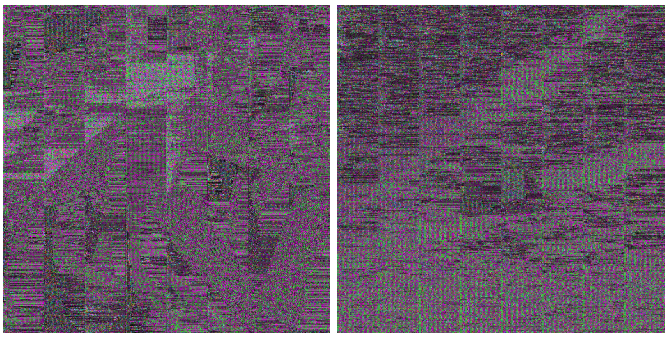
F. The Modulo-XOR technique

In order to increase the blocks being encrypted by the modulo technique, and to reduce the correlation between pixels, we used a second version where blocks are encrypted using XOR instead of skipping them. In this scheme, we used a simple XOR with CBC mode ciphering.

We started our experiments with higher modulo base S=55. The results were as follows:



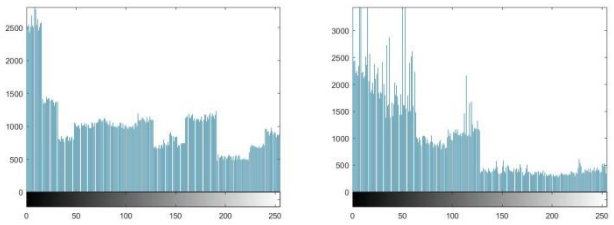
(a) Lena (b) Clown



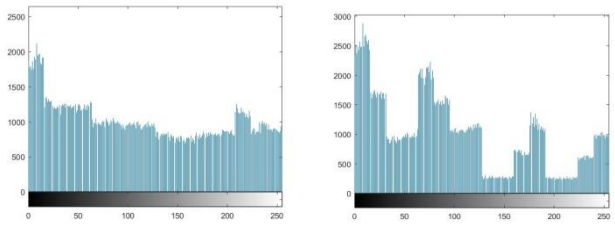
(c) Barbra (d) Truck

Figure 19. Encrypted image results of Modulo-XOR technique, S=55.

Figure 19 shows that the images are not recognizable even with a high modulo base, since we are encrypting the blocks instead of skipping them, resulting in a much better visual result.



(a) Lena (b) Clown



(c) Barbra (d) Truck

Figure 20. Histograms of image results of Modulo-XOR technique, S=55.

Figure 20 shows that the histograms are still uneven, although there is a noticeable change from the original ones. This means that xoring the rest of the data did improve the distribution of the values in case of modulo base 55.

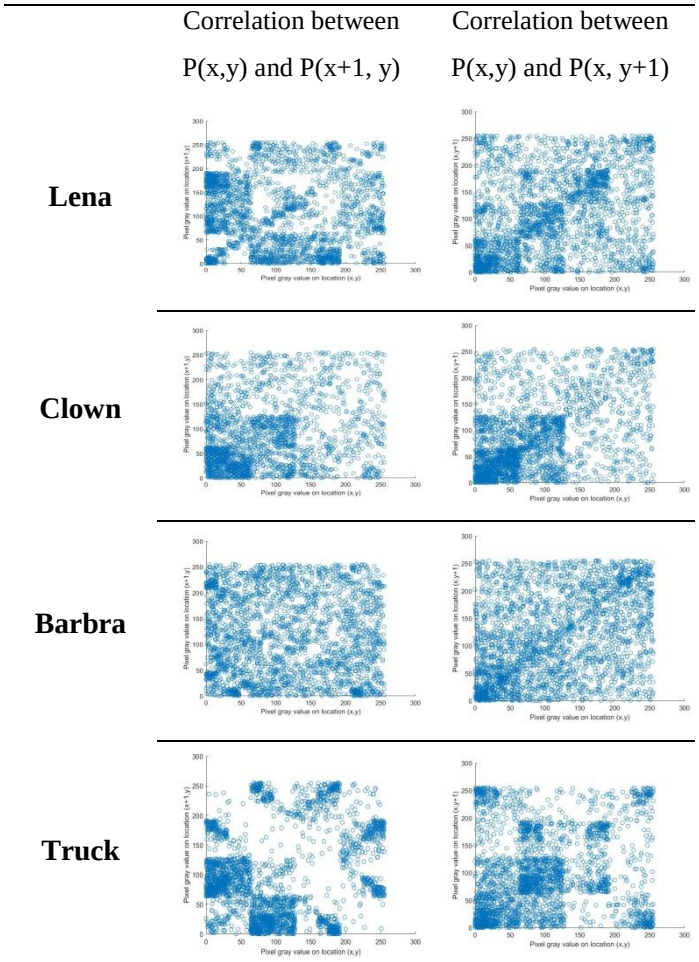


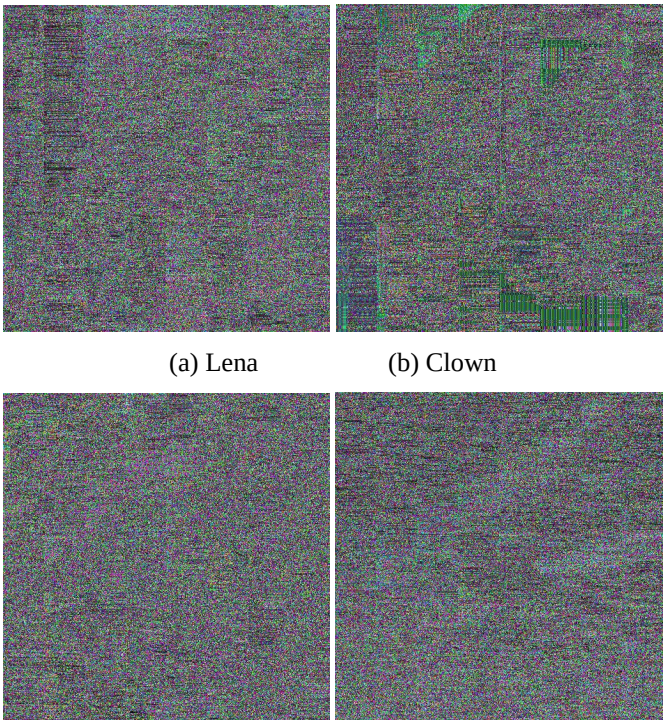
Figure 21. Encrypted image results of Modulo-XOR technique, S=55.

Figure 21 shows a much significant change than the previous technique. We notice that the values are more distributed across all the test images, despite the appearing of some focused areas in the case of Lena, Clown, and Truck. This is probably resulting from the xor operation.

TABLE V. PERFORMANCE RESULTS OF MODULO-XOR TECHNIQUE, S=55.

Image	Encryption ratio(%)	Time Improvement compared to AES	Encryption Duration (s)	Decryption Duration (s)
Clown	100% (AES 5.17%)	93,9%	4,32	3,85
Barbra	100% (AES 3.72%)	96,0%	2,89	2,65
Truck	100% (AES 4.45%)	94,4%	4,16	3,24

Next, we lowered the modulo base to 5 to obtain the following results:

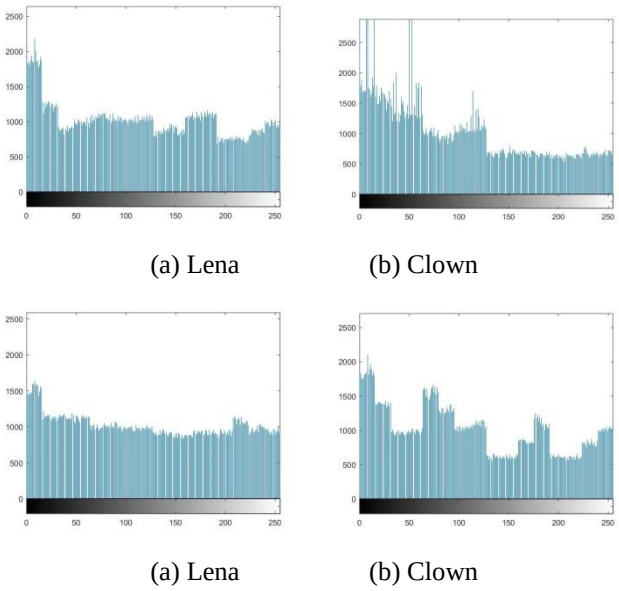


(a) Lena (b) Clown

(c) Barbra (d) Truck

Figure 22. Encrypted image results of Modulo-XOR technique, S=5.

Figure 22 show that the images are not visually recognizable unlike when tested the previous technique with modulo base of 5. Again, results prove that xoring the rest of data had a significant impact on the visual test.



(a) Lena (b) Clown

(a) Lena (b) Clown

Figure 23. Histograms of image results of Modulo-XOR technique, S=5.

Figure 23 shows that the histograms are getting more even as we lower the modulo base, which results in an acceptable distribution across all the test images, especially with Barbra.

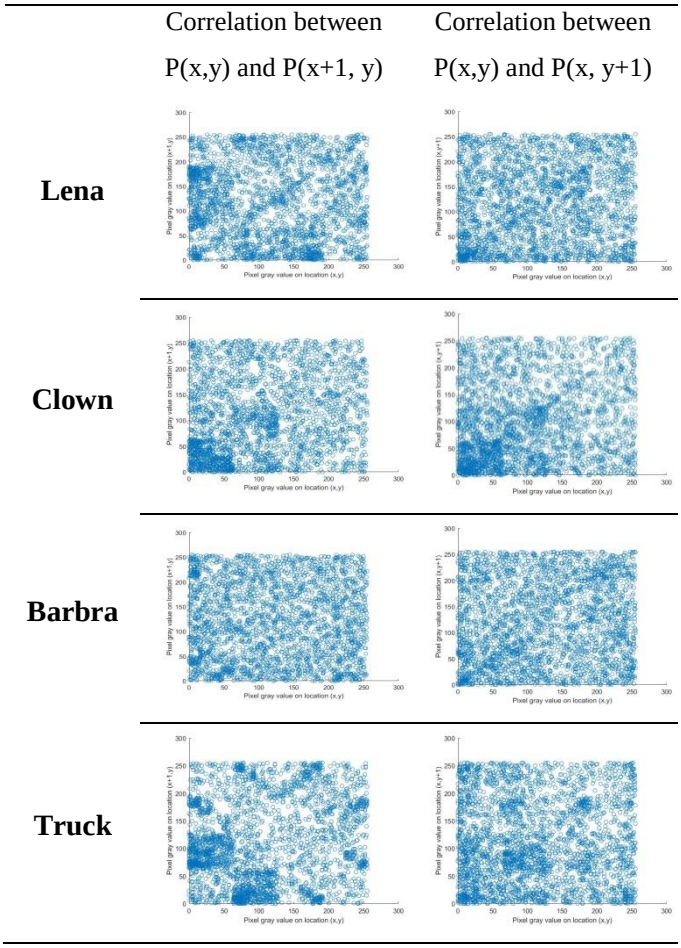


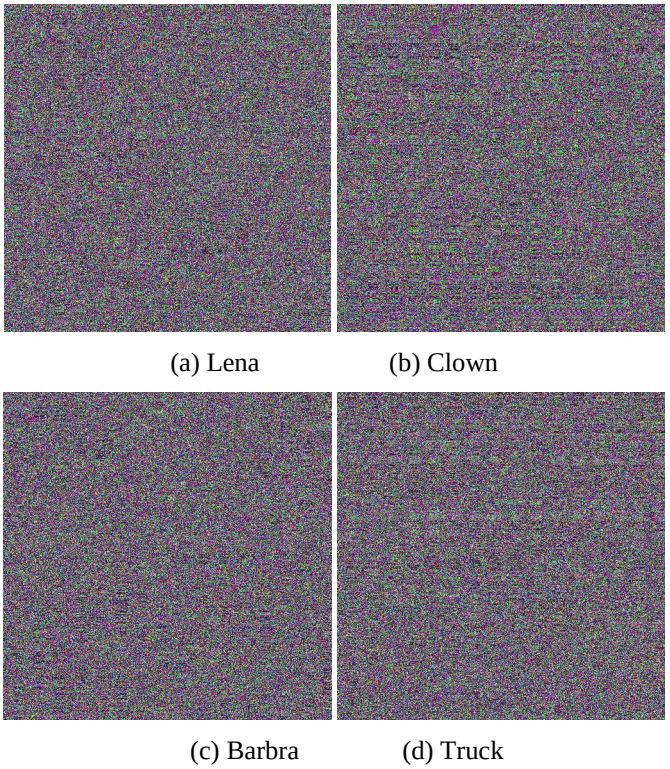
Figure 24. Correlation analysis of the Modulo-XOR technique, value S=5.

Results in Figure 24 show that scatter plots are much better distributed than the previous one (modulo 55). We can also see that focused areas are lesser indicating a much lower correlation values between the adjacent pixels in both the horizontal and the vertical directions.

TABLE VI. PERFORMANCE RESULTS OF MODULO-XOR TECHNIQUE, S=5.

Image	Encryption ratio(%)	Time Improvement compared to AES	Encryption Duration (s)	Decryption Duration (s)
Lena	100% (AES 33.56%)	65,8%	25,36	24,40
Clown	100% (AES 32.68%)	65.4%	25,96	23,88
Barbra	100 % (AES 33.51 %)	65,3%	25,35	24,14
Truck	100 % (AES 35.89 %)	64,5%	26,88	25,65

Next, we performed a third set of tests with modulo base equals to 2. The results are as follows:



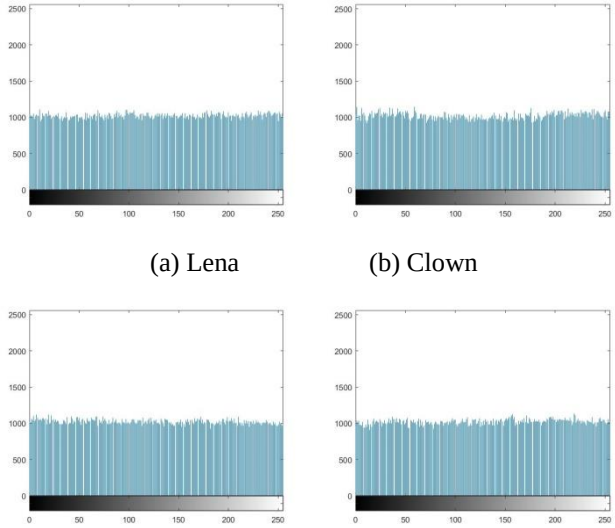
(a) Lena (b) Clown

(c) Barbra (d) Truck

Figure 25. Encrypted image results of Modulo-XOR technique, S=2.

Figure 25 shows that the images are not visually identifiable, and much better than the previous ones. In fact, all

images are looking similar, making the encryption closer to full AES technique.



(a) Lena (b) Clown

(c) Barbra (d) Truck

Figure 26. Histograms of image results of Modulo-XOR technique, S=2.

As we can see the histograms (Figure 26) are perfectly even, a very good distribution of the values, it almost matches the results from the AES technique.

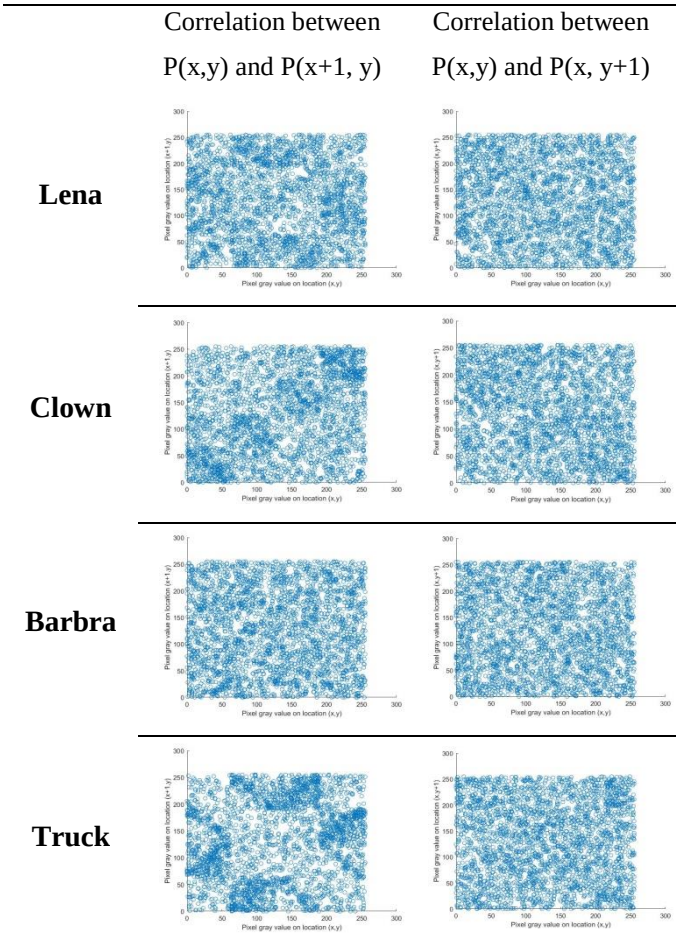


Figure 27. Correlation analysis of the Modulo-XOR technique, value S=2.

Figure 27 show that the best correlation result has been achieved so far with Modulo-XOR technique. The scatter plots are evenly distributed, and we do not see any patterns or focused areas meaning that the correlation is very low for both the horizontal and vertical directions.

TABLE VII. PERFORMANCE RESULTS OF MODULO-XOR TECHNIQUE, S=2.

Image	Encryption ratio(%)	Time Improvement compared to AES	Encryption Duration (s)	Decryption Duration (s)
Lena	100% (AES 67.09%)	31,9%	48,26	46,97
Clown	100% (AES 61.52%)	35,45%	44,96	43,18
Barbra	100 % (AES 61.59 %)	34,3%	45,35	44,01
Truck	100 % (AES 67.50 %)	30,5%	49,62	47,64

In this section, we can say that we have achieved our goal of improving the modulo base technique compared to the previous version (modulo-skipping), since the encrypted images were visually unrecognizable even with high modulo base. Furthermore, the histograms also showed even distributions when xor was involved in encryption.

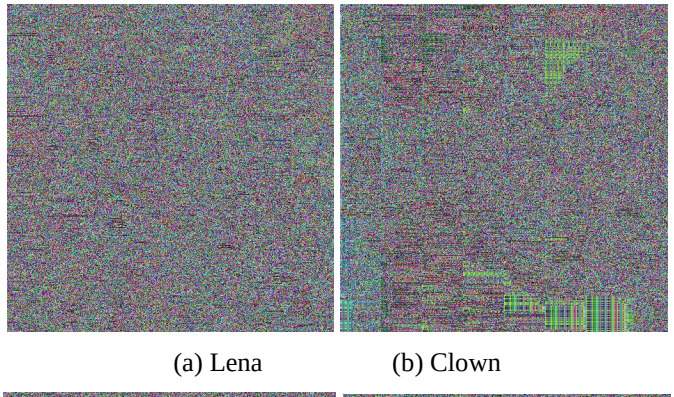
Also, the correlation analysis proved that adding the xor decreased the correlation between adjacent pixels in both horizontal and vertical directions. The time improvement compared to the AES technique was still significant in the vicinity of 31 %.

G. Block Selection using SHA256 Hash Technique

As a third implementation of our model, we tried to use a different hash function. The idea was to hash the image and returns a series of strings that we transform into a series of bits that serve as selectors of blocks to be encrypted (1=encrypted using AES, 0=encrypted using xor CBC cipher).

The digest from SHA256 as the name imply is 256 bits in size (a string of 64 characters in case of hexadecimal representation). We convert the digest into 256 digit long series of ones and zeros that we use to select the blocks we want to encrypt. However, an image could surpass 256 blocks, so to encrypt the rest of the blocks we simply repeated the series until the number of blocks in the image to be encrypted is reached.

The results are as follows:

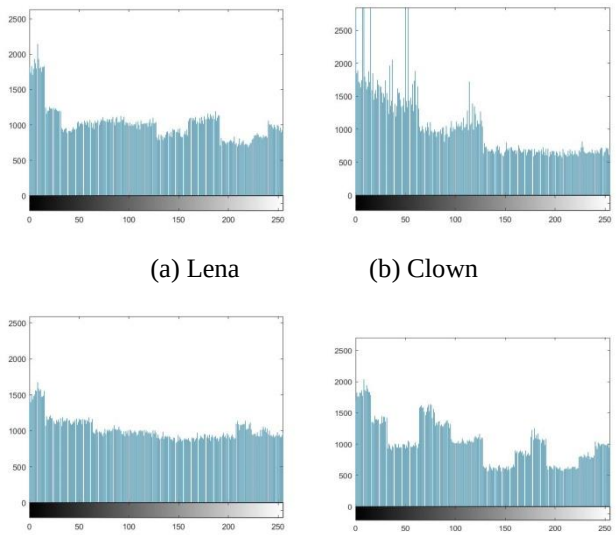


(a) Lena (b) Clown

(c) Barbra (d) Truck

Figure 28. Encrypted image results of SHA256 block selection technique.

We can clearly see in Figure 28 that the resulting images using this technique are not visually recognized giving the block selection technique the passing mark on the visual test.



(a) Lena (b) Clown

(c) Barbra (d) Truck

Figure 29. Histograms of image results of SHA256 block selection technique.

Figure 29 shows that the distribution of the values is well uniformed resulting in an almost even histograms across all test images, with Barbra having the most even histogram.

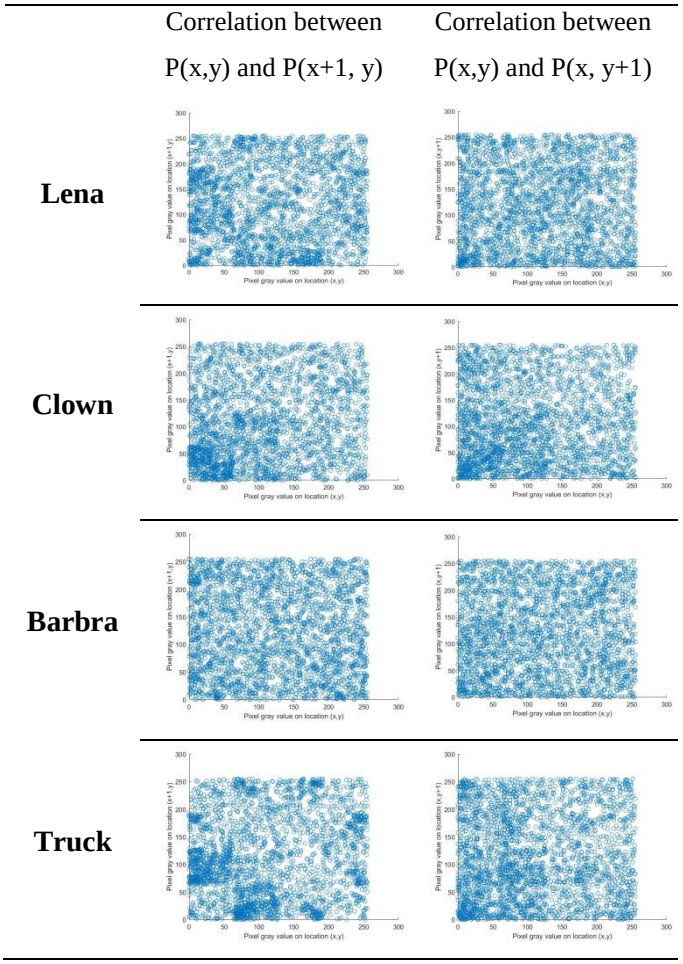


Figure 30. Correlation analysis of the SHA block selection technique.

We can see in Figure 30 that the scatter plots are well distributed. We can also see that there is no focused areas across the test images expect for Truck and Clown where we can notice some small focused areas in the horizontal direction. However, these results indicate a low correlation values in both the horizontal and the vertical axes.

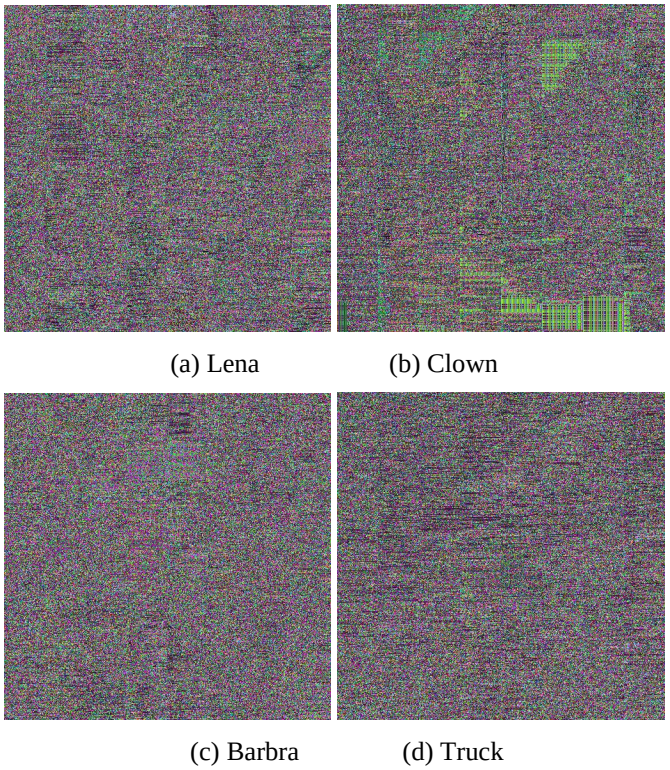
TABLE VIII. PERFORMANCE RESULTS OF SHA256 BLOCK SELECTION TECHNIQUE.

Image	Encryption ratio(%)	Time Improvement compared to AES	Encryption Duration (s)	Decryption Duration (s)
Lena	56,01%	42,6%	39,51	40,81
Clown	55,68%	43.1%	39,81	39,87
Barbra	56,71%	42,7%	39,85	40,51
Truck	54,61%	44,5%	38,32	39,41

Similarly as the previous technique, we used the MD5 has function. The same procedure is maintained of generating a series of strings from the image and then transformed it into series of bits that decide which blocks to be encrypted with AES (bit=1) and those to be encrypted with xor CBC cipher (bit=0).

MD5 produces a 128-bit digest (a 32 string series); we used this output in the same manner as the previous technique.

The results were as follows:

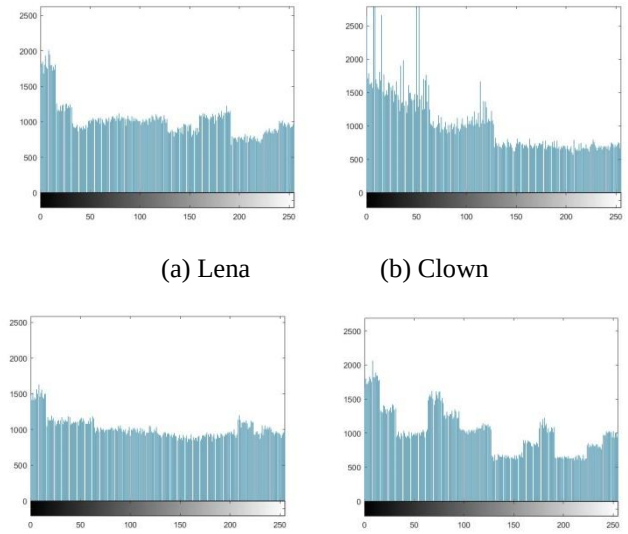


(a) Lena (b) Clown

(c) Barbra (d) Truck

Figure 31. Encrypted image results MD5 block selection technique

Figure 31 shows that this technique also passes the visual recognition test, with all the test images being unrecognizable.



(a) Lena (b) Clown

(c) Barbra (d) Truck

Figure 32. Histograms of image results of MD5 block selection technique.

Figure 32 shows a good distribution of the values across all test samples: histograms are well uniformed, expect for some values; but in general, results are acceptable.

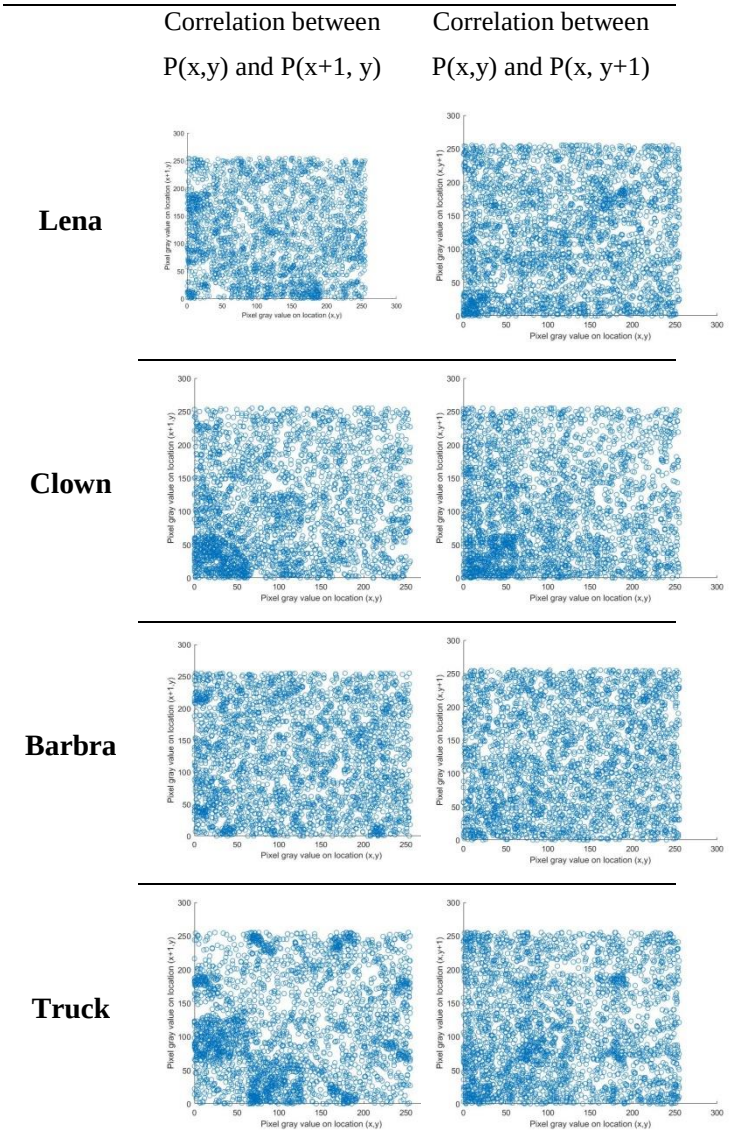


Figure 33. Correlation plots of the MD5 block selection technique.

Quite similar to the previous technique, the scatter plots of test images are well distributed (Figure 33): There is no sign of patterns or focused areas, which indicates a low correlation values between adjacent pixels in both the horizontal and the vertical directions.

TABLE IX. PERFORMANCE RESULTS OF MD5 BLOCK SELECTION TECHNIQUE.

Image	**Encryption ratio(%)**	**Time Improvement compared to AES**	**Encryption Duration (s)**	**Decryption Duration (s)**
Lena	52,36%	48,6%	36,21	35,32
Clown	52,84%	46,8%	38,11	36,32
Barbra	52,33%	47,7%	36,67	36,31
Truck	53,16%	48,7%	36,58	35,12

Both SHA256 and MD5 block selection technique showed a similar result overall, with MD5 having a slight advantage in the Encryption/Decryption time. However, there was almost no difference between those techniques in both the histogram analysis and the correlation analysis: They all showed good distribution and low correlation values. Both techniques had an encryption ratio of around 50 %, with high security as that of AES.

V. CONCLUSION

Security has been and will be a major concern when transmitting images. Therefore, new encryption techniques are being implemented every year. Most of them are used for a while until they are cracked or surpassed by computational power. In the past decade, image encryption is given much attention in research of information security and a lot of image encryption algorithms have been introduced. Due to some specific features of images like huge data capacity and high redundancy, the encryption of an image is different from other file formats like texts. Therefore, new encryption mechanisms are required to replace the traditional methods.

In this work study, we proposed multiple encryption techniques based on AES and other hashing methods like Modulo, SHA256 and MD5. The experiments showed that by selecting a modulo base higher than 5 in the case of modulo-skipping technique the results were not acceptable with the images staying unencrypted. However, if the modulo base is below 5, the technique showed a decent visual results, and a faster encryption/decryption durations compared to AES (around 33 percent faster). In case of Modulo-XOR technique, the visual results were good even with a high modulo base, proving that adding the xor improved the security of the technique. This technique also had a time improvement compared to AES (around 30 percent). For the SHA256 and MD5 selection techniques, they both showed similar time improvement over the AES technique in terms of visual recognition and encryption/decryption time, while the security was maintained based on histograms and correlation analysis.

As a future work, we suggest performing other hashing mechanisms that produces series more zeros than ones, in order to increase the AES block encryption and hence enhancing the time required for image ciphering.

REFERENCES

[1] Al-khassaweneh, Mahmood. "Image encryption method based on using least square error techniques at the decryption stage." International Journal of Information and Computer Security 4.4 (2011): 332-344.

[2] Al-Khassaweneh, Mahmood, and Shefa Tawalbeh. "A value transformation and random permutation-based coloured image encryption technique." International Journal of Information and Computer Security 5.4 (2013): 290-300.

[3] Selected Images. https://www.cs.cmu.edu/ ~chuck/ lennapg/editor.html, visited on 14/09/2018.

[4] Jolfaei, Alireza, Ahmadreza Matinfar, and Abdolrasoul Mirghadri. "Preserving the confidentiality of digital images using a chaotic encryption scheme." International Journal of Electronic Security and Digital Forensics 7.3 (2015): 258-277.