# Security Analysis of Image Cryptosystem Using Stream Cipher Algorithm with Nonlinear Filtering Function

Belmeguenaï Aïssa
Laboratoire de Recherche en Electronique de Skikda
Université 20 Août 1955- Skikda
BP 26 Route d'El-hadaeik
Skikda, Algeria

Derouiche Nadir
Laboratoire de Recherche en Electronique de Skikda
Université 20 Août 1955- Skikda
BP 26 Route d'El-hadaeik
Skikda, Algeria

Mansouri Khaled
Département d'Electronique
Université Badji Mokhtar
Annaba, Algeria

***Abstract*— In this work a new algorithm for encryption image is introduced. This algorithm makes it possible to cipher and decipher images by guaranteeing a maximum security. The algorithm introduced is based on stream cipher with nonlinear filtering function. The Boolean function used in this algorithm is resilient function satisfying all the cryptographic criteria necessary carrying out the best possible compromises. In order to evaluate performance, the proposed algorithm was measured through a series of tests. Experimental results illustrate that the scheme is highly key sensitive, highly resistance to the noises and shows a good resistance against brute-force, Berlekamp-Massey Attack and algebraic attack.**

***Keywords- cipherImage; cryptosystem; key-stream; nonlinear filtering function; stream cipher.***

## I. INTRODUCTION

In this paper, we are interested in the security of the data images, which are regarded as particular data because of their sizes and their information which is two-dimensional and redundant natures. These characteristics of the data make the classical cryptographic algorithms such as DES, RSA, and ... are inefficient for image encryption due to image inherent features, especially high volume image data. Many researchers proposed different image encryption schemes to overcome image encryption problems [1], [2], [3], [4]. In this work, we present a new algorithm for encryption and decryption images by using a stream cipher algorithm with filtering the linear feedback shift registers (LFSRs). The main advantages of such systems are their extreme speed and the change of the key of encryption for each symbol of the plaintext. In term of application, it is still the type of encryption preferentially and quasi-exclusively used in the industrial world (in particular in telecommunications and governmental). It allows implementations in hardware much easier, economic (less complexity). These algorithms are thus used in a privileged way in the case of communications likely to be strongly disturbed because they have the advantage of not propagating the errors [5]. This type of encryption is much faster than block ciphers.

The Boolean function used in this scheme is resilient function satisfying all the criteria cryptographic necessary to carry out a maximum security and can resist to certain attacks [6], [7], [8], [9].

## II. NON LINEAR FILTERING FUNCTION

This system was proposed by Siegenthaler [10] to increase the linear complexity of the binary sequence produced by linear feedback shift register (LFSR). A single register (LFSR) is used, length $L$, producing a binary sequence in maximum period. Certain stages of this register (LFSR) are combined by a nonlinear function $g$.

Such function is called filtering function. The sequence produced by the function which will constitute the key-stream, combined with the clear text. We refer to [11], [12] for further details. The linear complexity of the key-stream is at most $\lambda(s) = \sum_{i=1}^{d} \binom{L}{i}$, where $d$ is the degree algebraic of $g$.

### A. Linear Feedback Register

Linear feedback shift register produce a sequence $s = s_0, s_1, \ldots,$ satisfying the linear recurrence relation $s_n = \sum_{i=1}^{L} c_i s_{n-i}$, $n \geq L$ where $L$ is the length of the LFSR, $c_i \in F_2$ for $i = 1, \ldots, L$ and $s_i \in F_q$, $i \geq 0$.

The $L$ stages, $S_n = (s_n, \ldots, s_{n+L-1})$, is called a state of the shift register and we note $S_n = (s_n)_{n=0}^{\infty}$ the state sequence. We define the feedback polynomial to be $p(X) = 1 + c_1 X + c_2 X^2 + \ldots + c_L X^L$.

The first output symbols $s_0, s_1, \ldots, s_{L-1}$, are initially loaded into the LFSR, these symbols are called the initial state. This is also the secret key of the LFSR.

The sequences $S = S_0, S_1, \ldots$ produced by linear feedback register have many interesting properties such as a

long periodicity. If the feedback polynomial $p$ is primitive the period is $2^L - 1$.

*B. Non Linear Boolean Function*

Nonlinear Boolean function purpose in key-stream generators is to hide the linearity introduced by the LFSRs. A Boolean function is function $g : F_2^n \to F_2$ .

The function $g$ can be represented uniquely by a multivariate polynomial over $F_2$ of the form:

$$g(x_1,...,x_n) = a_0 + \sum_{i=1}^{n} a_i x_i + \sum_{1 \le i \prec j \le n} a_{ij} x_i x_j + ... + a_{12..n} x_1 x_2 ... x_n .$$

Where the coefficients $a_0, a_i, a_{ij}, ..., a_{12..n}$ belong to $F_2$. The degree of this polynomial is called the algebraic degree or simply degree of $g$ , and it is denoted by $\deg(g)$ . The functions of degrees at most one are called affine functions.

## III. Algorithm Description

The fundamental objective of our contribution is to propose a cryptosystem images which allows two people, called traditionally *Alice* and *Bob* (for example), to transfer from the images through a not very sure channel so that a third nobody, pirate can't understand what is exchanged. It is supposed that *Alice* wishes to send in a way made safe by network a plain-image $imag$ of $n \times m$ pixels with *Bob*.

Initially *Alice* transforms the plain-image into binary flows of bits which one calls flow of bits of the plain-image. Then, starting from a secret key $k$ , *Alice* generates the key-stream $Y$ same size as the flow of bits of the plain-image for this session (see algorithm B). Lastly, *Alice* calculates the binary flow of the cipher-image and sends it to *Bob* as shown in the figure 1. *Alice* and *Bob* must exchange the secret key $k$ as a preliminary. *Bob* then receives the binary flow of the cipher-image $C$ , and of dimensioned sound, will use the secret key $k$ to generate the key-stream $Y$ , then, he calculates the binary flow of the deciphered image $X$ . *Bob* put the binary flow of the deciphered image $X$ in the form of an image of $n \times m$ pixels and stores it in $imgdech$ . *Bob* can then visualize $imgdech$ .

If *Alice* wishes to send a new image to *Bob*, he will use a new secret key $k_1$ for this new session.

*A. Encryption and Decryption Image Algorithm*

*Encryption*

*Alice ciphers the plain-image $imag$ while passing by the following stages:*

1. *To read the plain-image $imag$ of $n \times m$ pixels;*
2. *To transform the plain-image into binary values and to store them in $X$ ;*
3. $N \leftarrow$ *the size of* $X$ ;
4. *for* $i = 1$ *to* $N$ *to make ;*
5. *To generate the key-stream* $Y(i)$ *by using the algorithm B ;*
6. *End to make ;*
7. *for* $i = 1$ *to* $N$ *to make*
8. $C(i) = xor\left(X(i), Y(i)\right)$ ;
9. *End to make ;*
10. *The binary flow of the cipher-image* $C$ *is sent.*

*Decryption*

*Bob deciphers the binary flow of the cipher-image $C$ while passing by the following stages:*

1. $N \leftarrow$ *the size of* $C$ ;
2. *for* $i = 1$ *to* $N$ *to make ;*
3. *To generate the key-stream* $Y(i)$ *by using the algorithm B ;*
4. *End to make ;*
5. *for* $i = 1$ *to* $N$ *to make;*
6. $Z(i) = xor\left(C(i), Y(i)\right)$ ;
7. *End to make ;*
8. *To put the binary flow of the deciphered image* $Z$ *in the form of an image of* $n \times m$ *pixels and to store it in* $imgdech$ ;
9. *To post the deciphered image* $imgdech$ .

*B. Key-Stream Calculation Algorithm*

*Inputs:*

- $imag$ : *plain-image;*
- $s_0, s_1, ..., s_{L-1}$ *are initially loaded into the LFSR;*
- $g$ : *filtering function with a 13 variables.*

*Results:*

- $s$ : *binary sequence produced by LFSR ;*
- $Y$ : *Key-stream produced by* $g$ .

*Treatment:*

1. *To read* $N$ *, the size of* $X$ ;
2. *To introduce the secret key, the value of initialization of LFSR* $s_0, s_1, ..., s_{L-1}$ ;
3. *for* $i = 1$ *to* $N + L - 1$ *to make;*
4. *To generate the binary sequence* $s(i)$ *produced by LFSR ;*
5. *End to make ;*
6. *for* $i = 1$ *to* $N$ *to make;*
7. *To generate the key-stream* $Y(i)$ *produced by function* $g$ ;
8. *End to make.*

## IV. THE PROPOSED LFSR AND FILTRING FUNCTION

After the text edit has been completed, the paper is ready for the template. Duplicate the template file by using the Save As command, and use the naming convention prescribed by your conference for the name of your paper. In this newly created file, highlight all of the contents and import your prepared text file. You are now ready to style your paper; use the scroll down window on the left of the MS Word Formatting toolbar.

The realization a stream cipher system which is as resistant as possible to the known attacks requires having an important mathematical tool which makes it possible to generate robust and unforeseeable key-stream on the formal level but also in the field of the implementation.

We considered the linear feedback shift registers of length 521bits to produce a binary sequence. The feedback polynomial of LFSR is chosen to be the primitive polynomial $p(x) = 1 + x^{48} + x^{521}$ and the initial state of LFSR is never allowed to be the all zero state. It follows that LFSR produces a maximum-length sequence of period $T = 2^{521} - 1.$

The filtering function $g$ that we used here is drawn from [13]. This function must be a high algebraic degree, balancedness, good correlations immunity, high non linearity and preferably to have good algebraic immunity to resist certain attacks.

Let $G^0(x_1,...,x_9,x_{10})$ be a function on $F_2^{10}$ proposed for standard LILI-128 (called function $f_d$ [14]) is 3-resilient of algebraic degree 6 and nonlinearity $NG^0 = 480$ with algebraic immunity 4.

Let $G^1(x_1,...,x_9,x_{11},x_{12}) = G^0(x_1,...,x_9,x_{11} \oplus x_{12}).$

Let $F(x_1,...x_{12}) = x_{12} \oplus x_{11} + G^0(x_1,...x_{10})$ and

$H(x_1,...x_{12}) = x_{12} \oplus x_{10} + G^0(x_1,...x_9,x_{11} \oplus x_{12}).$

We construct a function $g$ in 13-variables in the following way, $g(x_1,x_2,...,x_{13}) =$

$(1 \oplus x_{13})F(x_1,...,x_{12}) \oplus x_{13}H(x_1,...x_{12})$ is 5-resilient function, of algebraic degree 7 and nonlinearity $Ng = 2^{12} - 2^7$ with algebraic immunity 6. This function is optimal for the compromise between the degree and the order of resiliency, we have $7 + 5 \le 13 - 1$. This function satisfies all the cryptographic criteria necessary carrying out the best possible compromises.

## V. SIMULATION AND RESULTS

Simulation was carried out using MATLAB V 7.5. The proposed crypto-data hiding methodology was tested in different images. However, we present the results for the four bringing images, illustrated figures. 2.a, 3.a, 4.a and 5.a. They were ciphered with the same key of size 521-bit.

We first, we applied our cryptosystem to different images, we have the following results: From the original images illustrated by the figures 2.a, 3.a, 4.a and 5.a, we applied our Encryption algorithm with a secret key 521 bits in order to obtain the cipher-images illustrated by the figures 2.b, 3.b, 4.b and 5.b. We notice that initial information is not any more visible. From the cipher-images illustrated by the figures 2.b, 3.b, 4.b and 5.b, we apply the algorithm of decryption algorithm (the rebuilding of the original images) with the same key 521 bits in order to obtain the deciphered images illustrated in figures 2.c, 3.c, 4.c and 5.c. Difference between plain images and its corresponding decrypted images shown in figures 2, 3, 4 and 5, and their histograms are shown in figure 6 are prove that, there is no loss of information, the difference is always 0.

## VI. SECURITY ANALYSIS

A good encryption procedure should be robust against all kinds of cryptanalytic, brute-force (exhaustive research) and principal attacks (Berlekamp-Massey Attack, algebraic attack). In this section, the performance of the proposed image cryptosystem is analyzed in detail. We discuss the security analysis of the proposed image encryption scheme including some important ones like key sensitivity analysis, key space analysis, statistical attacks etc. to prove the proposed cryptosystem is secure against the most common attacks.

### A. Key Space Analysis

For secure image encryption, the key space should be large enough to make the exhaustive research attack infeasible. Since the algorithm has a 521 bits key, the intruder needs $2^{521}$ tests by exhaustive research. An image cipher with such as a long key space is sufficient for reliable practical use.

### B. Berlekamp-Massey Attack

For a filtering function of degree $d$, the linear complexity $\lambda(s)$ of the resulting key stream is upper bounded by $\sum_{i=1}^{d} \binom{L}{i}$. Moreover, it is very likely that the $\lambda(s)$ of the key stream $(Y_i)_{i \ge 0}$ is lower bounded by $\binom{L}{d}$ and that its period remains equal to $2^L - 1$. The Berlekamp-Massey attack [15] requires $2\lambda(s)$ data and has a complexity of $\lambda(s)^2$. Using the parameters L = 521; d = 7, linear complexity $\lambda(s)$ is between $2.0125e^{15}$ and $1.9854e^{15}$, it is sufficiently large. This complexity completely excludes to use the Berlekamp-Massey attack.

### C. Algebraic Attack

The complexity $C(L,d)$ of the algebraic attack on the stream cipher system with a key of size $L$ bits and equations of $d$ degree is given by $C(L,d) = \left( \sum_{i=0}^{d} \binom{L}{i}^w \right) = L^{w.d}$ ,

where $w$ corresponds to the coefficient of the method of the solution most effective by the linear system and $d$ is equal to algebraic immunity of the filtering function. We employ here the expression of Strassen [16] which is $w = \log_2(7) \approx 2.807$.

In our cryptosystem the secret key is 521 bits and the algebraic immunity of the filtering function is equal to 6. This leads to algebraic attack with a complexity which is $5.7145e^{45}$, which is sufficiently large. It is not easy to make a linear approximation of the filtering function within the framework of algebraic attack.

*D. Noise Analysis*

We also tested the resistance our cryptosystem to the noise by adding to the cipher-images a noise. From the cipher-images illustrated in the figures 2.b, 3.b, 4.b and 5.b we added a noise of the same size of plain-images. The results are given in the figure 2.d, 3.d, 4.d and 5.d. From the images 2.d, 3.d, 4.d and 5.d, we apply the decryption algorithm presented in section A; we have the results illustrated in figure 2.f, 3.f, 4.f and 5.f. The noise added to ciphers-images 2.b, 3.b is a matrix containing pseudo-random values drawn from a uniform distribution on the unit interval, generates with function "rand".

The noise added to ciphers-images 4.b and 5.b is a matrix containing pseudo-random values drawn from a normal distribution with mean zero and standard deviation one, generates with function "randn". In two cases examined, we can note that the deciphered images presented in figures 2.f, 3.f, 4.f and 5.f are identical to the original images (see 2.a, 3.a, 4.a and 5.a), there is no difference pixel with pixel has indeed between the deciphered images and plain-images because of reversibility of our technique of encryption. Figures 2.e, 3.e, 4.e and 5.a are representing difference image between cipher-images and cipher-images with additive noise.

*E. Sensitivity Analysis*

Thus, we tested our cryptosystem to the sensibility to the keys, for example, we cipher the images 2.a, 3.a, 4.a and 5.a with the secret key $K_1 = 521$ bits and, we decipher it with different key; $K_2 = 521$ bits. The result is given by figure 7.

*F. Correlation Coefficient Analysis*

Table 1 gives the correlation coefficient results. In table 1, we denoted respectively by $Cor_1$, $Cor_2$, $Cor_3$, and $Cor_4$ correlation coefficient between plain-images and encrypted images, correlation coefficient between plain-images and their decrypted images, correlation coefficient between encrypted images and decrypted images with different key; $K_2$, and correlation coefficient between plain-images and decrypted images with different key; $K_2$. It is observed that the correlation coefficient is a small correlation between plain-images and encrypted image, encrypted images and decrypted images with different key; $K_2$, and plain-images and decrypted images with different key; $K_2$.

*G. Entropy Analysis*

Table 2 gives entropy results. In table 2, we denoted respectively by $E_1$, $E_2$, $E_3$, and $E_4$ entropy values: of plain-images, encryptions images, decrypted images and decrypted images with different key; $K_2$. The entropy values of encryptions images, decrypted images with different key; $K_2$ obtained are very close to the theoretical value of 8. This means that information leakage in the encryption process is negligible and the encryption system is secure upon the entropy attack.

*H. Histogramm Analysis*

In the experiments, the original images and its corresponding encrypted images are shown in figure 2, 3, 4 and 5, and their histograms are shown in figure 8. It is clear that the histogram of the encrypted image is nearly uniformly distributed, and significantly different from the respective histograms of the original image. So, the encrypted image does not provide any clue to employ any statistical attack on the proposed encryption of an image procedure, which makes statistical attacks difficult.

These properties tell that the proposed image encryption scheme has high security against statistical attacks. In the original image (i.e. plain image), some gray-scale values in the range [0, 255] are still not existed, but every gray-scale values in the range [0, 255] are existed and uniformly distributed in the encrypted image. Some gray-scale values are still not existed in the encrypted image although the existed gray-scale values are uniformly distributed. Different images have been tested by the proposed image encryption procedure.

## VII. CONCLUSION

In this Work, a new algorithm based encryption scheme for image data was introduced; simulations were carried out for different images. The visual test indicates that the encrypted image was very different and no visual information can be deduced about the original image for all images. In addition, this method is very simple to implement, the encryption and decryption of an image.

Here the security aspects like key space, Berlekamp-Massey attack, algebraic attack, noise analysis, statistical attacks and sensitivity with respect to key, are discussed with examples. It is seen that the present cryptosystem is secure against the statistical attacks, brute force attack, Berlekamp-Massey attack, algebraic attack and to resists the additive noises.

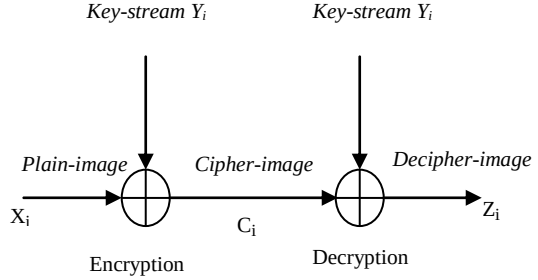



Figure 1. Principal encryption and decryption

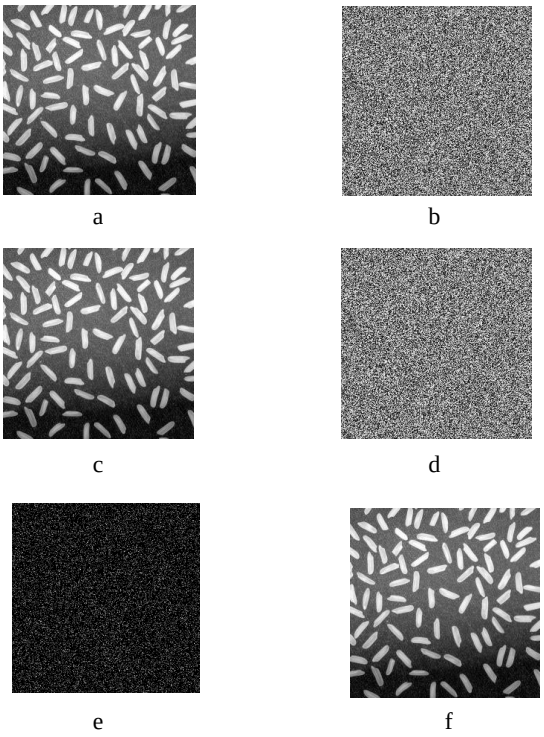



Figure 2. (a) Plain-image, (b) Cipher-image, c) Decipher image, d) Cipher-image with noise added, e) Difference image between image (b) and image (d), f) Decipher image (d).

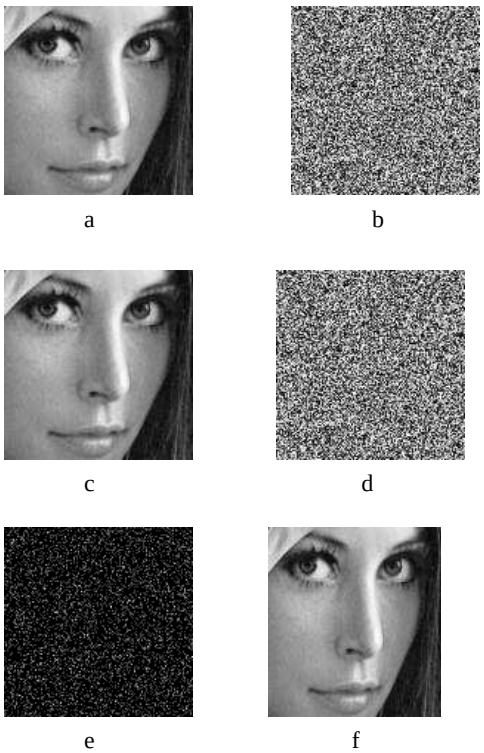



Figure 3. (a) Plain-image, (b) Cipher-image, c) Decipher image, d) Cipher-image with noise added, e) Difference image between image (b) and image (d), f) Decipher image (d).

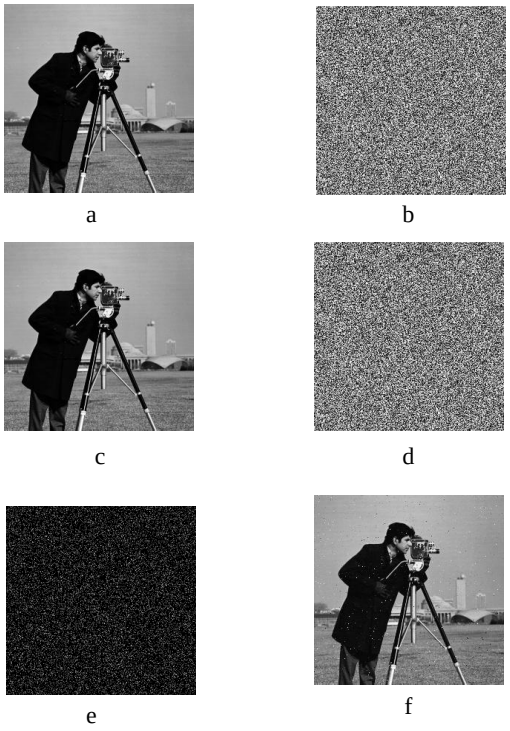



Figure 4. (a) Plain-image, (b) Cipher-image, c) Decipher image, d) Cipher-image with noise added, e) Difference image between image (b) and image (d), f) Decipher image (d).

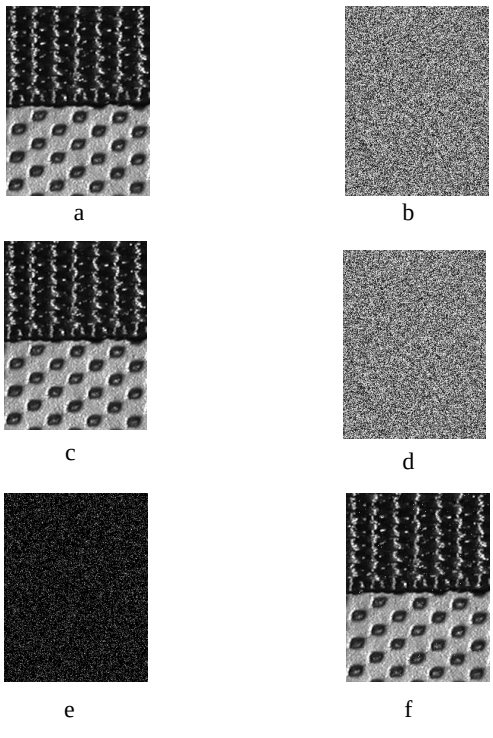



Figure 5. (a) Plain-image, (b) Cipher-image, c) Decipher image, d) Cipher-image with noise added, e) Difference image between image (b) and image (d), f) Decipher image (d).

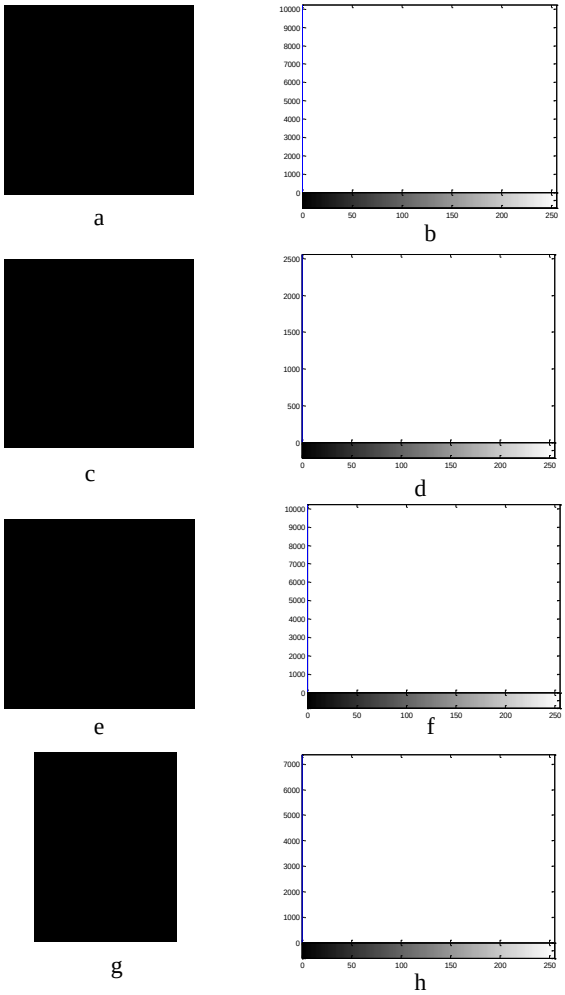
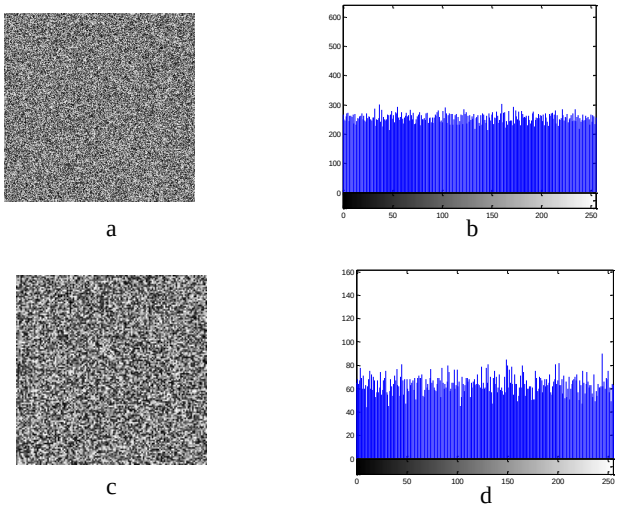



Figure 6. Frame (a), (c), (e) and (g) respectively show the difference between original images shown in figures 2.a, 3.a, 4.a and 5.a, and their decrypted image shown in fig 2.c, 3.c, 4.c and 5.c. Frame (b), (d), (f) and (h) respectively show their histogram.



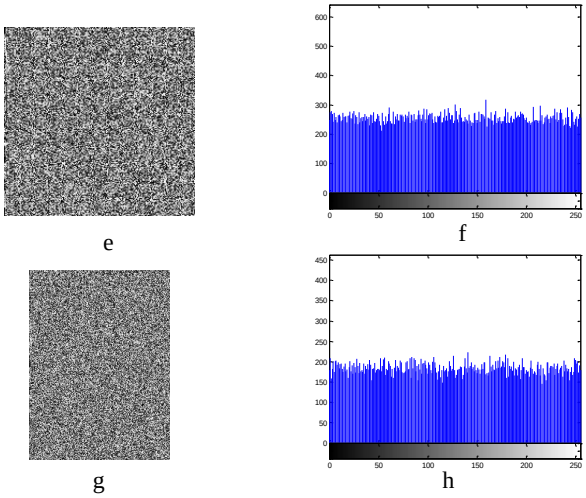




Figure 7. Sensitivity analysis: Frame (a), (c), (e) and (g) respectively, show decrypted image with wrong key ($K_2$) of the encryption images shown in figures 2.b, 3.b, 4.b and 5.b. Frame (b), (d), (f) and (h) respectively, show histogram of images ((a), (c), (e) and (g).

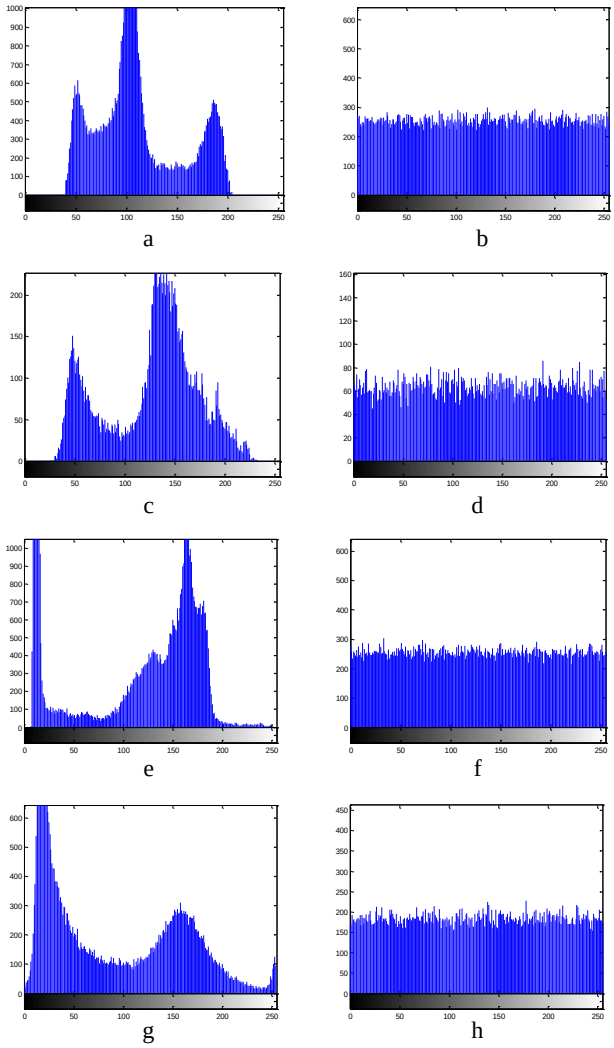



Figure 8. Histogram analysis: Frame (a), (c), (e) and (g) respectively, show the histogram of the plain images shown in figures 2.a, 3.a, 4.a and 5.a. Frame

(b), (d), (f) and (h) show the histogram of the decrypted image shown in figures 2.c, 3.c, 4.c and 5.c.

TABLE I. CORRELATION COEFFICIENTS

| Cases | $E_1$ | $E_2$ | $E_3$ | $E_4$ |
|---|---|---|---|---|
| Image 2.a | 7,0115 | 7,9973 | 7,0115 | 7,9973 |
| Image 3.a | 7,2631 | 7,9904 | 7,2631 | 7,9894 |
| Image 4.a | 7,0097 | 7,9977 | 7,0097 | 7,9972 |
| Image 5.a | 7,4864 | 7,9962 | 7,4864 | 7,9958 |

TABLE II. IMAGES ENTROPY

| Cases | $COR_1$ | $COR_2$ | $COR_3$ | $COR_4$ |
|---|---|---|---|---|
| Image 2.a | 0,0975 | 1 | -0,0055 | -0,0032 |
| Image 3.a | -0,0050 | 1 | -0,0022 | -0,0018 |
| Image 4.a | -0,0068 | 1 | -0,0046 | 0,0024 |
| Image 5.a | -0,0066 | 1 | -0,0022 | -0,0030 |

## REFERENCES

[1] M. Sharma and M. K. Kowar, "Image encryption techniques using chaotic schemes: a review", International Journal of Engineering Science and Technology, vol. II, no. 6, 2010, pp. 2359–2363.

[2] A. Jolfaei and A. Mirghadri, "An applied imagery encryption algorithm based on shuffling and baker's map," Proceedings of the 2010 International Conference on Artificial Intelligence and Pattern Recognition (AIPR-10), Florida, USA, 2010, pp. 279–285.

[3] A. Jolfaei and A. Mirghadri, "A novel image encryption scheme using pixel shuffler and A5/1," Proceedings of The 2010 International Conference on Artificial Intelligence and Computational Intelligence (AICI10), Sanya, China, 2010.

[4] L. Xiangdong, Z. Junxing, Z. Jinhai and H. Xiqin, "Image scrambling algorithm based on chaos theory and sorting transformation," IJCSNS International Journal of Computer Science and Network Security, vol. 8, no. 1, 2008, pp. 64–68.

[5] C. Carlet, "On the cost weight divisibility and non linearity of resilient and correlation immune functions", Proceeding of SETA'01 (Sequences and their applications 2001), Discrete Mathematics, Theoretical Computer Science, Springer p 131-144, 2001.

[6] T. Siegenthaler, "Decrypting a class of stream ciphers using cipher text only", *IEEE Transactions on Computers*, C-34(1):81–85, January 1985.

[7] C. Ding, G. Xiao, and W. Shan, "The stability theory of stream ciphers", Lecture Notes in Computer Science, Number 561, Springer Verlag, August 1991.

[8] N. Courtois and W. Meier, "Algebraic attacks on stream ciphers with linear feedback", Advances in cryptology– EUROCRYPT 2003, Lecture Notes in Computer Science 2656, pp. 346-359, Springer,2002.

[9] N. Courtois, "Fast algebraic attacks on stream ciphers with linear feedback", advances in cryptology–CRYPTO 2003, Lecture Notes in Computer Science 2729, pp. 177-194, Springer, 2003.

[10] T. Siegenthaler, "Cryptanalysis representation of nonlinearly filtered ML-sequences", In : Advances in cryptology- EUROCRYPT' 85, Lectures Notes in Computer science 219,pp 103-110,Springer Verlag, 1986.

[11] P.van Oorschot A. Menezes and S. Vantome, "Handbook of applied cryptography", Available: htt.www.cacr.math.uwaterloo.ca/,1996.

[12] G. Ars, "Une application des bases de Gröbner en cryptographie", DEA de Renne I, 2001.

[13] E. Pasalic, S. Maitra, T. Johansson and P. Sarkar, "New constructions of resilient and correlation immune Boolean functions achieving upper bounds on nonlinearity", In Workshop on Coding and Cryptography - WCC 2001, Paris, January 8–12, 2001. Electronic Notes in Discrete Mathematics, volume 6, Elsevier Science, 2001.

[14] L. Simpson, E. Dawson, J. Golic, and W. Millan, "LILI-128 key-stream generator", In Selected Areas in Cryptography, 7th Annual International Workshop, SAC2000, volume 2012 of Lecture Notes in Computer Science, pages 248–261. Springer-Verlag, Berlin, Heidelberg, New York, 2001.

[15] E.R Berlekamp. "Algebraic coding theory", *Mc Grow- Hill, New- York*, 1968.

[16] V. Strassen, "Gaussian elimination is not optimal", *Numerische Mathematik*, 13:354-356, 1969.