Image Encryption using Improved Keystream Generator of Achterbahn-128

Aissa Belmeguenai[1], Oulaya Berrak[2] and Khaled Mansouri[2]

[1] *Laboratoire de Recherche en Electronique de Skikda, Université 20 Août 1955- Skikda, BP 26 Route d'El-hadaeik Skikda, Algeria*

[2] *Department of Electronics, Faculty of Science and Engineering, Badji Mokhtar University, LP 12 Annaba, Algeria*

Keywords: Boolean Function, Image Encryption, Keystream Generator, Stream Cipher, Security Analysis.

Abstract: The images transmission become more and more widely used in everyday life and even have been known to be vulnerable to interception and unauthorized access. The security of their transmission became necessary. In this paper an improved version of the Achterbahn -128 for image encryption and decryption have been proposed. The proposed design is based on seventeen binary primitive nonlinear feedback shift registers (NLFSRs) whose polynomials are primitive and a nonlinear Boolean function. The outputs of seventeen registers are combined by the nonlinear Boolean function to produce keysteam sequence. The proposed scheme is compared to a Achterbahn-128. The results of several experimental, statistical analysis and sensitivity analysis show that the proposed image encryption scheme is better than Achterbahn-128 and provides an efficient and secure way for image encryption and transmission.

1 INTRODUCTION

The images are very largely used in our daily life; with recent development of information and communication technology, images transmission becomes more critical day by day. Higher security for transmitting data is highly required. Therefore, the stream cipher is an important issue.

Stream cipher is secret key encryption system, which combines plain text bits with a pseudo-random bit sequence. Stream ciphers are widely used in many domains (industrial, governmental, telecommunications and individuals), because they have the advantage of no error propagation, and are particularly suitable for use in environments where no buffering is available and /or plaintext elements need to be processed individually.

The multiplied number of attacks concerning the stream cipher systems based on linear feedback shift registers (combination model or filtering model) (Berlekamp, 1968), (Massey, 1969), (Siegenthaler, 1985), (Meier and Staffelbach, 1988), (Golic, 1994), (Courtois and Meier, 2003), and (Courtois, 2003) have led many researchers to be interested to the design based on primitive nonlinear feedback shift registers (NLFSRs), primarily motivated by eSTREAM, the ECRYPT stream cipher project (eSTREAM, 2002). Here we can mention the research works (Gottfert and Kniffler, 2006) and (Johansson and Meier, 2006).

In this paper a new version of Achterbahn-128 based on primitive NLFSRs oriented stream cipher and also the implementation of this generator for images encryption is introduced. The new version is based on seventeen binary primitive nonlinear feedback shift registers and a Boolean combining function. All feedback shift registers (NLFSRs) employed are primitive and nonlinear. The combining function achieves the best possible trade-offs between algebraic degree, resiliency order and nonlinearity (that is, achieving Siegenthaler's bound and Sarkar et al.'s bound).

The proposed version is compared with the Achterbahn-128. The comparison of the performance of the two designs is investigated for different images.

The paper is organized as follows. In Section 2 we recall the Achterbahn-128. Section 3 gives the specification of the proposed design. In Section 4 we consider the software implementation of the proposed design for image encryption and decryption and in section 5 we give the results of our visual testing. In section 6 we give the security analysis. Section 7 concludes the paper.

Belmeguenai, A., Berrak, O. and Mansouri, K.
Image Encryption using Improved Keystream Generator of Achterbahn-128.
DOI: 10.5220/0005713503330339
In *Proceedings of the 11th Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2016) - Volume 3: VISAPP*, pages 335-341
ISBN: 978-989-758-175-5

2 ACHTERBAHN-128 CIPHER

The Achterbahn-128 (Gottfert and Kniffler, 2006) is a binary additive stream ciphers as candidates to eSTREAM. The Achterbahn-128 cipher is designed to be small, simple and efficient in hardware.

The Achterbahn-128 consists of thirteen binary nonlinear feedback shift registers (NLFSRs) of lengths between 21 and 33 plus a Boolean function F. We denote by x_k for $1 \leq k \leq 13$ the output sequence generated by the k-th constituent NLFSR. All NLFSRs deployed in the Achterbahn-128 are primitive and nonlinear; they can produce binary sequences of period $2^{L_k} - 1$, where L_k is the length of register R_k. The output sequences of the thirteen NLFSR's x_k are combined by a Boolean combining function F of thirteen variables. The combining function F has resiliency 8, nonlinearity 3584, algebraic degree 4 and algebraic immunity 4.

The output of the keystream generator of Achterbahn-128 at time i, denoted by $Z(i)$, is generated as:

$$Z(i) = F(x_1(i), \ldots, x_{13}(i)). \qquad (1)$$

3 IMPROVED VERSION

In this section we give an improved version of the keystream generator of Achterbahn-128. The proposed design consists of seventeen binary primitive NLFSRs denoted R_j of lengths L_j, where $1 \leq j \leq 17$ and a nonlinear Boolean function $G : F_2^{17} \rightarrow F_2$. All feedback shift registers employed in the proposed version are primitive and nonlinear. The NLFSR's are such that they can produce binary sequences of period $2^{L_j} - 1$. Each shift register is described by its feedback function g_j. For the eleven NLFSR's whose lengths $25 \leq L_j \leq 33$, we us the feedback shift registers used in the keystream generator of Achterbahn-128/80 (Gottfert and Kniffler, 2006).

For the others NLFSR's the algebraic normal forms of the feedback functions are given in appendix. The nonzero output sequences of the seventeen binary primitive NLFSRs are taken as input to a combining function, G. It is defined as:

$$G = (1 \oplus P) f \oplus P f^* \oplus Q = f \oplus (f \oplus f^*) P \oplus Q. \qquad (2)$$

Where $f, f \oplus f^*, P$ and Q are given in appendix. For more detaill on G refer to construction 1 page 49 presented in (Dalai, 2006). The combining function G is balanced, has algebraic degree 7, correlation immune of order 9 and has nonlinearity $2^{16} - 2^{10}$.

The keystream sequence $(Y_i)_{i \geq 0}$ is computed as:

$$Y(i) = G(x_1(i), x_2(i), \cdots, x_{17}(i)), \forall i \geq 0, \qquad (3)$$

where $(x_j(i))_{i \geq 0}$ denotes the output sequence generated by the j-th constituent NLFSR. The variables $x_1(i), x_2(i), \ldots, x_{17}(i)$ corresponds to the tap positions $19, 39, 34, 23, 25, 36, 26, 29, 27, 28, 30, 40, 31, 44, 45, 32$ and 33 respectively.

4 IMPLEMENTATION

In this section we present the implementation of proposed design for image encryption and decryption. The implementation of the scheme is written by MATLAB.7.5. By $M(i)$, $Y(i)$ and $C(i)$ we denote respectively original image digits, keystream digits and encrypted image digits at time i. The flow chart of the encryption and decryption process are depicted in figures 1 and 2. Figure 3 illustrates the flow chart diagram for keystream generator.

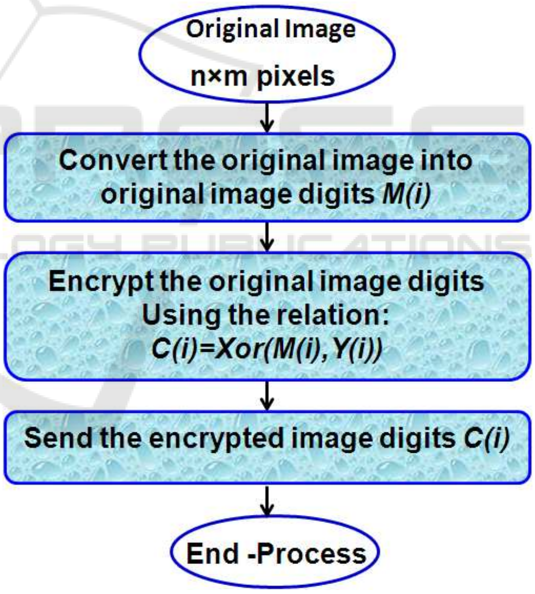
Original Image
n×m pixels
Convert the original image into original image digits M(i)
Encrypt the original image digits Using the relation: C(i)=Xor(M(i),Y(i))
Send the encrypted image digits C(i)
End -Process

Figure 1: Flow chart of the encryption process.

5 VISUAL TESTING RESULTS AND SECURITY ANALYSIS

In this section we discuss the obtained results from implementing the proposed scheme system and the Achterbahn-128. Two images Baboon and House indicated in figures 4 and 5 are used. After loading and processing the original image, it is converted into original image digit then encrypted by the proposed

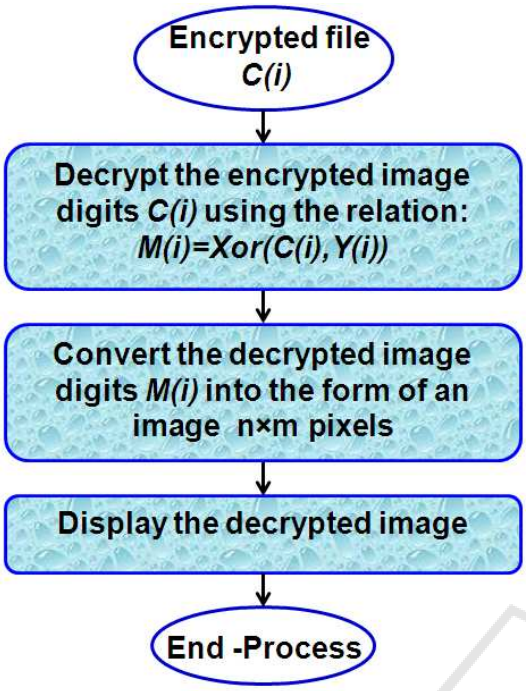
Encrypted file
C(i)
Decrypt the encrypted image digits C(i) using the relation:
M(i)=Xor(C(i),Y(i))
Convert the decrypted image digits M(i) into the form of an image n×m pixels
Display the decrypted image
End -Process

Figure 2: Flow chart of the decryption process.

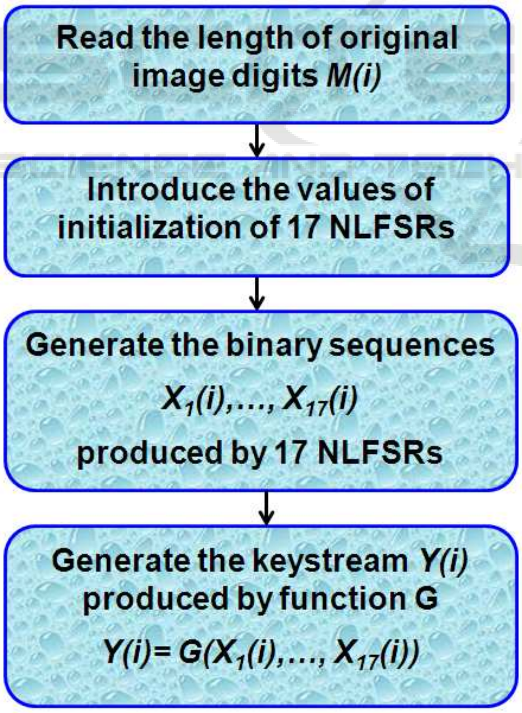
Read the length of original image digits M(i)
Introduce the values of initialization of 17 NLFSRs
Generate the binary sequences
X1(i),..., X17(i)
produced by 17 NLFSRs
Generate the keystream Y(i) produced by function G
Y(i)= G(X1(i),..., X17(i))

Figure 3: Flow chart of the keystream generator.

keystream generator and sent to the receiver. The function of the receiver is to decrypt the encrypted image with the same keystream in order to obtain the original image.

As a general requirement for all the image encryption schemes, the encrypted image should be greatly different from its original form.

From the original images shown in figures 4 and 5, we applied the proposed encryption algorithm in order to obtain the encrypted images illustrated by figures 4 and 5.

By comparing the original images and their encrypted images in figures 4 and 5, there is no visual information observed in the encrypted images, and the encrypted images are visual indistinguishable even with a big difference found in the original images.

From the encrypted images illustrated by the figures 4 and 5, we apply the decryption algorithm with the same key in order to obtain the decrypted images shown in figures 4 and 5.

An image histogram illustrates how pixels in an image are distributed by graphing the number of pixels at each gray intensity level. The histograms of the original images and the encrypted images are compared in figures 4 and 5. Our ideal goal is the encrypted image has histogram with random behavior. It is clear that the histograms of the encrypted images are almost uniformly distributed in gray scale [0-255] and significantly different from the original images histograms. Thus, our approach is more robust against statistical analysis.

Difference between original images and their corresponding decrypted images and their histograms are prove that, there is no loss of information, the difference is always 0.

Figures 6 and 7. show the experimental results of encryption and decryption for Baboon and House using the Achterbahn-128.

5.1 Correlation Coefficient

As an example we take the original image M and the encrypted image C. For gray scale image, the Pearson correlation coefficient between the original image M and encrypted image C is defined as:

$$Cor = \frac{\Sigma_j (M_j - E(M))(C_j - E(C))}{\sqrt{\Sigma_j (M_j - E(M))^2}\sqrt{\Sigma_j (C_j - E(C))^2}}. \quad (4)$$

Where M_j is the intensity of the j-th pixel in image M, $E(M)$ is the mean intensity of the image M.

Table 1 gives the correlation coefficient results. It is observed that the values shown in the table 1 are quite close to the value of zero, which implies that the original images and its corresponding encrypted images are totally different i.e. the encrypted image has no features and highly independent on the original image.

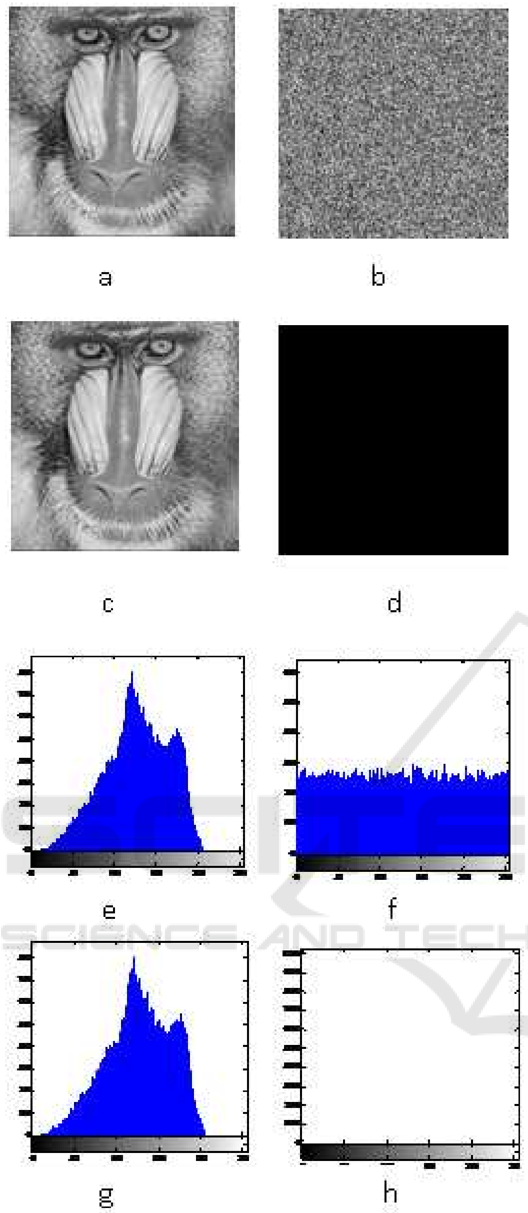

Figure 4: Experimental results using proposed scheme: Frame (a) show image Baboon, frame (b) show encrypted image, frame (c) show decrypted image, frame (d) show the difference between Baboon image and the corresponding decrypted image, frames (e), (f), (g) and (h) respectively; show the histograms of images shown in figures 4(a), 4(b), 4(c) and 4(d).

5.2 Information Entropy

The entropy of a message can be computed by the formula:

$$E(m) = -\Sigma_{i=0}^{255} Pr(m_i) \log_2 Pr(m_i). \qquad (5)$$

Where $Pr(m_i)$ represents the probability of symbol m_i. The entropy is expressed in bits.

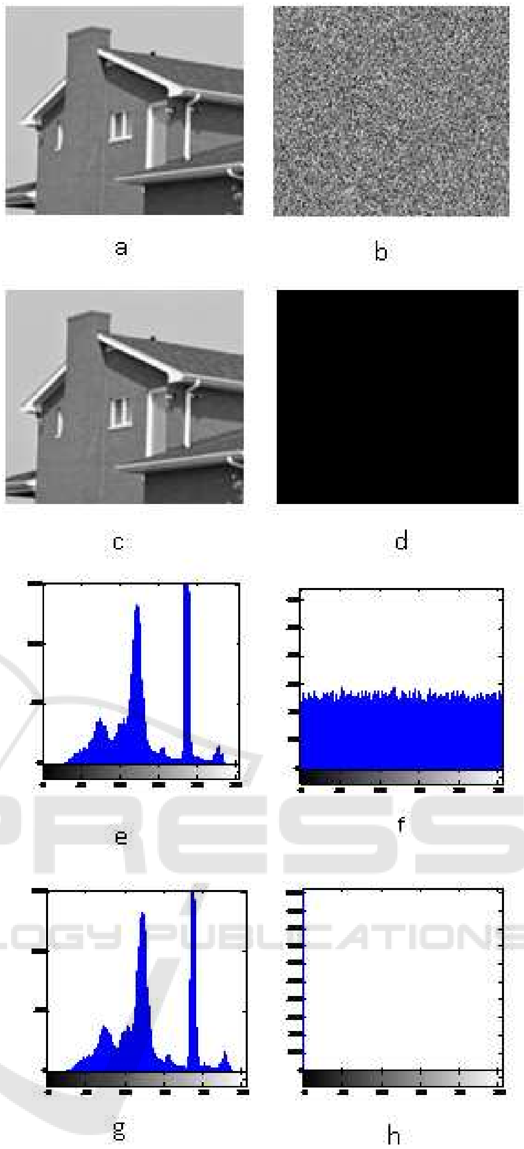

Figure 5: Experimental results using proposed scheme: Frame (a) show image House, frame (b) show encrypted image, frame (c) show decrypted image, frame (d) show the difference between House image and the corresponding decrypted image, frames (e), (f), (g) and (h) respectively; show the histograms of images shown in figures 5(a), 5(b), 5(c) and 5(d).

Table 2 gives the entropy results. The values presented in the table 2 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.

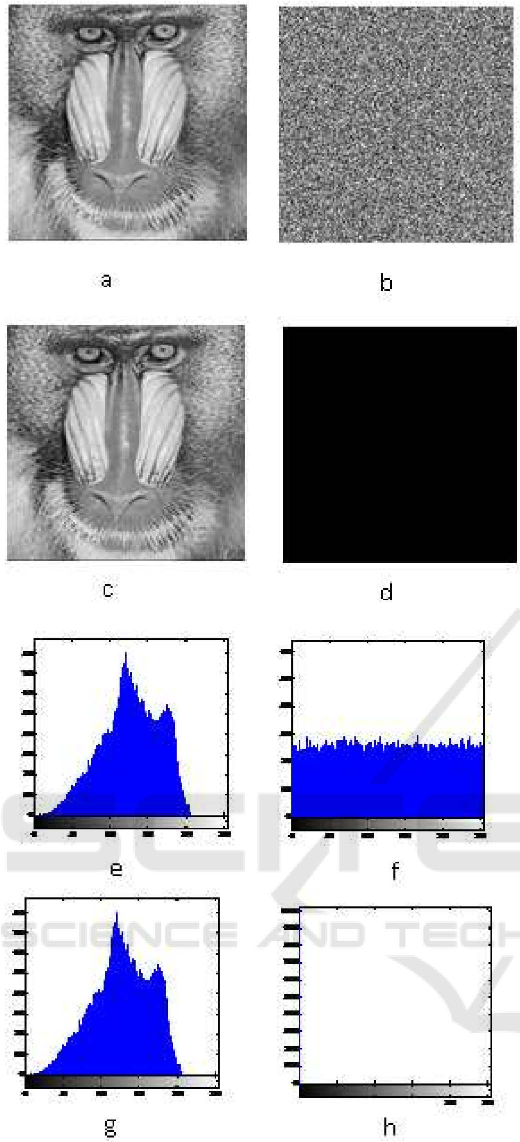

Figure 6: Experimental results using Achterbahn-128: Frame (a) show image Baboon, frame (b) show encrypted image, frame (c) show decrypted image, frame (d) show the difference between Baboon image and the corresponding decrypted image, frames (e), (f), (g) and (h) respectively; show the histograms of images shown in figures 4(a), 4(b), 4(c) and 4(d).

5.3 Sensitivity Analysis

To test the influence of one-pixel change on the overall image encrypted by the proposed algorithm, three common measures were used: the Mean Absolute Error (MAE), number of pixels change rate (NPCR) and unified average changing intensity (UACI).

The MAE, NPCR and UACI are defined respectively by:

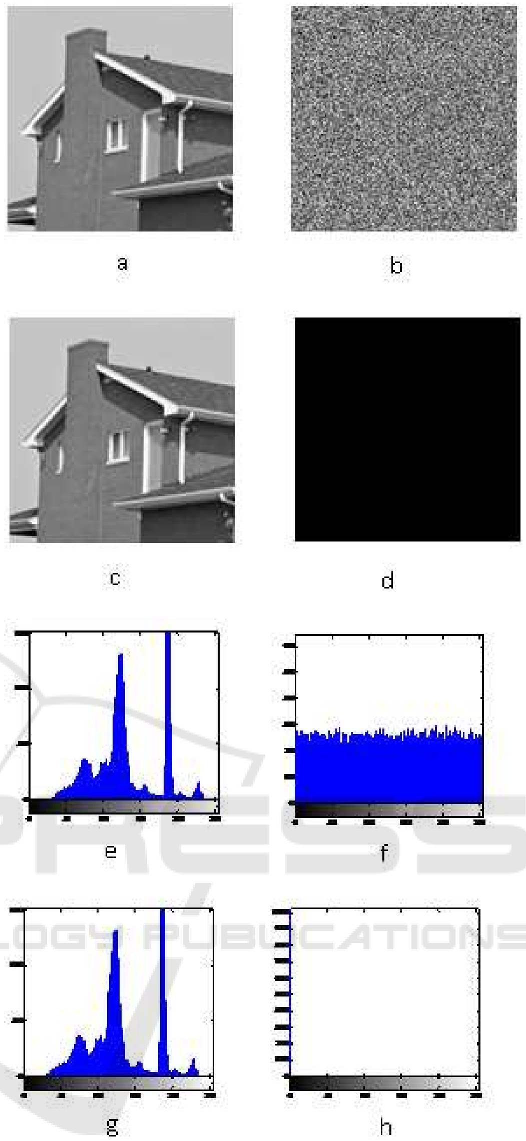

Figure 7: Experimental results using Achterbahn-128: Frame (a) show image House, frame (b) show encrypted image, frame (c) show decrypted image, frame (d) show the difference between House image and the corresponding decrypted image, frames (e), (f), (g) and (h) respectively; show the histograms of images shown in figures 5(a), 5(b), 5(c) and 5(d).

$$MAE = \frac{1}{h \times w} \Sigma_{i=0}^{h-1} \Sigma_{j=0}^{w-1} |C(i,j) - M(i,j)|. \qquad (6)$$

Where $M(i,j)$ and $C(i,j)$ be the gray level of the pixels at the i-th row and j-th column of a $h \times w$ original image and encrypted image, respectively. $|.|$ denotes absolute value function. The NPCR of these two images is defined by:

$$NPCR(M,C) = \frac{1}{h \times w} \Sigma_{ij} D(i,j) \times 100\%. \qquad (7)$$

Table 1: Correlation Coefficients Analysis.

Images	Proposed scheme	Achterbahn-128
Baboon	0.0007805	0.0014
House	0.000070905	0.0066

Table 2: Entropy.

Cases	Proposed scheme	Achterbahn-128
Baboon	7.9985	7.9971
House	7.9992	7.9973

Table 3: MAE, NPCR and UACI between original image and encrypted image using proposed scheme.

Image	MAE	NPCR (%)	UACI(%)
Baboon	34.89	27,38	99,60
House	30.99	28,45	99,60

Table 4: MAE, NPCR and UACI between original image and encrypted image using Achterbahn-128.

Image	MAE	NPCR (%)	UACI(%)
Baboon	35.03	27,34	99,58
House	30.95	28,39	99,58

Where $D(i,j) = 0$, if $C_1(i,j) = C_2(i,j)$ and $D(i,j) = 1$, if $C_1(i,j) \neq C_2(i,j)$, $C_1(i,j)$ and $C_2(i,j)$, whose corresponding original image have only one pixel difference.

$$UACI(M,C) = \frac{1}{h \times w} \Sigma_{ij} \frac{D(i,j)}{255} \times 100\%. \quad (8)$$

A sufficiently high NPCR/UACI score is usually considered to be a strong resistance against attacks.

The MAE,NPCR and UACI test results are shown in table 3 and table 4. Higher NPCR values are desired for ideal encryption schemes. The UACI values must be in the range of 33%. The results demonstrate that the proposed scheme can resist to differential attack.

5.4 Peak Signal to Noise Ratio (PSNR (dB)) Test

Peak signal to noise ratio can be used to evaluate an encryption scheme. PSNR reflects the encryption quality. This is a measurement which indicates the changes in pixel values between the original image and the encrypted image. The formula to calculate PSNR is:

$$PSNR = 10 \times \log_{10} \left[\frac{n \times m \times 255^2}{\Sigma_{i=0}^{n-1} \Sigma_{j=0}^{m-1} (C(i,j) - M(i,j))^2} \right]. \quad (9)$$

Where n and m are the width and height of the original image. The lower value of PSNR represents better encryption quality. The PSNR value of the proposed scheme are 9.31 (dB) for image House and 9.74 (dB) for Baboon. The PSNR value of the Achterbahn-128 are 9.33 (dB) for image House and 9.75 (dB) for Baboon.

6 CONCLUSION

In this Work, a new version of Achterbahn-128 for images encryption was introduced. Simulations were carried out with two 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. The design is very simple, fast and easy to implement for the encryption and decryption of image.

Several tests have been performed to compare the proposed scheme with Achterbahn-128; likely, statistical analysis, which includes information entropy analysis, correlation analysis and Histogram analysis; and MAE, NPCR, UACI and PSNR analysis. The experimental values show that the proposed scheme yield a very good performance over the Achterbahn-128.

REFERENCES

Berlekamp, E. R. (1968). *Algebraic Coding Theory*. Mc Grow-Hil, New- York.

Courtois, N. (2003). Fast algebraic attacks on stream ciphers with linear feedback. In *Advances in cryptology CRYPTO 2003*. Lecture Notes in Computer Science, 2729, p 177–194. Springer.

Courtois, N. and Meier, W. (2003). Algebraic attacks on stream ciphers with linear feedback. In *Advances in cryptology EUROCRYPT 2003*. Lecture Notes in Computer Science, 2656, p 345–359. Springer.

Dalai, D. K. (2006). On some necessary conditions of boolean functions to resist algebraic attacks. Thesis, Applied Statistics Unit Indian Statistical Institute Kolkata, India.

eSTREAM (2002). Ecrypt stream cipher project. In *IST-2002-507932. Available at.* http://www.ecrypt.eu.org/stream/.

Golic, J. D. (1994). Linear cryptanalysis of stream ciphers. In *Fast Software Encryption 1994*. Lecture Notes in Computer Science, 1008, p 154–169, Springer-Verlag.

Gottfert, B. G. R. and Kniffler, O. (2006). Achterbahn-128/80. eSTREAM, ECRYPT Stream Cipher Project, Report 2006/001, http://www.ecrypt.eu.org/stream/p2ciphers/achterbahn/achterbahn-p2.pdf.

Johansson, M. H. T. and Meier, W. (2006). A stream cipher proposal: Grain-128. In *IEEE International Symposium on Information Theory*. ISIT 2006.

Massey, J. L. (1969). Shift-register synthesis and bch decoding. In *IEEE Transactions on information Theory*. vol IT–15, p 122–127,1969.

Meier, W. and Staffelbach, O. (1988). Fast correlation attacks on stream chiper. In *Advances in cryptology-EUROCRYPT'88*. Lectures Notes in Computer science, 430, p 301–314, Springer Verlag, 1988.

Siegenthaler, T. (1985). Decrypting a class of stream ciphers using cipher text only. In *IEEE Transactions on Computers*. C–34, N 1, P 81–85, 1985.

APPENDIX

The algebraic normal form of the function f_0 is: $f_0 = (x_5 \oplus x_8x_5 \oplus x_8x_6 \oplus x_8x_7)(x_1x_4 \oplus x_3x_4 \oplus x_2x_3) \oplus x_6 \oplus x_7x_8x_5 \oplus x_8x_6 \oplus x_1x_4 \oplus x_3x_4 \oplus x_2 \oplus x_1$.

f_0 is balanced and is correlation immune of order 3,it has algebraic degree 4 and nonlinearity 112.

The algebraic normal form of the polynomial $f_0 \oplus f_0^*$ is:

$f_0 \oplus f_0^* = x_5x_8 \oplus x_5x_9 \oplus x_5x_{10} \oplus x_6x_8 \oplus x_6x_9 \oplus x_6x_{10} \oplus x_1x_4x_5x_8 \oplus x_1x_4x_5x_9 \oplus x_1x_4x_5x_{10} \oplus x_3x_4x_5x_8 \oplus x_3x_4x_5x_9 \oplus x_3x_4x_5x_{10} \oplus x_2x_3x_5x_8 \oplus x_2x_3x_5x_9 \oplus x_2x_3x_5x_{10} \oplus x_1x_4x_6x_8 \oplus x_1x_4x_6x_9 \oplus x_1x_4x_6x_{10} \oplus x_3x_4x_6x_8 \oplus x_3x_4x_6x_9 \oplus x_3x_4x_6x_{10} \oplus x_2x_3x_6x_8 \oplus x_2x_3x_6x_9 \oplus x_2x_3x_6x_{10} \oplus x_1x_4x_7x_8 \oplus x_1x_4x_7x_9 \oplus x_1x_4x_7x_{10} \oplus x_3x_4x_7x_8 \oplus x_3x_4x_7x_9 \oplus x_3x_4x_7x_{10} \oplus x_2x_3x_7x_8 \oplus x_2x_3x_7x_9 \oplus x_2x_3x_7x_{10}$.

Where f_0^* is Boolean function generated from f_0 by replacing the variable x_8 by $(x_9 \oplus x_{10})$.

The algebraic normal form of the polynomial P is:

$P = x_{11} \oplus x_{11}x_{14} \oplus x_{12}x_{14} \oplus x_{13}x_{14} \oplus x_{11}x_{14}x_{17} \oplus x_{12}x_{14}x_{17} \oplus x_{13}x_{14}x_{17} \oplus x_{11}x_{15}x_{17} \oplus x_{12}x_{15}x_{17} \oplus x_{13}x_{15}x_{17} \oplus x_{11}x_{16}x_{17} \oplus x_{12}x_{16}x_{17} \oplus x_{13}x_{16}x_{17}$.

The algebraic normal form of the polynomial Q is:

$Q = x_9 \oplus x_{10} \oplus x_{12} \oplus x_{13} \oplus x_8x_{11} \oplus x_9x_{11} \oplus x_{11}x_{14} \oplus x_{12}x_{14} \oplus x_{14}x_{17} \oplus x_{15}x_{17} \oplus x_8x_{11}x_{14} \oplus x_8x_{12}x_{14} \oplus x_8x_{13}x_{14} \oplus x_9x_{11}x_{14} \oplus x_9x_{12}x_{14} \oplus x_9x_{13}x_{14} \oplus x_{11}x_{14}x_{17} \oplus x_{12}x_{14}x_{17} \oplus x_{11}x_{15}x_{17} \oplus x_{12}x_{15}x_{17} \oplus x_{11}x_{16}x_{17} \oplus x_{12}x_{16}x_{17} \oplus x_8x_{11}x_{14}x_{17} \oplus x_9x_{11}x_{14}x_{17} \oplus x_8x_{12}x_{14}x_{17} \oplus x_9x_{12}x_{14}x_{17} \oplus x_8x_{13}x_{14}x_{17} \oplus x_9x_{13}x_{14}x_{17} \oplus x_8x_{11}x_{15}x_{17} \oplus x_9x_{11}x_{15}x_{17} \oplus x_8x_{12}x_{15}x_{17} \oplus x_9x_{12}x_{15}x_{17} \oplus x_8x_{13}x_{15}x_{17} \oplus x_9x_{13}x_{15}x_{17} \oplus x_8x_{11}x_{16}x_{17} \oplus x_9x_{11}x_{16}x_{17} \oplus x_8x_{12}x_{16}x_{17} \oplus x_9x_{12}x_{16}x_{17} \oplus x_8x_{13}x_{16}x_{17} \oplus x_9x_{13}x_{16}x_{17}$.

Agebraic normal forms of the feedback functions used in proposed version

$g_1(u_0, u_1, \cdots, u_{18}) = u_0 \oplus u_2 \oplus u_3 \oplus u_5 \oplus u_8 \oplus u_{12} \oplus u_1u_6 \oplus u_2s_6 \oplus u_2u_9 \oplus u_4u_7 \oplus u_5u_6 \oplus u_9u_{10} \oplus u_9u_{11} \oplus u_2u_4u_6 \oplus u_2u_4u_{10} \oplus u_2u_6u_9 \oplus u_4u_9u_{10} \oplus u_6u_9u_{10} \oplus u_9u_{10}u_{11} \oplus u_2u_4u_6u_9 \oplus u_2u_4u_9u_{10} \oplus u_4u_6u_9u_{10}$.

$g_2(u_0, u_1, \cdots, u_{38}) = u_0 \oplus u_1 \oplus u_3 \oplus u_5 \oplus u_{12} \oplus u_{16} \oplus u_{37} \oplus u_{38} \oplus u_1u_5 \oplus u_5u_8 \oplus u_6u_{10} \oplus u_{11}u_{37} \oplus u_4u_6u_{16} \oplus u_4u_9u_{14} \oplus u_6u_{15}u_{16} \oplus u_1u_9u_{14}u_{36}$.

$g_3(u_0, u_1, \cdots, u_{33}) = u_0 \oplus u_3 \oplus u_6 \oplus u_9 \oplus u_{33} \oplus u_3u_4 \oplus u_4u_8 \oplus u_4u_9 \oplus u_7u_9 \oplus u_1u_3u_4 \oplus u_2u_4u_5 \oplus u_5u_3u_9 \oplus u_1u_4u_9 \oplus u_2u_3u_4u_5 \oplus u_2u_4u_5u_9 \oplus u_2u_3u_4u_5u_9$.

$g_6(u_0, u_1, \cdots, u_{35}) = u_0 \oplus u_2 \oplus u_3 \oplus u_5 \oplus u_8 \oplus u_{35} \oplus u_1u_6 \oplus u_2u_6 \oplus u_2u_9 \oplus u_4u_7 \oplus u_5u_6 \oplus u_9u_{10} \oplus u_9u_{11} \oplus u_2u_4u_6 \oplus u_2u_4u_{10} \oplus u_2u_6u_9 \oplus u_4u_9u_{10} \oplus u_6s_9u_{10} \oplus u_9u_{10}u_{11} \oplus u_2u_4u_6u_9 \oplus u_2u_6u_9u_{10} \oplus u_4u_6u_9u_{10}$.

$g_{12}(u_0, u_1, \cdots, u_{39}) = u_0 \oplus u_2 \oplus u_3 \oplus u_4 \oplus u_5 \oplus u_6 \oplus u_8 \oplus u_{11} \oplus u_{37} \oplus u_1u_{11} \oplus u_2u_{11} \oplus u_3u_{12} \oplus u_4u_6 \oplus u_4u_7 \oplus u_5u_6 \oplus u_1u_2u_{11} \oplus u_1u_2u_{12} \oplus u_1u_9u_{11} \oplus u_9u_{10}u_{11} \oplus u_1u_2u_6u_{13} \oplus u_1u_2u_9u_{11} \oplus u_1u_2u_9u_{12} \oplus u_2u_9u_{10}u_{11} \oplus u_2u_9u_{10}u_{12} \oplus u_1u_2u_6u_9u_{13} \oplus u_2u_6u_9u_{10}u_{13}$.

$g_{14}(u_0, u_1, \cdots, u_{43}) = u_0 \oplus u_1 \oplus u_5 \oplus u_6 \oplus u_8 \oplus u_{35} \oplus u_{41} \oplus u_1u_3 \oplus u_1u_7 \oplus u_1u_{35} \oplus u_4u_{12} \oplus u_5u_{11} \oplus u_6u_{12} \oplus u_7u_9 \oplus u_1u_{11}u_{38} \oplus u_1u_4u_{11}u_{38} \oplus u_1u_7u_{11}u_{38} \oplus u_1u_4u_{10}u_{11}u_{38} \oplus u_1u_7u_9u_{11}u_{38} \oplus u_1u_{10}u_{11}u_{12}u_{38}$.

$g_{15}(u_0, u_1, \cdots, u_{44}) = u_0 \oplus u_1 \oplus u_6 \oplus u_7 \oplus u_9 \oplus u_{10} \oplus u_{11} \oplus u_{15} \oplus u_{16} \oplus u_{18} \oplus u_{22} \oplus u_{26} \oplus u_1u_8 \oplus u_6u_9 \oplus u_{16}u_{18} \oplus u_{17}u_{19} \oplus u_6u_{17}u_{19} \oplus u_9u_{11}u_{12} \oplus u_1u_5u_{16}u_{18} \oplus u_{10}u_{13}u_{16}u_{19}$.