Security Evaluation of Robust Chaotic Block Cipher

Abdul Hamid M. Ragab, Osama S. Farag Allah, Khalid W. Magld, Amin Y. Noaman

Abstract*—this paper investigates an enhanced robust chaotic block cipher (RCBC) which is used for potential increasing security. It makes heavy use of non-linear Boolean functions including parity and multiplexer functions, in addition to multiplication primitive operation with block size of 256-bits. It greatly guaranteed an increased diffusion achieved per round, allowing for greater security and fewer rounds. Comparative analysis of the cipher with different algorithms such as RC6, and RC5 is investigated; regarding design parameters and speed. The cipher is tested among its several design parameters including word size, number of rounds, and secret key length and their optimal choice values. Security estimation for digital imaging against brute-force, statistical, and differential attacks is explored from strict cryptographic viewpoint. Thorough experimental tests are carried out with detailed analysis, demonstrating the high security of the cipher.

***Index Terms* — Block ciphers, Symmetric encryption, Chaos, and Security analysis.**

I. INTRODUCTION

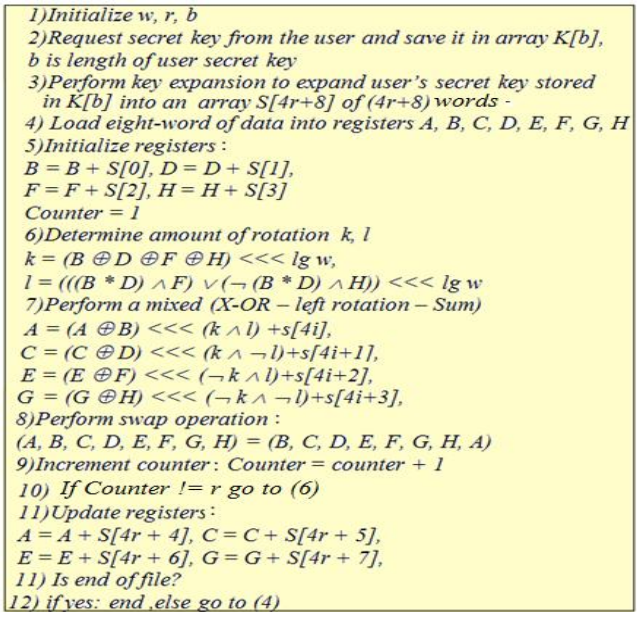
Chaotic encryption is studied in many several articles [1-4]. However, the problems related to security analysis of chaotic block ciphers hasn't been tackled yet, as investigated in details in this paper. An efficient encryption block cipher based on chaotic maps is presented, which is a symmetric encryption algorithm. It uses eight working registers providing capability to deal with 256-bits plaintext/ciphertext block sizes. It employs a chaotic logistic map in key scheduling to generate session key used in encryption/decryption. It uses heavy data-dependent rotations and the inclusion of integer multiplication as an additional primitive operation. The use of multiplication with eight working registers greatly increases the diffusion achieved per round, allowing for greater security, fewer rounds, and increased throughput. We examine the RCBC implementation for digital imaging, and security analysis with respect to brute-force, information entropy, cipher cycle, statistical, and differential attacks. The RCBC is compared with the well known block ciphers such as RC5 [5, 6] and RC6 [7, 8], with respect to its speed and cipher design parameters.

Experimental results for security analysis show that the RCBC block cipher has satisfactory security, which makes it a potential candidate for encryption of multimedia data such as images, audios and videos.

The rest of the paper is organized as follows: section II, explores the description of the RCBC including its features, design parameters, the primitive operations, and description for the RCBC architecture and specification. Implementation issues of key-expansion algorithm for the cipher are explained in section III. The detailed security analysis of the block cipher including key space analysis, information entropy analysis, cipher cycle analysis, statistical analysis, and sensitivity analysis are given in sections IV and V. Then, the paper conclusion is given.

II. DESCRIPTION OF THE RCBC ENC/DEC ALGORITHMS

Fig.1a. and Fig.1b show the RCBC cipher encryption and decryption algorithms, respectively. The cipher works on 256-bit input/output blocks, the design uses eight 32-bit registers**,** named A, B, C, D, E, F, G, and H. They contain the initial input plaintext as well as the output ciphertext at the end of the encryption. The advantage of that it contributes to increase number of rotations per round, and uses more bits of data to determine the rotation amounts in each round. The RCBC is a symmetric block cipher that has an iterative looping structure. All the rounds of encryption and decryption are identical in general; each round uses several operations with round keys to process the input data. A round of encryption (or decryption) performs a series of operations on the input data block and the round keys to generate the intermediate output data block. An output data block is then used as input data block for the next round. After predetermined number of rounds, and preprocessing and post-processing steps, then the ciphertext (or plaintext) is generated.



```
1)Initialize $w, r, b$
2)Request secret key from the user and save it in array $K[b]$,
$b$ is length of user secret key
3)Perform key expansion to expand user's secret key stored
  in $K[b]$ into an array $S[4r+8]$ of $(4r+8)$ words -
4) Load eight-word of data into registers A, B, C, D, E, F, G, H
5)Initialize registers :
$B = B + S[0], D = D + S[1],$
$F = F + S[2], H = H + S[3]$
Counter = 1
6)Determine amount of rotation $k, l$
$k = (B \oplus D \oplus F \oplus H) <<< \lg w,$
$l = (((B * D) \wedge F) \vee (\neg (B * D) \wedge H)) <<< \lg w$
7)Perform a mixed (X-OR – left rotation – Sum)
$A = (A \oplus B) <<< (k \wedge l) + s[4i],$
$C = (C \oplus D) <<< (k \wedge \neg l) + s[4i+1],$
$E = (E \oplus F) <<< (\neg k \wedge l) + s[4i+2],$
$G = (G \oplus H) <<< (\neg k \wedge \neg l) + s[4i+3],$
8)Perform swap operation :
$(A, B, C, D, E, F, G, H) = (B, C, D, E, F, G, H, A)$
9)Increment counter : Counter = counter + 1
10) If Counter != r go to (6)
11)Update registers :
$A = A + S[4r + 4], C = C + S[4r + 5],$
$E = E + S[4r + 6], G = G + S[4r + 7],$
11) Is end of file?
12) if yes: end ,else go to (4)
```

Fig.1a the RCBC Encryption Algorithm.

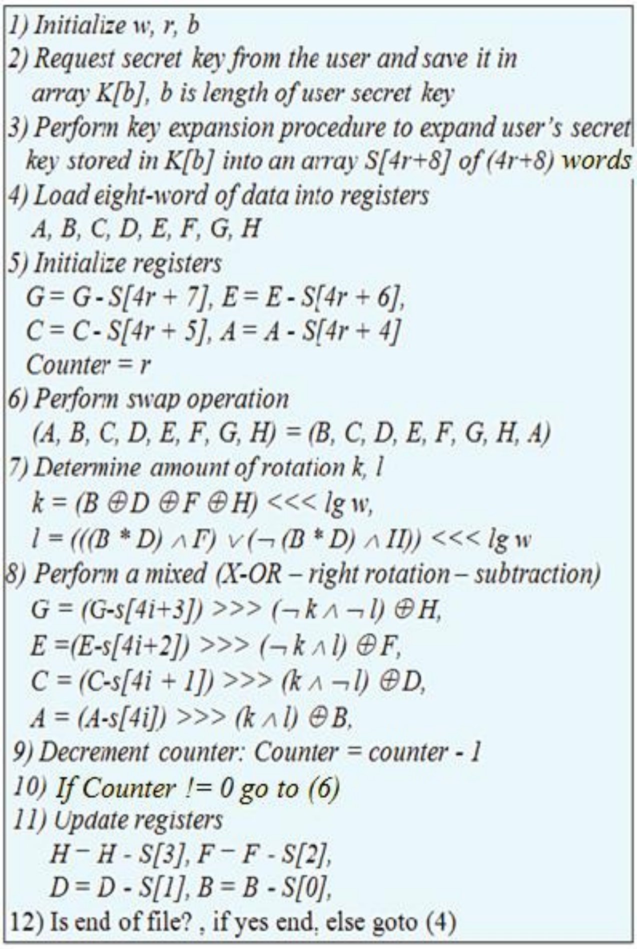
Manuscript received January 15, 2014.

Prof: Abdul Hamid M. Ragab, Prof. since 1995, Academic Staff Member & Consultants & Chairman of Computer Science and Eng. Referee for scientific Journals and Conferences.

Assoc. Prof: Osama S. Farag Allah, Associate Prof in Computer Science & Engineering. He got his Ph.D. in Computer Science & Engineering in 2007 from Menoufia University.

Assist. Prof: Khalid W. Magld, Computer Science University of Bradform, UK.

Assoc. Prof: Amin Y. Noaman, Computer Science from University of Manitoba, Canada.



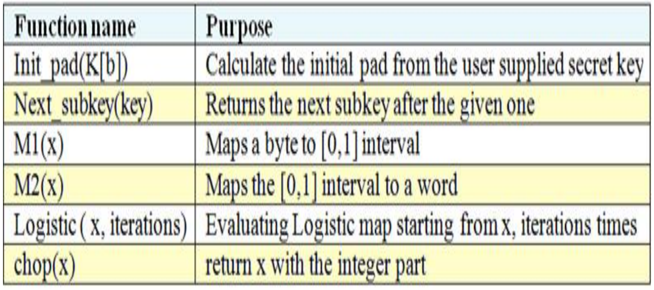
```
1) Initialize w, r, b
2) Request secret key from the user and save it in
   array K[b], b is length of user secret key
3) Perform key expansion procedure to expand user's secret
   key stored in K[b] into an array S[4r+8] of (4r+8) words
4) Load eight-word of data into registers
   A, B, C, D, E, F, G, H
5) Initialize registers
   G = G - S[4r + 7], E = E - S[4r + 6],
   C = C - S[4r + 5], A = A - S[4r + 4]
   Counter = r
6) Perform swap operation
   (A, B, C, D, E, F, G, H) = (B, C, D, E, F, G, H, A)
7) Determine amount of rotation k, l
   $k = (B \oplus D \oplus F \oplus H) <<< \lg w,$
   $l = (((B * D) \wedge F) \vee (\neg (B * D) \wedge H)) <<< \lg w$
8) Perform a mixed (X-OR – right rotation – subtraction)
   $G = (G-s[4i+3]) >>> (\neg k \wedge \neg l) \oplus H,$
   $E = (E-s[4i+2]) >>> (\neg k \wedge l) \oplus F,$
   $C = (C-s[4i+1]) >>> (k \wedge \neg l) \oplus D,$
   $A = (A-s[4i]) >>> (k \wedge l) \oplus B,$
9) Decrement counter: Counter = counter - 1
10) If Counter != 0 go to (6)
11) Update registers
    H − H - S[3], F − F - S[2],
    D = D - S[1], B = B - S[0],
12) Is end of file? , if yes end, else goto (4)
```

Fig.1b the RCBC Decryption Algorithm.

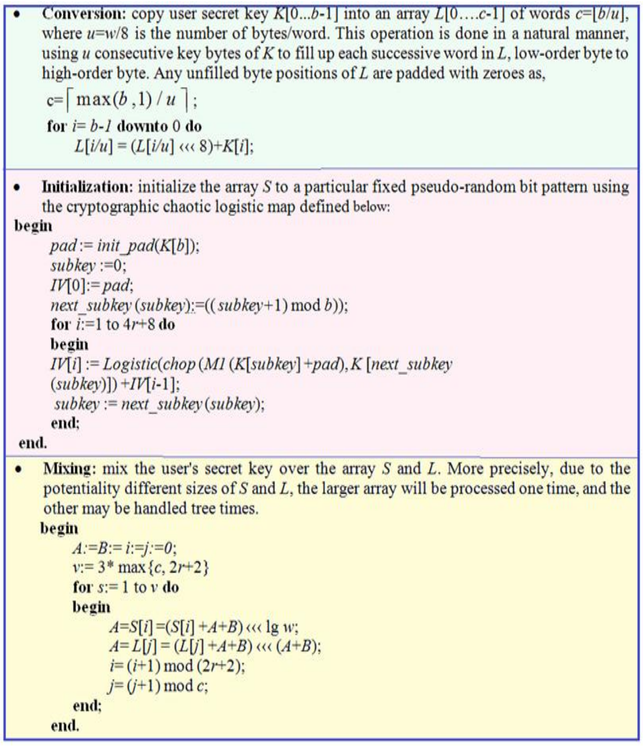
III. THE RCBC KEY-EXPANSION ALGORITHM

The key schedule for RCBC-w/r/b is used to generate the round keys that are derived from user-supplied key for use during encryption and decryption; where a user is supplied a key of b bytes. Sufficient zero bytes are appended to give a key length equal to a non-zero integral number of words; these key bytes are then loaded in little-endian fashion into an array of c "w-bit words" named L[0], …, L[c - 1]. Thus the first byte of key is stored as the low-order byte of L [0], etc., and L[c - 1] is padded with high-order zero bytes if necessary. The number of w-bit words that will be generated for the additive round keys is (4r + 8) and these are stored in the array S [0, 4r + 7]. The key expansion algorithm consists of three simple algorithmic parts. These parts are "convert", "initialize", and "mix" respectively, as shown in Fig. 2. The functions, defined in Table-I, are used in the RCBC implementation.

Table-I Functions used in RCBC Key scheduling implementation.



Function name	Purpose
Init_pad(K[b])	Calculate the initial pad from the user supplied secret key
Next_subkey(key)	Returns the next subkey after the given one
M1(x)	Maps a byte to [0,1] interval
M2(x)	Maps the [0,1] interval to a word
Logistic (x, iterations)	Evaluating Logistic map starting from x, iterations times
chop(x)	return x with the integer part



- **Conversion:** copy user secret key $K[0...b\text{-}1]$ into an array $L[0....c\text{-}1]$ of words $c=\lceil b/u \rceil$, where $u=w/8$ is the number of bytes/word. This operation is done in a natural manner, using u consecutive key bytes of K to fill up each successive word in L, low-order byte to high-order byte. Any unfilled byte positions of L are padded with zeroes as,

```
c = ⌈ max(b ,1) / u ⌉;
for i = b-1 downto 0 do
    L[i/u] = (L[i/u] <<< 8)+K[i];
```

- **Initialization:** initialize the array S to a particular fixed pseudo-random bit pattern using the cryptographic chaotic logistic map defined below:

```
begin
    pad := init_pad(K[b]);
    subkey :=0;
    IV[0]:= pad;
    next_subkey (subkey):=((subkey+1) mod b));
    for i:=1 to 4r+8 do
    begin
    IV[i] := Logistic(chop (M1 (K[subkey] +pad), K [next_subkey
    (subkey)]) +IV[i-1];
     subkey := next_subkey (subkey);
    end;
end.
```

- **Mixing:** mix the user's secret key over the array S and L. More precisely, due to the potentiality different sizes of S and L, the larger array will be processed one time, and the other may be handled tree times.

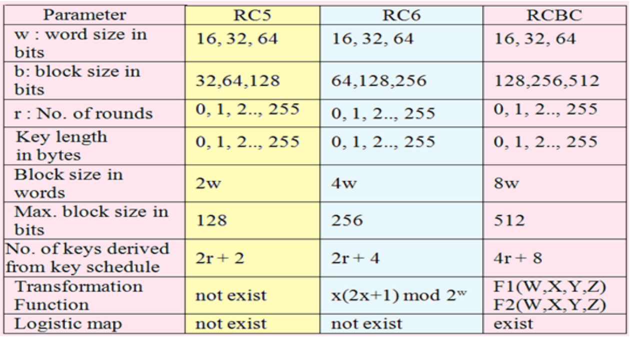
```
begin
    A:=B:= i:=j:=0;
    v:= 3* max{c, 2r+2}
    for s:= 1 to v do
    begin
        A=S[i] =(S[i] +A+B) <<< lg w;
        A= L[j] = (L[j] +A+B) <<< (A+B);
        i= (i+1) mod (2r+2);
        j= (j+1) mod c;
    end;
end.
```

Fig.2 RCBC key expansion algorithm.

IV. THROUGHPUT AND DESIGN PARAMETERS COMPARISONS BETWEEN RCBC, RC5, AND RC6

The design philosophy of the RCBC is to exploit operations such as rotations that are efficiently implemented on modern processors. The RCBC continues this trend, and takes advantage of the fact that 32 bits integer multiplication is now efficiently implemented on most processors. Integer multiplication is a very effective "diffusion" primitive, and is used in the cipher to contribute in computing rotation amounts, so that the rotation amounts are dependent on all of the bits of another register. As a result the RCBC has much faster diffusion than RC5 and RC6. Also, it runs with fewer rounds at increased security and with increased throughput. Table II shows a comparison between RCBC design parameters and RC5, and RC6 block ciphers. It seen from Table II that RCBC uses more cryptographic parameters (logistic maps). This means that the RCBC can be more robust than both RC5 and RC6. Detailed analyses are explained in next sections.

Table-II Comparison between RCBC, RC5, and RC6 design parameters.



Parameter	RC5	RC6	RCBC
w : word size in bits	16, 32, 64	16, 32, 64	16, 32, 64
b: block size in bits	32,64,128	64,128,256	128,256,512
r : No. of rounds	0, 1, 2.., 255	0, 1, 2.., 255	0, 1, 2.., 255
Key length in bytes	0, 1, 2.., 255	0, 1, 2.., 255	0, 1, 2.., 255
Block size in words	2w	4w	8w
Max. block size in bits	128	256	512
No. of keys derived from key schedule	2r + 2	2r + 4	4r + 8

Transformation Function	not exist	$x(2x+1) \bmod 2^w$	F1(W,X,Y,Z)
			F2(W,X,Y,Z)

Logistic map	not exist	not exist	exist

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

There are two inputs to the encryption function, which are the plain-image to be encrypted and the expanded secret key. In the RCBC image encryption, all parts of the file header are determined to know the start of the image pixels data array. The image header; on which the encryption is performed; is then excluded [9-11, 12]. The image data bit stream (not including the image header) is divided into blocks of 256-bit length. The first 256-bit block of image is entered as the plain-image to the encryption function of RCBC. The second input to the RCBC encryption algorithm is the expanded secret key that is derived from the user-supplied secret key; by a key schedule. This key schedule is an important component of a block cipher; since it computes the round keys from the user-supplied secret key [12]. Then, the next 256-bit plain-image block follows it, and so on as shown in Fig.3.

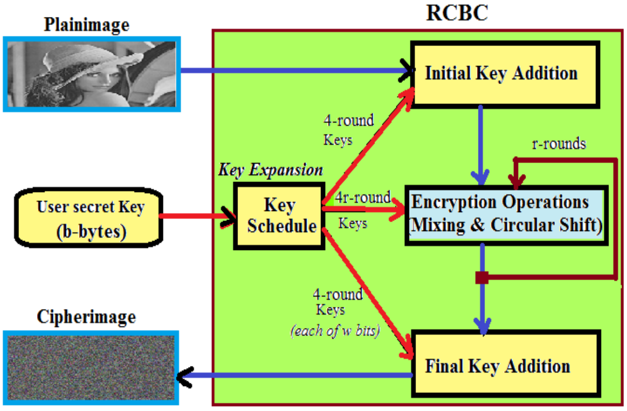


Fig. 3 the RCBC Block Diagram for Image Encryption.

In the decryption process, the encrypted image (or cipher-image) is also divided into 256-bit blocks. The 256-bit cipher-image is entered to RCBC decryption algorithm and the same expanded secret key is used to decrypt the cipher-image but the expanded secret key is applied in a reverse manner. Then the next 256-bit cipher-image block follows it, and so on with the same scan path.

Fig.4 shows Encryption/Decryption throughput for RC5, RC6 and the RCBC with electronic codebook mode "ECB mode [13, 14]"; as an example; at w=32, r=16, b=16. It is clear that RCBC leads to high throughput in both encryption and decryption processes.

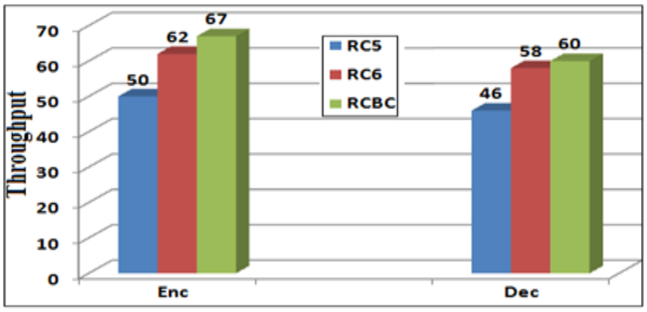


Fig.4 Enc/Dec throughput for RC5, RC6 and RCBC with ECB mode at w=32, r=16, b=16.

V. RCBC SECURITY ANALYSIS AND TEST RESULTS

The crucial measure for the quality of a crypto-system is its capability to withstand the attempts of an unauthorized participant opponent to gain knowledge about the unencrypted information. This measure is called security. The discussion of the security for discrete value crypto-systems is based on a model which was first introduced in [13] and was extended later in [14, 15]. To a certain extent, the resistance against attacks is a good measure of the performance of a crypto-system. So, it is often used to evaluate crypto-systems. The security of RCBC is estimated for digital images, under brute-force attack, statistical and differential attacks. It is shown that RCBC is secure from the strongly cryptographic viewpoint. Results show satisfactory security of the RCBC, as demonstrated in the following subsections. The security analysis results on this scheme are described, including the most important ones, like key space analysis, information entropy, statistical analysis, and differential analysis. The evaluation consisted of both theoretical derivations and practical experimentation.

A. Key Space Analysis

Key space size is the total number of different keys that can be used in the encryption. A good cipher should be sensitive to the cipher keys, and the key space should be large enough to make brute-force attacks infeasible. The RCBC is a 256-bit encryption scheme whose key space size is in the range (0-2040) bit. An exhaustive key search will take 2^k operations to succeed, where k is the key size in bits. This attack needs a few known plainimage-cipherimage pairs. An attacker simply tries all keys, one by one, and checks whether the given plainimage encrypts to the given cipherimage. For a block cipher with k-bit key and n-bit blocks, the number of pairs of images needed to determine the key uniquely is approximately [k/n] as shown in [6]. The key space size should be large enough to prevent such exhaustive searching. For practical use of RCBC, assume that the secret key length is 128-bit. Therefore, an opponent may try to bypass guessing the key and directly guessing all the possible combinations will need about 2^{128} operations to successfully determine the key. If an opponent employs a 3000 MIPS computer to guess the key by brute-force attack, the computational load can be calculated as:

$$\frac{2^{128}}{3x1000x10^{6}\,x\,60x60x24x365} > 3.5967610333\,x\,10^{21}\ \text{years} \qquad (1)$$

This is a very long time. No image can be closed-door after such years which are practically infeasible.

B. Information Entropy Analysis

Information theory is the mathematical theory of data communication and storage founded in [15-17]. Modern information theory is concerned with error-correction, data compression, cryptography, communications systems, and related topics. To calculate the entropy $H(m)$ of a source m , we have:

$$H(m) = \sum_{i=0}^{2^N-1} p(m_i)\log_2 \frac{1}{p(m_i)} \quad bits, \qquad (2)$$

And $p(m_i)$ represents the probability of symbol m_i and the entropy is expressed in bits. Suppose that the source emits 2^8 symbols with equal probability, i.e.

$m=\{m_1,\ m_2,...,\ m_8\}$. After evaluating Eq.2, we obtain its entropy $H(m)=8$, corresponding to a truly random source. Actually, given that a practical information source seldom generates random messages, in general its entropy value is smaller than the ideal one. However, when the messages are encrypted, their entropy should ideally be 8. If the output of such a cipher emits symbols with entropy less than 8, there exist certain degree of predictability, which threatens its security.

Let us consider the ciphertext of image encryption using the RCBC, the number of occurrence of each ciphertext block is recorded and the probability of occurrence is computed. The entropy is as follows:

$$H(m)=\sum_{i=0}^{2^N-1} p(m_i)\log_2\frac{1}{p(m_i)} = \sum_{i=0}^{255} p(m_i)\log_2\frac{1}{p(m_i)} = 7.9977\approx 8 \qquad (3)$$

The value obtained is 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.

C. Cipher Cycle Analysis

As a general requirement for all the image encryption schemes, the encrypted image should be greatly different from its original form. To quantify this requirement, two measures, including the number of pixels change rate (NPCR) and unified average changing intensity (UACI) can be adopted [18-20]. The value $NPCR_{R,G,B}$ is used to measure the number of pixels in difference of a color component in two images. Let $C(i,\ j)$ and $C'(i,\ j)$ be the ith row and jth column pixel of two images C and C', respectively, the $NPCR_{R,G,B}$ can be defined as:

$$NPCR_{R,G,B}=\frac{\sum_{i,j} D_{R,G,B}(i,j)}{N}\times 100\%, \qquad (4)$$

And N is the total number of pixels in the image and $D_{R,G,B}(i,\ j)$ is defined as:

$$D_{R,G,B}(i,j)=\begin{cases}0, & C_{R,G,B}(i,j)=C'_{R,G,B}(i,j)\\ 1, & C_{R,G,B}(i,j)\neq C'_{R,G,B}(i,j)\end{cases} \qquad (5)$$

And $C_{R,G,B}(i,\ j)$ and $C'_{R,G,B}(i,\ j)$ are the values of the corresponding color component Red (R), Green (G) or Blue (B) in the two images, respectively.

Considering two random images, the expected value of $NPCR_{R,G,B}$ according to [6] is found to be:

$$\varepsilon\left[NPCR_{R,G,B}\right]=(1-2^{-L_{R,G,B}})\times 100\% \qquad (6)$$

And $L_{R,G,B}$ is the number of bits used to represent the color component of red, green or blue. For example, for two random images with 512 × 512 pixels and 24-bit true color 8 bit for each RGB color component $L_R=L_G=L_B=8$ and hence:

$$\varepsilon\left[NPCR_R\right]=\varepsilon\left[NPCR_G\right]=\varepsilon\left[NPCR_B\right]=99.609375\%. \qquad (7)$$

The quantity $UACI_{R,G,B}$ is used to measure the average intensity differences in a color component and can be defined as:

$$UACI_{R,G,B}=\frac{1}{N}\left[\sum_{i,j}\frac{\left|C_{R,G,B}(i,j)-C'_{R,G,B}(i,j)\right|}{2^{L_{R,G,B}}-1}\right]\times 100\%, \qquad (8)$$

In the case of two random images, the expected value of $UACI_{R,G,B}$ can be computed as:

$$\varepsilon\left[UACI_{R,G,B}\right]=\left[\frac{\frac{1}{2^{2L_{R,G,B}}-1}\left(\sum_{i=1}^{2^{L_{R,G,B}}-1} i(i+1)\right)}{2^{L_{R,G,B}}-1}\right]\times 100\% \qquad (9)$$

Assuming each color component is coded with 8 bits, then:

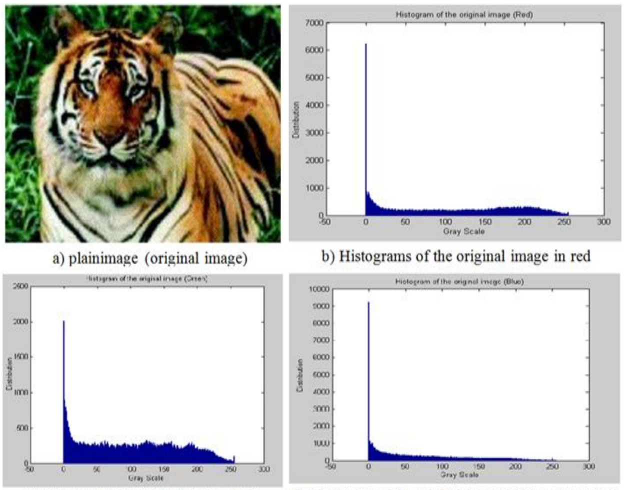
$$\varepsilon\left[UACI_R\right]=\varepsilon\left[UACI_G\right]=\varepsilon\left[UACI_B\right]=0.3346354 \qquad (10)$$

D. Statistical Analysis

In order to resist the statistical attacks, which are quite common nowadays, the encrypted images should possess certain random properties. To prove the robustness of the RCBC, we have performed statistical analysis by calculating the histograms and the correlations of two adjacent pixels in the plainimage/cipherimage. Different images have been tested, and similar results are obtained. The results are summarized in the following sections.

1) Color Histograms Analysis

An image-histogram illustrates how pixels in an image are distributed by graphing the number of pixels at each color intensity level. In order to appear random, the color histograms of the encrypted image should be uniform distributed in all three color components (RGB). We have calculated and analyzed the histograms of the several encrypted as well as its original colored images that have widely different content. One example of such histogram analysis is shown in Figs. 5 and 6. Particularly in Fig. 5a, we have shown the original image and in Figs. 5b, c and d respectively, the histograms of red, blue and green channels of the original image in Fig.5a. In Fig. 6a, we have shown the encrypted image of the original image in Figs 6a and in Figs 6b, c and d respectively, including the histograms of red, blue and green channels of the encrypted image 6a. It is clear from Fig. 6 that the histograms of the encrypted image are fairly uniform and significantly different from the respective histograms of the original image, and hence does not provide any clue to employ any statistical attack on the RCBC.



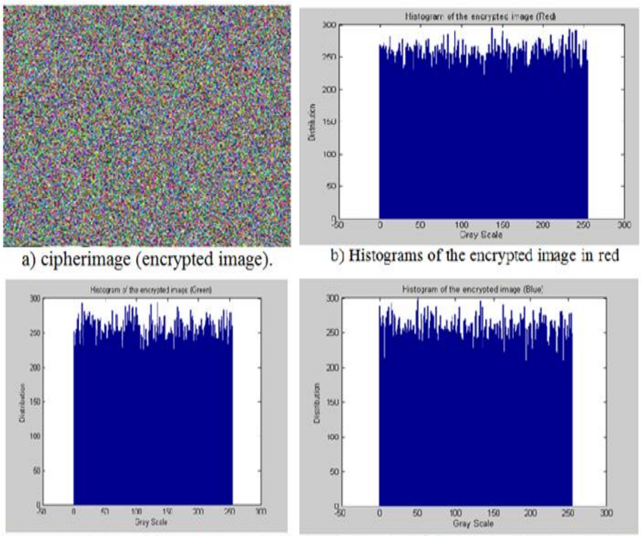
a) plainimage (original image)

b) Histograms of the original image in red

c) Histograms of the original image in green

d) Histograms of the original image in blue component.

Fig. 5 Histograms of the original image in (b) red, (c) green, and (d) blue components.



a) cipherimage (encrypted image).

b) Histograms of the encrypted image in red

c) Histograms of the encrypted image in green

d) Histograms of the encrypted image in blue

Fig. 6 Histograms of the encrypted image in (b) red, (c) green, and (d) blue components.

2) Correlation Coefficient Analysis

In addition to the histogram analysis, we have also analyzed the correlation between two vertically adjacent pixels, two horizontally adjacent pixels and two diagonally adjacent pixels in the several images and their encrypted images. For an ordinary image, each pixel is usually highly correlated with its adjacent pixels either in horizontal, vertical or diagonal directions. These high-correlation properties can be quantified as the correlation coefficient for comparison. The procedure for calculating correlation coefficient is as follows: First, randomly select 1000 pairs of two adjacent pixels from an image. Then, calculate their correlation coefficient using the following two formulas:

$$\operatorname{cov}(x,y) = E((x-E(x))(y-E(y))), \quad (11)$$

$$r_{xy} = \frac{\operatorname{cov}(x,y)}{\sqrt{D(x)}\sqrt{D(y)}}, \quad (12)$$

The quantities x and y are the values of two adjacent pixels in the image. In numerical computations, the following discrete formulas are used:

$$E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i \quad (13)$$

$$D(x) = \frac{1}{N}\sum_{i=1}^{N} (x_i - E(x))^2, \quad (14)$$

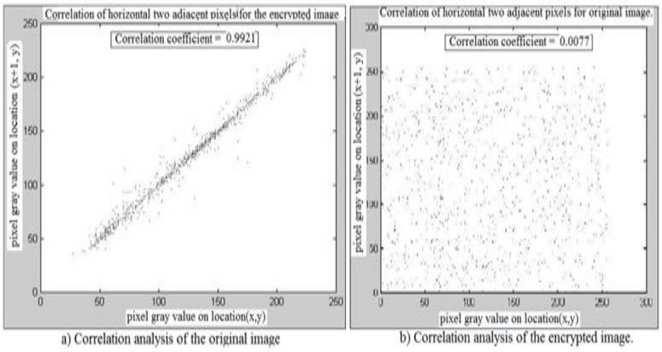
$$\operatorname{cov}(x,y) = \frac{1}{N}\sum_{i=1}^{N} (x_i - E(x))(y_i - E(y)), \quad (15)$$

In Fig. 6, we have outlined the distribution of two adjacent pixels in the original and encrypted images, as shown in Fig. 5a and Fig. 6a. Particularly, in Fig. 7a and Fig. 7b, we have depicted the distributions of two horizontally adjacent pixels in the original and encrypted images, respectively. Similarly, in Fig. 8a and Fig. 8b respectively, the distributions of two vertically adjacent pixels in the original and encrypted images have been depicted.

In Table III, we have given the correlation coefficients for the original and encrypted images shown in Fig. 6a and Fig. 7a respectively. It is clear from Figs. 7 and 8 and Table III that the encrypted image obtained from the RCBC retains small correlation coefficients (there is negligible correlation between the two adjacent pixels in the encrypted image) in all directions. However, the two adjacent pixels in the original image are highly correlated.

Table III Correlation coefficients in plainimage/cipherimage.

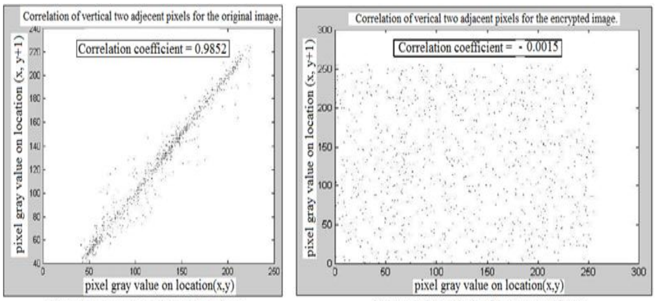
Direction of Adjacent pixels	Plainimage	Cipherimage
Horizontal	0.9921	0.0077
Vertical	0.9852	- 0.0015
Diagonal	0.9768	- 0.0064



a) Correlation analysis of the original image

b) Correlation analysis of the encrypted image.

Fig.7 the Correlations of two horizontally adjacent pixels in the original image and in the encrypted image.



a) Correlation analysis of the original image.

b) Correlation analysis of the encrypted image.

Fig.8 the Correlations of two vertically adjacent pixels in the original image and in the encrypted image.

E. Sensitivity Analysis

An ideal encryption procedure should be sensitive with respect to both the secret key and plain-image. The used crypto-system RCBC has high key and plaintext sensitivities. This means that a slight change in the key or in the plaintext will causes great changes in the ciphertext. These properties make various sensitivity-based (differential) attacks difficult. To prove the robustness of the RCBC, we will perform sensitivity analysis with respect to both key and plaintext.

1) Key Sensitivity Analysis

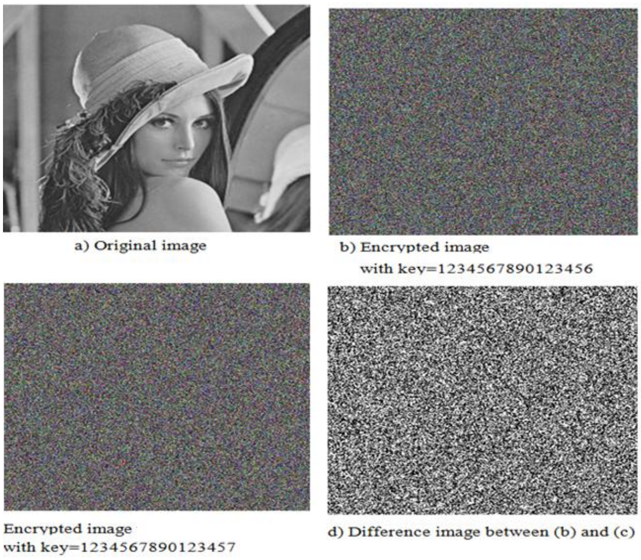
An encryption scheme has to be key-sensitive, meaning that a tiny change in the key will cause a significant change in the output. Assume that a 16-character (128-bit) ciphering key is used. For testing the key sensitivity of the RCBC encryption procedure, we have performed the following steps:

a- First, a 512x512 image is encrypted using the test key "1234567890123456".

b- Then, the least significant bit of the key is changed, so that the original key becomes, say "1234567890123457" in this example, which is used to encrypt the same image.

c- Finally, the above two ciphered images, encrypted by the two keys, are compared.

The result of key sensitivity analysis shows that changing one bit in encryption key will result in a completely different cipher-image by more than 99% in terms of pixel grey scale values. Fig. 9 shows the test results. High key sensitivity is required by secure crypto-systems, which means that the cipher-image cannot be decrypted correctly although there is only a slight difference between encryption and decryption keys. This guarantees the security of the RCBC against brute-force attacks. So, when a 16-character key is used to encrypt an image while another trivially modified key is used to decrypt the ciphered image, the decryption also completely fails. Fig. 10 has verified this, and it clearly shows that the image encrypted by the key "1234567890123456" is not correctly decrypted by using the key "1234567890123457", which has also only one bit difference between the two keys. It is clear that the decryption with a slightly different key, fails completely and hence the RCBC is highly key sensitive.


a) Original image

b) Encrypted image with key=1234567890123456

Encrypted image with key=1234567890123457

d) Difference image between (b) and (c)

Fig. 9 Key Sensitive test result_1 with RCBC-32/16/16.

2) Plain-image Sensitivity Analysis

In general, the opponent may make a slight change such as modifying only one pixel of the original image, and then observes the change of the result. In this way, we may be able to find out a meaningful relationship between the plain-image and the cipher-image. If one minor change in the plain-image can cause a significant change in the cipher-image, then this differential attack would become very inefficient and practically useless. In order to avoid the known-plaintext and the chosen-plaintext attacks (differential attacks), the changes in the cipher image should be significant even with a small change in the original one. According to the RCBC encryption process, this small difference should be diffused to the whole ciphered data. They can in fact be reflected by the $NPCR_{R,G,B}$.

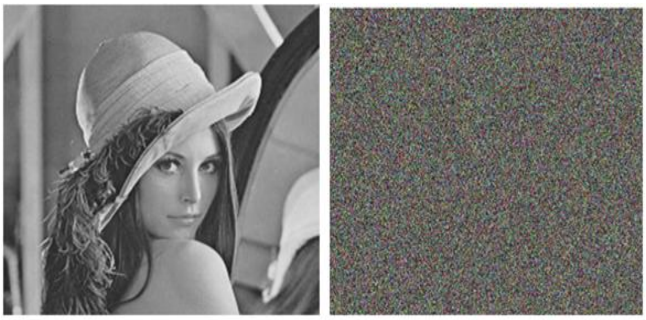


Fig. 10 Key Sensitive test result_2 with RCBC-32/16/16.

A desirable property for the RCBC is that it is highly sensitive to small change in the plain-image (single bit change in plain-image). The average pixel differences of some well-known images are computed for the two encrypted images that are their plain-images have only one-bit change. The results are tabulated in Table IV.

Table IV Encrypted images with 1-bit Pixel difference in their plain-images.

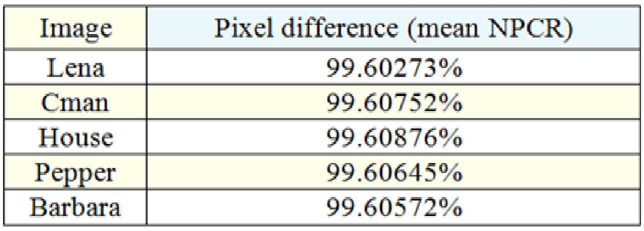
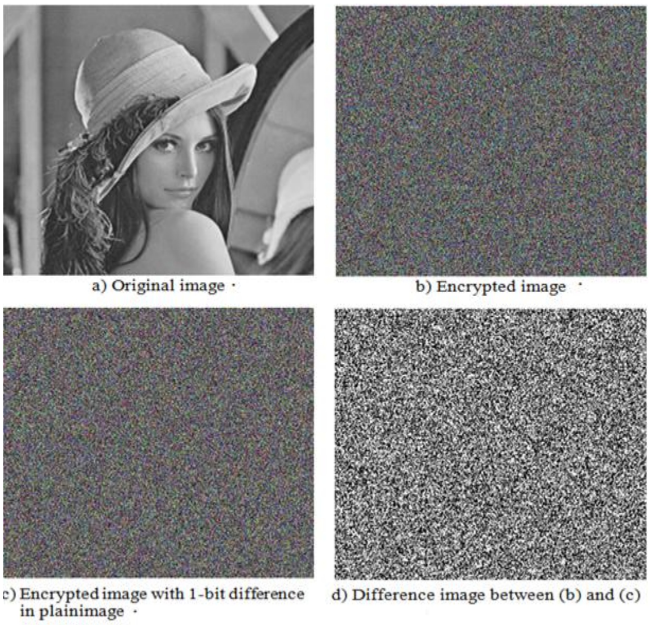


Image	Pixel difference (mean NPCR)
Lena	99.60273%
Cman	99.60752%
House	99.60876%
Pepper	99.60645%
Barbara	99.60572%

It can be observed that the values are very close to the expected value of pixel difference on two randomly generated images (99.609375%). Fig. 11 shows an example of two enciphered images from two plain-images with only 1-bit difference generated using the RCBC. The original and encrypted images are shown in Figs. 11a and 11b, respectively, Fig. 11c is the encrypted image with only one-bit change in the original image (a), while (d) is the difference-image between the two encrypted images: (b) and (c). As can be seen, most of the pixels in Fig. 11d are nonzero, which means that the difference between image (b) and image (c) is big enough. Thus, the RCBC has high plain-image sensitivity.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication



a) Original image

b) Encrypted image

c) Encrypted image with 1-bit difference in plainimage

d) Difference image between (b) and (c)

Fig.11 Plaintext Sensitivity test with RCBC-32/16/16.

VI. CONCLUSION

In this paper an efficient robust chaotic block cipher (RCBC) was examined and compared with both RC5 and RC6. The implementation of the cipher was analyzed and evaluated for digital images and its providing security. Experimental tests are carried out with detailed analysis, demonstrating the high security and throughput of the RCBC. So, the RCBC can be considered as near to a real-time fast and secure symmetric encryption for digital images. Security estimation for digital imaging against brute-force, statistical, and differential attacks is explored from strict cryptographic viewpoint. Thorough experimental tests are carried out with detailed analysis, demonstrated the high security of the cipher.

ACKNOWLEDGMENT

Authors would like to thank King Abdulaziz University in Jeddah, Saudi Arabia for offering the necessary facilities for accomplishing this research.

REFERENCES

1. A. Mariam Babu and K. J. Singh,” Performance Evaluation of Chaotic Encryption Technique”, American Journal of Applied Sciences, 10 (1): 35-41, 2013.
2. M.Surya B. Rao, V.S. G. Akula,” Chaotic Algorithms used for Encryption and Decryption on Moving Images”, International Journal of Advances in Computer Science and Technology, Volume 2, No.8, August 2013.
3. I. Bremnavas1, B.Poorna2 and I. R. Mohamed,” Secured medical image transmission using Chaotic map”, Elixir Comp. Sci. Eng. 54, 2013.
4. R. K. Purwar,” An Improved Image Encryption Scheme Using Chaotic Logistic Maps”, International Journal of Latest Trends in Engineering and Technology (IJLTET), Vol. 2 Issue 3 May 2013.
5. O. S. Farag Allah, “An enhanced chaotic key- based RC5 block cipher adapted to image encryption”, International Journal of Electronics, Volume 99, Issue 7, 2012.
6. H. H. Ahmed, H. M. Kalash, and O. S. Farag Allah,” Implementation of RC5 Block Cipher Algorithm for Image Cryptosystems”, International Journal of Information Technology, Vol. 3 No. 4, 2007.
7. A. H. M. Ragab, and N. A. Ismail, O. S. Farag Allah, “Enhancements and Implementation of RC6 Block Cipher for Data Security". IEEE Catalog Number: 01CH37239, 2001.
8. A. B. Mohamed1, Ghada Zaibi1, A. Kachouri,” Implementation of RC5 and RC6 block ciphers on digital”, The 8th International Multi-Conference on Systems, Signals & Devices,2011.
9. S. Lian, "A block cipher based on chaotic neural networks," Neuron-computing, doi:10.1016/j.neucom.2008.
10. S. Lian, "Efficient image or video encryption based on spatiotemporal chaos system," chaos, solutions and fractals, 2007.
11. S. Behnia, A. Akhshani, A. Akhavan, H. Mahmodi, "Applications of tripled chaotic maps in cryptography," Chaos, Solutions and Fractals, 2007.
12. O. S. Faragallah,”An Efficient Block Encryption Cipher Based on Chaotic Maps for Secure Multimedia Applications”, *Information Security Journal: A Global Perspective*, 20:135–147, 2011.
13. **ECB Mode,** http://www.cryptopp.com /wiki/ ECB_Mode.
14. K.T. Huang. , J.H. Chiu. and S. S. Shen,” a novel structure with dynamic operation mode for symmetric-key block ciphers” International Journal of Network Security & Its Applications (IJNSA), Vol.5, No.1,17-36, January 2013.
15. C.E. Shannon, “Communication Theory of Secrecy Systems,” Bell System Technical Journal, vol. 28, No. 4, pp. 656-715, October 1949.
16. S. Li, C. Li, G. Chen, K.T. Lo, “Crypto-analysis of the RCES/RSES image encryption scheme,” The journal of systems and software 811130-1143, 2008.
17. C. Li , S. Li, G. Alvarez , G. Chen, K.T. Lo, "Cryptanalysis of a chaotic block cipher with external key and its improved version," Chaos, Solutions and Fractals 37,299-307, 2008.
18. H. H. Ahmed, H. M. Kalash, and O. S. Farag Allah, "An Efficient Chaos-Based Feedback Stream cipher (ECBFSC) for Image Encryption and Decryption”, An International Journal of Computing and Informatics, Vol. 31, No. 1, PP. 121-129, ISSN 0350-5596, 2007.
19. T. Xiang, K.Wong, X. Liao, "An improved chaotic cryptosystem with external key", Communication in Nonlinear Science and Numerical Simulation 13, 1879-1887, 2008.
20. M. Amin, O. S. Faragallah, A. A. Abd El-Latif, "A Chaotic Block Cipher Algorithm for Image Cryptosystems," Journal of Communications in Nonlinear Science and Numerical Simulation, 2010.

AUTHORS PROFILE

Prof: Abdul Hamid M. Ragab, He got his PhD from Essex University, UK, in 1985 in e Systems. He is Prof. since 1995, Academic Staff Member & Consultants & Chairman of Computer Science and Eng. Referee for scientific Journals and Conferences. Supervised hundreds of PhD and Msc thesis. He has several highly cited Articles in IEEE Journals His research interests include: Multilevel Network Security, e-systems Applications, Adaptive E-Learning Systems, and Developed DSS. His eMail: ahm_ragab@yahoo.com

Assoc. Prof: Osama S. Farag Allah

He is Associate Prof in Computer Science & Engineering. He got his Ph.D. in Computer Science & Engineering in 2007 from Menoufia University. His research interests cover Computer networks, Network security, Cryptography, Internet Multimedia security, Image encryption, Watermarking, Steganographic, Data hiding, Chaos theory. He published several papers in these fields, and supervised many Msc and PhD Thesis. His eMail:osam_sal@yahoo.com

Assist. Prof: Khalid W. Magld, In 2007 he got his PhD in Computer Science University of Bradform, UK. Field of specialty: “Networking, and Information Systems Analysis and Applications”. He has several published work in this fields. His Research Interests: Unicast and multicast in mobile ad-hoc networks. Neural networks analysis and applications. Web-based simulation, training and education, E-Learning and applications. E-Government, e-Management Systems. His eMail: kmagld@kau.edu.sa

Assoc. Prof: Amin Y. Noaman, His Ph.D. in computer science from University of Manitoba, Canada, in 1999.He published many papers in the field, and supervised many Msc and PhD thesis. Currently he is associate professor in the computer science dept., faculty of computing and information technology, King Abdulaziz University. His current research focuses on data warehousing, bioinformatics, distributed database systems, DSS, mobile database and E-Learning. His eMail: anoaman @kau.edu.sa.