*Journal of Engineering Sciences, Assiut University, Vol. 34, No. 5, pp. 1575-1590, Sept. 2006*

# A MODIFICATION OF THE BLUETOOTH E0 STREAM CIPHER

***Ibrahim El-Docanny ; Nawal El-Fishawy and Eman Soltan***
*Department of Electronics & Electrical Communications Engineering, Faculty of Electronics Engineering, Menouf, Egypt*
*e-mail: nelfishawy@hotmail.com or eman_soltan@yahoo.com*

*(Received July 4, 2006 Accepted September 5, 2006)*

***ABSTRACT–*** *In this paper we propose an modified model for the E0 stream cipher, which is the Encryption system used in the Bluetooth specification. The modified model is manipulated by adding a fifth linear feed back shift register (LFSR) to the main key stream generator of the E0 cipher. The contribution of this modification is indicated in increasing the computational complexity of Ophir Levy and Avishai Wool attack from $2^{85}$ to $2^{125.25}$. Increasing the quality of encryption is also illustrated through different images encryption with the implementation of several measuring factors such as the correlation coefficient, the maximum deviation, and the irregular deviation.*

## 1- INTRODUCTION

Bluetooth is a new radio link which was developed to replace cables in a short range communications between Bluetooth devices [1]. The Bluetooth system specification must include cryptographic functionalities both authentication and encryption. Authentication is used to improve the user's legal identities and encryption to ensure privacy in communication. E0 stream cipher is used in Bluetooth system for data encryption, which is built around a relatively simple Key Stream Generator (KSG). Along the past few years many attacks against the E0 cipher appeared such as correlation, linearization and location attacks. Fluhrer and Luck have shown in [2] an optimized backtracking method to recover the secret key with a computational complexity of $2^{73}$ if $2^{43}$ bit key stream is available. Correlation attacks are based on a relation between the stream cipher output and one or more one of the inputs [3 – 5]. Linearization attack as in [6] showed that the initial value can be recovered by solving a system of nonlinear equations, which were transformed by linearization into a system of linear equations. Tracking the Bluetooth devices was showed by Jakobsson in [7].
In this paper we are concerned with Ophir Levy and Avishai Wool attack, which recovered the initial state of the LFSRs by solving $2^{86}$ Boolean linear system of equations [8]. Also in this paper the quality of encrypted images with the original and the modified cipher are tested with different quality measuring factors.

The paper is organized as follows. Section 2 describes the construction and the operation of the E0 stream cipher. Section 3 the proposed modified KSG is described. Section 4 Ophir Levy and Avishai Wool attack is applied to our new cipher. Section 5 gives more details about the images encryption and the methods of judging the quality of their encryption. In section 7 the paper is concluded.

## 2- THE DESCRIPTION OF THE ORIGINAL E0 STREAM CIPHER.

When two Bluetooth devices need to communicate securely, they first must agree on a shared secret link key which is used to generate the encryption key. This encryption key is combined with a public random value to generate the constraint link key $K_c$. The $K_c$ is used linearly with a 48 bit Bluetooth device address and a 26 bit clock master, which is distinct for each packet, to form the initial states of a two level key stream generator. More details describing of the Bluetooth encryption system can be found in [1].

The KSG consists of four LFSRs with lengths of 25, 31, 33, and 39, and a 4 bit finite state machine (called the Summation Combiner Logic and Blend, SCLB). The connecting polynomials of the four LFSRs are of Hamming weight 5, that in every LFSR, five (different) stages are used to produce a new LSB (Least Significant Bit) at every clock [9].

The connecting polynomial of the four LFSRs are:

| | | |
|---|---|---|
| $LFSR_1$: | $f_1(t) = t^{25} + t^{20} + t^{12} + t^8 + 1$ | output tape is bit 24 |
| $LFSR_2$: | $f_2(t) = t^{31} + t^{24} + t^{16} + t^{12} + 1$ | output tape is bit 24 |
| $LFSR_3$: | $f_3(t) = t^{33} + t^{28} + t^{24} + t^4 + 1$ | output tape is bit 32 |
| $LFSR_4$: | $f_1(t) = t^{39} + t^{36} + t^{28} + t^4 + 1$ | output tape is bit 32 |

For each output bit, each LFSR clocked once, and their output bits are XORed together with one bit of the finite state machine to produce the key stream bit. Then the output of the 4 LFSRs are summed together. The two most significant bits of this sum are used to update the state of the finite state machine [8], [2].

**Figure 1** shows the complete Bluetooth ciphering algorithm (E0). $Z^{-1}$ is a label for one delay element and $T_1$and $T_2$ are two different bijections over $IF_2^2$ where $T_1(x_1,x_0)= (x_1,x_0)$ and $T_2(x_1,x_0)= (x_0,x_1 \oplus x_0)$. $x_t^1$, $x_t^2$, $x_t^3$, $x_t^4$ are the outputs of the four LFSRs at time t. $z_t$ is the output of the key stream which is defined from the following equation:

$$z_t = x_t^1 \oplus x_t^2 \oplus x_t^3 \oplus x_t^4 \oplus c_t^0$$

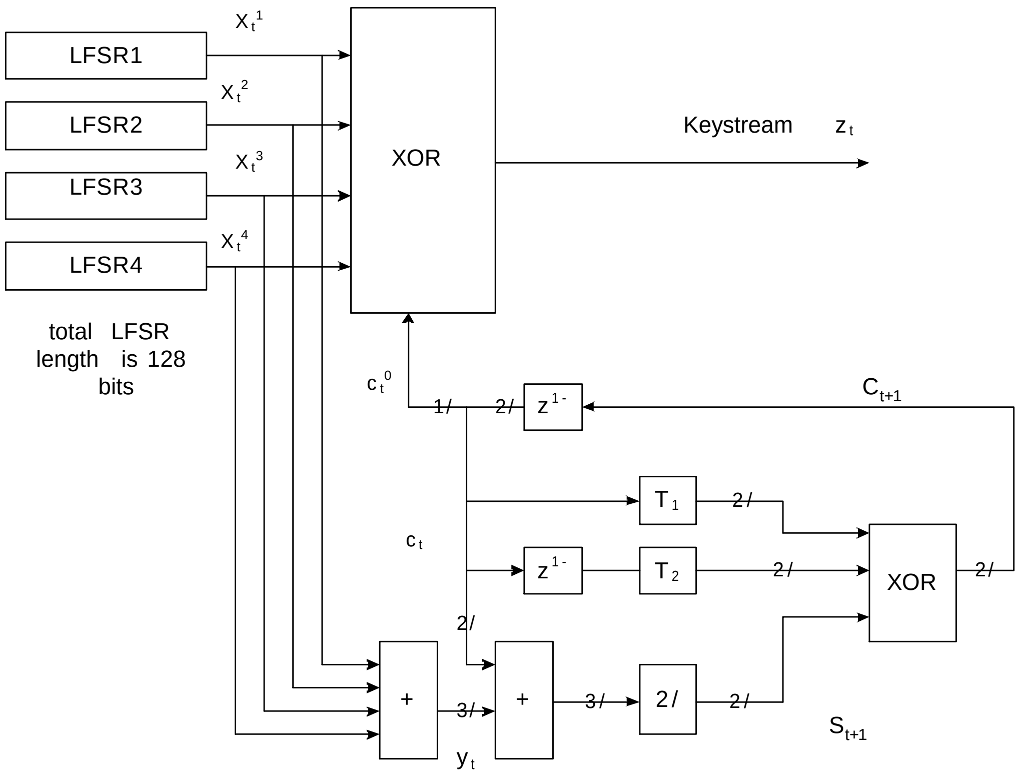
$$s_{t+1} = (s_{t+1}^1, s_{t+1}^0) = [(y_t + c_t)/2]$$

$$y_t = x_t^1 + x_t^2 + x_t^3 + x_t^4$$

$$c_{t+1} = (c_{t+1}^1, c_{t+1}^0) = (s_{t+1}^1, s_{t+1}^0) \oplus T_1(c_t) \oplus T_2(c_{t-1})$$

Where $y_t$ Є (0,1,2,3,4}and $S_t$ Є {0,1,2,3}. $(s_t^0,s_t^1)$ is a binary vector where $0 \longrightarrow (0,0)$ and $1 \longrightarrow (0,1)$, etc.

The implementation of the encryption algorithm is of two main levels, the first level is the initialization level which defines the initial contents of the LFSRs and the $c_0$ and $c_{-1}$ initial states. The second level is the main algorithm operation to generate the key stream output.



The small number indicates the number of the bits in the wire [10].

**Fig. 1:** The Bluetooth key stream generator (E0).

## 2-1 Initialization

The key stream generator needs to be loaded by the initial values of the four LFSRs (altogether 128 bits) and the 4 bits that specify the values of $c_0$ and $c_{-1}$. The 132 initial bits are derived from the constraint key ($k_c$) of length 128 bits, the 48 bits Bluetooth device address (ADR) and the 26 bits master clock (CLK). With the 6-bits(decimal) constant 113 (208 bits total). All the 208 bits will be loaded to the key stream generator to produce 200 bits output symbols, the last 128 bits generated will be fed back in parallel into KSG to form the initial contents of the LFSRs and the values of $c_0$ and $c_{-1}$ are kept. The initialization details will be as follows:

1- Set the initial states of all the LFSRs to zeros (t =0).
2- Open all the switches of the feed back path.
3- Arrange the input as shown in **Fig. 2**, ADR [I] denotes the ith byte of the BD-ADDR, $K_c$ [I] denotes also the ith byte of the encryption key and similarly CLK [I].
4- Start shifting the input bits in parallel in the four LFSRs.
5- Close the switch of the LFSR1 after 25 clock cycle, that of LFSR2 after 31 clock cycle, that of LFSR3 after 33 clock cycle and that of LFSR4 after 39 clock cycle.
6- At t = 39 set $c_{39}$=0 and $c_{38}$=0.

7- Continue shifting the remaining input bits until 200 output bits are produced.
8- Keep $c_t$ and $c_{t-1}$ and load the last generated 128 output bits in the four LFSRs by the order shown in **Fig. 3**.

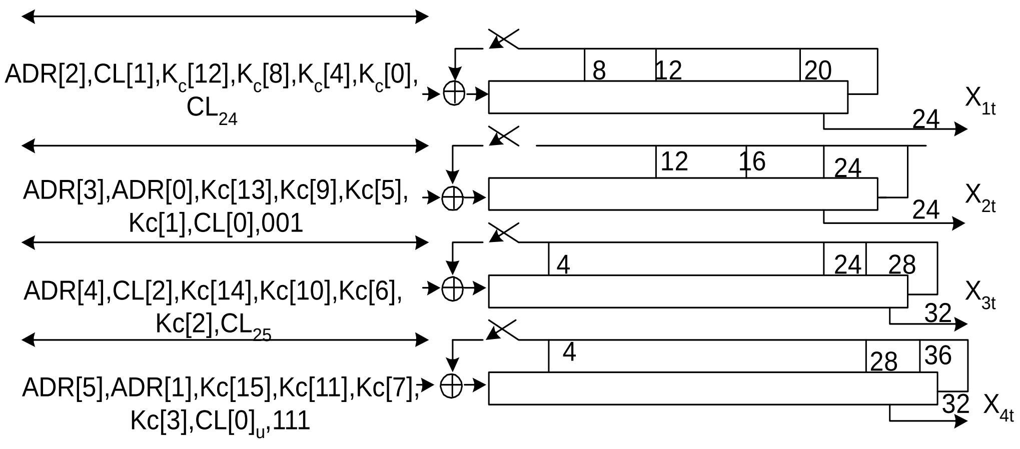

ADR[2],CL[1],Kc[12],Kc[8],Kc[4],Kc[0], CL24
8
12
20
24
X1t
ADR[3],ADR[0],Kc[13],Kc[9],Kc[5], Kc[1],CL[0],001
12
16
24
24
X2t
ADR[4],CL[2],Kc[14],Kc[10],Kc[6], Kc[2],CL25
4
24
28
32
X3t
ADR[5],ADR[1],Kc[15],Kc[11],Kc[7], Kc[3],CL[0]u,111
4
28
36
32
X4t

**Fig. 2:** The initial loading of the four LFSRs [10 ].

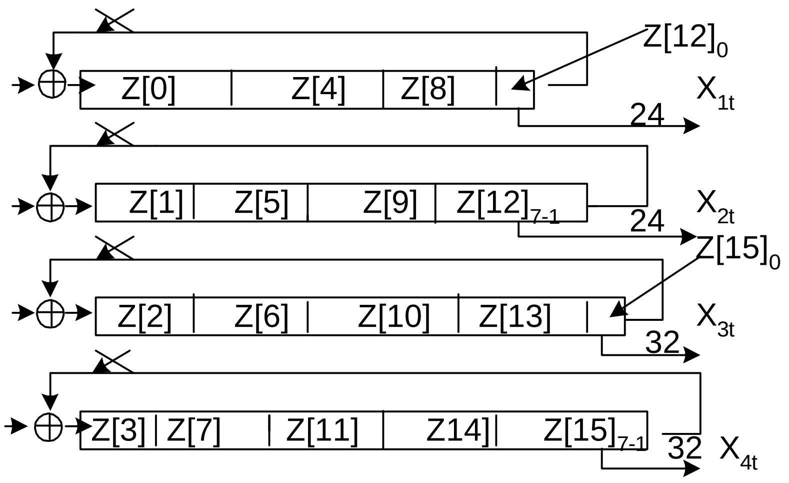


**Fig. 3:** The Second loading of the LFSRs with 128 last generated bits from the initialization process.

## 2-2 Operation

The 128 bits generated in the initialization level will represent the initial contents of the LFSRs then the generator will operate with the same sequence as before to generate the output key steam that will be used to encrypt the plaintext by simple XOR (addition module 2).

## 3- THE MODIFIED E0 STREAM CIPHER

Our proposed modified E0 stream cipher is based on adding a fifth LFSR to the KSG. This modification sustains more enforcement against uniform frame work attacks and increasing the quality of encryption.

The fifth added LFSR is supposed to be of 40 bits length and its connecting polynomial is:

$$LFSR_5 = t^{40} + t^{35} + t^{30} + t^{8} + 1 \qquad \text{the output tap is bit 39}$$

According to the new modified algorithm, **Fig. 1** will be modified by adding the fifth LFSR and the inputs to the adder will be the outputs of the five LFSRs. The output of the key stream will be defined by the following modified equations:

$$z_t = x_t^1 \oplus x_t^2 \oplus x_t^3 \oplus x_t^4 \oplus x_t^5 \oplus c_t^0$$
$$s_{t+1} = (s_{t+1}^1, s_{t+1}^0) = [(y_t + c_t)/2]$$
$$y_t = x_t^1 + x_t^2 + x_t^3 + x_t^4 + x_t^5$$
$$c_{t+1} = (c_{t+1}^1, c_{t+1}^0) = (s_{t+1}^1, s_{t+1}^0) \oplus T_1(c_t) \oplus T_2(c_{t-1})$$

Where $y_t$ Є (0,1,2,3,4,5)and $S_t$ Є (0,1,2,3,4). ($s_t^0$, $s_t^1$) is a binary vector where
$0 \longrightarrow (0,0)$ and $1 \longrightarrow (0,1)$etc.
The initial contents of the fifth added LFSR will be chosen by two different ways, which gives us two modified possibilities for the E0 cipher.

## A - The First New Modified Cipher (E0_1)

The initial content of the LFSR5 will be composed of the same inputs to the original cipher, which are ADDR, $K_c$, and CLK, but in a new arrangement. So the inputs of the first and third LFSRs will be 41 bits instead of 49, and the input for the second and the fourth LFSRs is 43 bits. The input for the fifth added LFSR is 40 bits. **Figure 4** indicates the initial loading of the five LFSRs for the first proposal.

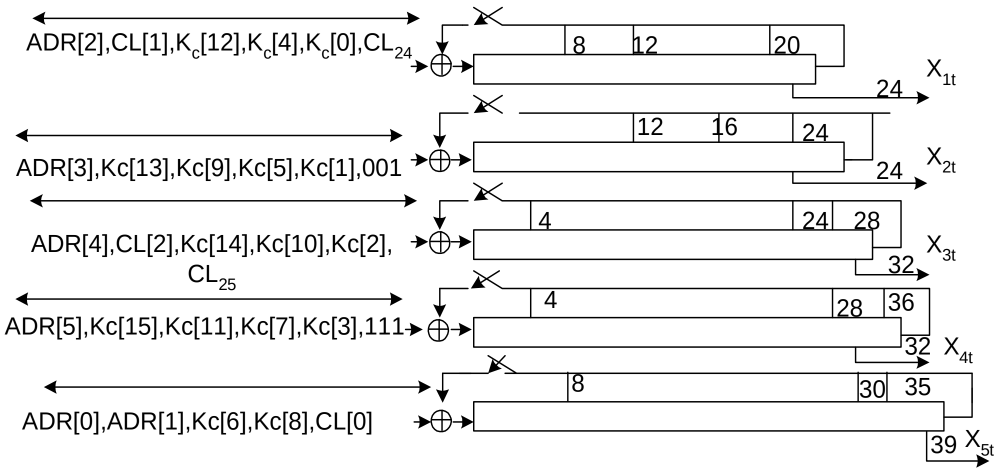

ADR[2],CL[1],Kc[12],Kc[4],Kc[0],CL24
8
12
20
24
X1t
ADR[3],Kc[13],Kc[9],Kc[5],Kc[1],001
12
16
24
24
X2t
ADR[4],CL[2],Kc[14],Kc[10],Kc[2], CL25
4
24
28
32
X3t
ADR[5],Kc[15],Kc[11],Kc[7],Kc[3],111
4
28
36
32
X4t
ADR[0],ADR[1],Kc[6],Kc[8],CL[0]
8
30
35
39
X5t

**Fig. 4:** The initial loading of the five LFSRs for the first modification.

## B-The Second New Modified Cipher (E0_2)

The initial contents of the fifth added LFSR will be chosen by adding a new input to the KSG, which is 128 bit random value (RAND). So the number of inputs to each of the LFSRs will be now increased and will have different arrangement as in **Fig. 5**. The input to first and the third LFSRs will be 65 bits and for the second and the fourth LFSRs will be 71 bits. The input to the fifth LFSR will be 64 bits.

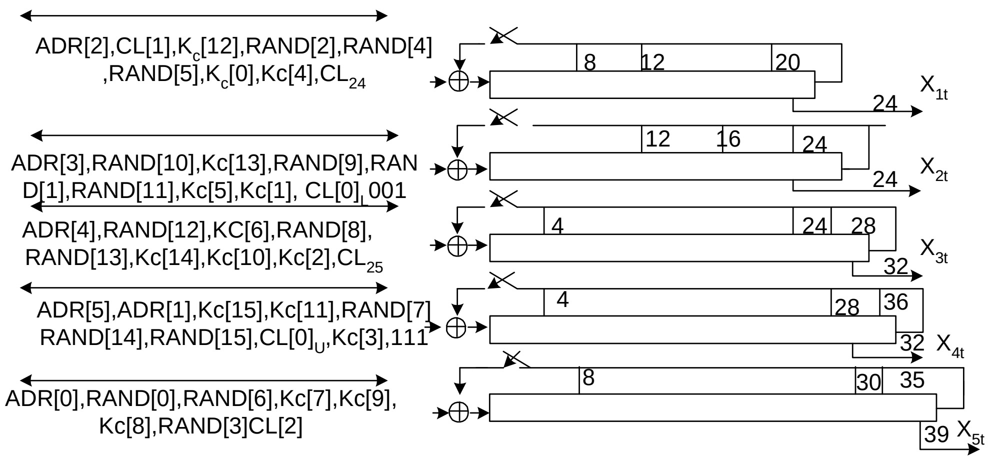

ADR[2],CL[1],Kc[12],RAND[2],RAND[4] ,RAND[5],Kc[0],Kc[4],CL24
8
12
20
24
X1t
ADR[3],RAND[10],Kc[13],RAND[9],RAND[1],RAND[11],Kc[5],Kc[1], CL[0]L001
12
16
24
24
X2t
ADR[4],RAND[12],KC[6],RAND[8], RAND[13],Kc[14],Kc[10],Kc[2],CL25
4
24
28
32
X3t
ADR[5],ADR[1],Kc[15],Kc[11],RAND[7] RAND[14],RAND[15],CL[0]U,Kc[3],111
4
28
36
32
X4t
ADR[0],RAND[0],RAND[6],Kc[7],Kc[9], Kc[8],RAND[3]CL[2]
8
30
35
39
X5t

**Fig. 5:** The initial loading of the five LFSRs for the second modification.

The key stream generator need to be loaded by the contents of the five LFSRs (altogether 168 bits) and the four bits that specify the values of $c_0$ and $c_{-1}$. The 172 initial bits are derived by the two different new modifications as shown in **Fig. 4** and **Fig. 5**. The initialization steps are the same as the original algorithm except adding the fifth LFSR into consideration in step 4 and 5. At step 6 at t= 40 set $c_{39}$=0 and $c_{40}$=0.
The generator will continue its operation to generate 200 bits, and the last 172 bits generated will be loaded to the five LFSRs as shown in **Fig. 6**.

After loading the key stream generator with its initial values, the generator will operate to produce the required key length ($Z_t$). The encryption is performed by taking the XOR of $Z_t$ and the bit sequence of the data packet. In our simulation program two values are assumed for the length of $Z_t$ which are 125 bits or 2745 bits.

## 4- THE IMPLEMENTATION OF OPHIR LEVY AND AVISHAI WOOL ATTACK TO OUR MODIFIED CIPHER

A framework for the cryptanalysis of the E0 cipher was introduced by Ophir Levy and Avishai wool for the derivation of the initial state of the E0 cipher [8].
Their attack was viewed as a generalization of the Bleichenbacher's attack and Fluhrer and Lucks attack [2]. This attack was based on guessing the initial states of some of the LFSRs, backtracking through the different possible states of the combiner logic, solving sets of linear equations, and checking for consistency with the given output stream cipher.

As declared in [8] to generate the output key stream bit ($Z_t$ )for the modified cipher we need five input bits (the outputs of the five LFSRs) and four bits that define the states of the Summation Combiner Logic and Blend.

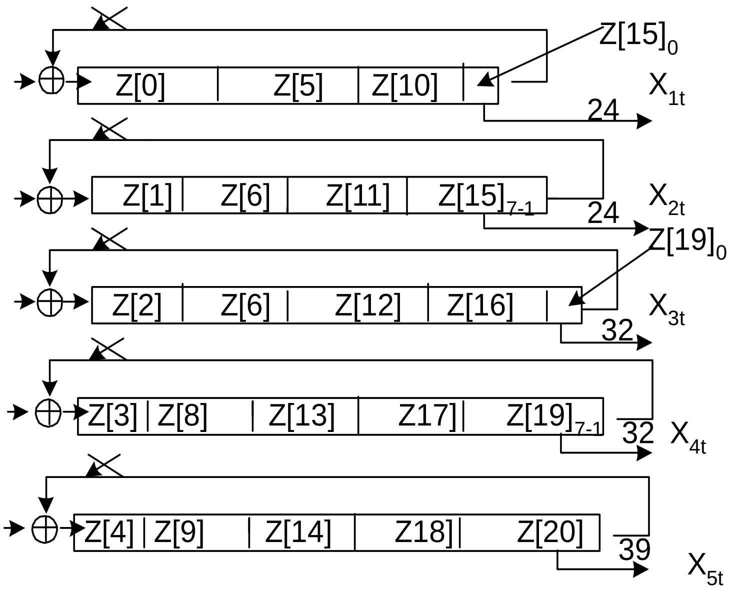


**Fig. 6:** The second loading of the LFSRs with 168 last generated bits from the initialization process.

A table will be derived as in [8] which arrange the output of the KSG ($Z_t$) to be either a “1” or “0”, according to four bit states input, (we have four bit states to get 16 combination states) . So each cell inside the table holds the next state and the output bit that is received for the relevant current state and the sum of input bits.

The sum of the outputs of the five LFSRs will be 0, 1, 2, 3, 4, and 5, and the four state bits are $(c_t^1, c_t^0, c_{t-1}^1, c_{t-1}^0)$ with $c_t^1$ as the MSB (Most Significant Bit).
Ophir Levy and Avishai Wool followed in their attack a derivation of some finding. They assumed knowing the current state of the SCLB, and the current output, then there are three possible next states. For each next state they can derive linear equations on the LFSRs initial state bits. They followed in their attack a basic model of five calculations. We will follow these calculations on in order to derive the computational complexity value.

**(1) Chains**

According to Ophir equations for a $Z_t$ output sequence, during the backtracking phase we produce a set of chains. For each chain we will get 168 linear equations of 168 unknown bits of the initial states. Solving these equations will gives us the initial states of the five LFSRs. For a given output and the guessed four initial states $(c_t^1, c_t^0, c_{t-1}^1, c_{t-1}^0)$ we will have a chain of sums according to **Table 1**. For each sum we calculate the next state. We have all the chains of equal length.

**Table 1:** The output of the KSG as function in four input state and the sum of the five inputs.

| Output bit | 0 | | 1 | |
|---|---|---|---|---|
| State | sum | Next state | sum | Next state |
| 0 | 0 | 0 | 1 | 0 |
| | 2 | 0 | 3 | 4 |
| | 4 | 4 | 5 | 4 |
| 1 | 0 | 0 | 1 | 12 |
| | 2 | 0 | 3 | 8 |
| | 4 | 8 | 5 | 8 |
| 2 | 0 | 4 | 1 | 4 |
| | 2 | 4 | 3 | 0 |
| | 4 | 0 | 5 | 0 |
| 3 | 0 | 5 | 1 | 8 |
| | 2 | 1 | 3 | 12 |
| | 4 | 13 | 5 | 12 |
| 4 | 1 | 19 | 0 | 5 |
| | 3 | 13 | 2 | 1 |
| | 5 | 1 | 4 | 1 |
| 5 | 1 | 9 | 0 | 9 |
| | 3 | 13 | 2 | 13 |
| | 5 | 1 | 4 | 13 |
| 6 | 1 | 1 | 0 | 1 |
| | 3 | 5 | 2 | 5 |
| | 5 | 9 | 4 | 5 |
| 7 | 1 | 13 | 0 | 13 |
| | 3 | 9 | 2 | 9 |
| | 5 | 5 | 4 | 9 |
| 8 | 0 | 10 | 1 | 14 |
| | 2 | 14 | 3 | 14 |
| | 4 | 2 | 5 | 2 |
| 9 | 0 | 6 | 1 | 2 |
| | 2 | 2 | 3 | 2 |
| | 4 | 13 | 5 | 14 |
| 10 | 0 | 14 | 1 | 2 |
| | 2 | 10 | 3 | 2 |
| | 4 | 6 | 5 | 14 |
| 11 | 0 | 2 | 1 | 6 |
| | 2 | 6 | 3 | 6 |
| | 4 | 10 | 5 | 10 |
| 12 | 1 | 11 | 0 | 11 |
| | 3 | 7 | 2 | 11 |
| | 5 | 7 | 4 | 7 |
| 13 | 1 | 7 | 0 | 7 |
| | 3 | 11 | 2 | 7 |
| | 5 | 11 | 4 | 11 |
| 14 | 1 | 15 | 0 | 15 |
| | 3 | 3 | 2 | 15 |
| | 5 | 3 | 4 | 3 |
| 15 | 1 | 3 | 0 | 3 |
| | 3 | 15 | 2 | 3 |
| | 5 | 15 | 4 | 15 |

## (2) Linear equations

For each sum in **Table 1**, we will get one linear equation according to the parity sum equation

$x_t^1 \oplus x_t^2 \oplus x_t^3 \oplus x_t^4 \oplus x_t^5 = c_t^0 \oplus z_t$ where $z_t$ is a known bit and $c_t^0$ is a guessed bit. According to the guessed sum value we can write the following equations. For a sum value 0 or 5 we get 5 linear equations. For a zero sum value, $x_t^1 = 0, x_t^2 = 0, x_t^3 = 0, x_t^4 = 0, x_t^5 = 0$ and a 5 sum value, $x_t^1 = 1, x_t^2 = 1, x_t^3 = 1, x_t^4 = 1, x_t^5 = 1$. For a guessed sum of 1 or 4 we get 5 different options of 5 linear equations each. And for a sum of 2 or 3 we will get 10 different options of 5 linear equations. As dedicated in [8] the backtracking cost of sum of 2 or 3 is more expensive than that of 0 or 5 and 1 or 4. The main goal is to go over all the possible chains which end with 168 equations.

## (3) The estimated number of chains

**Table 1** shows that we always have three next states for either value of output bit $Z_t$ so W (t) =3, and for depth of T backtracking W (t=T) = $(3)^T$, where W(T) is the number of the chains in the backtracking process.

## (4) The estimated backtracking chain lengths

The chain length is function in the number of the equations obtained during the backtracking process. Let N (t) denote the number of equations that was obtained using t output bits.

### a-Using sums 0/1/4/5:

From the previous discussion, there is one equation for every output and four additional equations for sums of 0/1/4/5. The six values of the sum are uniformly distributed so the probability for each sum is 1/6, then the probability of sums 0/1/4/5 is 4/6.

Referring to [8] the expectation of the number of equation can be defined by the following equation.

$N(t) = N(t-1) + 1 + 4/6 * 4 = N(t-1) + 22/6$

For backtracking of depth T

$N(T) = 22/6 * T$ equations

The number of equations that we get should not exceed 168.

$N(T) = 22/6 * T \geq 168$
$T \geq 45.82$ while it was in [1] 37.64

The minimum number of equations is 40 which is the largest LFSR length.

### b-Using only sums 0/5:

As explained in the previous section the probability of having sums of 0/5 is 2/6 so

$N(t-1) = N(t-1)+1+2/6 * 4= N(t-1)+14/6$

The recursive function can be translated into a linear one.

$$N(T) = 14/6 * T \geq 168$$
$$\text{So } T \geq 72.$$

The estimated depth of the backtracking chain is 72, while it was in [8] 59.

## (5) The computational complexity of the backtracking process

Ophir and Avishai follow in their estimation of the total number of linear system that can be solved during the total process a certain procedure, we will follow this procedure for our modified cipher. Let T the length of the chains. Assume R positions out of the total T position in the chain hold the sum 0/1/4/5 and the other T-R holds for 2/3 sums.

The probability of obtaining sums of 0 or 5 is ½ among R and the probability of sums 1 or 4 is ½. For sums of 0/5 there is only one option to backtrack through, but for sums 1 / 4 there are 5 options. So the expected number of linear equations will be

$$E=\sum_{i=0}^{R}\binom{R}{i}(1/2)^{i}(1/2)^{R-i}.1^{i}.5^{R-i}=$$

$$1/2R\sum_{i=0}^{R}\binom{R}{i}.5^{R-i}.1^{i}=1/2R(5+1)^{R}=(3)^{R}$$

As obtained in the previous section the total number of chains is $3^T$ for each output bit. We have one default and four additional equations, each chain has $(3)^R$ equations to be solved. We require 168 equations in order to solve the LFSRs initial states. So our constraints is

$$T+4R\geq 168$$
$$R=(168-T)/4$$

The number of equations is

$$(3)^{T}(3)^{R}=(3)^{T+R}=(3)^{|T+(168-T)/4}$$

For T = 40 (the length of the largest LFSR)
The total number of equations is

$$(3)^{40+(168-40)/4}=2^{121.25}$$

The total computational complexity reaches $2^{125.25}$ , this by adding the 4 bits initial states. We observe that adding the fifth LFSR to the E0 cipher enhance the cipher against Ophir Levy and Avishai Wool attack, where their computational complexity is $2^{90.63}$.

In the following section, the quality of encryption of the original and the modified cipher will be compared through measuring three factors on encrypted images.

## 5- QUALITY OF ENCRYPTION MEASURING FACTORS OF BITMAP IMAGES

Bitmap image is a type of uncompressed image format which preserve all information about the image data [11]. To encrypt an image by the original cipher or the modified cipher, the image will be read by Matlap program and will be set in one row one bit after each other and each bit will be XORed with the corresponding key bit. The input bits will be in packets with maximum length of 2745 bits . when the transmission of this packet is completed, the cipher is reinitialized. In our simulation program we have the option to change the packet length(encryption key) to be 125 bits or 2745 bits. Shorter key length provides more better result than longer one because of the reinitialization of the cipher after ending each packet length. Images quality of encryption can be measured by different methods. Visual inspection is one of these methods, where the highly disappeared featured of the image the better the encryption algorithm. As will be shown both the original and the modified ciphers has good visual inspection. So three other factors will be applied to measure the quality of encryption [11-13]. The first factor is the Maximum Deviation (MD) factor which measures the maximum deviation between the plaintext and the encrypted images. The second factor is the Correlation Coefficient (CC) between the plaintext image and its encryption. The third factor is the Irregular Deviation (ID) which measures the difference between the pixel value of the plaintext image and its corresponding pixel value of the encrypted one, for detail description refer to [11], [12].

The higher the maximum deviation the better the encrypted image is deviated from the original one. The lower value of the correlation coefficient indicates no relation between the plaintext image and its encryption. The lower the irregular deviation value the better encryption cipher.

## 6. RESULTS AND DISCUSSION

In our simulation programs three different BMP images will be considered. These images are Lena.bmp (**Fig. 7**) as it is the reference image used in image processing research, Nike.bmp (**Fig. 8**) as an example of an image containing very large areas of a single color and it is an example of a binary image and Barbara.bmp ( **Fig. 9**) an example of an image containing many high frequency components.

The three images will be encrypted with E0, E0_1, and E0_2 which are the original algorithm, the first modification, and the second modification respectively. Two key lengths will be implemented on each image, which are 125 and 2745 bits. The process of encryption will be done by XORing the bit value of the image with key bit. After the key length is ended an initialization process will start again. So the image encrypted with a 125 bit key will have more quality than that encrypted with 2745 bit key. Measured values of the three algorithms with a key length of 2745 bits and 125 bits are given in **Tables 2** and **3**.

**Table 2.** The measured values of the three algorithms with a key length of 2745 bits.

| Cipher | E0 | | | E0_1 | | | E0_2 | | |
|---|---|---|---|---|---|---|---|---|---|
| | MD | CC | ID | MD | CC | ID | MD | CC | ID |
| Lena | 134987943.5 | -0.0036889 | 16142 | 241878471.5 | 0.01268 | 16404 | 299810120 | 0.010585 | 16292 |
| Nike | 276247618.5 | 0.0026732 | 1936 | 441636858 | -0.0021906 | 2062 | 184859334 | -0.0022608 | 2954 |
| Barbara | 67890658.5 | -0.0070137 | 20434 | 94043955 | -0.0049528 | 20270 | 124703712 | -0.0020876 | 20594 |

**Table 3.** The measured values of the three algorithms with a key length of 125 bits.

| Cipher | E0 | | | E0_1 | | | E0_2 | | |
|---|---|---|---|---|---|---|---|---|---|
| | MD | CC | ID | MD | CC | ID | MD | CC | ID |
| Lena | 182935346 | 0.0058579 | 16458 | 380155660 | -0.003198 | 16364 | 299520903 | 0.0036831 | 16816 |
| Nike | 259002660 | 0.0078693 | 5686 | 602305461 | 0.0004527 | 6006 | 499156932 | 0.008994 | 14026 |
| Barbara | 118395597 | -0.00015569 | 20734 | 297073140 | -0.002324 | 20634 | 35899259.5 | 0.008894 | 16098 |

### Testing the results of the three images with the E0_1 compared with E0:

- **Lena image:** For a key length of 2745 bit, the results indicate that the value of MD is better for E0_1 than E0, but CC and ID values are better for E0 than E0_1. For a key length of 125 bit, the results indicate that the values of the three measured factors are better for the E0_1 than E0.

- **Nike image:** For a key length of 2745 bit, the results indicate that the values of MD and CC are better for E0_1 than E0, but ID value is better for E0 than E0_1. For a key length of 125 bit, the results indicate that MD and CC values are better forE0_1 than E0.
- **Barbara image:** For a key length of 2745 bit, the results indicate that the values of the three measured factors are better for E0_1 than E0. For a key length of 125 bit, the results indicate that the values of MD and ID are better for E0_1 than E0.

We have three measuring factors if two of them give better results for a certain cipher then we can say that it is the strongest one.

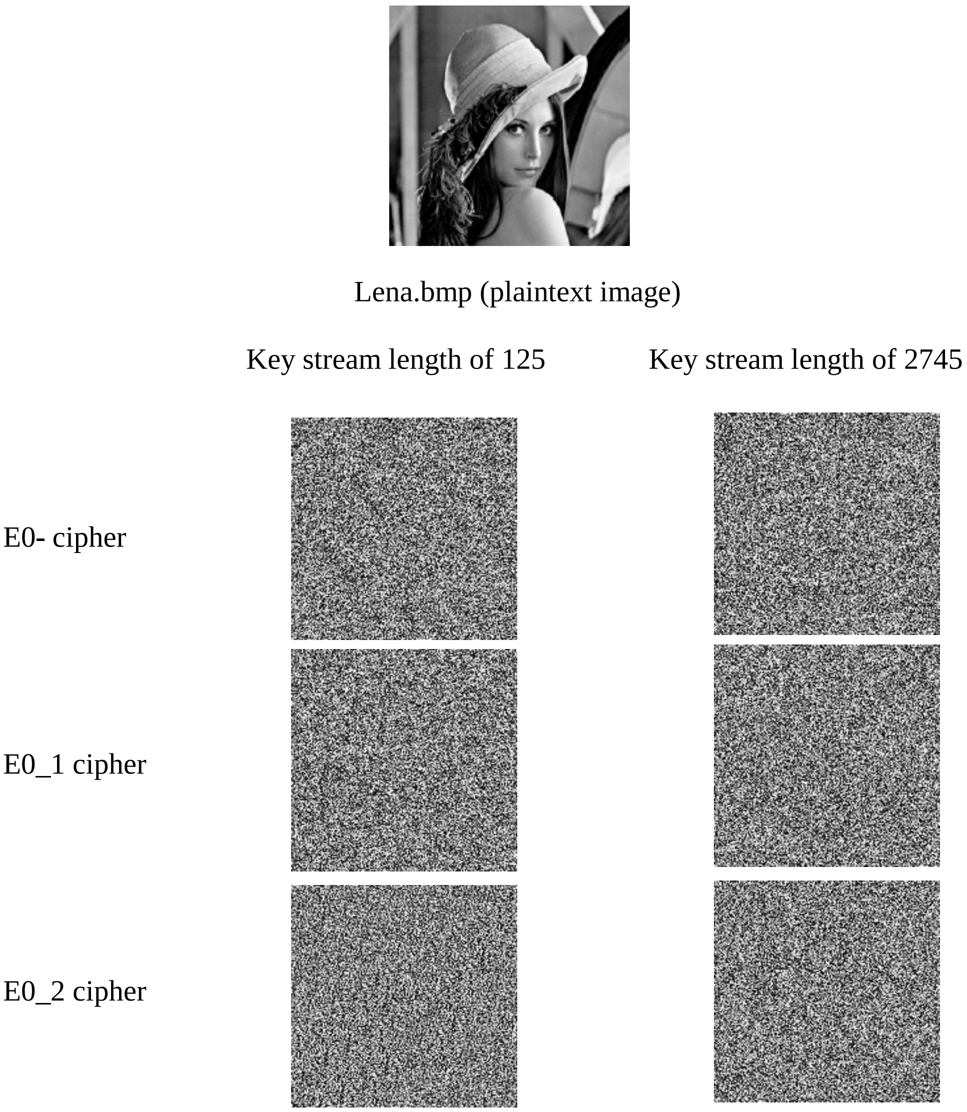


**Fig. 7** Encryption of Lena.bmp image by the original and the two modified E0 ciphers by using two different key stream lengths

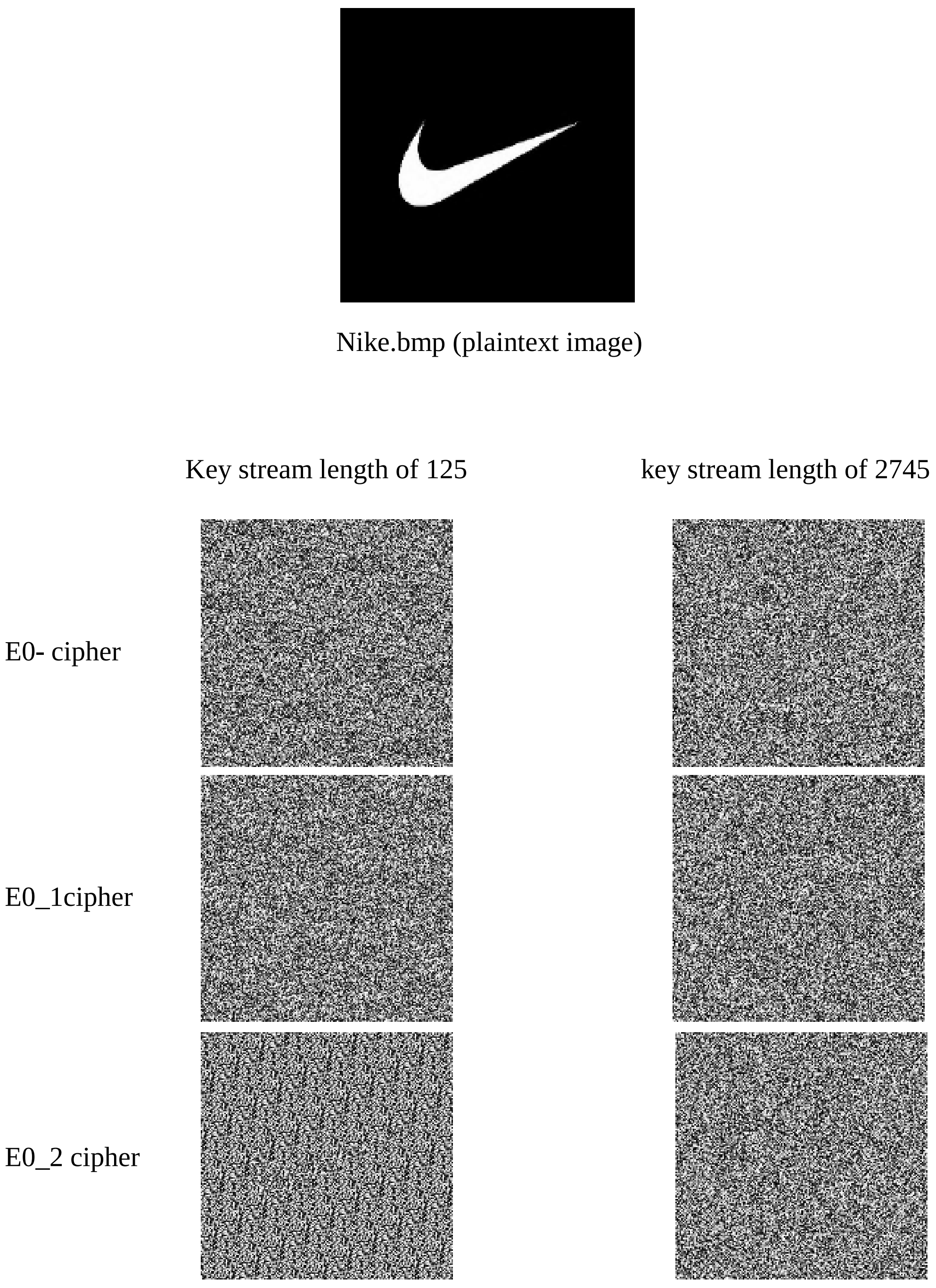
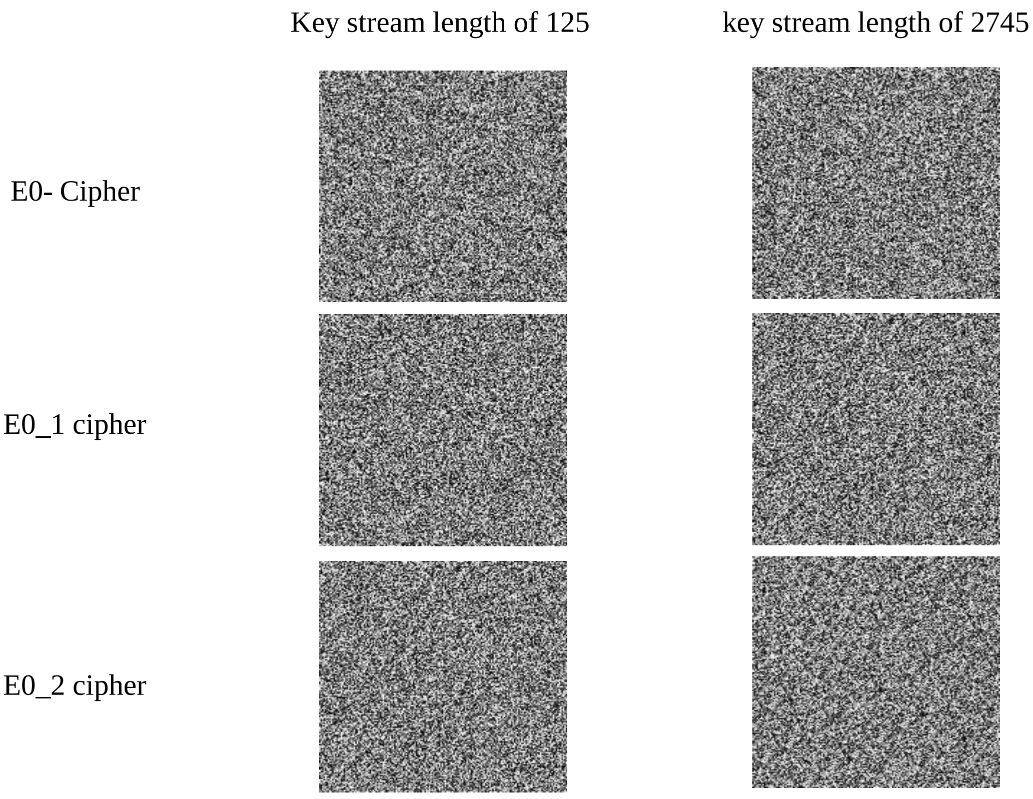


**Fig. 8** Encryption of Nike.bmp image by the original and the two modified E0 ciphers by using two different key stream lengths.

Barbara.bmp ( plaintext image)



**[Fig. 9** Encryption of Barara.bmp image by the original and the two new modified E0 ciphers by using two different key stream lengths.

## Testing the results of the three images with the E0_2 cipher compared with the E0

- **Lena image:** For a key of length of 125 bit, the results indicate that the values of the three measured values are better for the E0_2 than E0, but with a key length of 2745 bit E0 is better for values of CC and ID.
- **Nike image:** For a key of length 125 bit, the results indicate that the values of the three measured factors are better for E0_2 than that of E0, but with a key length of 2745 bit E0 is better for the values of MD and ID.

- **Barbara image:** For a key of 125 bit ,the results indicates the value of CC is better for E0_2 than E0, but the values of MD and ID are better for E0 than E0_2. For a key of 2745 bit, the values of MD and CC are better for E0_2 than E0, but ID value is better for E0 than E0_2.

## 7. CONCLUSION

The paper presented a modification to the Bluetooth E0 cipher which implemented through the addition of the fifth LFSR to the KSG. The modified cipher increased the computational complexity of a uniform frame work cryptanalysis from O ( $2^{94.63}$ ) to O ( $2^{125.25}$ ), image encryption inspected by the original and the modified cipher. The parameter measured on image encryption highlighted the higher degree of encryption achieved by the modified cipher.

## REFRENCES

[1] Specification of the Bluetooth System V.1.2. Core Specification, available from http://www.bluetooh.org/spec, 2003.

[2] Scott R. Fluhrer and Stefan Lucks, " Analysis of the E0 Encryption System" , Proc. of the 8th workshop on Selected Areas in Cryptography, LNCS 2259. Springer Verlag, 2001.

[3] P. EkdahI and T. Johansson, " Some Results on Correlation in Bluetooth Stream Cipher" Proc. Of the 10th joint conference on Communication and Coding. Obertauern, Austria, March 11-18, 2000.

[4] 4-Y. Lu and S. Vaudenay," Faster Correlation Attack on Bluetooth Key stream Generator E0", Advances in Cryptology – CRYPTO'04, LNCS 3152, pages. 406-425, Springer – Verlag, 2004.

[5] M. Hermeline and K. Nyberg, "Correlation Properties of the Bluetooth Combiner Generator", Information Security and Cryptology, LNCS 1687, pages. 17-29, Springer – Verlag, 1999.

[6] 6-F. Armknecht, " A Linearization Attack on the Bluetooth Key Stream Generator ", Cryptology eprint Archive, report 2001/191, available from http://www.eprint.iacr.org/2002/191/2002 .

[7] 7-Markus Jakobsson and Susanne Wetzel, " Security Weakness in Bluetooth ," In Proc. RSA Security Conference- Cryptographer's Track, LNCS 2020, pages 176-191. Springer – Verlag, 2001.

[8] O.Levy and A. Wool, "A Uniform Frame work For the Cryptanalysis of the Bluetooth E0 cipher", Proc. of the 1st International Conference on Security and Privacy for Emerging Areas in Communication Networks(Securecomm), pages 365-373, Athens, Greece, September 2005.

[9] C. Rechberger, Side Channel Analysis of Stream Ciphers, Master's Thesis, Institute for Applied Information Processing and Communications(IAIK), Graze University of Technology, 2004.

[10] C. Gehrmann, J. Persson, and B. Smeet, Bluetooth security, Artech House computer security series, 2004.

[11] N. El-Fishawy and Osama M. Abu Zaid, " Quality of Encryption Measurement of Bitmap Images with RC6, MRC6, and Rijndal Block Cipher Algorithms", In Minufia Journal of Electronic Research(MJEER), Vol.15, No. 2, July 2005.
[12] H. Elkamchouchi and Mina A. Makar," Measuring Encryption Quality of Bitmap Images Encrypted with Rijndal and KAMKAR Block Ciphers", In Proc. Twenty second National Radio Science Conference (NRSC,2005), pages C11, Cairo, Egypt, March 15-17, 2005
[13] Ibrahim Ziedan, Mohammed Fouad, Doaa H. Salem, "Application of Data Encryption Standard to Bitmap and JPEG images ", In Proc. Twentieth National Radio Science Conference ( NRSC, 2003), pages,C16, Egypt, March, 2003.

## تعديل في خوارزم التشفير المستخدم في شبكات السّنة الزرقاء

فـي هـذا البحـث : تـم التركيـز والشـرح الـوافي لخـوارزم التشـفير**(E0 stream cipher)** المستخدم في شبكات السّنة الزرقاء. وتم اقتراح نموذج معدل لهذا الخوارزم وذلك بإضـافة **linear feed back shift register** خامس إلى الأربعة المستخدمين بالفعل فـي الخـوارزم الأصلي داخل مولد مفتاح التشفير**(key stream generator)**

وتم دراسة نتائج هذا التعديل المقترح خلال اتجاهين مختلفين:

الاتجـاه الأول: ذلـك مـن خـلال تطبيـق الهجـوم الخـاص بالعـالمين **Ophir Levy and Avishai Wool** على الخوارزم المقترح المعدل فأدلـت نتـائج التطبيـق بزيـادة الجهـد المبـذول **(computational complexity)** فـي كسـر الخـوارزم المعـدل مـن $2^{90.63}$ إلى $2^{125.25}$.

الاتجاه الثانـي : تـم قيـاس جـودة تشفير الخـوارزم المعدل المقترح عن طريق تطبيق هذا الخوارزم المعدل على تشفير مجموعة من الصور ومقارنـة النتائج مع الخـوارزم الأصـلي وكانت النقاط التي تم القياس عليها وهى :

1- معامل الربط التقاربى **(correlation coefficient)**.
2- الحيود الأعظم **(maximum deviation)**.
3- الحيود الغير منتظم **(irregular deviation)**.

وقـد أدلـت نتـائج الثلاثـة عوامـل بـان الخـوارزم المعـدل المقتـرح لـه أفضـل نتـائج عـن الخوارزم الأصـلي.