A Proposed Non-**Linear** Shifted Box

Prof. DR. Imad Hussein Al-Husaini - Iraqi commission for Computer and Informatics

Dr. Jane Jaleel Stephan - Iraqi commission for Computer and Informatics

Ahmed Hussein Ali - Baghdad College for Economic Sciences

Abstract

The Objective of this research is to make the **Shifted Box** (***SB***) which can be defined as two-dimensional matrix of (N*M), unlimited and random content by making the values of rows and columns, changing at a time and before the shifting.

There are two methods to process the Shifted Box namely: -

1. **(Fixed positions movement)**
2. **(Variable positions movement).**

1. Introduction

Cryptography (or cryptology) is the practice and study of hiding information. In modern times, cryptography is considered a branch of both mathematics and computer science, and is affiliated closely with information theory, computer security, and engineering. Cryptography is used in applications present in technologically advanced societies [1].

Information security uses cryptography to transform usable information into a form that renders it unusable by anyone other than an authorized user.

Cryptography is used in information security to protect information from unauthorized or accidental disclosure while the information is in transit (either electronically or physically) and while information is in storage Until modern times, cryptography referred almost exclusively to **encryption**, the process of converting ordinary information (**plain text**) into unintelligible gibberish (**cipher text**). **Decryption** is the reverse, moving form unintelligible cipher text to plain text.

A cipher is a pair of algorithms which creates the encryption and the reversing decryption, the detailed operation of a cipher is controlled both by the algorithm and, in each instance, by a **key**. Keys are important, as ciphers without variable keys are trivially breakable and therefore less than useful for most purposes. This is all shown in Figure 1 [2].

2. Stream Cipher Structure

A stream cipher is a symmetric key cipher where plain text bits are combined with a pseudorandom cipher bit stream (key stream), typically by an exclusive-or (XOR) operation. In a stream cipher the plain text digits are encrypted one at a time, and the transformation of successive digits varies during the encryption. An alternative name is a state cipher, as the encryption of each digit is dependent on the current state. In practice, the digits are typically single bits or bytes.

Stream ciphers represent a different approach to symmetric encryption from block ciphers. Block ciphers operate on large blocks of digits with a fixed, unvarying transformation. This distinction is not always clear-cut: in some modes of operation, a block cipher primitive is used in such a way that it acts effectively as a stream cipher.

Stream ciphers typically are executed at a higher speed than block ciphers and have lower hardware complexity.

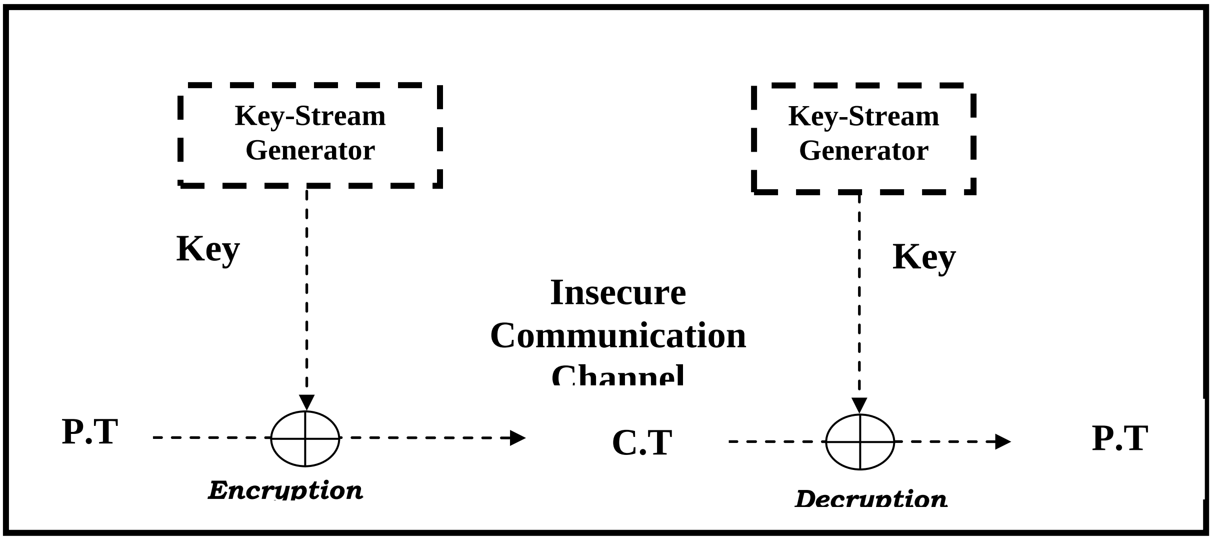


Figure 1 Stream Cipher Structure

There is a vast body of theoretical knowledge on stream ciphers, and various design principles for stream ciphers have been proposed and extensively analyzed. However, there are relatively few fully-specified stream cipher algorithms in the open literature. This unfortunate state of affairs can partially be explained by the fact that most stream ciphers used in practice tend to be proprietary and confidential. By contrast, numerous concrete block cipher proposals have been published, some of which have been standardized or placed in the public domain. Nevertheless, because of their significant advantages, stream ciphers are widely used today, and one can expect increasingly more concrete proposals in the coming years Block ciphers process plaintext in relatively large blocks (e.g., n _ 64 bits). The same function is used to encrypt successive blocks; thus (pure) block ciphers are memory less [3].

In contrast, stream ciphers process plaintext in blocks as small as a single bit, and the encryption function may vary as plaintext is processed; thus stream ciphers are said to have memory. They are sometimes called state ciphers since encryption depends on not only the key and plaintext, but also on the current state. This distinction between block and stream ciphers is not definitive; adding a small amount of memory to a block cipher (as in the CBC mode) results in a stream cipher with large blocks [4].

3. The Proposed Non-Linear Shifted Box Cipher

Non-**L**inear **S**hifted **B**ox (**NLSB**) can be defined as two dimensional matrix of (N*M), where **N** is a number of rows and **M** is the number of columns, each row can be considered as a **L**inear **F**eedback **S**hift **R**egister (**LFSR**) and each column can be considered as an **LFSR** in the same time. All rows have the same predefined feedback function defined for a predetermined two tapping stages (see figure 2)

LFSR-1
LFSR-2
LFSR-3
.
.
.
.
LFSR-M

Figure 2 Non-Linear Shifted Box

In the second hand, all columns have the same predefined feedback function defined for a predetermined two tapping stages (see figure 3).

LFSR-1	**LFSR-2**	**LFSR-3**	.	.	.	.	.	.	.	.	**LFSR-N**

Figure 3 Non-Linear Shifted Box

The movement of these registers must be performed for the rows before columns or columns before rows and not for both at the same time. The initialization of the shifted box must be done by a predefined procedure with a sequence of bits of length N*M depending on the application that uses it.

The nonlinearity of the shifted box can be obtained from making the feedback function of these registers depend on a variable tapping stages.

One of the two tapping stages must be the last stage (stage number N for rows and stage number M for columns), so, the variability is by selecting the second stage for the tapping of the feedback function.

In this paper, two procedures for selecting the variable stage will be produced as discussed in the following sections.

The operation of shifting for any row of (N) is the same of operation of all rows of the N in the box because its have the same tapping and the operation of shifting for any column of (M) is the same of operation of all columns of the M in the box because its have the same tapping.

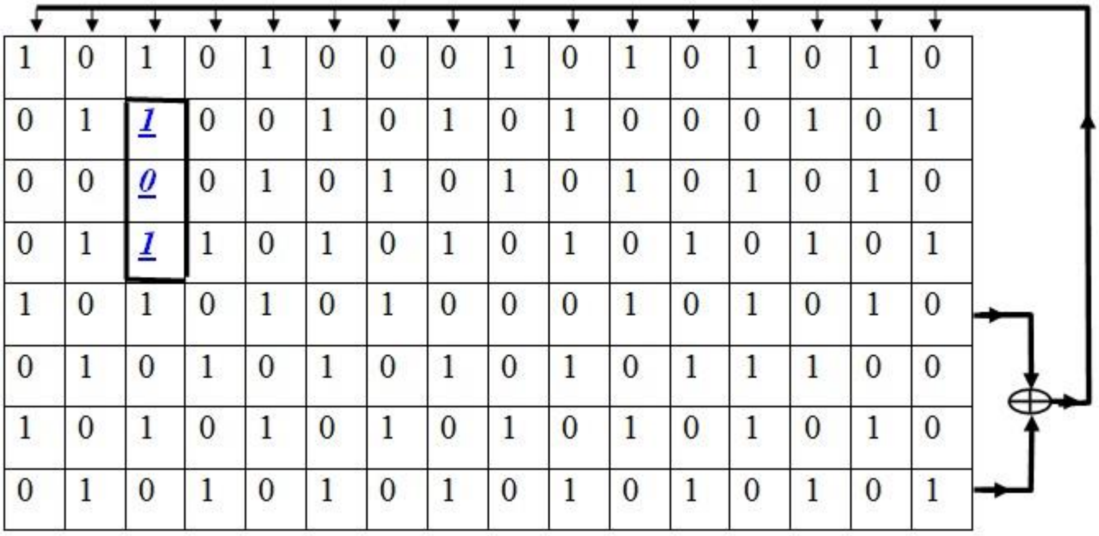
We can say that this method is a linear because the operation of shifting is the same of all operation, but if we want to be change the function from linear to nonlinear we must change the operation and became depend on input of the function or from out of the function as using specified locations and fixed and called it **(Fixed Positions Movement)** or using large LFSR Relatively and call it **(Variable Positions Movement).**

4. Fixed positions movement

In this class the shifted box stages that selected to determine the second stage of the feedback function selected from a fixed positions from the entire stages of the shifted box. For example, assuming that we have a shifted box of 8x16 dimensions, in this case the stage required to determine the number of the stage of the vertical LFSR's is one from 0-7 and the stage of the horizontal LFSR's is one from 0-15, so we need three bits to represent the vertical stage and four bits to horizontal stage.

4.1. Vertical Registers

The movement of the rows registers must be performed for a number of shifts determined by fixed selected positions. For example, if the fixed positions are selected randomly as described in figure 4, the number 101 will select the row number 5 to be tapped in the feedback function, so all content of row number 8 and the content of row number 5 will be Xored correspondingly resulted a new row will be stored at row number 1 after shifting the rows down by 1, the last row will be ignored.

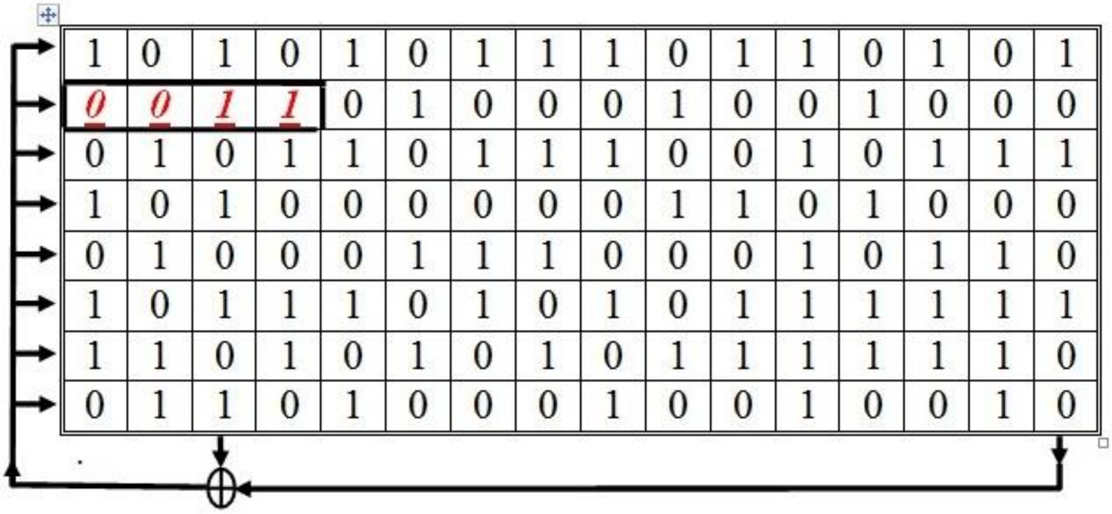


1	0	1	0	1	0	0	0	1	0	1	0	1	0	1	0
0	1	*1*	0	0	1	0	1	0	1	0	0	0	1	0	1
0	0	*0*	0	1	0	1	0	1	0	1	0	1	0	1	0
0	1	*1*	1	0	1	0	1	0	1	0	1	0	1	0	1
1	0	1	0	1	0	1	0	0	0	1	0	1	0	1	0
0	1	0	1	0	1	0	1	0	1	0	1	1	1	0	0
1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Figure 4 Vertical shift steps

4.2. Horizontal Shifting

The movement of the columns registers must be performed for a number of shifts determined by fixed selected positions. For example, if the fixed positions are selected randomly as described in figure 5, the number 0011 will select the column number 3 to be tapped in the feedback function, so all content of column (**N**) number 16 and the content of column number 3 will be Xored correspondingly resulted a new column will be stored at column number 1 after shifting the columns to the right by 1, the last column will be ignored.



1	0	1	0	1	0	1	1	1	0	1	1	0	1	0	1
0	*0*	*1*	*1*	0	1	0	0	0	1	0	0	1	0	0	0
0	1	0	1	1	0	1	1	1	0	0	1	0	1	1	1
1	0	1	0	0	0	0	0	0	1	1	0	1	0	0	0
0	1	0	0	0	1	1	1	0	0	0	1	0	1	1	0
1	0	1	1	1	0	1	0	1	0	1	1	1	1	1	1
1	1	0	1	0	1	0	1	0	1	1	1	1	1	1	0
0	1	1	0	1	0	0	0	1	0	0	1	0	0	1	0

Figure 5 Horizontal shift steps

5. Variable positions movement

In this class, we use an LFSR as a generator (see figure 6) to generate a group of bits to determine the second tapping stage of the feedback function 0f the shifted box stages. For example, assuming that we have a shifted box of 8x16 dimensions, in this case the stage required to determine the number of the stage of the vertical LFSR's is one from 0-7 and the stage of the horizontal LFSR's is one from 0-15, so we need three bits to represent the vertical stage and four bits to horizontal stage.

For example we select an LFSR of length 42, the tapping stages can be defined as 42, 7, 4, and 3 which produce a maximum period [3].

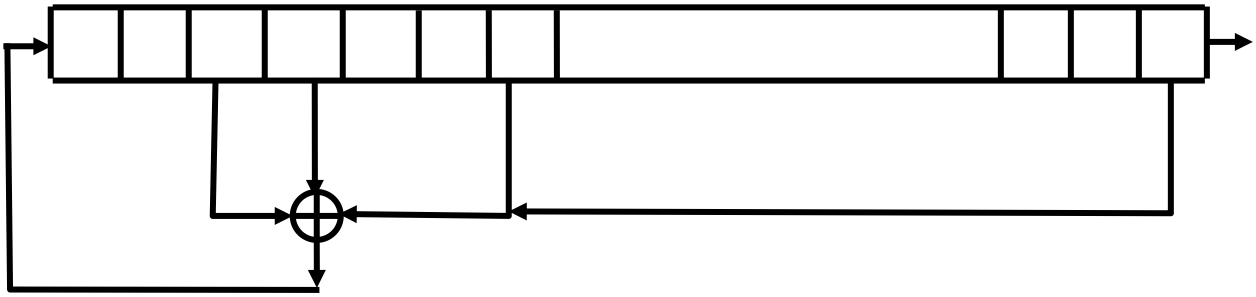
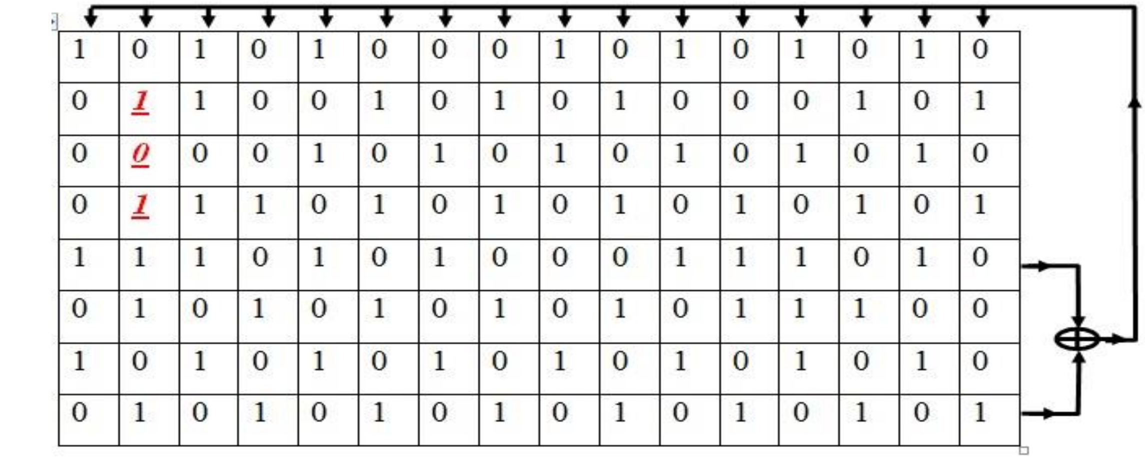


Figure 6 LFSR Generator

The initialization process is depending on the application which will produce a random sequence of length 42. To obtain the position of the second stage of the shifted box we must generate $\log_2 M$ for rows LFSR's and $\log_2 N$ for columns LFSR's. Assuming that we use a shifted box of 8x16 so, we need to produce three bits to obtain the position of the second stage of the shifted box, and 4 bits to obtain the second stage of the shifted box.

5.1 Vertical Shifting

The movement of the columns registers must be performed for a number of shifts determined by output of the LFSR. For example, if the output bits that selected randomly as 101 and converted to a whole number will determine the row number 5 to be tapped in the feedback function see figure 5, so all content of row (**M**) number 8 and the content of row number 5 will be Xored correspondingly resulted a new row will be stored at row number 1 after shifting the rows down by 1, the last row will be ignored.

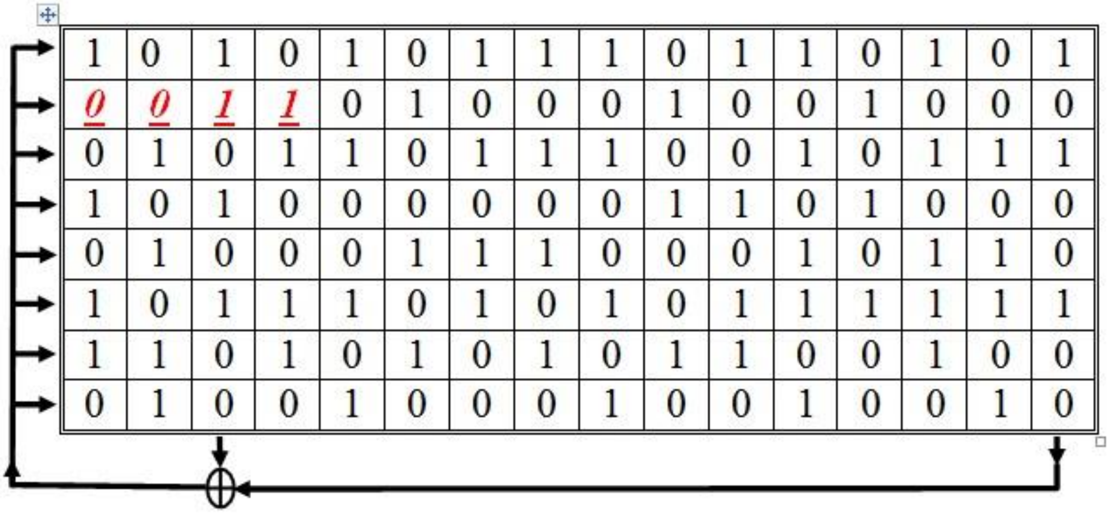


1	0	1	0	1	0	0	0	1	0	1	0	1	0	1	0
0	*1*	1	0	0	1	0	1	0	1	0	0	0	1	0	1
0	*0*	0	0	1	0	1	0	1	0	1	0	1	0	1	0
0	*1*	1	1	0	1	0	1	0	1	0	1	0	1	0	1
1	1	1	0	1	0	1	0	0	0	1	1	1	0	1	0
0	1	0	1	0	1	0	1	0	1	0	1	1	1	0	0
1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Figure 7 Vertical shift steps

5.2. Horizontal Shifting

The movement of the columns registers must be performed for a number of shifts determined by output of the LFSR. For example, if the output 0011 that selected randomly as described in figure 8, the number 0011 will determine the column number 3 to be tapped in the feedback function, so all content of column (**N**) number 16 and the content of column number 3 will be Xored correspondingly resulted a new column will be stored at column number 1 after shifting the columns to the right by 1, the last column will be ignored



1	0	1	0	1	0	1	1	1	0	1	1	0	1	0	1
0	*0*	*1*	*1*	0	1	0	0	0	1	0	0	1	0	0	0
0	1	0	1	1	0	1	1	1	0	0	1	0	1	1	1
1	0	1	0	0	0	0	0	0	1	1	0	1	0	0	0
0	1	0	0	0	1	1	1	0	0	0	1	0	1	1	0
1	0	1	1	1	0	1	0	1	0	1	1	1	1	1	1
1	1	0	1	0	1	0	1	0	1	1	0	0	1	0	0
0	1	0	0	1	0	0	0	1	0	0	1	0	0	1	0

Figure 8 Horizontal shift steps

6. Complexity Measurement

To calculate the complexity of the algorithm we need to specify the period of the key generator algorithm which will be equivalent to the complexity of the system [6]. The period of the **Fixed positions movement** algorithm can be computed by $\mathbf{2^{N*M}}$ (all possible length) $\mathbf{*\ 2^{\log_2 N} * 2^{\log_2 M}}$ (the fixed positions of the SB) and the period of the **Variable positions movement** algorithm can be computed by $\mathbf{2^{N*M}}$ (all possible length) $\mathbf{*\ 2^{\log_2 N} * 2^{\log_2 M}}$ (the Variab**le** positions of the SB) $\mathbf{*\ 2^{42}}$ (the period of the LFSR).

The complexity of the two classes of the movement algorithm approximately exceeds any input sequence in an applicable algorithm and might not be attacked in a brute-force attack. The algorithm cannot be attacked by any other known attacking method because of the high computational complexity of the permutation and matrix shifting.

7. Conclusions and Suggestion for Future Work

We can conclude that the proposed algorithm has a high complexity, long periodicity, high nonlinearity:-

1. Is a high complexity, long periodicity, high nonlinearity.
2. The period of the **Fixed positions movement** algorithm can be computed by $\mathbf{2^{N*M}}$ (all possible length) ***** $\mathbf{2^{\log_2 N}}$ ***** $\mathbf{2^{\log_2 M}}$ (the fixed positions of the SB) and the period of the **Variable positions movement** algorithm can be computed by $\mathbf{2^{N*M}}$ (all possible length) ***** $\mathbf{2^{\log_2 N}}$ ***** $\mathbf{2^{\log_2 M}}$ (the Variabl**e** positions of the SB) ***** $\mathbf{2^{42}}$ (the period of the LFSR).The periodicity of the algorithm obtained will exceed the length of any input media might be encrypted by any other algorithm.
3. The recommendation is that if the movement of SB implements using LFSR with another SB that contain a random number that used to determine the position of the second tap as permutation table and this class of movement will increase the complicity.

8. References

[1] Wikipedia the free encyclopedia: retrieved on 7/10/2008 from Http://en.wikipedia.org/wiki/Cryptoghrapy.

[2] O. Goldreich, "Foundations of Cryptography", Department of Computer Science and Applied Mathematics, Weizmann Institute, 1995.

[3] Rolf Oppliger (**Contemporary Cryptography**) Artech House Boston London **2007**

[4] Beker and Piper, “**Cipher Systems**”, Northwood Publication, U.K., 1982.

[5]: Niels Ferguson , Doug Whiting , Bruce Schneier , John Kelsey , Stefan Lucks , and Tadayoshi Kohno, " **Fast Encryption and Authentication in a Single Cryptographic Primitive**", University at Mannheim.

[6] Paris Kitsos and Ulrich Kaiser, " **A High-Speed Hardware Implementation of the Hermes 8-128 Stream Cipher**", School of Science and Technology in Hellenic Open University, Patras, Greece.