Security Analysis of Exclusive 128 Bit NLFSR Based Stream Cipher

Biju N[#1], Jegadish Kumar K. J [#2]

[#1#2]*SSN College of engineering*
Anna University,
Kalavakkam, 603110,India
[1]bijun46@gmail.com
[2]jegadishkj@ssn.edu.in

***Abstract*-This paper states the analysis of security for Exclusive 128-bit Non-Linear Feedback Shift Register (NLFSR) based Stream Cipher. Stream Cipher is a common method to protect confidential information from an unauthorized intrusion. NLFSRs are a generalization of Linear Feedback Shift Register (LFSR) in which a current state is a non-linear function of the previous state. NLFSR provide better trade off between security and hardware capability. NLFSR output sequences are normally very hard to predict and existing attacks such as algebraic attacks are not applicable. NLFSR based Stream Ciphers are mostly employed in RFID and smartcard applications. In this paper, Exclusive 128-bit stream cipher NLFSR is implemented by two configurations: Fibonacci Configurations and Galois Configurations. The security analysis is done based on *National Institute of Standards and Technology (NIST)*.**

***Keywords*-Cryptography, Stream cipher, NLFSRs, Fibonacci configuration, Galois configuration, NIST.**

I. INTRODUCTION

Cryptographic methods are applied in order to protect confidential information from an unauthorized or accidental disclosure. Today our society largely depends on security of electronic communications. Whether the communication path could be as short as a wire between two chips or as long as the World Wide Web, communications securities are important for integrity of our business.

Stream cipher is one of a common method that is employed for protecting the secret or private information from an unauthorized intrusion. Security solutions in mobile communications generally rely on the use of stream ciphering techniques [1]. Stream ciphers are symmetric-key ciphers that generate pseudorandom bit sequences that are used to encrypt the message signals on bit-by-bit basis. In a stream cipher, each plaintext digit is encrypted one at a time with the corresponding digit of the key stream, to create a digit of the ciphertext stream. The resulting encrypted information can be transformed back into its original form only by an authorized user possessing the cryptographic key [2]. The security of these ciphers depends upon the randomness of the bit sequences produced by them. The pseudo random keystream is typically generated serially from a key and an Initialization Vector (IV) or seed value using digital shift registers. The seed value serves as the cryptographic key for decrypting the ciphertext stream.

A stream cipher generates successive elements of the keystream based on an internal state. This state is updated in essentially two ways: if the state changes independently of the plaintext or ciphertext messages, the cipher is classified as a synchronous stream cipher. By contrast, self-synchronizing stream ciphers update their state based on previous ciphertext digits. Some of the known stream Ciphers are E0 used in Bluetooth, A5/1 used in GSM communications, RC4 and Grain in RFID applications. A5/1 is a stream cipher used in GSM standard to encrypt the information over the air transmission[3][4]. Encryption in mobile communication is very crucial to protect information of the subscribers and avoid fraud.

II. RELATED BACKGROUND

A pseudo-random sequence can be generated using a *Linear Feedback Shift Register (LFSR)*. LFSR is a shift register whose input bit is a linear function of its previous state. LFSRs are simple, fast, and easy to implement in both, software and hardware. They are capable of generating pseudo-random sequences with the same uniform statistical distribution of 0's and 1's as in a truly random sequence[5]. However, they are not cryptographically secure because the structure of an *n*-bit LFSR can be easily deduced by observing 2^n consecutive bit of its sequence [6].

One solution to this problem is to feed the outputs of several parallel LFSRs into a non-linear Boolean function to form a combination generator[7]. The combining function has to be carefully selected to ensure the security of the resulting scheme, for example, in order to prevent correlation attacks[8]. Other solutions are to combine several bits from the LFSR state using a

non-linear function, or to use the irregular clocking of the LFSR [9][10]. Examples of LFSR-based stream ciphers include A5/1 stream cipher that used to provide over-the-air communication privacy in the GSM cellular telephone standard, and E0 stream cipher that is used in the Bluetooth protocol. As another alternative, a *Non-Linear Feedback Shift Register (NLFSR)* whose current state is a non-linear function of its previous state can be used. NLFSRs output sequences are normally very hard to predict and existing cryptanalytic methods, such as algebraic attacks are usually not applicable. An adversary might need $O(2^n)$ bits of the sequence to determine the structure of an *n*-bit NLFSR. A number of different implementations of NLFSR-based stream ciphers for RFID and smartcards applications have been proposed.

Some of the common applications of LFSR include Counters, Built In Self Test (BIST) and Encryption. Few techniques to improve the linear complexity of LFSRs include (a) Non Linear Combining Functions: This technique employs the idea of feeding the outputs of all the LFSRs into a Non Linear Boolean Function to create a Combinational Generator. (b) Clock Controlled Generators: This is another method in which the LFSRs are clocked irregularly, controlled by the output of the second LFSR. Some of the generators include Stop and Go Generator, Alternating Step Generator, Shrinking Generator. The Stop and Go generator consist of two LFSRs, and if the output of one of the LFSRs is '1' the other LFSR is clocked otherwise it repeats the last output. The Alternating Step Generator uses three LFSRs namely LFSR 0, LFSR1, LFSR2. Here, if the output of the third LFSR is '0', then LFSR 0 is clocked and if the output of the LFSR2 is 1 then the LFSR1 is clocked. There are some disadvantages like Fast Correlation Attack and Generalized Inversion Attack. Examples of LFSR based stream ciphers include A5/1 and E0 Stream Ciphers. A5/1 [11] stream Cipher uses three LFSRs that are of different bit length and employing a different feedback functions [12][13].

Generally, NLFSR can be interested by two ways: in the Fibonacci configuration, or in the Galois configuration. The *Fibonacci* configuration, shown in Figure 1, is conceptually more simple. The Fibonacci type of NLFSRs consists of a number of bits numbered from left to right as $n-1, n-2, \ldots, 0$ with feedback from each bit to the $n-1$th bit. At each clocking instance, the value of the bit *I* is moved to the bit $i-1$. The value of the bit 0 becomes the output of the register. The new value of the bit $n-1$ is computed as some function of the previous values of other bits.

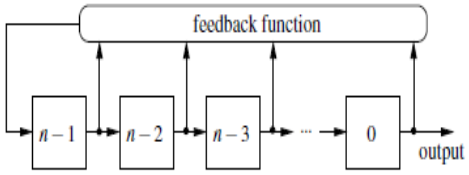

Fig. 1 Fibonacci Configurations

In the *Galois* type of NLFSR, shown in Figure 2, each bit *i* is updated according to its own feedback function [14]. Thus, in contrast to the Fibonacci NLFSRs in which feedback is applied to the $n-1$th bit only, in the Galois NLFSRs feedback can be applied to every bit. Since the depth of the circuits implementing feedback functions of individual bits is usually smaller than the depth of the circuits implementing the feedback function of the Fibonacci NLFSR, the propagation time can potentially be reduced. This makes Galois NLFSRs particularly attractive for stream cipher applications in which high keystream generation speed is important.

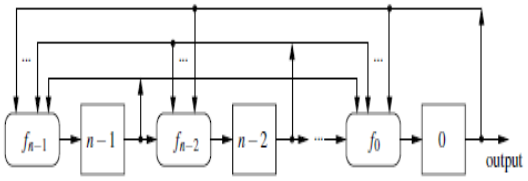

Fig. 2 Galois Configurations

However, Galois NLFSRs also have several drawbacks:

1) The period of the output sequence of a Galois NLFSR is not necessarily equal to the length of the longest cyclic sequence of its consecutive states [15].

2) An *n*-bit Galois NLFSR with the period 2^{n-1} does not necessarily satisfy the 1st and the 2^{nd} randomness postulates

of Golomb. An *n*-bit Fibonacci NLFSR with the period 2^{n-1} always satisfies both postulates.

These drawbacks do not create any problems in the linear case because, for LFSRs, there exist a mapping between the Fibonacci and the Galois configurations. A Galois LFSR generating the same output sequence as a given Fibonacci LFSR (and therefore possessing none of the above mentioned drawbacks) can be obtained by reversing the order of the feedback taps and adjusting the initial state. In the non-linear case, however, no mapping between the Fibonacci and the Galois configurations has been known until now. The problem of finding such a mapping is addressed in this paper. We demonstrate that, for each Fibonacci NLFSR, there exist a class of equivalent Galois NLFSRs, which produce the same output sequence, and show how to transform a given Fibonacci NLFSR into an equivalent Galois NLFSR. This is

carried out in the following three steps. First, we investigate under which conditions a non-linear recurrence of order *n* describing the output sequences of an *n*-bit Galois NLFSR exists. We introduce a structure called *feedback graph*, which reflects the relation between variables of feedback functions. We show that a recurrence of order *n* exists if the feedback graph can be reduced to a single vertex. Second, we examine what kind of feedback functions has feedback graphs which are reducible to a single vertex. We derive a sufficient condition characterizing these feedback functions. We call NLFSRs satisfying this condition *uniform*. Finally, the proof of equivalence of two uniform NLFSRs is done by showing that two systems of non-linear equations describing the sequences of NLFSR's states can be reduced to the same non-linear recurrence.

III. ARCHITECTURE OF STREAM CIPHER

The proposed Exclusive-128 NLFSR (Nonlinear Feedback Shift Register) stream cipher is a 128 bit that generates a 128 bit keystream. The cipher is implemented in such a way that each configuration is sandwiched between the other two configurations. The idea of implementing this new stream cipher is adopted from the A5/1 and modified A5/1 stream ciphers. This generated keystream when XORed with the plaintext produces the ciphertext. The 128-bit keystream is produced using six non-linear feedback shift registers, each used in either the Fibonacci or the Galois configuration [16]. The variant size of each configuration is 32 bit Fibonacci, 31 bit Galois, 25 bit Fibonacci, 32 bit Galois, 4 bit Fibonacci and 4 bit Galois NLFSRs [17]. The architecture of the proposed stream cipher is shown in fig.3

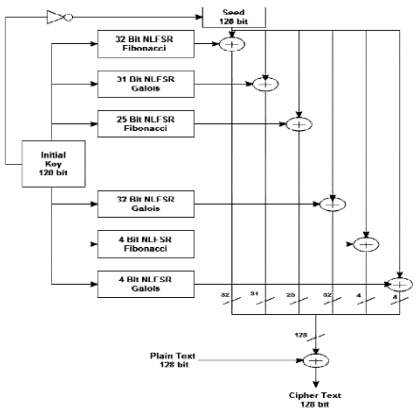

Fig. 3 Architecture if Exclusive 126-bit NLFSR

In this cipher, each configuration of NLFSR is arranged alternately, and their outputs are XORed with the seed value that is the complement of the initial key. This approach enabled us to encrypt and decrypt when both the key and plaintext vectors are simply 0.

IV. RANDOMNESS OF TEST RESULTS

The architecture of Exclusive 128 bit NLFSR based stream cipher contains 6 configurations (each one is either Fibonacci or Galois configuration). Each configuration has a function and this functions are tested by NIST Test Suite such as Frequency Monobit, Frequency within a Block and Run Test in order to test the randomness. Each test suite consists of an algorithm to check the randomness. The objective of this algorithm is to compute P-value. each *P-value* is the probability that a perfect random number generator would have produced a sequence less random than the sequence that was tested, given the kind of non-randomness assessed by the test. If a *P-value* for a test is determined to be greater than 0.01, then the sequence appears to have perfect randomness.

TABLE I
FREQUENCY MONOBIT TEST RESULTS

Configurations	**Functions**	**Frequency Monobit**
32bit Fibonacci	$f_{31} = x_0 \oplus x_2 \oplus x_6 \oplus x_7 \oplus x_{12} \oplus x_{17} \oplus x_{20} \oplus x_{27} \oplus x_{30} \oplus x_3x_9 \oplus x_{12}x_{15} \oplus x_4x_5x_{16}$	0.7237
25 bit Galois	$f_{24} = x_0 \oplus x_1 \oplus x_3 \oplus x_5 \oplus x_6 \oplus x_7 \oplus x_9 \oplus x_{12} \oplus x_{14} \oplus x_{15} \oplus x_{17} \oplus x_{18} \oplus x_{22} \oplus x_1x_6 \oplus x_4x_{13} \oplus x_8x_{16} \oplus x_{12}x_{15} \oplus x_5x_{11}x_{14} \oplus x_1x_4x_{11}x_{15} \oplus x_2x_5x_8x_{10}$	0.4795
4bit Fibonnaci	$f_3 = x_0 \oplus x_1x_3$	0.3173
4 bit Galois	$f_2 = x_3 \oplus x_0x_2$	0.5584
31 bit Galois	$f_{30} = x_0 \oplus x_3 \oplus x_5 \oplus x_7 \oplus x_{10} \oplus x_{16} \oplus x_{17} \oplus x_{18} \oplus x_{19} \oplus x_{20} \oplus x_{21} \oplus x_{24} \oplus x_{30} \oplus x_5x_5 \oplus x_{11}x_{18} \oplus x_{16}x_{22} \oplus x_{17}x_{21} \oplus x_1x_2x_{19} \oplus x_1x_{12}x_{14}x_{17} \oplus x_2x_5x_{13}x_{20}$	0.5373
32 bit Galois	$f_{27} = x_{28} \oplus x_0x_1x_{12}$	1.1427

Table 1 shows that 32 bit Galois is very highly random compared to the remaining configurations. The randomness of each function is tested by Frequency Monobit algorithm.

TABLE II
FREQUENCY WITHIN A BLOCK
TEST RESULTS

Configurations	Functions	Freq within a block
32bit Fibonacci	$f_{31} = x_0 \oplus x_2 \oplus x_6 \oplus x_7 \oplus x_{12} \oplus x_{17} \oplus x_{20} \oplus x_{27} \oplus x_{30} \oplus x_3x_9 \oplus x_{12}x_{15} \oplus x_4x_5x_{16}$	1
25 bit Galois	$f_{24} = x_0 \oplus x_1 \oplus x_3 \oplus x_5 \oplus x_6 \oplus x_7 \oplus x_9 \oplus x_{12} \oplus x_{14} \oplus x_{15} \oplus x_{17} \oplus x_{18} \oplus x_{22} \oplus x_1x_6 \oplus x_4x_{13} \oplus x_8x_{16} \oplus x_{12}x_{15} \oplus x_5x_{11}x_{14} \oplus x_1x_4x_{11}x_{15} \oplus x_2x_5x_8x_{10}$	2
4bit Fibonnaci	$f_3 = x_0 \oplus x_1x_3$	2
4 bit Galois	$f_2 = x_3 \oplus x_0x_2$	4
31 bit Galois	$f_{30} = x_0 \oplus x_3 \oplus x_5 \oplus x_7 \oplus x_{10} \oplus x_{16} \oplus x_{17} \oplus x_{18} \oplus x_{19} \oplus x_{20} \oplus x_{21} \oplus x_{24} \oplus x_{30} \oplus x_5x_5 \oplus x_{11}x_{18} \oplus x_{16} x_{22} \oplus x_{17}x_{21} \oplus x_1x_2x_{19} \oplus x_1x_{12}x_{14} x_{17} \oplus x_2x_5x_{13}x_{20}$	16
32 bit Galois	$f_{27} = x_{28} \oplus x_0x_1x_{12}$	0.25

TABLE III
RUN TEST RESULTS

Configurations	Functions	Run Test
32bit Fibonacci	$f_{31} = x_0 \oplus x_2 \oplus x_6 \oplus x_7 \oplus x_{12} \oplus x_{17} \oplus x_{20} \oplus x_{27} \oplus x_{30} \oplus x_3x_9 \oplus x_{12}x_{15} \oplus x_4x_5x_{16}$	0.4641
25 bit Galois	$f_{24} = x_0 \oplus x_1 \oplus x_3 \oplus x_5 \oplus x_6 \oplus x_7 \oplus x_9 \oplus x_{12} \oplus x_{14} \oplus x_{15} \oplus x_{17} \oplus x_{18} \oplus x_{22} \oplus x_1x_6 \oplus x_4x_{13} \oplus x_8x_{16} \oplus x_{12}x_{15} \oplus x_5x_{11}x_{14} \oplus x_1x_4x_{11}x_{15} \oplus x_2x_5x_8x_{10}$	1.1033
4bit Fibonnaci	$f_3 = x_0 \oplus x_1x_3$	8.5812
4 bit Galois	$f_2 = x_3 \oplus x_0x_2$	4
31 bit Galois	$f_{30} = x_0 \oplus x_3 \oplus x_5 \oplus x_7 \oplus x_{10} \oplus x_{16} \oplus x_{17} \oplus x_{18} \oplus x_{19} \oplus x_{20} \oplus x_{21} \oplus x_{24} \oplus x_{30} \oplus x_5x_5 \oplus x_{11}x_{18} \oplus x_{16} x_{22} \oplus x_{17}x_{21} \oplus x_1x_2x_{19} \oplus x_1x_{12}x_{14} x_{17} \oplus x_2x_5x_{13}x_{20}$	0
32 bit Galois	$f_{27} = x_{28} \oplus x_0x_1x_{12}$	1.8427

Table 2 shows that 32 bit Galois is enormously random compared to the remaining configurations whereas Table 3 shows 4 bit Fibonacci function is highly random.

V. CONCLUSIONS

In this paper, we analyzed security for Exclusive 128-bit Non-Linear Feedback Shift Register (NLFSR) based Stream Cipher. The security analysis is done based on NIST standard Test Suite. This test suite tests randomness of Fibonacci and Galois functions from the architecture of Exclusive 128-bit NLFSR out. Prior to testing the randomness of each function, an algorithm is used to compute the P-value. That means the randomness depends on the P-value, i.e., if P-value > 0.01 then the function is random. In this paper it is clear that functions are certainly random.

REFERENCES

1. M. Hell, T. Johansson, and W. Meier, "Grain - a stream cipher for constrained environments,"

2. M. Robshaw, "Stream ciphers," Tech. Rep. TR - 701, July 1994.

3. R. Mita, G. Palumbo and M. Poli, Pseudo-random sequence generators with improved inviolability performance, IEE Proceedings of Circuits, Devices and Systems, 2006, volume. 153, pp 375-382, 2006.

4. S. E. AlAschkar and M. T. El-Hadidi, Known attacks for the A5/1 algorithm: A Tutorial, International Conference on Information and Communications Technology (ICICT03), pp. 229-251,2003.

5. J. Massey, "Shift-register synthesis and bch decoding," IEEE Transactions on Information Theory, vol. 15, pp. 122–127, 1969.

6. J. D. Golic, "On the linear complexity of functions of periodic GF(q) sequences," IEEE Transactions on Information Theory, vol. 35, no. 1, pp. 69–75, 1989.

7. Y. Tarannikov, "New constructions of resilent Boolean function with maximum nonlinearity," Lecture Notes in Computer Science, vol. 2355, pp. 66–77, 2001.

8. W. Meier and O. Staffelbach, “Fast correlation attacks on certain stream ciphers,” J. Cryptol., vol. 1, no. 3, pp. 159–176, 1989.

9. R. Bialota and G. Kawa, “Modified alternating k generators,” Des.Codes Cryptography, vol. 35, no. 2, pp. 159–174, 2005.

10. K. Zeng, C. Yang, D. Wei, and T. R. N. Rao, “Pseudo-random bit generators in stream-cipher cryptography,” Computer, 1991.

11. Nur Hafiza Zakaria, Kamaruzzaman Seman and Ismail Abdullah, “Modified A5/1 Based Stream Cipher For Secured GSM Communication”, IJCSNS International Journal of Computer Science and Network Security, vol.11 No.2, February 2011.

12. Elena Dubrova,”ATransformation From the Fibonacci to the Galois NLFSRs”, IEEE transactions on information theory, vol.55, no. 11,pp. 5263-5271, November 2009.

13. M. Hell, T. Johansson, and W. Meier, “Grain - a stream cipher for constrained environments,” citeseer.ist.psu.edu/732342.html.

14. K. Zeng, C. Yang, D. Wei, and T. R. N. Rao, “Pseudo-random bit generators in stream-cipher cryptography,” Computer, 1991.

15. E. Dubrova, M. Teslenko, and H. Tenhunen, “On analysis and synthesis of (n, k)-non-linear feedback shift registers,” in Design and Test in Europe, pp. 133–137, 2008.

16. Elena Dubrova, ”Finding Matching Initial States for Equivalent NLFSRs in the Fibonacci and the Galois Configurations”, IEEE transactions on information theory, vol. 56, no. 6, June 2010.

17. Andrew Rukhin, Juan Soto,” A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications”, National Institute of Standards and Technology (NIST), US department of Commerce, April 2010.

