Circuits and Systems, 2013, 4, 165-168
http://dx.doi.org/10.4236/cs.2013.42022 Published Online April 2013 (http://www.scirp.org/journal/cs)

The Nonlinear Filter Boolean Function of LILI-128 Stream Cipher Generator Is Successfully Broken Based on the Complexity of Nonlinear 0 1 Symbol Sequence

Xiangao Huang[1], Chao Wang[2], Wei Huang[3], Junxian Li[1]
[1]Beijing Institute of Technology, Zhuhai Campus, Zhuhai, China
[2]Computer Center, Chang'an University, Xi'an, China
[3]Research Institute of Electronics, Xi'an, China
Email: xiangaohuang@yahoo.com.cn

Received January 24, 2013; revised February 24, 2013; accepted March 4, 2013

ABSTRACT

The nonlinear filter Boolean function of LILI-128 stream cipher generator is studied in this paper. First we measure the complexity of the stream ciphers sequence of LILI-128 stream cipher generator and obtain the shortest bit stream sequence reconstructed Boolean function of nonlinear filter in LILI-128 stream cipher generator. Then the least nonlinear Boolean function of generating stream cipher sequence is reconstructed by clusterig, nonlinear predictive and nonlinear synchronization from shortest bit stream sequence. We have verified the correctness of our reconstruction result by simulating the block diagram of Lili-128 keystream generator using our getting Boolean function and implement designers' reference module of Lili-128 stream cipher public online, and two methods produce the same synchronous keystream sequence under same initial state, so that our research work proves that the nonlinear Boolean function of LILI-128 stream cipher generator is successfully broken.

Keywords: LILI-128 Stream Cipher; Clock Control; Boolean Function; Complexity; Attack

1. Introduction

Our society greatly depends on security of communications, financial transactions, telematic services, internet and mobile networks [1] which in turn present new challenges for protecting the information from unauthorized eavesdropping. Cryptography mainly uses two types of symmetric algorithms, block ciphers and stream ciphers. The block ciphers have become widely used technology. As an example AES is a secure block cipher that offers excellent performance on a variety of hardware and software environments. On the other hand, the stream ciphers are widely used in secure communication because of high throughput, less complex hardware circuitry and very little error propagation which has attracted much attention. An important class of stream ciphers is based on a mixture of linear feedback shift register (LFSR), nonlinear filter generators and also clock-controlled generators [2,3]. LILI-128 stream cipher is an example which is designed by Dawson, Clark, Golic, Millan, Penna and Simpson, which submitted to NESSIE (New European Schemes for Signatures, Integrity and Encryption) as a candidate cipher [3]. According to the final report, LILI-128 stream cipher was rejected in the first round of NESSIE. In this work we will discuss basic structure and the attack of LILI-128 stream cipher. We particularly study the filter Boolean function of LILI-128 stream cipher generator.

2. The Structure of LILI-128

The structure of the LILI-128 keystream generators is illustrated in **Figure 1**. It uses two binary $LFSR_c$ and $LFSR_d$ and two functions f_c and f_d to generate a pseudo-random binary keystream sequence. At initialization, 128 bit key provides the initial states of the $LFSR_c$ and $LFSR_d$.

The generator can be divided into two subsystems based on the functions they perform: the clock control subsystem and data generation subsystem. The clock control subsystem produces an integer sequence that is used to control data generation subsystem. The feedback polynomial of the $LFSR_c$ is chosen to be the primitive polynomial

$$G_c(x) = x^{39} + x^{35} + x^{33} + x^{31} + x^{17} + x^{15} + x^{14} + x^2 + 1 \quad (1)$$

Since $G_c(x)$ is primitive, the $LFSR_c$ produce a maximum-length sequence of period $P_c = 2^{39} - 1$. The function f_c takes two bits as input and produced an integer c_k such that $c_k \in \{1,2,3,4\}$. The value of c_k is calculated as

$$c_k = f_c(y_1, y_2) = 2y_1 + y_2 + 1, k \geq 1 \quad (2)$$

The $LFSR_d$ is clocked by c_k at least once and at most four times, which is given as fllows:

$$G_d(x) = x^{89} + x^{83} + x^{80} + x^{55} + x^{53} + x^{42} + x^{39} + x + 1 \quad (3)$$

since $G_d(x)$ is a primitive polynomial, a period of $P_d = 2^{89} - 1$ at maximum is guaranteed for $LFSR_d$ output sequence. The contents of 10 different stages of $LFSR_d$ are input to a nonlinear filter function f_d. The output z_k of nonlinear filter function f_d is the keystream sequence.

3. Security Analysis

LILI-128 stream cipher are a long period around 2^{128}, high linear complexity which is conjectured to be at least 2^{68}, and good statistics regarding the distribution of zeroes and ones, so designers claim that the LILI-128 keystream generator can resist currently known styles of attack. Some methods [4-7] of breaking it have been proposed, since LILI-128 stream cipher was publicized in 2000. The methods have already been shown that some attack break the LILI-128 stream cipher more efficiently than an exhaustive search for its secret key. However, most of the attack methods consider only the complexity of time or memory for search for its secret key. For example, Time-Memory Tradeoff Attack [4] needs approximately 2^{46} bits, and Correlation Attack [5] needs approximately about 2^{23} bits. While algebraic Attack [6] needs approximately 2^{18} bits. Even a new attack method [7] requires a mere 2^7 bits of keystream, but needs $2^{99} - 1$ computations. The above styles of attack are only quailtative analysis, without an actual example of the successful attack. In 2005, designers summarize recently published styles of attack on the LILI-128 stream cipher, and assert that LILI-128 remains unbroken [8]. They encourage further analysis of the LILI-128 stream cipher.

4. The Expression of the Nonlinear Filter Function f_d

In **Figure 1**, designers do not publicize the expression of the nonlinear filter function f_d. Up to now, all attacks explain only that LILI-128 has lower complexity as designers claim, and do not get the expression of the nonlinear filter function f_d. The aim which we attack LILI-128 is the expression of the nonlinear filter function f_d. The reference module of LILI-128 is got from the Information Security Institute's webpage: http://www.isi.qut.edu.au/resources/lili/. We implement the reference module on ASCII initial value "yyyy yyyyyyyyyy y" and get the keystream sequence of LILI-128, and study the expression of the nonlinear filter function f_d. First, we determine the least keystream bit amount of expressing the nonlinear filter function f_d from the LILI-128 kystream sequence by means of measuring the complexity of the LILI-128 kystream sequence. The least keystream bit amount of expressing the nonlinear filter function f_d is not equal for differ initial state. The least keystream bit amount is approximately 2^{12} - 2^{13} bits. We reconstruct the expression f_d from Shortest bit stream sequence by clusterig, nonlinear predictive and nonlinear synchronization, which has 46 items from liner items to nonlinear polynomials with 6 orders as follows:

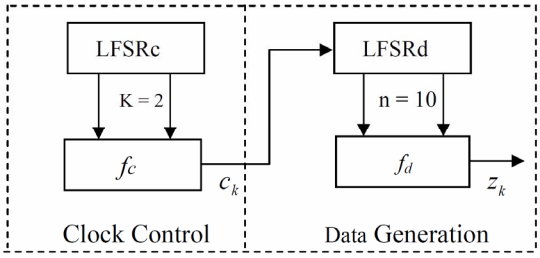

Figure 1. Overview of LILI-128 keystream generators.

$$\begin{aligned} f_d = {} & x_6 + x_7 + x_8 + x_9 + x_1 \cdot x_5 + x_1 \cdot x_7 + x_2 \cdot x_8 \\ & + x_2 \cdot x_{10} + x_3 \cdot x_9 + x_3 \cdot x_{10} + x_4 \cdot x_5 + x_1 \cdot x_2 \cdot x_6 \\ & + x_1 \cdot x_2 \cdot x_7 + x_1 \cdot x_2 \cdot x_8 + x_1 \cdot x_2 \cdot x_9 + x_1 \cdot x_3 \cdot x_7 \\ & + x_1 \cdot x_3 \cdot x_8 + x_1 \cdot x_4 \cdot x_5 + x_1 \cdot x_4 \cdot x_6 + x_1 \cdot x_4 \cdot x_7 \\ & + x_2 \cdot x_3 \cdot x_5 + x_2 \cdot x_3 \cdot x_8 + x_2 \cdot x_4 \cdot x_5 + x_2 \cdot x_4 \cdot x_7 \\ & + x_2 \cdot x_4 \cdot x_8 + x_1 \cdot x_2 \cdot x_3 \cdot x_5 + x_1 \cdot x_2 \cdot x_3 \cdot x_7 \\ & + x_1 \cdot x_2 \cdot x_3 \cdot x_8 + x_1 \cdot x_2 \cdot x_3 \cdot x_{10} + x_1 \cdot x_2 \cdot x_4 \cdot x_5 \\ & + x_1 \cdot x_2 \cdot x_4 \cdot x_7 + x_1 \cdot x_2 \cdot x_4 \cdot x_9 + x_1 \cdot x_3 \cdot x_4 \cdot x_6 \\ & + x_1 \cdot x_3 \cdot x_4 \cdot x_8 + x_2 \cdot x_3 \cdot x_4 \cdot x_7 + x_2 \cdot x_3 \cdot x_4 \cdot x_9 \\ & + x_2 \cdot x_4 \cdot x_5 \cdot x_6 + x_2 \cdot x_4 \cdot x_5 \cdot x_7 + x_1 \cdot x_2 \cdot x_3 \cdot x_4 \cdot x_7 \\ & + x_1 \cdot x_2 \cdot x_3 \cdot x_4 \cdot x_8 + x_1 \cdot x_2 \cdot x_4 \cdot x_5 \cdot x_6 \\ & + x_1 \cdot x_2 \cdot x_4 \cdot x_5 \cdot x_7 + x_2 \cdot x_3 \cdot x_4 \cdot x_5 \cdot x_6 \\ & + x_2 \cdot x_3 \cdot x_4 \cdot x_5 \cdot x_7 + x_1 \cdot x_2 \cdot x_3 \cdot x_4 \cdot x_5 \cdot x_6 \\ & + x_1 \cdot x_2 \cdot x_3 \cdot x_4 \cdot x_5 \cdot x_7 \end{aligned} \quad (4)$$

5. Verifying the Correctness for the Filter Boolean Function f_d

If we describe LILI-128 keystream generator by MATLAB, the stages of $LFSR_c$ be labeled $s[1]$, $s[2]$,…, $s[39]$ from left to right. At every time, we have the following formula to calculate the feedback bit:

$$\begin{aligned} w = {} & s[38] \oplus s[26] \oplus s[25] \oplus s[23] \\ & \oplus s[9] \oplus s[7] \oplus s[5] \oplus s[1] \end{aligned} \quad (5)$$

where $\oplus$ indicates the addition modulo 2. Let the $LFSR_c$ shift left, and $s[38] = w$. Sequentially circulating, the $LFSR_c$ will produce a linear pseudorandom sequence. The function f_c is given by

$$f_c(x_{13}, x_{21}) = 2 \cdot x_{13} + x_{21} + 1 \tag{6}$$

The stages of $LFSR_d$ be labeled $u[1]$, $u[2]$, ⋯, $u[89]$ from left to right. At time k, the feedback bit is calculated by the following formula

$$\begin{aligned} w = {} & u[89] \oplus u[51] \oplus u[48] \oplus u[37] \\ & \oplus u[35] \oplus u[10] \oplus u[7] \oplus s[1] \end{aligned} \tag{7}$$

where $\oplus$ indicates the addition modulo 2. Let the $LFSR_d$ shift left, and $u[89] = w$. According to above circulation, the $LFSR_d$ will produce a linear pseudorandom sequence. We map the set: (1, 2, 3, 4, 5, 6, 7, 8, 9, 10) into the set: (1, 2, 4, 8, 13, 21, 31, 45, 66, 81), the f_d is given by:

$$\begin{aligned} f_d = {} & x_{21} + x_{31} + x_{45} + x_{66} + x_1 \cdot x_{13} + x_1 \cdot x_{31} + x_2 \cdot x_{45} \\ & + x_2 \cdot x_{81} + x_4 \cdot x_{66} + x_4 \cdot x_{81} + x_8 \cdot x_{13} + x_1 \cdot x_2 \cdot x_{21} \\ & + x_1 \cdot x_2 \cdot x_{31} + x_1 \cdot x_2 \cdot x_{45} + x_1 \cdot x_2 \cdot x_{66} + x_1 \cdot x_4 \cdot x_{31} \\ & + x_1 \cdot x_4 \cdot x_{45} + x_1 \cdot x_8 \cdot x_{13} + x_1 \cdot x_8 \cdot x_{21} + x_1 \cdot x_8 \cdot x_{31} \\ & + x_2 \cdot x_4 \cdot x_{13} + x_2 \cdot x_4 \cdot x_{45} + x_2 \cdot x_8 \cdot x_{13} + x_2 \cdot x_8 \cdot x_{31} \\ & + x_2 \cdot x_8 \cdot x_{45} + x_1 \cdot x_2 \cdot x_4 \cdot x_{13} + x_1 \cdot x_2 \cdot x_4 \cdot x_{31} \\ & + x_1 \cdot x_2 \cdot x_4 \cdot x_{45} + x_1 \cdot x_2 \cdot x_4 \cdot x_{81} + x_1 \cdot x_2 \cdot x_8 \cdot x_{13} \\ & + x_1 \cdot x_2 \cdot x_8 \cdot x_{31} + x_1 \cdot x_2 \cdot x_8 \cdot x_{66} + x_1 \cdot x_4 \cdot x_8 \cdot x_{21} \\ & + x_1 \cdot x_4 \cdot x_8 \cdot x_{45} + x_2 \cdot x_4 \cdot x_8 \cdot x_{31} + x_2 \cdot x_4 \cdot x_8 \cdot x_{66} \\ & + x_2 \cdot x_8 \cdot x_{13} \cdot x_{21} + x_2 \cdot x_8 \cdot x_{13} \cdot x_{31} + x_1 \cdot x_2 \cdot x_4 \cdot x_8 \cdot x_{31} \\ & + x_1 \cdot x_2 \cdot x_4 \cdot x_8 \cdot x_{45} + x_1 \cdot x_2 \cdot x_8 \cdot x_{13} \cdot x_{21} \\ & + x_1 \cdot x_2 \cdot x_8 \cdot x_{13} \cdot x_{31} + x_2 \cdot x_4 \cdot x_8 \cdot x_{13} \cdot x_{21} \\ & + x_2 \cdot x_4 \cdot x_8 \cdot x_{13} \cdot x_{31} + x_1 \cdot x_2 \cdot x_4 \cdot x_8 \cdot x_{13} \cdot x_{21} \\ & + x_1 \cdot x_2 \cdot x_4 \cdot x_8 \cdot x_{13} \cdot x_{31} \end{aligned} \tag{8}$$

We simulate the keystream sequence of LILI-128 in **Figure 1** using (6) by 128 bits initial values which are ASCII "yyyyyyyyyyyyyyyy". Compare simulating result with the result of implementing the reference module, and two methods produce the same synchronous keystream sequence under same initial state "yyyyyyyyyyyyyyyy". We verify the nonlinear filter function f_d again by initial state "gggggggggggggg" and "123456789abcdefg", and obtain all the same synchronous keystream sequence, so that the validation work indicates we have successfully broken the nonlinear filter Boolean function of LILI-128 stream cipher generator.

6. Conclusion

In the design of LILI-128, Designers made an attempt to confuse the linear pseudorandom binary sequence with the long period $P_c = 2^{89} - 1$ by the clock-control with the long period $P_c = 2^{39} - 1$ and get high linear complexity of keystream sequence. The nonlinear sequence is generated through the nonlinear filter function f_d with 46 items and the most algebraic 6 orders to withstand all kinds of currently known attack. The LILI-128 keystream generator certainly resists currently known styles of attack, but it does not withstand our attack. The currently known styles of attack based on time complexity and memory complexity of arithmetic. The attacks do not consider the complexity of keystream sequence oneself. We get the least bit amount of the attack by measuring the complexity of the keystream sequence of LILI-128 and the nonlinear filter function f_d from the least keystream bit amount by the phase space reconstruction, Clustering, nonlinear prediction and nonlinear synchronization. In this paper, our research work has made a great breakthrough in stream cipher analysis, breaks the dream that stream cipher of LILI-128 is not attacked, and will bring the importance influence on stream cipher design. We only publish our research result and do not expatiate the specific theory and algorithm of attacking LILI-128 stream cipher because our attack method has important application in attacking military stream cipher. Reader verifies above f_d first of all. And then f_d is redesigned. The stream cipher sequence of LILI-128 output is obtained by simulating **Figure 1** given an ASCII initial state. Reader sends the ASCII initial state and the stream cipher sequence whose length is 2^{13} bits to us. We will return f_d to him. The Boolean function f_d of nonlinear filter is got from known stream ciphers sequence, which belongs to research areas of blind signal processing.

REFERENCES

[1] B. Schneier, "Applied Cryptography," 2nd Edition, John Wiley and Sons, Hoboken, 1996.

[2] R. E. Atani, *et al.*, "Alamout: A New Synchronous Stream Cipher with Authentication," *IEEE Proceedings of the* 1*st International Conference on Wireless Communication*, *Vehicular Technology*, *Information Theory and Aerospace & Electronic Systems Technology*, 17-20 May 2009, pp. 4244-4067.

[3] E. Dawson, A. Clark, J. Golic, W. Millan, L. Penna and L. Simpson, "The LILI-128 Keystream Generator," *NESSIE Proceedings of the Fist Open NESSIE Workshop*, Leuven, November 2000. Http://www. cryponessie.org.

[4] M. J. O. Saarinen, "A Time-Memory Trade of Attack against LILI-128," In: *Fast Software Encryption* (*Lecture Notes in Computer Science*), Springer-Verlag, Berlin, 2002, pp. 231-236. doi:10.1007/3-540-45661-9_18

[5] H. Molland and T. Helleseth, "An Improve Correlation Attack against Irregular Clocked and Filtered Keystream Generators," In: *Advance in Cryptology-Crypto* 2004

(*Lecture Notes in Computer Science*), Springer-Verlag, Berlin, 2004, pp. 373-389. doi:10.1007/978-3-540-28628-8_23

[6] N. T. Courtois, "Fast Algebraic Attack on Stream Ciphers with Linear Feedback," In: *Advance in Cryptology-Crypto* 2003 (*Lecture Notes in Computer Science*), Springer-Verlag, Berlin, 2003, pp. 176 - 194.

[7] Y. Tsunoo, T. Saito, M. Shigeri, H. Kubo and K. Minematsu, "Shorter Bit Sequence Is Enough to Break Stream Cipher LILI-128," *IEEE Transactions on Information Theory*, Vol. 51, 2005, pp. 4312-4319. doi:10.1109/TIT.2005.859285

[8] W. Millan and E. Dawson, "LILI-II Is Not Broken," 2005. http//eprint.iacr.org/complete/2005/234

