Enhanced Stream Cipher Algorithm using Consecutive Nonlinear Functions

Hisham S. Elganzoury
Al-Azhar University
Nasr City
Cairo, Egypt

Talaat A. El-Garf
Military Tech. Col.
Nasr City
Cairo, Egypt

A.A. Hafez
Military Tech. Col.
Nasr City
Cairo, Egypt

Ahmed Safwat
Al-Azhar University
Nasr City
Cairo, Egypt

ABSTRACT
Confidentiality is a security service that keeps the information from all but those authorized to have it. It needs an efficient cryptographic algorithm. Stream cipher is considered a very important class of symmetric encryption algorithms used to achieve that goal. Its basic design philosophy is inspired by the one-time-pad cipher, which encrypts by XOR'ing the plaintext with a random key. However, the need for a key of the same size as the plaintext makes the one-time-pad impractical for most applications. Instead, stream ciphers expand a given short random key into a pseudo-random key stream, which is then XOR'ed with the plaintext to generate the output ciphertext. This paper suggests a nonlinear balanced stream cipher algorithm which provides high nonlinearity, high linear complexity, high correlation immunity, large Hamming Distance, long key period and good randomness properties exploiting consecutive nonlinear functions. This algorithm is then implemented on a FPGA Kit using VHDL to illustrate its applicability to modern communication systems such as smart phones and PDAs.

General Terms
Security, Symmetric Encryption Algorithms, Confidentiality

Keywords
Reduction Function, Consecutive Nonlinear Functions, Non-linear Stream Cipher Algorithms, Random Sequence Tests

1. INTRODUCTION
The high increasing demand of securing communicated information has forced researchers and system designers to develop more complicated cryptographic algorithms. The design goal of stream ciphers is to efficiently generate pseudorandom bits which are indistinguishable from truly random bits [1]. Stream Cipher algorithms have the advantage of software efficiency and short processing time [2]. The construction of secure running key generators necessarily implies the introduction of nonlinear transformations which greatly complicates the indispensable analysis. There exist implicit and explicit methods for introducing nonlinear effects. To exploit an algorithm in most modern communication devices such as smart phones and PDAs, it has to achieve minimum key size, and processing complexity. These requirements are not offered by most stream cipher algorithms. Explicit methods directly apply nonlinear functions; in the autonomous automaton description of the running key generator (Fig.1) where σ_j is the current state of the cell, f_0 is the output function, f_s is the next state function, K is the key, and Z_j is the output symbol.

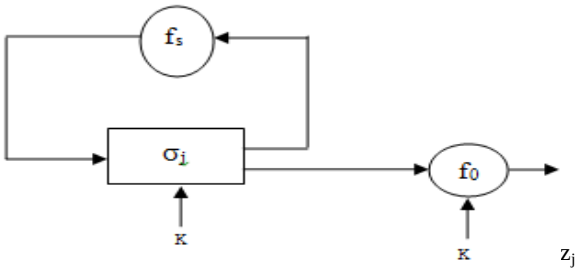

Figure 1: The key generator as an autonomous finite state machine

The next state and the output mappings are candidates for nonlinear transformations. Unfortunately, the theory of autonomous automaton whose change of state function is nonlinear (e.g. Nonlinear Feedback Shift Register) is not very well developed so that a running key generator employing nonlinear feedback will only be very limitedly analyzable.

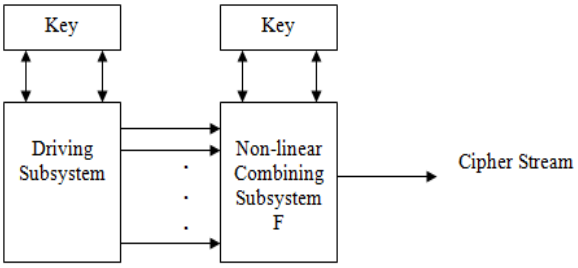

Figure 2: The conceptual distinction between driving and nonlinear combining subsystem in a general key stream generator

Different is the situation when a linear autonomous automaton is combined with a nonlinear output mapping to be used as running key generator. In this paper, it is convenient to subdivide the running key generator into a driving part and a combining part (Fig.2) [3]. The remainder of the paper is structured as follows: Section 2 introduces related works, while Section 3 describes the proposed stream cipher algorithm. Then section 4 shows the security analysis applied to the proposed algorithm. At last, Section 5 concludes the paper and indicates future work.

2. RELATED WORKS
Many stream ciphers were developed to achieve the high nonlinearity and complexity. Here are some of those proposals. In [2] the author proposed a new stream cipher algorithm comprising two parts, driving and combining parts. The driving part consists of 16 LFSRs of 32 cells (bits). A middle stage is introduced for bit-reorganization; it extracts

four outputs of length 32 bits from the cells of the LFSR and combines them. Although the algorithm provided a good way to achieve nonlinearity and maximal length properties, it doesn't completely remove the input stream effect on the output which, in fact, makes the communication systems vulnerable to attacks especially when the attacker has some pieces of input and output streams. In addition, nonlinearity is based on one stage of combination which is not efficient to conceal the input stream like using MUX'ing. That is why the author required the S-Box and Q-Box (Arrays of fixed elements to use in the combining stage) to be kept as secret as the secrete key. Regarding implementation in limited environments such as smart phones, 16 LFSRs is a large number while the same security level can be achieved using fewer number of LFSRs. At last, using LFSRs with same lengthes may reveal the code word length that is used in some attacks such as guess and determine attacks. In [4] the Author proposed a solution to encrypt plain images using stream cipher algorithm. Fourteen NLFSRs of lengths between 20 and 33 bits are used with 8th-order correlation immune Boolean combining function of fourteen variables and has algebraic degree 4 and nonlinearity. Regarding applicability to smart phones, tablets and PDAs, fourteen NLFSRs is a large number whose security level can be achieved with less number of NLFSRs. The nonlinear function is ACHTERBAHN-128/80[5] which has a maximal key stream length of limited to 2^{63} bits. The used nonlinear function doesn't conceal the input effect on the output stream. At last the algebraic immunity degree of 4 makes this approach vulnerable to algebraic attacks.

3. PROBLEM DESCRIPTION AND PROPOSAL

The combining Subsystem should transfer the statistical properties of the periodic driving sequences to the generated running key in the sense that, when the input sequence are true q-ary coin tossing sequences, so is the output sequence, maximize the period of the running key relative to the periods of driving sequences which provide the needed high potential for a large linear complex of the key stream, maximize the linear complexity of the running key, prevent leakage to avoid any modularizing attack directed towards the sub modules of the driving subsystem of the key stream generator. Our proposed algorithm aims to achieve those requirements by introducing more than one high nonlinear part such as rolling drivers, multiplexers, nonlinear reduction/selection functions and combination functions. The algorithm accepts 192 input bits and XOR's them with 192 key bits. As shown in Fig.3 the algorithm consists of driving part and nonlinear combining part. Driving part consists of seven MLFSRs with co-prime lengths, these MLFSRs are combined to ensure the confusion and diffusion before being fed to the combining part.

The feedback (characteristic) polynomials of the seven LFSRs are:

$F1=1+X+X^2+X^3+X^4+X^5+X^{37}$

$F2=1+X^{24}+X^{31}$

$F3=1+X^{27}+X^{29}$

$F4=1+X^{18}+X^{23}$

$F5=1+X^{13}+X^{17}+X^{18}+X^{19}$

$F6=1+X^{38}+X^{41}$

$F7=1+ X^{37}+ X^{38}+X^{52}+X^{53}$

Rolling Drivers

Rolling drivers are based on "Rolling Arrays "introduced in [8] with some modifications. The first rolling driver array is filled with numbers from 0 to 127 (i.e. 128 elements) in a random manner while the other rolling driver array is filled with numbers from 128 to 159 (i.e. 32 elements). With ref to Fig.3, Two Rolling drivers are involved; the first is at the input of (CF1/COMB1) while the second is at the input of (CF2/COMB2). The indirect accesses through rolling driver add a greater amount of complexity to the mixing process, in a way that is highly nonlinear and hard for cryptanalysts to follow.

The Nonlinear and Reduction function

Referring to Fig.4, The nonlinear and Reduction function consists of one LFSR with maximum length and a nonlinear Reduction/Selection function S. The maximum LFSR has the length of 192 bits and its primitive feedback polynomial is: $F8=1+X_{59}+X_{94}+X_{113}+X_{143}+X_{181}$. Referring to table 1, function S is permanently fixed and highly nonlinear combining function. Its selection is based on the requirement of producing sequences with controllable complexity. Table 1 indicates the look up table of the S-function. The Algebraic Normal Form (ANF) of the S-function contains nonlinear terms of different degrees (Zero degree and up to eight degrees) whose values are very close to the values expected for random function as illustrated in table 1.

The CF1 (COMB 1)

This stage is to ensure removal of the input statistics from the output stream. It accepts seven input bits per time from the Rolling Driver and then handles the output bits as follows: F_0 is a fixed 0; F1 is a selective function which uses the last 5 bits of the input to select one bit per clock from a varying 2X16 array. It is a 2X16 array, which uses the first bit to select between the rows and the other 4 bits to select among the columns, then the intersecting bit is the required bit. The other outputs start as follows: the third output (no.2) =input (no.5); fourth output (no.3) = input (4); fifth output (no. 4) = input (3); sixth output (no. 5) =input (2); seventh out (no.6) = input (1); and eighth out (no. 7) = input (0); each time the algorithm is run the output/input mapping is shifted by one. I.e. the second run will give the following: fourth output (no.3) =input (3); fifth output (no.4) =input (2), and so on. F2 is delayed by T (one clock); XOR'ed with F0 and then shifts the contents of the Shift Register SR.

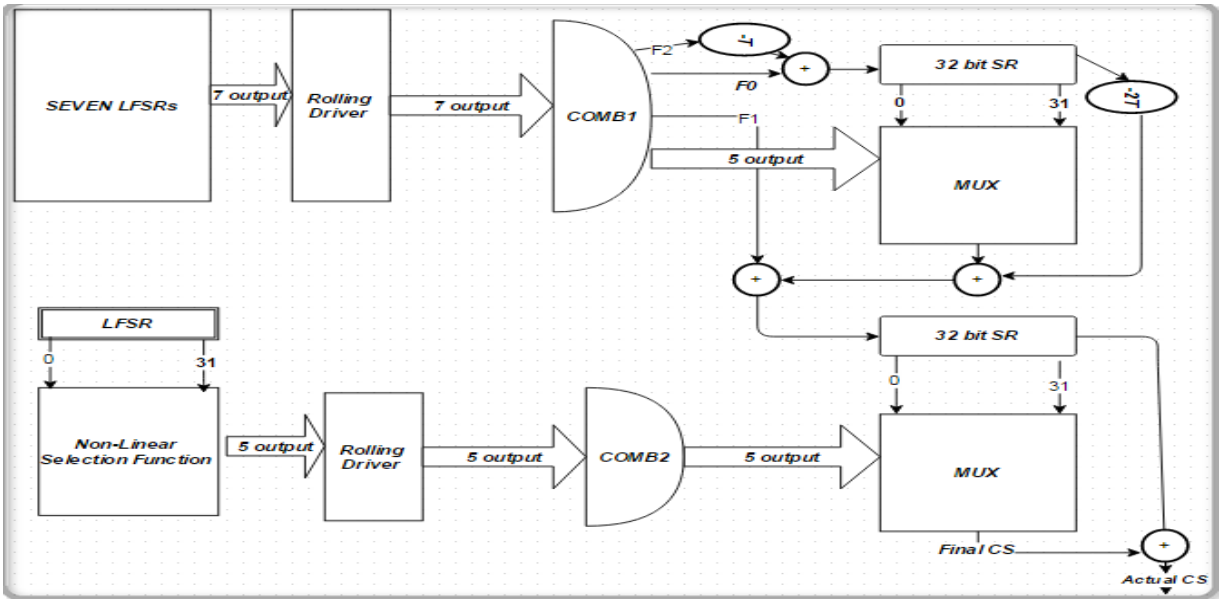

Figure 3: Proposed Stream Cipher Algorithm

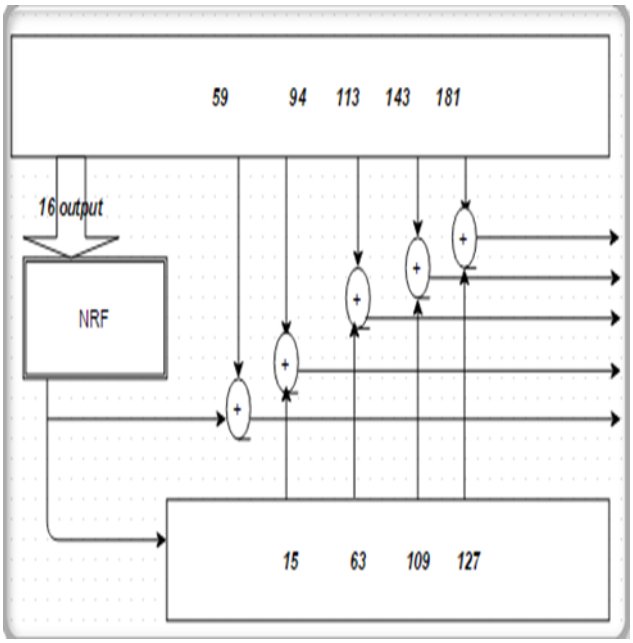

Figure 4: Non-linear Reduction Function

The CF2 (COMB2)

It works as follows: The output (0) = the input (no.3); the output (1) = the input (no. 0); the output (2) = the input (no.2); the output (3) = the input (no. 4); and the output (4) = the input (no.1).

MUX'ing and XOR'ing functions

Referring to Fig.3, two MUXs and two SRs are used to provide an additional stage of nonlinearity, diffusion, and complexity. Both MUX's are 32X1 multiplexers which use five bits as selector. Each uses the combination (COMB1 or CF1and COMB2 or CF2) output to select one of their inputs.

At last the output of the second 32-bits SR is XOR'ed with the second MUX output to produce the actual code sequence (Actual_CS).

Table 1.Nonlinear Function(S)

S	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	1	0	0	1	1	0	1	0	1	1	0	0	0	0	0	1
1	1	1	1	0	0	1	1	0	0	0	1	1	1	1	1	0
2	0	1	0	1	1	1	1	0	1	1	1	1	1	0	0	1
3	1	0	0	1	0	0	0	0	0	1	0	1	1	1	1	0
4	1	1	1	0	0	1	1	1	0	0	1	0	1	1	0	0
5	0	1	0	1	0	1	0	1	0	1	0	1	0	1	1	0
6	0	1	0	1	0	0	0	1	0	0	1	0	0	1	0	0
7	0	0	0	1	0	0	1	1	0	0	1	1	0	1	0	1
8	0	0	0	0	1	0	1	1	1	0	0	1	1	0	1	1
9	1	0	0	1	0	0	1	0	0	0	0	1	0	0	0	0
A	0	1	1	0	1	0	1	1	1	0	0	1	1	1	0	1
B	1	0	0	1	0	0	1	0	0	1	1	0	1	0	1	1
C	1	0	0	1	1	1	1	0	0	1	0	1	0	0	1	0
D	0	1	1	0	1	1	1	0	1	1	1	0	1	1	0	1
E	0	1	1	0	0	0	1	0	0	1	0	1	0	0	1	1
F	1	1	0	1	1	1	0	1	0	0	0	1	0	0	0	1

Table 2.ANF of different degrees

Degree	No. of appearing terms in the LUT	No. of expected terms for random functions
0	**1**	**0.5**
1	**5**	**4**
2	**13**	**14**

3	**30**	**28**
4	**33**	**35**
5	**27**	**28**
6	**17**	**14**
7	**7**	**4**
8	**1**	**0.5**
9	**5**	**4**
10	**13**	**14**

4. SECURITY ANALYSIS

The proposed algorithm is designed to provide a high level of complexity, nonlinearity, randomness, balance and diffusion. Those requirements are satisfied. For example, diffusion level is enhanced by using Non-linear reduction function, Rolling driver, Combination function, and MUX'ing functions. In the below sub-sections a detailed analysis is performed as follows.

4.1 Possible Attacks

4.1.1 Guess and Determine Attack

In [6] the guess-and-determine attack is defined as getting some internal states of LFSRs and then determining the remaining states using a suitable interpolation technique. The number of internal states of the algorithm has 37+31+29+23+19+41+53+192+32×2=489 possible states with different LFSRs lengths, so it is very hard to guess the keyword length that is required to guess the internal states.

4.1.2 BDD Attack

Binary Decision Diagram (BDD) attack, this attack is no longer available for states that are more than 128bits [7].

4.1.3 The Berlekamps-Massey Attack

The Berlekamps-Massey attack [9] requires 2A(y) data Successive. In order to mount a Berlekamp-Massey attack, the key stream generator must produce a key stream with linear complexity as high as possible. The lower bound of the linear complexity A(y) used in this encryption scheme satisfies A(y) >> 2^{200}, this bound is sufficiently large which implies that we completely excludes to use the Berlekamp-Massey attack

4.1.4 Correlation Attack

The correlation attack of Sigenthaler [10] can be evaluated using the sum of lengths of the shortest NLFSRs of the key stream generator that is 37+31+29+23+19+41+53+32×2= 297. Therefore, the complexity of Siegenthaler's correlation attack against this proposal is at least O (2^{297}). This type of attack is theoretically too hard.

4.1.5 Algebraic Attack

Algebraic attack problem due to low nonlinear degree of the combination function, as the values vary from 0000 to 1111 can be compensated using consecutive levels of nonlinear elements such as rolling driver and MUX'ing

4.2 Statistical Tests

Now the results of statistical tests performed using NIST test package [11] are introduced. These tests aim to examine the randomness of the output stream. The Significance probability P-value represents the probability of obtaining a result further than the test statistic lies from the expected, if the algorithm produces a random stream. Very small P-values would support non-randomness for the given measure. The tests which are performed on the output of 2MByte file are Frequency test, Linear Complexity test, Runs test, Chi-Square test, Binary Derivative test, Serial test, Block Frequency test, and Fast Fourier transform test. The following table shows a comparison between some other algorithms and the current proposal

Table 3.Comparison between the proposed algorithm and some other algorithms

Test	Proposed algorithm	A5	RC4	Rabbit	LEA
Frequency test	0.6891	0.436	0.83	0.214	0.624
Linear complexity	0.9003	0.8341	0.500	0.8341	0.9739
Runs test	0.8070	0.7696	0.779	0.3083	0.8859
Chi-Square Test	0.913	0.5477	0.6960	0.0278	0.9055
Binary derivative test	0.6154	0.3404	0.753	0.5158	0.5490

While the Serial test shows P-value=0.6891, the Block Frequency Test shows P-value = 0.9505, and the Fast Fourier transform test shows P-value =0.9003.

4.3 Theoretical Analysis [3]

4.3.1 Nonlinearity

The proposed algorithm possesses more than one nonlinear function such as COMB1 and COMB2, Multiplexers, nonlinear S-function as shown in table.1, and the rolling driver.

4.3.2 The key period

The period of the produced sequence can be calculated as follows, P= (2^{37}-1) (2^{31}-1) (2^{29}-1) (2^{23}-1) (2^{19}-1) (2^{41}-1) (2^{53}-1) (2^{192}-1) ≅ 2^{425}=8×10^{127}>>10^{120} which is a very large value as required

4.3.3 Key Diversity

Key diversity is the possible keys can the algorithm produce and mathematically is $2^{keySize}$. For the current proposal the key diversity is 2^{192} keys

4.3.4 Linear Complexity

Linear Complexity (LC) is an important parameter used to judge the cryptographic performance of the bit sequence produced by stream ciphers. The value of LC depicts the cryptographic strength of the cipher. The linear complexity of parallel LFSRs with their periods of n_1, n_2, n_3… where (n_1, n_2, n_3…) =1, can be calculated as follows LC=n_1n_2+n_2n_3 +n_3n_4+ n_4n_5+n_5n_6+n_6n_7+n_7n_8+n_8.For the mentioned LFSRs' lengths; LC≈13259. This means that 13259-bit LFSR is required to generate the sequence produced by the proposed cipher

5. CONCLUSION AND FUTURE WORK

Referring to Sec.3 and 4, the proposed modified algorithm satisfied the given assumptions as it passed the statistical tests and showed an enhancement to the previously developed

ciphers. It walked a step towards reducing the processing time and capacity required for applicability on resource limited environments such as smart phones and PDAs. In addition, using different length LFSRs hardened the code word size guess. Also, using consecutive nonlinear functions with the rolling manner increased the rigidness, complexity, nonlinearity, key period and randomness of the cipher. The upcoming efforts will be concentrated on reducing the key size to have the optimum size for smart phones, tablets and PDAs as the modified proposed algorithm is planned to provide the random initialization vector sets to the proposed Android security scheme presented in [12].

6. ACKNOWLEDGMENTS

Special thanks to Dr. Mohamed Zahra (Allah bless his soul) and Dr. M.I. Mohamed for their cooperation and support.

7. REFERENCES

[1] Martin Boesgaard, Mette Vesterager, Thomas Pedersen, Jesper Christiansen, and Ove Scavenius, 2002 “Rappit: A New High-Performance Stream Cipher”.Cryptico A/S FRUBEJERGVEJ, Denmark.

[2] Hadia M. El Hennawy, Alaa E. Omar , Salah M. Kholaif, “Design of LEA: Link Encryption Algorithm NEW PROPOSED STREAM CIPHER Algorithm”, 31st National Radio Science Conference (NRSC2014) April 28 – 30, 2014.

[3] Rainer A.Rueppel 1986.Analysis and Design of Stream Ciphers.

[4] B. Aissa, D. Nadir, M. AmmarAn ,“Approach Using Stream Cipher Algorithm for Image Encryption and Decryption”, 15th international conference on Sciences and Techniques of Automatic control & computer engineering - STA'2014

[5] Cryptanalysis of Achterbahn-128/80. María Naya-Plasencia? ,IST Programme under Contract IST-2002-507932 ECRYPT.

[6] H. Ahmadi, T. Eghlidos and S. Khazaei, “Improved guess and determine Attack on SOSEMANUK”, Tehran, Iran, 2006.

[7] M. Krause, “BDD-Based Cryptanalysis of Keystream Generators”, EUROCRYPT 2002, volume 2332.

[8] E. Beham and J. Seberry,“A Fast and Secure stream cipher using Rolling arrays”, April 29, 2005.

[9] R. Berlekamp (1968) Algebraic Coding Theory. Mc Grow-Hill. New- York

[10] T. Sigenthaler.,” Correlation-immunity of nonlinear combining functions for cryptographic applications”, IEEE Transactions on Information Theory, p 776-780. September 1984.

[11] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks,A. Heckert and J. Dray May 15,2001 ,“A statistical Test Suite for Random and Pseudorandom number Generators for Cryptographic Applications”, NIST Special Publication 800-22.

[12] H.Sarhan, A.A.Hafez, A.Safwat, and A.Hegazy, “Secure Android-Based Mobile Banking Scheme”, International Journal of Computer Applications (0975 – 8887) Volume 118–No.12, May, 2015.