Australian Journal of Basic and Applied Sciences, 7(6): 34-37, 2013
ISSN 1991-8178

Key Generator Based On Operation InSAFER FamilyOf Ciphers

[1]G. Khachatrian, [2]P. Abdollahifard

[1]Computer Engineering Department, Faculty of Engineering, American UniversityOf Armenia(AUA), Yerevan, Armenia
[2]Institute for Informatics and Automation Problems, National Academy of Sciences(NAS RA), Yerevan, Armenia.

Abstract: In this paper a mechanism for random number generation based on some operations used in SAFER Family of ciphers is introduced. It is shown how some kind of 16-bit shift register can be designed that has a nearly maximum possible period. That design is not based on traditional feedback primitive polynomials but is based on special XOR shift operation using nonlinear operational blocks used in SAFER Family. The presented mechanism can be used for generation's random 128-bit keys (or more) used in symmetric encryption algorithms by combining some of them such shift registers.

Key words: Random NumberGenerator, XOR shift register, Key generator, PRNG.

INTRODUCTION

Random numbers are useful for a variety of purposes, such as generating data encryption keys, simulating and modeling complex phenomena and for selecting random samples from larger data sets. They have also been usedaesthetically, for example in literature and music, and are of course ever popular for games and gambling. When discussing single numbers, a random number is one that is drawn from a set of possible values, each of which is equally probable, i.e., a uniform distribution. When discussing a sequence of random numbers, each number drawn must be statistically independent of the others.

Generation of random numbers plays a crucial role in cryptographic applications and many other related areas. One of important problems in cryptographic implementations is a very fast generation of possibly maximum number of different keys from some master key which are not correlated with each other. In this paper we will show how to design some kind of a 16-bit shift register which is not based on traditional feedback shift registers based on primitive polynomials, but is based on some nonlinear operations used in SAFER family of ciphers (James, L., Massey, 1994; Massey, J., *et al*., 1998; Massey, J., *et al*., 2000; Massey, J., 1995). Safer Family has two nonlinear byte to byte transformation tables which will be used in our design. One table denoted by *EXP*is based on exponentiation function $45^X \equiv Y \bmod 257$where *X* and *Y* are any numbers between 0 and 255. The second one denoted by LOG is based on logarithm function $\log_{45}(X) \equiv Y \bmod 257$ **.**

$$L(a) = \begin{cases} \log_{45}(a) \mod 257 & a \neq 0 \\ 128 & a = 0 \end{cases}$$

$$X(a) = (45^a \mod 257) \bmod 256$$

We used L and X function form safer family but we change those functions for 4 bit (as byte) to 4 bit, so our function will be:

$$L(a) = \begin{cases} \log_{q}(a) \mod 17 & a \neq 0 \\ 8 & a = 0 \end{cases}$$

$$X(a) = (q^a \mod 17) \bmod 16$$

For notation of 4 bits we will use numbers from 0 to 15 and q in this function is primitive number of number 17.

Design Of Random Number Generator:

The algorithm used by the generator is based on common cryptographic primitives, SAFER. We have investigated the following scheme of transformation of 16 bits as 4 blocks. The first and forth byte of an input combination are processed by using EXP function ($X(a)$) and second and third bytes are processed by using LOG function($L(a)$). After this transformation all 16 bits vector at the output (*Figure A)* are shifted 2 times, first 16 bits circular shift J1 bits (*Figure B)*

Corresponding Author: G. Khachatrian, Computer Engineering Department, Faculty of Engineering, American UniversityOf Armenia(AUA), Yerevan, Armenia

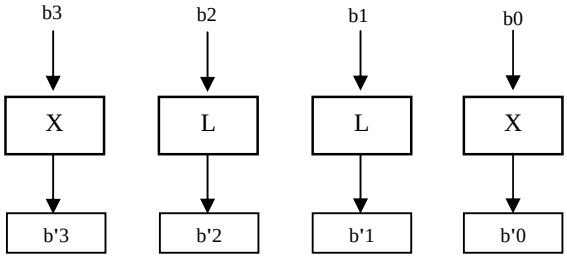

Fig. A

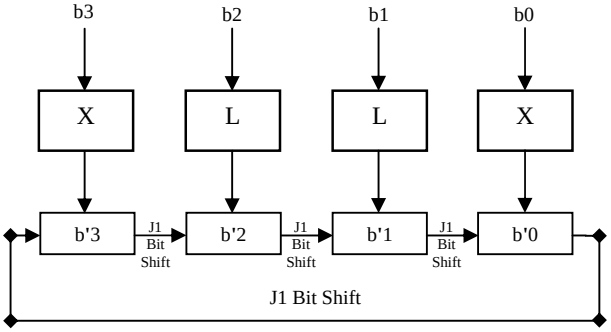

Fig. B

Second circular shift J2 (*Figure C)*, and then make *XOR* main 16 bits with result of J1 and J2 circular shift. So the schematic of transformation is depicted below.

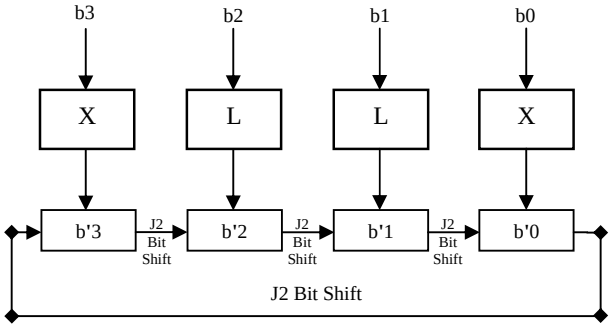

Fig. C

In Continue, make new B witch is

f(B)= B XOR (B>>j1) XOR (B>>j2)

(Notation: symbol (>>) meaning circular Shift)

After that each of the new bytes goes to the corresponding input and overall transformation is repeated again (*Figure D*).

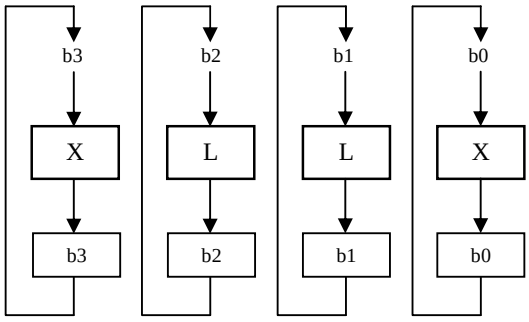

Fig. D

Obviously for each initial state there will be some number of different 16 bits generated after which the machine comes back to the initial state. All different 16 bits vectors generated after an initial state will be called to be a cycle.

Experiments on Our Generator:

We have written a program to generate all possible cycles with all primitive numbers of 17 and try on different J1 and J2 bits cycle shift, and find out that there isone major cycle that has "almost" all 16-bit combinations, if q=5, J1=6 and j2=13 namely 65533 out of 2^{16} = 65536.

So this model can generate more than %99.99 numbers in one loop. So there just 3 number aren't in our loop, these number are 16627, 33971 and 45364. Quite surprisingly an initial 16 bits state is 0,0,0,0.

We have also investigated the same schematic but with other primitive number of 17 and different bit shifts and the resulting are show in table 1 and figure f.

Table 1:

Q *(Primitive number 17)*	*J1* *Bits Shift*	*J2* *Bits Shift*	*Max Loop*
3	5	12	65290
5	6	13	65533
6	5	10	64812
7	3	8	65505
10	3	14	63879
11	2	13	65271
12	3	6	63246
14	5	6	65475

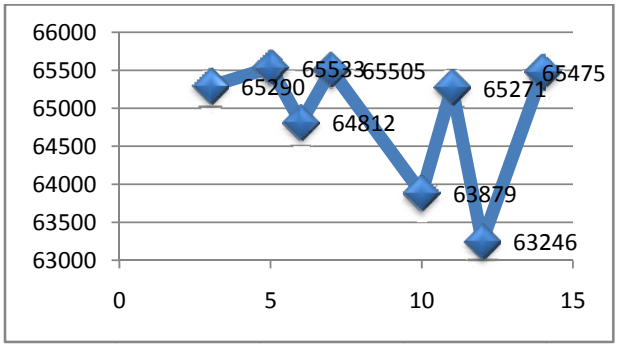

Fig. E

Conclusion:

It is interesting to see how we wereable to generate a shift register without using any primitive polynomials but getting a period of repetition close to maximum possible. It would be very interesting also to investigate some additional structural properties of combinations inside a largest cycle, in particular an Hamming weight distribution, etc. We can use above mentioned schematic to generate 128-bit keys, a typical setting for the key lengths, by combining 8 above described generators.

REFERENCES

AlfredMenezes, *et al*., 1997. Handbook of Applied Cryptography, CRC Press.

James, L., Massey, 1994. "SAFER K-64: A byte-oriented block-ciphering algorithm". In R. Anderson, editor, Fast Software Encryption, volume 809 of Lecture Notes in Computer Science, pages 1{17. Springer-Verlag.

Khachatrian, G., M. Kuregian, P. Abdollahifard, 2011." Random Number Generator Based on Operation In

Kohlbrenner, P. and K. Gaj, 2004. An embedded true randomnumber generator for fpgas.In FPGA '04: Proceeding of the2004 ACM/SIGDA 12th international symposium on Fieldprogrammable gate arrays, pp: 71-78.

Lim, D., 2004. Extracting Secret Keys from Integrated Circuits.Master's thesis, Massachusetts Institute of

Massey, J., 1995. "Announcement of a Strengthened Key Schedule for the Cipher SAFER", Revision.

Massey, J., G. Khachatrian, M. Kuregian, 1998. "Nomination of SAFER+ as a Candidate Algorithm for Advanced Encryption Standard (AES)"- Represented at the first AES conference, Ventura, USA,

Massey, J., G. Khachatrian, M. Kuregian, 2000. "Nomination of SAFER++ as a Candidate Algorithm for NESSIE project " - first NESSIE conference, , Leuven Belgium(37p)

SAFER Family of Ciphers",CSIT,ComputerScince and information technologies conference,yarvan, ARMENIA

Technology.

Wichmann, B.A. and I.D. Hill, 2006. Generating good pseudo-random numbers Computational Statistics and Data Analysis, 51: 1614-1622.