JOURNAL OF CRITICAL REVIEWS

ISSN- 2394-5125 VOL 7, ISSUE 15, 2020

# A Two Phase Block Cipher Based On Graph Structure With Linear Feedback Shift Register (LFSR) As Diffusion Layer

[1]Ch. Suneetha [2]MPR Murthy [3]CH. Neelima [4]S. Sarvalakshmi [5]G. Lalitha Devi

[1] Associate Professor in Mathematics, GITAM University, Visakhapatnam, India

[2] Sr. Assistant professor in Mathematics, GVP college forDegree and PG Courses(A), Visakhapatnam, India

[3]Assistant Professor in mathematics, Sankethika Vidya Parishad, Visakhapatnam, India

[4]Assistant Professor in mathematics, MR PG College, Vizianagaram, India

[5] Assistant Professor in Mathematics, Prism Degree and PG College, Visakhapatnam, India

[1]Author For Correspondence: schivuku@gitam.edu

Received: 14 April 2020 Revised and Accepted: 8 August 2020

**Abstract**: Extensive use of internet in technology and communications leads to hardship of transferring sensitive data between two or more entities. Disguising the sensitive information is the only tool to protect the data against cyber-attacks. Graph theory plays a major role in encryption for a long time. The present paper designs a block cipher using graph theory, especially the adjacency matrices of un directed graphs. The data is encrypted in blocks, each block in three rounds and each round at two different phases: using adjacency matrix at the first phase and applying logical XOR operation in a special pattern on every element of the message at the second phase. Here the adjacency matrix is public known to all, the entities generate new matrices for encrypting each block with the help of linear feedback shit register LFSR. Here LFSR acts as diffusion layer. The LFSR polynomial is secret key where the entities can communicate among themselves or they rely on trusted third party.

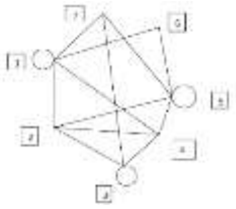
**Key Words**: Adjacency matrix, Linear Feedback Shift Register (LFSR), Logical XOR, Encryption, Decryption

**Introduction**: As the role of internet is ever expanding in communication networks nowadays there is always a threat of stealing, hacking and eavesdropping of the sensitive data. One of the means of protecting the information against active and passive attacks in transmission process is encrypting. Recently, graph theory is integrated in designing efficient cryptographic algorithms. Cayley graphs, Ramanujan graphs, expander graphs etc are widely used for encryption process due to the reason that finding the path is hard in these graphs for classical and quantum computers.

**Literature Survey**: Graph theory is playing an active role in cryptography nowadays. Several researchers of cryptography explored many graph theory concepts applied to different branches of Cryptography. Natalia Tokareva [1] explained the connection between graph theory and cryptography. There he discussed about sparse graphs, Cayley graphs and bent functions applied to mobile networks. Wael Mahmoud Al Etaiwi [2] designed a new complex cipher using cyclic graphs, complete graphs and minimum spanning tree. P. Amudha et.al. [3] explored several applications of graph theory in cryptography. In that chapter they proposed an encryption algorithm using Euler graphs. Yamuna et.al [4] designed a double encryption scheme using Hamilton path and complete graph. Steve Lu et.al. [5] applied graph theory to image cryptography by assigning nodes and edges to images. Charles et.al. [6] designed new collision resistant hash functions from expander graphs. Anmaria Costache et.al. [7] studied the security of a proposal for post quantum cryptography using super singular isogeny graphs. Later, Hyungrok Jo [8] introduced cryptographic hash functions based on Charles expander graph in post quantum cryptography. In view of the earlier work on cryptography using graph theory the present paper proposes an innovative technique of encryption using adjacency matrix and simple logical XOR operation.

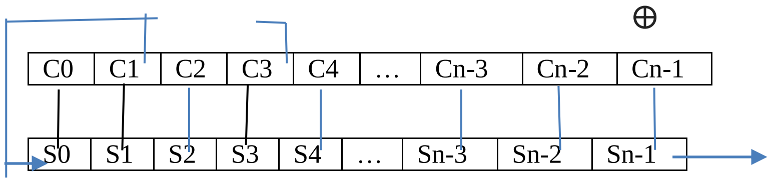
**Adjacency matrix of un directed graph**: An undirected graph G (V, E) with set of vertices V, set of edges E connecting the vertices is cyclic if the path from one vertex returns to the same and complete if an edge exists between two vertices [9,10]. A square matrix used to represent a finite graph is known as adjacency matrix with elements either 0 or 1 [11,12]. It is a symmetric matrix representing the connection between vertices, M = {$m_{ij}$ / $m_{ij}$ = 0 or 1} . General notation of adjacency matrix of un directed graph is

$[0\ m12 \ldots\ldots m1l\ m21\ 0\ m23 \ldots m2l \ldots\ldots\ldots\ldots\ldots ml1\ ml2 \ldots\ldots 0\ ]$ . Since it is symmetric having the entries either 0 or 1 the whole matrix can be represented by just the entries on the principal diagonal and the elements above the principal diagonal. Adjacency matrix is the representation of connectivity of pairs of vertices in finite graphs. For instance, the adjacency matrix of the below graph is



0 1 0 1 0 1 1 1 0 1 1 1 0 0 0 1 0 1 0 0 1 1 1 1 0 1 0 0 0 1 0 1 0 1 1 1 0 0 0 1 0 0 1 0 1 0 1 0 0

**Linear Feedback Shift Register (LFSR):** Linear Feedback Shift Register acts like Ouroboros in electronic systems which is an efficient design for output response analyser. Here the output is changed and fed back to the input to form an endless cycle in the form of sequence pattern. Initial or starting state of LFSR is called seed. Since it is periodic function the seed should be in nonzero state. The maximum length of the period of a p-bit LFSR is
$2^{p-1}$. There is one polynomial called LFSR polynomial which is a primitive polynomial. The seed or zero state is a sequence of bits with entries 0 and 1. Logical XOR operation is applied on the binary bits at the positions of the given LFSR polynomial, the output is fed to the input by shifting the bits linearly from left to right and replacing the first bit by the output of XOR operation [15,16]. The general form of LFSR polynomial is
$1+C_0x+c_1x^2+C_2x^3+\ldots\ldots C_nx^{n+1}$
$C_i \in \{0,1\}$ i = 0 to n-1. An n bit LFSR with primitive polynomial $1+x^2+x^4$ is in the form



Linear Feedback Shift Register (LFSR) has wide range of applications in electronics. It is used to generate sequences of parallel pseudo random numbers in cryptography. For light weight application stream cipher is more applicable for fast encryption of the message [13,14]. LFSR based stream cipher completes the task of speedy encryption with required cryptographic properties and with low computational overhead. This is the reason that LFSR stream cipher is popular at the initial stages. Later, it was found to be insecure as it is periodic, the sequence eventually repeats. Research on LFSR proved that the periodicity can be avoided by considering large LFSR and the security of the cipher can be enhanced by considering multiple LFSR's. In the present algorithm LFSR is used as diffusion layer for key stream generation of encryption of different blocks using the primary master key.

**Proposed Method**: Here an un directed graph and its adjacency matrix A are public. If two users want to communicate messages, they share a secret key in the form of an irreducible polynomial whose order is less than the order of the adjacency matrix called the primary or master key. The secret key sharing is done either by a mutual agreement or getting the key by a trusted third party. The message to be communicated is divided into blocks each of length equal to the square of the order of the adjacency matrix. If the adjacency matrix has order l, then the block size is $l^2$ . Since the adjacency matrix is symmetric the entire matrix can be represented by the elements on the principal diagonal and the elements above the principal diagonal. The adjacency matrix of order 5x5 is
$[a11\ a12\ a13\ a14\ a15\ a21\ a22\ a23\ a24\ a25\ a31\ a32\ a33\ a34\ a35\ a41\ a42\ a43\ a44\ a45\ a51\ a52\ a53\ a54\ a55\ ]$
the entire matrix ix represented by a sequence of binary stream

a11a12a13a14a15a22a23a24a25a33a34a35a44a45a55. For instance, the adjacency matrix
0 1 0 1 1 0 0 0 0 0 0 1 1 0 1 0 is represented by a binary stream 0101000010.

This key stream is master key stream. The key for three rounds of each block are generated by applying Linear Feedback Shift Register LFSR on the master key stream. The LFSR polynomial in the form of bits is secret between the communicating parties. The equivalent ASCII decimals of the message characters are divided into blocks of length $l^2$ each and arranged as entries of a lxl square matrix M1, M2, M3…. Each block matrix is encrypted at two different phases using adjacency matrix at the first phase and applying logical XOR operation on each element of the first stage encryption with the surrounding elements in a special pattern at the second phase. The two-phase encryption of each block is repeated three times with different keys; adjacency matrix and power. To add on more security for the cipher the first stage cipher is flipped before applying the XOR operation.

**Encryption**:

**Notation used in the algorithm**:

1. C[(p) (q) (s)] is the cipher for pth block, qth phase, sth round
2. A[(p) (s)] is adjacency matrix used for pth block sth round
3. r[(p) (s)] is a decimal number, the power to which the adjacency matrix is raised for pth block sth round

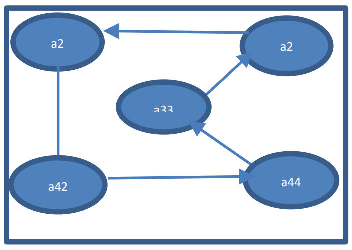
p = 1,2 3,4……. represents the number of blocks; q = 1,2 represents the number of phases of
encryption and s = 1,2,3 represents the number of rounds of encryption.

**I Phase Encryption**: At the first stage the binary stream of the adjacency matrix A is considered as starting state or seed of LFSR. Then using the irreducible LFSR polynomial the bits are shifted to the right replacing the first bit with new bit and leaving the last bit of the seed. The output binary stream of the first state is expanded as a new adjacency matrix A [(1)(1)] used to encrypt the first block matrix M1 at the first round. M1 is multiplied with A [(1)(1)] raised to the power r [(1)(1)], the decimal equivalent of the first four bits of the first state binary stream of LFSR

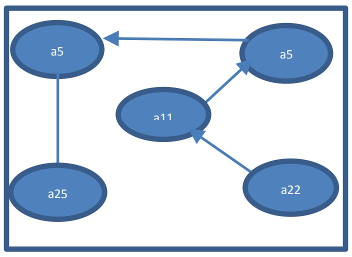
$$C[(1)(1)(1)] = M1 * A[(1)(1)]^{r[(1)(1)]}$$

The outcome matrix is flipped by interchanging the rows to have $C[(1)(1)F(1)]$

**II Phase encryption**: In this phase each decimal entry of the lxl matrix $C[(1)(1)F(1)]$ will undergo exclusive XOR operation with surrounding diagonal elements in a special pattern in counter clock wise direction and the original entry is replaced by new element. For instance, the element a33 is first XORed with a24, then with a22, with a42, finally with a44 and the resulting decimal number is placed in a33 position.



While applying the XOR operation on the end or boarder elements of the matrix the corresponding element of the last or first row/column are used. For instance, the XOR operation on the first element a11 is



After undergoing each element of the matrix $C[(1)(1)F(1)]$the XOR operation in the above pattern the outcome matrix is $C[(1)(2)(1)]$. The same procedure of the above two phases is repeated in two more rounds on $C[(1)(2)(1)]$ with adjacency matrices A [(1) (2)], A [(1) (3)] expanded by binary streams of second and third states of LFSR. Here r [(1) (2)], r [(1) (3)] are the exponents of adjacency matrices, the decimal equivalents of second and third states LFSR. The algorithm for one block encryption is

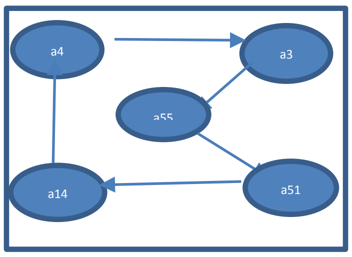
| | | |
|---|---|---|
| $C[(1)(1)(1)] = M1 * A[(1)(1)]^{r[(1)(1)]}$ | $C[(1)(1)(2)] = C[(1)(2)(1)] * A[(1)(2)]^{r[(1)(2)]}$ | $C[(1)(1)(3)] = C[(1)(2)(2)] * A[(1)(3)]^{r[(1)(3)]}$ |
| $C[(1)(1)F(1)] = FlipC[(1)(1)(1)]$ | $C[(1)(1)F(2)] = FlipC[(1)(1)(2)]$ | $C[(1)(1)F(3)] = FlipC[(1)(1)(3)]$ |
| $C[(1)(2)(1)] = XOR\ C[(1)(1)F(1)]$ | $C[(1)(2)(2)] = XOR\ C[(1)(1)F(2)]$ | $C[(1)(2)(3)] = XOR\ C[(1)(1)F(3)]$ |

All the data blocks are encrypted at two different phases in three rounds. The first, second, third states of LFSR are the adjacency matrices and exponents for first block. Fourth, fifth, sixth states of LFSR are the adjacency matrices and exponents for second block; seventh, eighth, ninth states for third block and so on. The LFSR is a periodic sequence, eventually coincides with the seed. For encrypting each block three states of LFSR are needed. So, at some point of the data there may be a coincidence between data block, adjacency matrix pair. That is an event of same data encrypted with same adjacency matrix may occur. Due to the repetition of data, adjacency matrix pair whenever it happens the cipher will become vulnerable against linear cryptanalysis and differential cryptanalysis leading to correlation attack. One way to avoid this is choosing the seed of the LFSR which is the binary stream of the adjacency as large as possible. Another way is after exhausting the LFSR, i.e., if the state of LFSR coincides the seed, the parity bit is applied on the block of original LFSR then the same procedure is repeated to generate key streams. After encrypting all the data blocks at two different phases in three rounds the output decimals are coded to ASCII text and communicated as cipher via public channel to the receiver.

**Decryption**:

The receiver after receiving the cipher divides it into blocks of length $l^2$, converts to decimal equivalents and writes as lxl square matrices. Then starts decryption from the first block

**I Phase Decryption**: Logical XOR operation is applied on each element of the first block matrix starting from the last element. Then the entries are flipped by interchanging the rows.



$$C[(1)(1)F(3)] = XOR\ C[(1)(2)(3)]$$

$$C[(1)(1)(3)] = Flip\ C[(1)(1)F(3)]$$

Then it is multiplied with the inverse of the adjacency matrix of the corresponding round raised to the power, the decimal equivalent of first four bits output binary stream of third state LFSR.

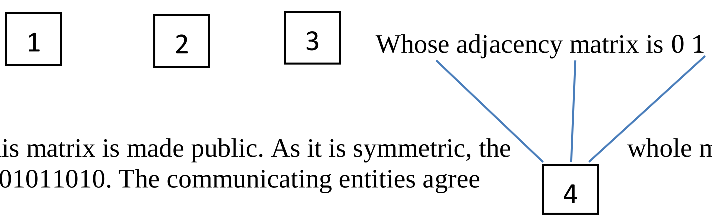
$$C[(1)(2)(2)] = C[(1)(1)(3)] * inv\{A[(1)(3)]\}^{r[(1)(3)]}$$

The two-phase decryption is implemented in two more rounds to obtain the first block matrix M1. The same process is applied to all the blocks and the corresponding ASCII characters written consecutively gives us the original message. The decryption algorithm for one block encryption is

| | | |
|---|---|---|
| $C[(1)(1)F(3)] = XOR\ C[(1)(2)(3)]$<br>$C[(1)(1)(3)] = Flip\ C[(1)(1)F(3)]$<br>$C[(1)(2)(2)] = C[(1)(1)(3)] * inv\{A[(1)(3)]\}^{r[(1)(3)]}$ | $C[(1)(1)F(2)] = XOR\ C[(1)(2)(2)]$<br>$C[(1)(1)(2)] = Flip\ C[(1)(1)F(2)]$<br>$C[(1)(2)(1)] = C[(1)(1)(2)] * invA\{[(1)(2)]\}^{r[(1)(2)]}$ | $C[(1)(1)F(1)] = XOR\ C[(1)(2)(1)]$<br>$C[(1)(1)(1)] = Flip\ C[(1)(1)F(1)]$<br>$M1 = C[(1)(1)(1)] * invA\{[(1)(1)]\}^{r[(1)(1)]}$ |

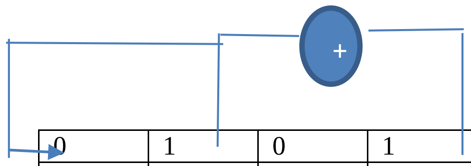
**Implementation of algorithm**: The algorithm is implemented on intel®, Core ™i3 with speed 2.10 GHz processor using MATLAB 2020 trial version.

**Example 1**: Consider the message ‘SUNEETHA’ consisting of 8 letters and an un directed graph



Whose adjacency matrix is 0 1 0 1 1 0 1 1 0 1 0 1 1 1 1 0

This matrix is made public. As it is symmetric, the whole matrix is represented by a binary stream 0101011010. The communicating entities agree upon to use an irreducible polynomial 1+ x^2 + x^4 either personally exchanging or getting it from a trusted third party. First three states of LFSR are



| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | Seed |
| 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 I state |
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 II state |
| 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 III state |

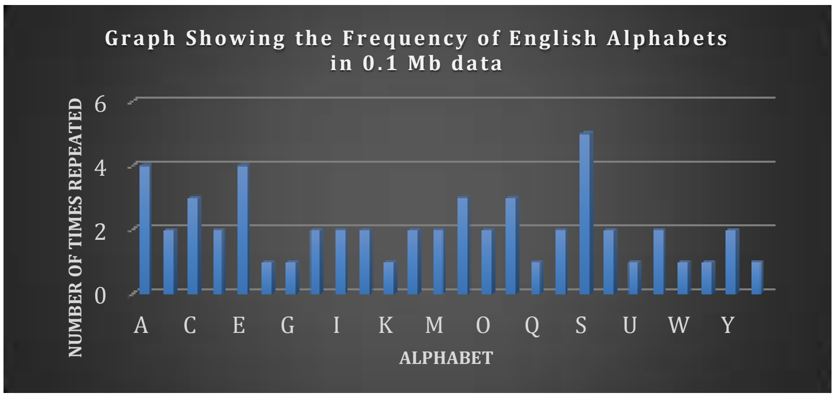
The adjacency matrices used in three rounds of the first block are formed by the binary streams

0010101101; 0001010110;1000101011. The powers to which the adjacency matrices raised are the equivalent decimals of first four bits of the above streams 2, 1,8. Since the message contains only 8 characters, block length is 16 others may be filled at random or consider the decimal number ‘32’. After three rounds of two phase encryption the cipher is ‘s ! É ³ Å DC 3 Y J’.

**Example 2**: In this example consider the repetition of the characters ‘SSSSSSSSSS’. After three rounds of encryption with different keys as adjacency matrices the cipher is

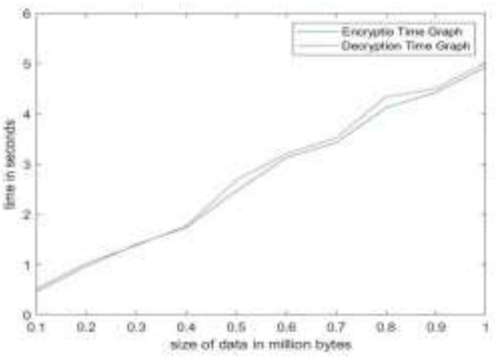
‘§ ý ³ N — l VT l { ¹’

**Security Analysis**: In the present paper an adjacency matrix of an un directed graph is made public. The message is encrypted in block at two different phases in three rounds using adjacency matrix and logical XOR operation. The adjacency matrix for each block and for each round are generated from the publicly known adjacency matrix using LFSR scheme. The irreducible LFSR polynomial is secret key between the users. The seed of LFSR is the original binary stream of the adjacency matrix. Here LFSR is used as diffusion layer to generate the adjacency matrices and exponents. Correlation attack is the general attack on LFSR. In LFSR the basic aim of an adversary is to recover the seed or the secret state bits. Here the seed of LFSR is public and it is only a device to generate adjacency matrices. So, no question of correlation attack arises here. Public key cryptographic algorithm is more Secure when it relies on some hard and well-defined mathematical problem. Here multiplication of each data matrix with adjacency matrix raised to some power and applying logical XOR operation on each decimal element of the matrix with all neighbouring elements in a special pattern is a hard-mathematical problem. Due to the concatenation of each element with its neighbouring elements by XOR operation it becomes a tough job for an adversary to break the cipher after three rounds of encryption. In first example 'SUNEETHA', E is repeated twice but no repetition is there in the cipher. In the second example 'SSSSSSSSSS', the same character 'S' is mapped to 's ! É ³ Å DC 3 Y $ J'. So, the linear cryptanalysis and differential cryptanalysis are not possible to execute in the present algorithm. The following graph shows English letter frequency in 0.1 Mb data, with 100 characters. The letter 'S' is repeated maximum of 5 times, next 'E' and 'A' are repeated 4 times. In 0.1 Mb data the English letter frequency if highly negligible.



**Performance Analysis**: For different size of the data the encryption and decryption time are calculated, tabulated and the plot is shown in figure 1.

| Data size (in Million bytes) | Time of Encryption (in seconds) | Time of Decryption (in seconds) |
|---|---|---|
| 0.1 | 0.463 | 0.512 |
| 0.2 | 0.972 | 1.023 |
| 0.3 | 1.399 | 1.382 |
| 0.4 | 1.743 | 1.768 |
| 0.5 | 2.465 | 2.679 |
| 0.6 | 3.131 | 3.201 |
| 0.7 | 3.427 | 3.523 |
| 0.8 | 4.123 | 4.342 |
| 0.9 | 4.431 | 4.503 |
| 1.0 | 4.935 | 5.021 |



The above graph shows that the execution time is almost linear and 1Mb data requires roughly 5 seconds for encryption/decryption. So, the execution time is nearly close to those of conventional block ciphers like AES and DES (Literature shows that encryption time for 1Mb data in AES is 5.66 and for DES is 4.1 approximately).

**Conclusions:** So, the present block cipher using adjacency matrix and logical XOR operation is more secure against all types of active and passive attacks that compete with the conventional block ciphers.

**References**:

1. Natalia Tokareva, "Connection between graph theory and cryptography G2C2:Graphs and groups, Cyclic and Coverings, Sep 2014, Novosibirsk Russia.
2. Wael Mahmoud Al Etaiwi, "Encryption algorithm using graph theory", Journal of scientific research and reports, 3(19) 2519-2527, 2014.
3. P. Amudha, A.C. Charles Sagayaraj, A.C. shantha Sheela, "An application of graph theory in cryptography" , International journal of pure and applied mathematics IJPAM, 119(3), 2018, pp 375-383.
4. Yamuna M, Meenal Gogia, Ashish sikka,Md.Jazib Hayat Khan, "Encryption using graph theory and linear algebra", International journal of computer applications IJCA, 2012.
5. S Lu, Rafael Ostrovsky, Daniel Manchala, "Visual cryptography on graphs", Cite Seeix COCOON, 2008, 225-234.
6. Denis X Charles, Eyal Z. Goren, Kristic E. Lauter, "Cryptographic hash functions from expander graphs", Journal of Cryptology, 22, 93-113, 2009.
7. Anmaria Costache, Brooke Feigon, Kristin Lauter, Maike Masierer, Anna Puskar, "Ramanujan graphs in cryptography", arXiv:1806. 05709V2 [math.NT]18th December 2018.
8. Hyungrok Jo, "Ramanujan graphs for post quantum cryptography", http://www.researchgate.net/publication/337150777.
9. Yvo Desmedt, Josef Piprzyk, Ron Sternfield, Xiaoming Sun and Tartary, "Graph Coloring Applied to Secure Computation in Non-Abilian Groups", Journal of Cryptology, Springer, online from 5th September 2011
10. C. Blundo, A. Santis, D.R. Stinson and U. Vaccaro "Graph decompositions and secret sharing schemes", Journal of Cryptology, 1995, Vol.8, No.1, pgs 39-64.
11. Connections between graph theory and cryptography http://en.wikipedia.org/wiki/Graph_theory
12. http://mathworld.wolfram.com/AdjacencyMatrix.html
13. Yi Lu and Serge Vaudenay. Cryptanalysis of Bluetooth keystream generator twolevel E0. In Advances in Cryptology - ASIACRYPT 2004, volume 3329 of Lecture Notes in Computer Science, pages 483–499. Springer-Verlag, 2004.
14. Kocher, P., Lee, R., McGraw, G., Raghunathan, A., Ravi, S.: Security as a New Dimension in Embedded System Design. In: Proc. of IEEE Design Automation Conference - DAC 2004, pp. 753–761. IEEE Computer Society Press, Los Alamitos (2004).
15. http://datagenetics.com/blog/november12017/index.html
16. http://homepages.cae.wisc.edu/~ece553/handouts/LFSR-notes.PDF