A FAST AND SECURE STREAM CIPHER BASED ON CELLULAR AUTOMATA OVER GF(q)

Miodrag Mihaljević
Academy of Science and Arts
Kneza Mihaila 35, Belgrade
Yugoslavia
emihalje@ubbg.etf.bg.ac.yu

Yuliang Zheng
Monash University
McMahons Road, Frankston
VIC 3199, Australia
yzheng@fcit.monash.edu.au

Hideki Imai
University of Tokyo
7-22-1 Roppongi, Minato-ku
Tokyo, 106-8558 Japan
imai@iis.u-tokyo.ac.jp

Abstract

The problem of designing a family of pseudorandom number generators for cryptographic applications, called key stream generators, is considered for word-oriented CPU platforms. A novel key stream generator, together with a new application of linear cellular automata over GF(q), is proposed. Construction of the generator is based on the use of very recently published results on cellular automata theory and its applications in cryptography, as well as on core principles employed in a number of existing key stream generators. Analysis indicates that the proposed generator satisfies standard minimal security requirements including a large period and good statistical properties, and that it is secure against all known attacks. An important feature of the proposed generator is that it is compact and suitable for high speed applications.

1 Introduction

Cryptographic techniques play an important role in information protection, and stream ciphers are an important class of encryption algorithms (see [28], [30] and [18]). A stream cipher encrypts one individual character in a plaintext message at a time, using an encryption transformation which varies with time. Such a cipher is typically implemented by the use of a so-called pseudorandom number generator or a key stream generator which expands a short secret key into a long running key sequence. Mathematically, a key stream generator is equivalent to a finite state machine that, based on a secret key, generates a key stream for controlling an encryption transformation. Let x_i, y_i, z_i, and s_i denote the plaintext digit, the ciphertext digit, key stream digit, and the internal state of the finite state machine at time i, and k denotes the secret key. Then the encryption procedure of the stream cipher can be described by the following: $y_i = x_i + z_i$, $z_i = f(k, s_i)$, $i \geq 1$, where $\{z_i\}$ is the key stream or running key sequence, $f(\cdot)$ is the next state function of the key stream generator, and "+" denotes a modulo addition.

According to [18], [30], and [28], for example, there is a vast body of theoretical knowledge on stream ciphers, and various design principles for stream ciphers have been proposed and extensively analyzed. However, there are relatively few fully-specified stream cipher algorithms in the open literature. This undesirable state of affairs can be partially explained by the fact that most stream ciphers used in practice tend to be proprietary and confidential [18]. By contrast, numerous concrete block cipher proposals have been published, some of which have been standardized or placed in public domain. Nevertheless, because of their significant advantages, stream ciphers are widely used today, and one can expect an increasing number of concrete proposals in the coming years, [18]. This paper represents a contribution to this line of research.

In this paper the main lines of a novel stream cipher are given which can be used as a tool for constructing particular stream ciphers appropriate to given conditions. Main aim of this paper is to propose a novel building block for stream cipher and to point out a possibility for combining reported design principles to obtain a new more secure and more efficient scheme.

The published proposals for key stream generators which can be used on theirs own to expand a short secret key into a long key stream or as a building blocks for more complex generators, include the following: nonlinear filter generator (see [31], [28], [30] and [18], for example), generators with time-variant tables including the alleged RC4 algorithm, (see [15], [9], [30]-[27], for example), shrinking and self-shrinking generators (see [6], and [17], for example) and cellular automata based key stream generators (see [32], [25],

for example). On the other hand, according to the reported results, it appears that all of these proposals also have certain weaknesses.

An aim of this paper is to propose a key stream generator which employs the good characteristics of published structures and overcome theirs weaknesses.

Note that, despite the weaknesses of certain proposed cryptographic applications of cellular automata (CAs), they appear to be a promising building block for cryptographic systems with certain advantages over linear feedback shift registers (LFSRs). CA is a more general linear finite state machine than a LFSR, and a LFSR can be considered as a particular CA. Also, CA is a means for fast generation of streams with good statistical characteristics and a large period. Finally, the CA can be considered as a more cryptographically secure generator than a LFSR, as a number of methods for LFSR initial state reconstruction based on certain LFSR output sequence can not work on the corresponding CA problem.

In this paper, the construction of a novel family of key stream generators is proposed and discussed.

Section 2 points out the relevant background. The novel key stream generator is proposed in Section 3. Its security together with efficiency is discussed in Section 4. Some concluding remarks are made in Section 5.

2 Background

This section summarize previous main works and results relevant for this paper. As the first, basic properties of the linear cellular automata over GF(q) are presented. Then, four classes of the key stream generators are pointed out, each of which employs certain design principle relevant for this work.

2.1 Linear Cellular Automata over GF(q)

A linear finite state machine (LFSM) is a realization or an implementation of certain linear operator. Linear feedback shift registers (LFSRs) and Linear Cellular Automata (CAs) are particular LFSMs. Following [3] this section summarize the main characteristics of the CA over GF(q).

A null-boundary linear hybrid cellular automata is a LFSM composed of a one-dimensional array of n cells with the following characteristics. Each cell consists of a single memory element capable of storing a member of GF(q), and a next-state computation function. We consider a situation when the communication between cells is nearest-neighbor, so that each cell is connected to only its left and right neighbors. The leftmost and rightmost cells behave as though their left and right neighbors, respectively, are in state 0, and this make the CA null-boundary. At each time step t, cell i has a state $s_i^{(t)}$ (that is a member of GF(q)). The next-state function of a cell is its computation rule, or just rule. A linear CA employs the linear next-state functions.

For time step $t+1$, each cell i computes its new state $s_i^{(t+1)}$, using its next-state function f_i. In a CA, this function can depend on only the information available to the cell, and in the here considered case, it is the states of cells $i-1$, i, and $i+1$ at the time t. Since we require that f_i be linear, $s_i^{(t+1)} = f_i(s_{i-1}^{(t)}, s_i^{(t)}, s_{i+1}^{(t)}) = c_i s_{i-1}^{(t)} + d_i s_i^{(t)} + b_i s_{i+1}^{(t)}$, and b_i, d_i, and c_i are constants dependent on the particular machine. The multiplication and addition operations are performed in the field GF(q).

We define the state of a CA at time t to be the n-tuple formed from the states of the individual cells, $s^{(t)} = [s_1^{(t)}, ..., s_n^{(t)}]$. The next-state function of the CA is computed as $s^{(t+1)} = [f_1(0, s_1^{(t)}, s_2^{(t)}), ..., f_i(s_{i-1}^{(t)}, s_i^{(t)}, s_{i+1}^{(t)}), ...]$. Since each f_i is a linear function, f is also a linear function, mapping n-tuples to n-tuples. Linearity implies that f has an n by n matrix formulation A, so that the previous expression can be rewritten as a matrix-vector product

$$s^{(t+1)} = f(s^{(t)}) = A s^{(t)} \quad , \tag{1}$$

where A is the transition matrix for the CA, and the product is a matrix-vector multiplication over GF(q).

Because the CA communication is restricted to nearest-neighbor, the matrix A is tridiagonal. The sub-diagonal contains the multipliers on the left inputs of the cells; likewise, the super-diagonal contains the right-input multipliers. The main diagonal consists of the self-input multipliers, and the rest of the matrix is 0:

$$A = \begin{bmatrix} d_1 & b_1 & 0 & ... & 0 & 0 \\ c_2 & d_2 & b_2 & ... & & 0 \\ 0 & c_3 & d_3 & ... & & . \\ & . & . & ... & & . \\ 0 & & & ... & d_{n-1} & b_{n-1} \\ 0 & 0 & ... & ... & c_n & d_n \end{bmatrix} \tag{2}$$

A CA has a maximum length cycle if the sequence of states $s^{(0)}$, $s^{(1)}$, $s^{(2)}$, ... , $s^{(0)}$ includes all $q^n - 1$ nonzero states for any nonzero starting state $s^{(0)}$, and its characteristic polynomial is primitive if and only if the CA has a maximal length cycle. In [3], the underlying theoretical results which are required for the design and analysis of linear hybrid CA over GF(q) are derived, and a probabilistic algorithm is proposed for obtaining a CA with a given characteristic polynomial. The algorithm provides a good

practical method to the finding of any required maximal length CA.

2.2 Certain Key Stream Generators

This subsection summarizes relevant results on constructions of the key stream generators from which the novel proposal originates.

2.2.1 Cellular Automata Based Generators

The first cryptographic application of a cellular automata was published in [32]. Two key stream generators based on the linear cellular automata over GF(2), called PCA with ROM and Two Stage PCA were proposed in [25] (also see [5]). The weaknesses oh these generators have been reported in [16], [13], [22], [23],[24], and [2]. Recently, an improved key stream generator based on PCA with ROM was proposed and analyzed in [24] assuming operations over GF(2). Note that no one cryptographic application of CA over GF(q), $q > 2$, has been reported yet.

2.3 Nonlinear Filter Generator

The nonlinear filter generator (NLFG) is a well known type of key stream generators. The NLFG consists of a single regularly clocked binary linear feedback shift register (LFSR) and a nonlinear Boolean function f of n input variables. The key stream is generated by applying f to the output of n stages of the LFSR. The weaknesses of NFLG have been reported in [31], [8], [1], [11], [14], and [29], for example.

2.3.1 Generators with Time-Variant Tables

A well known method for combining certain pseudorandom sub-generators to obtain a key stream generator is the shuffler [15]. One pseudorandom generator is used to produce the values for the final key stream sequence, but the values are first saved in a table. The second generator is used to produce pointers into the table. At each cycle, the pointer generator produces a new pointer into table, and the value at that location is output. Then the value generator produces a new value, which is inserted into the table, replacing the value that was just removed. The random delaying of the values in the table has the effect of shuffling the sequence elements. A variant of this approach is reported in [9].

Another type of time-variant table is employed in the alleged RC4 key stream generator [27], [30]. According to [30] the internal state of RC4 at time t consists of a table $S_t = (S_t(\ell))_{\ell=1}^{2^n-1}$ of 2^n n-bit words and two pointer n-bit words i_t and j_t. Let initially $i_0 = j_0 = 0$. The next-state and output functions of RC4 are for every $t \geq 1$ defined by $i_t = i_{t-1} + 1$, $j_t = j_{t-1} + S_{t-1}(i_t)$, $S_t(i_t) = S_{t-1}(j_t)$, $S_t(j_t) = S_{t-1}(i_t)$, $Z_t = S_t(S_t(i_t) + S_t(j_t))$, where all the additions are modulo 2^n. It is assumed that all the words except for the swapped ones remain the same (swapping itself is effective only if $i_t \neq j_t$). The output n-bit word sequence is $Z = (Z_t)_{t=1}^{\infty}$. Note that the time-variant table in alleged RC4 is a slowly-varying one.

Certain weaknesses of the generators based on time-variant tables have been reported in [26], [20], and [12].

2.3.2 The Shrinking Generators

Construction of a key stream generator, called the shrinking generator is proposed in [6] (noting that the same idea in a cryptanalytic context is considered in [19]). The construction uses two sources of pseudorandom bits to create a third source of pseudorandom bits of (potentially) better quality than the original sources. Here quality stands for difficulty of predicting the pseudorandom sequence. The resulting sequence is a subsequence from the first source where the subsequence elements are chosen according to the positions of "1" in the the second source. Therefore the resultant sequence is a "shrunken" version of the first one.

A key stream generator based on the shrinking principle and called self-shrinking generator was proposed and considered in [17]. The self-shrinking generator employs only one linear feedback shift register (LFSR) and the generator output is produced from the LFSR output sequence according to the following: If a pair happens to take value "10" or "11", this pair is taken to produce the pseudorandom bit "0" or "1", depending on the second bit of the pair. On the other hand, if pair happens to be "01" or "00", it will be discarded.

Certain weaknesses of the shrinking generators have been reported in [19], [6], [17], [21], and [10].

3 Novel Key Stream Generator

3.1 Underlying Design Criteria

Intention of any construction of a key stream generator is to obtain an efficient and secure scheme. Key stream generators are required to be practically secure with the respect to computationally bounded cryptanalytic attacks in the known/ciphertext scenario. Accordingly, the practical security criterion for a key stream generator is the key stream unpredictability criterion which means that without knowing the secret key it should be computationally infeasible to reconstruct a key stream sequence from its portions.

In practice, the key stream generator security is checked only with respect to particular cryptanalytic attacks, and the required immunity to these attacks gives rise to various practical design criteria. In general, insisting on satisfying or optimizing certain particularly chosen design criteria does not appear to be a good strategy, as the key stream generator may then become vulnerable to other cryptanalytic attacks.

Cryptanalytic attacks can be classified into three general types. The attacks of the first type use statistical weaknesses of the key stream sequence for the prediction, and the resulting design criteria is requirement for good statistical properties of the the key stream sequence. The attacks of the second type aim at reconstructing the key stream sequence by using an equivalent key stream generator of a simple structure and typically much larger internal state size whose parameters have to be defined from known portions of the key stream sequence. The corresponding design criteria include long period of the key stream sequence and the high complexity measures of various kinds. The attacks of the third type aim at reconstructing the secret key and they are the most dangerous. Accordingly, the corresponding design criteria include the resistance on all known approaches for secret key reconstruction.

3.2 Main Ideas for Construction

The novel generator is designed based on the following principles:

- finite state machine principle employing linear CA over GF(q),
- nonlinear filter principle with time variant mapping - filter function,
- a variant of the self shrinking principle.

Also, the underlying idea for the novel construction could be considered in the following way: Generate the key stream starting from two appropriate sources of pseudorandom patterns defined by the following. The sequences of states of both sources should have good statistical properties, and each source should control the another one in certain manner such that the key stream has better cryptographic quality than the sequences of patterns generated by the sources, assuming that quality stands for difficulty of predicting the key stream.

3.3 The Generator Algorithm

The novel generator is a finite state machine which operates according certain clock and generates a sequence with elements from GF(q).

The main components of the generator are the following:

1. linear CA over GF(q), q prime, with L cells, and primitive characteristic polynomial;
2. RAM with q cells for a permutation of all elements from GF(q);
3. control logic.

The secret key determines the CA initial state and the RAM initial state. Also, we assume that for a particular application, an appropriate selection of CA state-transition matrix could be done based on [3].

In order to minimize the coast of generator realization we restrict the construction on employment the CA transition matrix (2) assuming the following constraint set.

Constraint 1: (i) $b_i = 1$, $1 \leq i \leq n-1$, (ii) $c_i = -1$, $2 \leq i \leq n$, (iii) $d_i \in \{0,1\}$, $1 \leq i \leq n$, and (iv) the number of d_i, $1 \leq i \leq n$ that are 1 is minimal.

The field size q is restricted to be prime, since any CA that has first three properties over non-prime field is reducible, [3].

We assume the following notation:

- CA_i is content of the ith CA cell which is an element of GF(q), $i = 1, 2, ..., L$;
- $RAM(a_i)$ is a content of the RAM cell at address a_i, $i = 0, 1, ..., q-1$;
- $SWAPP(RAM(a_i), RAM(a_j))$ denotes operation of exchanging the contents of RAM locations at address a_i and a_j.

After each clock, the generator realizes the following steps, and generates an output symbol.

The Generator Clock Cycle

1. transition from the current CA state into the next one;
2. redefining of the RAM according to the following, for each $i = 1, 2, ..., L/2$:

$$SWAPP(RAM(CA_i), RAM(CA_{i+L/2})) \quad (3)$$

(assuming that L is an even integer);

3. calculation of a value S:

$$S = \sum_{i=1}^{I} RAM(CA_{i\Delta}) \ , \quad (4)$$

where $\sum$ denotes modulo q addition and I, Δ are certain constants, $I\Delta \leq L$;

- repeat the Step 1 if S is greater than certain threshold $\alpha = S_{max}/2$, where S_{max} is the biggest element of GF(q);

4. calculation of a value ADS:

$$ADS = \sum_{\ell=1}^{L} CA_\ell \ , \qquad (5)$$

where $\sum$ denotes modulo q addition;

- the generator output at the end of current cycle is the RAM content at the address ADS.

4 Discussion of the Generator Characteristics

This section points out main characteristics of the proposed key stream generator. The analysis implies that the generator is a cryptographically secure one, and that it can be efficiently realized.

4.1 Cryptographic Security

4.1.1 Period and Statistical Characteristics

Basic requirements on a key stream generator include large period, high complexity, and good statistical properties of the key stream sequence. Deriving the period and complexity of a pseudorandom sequence is generally a difficult algebraic problem which seems to be tractable only for relatively simple sequences and under special constraints. Due to the unpredictability criterion, key stream sequence should not have simple structure and, accordingly, its basic characteristics of period, complexity and statistical properties are unlikely easy to be established in practical schemes. The generator proposed in previous section does not belong to a class of simple schemes, so that, according to the results known so far, it seems unlikely that certain characteristics of the output sequences can be derived in a deterministic manner, and only probabilistic results could be expected. These probabilistic results should be based on relevant underlying assumptions formulated according to the well established results regarding to the random mappings as well as the characteristics of sequences generated by the cellular automata.

Period. Following [7] and [12], it can be shown that the state diagram of the proposed generator consists of cycles only, which can be expected to have average length approximately equal to $2^{qlog_2q+Llog_2q-1}$.

Statistical Characteristics. Assumption that CA with primitive characteristic polynomial generate sequences with good statistical properties (which is a reasonable one, see [5], for example), and following the appropriate statistical model imply that the key stream sequences generated by the novel scheme have good statistical properties over GF(q).

4.1.2 Resistance on Known Attacks

A main criterion for the security evaluation is resistance against the known cryptanalytic attacks. Accordingly note that it can be directly shown that the proposed generator is resistant against all the cryptanalytic approaches reported so far, and particularly it is resistant against:

- all the attacks on cellular automata based structures (see [16], [13], [22], [23],[24], [2], for example),
- all the attacks on nonlinear filter generators (see [31], [8], [1], [11], [14], [29], for example),
- all the attacks on the generators based on time-variant tables (see [26], [20], [12], [30], [18], for example), and
- all the attacks on the shrinking based generators (see [19], [6], [17], [21], [10], [18], for example).

Accordingly, the novel generator is resistant against all cryptanalytic attacks published so far, and its effective key size is equal to its formal size.

4.2 Complexity of Realization

Recall that the generator construction is restricted on linear CA with transition matrix (2) with the Constraint 1. Counting the operations required for realization of the each generator clock cycle yields the following upper bound on complexity C of generating a key stream symbol:
$C \leq 8L(modq\ add.) + L(read/write\ op.)$.
Accordingly, the complexity of a bit generation is upper bounded by C/log_2q.

Note that the generator structure can be efficiently implemented in both: software and/or hardware, noting that the VLSI CA chips are available (see [25]).

5 Conclusions

In this paper, the construction for a novel family of key stream generators is proposed and discussed.

The proposal is based on certain recently introduced approaches which enable design of secure and efficient key stream generators. These approaches include employment of the linear cellular automata over GF(q), time variant nonlinear mapping / filtering, and the shrinking principle.

It is pointed out that the novel construction satisfies the standard - basic requirements for the cryptographic security and it ensures fast generation of the key stream. The proposed scheme generates key stream sequences of period exponential with the main

generator parameters and with good statistical characteristics. The novel generator is resistant against all up-to now known attacks, and its effective key-size length is equal to its formal length. Also, the proposed scheme can be used as a building block for more complex systems. Finally, note that the proposed generator is a regular and compact structure suitable for high speed applications.

References

[1] R.J. Anderson, "Searching for optimum correlation attack", Fast Software Encryption '94, *Lecture Notes in Computer Science*, vol. 1008, pp. 137-143, 1995.

[2] S.R. Blackburn, S. Murphy and K.G. Peterson, "Comments on "Theory and Applications of Cellular Automata in Cryptography"", *IEEE Trans. Comput.*, vol. 46, pp. 637-638, May 1997.

[3] K. Cattell and J.C. Muzio, "Analysis of one-dimensional linear hybrid cellular automata over GF(q)", *IEEE Trans. Comput.*, vol. 45, pp. 782-792, 1996.

[4] K. Cattell and J.C. Muzio, "Synthesis of one-dimensional linear hybrid cellular automata", *IEEE Trans. Computer-Aided Design*, vol. 15, pp. 325-335, March 1996.

[5] P.P. Chaudhuri, D.R. Chaudhuri, S. Nandi and S. Chattopadhyay, *Additive Cellular Automata: Theory and Applications*. New York: IEEE Press, 1997.

[6] D. Coppersmith, H. Krawczyk and Y. Mansour, "The shrinking generator", Advances in Cryptology - CRYPTO '93, *Lecture Notes in Computer Science*, vol. 773, pp. 22-39, 1994.

[7] P. Flajolet and A.M. Odlyzko, "Random mapping statistics", Advances in Cryptology - EUROCRYPT '89, *Lecture Notes in Computer Science*, vol. 434, pp. 329-354, 1990.

[8] R. Forre, "A fast correlation attack on nonlinearly filtered shift-register sequences", Advances in Cryptology EUROCRYPT '89, *Lecture Notes in Computer Science*, vol. 434, pp. 586-595, 1990.

[9] J. Golić and M. Mihaljević, "Minimal linear equivalent analysis of a variable memory binary sequences generator", *IEEE Trans. Inform. Theory*, vol. 36, pp. 190-192, 1990.

[10] J. Golić and M. Mihaljević, "A generalized correlation attack on a class of stream ciphers based on the Levenshtein distance", *Journal of Cryptology*, vol. 3, pp. 201-212, 1991.

[11] J. Golić, "On the security of nonlinear filter generators", Fast Software Encryption '96, *Lecture Notes in Computer Science*, vol. 1039, pp. 173-188, 1996.

[12] J. Golić, "Linear statistical weakness of alleged RC4 key stream generator", Advances in Cryptology - EUROCRYPT '97, *Lecture Notes in Computer Science*, vol. 1233, pp. 226-238, 1997.

[13] C.K. Koc and A.M. Apohan, "Inversion of cellular automata iterations", *IEE Proc. - Comput. Digit. Tech.*, vol. 144, pp. 279-284, 1997.

[14] S. Lee, S. Chee, S. Park, and S. Park, "Conditional correlation attack on nonlinear filter generators", Advances in Cryptology - ASIACRYPT '96, *Lecture Notes in Computer Science*, vol. 1163, pp. 360-367, 1996.

[15] M.D. MacLaren and G. Marsaglia, "Uniform random number generators", *Journal ACM*, vol. 12, pp. 83-89, 1968.

[16] W. Meier and O. Staffelbach, "Analysis of pseudo random sequences generated by cellular automata", Advances in Cryptology - EUROCRYPT 91, *Lecture Notes in Computer Science*, vol. 547, pp. 186-189, 1992.

[17] W. Meier and O. Staffelbach, "The self-shrinking generator", Advances in Cryptology - EUROCRYPT '94, *Lecture Notes in Computer Science*, vol. 950, pp. 205-214, 1995.

[18] A.J. Menezes, P.C. van Oorschot and S.A. Vanstone, *Handbook of Applied Cryptography*. Boca Roton: CRC Press, 1997.

[19] M. Mihaljević, "An approach to the initial state reconstruction of a clock-controlled shift register based on a novel distance measure", Advances in Cryptology - AUSCRYPT '92, *Lecture Notes in Computer Science*, vol. 718, pp. 349-356, 1993.

[20] M. Mihaljević, "A correlation attack on the binary sequence generators with time-varying output function", Advances in Cryptology - ASIACRYPT '94, *Lecture Notes in Computer Science*, vol. 917, pp. 67-79, 1995.

[21] M. Mihaljević, "A faster cryptanalysis of the self-shrinking generator", Information Security and Privacy - ACISP '96, *Lecture Notes in Computer Science*, vol. 1072, pp. 182-189, 1996.

[22] M. Mihaljević, "Security examination of certain cellular automata based key stream generator", *ISITA 96 - 1996 IEEE Int. Symp. Inform. Theory and Appl.*, Canada, Victoria, B.C., Sept. 1996, Proc. pp. 246-249.

[23] M. Mihaljević, "Security examination of a cellular automata based pseudorandom bit generator using an algebraic replica approach", Applied Algebra, Algorithms and Error Correcting Codes - AAECC 12, *Lecture Notes in Computer Science*, vol. 1255, pp. 250-262, 1997.

[24] M. Mihaljević, "An improved key stream generator based on the programmable cellular automata", Information and Communication Security - ICICS '97, *Lecture Notes in Computer Science*, vol. 1334, pp. 181-191, 1997.

[25] S. Nandi, B.K. Kar and P. Pal Chaudhuri, "Theory and applications of cellular automata in cryptography", *IEEE Trans. Comput.*, vol. 43, pp.1346-1357, 1994.

[26] C.T. Retter, "A key-search attack on MacLaren-Marsaglia system", *Cryptologia*, vol. 9, pp. 114-130, 1985.

[27] R.L. Rivest, "The RC4 encryption algorithm", RSA Data Security, Inc., March 1992.

[28] R.A. Rueppel, "Stream ciphers", in *Contemporary Cryptology: The Science of Information Integrity*, G. Simmons ed., pp. 65-134. New-York: IEEE Press, 1991.

[29] M. Salmasizadeh, L. Simpson, J. Golić, and E. Dawson, "Fast correlation attacks and multiple linear approximations", Information Security and Privacy - ACISP '97, *Lecture Notes in Computer Science*, vol. 1270, pp.228-239, 1997.

[30] B. Schneier, *Applied Cryptography: Protocols, Algorithms, and Source Codes in C*. New-York: John Wiley, 1996, 2nd edn.

[31] T. Siegenthaler, "Cryptanalyst's representation of nonlinearly filtered ml-sequences", Advances in Cryptology - EUROCRYPT '85, *Lecture Notes in Computer Science*, vol. 219, pp.103-110. 1986.

[32] S. Wolfram, "Cryptography with Cellular Automata", Advances in cryptology - CRYPTO 85, *Lecture Notes in Computer Science*, vol. 218, pp. 429-432, 1985.