Deterministic Cryptanalysis of some Stream Ciphers [⋆]

P. Caballero-Gil[1], A. Fúster-Sabater[2] and C. Hernández-Goya[1]

[1] Faculty of Maths, D.E.I.O.C., University of La Laguna, 38271 Tenerife, Spain

[2] Institute of Applied Physics, C.S.I.C., Serrano 144, 28006 Madrid, Spain

Abstract. A new graph-based approach to edit distance cryptanalysis of some clock-controlled generators is here presented in order to simplify search trees of the original attacks. In particular, the proposed improvement is based on cut sets defined on some graphs so that only the most promising branches of the search tree have to be analyzed because certain shortest paths provide the edit distances. The strongest aspects of the proposal are: a) the obtained results from the attack are absolutely deterministic and b) many inconsistent initial states are recognized beforehand and avoided during search.

1 Introduction

The main goal in the design of stream ciphers is to generate long pseudorandom keystream sequences from a short key in such a way that it is not possible to rebuild the short key from the keystream sequence. This work focuses on stream ciphers based on Linear Feedback Shift Registers ($LFSRs$), and more precisely on Shrinking [2] and Alternating Step [10] generators. Both generators produce keystream sequences with good cryptographic properties [8].

Most types of cryptanalysis on stream ciphers are performed under a known plaintext hypothesis, that is to say, it is assumed that the attacker has direct access to the keystream output from the generator [11]. The computational complexity of such attacks is always compared with the complexity of the exhaustive search, and if the former is smaller, then the cipher is said to be broken. Although this theoretical definition can look useless, in fact it is very important for the development and understanding of the security of stream ciphers because many times it reveals weaknesses that might lead to practical attacks.

The main idea behind this paper is to propose a deterministic improvement of a known plaintext divide-and-conquer attack consisting of three steps: 1) Guess the initial state of an $LFSR$ component of the generator. 2) Try to determine the other variables of the cipher based on the intercepted keystream. 3) Check that guess was consistent with observed keystream sequence.

[⋆] This work was supported in part by CDTI (Spain) and the companies INDRA, Unión Fenosa, Tecnobit, Visual Tool, Brainstorm, SAC and Technosafe under Project Cenit-HESPERIA as well as supported by Ministry of Science and Innovation and European FEDER Fund under Project TIN2008-02236/TSI.

Caballero-Gil P., Fúster-Sabater A. and Hernández-Goya C. (2009).
Deterministic Cryptanalysis of some Stream Ciphers.
In *Proceedings of the 7th International Workshop on Security in Information Systems*, pages 16-25
DOI: 10.5220/0002175500160025

This three-step attack was first proposed in [6] and [7] by means of a distance function known as Levenshtein or edit distance. Nevertheless, the approach considered in this work may be seen as an extension of the constrained edit distance attack to clock-controlled $LFSR$-based generators presented in [12] and generalized in [1]. Our main aim here is to investigate whether the number of initial states to be analyzed can be reduced. This feature was pointed out in [4] as one of the most interesting problems in the cryptanalysis of stream ciphers. According to the original method, the attacker needs to traverse an entire search tree including all the possible $LFSR$ initial states. However, in this work the original attack is improved by simplifying the search tree in such a way that only the most efficient branches are retained. In order to achieve such a goal, cut sets are defined in certain graphs that are here used to model the original attack. This new approach produces a significant improvement in the computing time of the original edit distance attack since it implies a dramatic reduction in the number of initial states that need to be evaluated. Furthermore, it is remarkable that, unlike previous attacks, the results obtained from this proposal are fully deterministic.

2 Preliminaries

The Shrinking Generator (SG) [2] and the Alternating Step Generator (ASG) [10] are two well known keystream generators with cryptographic applications. The notation used within this work is as follows. The lengths of the $LFSRs$ S, A and B are denoted respectively by L_S, L_A and L_B. Their characteristic polynomials are respectively $P_S(x)$, $P_A(x)$ and $P_B(x)$, and the sequences they produce are denoted by $\{s_i\}$, $\{a_i\}$ and $\{b_i\}$. The output keystream is $\{z_j\}$.

Despite their simplicity and the large number of published attacks [3], [5], [9] and [13], both generators remain remarkably resistant to practical cryptanalysis and the previous references are just theoretical attacks.

The edit or Levenshtein distance is the minimum number of elementary operations (insertions, deletions and substitutions) required to transform one sequence X of length N into another sequence Y of length M, where $M \leq N$. Some applications of the edit distance are file checking, spell correction, plagiarism detection, molecular biology and speech recognition. The dynamic programming approach (like the shortest-distance graph search and Viterbi algorithm) is a classical solution for computing the edit distance matrix where the distances between prefixes of the sequences are successively evaluated until the final result is achieved. When applying an edit distance attack on a clock-controlled stream cipher, the objective is to compute the initial state of a target $LFSR$ that is a component of the attacked generator. As in Viterbi search, this problem has the property that the shortest path to a state is always part of any solution of which such a state is a part. We will be able to see this fact quite clearly by the formalization of the algorithm as a search through a graph.

Clock-controlled registers are said to work with constrained clocking when a restriction exists on a maximum number of times that the register may be clocked before an output bit is produced. For these registers, attacks based on the so-called constrained edit distance have been proposed and analyzed in [6] and [7].

In this work, two different possible models for the attacked generator are considered. In both cases it is assumed that the feedback polynomial of the target $LFSR$ is known. According to the first model, it is assumed that $Y = \{y_n\}$ is an intercepted keystream segment of length M, which is seen as a noisy decimated version of a segment $X = \{x_n\}$ of length N produced by a target $LFSR$. On the other hand, according to the second model, it is assumed that $X = \{x_n\}$ is an intercepted keystream segment of length N, which is seen as a noisy widened version of a segment $Y = \{y_n\}$ of length M produced by a target $LFSR$. In this latter case, insertions in the sequence Y are indicated by two sequences S and B so that S points the locations where the bits of B must be inserted.

The main objective of the attack according to these models will be to deduce some initial state of the target $LFSR$ that allows one to produce an intercepted keystream sequence through decimation or insertion, respectively, without knowing the decimation or insertion sequences.

An essential step in edit distance attacks is the computation of edit distance matrices $W = (w_{i,j}), i = 0, 1, \ldots, N - M, j = 1, 2, \ldots, M$ associated each one with a couple of sequences X and Y where Y is the intercepted keystream sequence and X is a $LFSR$ sequence produced by one possible initial state.

In the first model, the intercepted sequence is Y while X is the candidate sequence. In the second model, the intercepted sequence is X while Y is the candidate sequence. Also note that from the computation of the edit distance between X and Y, the edit sequences that are computed in the first case correspond to decimation sequences while in the second case they correspond to insertion sequences.

Some of the parameters of such a matrix are described below. Firstly, its dimension is $(N - M + 1) \cdot M$. Furthermore, its last column gives the edit distance between X and Y thanks to the value $\min_{i=0,\ldots,N-M}\{w_{i,M} + N - M - i\}$. Lastly, each element of the matrix, apart from the last column $w_{i,j},\ i = 0, \ldots, N - M,\ j = 1, \ldots, M - 1$, corresponds exactly to the edit distance between prefix sub-sequences $x_1, x_2, \ldots, x_{i+j}$ and $y_1, y_2, \ldots, y_j$. The edit distance between prefix sub-sequences $x_1, x_2, \ldots, x_{i+M}$ and Y are given by $w_{i,M} + N - M - i,\ i = 0, \ldots, N - M$.

In the edit distance attack here analyzed only deletions and substitutions are allowed. Consequently, each element $w_{i,j}$ of the edit distance matrix W may be recursively computed from the elements of the previous columns according to the formulas in Equation (1), which depend exclusively on the coincidence or difference between the two bits x_{i+j} and y_j.

$$\begin{gathered}
w_{i,1} = P_i(x_{i+1}, y_1),\ i = 0, \ldots, N - M \\
w_{0,j} = w_{0,j-1} + P_0(x_j, y_j),\ j = 2, \ldots, M \\
w_{i,j} = \min_{k=0,\ldots,i}\{w_{i-k,j-1} + P_k(x_{i+j}, y_j)\},\ i = 1, \ldots, N - M,\ j = 2, \ldots, M \\
P_k(x_{i+j}, y_j) = \begin{cases} k & if\ x_{i+j} = y_j \\ k+1 & if\ x_{i+j} \neq y_j \end{cases}, k = 0, ..., i
\end{gathered} \tag{1}$$

$P_k(x_{i+j}, y_j)$ gives the cost of the deletion of k bits previous to x_{i+j} plus its substitution by its complementary if $x_{i+j} \neq y_j$. Note that a maximum length k of possible runs of decimations is assumed for constrained edit distance matrices. It is also remarkable that at each stage the minimum has to be obtained in order to extend the search at a next stage, which implies the need to maintain a record of the search in the same way that Viterbi algorithm saves a back pointer to the previous state on the maximum probability path.

In order to avoid the computation of the edit distances for all possible initial sequences, we propose a graph-theoretical approach so that the computation of edit distances may be seen as a search through a basic graph. Such a basic graph is a directed rooted tree where each non-root vertex $(i + j, j), i = 0, 1, , N - M; j = 1, 2, , M$, indicates a correspondence between the bits x_{i+j} and y_j and each edge indicates either a deletion of the bit x_{i+j} when $j = 0$, or a possible transition due to a deletion (D) or a substitution (S), in the remaining cases. In this way, the computation of edit distances consists in finding the shortest paths through the graph.

For the description of our improvement, we now define a new weighted directed graph, here called induced graph, where the costs of shortest paths come directly from the elements of the matrix W. This induced graph is computed from the basic graph such as follows: if we eliminate vertical edges in the previous graph by computing the partial transitive closure of every pair of edges of the form $((i + j - 2, j - 1), (i + j - 1, j - 1))$ and $((i + j - 1, j - 1), (i + j, j))$ and by substituting them by the edge $((i + j - 2, j - 1), (i + j, j))$, then we get the graph that will be called induced graph.

In this graph there are as many vertices as elements in the matrix W, plus an additional source and an additional sink. On the other hand, the directed edges in this induced graph are defined from the computation of the edit distances described in Equation (1), plus additional edges joining the source with the vertices associated to the first column of W and additional edges joining the vertices associated to the last column of W with the sink. For instance, the induced graph corresponding to a constrained edit distance matrix with runs of decimations of maximum length 1 has $(N - M + 1) \cdot (2M - N + 2)$ vertices and $2 \cdot (N - M + 1) \cdot (2M - N + 2) - M - 3$ edges. Moreover, edges in the induced graph have different costs depending on the specific pair of sequences X and Y, and particularly on the coincidences between the corresponding bits of both sequences, as described in Equation (1). Note that in the induced graph, the shortest paths between the source and the sink give us the solution of the cryptanalytic attack through the specification of both decimation and noise sequences that can be extracted from them.

3 Search of Promising Initial States

The main idea behind the method shown in this Section comes directly from the association between bits x_{i+j} and edges of the induced graph. Since the calculation of the minimum edit distance implies the computation of some shortest path in such a graph, cut sets between the source and the sink in the induced graph may be useful in order to define a set of conditions for candidate sequences so that it is possible to establish a minimum threshold edit distance. In this way, once an intercepted sequence fulfills some of those stated conditions, the cost of the corresponding cut set can be guaranteed to be minimal for some possible candidate sequence, what has direct consequences on the costs of the shortest paths, that is to say, on the edit distances.

In this way, as soon as an intercepted sequence fulfills some specific condition defined below, and this fact allows the description of a candidate and feasible initial sequence, we will know that such an initial sequence will provide us with a useful upper threshold for the edit distance and even in many cases, such a sequence will be a minimum edit distance sequence.

The specific cut sets that we have used for the numerical results shown in this work are defined as follows. Each cut set $C_{i+j}, 2 \leq i+j \leq N-1$ contains:

1. The set of all the arcs corresponding to the vertex x_{i+j}.
2. All those edges corresponding to bits x_w with $w > i+j$ whose output vertex is one of the output vertices of the former set.

For the first model, these cut sets may be characterized by several independent conditions on the intercepted sequence Y that may be used to guarantee a decrease on the edit distances of different candidate sequences X. After having checked each hypothesis separately, the tools used to check both sets of conditions on candidate sequences X are described in terms of a pattern that is made out of independent bits of X according to the formulas in Equation (2).

If $\forall j : 2,3,\ldots,N-M+1;\ y_1 = y_2 = \cdots = y_j$ then $y_j = x_1 = x_2 = \cdots = x_{j+N-M}$
If $\forall j : N-M+2, N-M+3,\ldots,M; y_{M-N+j} = \cdots = y_{j-1} = y_j$ then $y_j = x_j = x_{j+1} = \cdots = x_{j+N-M}$
If $\forall j : M+1, M+2,\ldots,N-1;\ y_{M-N+j} = \cdots = y_{M-1} = y_M$ then $y_M = x_j = x_{j+1} = \cdots = x_N$
(2)

For the second model, the cut sets may be characterized by different independent conditions on the intercepted sequence X that may be used to guarantee a decrease on the edit distances of candidate sequences Y. After having checked each hypothesis separately, the tools used to check both sets of conditions on candidate sequences Y are described in terms of a pattern that is made out of independent bits of Y according to the formulas in Equation (3).

If $\forall j : 2,3,\ldots,N-M+1;\ x_1 = x_2 = \cdots = x_{j+N-M}$ then $x_1 = y_1 = y_2 = \cdots = y_j$
If $\forall j : N-M+2, N-M+3,\ldots,M; x_j = x_{j+1} = \cdots = x_{j+N-M}$ then $x_j = y_{M-N+j} = \cdots = y_{j-1} = y_j$
If $\forall j : M+1, M+2,\ldots,N-1;\ x_j = x_{j+1} = \cdots = x_N$ then $x_j = y_{M-N+j} = \cdots = y_{M-1} = y_M$
(3)

For checking previous equations (2) and (3), it is necessary to determine the value of N, which depends on k that is the maximum length of possible runs of decimations. For example, if $k = 1$, then $N = 3M/2$, which is the mathematical expectation of N in such a case.

Note that the checking procedure of hypothesis described with the previous Equations, applied on the intercepted sequence takes polynomial time as it implies a simple verification of runs. The previous patterns allow one to discover promising initial states producing sequences with a low edit distance. In fact, such a pattern provides a good quality threshold for the method that will be described in the following Section.

4 General Attack

The threshold obtained through the pattern described in the previous Section is a fundamental ingredient of the general attack described below. The algorithm here developed also makes use of a new concept, the so-called *stop column*, which leads to a considerable saving in the computation of the edit distance matrices. Indeed, a *stop column* with respect to a threshold T may be defined as a column j_0 of the edit distance matrix W such that each one of their elements fulfills the Equation (4).

$$w_{i,j_0} > T - (N - M - i), \forall i \tag{4}$$

Once a minimum edit distance threshold has been obtained, we may use such a threshold to stop the computation of any matrix W as soon as a *stop column* has been detected. This is due to the fact that the edit distance corresponding to the candidate initial state will be worse than the threshold. In this simple way, two new improvements on the original attack may be achieved. On the one hand, as yet mentioned, the computation of any matrix may be stopped as soon as a *stop column* is obtained. On the other hand and thanks to the association between bits x_{i+j} and edges of the graph, we may define a new anti-pattern on the initial states of the target $LFSR$, the so-called IS-anti-pattern. This new parameter allows us to discard the set of initial states fulfilling such an IS-anti-pattern when an early *stop column* has been detected. This is so because once a *stop column* has been obtained, it is possible to discard directly all the initial states whose first bits coincide with those that produce the *stop column*. In order to take full advantage of *stop columns*, it is convenient to have some efficient way of obtaining a good threshold. That is exactly the effect of the pattern described in the previous Section.

Since it is possible that the described pattern correspond only to sequences that may not be produced by the target $LFSR$, in practice it is convenient to restrict the pattern to the length of the target $LFSR$. So, the pattern obtained from the first formulas of Equations (2) and (3) limited to the length of the target $LFSR$ is what we call IS-pattern. On the other hand, although sequences generated through the IS-pattern have minimum edit distance, it is possible that the corresponding obtained decimation or insertion sequences and noise sequences are not consistent with the description of the attacked generator. This is the reason why the proposed algorithm includes a process of hypothesis relaxation, which implies the successive complementation of bits of the IS-pattern until getting a positive result.

Finally, since the IS-pattern is determined by the runs at the beginning of the intercepted sequence, if no long run exists at the beginning of the sequence, initially the algorithm discards the first bits in the intercepted sequence before a long run, and use those discarded bits to confirm the result of the attack. This idea is expressed within the

algorithm by a parameter $H \in [0, L]$, chosen by the attacker depending on its computational capacity (the greater capacity, the fewer H).
The full description of the proposed general edit distance attack is as follows.

Algorithm.

Input: The intercepted keystream sequence and the feedback polynomial of the target $LFSR$ of length L.

Output: The initial states of the target $LFSR$ producing sequences with a low edit distance with the intercepted sequence, and the corresponding decimation or insertion sequence and noise sequence.

1. Verification of hypothesis on the intercepted sequence described in Equation (2) or (3).
2. While fewer than H hypothesis are fulfilled, discard the first bit and consider the resulting sequence as new intercepted sequence.
3. Definition of the IS-pattern according to the first L formulas in Equation (2) or (3).
4. Initialization of the threshold $T = N$.
5. For each initial state fulfilling the IS-pattern, which has not been previously rejected:
   (a) Computation of the edit distance matrix, stopping after detecting a *stop column* according to threshold T and Equation (4).
   (b) Definition of the IS-anti-pattern and rejection of all initial states fulfilling it.
   (c) Updating of the threshold T.
6. For each initial state producing a sequence with minimum edit distance:
   (a) Computation of the shortest paths from the graph induced by the edit distance matrix.
   (b) Translation from each shortest path into decimation or insertion sequences and noise sequences.
   (c) Checking that the obtained decimation or insertion sequences, and noise sequences are consistent with the attacked generator. Otherwise, updating of the IS-pattern by complementing one of the bits in the original IS-pattern.

Note that if the output is not the minimum edit distance sequence, the obtained edit distance can be used as threshold for the stop column method in order to find such a sequence quickly.

5 Attack on Shrinking and Alternating Step Generators

In this Section a specific implementation of the general attack presented in the previous Section for the cases of the SG and the ASG is considered.

One of the first questions that have to be taken into account in both cases is the limitation on the number of consecutive deletions because the longest run of consecutive deletions in X to get Y is always shorter than the length L_S of the selector register S. This restriction implies that the equation (1) corresponding to the computation of the edit distance matrix should be modified in the following way:

$$\begin{gathered}
w_{i,1} = P_i(x_{i+1}, y_1),\ i = 0, \ldots, L_S \\
w_{0,j} = w_{0,j-1} + P_0(x_j, y_j),\ j = 2, \ldots, M \\
w_{i,1} = \infty,\ i = L_S + 1, \ldots, N - M \\
w_{i,j} = \min_{k=0,\ldots,L_S-1}\{w_{i-k,j-1} + P_k(x_{i+j}, y_j)\},\ i = 1, \ldots, N - M,\ j = 2, \ldots, M \\
P_k(x_{i+j}, y_j) = \begin{cases} k & \text{if } x_{i+j} = y_j \\ k+1 & \text{if } x_{i+j} \neq y_j \end{cases}\ k = 0, ..., L_S - 1
\end{gathered} \tag{5}$$

Equations (2) and (3) corresponding to the definition of the pattern in the first and the second model, respectively must be also adapted to the SG and the ASG, producing the Equations (6) and (7) respectively:

If $\forall j : 2, 3, \ldots, N - M + 1; y_{1+j/L_S} = \cdots = y_{j-1} = y_j$ then $y_j = x_{L_S(j/L_S)} = x_{L_S(j/L_S)+1} = \cdots = x_{L_S(j/L_S)+L_S-1}$
If $\forall j : N - M + 2, N - M + 3, \ldots, M; y_{M-N+j} = \cdots = y_{j-1} = y_j$ then $y_j = x_j = x_{j+1} = \cdots = x_{j+L_S-1}$
If $\forall j : M + 1, M + 2, \ldots, N - 1;\ y_{M-N+j} = \cdots = y_{M-(N-j)/L_S}$ then $y_M = x_j = x_{j+1} = \cdots = x_{\min(j+L_S-1,N)}$

(6)

If $\forall j : 2, 3, \ldots, N - M + 1; x_{L_S(j/L_S)} = x_{L_S(j/L_S)+1} = \cdots = x_{L_S(j/L_S)+L_S-1}$ then $x_{L_S(j/L_S)} = y_{1+j/L_S} = \cdots = y_{j-1} = y_j$
If $\forall j : N - M + 2, N - M + 3, \ldots, M; x_j = x_{j+1} = \cdots = x_{j+L_S-1}$ then $x_j = y_{M-N+j} = \cdots = y_{j-1} = y_j$
If $\forall j : M + 1, M + 2, \ldots, N - 1;\ x_j = x_{j+1} = \cdots = x_{\min(j+L_S-1,N)}$ then $x_j = y_{M-N+j} = \cdots = y_{M-(N-j)/L_S}$

(7)

Finally, the process of hypothesis relaxation explained in the last Section must also be used for the cases of SG and ASG when the minimum obtained edit distance is greater than $N - M$ since it corresponds to the presence of noise.

6 Simulation Results

From several randomly generated examples, we may deduce a general classification of inputs into several cases. The best ones correspond to IS-patterns which directly identify solutions. On the contrary, bad cases are those ‘missing the event’ cases in which the IS-pattern fails to identify any correct initial state. Such cases are generally associated with long runs at the beginning of the sequences Y. Finally, the medium cases are those for which, despite the non existence of solutions fulfilling the pattern, a good threshold is obtained. Such cases allow a good percentage of saving in computing thanks to the detection of many early *stop columns*.

From the obtained results and the relationship between L_A and Seq.pat. we may deduce that the proposed algorithm produces the solution in $O(2^{L_A/2})$ time instead of the $O(2^{L_A})$ time corresponding to the exhaustive search that implied the original attack [13]. Furthermore, it is clear that the worst outputs appear when the initial results in steps 1 to 4 are not adequate as there are no initial states fulfilling the IS-pattern. However, even in these cases that require more computation, it is guaranteed that the solution is always obtained.

Note that as aforementioned, the proposed algorithm not always output the minimum edit distance sequence. However, since the hypothesis on Y are independent, the groups of bits in the IS-pattern are also independent and, consequently, the conditions might be considered separately in such way that we might define in this way a relaxed IS-pattern which might lead to sequences that fulfill them. In addition, empirical results have shown that intercepted sequences Y with short runs at the beginning cause a greater improvement in the time complexity of the attack. Thus, another way to avoid a bad behavior of the original algorithm is by choosing sub-sequences from the intercepted sequence Y that have no too long runs at the beginning, and by applying the algorithm to each one of these sub-sequences.

7 Conclusions

In this work a new deterministic approach to the cryptanalysis of $LFSR$-based stream ciphers has been proposed. In particular, a practical improvement on the edit distance attack on certain clock-controlled $LFSR$-based generators has been proposed, which reduces the computational complexity of the original attack because it does not require an exhaustive search over all the initial states of the target $LFSR$. The main tool used for the optimization of the original attack was the definition of graphs where optimal paths provide cryptanalytic results and of cut sets on them that have been used to obtain a useful threshold to cut the search tree.

An extension of this article, which is being part of a work in progress, takes advantage of the basic idea of using cut sets to improve edit distance attacks against generalized clock-controlled $LFSR$-based generators. In order to do this, the three edit operations are considered and the resulting cut sets on the corresponding induced graph allow the identification of promising initial states.

References

1. Caballero-Gil, P. and Fúster-Sabater, A. (2005): Improvement of the Edit Distance Attack to Clock-Controlled LFSR-Based Stream Ciphers. *Lecture Notes in Computer Science* 3643 Springer-Verlag: 355-364.
2. Coppersmith, D., Krawczyk, H. and Mansour, H. (1994): The Shrinking Generator. *Lecture Notes in Computer Science* 773 Springer-Verlag: 22-39.
3. Daemen, J. and Van Assche G. (2005): Distinguishing Stream Ciphers with Convolutional Filters.*Cryptology ePrint Archive* Report 039.
4. Golic, J.D. (1998): Recent Advances in Stream Cipher Cryptanalysis. *Publication de l'Institut Mathematique* Tome 64 (78) : 183-204.
5. Golic, J.D. (2005): Embedding probabilities for the Alternating Step Generator. *IEEE Transactions on Information Theory*, Volume 51, Issue 7, July: 2543-2553.
6. Golic, J.D. and Mihaljevic, M.J. (1991): A Generalized Correlation Attack on a Class of Stream Ciphers Based on the Levenshtein Distance. *Journal of Cryptology* 3, No. 3: 201-212.

7. Golic, J.D. and Petrovic, S. (1993): A Generalized Correlation Attack with a Probabilistic Constrained Edit Distance. *Lecture Notes in Computer Science* 658 Springer-Verlag: 472-476.
8. Gollmann, D. and Chambers, W.C. (1989): Clock-Controlled Shift Registers: A Review. *IEEE Transactions on Selected Areas in Communications* SAC-7 May: 525-533.
9. Gomulkiewicz, M., Kutylowski, M. and Wlaz, P. (2006): Fault Jumping Attacks against Shrinking Generator. N. 06111, Dagstuhl Seminar Proceedings.
10. Günther, C.G. (1988): Alternating Step Generators Controlled by De Bruijn Sequences. *Lecture Notes in Computer Science* 304, Springer-Verlag: 5-14.
11. Jiang, S. and Gong, G. (2002): On Edit Distance Attack to Alternating Step Generator. *Technical Report* Corr2002-28, University of Waterloo
12. Kanso, A. (2003): Clock-Controlled Shrinking Generator of Feedback Shift Registers. *Lecture Notes in Computer Science* 2727, Springer-Verlag: 443-451.
13. Petrovic, S. and Fúster, A. (2004): Clock Control Sequence Reconstruction in the Ciphertext Only Attack Scenario. *LNCS* 3269, Springer-Verlag: 427-439.