Correlation Analysis of the Shrinking Generator

Jovan D. Golić

GEMPLUS
Rome CryptoDesign Center, Technology R&D
Via Pio Emanuelli 1, 00143 Rome, Italy
jovan.golic@gemplus.com

Abstract. The shrinking generator is a well-known keystream generator composed of two linear feedback shift registers, LFSR_1 and LFSR_2, where LFSR_1 is clock-controlled according to regularly clocked LFSR_2. A probabilistic analysis of the shrinking generator which shows that this generator can be vulnerable to a specific fast correlation attack is conducted. The first stage of the attack is based on a recursive computation of the posterior probabilites of individual bits of the regularly clocked LFSR_1 sequence when conditioned on a given segment of the keystream sequence. Theoretical analysis shows that these probabilities are significantly different from one half and can hence be used for reconstructing the initial state of LFSR_1 by iterative probabilistic decoding algorithms for fast correlation attacks on regularly clocked LFSR's. In the second stage of the attack, the initial state of LFSR_2 is reconstructed in a similar way, which is based on a recursive computation of the posterior probabilites of individual bits of the LFSR_2 sequence when conditioned on the keystream sequence and on the reconstructed LFSR_1 sequence.

Keywords. Stream ciphers, unconstrained irregular clocking, posterior probabilities, fast correlation attacks.

1 Introduction

The shrinking generator [1] is a well-known keystream generator for stream cipher applications. It consists of only two linear feedback shift registers (LFSR's). The clock-controlled LFSR, LFSR_1, is irregularly clocked according to the clock-control LFSR, LFSR_2, which is regularly clocked. More precisely, at each time, both LFSR's are clocked once and the bit produced by LFSR_1 is taken as the output bit if the clock-control bit produced by LFSR_2 is equal to 1. Otherwise, the output bit is not produced. The output sequence is thus a nonuniformly decimated LFSR_1 sequence. It is recommended in [1] that the LFSR initial states and the feedback polynomials be defined by the secret key. Under certain conditions, the output sequences possess a long period, a high linear complexity, and good statistical properties.

As pointed out in [1], a basic divide-and-conquer attack on the shrinking generator is the linear consistency attack [17] on LFSR_2 which requires the exhaustive search through all possible initial states and feedback polynomials of

J. Kilian (Ed.): CRYPTO 2001, LNCS 2139, pp. 440–457, 2001.

LFSR_2. On the other hand, a probabilistic correlation attack targeting LFSR_1 which requires the exhaustive search through all possible initial states and feedback polynomials of LFSR_1 is proposed in [4] and analyzed by computer simulations in [15]. A reduced complexity method based on searching for specific subsequences of the output sequence is suggested in [9], but both the complexity and the required keystream segment length are exponential in the length of LFSR_1.

It is shown in [3] that the output sequence may have a detectable linear statistical weakness if the feedback polynomial of LFSR_1 has low-weight polynomial multiples of moderately large degrees. It is suggested in [5] that this weakness may even be used for recovering the LFSR_1 feedback polynomial. A theoretical framework for a fast correlation attack targeting the initial state of LFSR_1 is also proposed in [5], but the attack is not implemented as it requires a search for specific polynomial multiples of the LFSR_1 feedback polynomial.

The objective of this paper is to investigate if the initial states of LFSR_1 and LFSR_2 can be reconstructed by an algorithm that would not require the exhaustive search through all possible initial states and whose complexity can be sufficiently small even for large LFSR lengths. The LFSR feedback polynomials are assumed to be known. The basic point of our approach is to consider the posterior probabilites of individual bits of the regularly clocked LFSR_1 sequence when conditioned on a given segment of the keystream sequence. In the probabilistic model where the LFSR sequences are assumed to be independent and purely random,[1] a recursion and an explicit expression for computing these probabilities with complexity quadratic in the keystream segment length are both derived. A theoretical analysis shows that the computed posterior probabilities can be significantly different from one half for a purely random output sequence. In a more general probabilistic model, in which the LFSR_1 sequence is assumed to be a sequence of independent, not necessarily uniformly distributed, binary random variables, it is proved that the posterior probabilities can be recursively computed with complexity cubic in the keystream segment length.

Accordingly, as these probabilities represent soft-valued estimates of the corresponding bits of the regularly clocked LFSR_1 sequence, they can be used in an iterative probabilistic decoding algorithm for fast correlation attacks on regularly clocked LFSR's (e.g., see [11], [12], and [8]). It is known that the complexity of such an algorithm primarily depends on the degrees and numbers of low-weight polynomial multiples of the feedback polynomial of LFSR_1 which, according to [10], [7], and [14], may also contain an additional number of concentrated nonzero terms. The initial state of LFSR_1 can thus be recovered. A more sophisticated method in which the posterior probabilities are iteratively updated by intertwining the probabilistic decoding with the recursive computation is also introduced.

In addition, a composite method that effectively enhances the posterior probabilities for longer keysteam segments is proposed. Essentially, it consists in

[1] A sequence of independent uniformly distributed random variables over a finite set is called purely random.

dividing a longer keystream segment into subsegments of equal length, in computing the posterior probabilities for the subsegments, and then in combining these posterior probabilities appropriately.

If the posterior probabilities corresponding to a given keystream sequence are not sufficiently different from one half, they can be computed for subsequences of the keystream sequence obtained by discarding the initial segment of variable length until the significant posterior probabilities are obtained. This will improve the performance of the fast correlation attacks explained above, but the length of the initial LFSR_1 segment has to be guessed. For the initial output segment of length $j-1$, one has to make $O(\sqrt{2j})$ guesses around the expected value $2j-1$. Moreover, one can thus also search for the outstanding posterior probabilities and then apply an information set decoding algorithm to recover the LFSR_1 initial state. The success of such an algorithm is independent of the LFSR_1 feedback polynomial, but the achievable complexity is still exponential in the length of LFSR_1. This improves the reduced complexity method [9].

The second point of our approach is to consider the posterior probabilites of individual bits of the regularly clocked LFSR_2 sequence when conditioned on a given segment of the keystream sequence and on the reconstructed LFSR_1 sequence, as suggested in [9]. It is proved that these probabilities can be recursively computed with complexity cubic in the keystream segment length, thus showing that the expression given in [9] is incorrect. As the LFSR_1 sequence is assumed to be known, the computed posterior probabilities are more distinguished from one half than in the case of LFSR_1. This makes the reconstruction much easier. Consequently, the initial state of LFSR_2 can be recovered either by an iterative probabilistic decoding algorithm or by a simple information set decoding algorithm using a subset of the probabilities close to zero or one.

Section 2 contains an overview of known results concerning the posterior probabilities of blocks of LFSR_1 bits. The results regarding the posterior probabilities of individual LFSR_1 and LFSR_2 bits are presented in Sections 3 and 4, respectively. These posterior probabilities are theoretically analyzed in Section 5. The combined fast correlation attacks are proposed in Section 6, and conclusions are given in Section 7. Proofs of two underlying theorems are presented in Appendices A and B.

2 Posterior Probabilities of Blocks of LFSR_1 Bits

We use the notation $A = a_1, a_2, \ldots$ for a binary sequence, A_k for its subsequence $a_k, a_{k+1}, \ldots$, A^n for its prefix $(a_i)_{i=1}^n = a_1, a_2, \ldots a_n$, and A_k^n for its subsequence $(a_i)_{i=k}^n = a_k, a_{k+1}, \ldots, a_n$. If its length is finite, then A is called a string. Let $w(A)$ and $d(A)$ denote the numbers of 1's and 0's in A, respectively. For simplicity, we keep the same notation for random variables and their values.

Let X, C, and Y denote the output sequences of LFSR_1, LFSR_2, and the shrinking generator itself, respectively. In a general model, let X and C be arbitrary binary sequences. Then Y is obtained from X by the nonuniform decimation according to C, that is, a bit x_i is deleted from X iff $c_i = 0$. Accordingly,

Y is a function of X and C, $Y = F(X, C)$, where the length of Y may be finite and is equal to $w(C)$. Thus Y^n is a function of X and C, $Y^n = F^n(X, C)$, for any $1 \le n \le w(C)$. If $w(C) = 0$, then Y is not produced. If $w(C^n) = l \ge 1$ and $c_n = 1$, then $y_l = x_n$. It follows that y_n is a function of X_n and C, $f_n(X_n, C)$.

We assume a probabilistic model where X and C are independent and purely random binary sequences. It then follows that the output sequence Y is also purely random. We are first interested in deriving the posterior probability $\Pr\{X^n \mid Y\}$ which is in this model equal to $\Pr\{X^n \mid Y^n\}$. To this end, according to [4], define the following conditional probability for prefixes of X and Y

$$Q(e, s) \stackrel{def}{=} \Pr\{Y^s, d(C^{e+s}) = e \mid X^{e+s}\}. \tag{1}$$

It is in fact the probability that Y^s is obtained by deleting e bits from a given string X^{e+s}. The permissible values of s and e are $0 \le s \le n$ and $0 \le e \le n - s$, where Y^0 denotes an empty set and, formally, $Q(0, 0) = 1$. This probability can be computed recursively by

$$Q(e, s) \;=\; \frac{1}{2}\, Q(e-1, s) \;+\; \frac{1}{2}\, \delta(x_{e+s}, y_s)\, Q(e, s-1) \tag{2}$$

where the terms on the right-hand side corresponding to unpermissible values of e or s (i.e., for $e = 0$ or $s = 0$) are assumed to be equal to zero (see [4] and Appendix B). Here, $\delta(i, j)$ or $\delta_{i,j}$ is the Kronecker symbol, i.e., $\delta(i, j) = 1$ if $i = j$ and $\delta(i, j) = 0$ if $i \ne j$.

Consequently, we have

$$\begin{aligned}
\Pr\{Y^n \mid X^n\} &= \sum_{e=0}^{n} \Pr\{Y^n, d(C^n) = e \mid X^n\} \\
&= \sum_{e=0}^{n} \Pr\{Y_{n-e+1}^n \mid Y^{n-e}, d(C^n) = e, X^n\}\, Q(e, n-e) \\
&= \sum_{e=0}^{n} 2^{-e}\, Q(e, n-e)
\end{aligned} \tag{3}$$

in view of the fact that, on the condition that $d(C^n) = e$, the string Y_{n-e+1}^n is obtained by decimating X_{n+1} according to C_{n+1}, where X_{n+1} and C_{n+1} remain to be mutually independent and purely random (even when conditioned on X^n and Y^{n-e}). Therefore, under the given conditions, Y_{n-e+1}^n remains to be uniformly distributed. Further, as X^n and Y^n are both uniformly distributed, we have

$$\Pr\{X^n \mid Y^n\} \;=\; \Pr\{Y^n \mid X^n\} \;=\; \sum_{e=0}^{n} 2^{-e}\, Q(e, n-e) \tag{4}$$

which is computed in $O(n^2)$ time and $O(n)$ space. The probability (4) can be found in [9], and also corresponds to the probability derived in [6] for the alternating step generator, because the nonuniform decimation of a purely random

sequence can be regarded as the inverse operation to the nonuniform interleaving of two purely random sequences which is inherent to this generator.

For ease of computation, one can introduce $N(e,s) = 2^{e+s}\,Q(e,s)$ which represents the number of clock-control strings C^{e+s} that result in Y^s from a given X^{e+s}. These integers can be computed by the recursion

$$N(e,s) \;=\; N(e-1,s) \;+\; \delta(x_{e+s}, y_s)\, N(e,s-1). \tag{5}$$

Then

$$\Pr\{X^n \mid Y^n\} \;=\; 2^{-n} \sum_{e=0}^{n} 2^{-e}\, N(e, n-e). \tag{6}$$

It is proposed in [4] to use the probability $Q(m-n,n)$, where $m \approx 2n$, in order to reconstruct the LFSR_1 initial state from a given keystream segment Y^n. This probability is computed in $O(n(m-n)) = O(n^2)$ time. Statistical experiments from [15] show that $n \approx 20r_1$ is sufficient for a successful reconstruction.[2] Here, $Q(m-n,n)$ is used as a measure of correlation between Y^n and X^m, where X^m is produced from an assumed LFSR_1 initial state. It would be interesting to compare $Q(m-n,n)$ with the posterior probability (4) with respect to the minimum keystream segment length and the complexity required. However, the exhaustive search over all possible LFSR_1 initial states is required for both measures. It is worth mentioning that a conclusion from [9] that the required n is independent of r_1 is incorrect, because, according to the deletion channel capacity argument, n must be linear in r_1 (see [4] and [15]).

3 Posterior Probabilities of Individual LFSR_1 Bits

In this section, the posterior probabilities of individual bits of the regularly clocked LFSR_1 sequence when conditioned on a given segment of the keystream sequence are introduced. In Section 3.1, it is shown that these probabilities can be computed recursively in a probabilistic model in which the LFSR_2 sequence is assumed to be purely random, the LFSR_1 sequence is assumed to be a sequence of independent binary random variables, and both sequences are assumed to be mutually independent. This general model is relevant for a fast correlation attack on LFSR_1 in which the posterior probabilities are iteratively updated by intertwining the recursive computation with a probabilistic decoding algorithm used in fast correlation attacks on regularly clocked LFSR's. In Section 3.2, a special case of this model in which the LFSR_1 sequence is assumed to be purely random is considered. This case is especially relevant for a fast correlation attack on LFSR_1 in which the posterior probabilities recursively computed in the first stage are then processed by an iterative probabilistic decoding algorithm in the second stage.

[2] The length of LFSR_i is denoted as r_i, $i = 1, 2$.

3.1 General Probabilistic Model

Generalize the probabilistic model from Section 2 in such a way that a prefix of X need not be purely random. More precisely, let X be a sequence of independent binary random variables (bits) such that $\Pr\{x_i = 1\} = p_i$ for $1 \le i \le n$ and $\Pr\{x_i = 1\} = 0.5$ for $i > n$, where n is a given positive integer. Our objective here is to determine the posterior probabilities $\hat{p}_i = \Pr\{x_i = 1 \mid Y^n\}$ for $1 \le i \le n$. It follows that

$$\hat{p}_i \;=\; p_i \,\frac{\Pr\{Y^n \mid x_i = 1\}}{\Pr\{Y^n\}}. \tag{7}$$

The problem is how to compute the probabilities $\Pr\{Y^n \mid x_i = 1\}$ and $\Pr\{Y^n\}$ efficiently. To this end, introduce the following partial probabilities, for prefixes of Y,

$$P_i(e,s) \;\stackrel{def}{=}\; \Pr\{Y^s, d(C^{e+s}) = e \mid x_i = 1\} \tag{8}$$

$$P(e,s) \;\stackrel{def}{=}\; \Pr\{Y^s, d(C^{e+s}) = e\} \tag{9}$$

for $0 \le s \le n$ and $0 \le e \le n - s$, where formally $P(0,0) = 1$ and $P_i(0,0) = 1$.

The following theorem, proved in Appendix A, shows that the partial probabilities can be computed recursively and then used to obtain the desired posterior probabilities by (7).

Theorem 1. *For any given Y^n and each $1 \le i \le n$, we have*

$$\hat{p}_i \;=\; p_i \,\frac{\sum_{e=0}^{n} 2^{-e}\, P_i(e, n-e)}{\sum_{e=0}^{n} 2^{-e}\, P(e, n-e)} \tag{10}$$

where the partial probabilities are determined recursively by

$$\begin{aligned} P_i(e,s) = {} & \frac{1}{2}\, P_i(e-1,s) \\ & + \frac{1}{2}\,(\delta_{i,e+s} y_s + (1-\delta_{i,e+s})(y_s p_{e+s} + (1-y_s)(1-p_{e+s})))\, P_i(e,s-1) \end{aligned} \tag{11}$$

$$P(e,s) = \frac{1}{2}\, P(e-1,s) \;+\; \frac{1}{2}\,(y_s p_{e+s} + (1-y_s)(1-p_{e+s}))\, P(e,s-1) \tag{12}$$

for $0 \le s \le n$, $0 \le e \le n - s$, and $(e,s) \ne (0,0)$, from the initial values $P_i(0,0) = P(0,0) = 1$. (The terms on the right-hand sides of these equations corresponding to unpermissible values of e or s, i.e., for $e = 0$ or $s = 0$, are assumed to be equal to zero.)

The time and space complexities of the corresponding algorithm are clearly $O(n^3)$ and $O(n)$, respectively. The algorithm may thus be feasible even if n is large. For computational convenience, the multiplicative factor 0.5 can be removed from the recursions without affecting the values of the posterior probabilities. The time complexity can be reduced to $O(n^2\sqrt{n})$ if $P_i(e,s)$ and $P(e,s)$ are computed approximately, only for $O(\sqrt{2s})$ values of e around s.

3.2 Purely Random String Probabilistic Model

Consider now the model in which X is a purely random sequence. It is a particular instance of the general model from Section 3.1 in which $p_i = 0.5$, $1 \le i \le n$. In this model, the recursion (12) can be explicitly solved as $P(e,s) = \binom{e+s}{e} 2^{-(e+2s)}$, so that $\Pr\{Y^n\} = 2^{-n}$, as to be expected. Accordingly, the posterior probabilities can be computed by the following corollary to Theorem 1.

Corollary 1. *If X is purely random, then for any given Y^n and each $1 \le i \le n$, we have*

$$\hat{p}_i \;=\; 2^{n-1} \sum_{e=0}^{n} 2^{-e}\, P_i(e, n-e) \tag{13}$$

where the partial probability is determined recursively by

$$P_i(e,s) \;=\; \frac{1}{2}\, P_i(e-1,s) \;+\; \frac{1}{4}\,(1 + \delta_{i,e+s}(2y_s - 1))\, P_i(e,s-1) \tag{14}$$

for $0 \le s \le n$, $0 \le e \le n-s$, and $(e,s) \ne (0,0)$, from the initial value $P_i(0,0) = 1$.

Further simplification and an explicit expression can be obtained by using the fact that X is purely random. Namely, in a similar way as (34) in Appendix A, we obtain

$$\begin{aligned}
\Pr\{Y^n \mid x_i = 1\} &= \sum_{e=0}^{i} \Pr\{Y^n, d(C^i) = e \mid x_i = 1\} \\
&= \sum_{e=0}^{i} \Pr\{Y_{i-e+1}^n, Y^{i-e}, d(C^i) = e \mid x_i = 1\} \\
&= \sum_{e=0}^{i} \Pr\{Y_{i-e+1}^n \mid Y^{i-e}, d(C^i) = e, x_i = 1\}\, P_i(e, i-e) \\
&= 2^{-(n-i)} \sum_{e=0}^{i} 2^{-e}\, P_i(e, i-e) \;=\; 2^{-(n-i)}\, \Pr\{Y^i \mid x_i = 1\}.
\end{aligned} \tag{15}$$

As a consequence, we have

$$\Pr\{x_i = 1 \mid Y^n\} \;=\; \Pr\{x_i = 1 \mid Y^i\}. \tag{16}$$

Also, it follows that

$$P_i(e, i-e) \;=\; \frac{1}{2}\, P(e-1, i-e) \;+\; \frac{1}{2}\, y_{i-e}\, P(e, i-e-1) \tag{17}$$

where $P(e,s) = 2^{-(e+2s)}\, M(e,s)$, $M(e,s) = \binom{e+s}{e}$, and the binomial coefficients can be computed recursively by

$$M(e,s) \;=\; M(e-1,s) \;+\; M(e,s-1) \tag{18}$$

for $0 \le s \le n-1$, $0 \le e \le n-1-s$, and $(e,s) \ne (0,0)$, from the initial value $M(0,0) = 1$. Then (13) and (17) imply that

$$\hat{p}_i \;=\; \frac{1}{2}\,2^{-i} \sum_{e=0}^{i} \left(\binom{i-1}{e-1} \;+\; 2 \binom{i-1}{e}\, y_{i-e} \right). \tag{19}$$

Finally, we obtain the following theorem.

Theorem 2. *If X is purely random, then for any given Y^n and each $1 \le i \le n$, we have*

$$\hat{p}_i \;=\; \frac{1}{2} \left(\frac{1}{2} \;+\; 2^{-(i-1)} \sum_{e=0}^{i-1} \binom{i-1}{e}\, y_{i-e} \right). \tag{20}$$

The time and space complexities of the algorithm corresponding to Theorem 2 are $O(n^2)$ and $O(n)$, respectively, where the binomial coefficients can be recursively precomputed in $O(n^2)$ time by using (18). However, (20) shows that $\hat{p}_i$ can be numerically approximated with an arbitrarily small error by using only $O(\sqrt{i-1}/2)$ values of e around $(i-1)/2$. This reduces the time complexity to $O(n\sqrt{n})$.

The following immediate corollary to Theorem 2 shows that the posterior probabilities cannot approach 0 or 1.

Corollary 2. *If X is purely random, then for any given Y^n and each $1 \le i \le n$, we have*

$$\frac{1}{4} \;\le\; \hat{p}_i \;\le\; \frac{3}{4} \tag{21}$$

where the lower and upper bounds are achieved if and only if Y^i consists of all 0's and of all 1's, respectively.

4 Posterior Probabilities of Individual LFSR$_2$ Bits

In this section, it is shown that the posterior probabilities of individual bits of the regularly clocked LFSR$_2$ sequence when conditioned on a given segment of the keystream sequence and on a segment of the reconstructed LFSR$_1$ sequence can be computed recursively with complexity cubic in the segment length.

Assuming that X and C are independent and purely random, our objective is to determine the posterior probabilities $\hat{q}_i = \Pr\{c_i = 1 \mid Y^n, X^n\}$ for $1 \le i \le n$. It follows that

$$\hat{q}_i \;=\; \frac{1}{2}\, \frac{\Pr\{Y^n \mid c_i = 1, X^n\}}{\Pr\{Y^n \mid X^n\}}. \tag{22}$$

In Section 2, it is shown that $\Pr\{Y^n \mid X^n\}$ can be computed recursively. The problem is how to compute $\Pr\{Y^n \mid c_i = 1, X^n\}$ efficiently. Similarly to (1), define the following conditional probability for prefixes of X and Y

$$Q_i(e,s) \;\stackrel{def}{=}\; \Pr\{Y^s, d(C^{e+s}) = e \mid c_i = 1, X^{e+s}\} \tag{23}$$

for $0 \le s \le n$ and $0 \le e \le n-s$, with $Q_i(0,0) = 1$.

The following theorem, proved in Appendix B, shows that this probability can be computed recursively and then used to obtain the desired posterior probabilities by (22). This theorem shows that the expression for the posterior probabilities given in [9] is incorrect, not only in general, but also in a special case of the probabilities $\Pr\{c_i = 1 \mid Y^i, X^i\}$.

Theorem 3. *For any given Y^n and X^n and each $1 \le i \le n$, we have*

$$\hat{q}_i \;=\; \frac{1}{2}\,\frac{\sum_{e=0}^{n} 2^{-e}\, Q_i(e, n-e)}{\sum_{e=0}^{n} 2^{-e}\, Q(e, n-e)} \tag{24}$$

where $Q(e,s)$ and $Q_i(e,s)$, respectively, are determined recursively by (2) *and by*

$$Q_i(e,s) \;=\; \frac{1}{2}\,(1-\delta_{i,e+s})\,Q_i(e-1,s) \;+\; \frac{1}{2}\,(1+\delta_{i,e+s})\,\delta(x_{e+s}, y_s)\,Q_i(e,s-1) \tag{25}$$

for $0 \le s \le n$, $0 \le e \le n-s$, and $(e,s) \ne (0,0)$, from the initial value $Q_i(0,0) = 1$.

The time and space complexities of the corresponding algorithm are clearly $O(n^3)$ and $O(n)$, respectively. For ease of computation, one can introduce the integers $N_i(e,s) = 2^{e+s}\,Q_i(e,s)$ which can be computed by the recursion

$$N_i(e,s) \;=\; (1-\delta_{i,e+s})\,N_i(e-1,s) \;+\; (1+\delta_{e,i+s})\,\delta(x_{e+s}, y_s)\,N_i(e,s-1). \tag{26}$$

Then

$$\hat{q}_i \;=\; \frac{1}{2}\,\frac{\sum_{e=0}^{n} 2^{-e}\, N_i(e, n-e)}{\sum_{e=0}^{n} 2^{-e}\, N(e, n-e)} \tag{27}$$

where the integers $N(e,s)$ satisfy the recursion (5). The time complexity can be reduced to $O(n^2\sqrt{n})$ if $N_i(e,s)$ and $N(e,s)$ are computed approximately, only for $O(\sqrt{2s})$ values of e around s.

5 Analysis of Posterior Probabilities

The posterior probabilities of individual LFSR_1 bits computed according to Theorem 2 may be useful for reconstructing the unknown LFSR_1 sequence from a known segment of the output sequence if they are sufficiently different from one half. According to Theorem 2 and Corollary 2, the posterior probability $\hat{p}_i$ will be close to 1/4 (3/4) if there is an output segment of length relatively close to $\sqrt{i-1}/2$ around the position $(i-1)/2$ in the output string such that the relative number of 0's (1's) on this segment is considerably different from one half. More generally, if Y^j is relatively unbalanced, that is, if the relative number of 0's in Y^j is considerably different from one half, then most of the posterior probabilities of bits in X^{2j} will be significant.

As $\hat{p}_i$ depends on the output string Y^i, it is interesting to analyze the average value of the absolute difference $|\Delta\hat{p}_i| = |\hat{p}_i - 0.5|$ over purely random Y^i. In view of (20), we get

$$|\Delta\hat{p}_i| \;=\; \frac{1}{2}\,2^{-(i-1)}\,|\sum_{e=0}^{i-1}\binom{i-1}{e}\,(y_{i-e}-0.5)|. \tag{28}$$

Exact analysis of (28) appears to be difficult. However, the following approximate analysis establishes that $|\Delta\hat{p}_i|$ is significantly different from zero for a uniformly distributed Y^i.

The analysis is based on approximating a binomial distribution $B(n, 0.5)$ by a uniform distribution, with the same expected value and standard deviation, over a segment of length $\sqrt{3n}$ centered around $0.5n$. Consequently, let $I(i)$ denote a segment of length $m(i) \approx \sqrt{3(i-1)}$ centered around $0.5(i+1)$. Then (28) reduces to

$$\begin{aligned} |\Delta\hat{p}_i| &\approx \frac{1}{2}\,\frac{1}{m(i)}\,|\sum_{j\in I(i)}(y_j - 0.5)| \\ &\approx \frac{1}{2}\,\frac{1}{m(i)}\,|m_1(i) - 0.5m(i)| \end{aligned} \tag{29}$$

where $m_1(i)$ is the number of 1's in Y^i on the segment $I(i)$. Now, as $m_1(i)$ is binomially distributed, we further get the following average values over Y^i

$$|m_1(i) - 0.5m(i)|_{\rm av} \;\approx\; \frac{1}{\sqrt{2\pi}}\,\sqrt{m(i)} \tag{30}$$

$$\begin{aligned} |\Delta\hat{p}_i|_{\rm av} &\approx \frac{1}{2\sqrt{2\pi}}\,\frac{1}{\sqrt{m(i)}} \\ &\approx \frac{1}{2\sqrt{2\pi\sqrt{3}}}\,\frac{1}{\sqrt[4]{i-1}} \;\approx\; 0.1515\,\frac{1}{\sqrt[4]{i}}. \end{aligned} \tag{31}$$

Except maybe for the multiplicative constant, the approximation is very good for $i \geq 100$. Thus, as i increases, it turns out that $|\Delta\hat{p}_i|_{\rm av}$ decreases approximately like $0.1515/\sqrt[4]{i}$. The decrease is to be expected, because of a loss of synchronization between the original and the decimated sequence. However, it may be surprising that the decrease is very slow, so that the posterior probabilities remain significant even for relatively large values of i. For example, $|\Delta\hat{p}_i|_{\rm av}$ is approximately 0.01515 for $i = 10000$ and 0.01 for $i = 50000$.

The posterior probabilities of individual LFSR_2 bits computed according to Theorem 3 depend on both the output sequence and on the reconstructed LFSR_1 sequence. They are harder to analyze theoretically, but should be much more different from one half than the posterior probabilities of individual LFSR_1 bits, because the LFSR_1 sequence is assumed to be known. They can be used for reconstructing the unknown LFSR_2 sequence from a known segment of the output sequence and a segment of the reconstructed LFSR_1 sequence.

6 Combined Fast Correlation Attacks

It is assumed that the LFSR feedback polynomials and a sufficiently long segment of the keystream sequence, in the known-plaintext scenario, are known. The objective of cryptanalysis is to reconstruct the secret-key-dependent initial states of LFSR_1 and LFSR_2 by an algorithm whose complexity can be relatively small even for large LFSR lengths.

6.1 Basic Attack on LFSR_1

Let Y^n be a given segment of the keystream sequence and let X^n be the corresponding segment of the regularly clocked output sequence of LFSR_1 whose initial state is to be recovered. The basic attack on LFSR_1 consists of two stages.

In the first stage, compute the posterior probabilities of individual bits of X^n by using the probabilistic model in which the input strings are assumed to be purely random. This is achieved in $O(n\sqrt{n})$ time by applying Theorem 2 from Section 3.2. The obtained sequence of posterior probabilities, $(\hat{p}_i)_{i=1}^n$, is a soft-valued estimate of X^n. A hard estimate, $\bar{X}^n = (\bar{x}_i)_{i=1}^n$, of X^n can be obtained by applying the maximum posterior probability decision rule for individual bits, i.e., $\bar{x}_i = 1$ if $\hat{p}_i > 0.5$ and $\bar{x}_i = 0$ otherwise. Therefore

$$\Pr\{\bar{x}_i \neq x_i \mid Y^i\} \;=\; \min(\hat{p}_i, 1-\hat{p}_i). \tag{32}$$

The correlation coefficient between $\bar{x}_i$ and x_i, conditioned on Y^i, is then

$$c_i \;= 1 - 2\Pr\{\bar{x}_i \neq x_i \mid Y^i\} \;=\; |1-2\hat{p}_i|. \tag{33}$$

The analysis conducted in Section 5 shows that the expected value of c_i over Y^i slowly decreases approximately like $0.303/\sqrt[4]{i}$ as i increases. So, it remains to be significantly large even for relatively large i such as $i = 10000$.

In the second stage, X^n is reconstructed from $(\hat{p}_i)_{i=1}^n$ by using the LFSR_1 linear recursion. Equation (32) means that $\bar{X}^n$ can be modeled as a noisy output of a time-varying binary symmetric channel when X^n is applied to its input, where the errors are approximately independent. As X^n is a codeword of the corresponding (truncated cyclic) linear block code, the problem of reconstructing X^n is thus essentially a decoding problem. It can be solved by using parity-check based iterative probabilistic decoding algorithms for fast correlation attacks on regularly clocked LFSR's (e.g., see [11], [12], and [8]). The time-variant correlation coefficient should improve the performance of these attacks.

It is known that the complexity of fast correlation attacks on a regularly clocked LFSR and the required output string length n mainly depend on the magnitude of the correlation coefficient and on the degrees and numbers of low-weight polynomial multiples of the LFSR feedback polynomial (e.g., see [11], [13], [7], and [8]). Successful fast correlation attacks are reported in [8], for random feedback polynomials, and in [16], for low-weight feedback polynomials, for the correlation coefficients as small as 2/15 and 1/16, respectively. For the shrinking generator, according to Section 5, the expected value of the correlation coefficient

c_i is considerably different from zero even if i is relatively large. For example, this expected value is approximately equal to 1/10, 1/20, 1/35, and 1/50 for $i = 100$, 1000, 10000, and 50000, respectively.

Since the expected value of c_i slowly decreases as i increases, it is of interest to keep n reasonably small. To this end, the so-called parity checks with memory [10] (also see [7]) or the parity checks sharing a given number of bits in common [14] may be utilized. In conclusion, the second stage of the basic fast correlation attack on the shrinking generator may be successful for a large class of LFSR_1 feedback polynomials.

If an information set decoding (e.g., error-free sliding window) technique is applied at the end, then the reconstructed string $\hat{X}^n$ will satisfy the LFSR_1 recursion, but should be tested for correlation with $\bar{X}^n$. Alternatively, one may use the posterior probability (4) of blocks of LFSR_1 bits as a measure of correlation.

6.2 Iterative Attack on LFSR_1

The iterative probabilistic decoding algorithms in the second stage of the basic attack from Section 6.1 iteratively update the posterior probabilities of individual bits of X^n. Therefore, the basic attack can be (considerably) improved if the first stage of the attack is incorporated in iterations of the iterative probabilistic decoding algorithm chosen. For example, we propose an iterative attack whose first iteration coincides with the basic attack and every subsequent iteration consists of two stages. First, update the posterior probabilities of individual bits of X^n by Theorem 1 from Section 3.1 where the posterior probabilities from the preceding iteration are used as the prior probabilities. Second, update the posterior probabilities of individual bits of X^n by applying the iterative probabilistic decoding algorithm.

6.3 Composite Attack on LFSR_1

As the posterior probabability $\hat{p}_i$ slowly approaches one half as i increases, it makes sense to divide a longer keystream segment into subsegments of equal length, to compute the posterior probabilities for the subsegments, and then to combine these posterior probabilities appropriately.

To this end, consider m overlapping output subsegments $Y_{jn+1}^{jn+2n+\tau_j}$, $0 \le j \le m-1$, where $\tau_j \approx \sqrt{2(j+1)n}$, $0 \le j \le m-2$, and $\tau_{m-1} = 0$. Compute $2n+\tau_j$ posterior probabilities for the corresponding LFSR_1 segment $X_{i_j+1}^{i_j+2n+\tau_j}$, for each $0 \le j \le m-1$. Here, $i_0 = 0$ and for $j > 0$, i_j is unknown, but is expected to be around $2jn+1$ within an interval of length proportional to $\sqrt{2jn}$. So, a segment of $2mn$ posterior probabilities can be composed by guessing i_j, $1 \le j \le m-1$, and by taking the posterior probabilities more different from one half for the overlapping parts of the LFSR_1 subsegments. Additional τ_j bits for the j-th subsegment serve to fill in a possible gap between the j-th and $(j+1)$-th subsegments. As $\hat{p}_i$ slowly changes with i, the method is not sensitive to $m-1$ guesses of unknown positions i_j.

Finally, a fast correlation attack is run by using the composed segment of $2mn$ consecutive posterior probabilities. It has to be run for each of about $\sqrt{(m-1)!}(2n)^{(m-1)/2}$ guesses. For example, $n \leq 20000$ and $m \leq 5$ are realistic choices of the parameters.

6.4 Subsequence Attack on LFSR$_1$

Suppose that the posterior probabilities corresponding to a given keystream segment Y^n are not sufficiently different from one half, because the length n required for the success of fast correlation attacks explained above is too large. One can then compute the posterior probabilities for a number of subsequences of the keystream sequence obtained by discarding the initial segment of variable length until more significant posterior probabilities are obtained. This will improve the performance of the fast correlation attacks, but the length of the initial LFSR$_1$ segment has to be guessed. More precisely, if a segment $X_{j'}^{j'+n-1}$ of the LFSR$_1$ sequence is reconstructed from the output segment Y_j^{j+n-1}, one has to make $O(\sqrt{2j})$ guesses around the expected value $2j$ in order to find the unknown initial position j'. The number of tested subsequences is j/δ if one skips $\delta - 1$ output bits at a time. Testing can be simplified by searching for relatively unbalanced output subsequences instead of the significant posterior probabilities.

In particular, one can also search for about r_1, not necessarily consecutive, outstanding posterior probabilities (close to 1/4 or 3/4) and then apply an information set decoding algorithm to recover the LFSR$_1$ initial state, where the posterior probability (4) of blocks of LFSR$_1$ bits is used as a measure of correlation. The success of such an algorithm is independent of the LFSR$_1$ feedback polynomial, but, according to the information set decoding arguments, the achievable complexity cannot be smaller than about $2^{0.5\,r_1}$ corresponding steps. This improves the reduced complexity method [9] based on specific subsequences of the output sequence. Namely, as the class of usable subsequences is effectively enlarged, the required keystream segment length, around $2^{0.5\,r_1}$, can be considerably reduced. The expression given in [9] is approximative, whereas the accurate expression for the posterior probabilities is provided by Theorem 2. Moreover, the need for guessing the length of the initial LFSR$_1$ segment is overlooked in [9].

6.5 Reinitialization Attack on LFSR$_1$

Suppose that for resynchronization purposes the shrinking generator is reinitialized by bitwise addition of a reinitialization vector to the secret-key-controlled LFSR initial states, in view of the fact that the nonlinear next-state function prevents the resynchronization attack [2]. The posterior probabilities of individual LFSR$_1$ bits produced from the secret-key-controlled initial state can then be computed for different initialization vectors and all combined into values more different from one half, so that the corresponding fast correlation attack is easier.

6.6 Attack on LFSR_2

After reconstructing a candidate initial state of LFSR_1, the initial state of LFSR_2 can be recovered by computing the posterior probabilities of individual LFSR_2 bits by Theorem 3 from Section 4. More precisely, the posterior probabilities of individual bits of C^m are computed in $O(m^2\sqrt{m})$ time from given Y^m and reconstructed $\hat{X}^m$, $m \leq n$. Here, C^m is the corresponding segment of the regularly clocked output sequence of LFSR_2 whose initial state is to be recovered. As $\hat{X}^m$ is assumed to be known, the obtained posterior probabilities are much more distinguished from one half than in the case of LFSR_1. The reconstruction problem is then much easier and m can be much smaller than n. The posterior probabilities can be further enhanced by the reinitialization method described in Section 6.5. Accordingly, the initial state of LFSR_2 can be reconstructed by iterative probabilistic decoding algorithms in the same way as in the basic attack on LFSR_1 explained in Section 6.1. Moreover, as the posterior probabilities can be close to 0 or 1, simple information set decoding algorithms may also be applicable.

One should repeat the attack on LFSR_2 for several small phase shifts, positive or negative, of the reconstructed LFSR_1 sequence until the correct initial states of both LFSR's are reconstructed. Note that the number of solutions for the LFSR initial states is the number of 0's in a cycle of the LFSR_2 sequence preceding the first clock-control bit equal to 1 (see [15]).

7 Conclusions

The introduced probabilistic analysis of the shrinking generator shows that the irregularly clocked LFSR's, unlike a common belief in the open literature, may be vulnerable to fast correlation attacks. The analysis can be generalized to deal with arbitrary keystream generators based on clock-controlled LFSR's.

In order to reconstruct the initial state of the clock-controlled LFSR, LFSR_1, in the shrinking generator, the new idea is to compute the posterior probabilities of individual bits of the regularly clocked LFSR_1 sequence when conditioned on a given segment of the output sequence. Perhaps surprisingly, a theoretical analysis indicates that these probabilities can be significantly different from one half even for relatively long segments of the LFSR_1 sequence. Accordingly, the initial state of LFSR_1 may be recovered by a fast correlation attack, applicable to a regularly clocked LFSR, based on the computed posterior probabilities. It is known that such an attack can be successful for certain LFSR feedback polynomials. More sophisticated fast correlation attacks including the iterative attack, the composite attack, the subsequence attack, and the reinitialization attack are also proposed.

The initial state of the clock-control LFSR, LFSR_2, can be reconstructed in a similar way, but based on the computed posterior probabilities of individual bits of the regularly clocked LFSR_2 sequence when conditioned on a given segment of the output sequence and on a segment of the reconstructed LFSR_1 sequence. As these probabilities are more distinguished from one half, the corresponding fast correlation attack is easier.

Appendix

A Proof of Theorem 1

To prove (10), we start from (7). First, in view of (8), we get

$$\begin{aligned}\Pr\{Y^n \mid x_i = 1\} &= \sum_{e=0}^{n} \Pr\{Y^n, d(C^n) = e \mid x_i = 1\} \\ &= \sum_{e=0}^{n} \Pr\{Y_{n-e+1}^n, Y^{n-e}, d(C^n) = e \mid x_i = 1\} \\ &= \sum_{e=0}^{n} \Pr\{Y_{n-e+1}^n \mid Y^{n-e}, d(C^n) = e, x_i = 1\}\, P_i(e, n-e) \\ &= \sum_{e=0}^{n} 2^{-e}\, P_i(e, n-e). \end{aligned} \tag{34}$$

Namely, on the condition that $d(C^n) = e$, the string Y_{n-e+1}^n is obtained by decimating X_{n+1} according to C_{n+1}, where X_{n+1} and C_{n+1} are mutually independent and purely random even when conditioned on x_i and Y^{n-e}. Therefore, under the given conditions, Y_{n-e+1}^n is uniformly distributed. Similarly, in view of (9), we have

$$\Pr\{Y^n\} \;=\; \sum_{e=0}^{n} 2^{-e}\, P(e, n-e). \tag{35}$$

Consequently, (7) together with (34) and (35) result in (10).

As for the recursions, we only prove (11), whereas (12) is proved analogously. For $(e, s) \neq (0, 0)$, (8) results in

$$\begin{aligned} P_i(e,s) &= \Pr\{Y^s, d(C^{e+s}) = e \mid x_i = 1, c_{e+s} = 0\} \cdot \Pr\{c_{e+s} = 0 \mid x_i = 1\} \\ &\quad + \Pr\{Y^s, d(C^{e+s}) = e \mid x_i = 1, c_{e+s} = 1\} \cdot \Pr\{c_{e+s} = 1 \mid x_i = 1\} \\ &= \Pr\{Y^s, d(C^{e+s-1}) = e-1 \mid x_i = 1, c_{e+s} = 0\} \cdot \frac{1}{2} \\ &\quad + \Pr\{Y^s, d(C^{e+s-1}) = e \mid x_i = 1, c_{e+s} = 1\} \cdot \frac{1}{2}. \end{aligned} \tag{36}$$

Now, as $d(C^{e+s-1})$ is independent of c_{e+s}, and Y^s is independent of c_{e+s} on the condition that $d(C^{e+s-1}) = e-1$, we get

$$\begin{aligned} &\Pr\{Y^s, d(C^{e+s-1}) = e-1 \mid x_i = 1, c_{e+s} = 0\} \;= \\ &\qquad \Pr\{Y^s, d(C^{e+s-1}) = e-1 \mid x_i = 1\} \;=\; P_i(e-1, s). \end{aligned} \tag{37}$$

On the other hand, if $c_{e+s} = 1$ and $d(C^{e+s-1}) = e-1$, then $y_s = x_{e+s}$. Thus, we get

$$\begin{aligned}
&\Pr\{Y^s, d(C^{e+s-1}) = e \mid x_i = 1, c_{e+s} = 1\} \\
&\quad = \Pr\{x_{e+s} = y_s, Y^{s-1}, d(C^{e+s-1}) = e \mid x_i = 1, c_{e+s} = 1\} \\
&\quad = \Pr\{x_{e+s} = y_s \mid Y^{s-1}, d(C^{e+s-1}) = e, x_i = 1, c_{e+s} = 1\} \\
&\qquad \cdot \Pr\{Y^{s-1}, d(C^{e+s-1}) = e \mid x_i = 1, c_{e+s} = 1\} \qquad (38) \\
&\quad = \Pr\{x_{e+s} = y_s \mid x_i = 1\} \\
&\qquad \cdot \Pr\{Y^{s-1}, d(C^{e+s-1}) = e \mid x_i = 1\} \qquad (39) \\
&\quad = (\delta_{i,e+s} y_s + (1 - \delta_{i,e+s})(y_s p_{e+s} + (1 - y_s)(1 - p_{e+s}))) \cdot P_i(e, s-1). \\
&\qquad (40)
\end{aligned}$$

The first line of (39) follows from the first line of (38) because x_{e+s} is independent of C^{e+s} and, on the condition that $d(C^{e+s-1}) = e$, it is also independent of Y^{s-1}. In addition, as $d(C^{e+s-1})$ is independent of c_{e+s} and Y^{s-1} is independent of c_{e+s} on the condition that $d(C^{e+s-1}) = e$, the second line of (39) follows from the second line of (38).

Equation (11) directly follows from (36), (37), and (40). If $e = 0$, then the first term on the right-hand side of (11) is omitted, and if $s = 0$, then the second term on the right-hand side of (11) is omitted. The correct values of $P_i(1, 0)$ and $P_i(0, 1)$ are both obtained from the initial value $P_i(0, 0) = 1$.

B Proof of Theorem 3

The proof is essentially similar to the proof of Theorem 1, but should be conducted carefully. To prove (24), we start from (22). First, in view of (23), we get

$$\begin{aligned}
&\Pr\{Y^n \mid c_i = 1, X^n\} \\
&\qquad = \sum_{e=0}^{n} \Pr\{Y^n, d(C^n) = e \mid c_i = 1, X^n\} \\
&\qquad = \sum_{e=0}^{n} \Pr\{Y_{n-e+1}^n, Y^{n-e}, d(C^n) = e \mid c_i = 1, X^n\} \\
&\qquad = \sum_{e=0}^{n} \Pr\{Y_{n-e+1}^n \mid Y^{n-e}, d(C^n) = e, c_i = 1, X^n\} \, Q_i(e, n-e) \\
&\qquad = \sum_{e=0}^{n} 2^{-e} \, Q_i(e, n-e). \qquad (41)
\end{aligned}$$

Namely, on the condition that $d(C^n) = e$, the string Y_{n-e+1}^n is obtained by decimating X_{n+1} according to C_{n+1}, where X_{n+1} and C_{n+1} are mutually independent and purely random even when conditioned on c_i and Y^{n-e}. Therefore, under the given conditions, Y_{n-e+1}^n is uniformly distributed. Note that (3) is similarly derived from (1). Consequently, (22) together with (41) and (3) result in (24).

As for the recursions, we note that the proof of (2) is similar to the proof of (25) given below. For $(e,s) \neq (0,0)$, (23) results in

$$\begin{aligned} & Q_i(e,s) \\ &= \Pr\{Y^s, d(C^{e+s}) = e \mid c_i = 1, X^n, c_{e+s} = 0\} \cdot \Pr\{c_{e+s} = 0 \mid c_i = 1, X^n\} \\ &\quad + \Pr\{Y^s, d(C^{e+s}) = e \mid c_i = 1, X^n, c_{e+s} = 1\} \cdot \Pr\{c_{e+s} = 1 \mid c_i = 1, X^n\} \\ &= \Pr\{Y^s, d(C^{e+s-1}) = e-1 \mid c_i = 1, X^n, c_{e+s} = 0\} \cdot \frac{1}{2}\,(1 - \delta_{i,e+s}) \\ &\quad + \Pr\{Y^s, d(C^{e+s-1}) = e \mid c_i = 1, X^n, c_{e+s} = 1\} \cdot \frac{1}{2}\,(1 + \delta_{i,e+s}) \end{aligned} \tag{42}$$

where the conditional probability in the first term is computed only for $i \neq e+s$.

Now, as $d(C^{e+s-1})$ is independent of c_{e+s}, and Y^s is independent of c_{e+s} on the condition that $d(C^{e+s-1}) = e-1$, we get that for $i \neq e+s$

$$\begin{aligned} & \Pr\{Y^s, d(C^{e+s-1}) = e-1 \mid c_i = 1, X^n, c_{e+s} = 0\} \\ &\qquad = \Pr\{Y^s, d(C^{e+s-1}) = e-1 \mid c_i = 1, X^n\} \;=\; Q_i(e-1,s). \end{aligned} \tag{43}$$

On the other hand, if $c_{e+s} = 1$ and $d(C^{e+s-1}) = e-1$, then $y_s = x_{e+s}$. Thus, we get

$$\begin{aligned} & \Pr\{Y^s, d(C^{e+s-1}) = e \mid c_i = 1, X^n, c_{e+s} = 1\} \\ &\quad = \Pr\{x_{e+s} = y_s, Y^{s-1}, d(C^{e+s-1}) = e \mid c_i = 1, X^n, c_{e+s} = 1\} \\ &\quad = \Pr\{x_{e+s} = y_s \mid Y^{s-1}, d(C^{e+s-1}) = e, c_i = 1, X^n, c_{e+s} = 1\} \\ &\qquad \cdot \Pr\{Y^{s-1}, d(C^{e+s-1}) = e \mid c_i = 1, X^n, c_{e+s} = 1\} \qquad (44) \\ &\quad = \Pr\{x_{e+s} = y_s \mid x_{e+s}\} \\ &\qquad \cdot \Pr\{Y^{s-1}, d(C^{e+s-1}) = e \mid c_i = 1, X^n\} \qquad (45) \\ &\quad = \delta(x_{e+s}, y_s) \cdot Q_i(e, s-1). \qquad (46) \end{aligned}$$

The first line of (45) follows from the first line of (44) as x_{e+s} is contained in X^n. In addition, as $d(C^{e+s-1})$ is independent of c_{e+s} and Y^{s-1} is independent of c_{e+s} on the condition that $d(C^{e+s-1}) = e$, the second line of (45) follows from the second line of (44).

Equation (25) directly follows from (42), (43), and (46). If $e = 0$, then the first term on the right-hand side of (25) is omitted, and if $s = 0$, then the second term on the right-hand side of (25) is omitted. The correct values of $Q_i(1,0)$ and $Q_i(0,1)$ are both obtained from the initial value $Q_i(0,0) = 1$.

References

1. 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, 1993.
2. J. Daemen, R. Govaerts, and J. Vandewalle, "Resynchronization weakness in synchronous stream ciphers," Advances in Cryptology - EUROCRYPT '93, *Lecture Notes in Computer Science*, vol. 765, pp. 159-167, 1994.

3. J. Dj. Golić, "Intrinsic statistical weakness of keystream generators," Advances in Cryptology - ASIACRYPT '94, *Lecture Notes in Computer Science*, vol. 917, pp. 91-103, 1995.
4. J. Dj. Golić and L. O'Connor, "Embedding and probabilistic correlation attacks on clock-controlled shift registers," Advances in Cryptology - EUROCRYPT '94, *Lecture Notes in Computer Science*, vol. 950, pp. 230-243, 1995.
5. J. Dj. Golić, "Towards fast correlation attacks on irregularly clocked shift registers," Advances in Cryptology - EUROCRYPT '95, *Lecture Notes in Computer Science*, vol. 921, pp. 248-262, 1995.
6. J. Dj. Golić and R. Menicocci, "Edit probability correlation attack on the alternating step generator," Sequences and their Applications - SETA '98, *Discrete Mathematics and Theoretical Computer Science*, C. Ding, T. Helleseth, and H. Niederreiter eds., Springer-Verlag, pp. 213-227, 1999.
7. J. Dj. Golić, "Iterative probabilistic decoding and parity checks with memory," *Electronics Letters*, vol. 35(20), pp. 1721-1723, Sept. 1999.
8. J. Dj. Golić, M. Salmasizadeh, and E. Dawson, "Fast correlation attacks on the summation generator," *Journal of Cryptology*, vol. 13, pp. 245-262, 2000.
9. T. Johansson, "Reduced complexity correlation attacks on two clock-controlled generators," Advances in Cryptology - ASIACRYPT '98, *Lecture Notes in Computer Science*, vol. 1514, pp. 342-357, 1998.
10. T. Johansson and F. Jonnson, "Improved fast correlation attacks on stream ciphers via convolutional codes," Advances in Cryptology - EUROCRYPT '99, *Lecture Notes in Computer Science*, vol. 1592, pp. 347-362, 1999.
11. W. Meier and O. Staffelbach, "Fast correlation attacks on certain stream ciphers," *Journal of Cryptology*, vol. 1, pp. 159-176, 1989.
12. M. J. Mihaljević and J. Dj. Golić, "A comparison of cryptanalytic principles based on iterative error-correction," Advances in Cryptology - EUROCRYPT '91, *Lecture Notes in Computer Science*, vol. 547, pp. 527-531, 1991.
13. M. J. Mihaljević and J. Dj. Golić, "Convergence of a Bayesian iterative error-correction procedure on a noisy shift register sequence," Advances in Cryptology - EUROCRYPT '92, *Lecture Notes in Computer Science*, vol. 658, pp. 124-137, 1993.
14. M. J. Mihaljević, M. P. C. Fossorier, and H. Imai, "A low-complexity and high-performance algorithm for the fast correlation attack," Fast Software Encryption - New York 2000, *Lecture Notes in Computer Science*, vol. 1978, pp. 196-212, 2001.
15. L. Simpson, J. Dj. Golić, and E. Dawson, "A probabilistic correlation attack on the shrinking generator," Information Security and Privacy - Brisbane '98, *Lecture Notes in Computer Science*, vol. 1438, pp. 147-158, 1998.
16. L. Simpson, J. Dj. Golić, M. Salmasizadeh, and E. Dawson, "A fast correlation attack on multiplexer generators," *Information Processing Letters*, vol. 70, pp. 89-93, 1999.
17. K. Zeng, C. H. Yang, and T. R. N. Rao, "On the linear consistency test (LCT) in cryptanalysis with applications," Advances in Cryptology - CRYPTO '89, *Lecture Notes in Computer Science*, vol. 435, pp. 164-174, 1990.