J. Cryptology (2000) 13: 245–262
DOI: 10.1007/s001459910009

Journal of
CRYPTOLOGY

Fast Correlation Attacks on the Summation Generator[*]

Jovan Dj. Golić
School of Electrical Engineering, University of Belgrade,
Bulevar Revolucije 73, 10001 Belgrade, Yugoslavia
golic@galeb.etg.bg.ac.yu

Mahmoud Salmasizadeh
Electronic Research Centre, Sharif University of Technology,
P.O. Box 11365-8639, Tehran, Iran

Ed Dawson
Information Security Research Centre, Queensland University of Technology,
GPO Box 2434, Brisbane, Queensland 4001, Australia

Communicated by Othmar Staffelbach

Received 13 December 1996 and revised 7 October 1998

Abstract. The linear sequential circuit approximation method for combiners with memory is used to find mutually correlated linear transforms of the input and output sequences in the well-known summation generator with any number of inputs. It is shown that the determined correlation coefficient is large enough for applying a fast correlation attack to the output sequence to reconstruct the initial states of the input linear feedback shift registers. The proposed attack is based on iterative probabilistic decoding and appropriately generated low-weight parity-checks. The required output sequence length and the computational complexity are both derived. Successful experimental results for the summation generators with three and five inputs are obtained.

Key words. Summation generator, Correlation attacks, Linear approximations, Correlation coefficients, Parity-checks.

1. Introduction

A well-known type of keystream generators for stream cipher applications consists of a number of linear feedback shift registers (LFSRs) combined by a memoryless nonlinear function. It is shown in [15] and [14] that such structures may be vulnerable to divide-and-conquer correlation attacks based on the termwise correlation between the keystream

[*] Most of this work was carried out while the first two authors were with the Information Security Research Centre, Queensland University of Technology, Australia. Part of this work was done while the first author was on leave at the Isaac Newton Institute for Mathematical Sciences, Cambridge, England.

sequence and a subset of the LFSR sequences. More importantly, fast correlation attacks based on iterative probabilistic decoding are introduced in [9] (see also [1], [11], [12], and [3]). These attacks are successful if the correlation coefficient is large enough and if the involved feedback polynomials have sufficiently many low-weight polynomial multiples of moderately large degrees.

The use of combiners with memory to overcome the tradeoff [14] between the linear complexity and correlation immunity is suggested in [13]. It is shown that one can achieve the maximum-order correlation immunity, regardless of the linear complexity, with just one bit of memory. The *summation generator* proposed in [13] and [8] is such a nonlinear combiner with memory. For two inputs, it has only one bit of memory, and for n inputs it has $m = \lceil \log_2 n \rceil$ bits of memory.

The correlation properties of combiners with one bit of memory are investigated in [10]. For the summation generator with any number of inputs, the corresponding asymptotic correlation coefficient (both unconditional and conditioned on the output sequence) is determined in [16]. For a general binary combiner with m memory bits, the correlation properties are analyzed in [4]. It is shown that in such a combiner there exists a nonzero linear function (transform) of at most $m+1$ successive output bits that is correlated to a linear function of at most $m+1$ successive input bits. The linear sequential circuit approximation (LSCA) method [4] provides a feasible procedure for finding such pairs of linear functions with comparatively large correlation coefficients. The LSCA method consists in determining and solving a linear sequential circuit that approximates a binary combiner with memory.

It is shown in [4] that every combiner with memory can be rendered zero-order correlation immune by applying an appropriate linear transform to the output sequence. In this case the resulting correlation coefficient is reduced depending on the number of nonzero terms in the linear transform applied, but may still be large enough to perform a basic correlation attack [15] or even the fast correlation attack [9].

A divide-and-conquer attack on the summation generator is proposed in [2]. The required keystream sequence length is slightly larger than the sum of the input LFSR lengths, but the attack consists in a search over all possible initial states of all the LFSRs except for the longest one. Another attack, based on a specific, 2-adic complexity measure is introduced in [6]. The required keystream sequence length to predict the whole sequence is on average proportional to the sum of the LFSR periods and the computational complexity is roughly quadratic in this length.

The first objective of this paper is to apply the LSCA method to the summation generator with an arbitrary number of inputs and obtain all pairs of mutually correlated input and output linear functions/transforms with the maximum possible absolute value of the correlation coefficient. The second objective is to exploit this correlation weakness to mount a fast correlation attack [3] on the input LFSRs. For the attack to be successful, sufficiently many low-weight polynomial multiples of the least common multiple of the LFSR feedback polynomials have to be generated. For this purpose, a polynomial residue method (initiated in [9]) based on the birthday paradox is used. For the summation generators with three and five inputs, systematic successful experiments are conducted for the LFSR lengths chosen according to the computational power available.

Fast correlation attack techniques and the polynomial residue method are reviewed in Section 2 and the summation generator is defined in Section 3. Linear appoximations for the summation generator with any number of input LFSRs and the corresponding fast correlation attacks are theoretically investigated in Sections 4 and 5, respectively. The complexities of the proposed and known attacks are compared in Section 5. The experimental fast correlation attacks on the summation generators with three and five input LFSRs are presented in Section 6. Conclusions are given in Section 7.

2. Fast Correlation Attacks

The probabilistic model, a method for generating low-weight parity-checks, and an iterative error-correction algorithm used in fast correlation attacks are outlined in this section.

2.1. *Probabilistic Model*

The observed keystream sequence $z = \{z_i\}_{i=0}^{N-1}$ is modeled as the output sequence of a memoryless binary symmetric channel (BSC) with error probability p (corresponding to the known correlation coefficient $c = 1 - 2p$) when the unknown LFSR sequence $a = \{a_i\}_{i=0}^{N-1}$ is applied to its input. The LFSR feedback polynomial $f(x)$ of degree r is assumed to be known. The set of sequences a generated from all possible initial states is then a linear (N, r) code. The optimal decoding algorithm consists in finding the LFSR initial state giving rise to a codeword a lying at the minimum Hamming distance from the received codeword z. This is essentially the basic correlation attack, with computational complexity $O(2^r)$, proposed in [15]. The decoding error probability will be close to zero if $r/N < C$ where $C = 1 - H_2(p)$ is the capacity of the BSC. If c is small, then this condition reduces to $N > r\,O(1/c^2)$.

The objective of fast correlation attacks is to recover the original LFSR sequence without searching over all 2^r initial states. This can be achieved by using iterative probabilistic decoding procedures based on low-weight parity-checks.

2.2. *Parity-Checks*

A parity-check is any linear relationship satisfied by an LFSR sequence. It is known that the parity-checks correspond to polynomial multiples $h(x)$, $h(0) = 1$, of $f(x)$ (see [1]). Our objective is to obtain sufficiently many parity-check polynomials $h(x)$ of low weight (number of nonzero terms) and of as small degree as possible, because the maximum degree used determines the required keystream sequence length. Repeated squaring [9] of $f(x)$ is a simple weight-preserving technique that can be used if the weight of $f(x)$ is low.

To generate all $h(x)$ of weight at most $2k + 1$ and of degree at most M, $M \geq r$, we use the polynomial residue method [5] which is based on the birthday paradox method from [9]. First, in $O(M)$ time compute and store the residues of all the monomials $x^m \mod f(x)$, $1 \leq m \leq M$. Second, by bitwise summation, in $O(M^k)$ time compute and store the residues $x^{i_1} + \cdots + x^{i_k} \mod f(x)$ for all $\binom{M}{k}$ combinations $1 \leq i_1 < \cdots < i_k \leq M$. Third, by a fast sorting algorithm in $O(M^k \log_2 M^k)$ time sort these residues (as integers) and find all the matches of 0 (equal residues) and matches of 1 (binary sum of residues equal to 1). A match of 0 gives a polynomial multiple of even weight at most equal to $2k$, whereas a match of 1 gives a polynomial multiple of odd weight at most equal to $2k + 1$.

For a random $f(x)$, it is argued in [5] that the expected number of polynomial multiples of any given weight w, $w \geq 2$, is for large $M/(w - 1)$ given as

$$2^{-r} \binom{M}{w-1} \approx \frac{M^{w-1}}{(w-1)!\, 2^r}. \tag{1}$$

This implies that in order for a polynomial multiple of weight $2k + 1$ and degree at most M to exist, it is on average necessary and sufficient that $M \geq (2k)!^{1/(2k)} 2^{r/(2k)}$ (which is related to the birthday paradox). As a consequence, $M = O(2^{r/(2k)})$ yields that the required precomputation storage and time for finding all the parity-check polynomials of weight at most $2k + 1$ are $O(2^{r/2})$ and $O(r 2^{r/2})$, respectively.

Each polynomial multiple of weight w found can be used to form w parity-checks from the corresponding phase shifts. All the parity-checks obtained can then be tested for orthogonality (see [11] and [3]), so that some phases of some of the polynomials may be discarded.

2.3. *Iterative Error-Correction Algorithm*

We employ an iterative probabilistic parity-check-based decoding algorithm [11] with a modification given in [3]. The algorithm consists of several rounds, each composed of a number of iterations. For each of N observed keystream bits, a set of preferably orthogonal parity-checks is first determined. The algorithm starts with the observed keystream sequence $\{z_i\}_{i=0}^{N-1}$ and with $p < 0.5$ as the error probability for each bit. The keystream sequence is then iteratively modified to yield the reconstructed LFSR sequence.

In each iteration recalculate the parity-check values and compute the current error probabilities as the posterior probabilities of error given the previous error probabilities as the prior probabilities of error. Then complement all the bits with an error probability larger than one-half. If p is not too close to 0.5, then most of the error probabilities typically quickly converge to zero. The number of errors is thus reduced, but not necessarily to zero. In order to correct all the errors, the algorithm is repeated for several rounds by resetting all the error probabilities to p. At the end use a simple information set decoding technique which consists in searching for an error-free sliding window of r successive bits. In fact, we applied an improved algorithm with the so-called fast resetting and with the sliding window technique incorporated in rounds (see [3]).

3. Description of the Summation Generator

The summation generator [13], [8] is a binary nonlinear combiner with memory whose internal state variable, the carry, takes integer values from the set $[0, n - 1]$, where n is the number of inputs. The memory size in bits is thus $m = \lceil \log_2 n \rceil$.

Let $X_t = (x_{1,t}, \ldots, x_{n,t})$ and y_t denote the n input bits and the output bit at time t, respectively, and let S_t denote the carry at time t. For simplicity, we keep the same notation for the carry $S_t = \sum_{j=0}^{m-1} s_{j,t} 2^j$ and for the binary representation of the carry $S_t = (s_{0,t}, \ldots, s_{m-1,t})$. We also use the notation $S_t^{(j)} = s_{j,t}$, $0 \leq j \leq m - 1$, $S_t^{(0)}$

being the least significant bit of S_t. Then the output and the next-state functions of the summation generator are for $t \geq 0$ respectively defined by

$$y_t = \bigoplus_{i=1}^{n} x_{i,t} \oplus S_t^{(0)}, \tag{2}$$

$$S_{t+1} = \left\lfloor \left(\sum_{i=1}^{n} x_{i,t} + S_t \right) \Big/ 2 \right\rfloor, \tag{3}$$

with the modulo 2 summation in (2) and integer summation in (3). In other words, at time t, the input bits and the carry are summed as integers, the least significant bit of the sum is taken as the output bit, and the remaining part of the sum defines the next carry.

The input sequences, $x_i = \{x_{i,t}\}_{i=0}^{\infty}$, $1 \leq i \leq n$, are defined as the LFSR sequences typically generated from distinct primitive feedback polynomials, which are assumed to be known to the cryptanalyst. The LFSR initial states are controlled by the secret key, whereas the initial carry, S_0, is either fixed or is also controlled by the secret key.

The next-state function (3) is not balanced, that is, its output is not balanced (uniformly distributed) if its input is balanced. However, in the probabilistic model where the input sequences are regarded as purely random (that is, as mutually independent sequences of independent and uniformly distributed binary random variables), it is shown in [16] that (3) defines an ergodic Markov chain. Its stationary (asymptotic) probability distribution is given by (see [16] and [7])

$$p_s = \frac{1}{n!} \sum_{l=0}^{s} (-1)^l (s-l)^n \binom{n+1}{l}. \tag{4}$$

Here p_s denotes the probability that the carry is equal to $s-1$, $1 \leq s \leq n$.

The correlation coefficient between any two binary random variables a and b is defined as $c(a,b) = \Pr\{a = b\} - \Pr\{a \neq b\}$, and the correlation coefficient of a single binary variable a is defined as $c(a) = c(a,0)$. Assume that the random carry variable S has the probability distribution (4). Then the (asymptotic) correlation coefficient of the least significant bit $S^{(0)}$ depends on n and is given as

$$c_n(S^{(0)}) = \sum_{s=1}^{n} (-1)^{s+1} p_s. \tag{5}$$

It is shown in [16] that $c_n(S^{(0)}) = 0$ for even n and that for odd n, $c_n(S^{(0)})$ is different from zero and exponentially converges to zero with n. It turns out that for odd n, if n is not too big (e.g., $n \leq 9$), the correlation coefficient is large enough to apply successful fast correlation attacks.

For even n, however, it is shown in [16] that the asymptotic conditional correlation coefficient $c_n(S^{(0)})$, conditioned on a sufficiently long series of successive ones/zeros in the output sequence converges to $+/-\ c_{n-1}(S^{(0)})$, respectively, where $c_{n-1}(S^{(0)})$ is the unconditional correlation coefficient defined by (5). For $n = 2$, the asymptotic conditional correlation coefficient is thus equal to ±1, which is sufficient to mount successfully a (fast) conditional correlation attack [10].

Due to the binary summation in (2), the summation generator is maximum-order correlation immune [13]. That is, for any given initial carry, the output sequence is (statistically) independent of any proper subset of the input sequences assumed to be purely random. Consequently, any linear transform of the input sequences that is correlated to a linear transform of the output sequence must involve all the input sequences.

4. Linear Approximations and Correlation Coefficients

In this section the best linear approximations for the summation generator with an arbitrary number of inputs and the corresponding maximum correlation coefficients are determined.

It is shown in [16] that the asymptotic probability distribution (4) is symmetric, that is,

$$p_{n+1-s} = p_s, \qquad 1 \le s \le n. \tag{6}$$

As a consequence, the correlation coefficient (5) of the least significant carry bit $S^{(0)}$ vanishes for even n. It is also established in [16] that (5) can be put into the form

$$c_n(S^{(0)}) = 2^{-n}\left(\sigma_{\mathrm{odd}}(n+1)\sum_{l=0}^{\lfloor (n-1)/2\rfloor}(-1)^l\, p_{2l+1} + \sigma_{\mathrm{even}}(n+1)\sum_{l=1}^{\lfloor n/2\rfloor}(-1)^{l+1}\, p_{2l}\right), \tag{7}$$

where, for any positive integer ν,

$$\sigma_{\mathrm{odd}}(\nu) \;=\; \sum_{l=0}^{\lfloor (\nu-1)/2\rfloor}(-1)^l\binom{\nu}{2l+1}, \tag{8}$$

$$\sigma_{\mathrm{even}}(\nu) \;=\; \sum_{l=0}^{\lfloor \nu/2\rfloor}(-1)^l\binom{\nu}{2l}. \tag{9}$$

By using the identity $\sigma_{\mathrm{even}}(\nu) + \sigma_{\mathrm{odd}}(\nu)i = (1+i)^\nu = 2^{\nu/2}e^{i\nu\pi/4}$, where $i = \sqrt{-1}$ in the field of complex numbers, it is shown in [16] that for even ν, $\nu = 2k$,

$$\sigma_{\mathrm{even}}(2k), \sigma_{\mathrm{odd}}(2k) = \begin{cases} 2^k, 0 & \text{if} \quad k \equiv 0 \pmod 4, \\ 0, 2^k & \text{if} \quad k \equiv 1 \pmod 4, \\ -2^k, 0 & \text{if} \quad k \equiv 2 \pmod 4, \\ 0, -2^k & \text{if} \quad k \equiv 3 \pmod 4. \end{cases} \tag{10}$$

Analogously, for odd ν, $\nu = 2k-1$, we get

$$\sigma_{\mathrm{even}}(2k-1), \sigma_{\mathrm{odd}}(2k-1) = \begin{cases} 2^{k-1}, -2^{k-1} & \text{if} \quad k \equiv 0 \pmod 4, \\ 2^{k-1}, 2^{k-1} & \text{if} \quad k \equiv 1 \pmod 4, \\ -2^{k-1}, 2^{k-1} & \text{if} \quad k \equiv 2 \pmod 4, \\ -2^{k-1}, -2^{k-1} & \text{if} \quad k \equiv 3 \pmod 4. \end{cases} \tag{11}$$

For odd n, $n = 2k-1$, in view of (10), the correlation coefficient (7) reduces to

$$c_n(S^{(0)}) = 2^{-(n-1)/2}\begin{cases} \sum_{l=1}^{k-1}(-1)^{l+1}\, p_{2l} & \text{if} \quad k \equiv 0 \pmod 4, \\ \sum_{l=0}^{k-1}(-1)^l\, p_{2l+1} & \text{if} \quad k \equiv 1 \pmod 4, \\ -\sum_{l=1}^{k-1}(-1)^{l+1}\, p_{2l} & \text{if} \quad k \equiv 2 \pmod 4, \\ -\sum_{l=0}^{k-1}(-1)^l\, p_{2l+1} & \text{if} \quad k \equiv 3 \pmod 4. \end{cases} \tag{12}$$

It then follows [16] that $|c_n(S^{(0)})| < 2^{-(n-1)/2}$, and direct computation reveals that $c_n(S^{(0)})$ is different from zero (see [16] for odd $n \leq 11$).

For odd n, our main objectives here are to obtain all the linear functions of the input bits that are correlated to the output bit with the correlation coefficient $\pm c_n(S^{(0)})$ and also to examine whether this correlation coefficient has the maximum possible absolute value. For even n, we want to find all the linear functions of the input bits that are correlated to a linear function of the output bits with the correlation coefficient of the maximum possible absolute value which has to be determined too.

The LSCA method [4] can help us find such mutually correlated input and output linear functions for both even and odd n. The essence of this method is to find good linear approximations to the output boolean function and to the boolean components of the next-state function of a binary combiner with memory and to solve the resulting linear sequential circuit. Since the output function (2) is already linear, we have to find linear approximations to $S_t^{(0)}$ with nonzero correlation coefficients. For odd n, we can approximate $S_t^{(0)}$ as zero or one, but this does not exhaust all possibilities. For even n, we have to find other linear approximations to $S_t^{(0)}$ as a boolean function of X_{t-1} and S_{t-1} (see (3)).

It follows that

$$S_t^{(0)} = \left\lfloor \left(\sum_{i=1}^{n} x_{i,t-1} + S_{t-1}^{(0)} \right) \bigg/ 2 \right\rfloor^{(0)} \oplus S_{t-1}^{(1)}, \tag{13}$$

where, as before, $S_t = \sum_{j=0}^{m-1} S_t^{(j)} 2^j$, $m = \lceil \log_2 n \rceil$. It is assumed that the carry S_{t-1} (and hence S_t too) has the asymptotic probability distribution given by (4). In a simplified notation, we have to analyze the following boolean function:

$$z = \left\lfloor \left(\sum_{i=1}^{n} x_i + s_0 \right) \bigg/ 2 \right\rfloor^{(0)} \oplus s_1. \tag{14}$$

Here $X = (x_1, \ldots, x_n)$ is uniformly distributed and independent of (s_0, s_1), with the probability distribution p_{s_0,s_1} derived from (4) as $p_{0,0} = \sum_{l=0}^{\lfloor (n-1)/4 \rfloor} p_{4l+1}$, $p_{1,0} = \sum_{l=0}^{\lfloor (n-2)/4 \rfloor} p_{4l+2}$, $p_{0,1} = \sum_{l=0}^{\lfloor (n-3)/4 \rfloor} p_{4l+3}$, and $p_{1,1} = \sum_{l=0}^{\lfloor (n-4)/4 \rfloor} p_{4l+4}$.

Lemma 1. *Let $L(X)$ be any linear function of X and let $L(X) \oplus x_{i_1} \oplus x_{i_2}$ be a linear function such that x_{i_1} and x_{i_2} are distinct variables not appearing in $L(X)$. If the correlation coefficient between z and $L(X)$ is equal to c, then the correlation coefficient between z and $L(X) \oplus x_{i_1} \oplus x_{i_2}$ is equal to $-c$.*

Proof. Let $c_{a,b}$ denote the correlation coefficient between z and $L(X)$ when conditioned on any particular value (a, b) of (x_{i_1}, x_{i_2}). Then

$$c(z, L(X)) = \tfrac{1}{4}(c_{0,0} + c_{0,1} + c_{1,0} + c_{1,1}). \tag{15}$$

Since $L(X)$ is degenerate in both x_{i_1} and x_{i_2} and since (14) is symmetric with respect to X, we have that $c_{0,1} = c_{1,0}$. As for $c_{0,0}$ and $c_{1,1}$, if, for any X and (s_0, s_1) such that $(x_{i_1}, x_{i_2}) =$

$(0,0)$, only the value of (x_{i_1}, x_{i_2}) is changed into $(1,1)$, then the corresponding value of z is complemented. As a consequence, since $L(X)$ is degenerate in both x_{i_1} and x_{i_2}, we get that $c_{0,0} = -c_{1,1}$. Then (15) reduces to $c(z, L(X)) = c_{0,1}/2$.

On the other hand, let $c'_{a,b}$ denote the correlation coefficient between z and $L(X) \oplus x_{i_1} \oplus x_{i_2}$ when conditioned on any particular value (a,b) of (x_{i_1}, x_{i_2}). Then, analogously,

$$c(z, L(X) \oplus x_{i_1} \oplus x_{i_2}) = \tfrac{1}{4}(c'_{0,0} + c'_{0,1} + c'_{1,0} + c'_{1,1}), \tag{16}$$

where $c'_{0,0} = c_{0,0}$, $c'_{1,1} = c_{1,1}$, $c'_{0,1} = -c_{0,1}$, and $c'_{1,0} = -c_{1,0}$. Hence $c(z, L(X) \oplus x_{i_1} \oplus x_{i_2}) = -c_{0,1}/2$. □

Lemma 2. *Let* $\sigma_{\text{odd}}(n-1)$ *and* $\sigma_{\text{even}}(n-1)$ *be defined by* (8) *and* (9), *respectively. Then the correlation coefficient between* z *and* x_1 *is given as*

$$c_n(z, x_1) = 2^{-(n-1)} \left(\sigma_{\text{odd}}(n-1) \sum_{l=0}^{\lfloor (n-1)/2 \rfloor} (-1)^l p_{2l+1} + \sigma_{\text{even}}(n-1) \sum_{l=1}^{\lfloor n/2 \rfloor} (-1)^{l+1} p_{2l} \right). \tag{17}$$

Proof. Let $c_{s_0,s_1;x_1}$ denote the correlation coefficient between z and x_1 when conditioned on any particular values (s_0, s_1) and x_1, that is,

$$c_{s_0,s_1;x_1} = \Pr\{z = x_1 | s_0, s_1; x_1\} - \Pr\{z \neq x_1 | s_0, s_1; x_1\}. \tag{18}$$

Let c_{s_0,s_1} denote the correlation coefficient between z and x_1 when conditioned on any particular value (s_0, s_1), that is,

$$c_{s_0,s_1} = \tfrac{1}{2}(c_{s_0,s_1;0} + c_{s_0,s_1;1}). \tag{19}$$

Then

$$c_n(z, x_1) = \sum_{s_0,s_1} p_{s_0,s_1} c_{s_0,s_1}. \tag{20}$$

We first compute $c_{0,0}$. From (14) we directly get

$$\begin{aligned} \Pr\{z = 0 | 0,0;0\} &= \Pr\left\{ \left\lfloor \left(\sum_{i=2}^{n} x_i \right) \Big/ 2 \right\rfloor^{(0)} = 0 \right\} \\ &= \Pr\left\{ \sum_{i=2}^{n} x_i \equiv 0 \text{ or } 1 \pmod 4 \right\} \\ &= 2^{-(n-1)} \sum_{l=0}^{\infty} \left(\binom{n-1}{4l} + \binom{n-1}{4l+1} \right), \end{aligned} \tag{21}$$

where as usual $\binom{\nu}{\mu} = 0$ if $\mu > \nu$. Similarly, we have

$$\begin{aligned}\Pr\{z = 1|0, 0; 0\} &= \Pr\left\{\left\lfloor\left(\sum_{i=2}^{n} x_i\right) \Big/ 2\right\rfloor^{(0)} = 1\right\}\\ &= \Pr\left\{\sum_{i=2}^{n} x_i \equiv 2 \text{ or } 3 \pmod 4\right\}\\ &= 2^{-(n-1)} \sum_{l=0}^{\infty}\left(\binom{n-1}{4l+2} + \binom{n-1}{4l+3}\right). \end{aligned} \tag{22}$$

In view of (18), (21) and (22) imply

$$c_{0,0;0} = 2^{-(n-1)} \sum_{l=0}^{\infty}\left(\binom{n-1}{4l} + \binom{n-1}{4l+1} - \binom{n-1}{4l+2} - \binom{n-1}{4l+3}\right). \tag{23}$$

On the other hand, we similarly obtain

$$\Pr\{z = 1|0, 0; 1\} = 2^{-(n-1)} \sum_{l=0}^{\infty}\left(\binom{n-1}{4l+1} + \binom{n-1}{4l+2}\right), \tag{24}$$

$$\Pr\{z = 0|0, 0; 1\} = 2^{-(n-1)} \sum_{l=0}^{\infty}\left(\binom{n-1}{4l} + \binom{n-1}{4l+3}\right). \tag{25}$$

Hence

$$c_{0,0;1} = 2^{-(n-1)} \sum_{l=0}^{\infty}\left(-\binom{n-1}{4l} + \binom{n-1}{4l+1} + \binom{n-1}{4l+2} - \binom{n-1}{4l+3}\right). \tag{26}$$

Finally, (23) and (26) combined by (19) yield

$$c_{0,0} = 2^{-(n-1)} \sum_{l=0}^{\infty}\left(\binom{n-1}{4l+1} - \binom{n-1}{4l+3}\right) = 2^{-(n-1)}\sigma_{\text{odd}}(n-1). \tag{27}$$

The remaining three cases are treated in an analogous way and as a result we get

$$c_{1,0} = 2^{-(n-1)} \sum_{l=0}^{\infty}\left(\binom{n-1}{4l} - \binom{n-1}{4l+2}\right) = 2^{-(n-1)}\sigma_{\text{even}}(n-1), \tag{28}$$

$$c_{0,1} = -2^{-(n-1)}\sigma_{\text{odd}}(n-1), \tag{29}$$

$$c_{1,1} = -2^{-(n-1)}\sigma_{\text{even}}(n-1). \tag{30}$$

According to (20), (27)–(30) finally yield

$$c_n(z, x_1) = 2^{-(n-1)}\sigma_{\text{odd}}(n-1)(p_{0,0} - p_{0,1}) + 2^{-(n-1)}\sigma_{\text{even}}(n-1)(p_{1,0} - p_{1,1}), \tag{31}$$

which is equivalent to (17). □

Lemma 3. *For odd n, $c_n(z, x_1) = 0$, and for even n, $n = 2k$,*

$$c_n(z, x_1) = 2^{-n/2} \begin{cases} -\sum_{l=0}^{k-1}(-1)^l p_{2l+1} + \sum_{l=1}^{k}(-1)^{l+1} p_{2l} & \text{if } k \equiv 0 \pmod 4, \\ \sum_{l=0}^{k-1}(-1)^l p_{2l+1} + \sum_{l=1}^{k}(-1)^{l+1} p_{2l} & \text{if } k \equiv 1 \pmod 4, \\ \sum_{l=0}^{k-1}(-1)^l p_{2l+1} - \sum_{l=1}^{k}(-1)^{l+1} p_{2l} & \text{if } k \equiv 2 \pmod 4, \\ -\sum_{l=0}^{k-1}(-1)^l p_{2l+1} - \sum_{l=1}^{k}(-1)^{l+1} p_{2l} & \text{if } k \equiv 3 \pmod 4. \end{cases} \tag{32}$$

Proof. For even n, $n = 2k$, (32) is a direct consequence of (17) and (11). For odd n, $n = 2k + 1$, (17) and (10) result in $c_n(z, x_1)$ being given by the right-hand side of (12) with a difference that $n = 2k + 1$ instead of $n = 2k - 1$. The following concise form of (12),

$$c_n(z, x_1) = 2^{-(n-1)/2} \begin{cases} \pm \sum_{l=1}^{k}(-1)^{l+1} p_{2l} & \text{if } k \equiv 0 \pmod 2, \\ \pm \sum_{l=0}^{k}(-1)^l p_{2l+1} & \text{if } k \equiv 1 \pmod 2, \end{cases} \tag{33}$$

shows that this difference is essential. Namely, for both even and odd k, the symmetry equation (6) forces (33) to be equal to zero. □

Let $c(n)$ be defined as $c_n(S^{(0)})$, by (12), if n is odd and as $c_n(z, x_1)$, by (32), if n is even. Let also $c_{\max}(n) = |c(n)|$. Then, in view of (2), Lemmas 1–3 result in the following theorem, which completely specifies the correlation between the current output bit and linear functions of the current and preceding input bits.

Theorem 1. *For any time $t \geq 1$, assume that the current and the preceding inputs to the summation generator are mutually independent and uniformly distributed and that the preceding carry has the asymptotic probability distribution* (4).

If the number n of binary inputs is odd, then the correlation coefficient between the current output bit and the binary sum of the current input bits and any number μ, $0 \leq \mu \leq n$, of the preceding input bits is equal to $c(n)$ if $\mu \equiv 0 \pmod 4$, to $-c(n)$ if $\mu \equiv 2 \pmod 4$, and to zero if μ is odd.

If the number n of binary inputs is even, then the correlation coefficient between the current output bit and the binary sum of the current input bits and any number μ, $0 \leq \mu \leq n$, of the preceding input bits is equal to $c(n)$ if $\mu \equiv 1 \pmod 4$, to $-c(n)$ if $\mu \equiv 3 \pmod 4$, and to zero if μ is even.

For each $1 \leq n \leq 10$, the computed value of the maximum correlation coefficient $c_{\max}(n)$, a representative input linear function with the minimum number of terms, and the output affine function to be used in fast correlation attacks are displayed in Table 1.

We conjecture that Theorem 1 identifies all input linear functions that are correlated to the current output bit with the correlation coefficient having the maximum absolute value. This is supported by the experimental fast correlation attacks on the summation generators with three and five inputs which always converged to a linear (or affine) transform of the input sequences determined by Theorem 1.

Another argument for this conjecture is provided by the LSCA method itself. Namely, to obtain nonzero correlation coefficients between the current output bit and linear functions of more than two successive inputs, we should consider more than just a single

Table 1. Maximum correlation and best linear approximation.

n	$c_{\max}(n)$	Best linear approximation	Output
1	1.00000	$x_{1,t}$	z_t
2	$\frac{1}{2} = 0.50000$	$\bigoplus_{i=1}^{2} x_{i,t} + x_{1,t-1}$	z_t
3	$\frac{1}{3} \approx 0.33333$	$\bigoplus_{i=1}^{3} x_{i,t}$	$\bar{z}_t$
4	$\frac{5}{24} \approx 0.20833$	$\bigoplus_{i=1}^{4} x_{i,t} + x_{1,t-1}$	$\bar{z}_t$
5	$\frac{2}{15} \approx 0.13333$	$\bigoplus_{i=1}^{5} x_{i,t}$	z_t
6	$\frac{61}{6!} \approx 0.08472$	$\bigoplus_{i=1}^{6} x_{i,t} + x_{1,t-1}$	z_t
7	$\frac{272}{7!} \approx 0.05396$	$\bigoplus_{i=1}^{7} x_{i,t}$	$\bar{z}_t$
8	$\frac{1385}{8!} \approx 0.03435$	$\bigoplus_{i=1}^{8} x_{i,t} + x_{1,t-1}$	$\bar{z}_t$
9	$\frac{7936}{9!} \approx 0.02186$	$\bigoplus_{i=1}^{9} x_{i,t}$	z_t
10	$\frac{50521}{10!} \approx 0.01392$	$\bigoplus_{i=1}^{10} x_{i,t} + x_{1,t-1}$	z_t

round of the next-state function (3). Due to multiple linear approximation to the internal state bits required, the overall correlation coefficient is expected to decrease.

The conjecture is also confirmed by computing the correlation coefficients between all the linear functions of at most three successive outputs and inputs for $n = 3$ and $n = 5$. In both cases, the maximum correlation coefficient found is $c_{\max}(n)$. For three successive inputs and the current output, the correlation coefficient takes intermediate values and the largest ones computed are $\pm\frac{1}{8}$ and $\pm\frac{7}{128}$ for $n = 3$ and $n = 5$, respectively. More precisely, the number of such input linear functions with the correlation coefficient equal to $\frac{1}{8}$, $-\frac{1}{8}$, $\frac{7}{128}$, and $-\frac{7}{128}$ is 9, 3, 30, and 30, respectively. In addition, for $n = 3$, we also computed the correlation coefficients between the current output bit and linear functions of four successive inputs (effectively) and the largest one obtained is $\pm\frac{1}{16}$.

From (12) and (32) it follows that $c_{\max}(n) < 2^{-(n-1)/2}$ for odd n and $c_{\max}(n) < 2^{-n/2}$ for even n, respectively. It turns out that the values of $c_{\max}(n)$ are not much smaller than what can be obtained by a random memoryless combiner with n inputs.

5. Fast Correlation Attacks on Summation Generator

In this section a theoretical analysis of fast correlation attacks on the summation generator is presented. The general case of an arbitrary number of inputs and parity-check polynomials of an arbitrary weight is considered.

According to Section 4, the fast correlation attack based on iterative error-correction decoding uses the output sequence or the binary complement of the output sequence. The LFSR sequence to be reconstructed is a linear transform of the input sequences corresponding to one of the input linear functions with the maximum absolute value of the correlation coefficient, $c_{\max}(n)$. Let L^+ and L^- denote the numbers of input linear functions with the correlation coefficient $c_{\max}(n)$ and $-c_{\max}(n)$ to the current output bit, respectively. To minimize the number of possible input linear transforms, the attack is run on the output sequence if $L^+ \leq L^-$ and on its binary complement if $L^+ > L^-$.

The error probability for the associated BSC channel is $p = (1 - c_{\max}(n))/2$. In view of maximum-order correlation immunity, the corresponding LFSR feedback polynomial ($f(x)$ of degree r) is the least common multiple of the individual input LFSR feedback polynomials. If these polynomials are distinct and primitive, then $f(x)$ is their product. The parity-checks to be used in the attack can be obtained by the polynomial residue method described in Section 2.2.

If odd-weight parity-checks are predominant, then the iterative error-correction algorithm, if successful, is expected to converge randomly to a linear transform of the input sequences with the correlation coefficient $c_{\max}(n)$. If even-weight parity-checks are predominant, then the algorithm may also converge to the binary complement of a linear transform of the input sequences with the correlation coefficient $-c_{\max}(n)$. This is because every even-weight parity-check polynomial for the feedback polynomial $f(x)$ contains $1 + x$ as a factor and is thus also a parity-check polynomial for the feedback polynomial $(1 + x)f(x)$, which corresponds to the bitwise binary complement of any LFSR sequence satisfying $f(x)$. The experiments reported in the next section confirm this behavior.

A unique solution for the unknown LFSR initial states consistent with the keystream sequence is obtained as follows. For any assumed input linear/affine transform, all the LFSR initial states are recovered by solving the corresponding linear equations and are then tested for consistency with the given keystream sequence. If the initial carry is secret key controlled, then its correct value has to be guessed out of n possible values.

The computational complexity of the iterative error-correction algorithm is proportional to the keystream sequence length and to the number of parity-checks per bit used in the attack. The required keystream sequence length is proportional to the maximum degree of the parity-check polynomials used which is $O(2^{r/(w-1)})$ for average LFSR feedback polynomials, provided that the parity-check weight w is fixed. Since the required number of the parity-checks depends only on w and on the correlation coefficient (see (36) below), both the keystream sequence length and the computational complexity increase as $O(2^{r/(w-1)})$ if w is fixed.

Our objective now is to examine the case when r is large and w is varied. Let $J(w)$ and $J_{\max}(w)$ respectively denote the average and the maximum numbers of the parity-checks per bit needed for a successful fast correlation attack on a given summation generator and let M be the maximum degree of the associated parity-check polynomials. The required keystream sequence length is typically $N = \nu M$, where ν is a relatively small constant (e.g., $1 < \nu \leq 10$). The average required computational complexity per one round of the iterative error-correction algorithm is about $C = N(w-1)J(w)$ (in appropriate units), and the number of rounds is roughly independent of w. According to (1), since the required number of the parity-check polynomials is $J_{\max}(w)/w$, one may on average expect that

$$M \approx \left(\frac{(w-1)!}{w}\right)^{1/(w-1)} J_{\max}(w)^{1/(w-1)}\, 2^{r/(w-1)}, \tag{34}$$

where the orthogonality condition is for simplicity disregarded. If ν is not close to one (e.g., $\nu \approx 10$), then $J_{\max}(w) \approx J(w)$, so that

$$C \approx \nu(w-1)\left(\frac{(w-1)!}{w}\right)^{1/(w-1)} J(w)^{w/(w-1)}\, 2^{r/(w-1)}. \tag{35}$$

It remains to assess $J(w)$. Let $c = c_{\max}(n)$. Under the assumption that the parity-checks are orthogonal and that $J_{\max}(w) \approx J(w)$, the convergence condition [12] indicates that the iterative error-correction algorithm will on average be successful if

$$\left(\frac{1+c^{w-1}}{1-c^{w-1}}\right)^{J(w)} > \frac{1+c}{1-c}, \tag{36}$$

which for small c can be well approximated as

$$J(w) > \frac{1}{c^{w-2}}. \tag{37}$$

Note that the number of the parity-checks needed for a successful fast correlation attack is in practice larger than the value predicted by (36), because in the experiments the parity-checks are not necessarily orthogonal, the errors are not independent as in the BSC model, the constant ν is close to one, and the best linear approximation is not unique. However, apart from a multiplicative constant, (37) seems to be a good approximation as far as the dependence on w is concerned.

Consequently, (35) can be reduced to

$$C \approx \nu(w-1)\left(\frac{(w-1)!}{w}\right)^{1/(w-1)} c^{-w(w-2)/(w-1)}\, 2^{r/(w-1)}. \tag{38}$$

So, the computational complexity is given as the product of three factors, as functions of w: the first increases roughly as $\nu(w-1)^2/e$, the second exponentially increases as $c^{-(w-1)+1/(w-1)}$, and the third decreases as $2^{r/(w-1)}$. A similar expression holds for the keystream sequence length $N = \nu M$ with a difference that the first factor is linear in $w-1$ and that the second factor, $c^{-1+1/(w-1)}$, is roughly constant. Therefore, N is predominantly determined by the third factor $2^{r/(w-1)}$. Unlike N, C is predominantly determined by the product of the last two factors, $c^{-(w-1)+1/(w-1)}\, 2^{r/(w-1)}$, which initially decreases with w, then achieves its minimum value $2^{2\sqrt{(r-\log_2 c^{-1})\log_2 c^{-1}}}$ for $w_{\mathrm{opt}} \approx \sqrt{r/\log_2 c^{-1} - 1} + 1$, and increases with further increase of w. Accordingly, we get

$$C_{\min} \approx \frac{\nu}{e}\, 2^{2\sqrt{(r-\log_2 c^{-1})\log_2 c^{-1}} + \log_2 2\sqrt{r/\log_2 c^{-1} - 1}} \tag{39}$$

and the corresponding value of N is given as

$$N \approx \frac{\nu}{ec}\, 2^{\sqrt{(r-\log_2 c^{-1})\log_2 c^{-1}}+\log_2\sqrt{r/\log_2 c^{-1}-1}}. \tag{40}$$

To simplify the comparisons given below, we use the approximations

$$C_{\min} \approx 2^{2\sqrt{r\log_2 c^{-1}}} \tag{41}$$

and $N \approx 2^{\sqrt{r\log_2 c^{-1}}}$.

Assuming that the LFSR lengths are approximately equal, the minimum computational complexity (41) can be compared with the computational complexities of the divide-and-conquer attack [2], $2^{r(n-1)/n}$, and of the 2-adic complexity attack [6], $2^{2(r/n+\log_2 n)}$ (neglecting the multiplicative constants). Thus $C_{\min} < 2^{2r/n}$ if $r > n^2 \log_2 c_{\max}(n)^{-1}$.

Finally, we compare the keystream sequence length and the computational complexity required for the successful fast correlation attack with those required for the successful basic correlation attack based on the method [4] which reduces the summation generator to a zero-order correlation immune combiner. Namely, choose the shortest of the LFSRs, make the product, $\hat{f}(x)$, of the distinct feedback polynomials of the remaining LFSRs, and find a polynomial multiple $h(x)$ of $\hat{f}(x)$ of low weight w. Then apply the linear transform defined by $h(x)$ to the output sequence. According to Theorem 1, the linearly transformed output sequence is bitwise correlated to the same linear transform of the chosen LFSR sequence or of its binary derivative. The correlation coefficient can be well approximated as $c_{\max}(n)^w$ or $-c_{\max}(n)^w$, assuming that the correlation noise is memoryless.

The correlation attack then consists of guessing the initial state of the chosen LFSR and of estimating the bitwise correlation coefficient between the linearly transformed LFSR and output sequences. The guessed initial state is assumed as correct if the estimated correlation coefficient is consistent with $\pm c_{\max}(n)$. The required linearly transformed output sequence length is $r_i\, O(c_{\max}(n)^{-2w})$, where r_i is the chosen LFSR length.

Assume for simplicity that the LFSR lengths are equal and let $c = c_{\max}(n)$. The required keystream sequence length can then be approximated as

$$N \approx 2^{2w\log_2 c^{-1}+\log_2(r/n)} + 2^{r(n-1)/(n(w-1))}. \tag{42}$$

The first and the second terms stand for the length needed for testing the correlation and for the expected degree of the parity-check polynomial $h(x)$. The required computational complexity is $C \approx w\, 2^{r/n}\, 2^{2w\log_2 c^{-1}+\log_2(r/n)}$ and the required storage space for computing the input linear transform is $S \approx 2^{r(n-1)/(n(w-1))}$.

As w increases, C increases and S decreases and there is an optimal value w_{opt} minimizing the keystream sequence length N. $N_{\min}$ is approximately achieved if the two terms in (42) are equal. That is, $w_{\text{opt}} \approx \sqrt{(n-1)r/(2n\log_2 c^{-1})} + 1$ and

$$N_{\min} \approx 2^{\sqrt{2r(n-1)\log_2 c^{-1}/n}}\,(1+r/n), \tag{43}$$

$$C_{\text{opt}} \approx 2^{\sqrt{2r(n-1)\log_2 c^{-1}/n}+r/n}. \tag{44}$$

Note that the minimum computational complexity (41) is smaller than C_{opt} if

$$r > n^2 \log_2 c_{\max}(n)^{-1} \left(2 - \sqrt{2(n-1)/n}\right)^2 . \tag{45}$$

6. Experimental Results

The objective of this section is to examine experimentally the vulnerability of the summation generators with three and five inputs to fast correlation attacks.

6.1. *Three Inputs*

The summation generator with three inputs is a binary nonlinear combiner with two bits of memory. According to Theorem 1, $c_{\max}(3) = \frac{1}{3}$, which is large enough to apply the fast correlation attack. The current output bit is correlated with the correlation coefficient $-\frac{1}{3}$ to the binary sum of the three current input bits, and with the correlation coefficient $\frac{1}{3}$ to the binary sum of the three current and any two preceding input bits (there are three such linear functions). Consequently, the fast correlation attack is run on the binary complement of the output sequence.

Experiments were conducted on four summation generators. The tap settings $Taps_i$ for LFSR$_i$, $i = 1, 2, 3$, the degree r, and the weight w_{p} of the resulting product feedback polynomial are shown in Table 2.

In each case, the attack was performed for 20 randomly chosen initial contents of the LFSRs for two sets of (not necessarily orthogonal) parity-checks: one, I, with predominant weight 5 and the other, II, with predominant weight 7 or 6 (depending on r). The parity-check sets were obtained by the method described in Section 2.2 for $k = 2$ and $k = 3$, respectively. For each out of the 20 initial contents, the number of the parity-checks used and the keystream sequence length were both minimized.

The weight w, the average number K_{av}, the maximum number $K_{\max}$, and the maximum degree $M_{\max}$ of the parity-check polynomials used as well as the average value N_{av} and the standard deviation $\sigma(N)$ of the keystream sequence length in successful experiments are all shown in Tables 3 and 4, for the parity-check sets I and II, respectively. For comparison, the keystream sequence length obtained by theory is also shown in Tables 3 and 4 (i.e., $N_{\text{th}} = \nu M$, where M is given by (34) and $J(w)$ determined by (36), assuming that $\nu = 2$ and that w is the predominant weight).

In the case $r = 30$ shown in Table 4, where even parity-check weight, 6, is predominant, the algorithm converged randomly to one of the four input affine transforms identified above. In other cases, where the predominant parity-check weight is odd, it converged to the bitwise sum of the input LFSR sequences.

6.2. *Five Inputs*

The summation generator with five inputs is a binary nonlinear combiner with three bits of memory. According to Theorem 1, $c_{\max}(5) = \frac{2}{15}$, which is, although smaller than $\frac{1}{3}$, also large enough to apply the fast correlation attack. The current output bit is

Table 2. Input LFSRs and product feedback polynomial, $n = 3$.

$Taps_1$	$Taps_2$	$Taps_3$	r	w_p
0, 4, 9	0, 3, 10	0, 2, 11	30	17
0, 3, 9	0, 2, 10	0, 3, 15	34	19
0, 3, 10	0, 2, 11	0, 3, 17	38	19
0, 1, 3, 4, 13	0, 1, 15	0, 2, 3, 5, 16	44	17

Table 3. Parity-check polynomials, set I, and keystream sequence length.

r	w	K_{av}	K_{max}	M_{max}	N_{av}	$\sigma(N)$	N_{th}
30	4	1	1	838	$2^{10.7}$	$2^{8.5}$	$2^{10.3}$
	5	118	169	1, 495			
34	5	137	170	2,877	$2^{11.4}$	$2^{8.7}$	$2^{11.3}$
38	5	79	79	4,990	$2^{12.5}$	$2^{9.5}$	$2^{12.3}$
44	5	127	135	14,954	$2^{13.9}$	$2^{9.6}$	$2^{13.8}$

Table 4. Parity-check polynomials, set II, and keystream sequence length.

r	w	K_{av}	K_{max}	M_{max}	N_{av}	$\sigma(N)$	N_{th}
30	5	4	6	559	$2^{9.5}$	$2^{7.7}$	$2^{8.1}$
	6	425	518	585			
	7	282	282	250			
34	5	13	152	800	$2^{9.9}$	$2^{9.0}$	$2^{9.1}$
	6	146	240	849			
	7	5,360	5,945	650			
38	6	19	19	830	$2^{10.7}$	$2^{9.4}$	$2^{9.8}$
	7	5,360	5,945	650			
44	7	2,600	2,600	1,800	$2^{11.4}$	$2^{9.7}$	$2^{10.8}$

Table 5. Input LFSRs and product feedback polynomial, $n = 5$.

$Taps_1$	$Taps_2$	$Taps_3$	$Taps_4$	$Taps_5$	r	w_p
0, 1, 4	0, 2, 5	0, 1, 6	0, 1, 7	0, 2, 3, 4, 8	30	13
0, 1, 6	0, 1, 7	0, 2, 3, 4, 8	0, 4, 9	0, 3, 10	40	21

Table 6. Parity-check polynomials and keystream sequence length.

r	w	K_{av}	K_{max}	M_{max}	N_{av}	$\sigma(N)$	N_{th}
30	5	8,338	9,272	3,915	$2^{12.1}$	$2^{10.4}$	$2^{11.5}$
	6	598	598	604			
	7	130	130	223			
40	5	14,800	14,800	24,968	$2^{14.9}$	$2^{11.8}$	$2^{14.0}$

correlated to six and ten linear functions of the corresponding two successive inputs with the correlation coefficient equal to $\frac{2}{15}$ and $-\frac{2}{15}$, respectively. Any of the six functions is the binary sum of the five current and zero or four preceding input bits, whereas any of the ten functions is the binary sum of the five current and two preceding input bits. Consequently, the fast correlation attack is run on the output sequence.

The fast correlation attack is conducted in the same way as in the case of three inputs except that there are more multiple linear/affine approximations to be checked in the final stage. Experiments were conducted on two summation generators. The tap settings $Taps_i$ for LFSR$_i$, $1 \leq i \leq 5$, the degree r, and the weight w_{p} of the resulting product feedback polynomial are shown in Table 5.

In each case, the attack was performed for 20 randomly chosen initial contents of the LFSRs by using (not necessarily orthogonal) parity-checks of predominant weight 5. For each out of the 20 initial contents, the number of the parity-checks used and the keystream sequence length were both minimized. The results obtained by experiments and by theory are shown in Table 6. In all the cases, the predominant parity-check weight, 5, is odd and the algorithm converged randomly to one of the six input linear transforms identified above.

7. Conclusions

It is shown that the summation generators with three, five, and any moderately large number n of input LFSRs may be vulnerable to fast correlation attacks based on iterative probabilistic decoding. Mutually correlated linear transforms of the input and output sequences are identified by the linear sequential circuit approximation method using the known asymptotic probability distribution of the carry. The underlying maximum correlation coefficient, $c_{\max}(n)$, is derived for any n. For success, sufficiently many low-weight parity-check polynomials can be generated by a polynomial residue method.

For random feedback polynomials with the least common multiple of degree r and any fixed parity-check weight w, the average required keystream sequence length and the computational complexity are both $O(2^{r/(w-1)})$, whereas the precomputation storage and time for finding the parity-check polynomials are $O(2^{r/2})$ and $O(r2^{r/2})$, respectively. However, primitive feedback polynomials may exist for which these figures are much smaller. When w is varied, it is shown that the minimum computational complexity can be approximated as $C_{\min} \approx 2^{2\sqrt{r \log_2 c_{\max}(n)^{-1}}}$ and is achieved if $w \approx \sqrt{r/\log_2 c_{\max}(n)^{-1}} + 1$. The corresponding keystream sequence length is then $N \approx 2^{\sqrt{r \log_2 c_{\max}(n)^{-1}}}$.

Successful experimental results are systematically obtained for the summation generator with three inputs and various r by using the polynomial multiples of weight 5 and of weight 6 and 7 combined, respectively, and, also, for the summation generator with five inputs by using the polynomial multiples of predominant weight 5.

Acknowledgments

The authors wish to acknowledge the contribution of Abdollah Khodkar in assisting with the computer programming needed to obtain the experimental results. The authors are thankful to two anonymous referees for their helpful comments.

References

[1] V. Chepyzhov and B. Smeets, On a fast correlation attack on stream ciphers, *Advances in Cryptology - Eurocrypt* '91, Lecture Notes in Computer Science, vol. 547, Springer-Verlag, Berlin, 1991, pp. 176–185.

[2] E. Dawson and A. Clark, Divide and conquer attacks on certain classes of stream ciphers, *Cryptologia*, vol. 18(1), 1994, pp. 25–40.

[3] J. Dj. Golić, M. Salmasizadeh, A. Clark, A. Khodkar, and E. Dawson, Discrete optimisation and fast correlation attacks, *Cryptography*: *Policy and Algorithms - Brisbane* '95, Lecture Notes in Computer Science, vol. 1029, Springer-Verlag, Berlin, 1996, pp. 186–200.

[4] J. Dj. Golić, Correlation properties of a general binary combiner with memory, *Journal of Cryptology*, vol. 9(2), 1996, pp. 111–126.

[5] J. Dj. Golić, Computation of low-weight parity-check polynomials, *Electronics Letters*, vol. 32(21), Oct. 1996, pp. 1981–1982.

[6] A. Klapper and M. Goresky, Cryptanalysis based on 2-adic rational approximation, *Advances in Cryptology - Crypto* '95, Lecture Notes in Computer Science, vol. 963, Springer-Verlag, Berlin, 1995, pp. 262–273.

[7] S. Lloyd and C. Mitchel, Calculating some eigenvectors, Unpublished manuscript, 1990.

[8] J. L. Massey and R. A. Rueppel, Method of, and apparatus for, transforming a digital sequence into an encoded form, U.S. Patent No. 4,797,922, 1989.

[9] W. Meier and O. Staffelbach, Fast correlation attacks on certain stream ciphers, *Journal of Cryptology*, vol. 1(3), 1989, pp. 159–176.

[10] W. Meier and O. Staffelbach, Correlation properties of combiners with memory in stream ciphers, *Journal of Cryptology*, vol. 5(1), 1992, pp. 67–86.

[11] M. 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, Springer-Verlag, Berlin, 1991, pp. 527–531.

[12] M. 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, Springer-Verlag, Berlin, 1993, pp. 124–137.

[13] R. A. Rueppel, Correlation immunity and the summation generator, *Advances in Cryptology - Crypto* '85, Lecture Notes in Computer Science, vol. 218, Springer-Verlag, Berlin, 1986, pp. 260–272.

[14] T. Siegenthaler, Correlation immunity of nonlinear combining functions for cryptographic applications, *IEEE Transactions on Information Theory*, vol. IT-30, Sept. 1984, pp. 776–780.

[15] T. Siegenthaler, Decrypting a class of stream ciphers using ciphertext only, *IEEE Transactions on Computers*, vol. C-34, Jan. 1985, pp. 81–85.

[16] O. Staffelbach and W. Meier, Cryptographic significance of the carry for ciphers based on integer addition, *Advances in Cryptology - Crypto* '90, Lecture Notes in Computer Science, vol. 537, Springer-Verlag, Berlin, 1991, pp. 601–614.