Edit Distance Correlation Attack on the Alternating Step Generator

Jovan Dj. Golić[1] * and Renato Menicocci[2]

[1] School of Electrical Engineering, University of Belgrade
Bulevar Revolucije 73, 11001 Beograd, Yugoslavia
Email: golic@galeb.etf.bg.ac.yu
[2] Fondazione Ugo Bordoni
Via B. Castiglione 59, 00142 Roma, Italy
Email: rmenic@fub.it

Abstract. A novel edit distance between two binary input strings and one binary output string of appropriate lengths which incorporates the stop/go clocking in the alternating step generator is introduced. An efficient recursive algorithm for the edit distance computation is derived. The corresponding correlation attack on the two stop/go clocked shift registers is then proposed. By systematic computer simulations, it is shown that the minimum output segment length required for a successful attack is linear in the total length of the two stop/go clocked shift registers. This is verified by experimental attacks on relatively short shift registers.

Key words. Stream ciphers, clock-controlled shift registers, alternating step generator, edit distance, cryptanalysis, correlation attacks.

1 Introduction

Keystream generators for stream cipher applications consisting of a small number of clock-controlled shift registers combined by a linear function seem to provide an efficient means for producing sequences with long period, high linear complexity, and good statistical properties, see [7]. The stop-and-go clocking is particularly popular for high speed applications. At any time, a stop/go shift register is clocked once if the clock-control input bit is equal to 1 and is not clocked at all otherwise. The clock-control sequence can be generated by another, regularly clocked shift register, whereas the inherent autocorrelation weakness due to the repetition of bits in a stop/go shift register can be overcome by linearly combining its output with the output of an additional regularly clocked shift

* This work was done while the first author was with the Information Security Research Centre, Queensland University of Technology, Brisbane, Australia. Part of this work was carried out while the first author was on leave at the Isaac Newton Institute for Mathematical Sciences, Cambridge, United Kingdom. This research was supported in part by the Science Fund of Serbia, grant #04M02, through the Mathematical Institute, Serbian Academy of Science and Arts.

register [1], which may be the same as the clock-control one, as is the case in the well-known stop/go cascades [7]. Typically, all the shift registers have linear feedback. However, the additional linear feedback shift register (LFSR) introduced to improve the statistics is then vulnerable to a fast correlation attack based on the repetition weakness [12]. If the clock-control LFSR is itself linearly combined with the stop/go one to produce the output, then it succumbs to a specific, conditional correlation attack [10] which exploits the same repetition weakness, but in a different, perhaps unexpected way.

The alternating step generator (ASG) [8] is an interesting combination of three binary LFSRs, two of which, LFSR_1 and LFSR_2, are stop/go clocked in a special way by the third one, LFSR_3, which is regularly clocked. Instead of LFSR_3, one may also use any binary keystream generator. More precisely, if the clock-control bit is equal to 1, then LFSR_1 is clocked and LFSR_2 is not, and if the clock-control bit is equal to 0, then LFSR_2 is clocked and LFSR_1 is not. The output sequence is formed as the bitwise sum of the two stop/go clocked LFSR sequences. Although it was proposed before the appearance of [12] and [10], the ASG is not vulnerable to the attacks there introduced. It is shown in [8] that the initial state of the clock-control LFSR_3 can be recovered via a divide-and-conquer attack which can be regarded as a special kind of the linear consistency attack [11], which appeared later. Namely, if and only if the guess about the initial state of LFSR_3 is correct, then the first binary derivative of the output sequence gives rise to the first binary derivatives of both the regularly clocked LFSR_1 and LFSR_2 sequences which are then easily tested for linear complexity by the Berlekamp-Massey algorithm [9].

The objective of this paper is to investigate whether a divide-and-conquer attack on LFSR_1 and LFSR_2 is possible. A way of doing this is to take the edit distance [2] or edit probability [3] approaches developed for memoryless combiners (for combiners with memory, see [6]) and adapt them to deal with the stop/go clocking. The main idea is to define the edit distance between two binary input strings and one binary output string of appropriate lengths as the minimum possible number of effective substitutions (complementations) needed in the combination string, produced from the two input strings by the stop/go clocking in the ASG manner, to obtain the output string, where the minimum is taken over all binary clock-control strings. Our first result is to prove that this unusual edit distance can be computed efficiently by a recursive algorithm whose computational complexity is quadratic in the output string length. Our second contribution is to show by systematic experiments obtained by computer simulations that the minimum output sequence length required for the success of the corresponding edit distance correlation attack is linear in the total length of LFSR_1 and LFSR_2. The reconstruction of the LFSR_3 initial state is also discussed.

In Section 2, a more detailed description of the ASG along with its basic properties is provided. The edit distance and the recursive algorithm for its efficient computation are presented in Section 3, and the experimental results on the underlying embedding probabilities are shown in Section 4. The corresponding correlation attack is explained and experimentally verified in Section 5. Conclusions and some open problems are given in Section 6.

2 Alternating Step Generator

In this section, we recall the structure and basic properties of the alternating step generator (ASG), as presented in [8]. As shown in Fig. 1, the output of the ASG is obtained by bitwise addition (modulo two) of the output sequences of two binary linear feedback shift registers, LFSR$_1$ and LFSR$_2$, whose stop/go clocking is defined by a binary clock-control generator (CCG), which is typically another, but regularly clocked LFSR, denoted as LFSR$_3$. It is assumed that LFSR$_1$ and LFSR$_2$ have different irreducible feedback polynomials of respective degrees r_1 and r_2 and respective coprime periods P_1 and P_2. At every step, only one LFSR is stepped and the output bit is assumed to be produced in the step-then-add manner. Let c_t denote the output bit of the CCG at step $t \geq 1$. Then, in order to obtain the ASG output bit z_t at step t, we first step LFSR$_1$ or LFSR$_2$ depending on whether $c_t = 1$ or $c_t = 0$, respectively, and then we add modulo two the output bits of LFSR$_1$ and LFSR$_2$. Observe that LFSR$_1$ is stop/go clocked, whereas LFSR$_2$ is go/stop clocked. In [8], some good cryptographic properties of the ASG, such as a long period (P), a high linear complexity (L), and approximately uniform relative frequency of short output patterns on a period are established, under the assumption that the clock-control sequence is a de Bruijn sequence of period 2^k. More precisely, it is proven that $P = P_1 P_2 2^k$ and $(r_1 + r_2) 2^{k-1} < L \leq (r_1 + r_2) 2^k$, whereas for approximately uniform distribution of the output patterns of length not bigger than $\min(r_1, r_2)$, it is in addition required that the feedback polynomials of LFSR$_1$ and LFSR$_2$ be primitive as well. It is expected that similar results also hold if the CCG is a LFSR with a primitive feedback polynomial whose period is coprime to $P_1 P_2$.

As mentioned in the previous section, it is also shown in [8] that there exists a (linear consistency) divide-and-conquer attack on the clock-control generator.

3 Edit Distance

Let $X^{n+1} = x_1, x_2, \ldots, x_{n+1}$ and $Y^{n+1} = y_1, y_2, \ldots, y_{n+1}$ denote two binary input strings and let $Z^n = z_1, z_2, \ldots, z_n$ denote a binary output string. Given a binary clock-control string $C^n = c_1, c_2, \ldots, c_n$, let $\hat{Z}^n = \hat{z}_1, \hat{z}_2, \ldots, \hat{z}_n$ denote the combination string produced from X^{n+1} and Y^{n+1} by the step-then-add alternating stepping according to C^n (X^{n+1} and Y^{n+1} correspond to the regularly clocked LFSR$_1$ and LFSR$_2$ sequences of length $n + 1$, respectively). Accordingly, we initially have $\hat{z}_1 = x_1 \oplus y_2$ if $c_1 = 0$ and $\hat{z}_1 = x_2 \oplus y_1$ if $c_1 = 1$, whereas for any $1 \leq s \leq n - 1$ and $0 \leq l \leq s$, if l denotes the number of ones in C^s, then $\hat{z}_s = x_{l+1} \oplus y_{s+1-l}$ and we have $\hat{z}_{s+1} = x_{l+1} \oplus y_{s+2-l}$ if $c_{s+1} = 0$ and $\hat{z}_{s+1} = x_{l+2} \oplus y_{s+1-l}$ if $c_{s+1} = 1$.

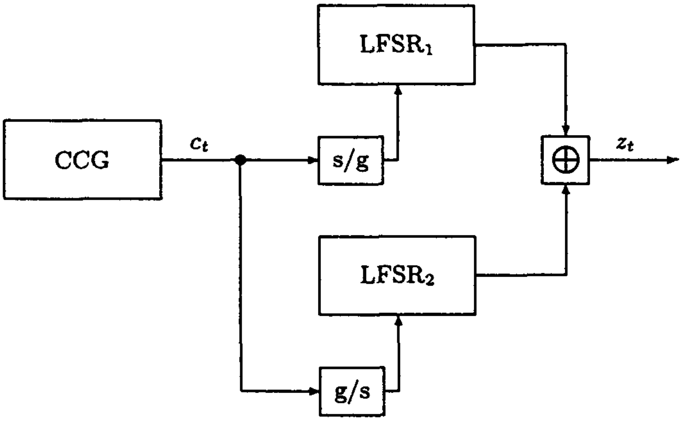

Fig. 1. The alternating step generator.

The *edit distance* between a given pair of strings (X^{n+1}, Y^{n+1}) and a given string Z^n, denoted as $D(X^{n+1}, Y^{n+1}; Z^n)$, is then defined by

$$D(X^{n+1}, Y^{n+1}; Z^n) = \min_{C^n \in \{0,1\}^n} d_H(Z^n, \hat{Z}^n) \qquad (1)$$

where $d_H(Z^n, \hat{Z}^n)$ denotes the Hamming distance between Z^n and $\hat{Z}^n$. In other words, the edit distance is defined as the minimum number of effective substitutions needed to obtain Z^n from $\hat{Z}^n$, where the minimum is over all 2^n binary clock-control strings C^n. Apart from the substitutions in the combination string $\hat{Z}^n$, the edit transformation of the input strings (X^{n+1}, Y^{n+1}) into the output string Z^n also contains one deletion of the first bit from one of the input strings and exactly $n-1$ repetitions of bits in the input strings, regardless of the clock-control string C^n. Consequently, the only informative part of the edit transformation are effective substitutions, as reflected in our definition of the edit distance.

Our basic objective is to examine whether the defined edit distance can be computed efficiently by an algorithm whose computational complexity is significantly smaller than 2^n, which corresponds to the computation of (1) by the exhaustive search over all C^n. To this end, for any $1 \leq s \leq n$ and $0 \leq l \leq s$, we define the *partial edit distance* $W(l, s)$ as $D(X^{s+1}, Y^{s+1}; Z^s)$ under an additional constraint that the binary clock-control string contains exactly l ones, that is,

$$W(l, s) = \min_{C^s : w_H(C^s) = l} d_H(Z^s, \hat{Z}^s) \qquad (2)$$

where $w_H(C^s)$ is the Hamming weight of C^s. Accordingly, the last bit $\hat{z}_s$ is in (2) always produced from the input bits x_{l+1} and y_{s+1-l}, so that the edit transformation involves the prefixes X^{l+1} and Y^{s+1-l} only. The partial edit distance can then be represented as

$$W(l,s) \;=\; \min_{C^s : w_H(C^s)=l} \sum_{k=1}^{s} (z_k \oplus \hat{z}_k) \tag{3}$$

where the Hamming distance is expressed as the integer sum of binary variables.

We are now ready to formulate a theorem which enables the efficient computation of the edit distance based on a recursive property of the partial edit distance.

Theorem 1. *For any X^{n+1}, Y^{n+1}, and Z^n, we have*

$$D(X^{n+1}, Y^{n+1}; Z^n) \;=\; \min_{0 \le l \le n} W(l,n) \tag{4}$$

where the partial edit distance $W(l,n)$ is computed recursively by

$$W(l,s) \;=\; (x_{l+1} \oplus y_{s+1-l} \oplus z_s) + \min\left(W(l-1, s-1), W(l, s-1)\right) \tag{5}$$

for $1 \le s \le n$ and $0 \le l \le s$, with the initial values $W(-1,s) = W(s+1,s) = \infty$ and $W(0,0) = 0$.

Proof. First observe that (4) is an immediate consequence of the definition of the partial edit distance. Second, for $s = 1$, (3) directly implies that $W(0,1) = x_1 \oplus y_2 \oplus z_1$ and $W(1,1) = x_2 \oplus y_1 \oplus z_1$ which can also be obtained by (5) from the given initial values.

Now, assume that $s > 1$. Since by definition, $\hat{z}_s = x_{l+1} \oplus y_{s+1-l}$, (3) can then be put into the form

$$\begin{aligned} W(l,s) \;&=\; (x_{l+1} \oplus y_{s+1-l} \oplus z_s) \\ &+ \min\left(\min_{C^{s-1} : w_H(C^{s-1})=l-1} \sum_{k=1}^{s-1} (z_k \oplus \hat{z}_k), \; \min_{C^{s-1} : w_H(C^{s-1})=l} \sum_{k=1}^{s-1} (z_k \oplus \hat{z}_k) \right) \end{aligned} \tag{6}$$

where the first and the second minima correspond to clock-control strings whose last bit c_s is equal to one and zero, respectively (the given initial values take care of the case when $l = 0$ or $l = s$). Equation (5) then follows directly in view of (3). □

The time and space complexities of the recursive algorithm corresponding to Theorem 1 are clearly $O(n^2)$ and $O(n)$, since only the values of the partial edit distance for the current and the preceding value of s have to be stored at a time. The algorithm is thus feasible even if n is very large. One can also store the whole matrix of the partial edit distances, indexed by s and l, in $O(n^2)$ space along with the value(s) of the clock-control bit c_s for which the minimum in (5) is achieved. By backtracking through the matrix, one can then recover all possible clock-control strings giving rise to the minimum number of effective substitutions representing the edit distance.

Some basic symmetry properties of the defined edit distance are captured by the following two propositions. For an arbitrary binary sequence or string $A = a_1, a_2, a_3, a_4, \ldots$, let $\bar{A} = \bar{a}_1, \bar{a}_2, \bar{a}_3, \bar{a}_4, \ldots$, $\tilde{A} = a_1, \bar{a}_2, a_3, \bar{a}_4, \ldots$, and $\tilde{\bar{A}} = \bar{a}_1, a_2, \bar{a}_3, a_4, \ldots$, where $\bar{a}$ denotes the complement of a bit a.

Proposition 2.

$$D(X^{n+1},Y^{n+1};Z^n) = D(\bar{X}^{n+1},\bar{Y}^{n+1};Z^n) = D(\bar{X}^{n+1},Y^{n+1};\bar{Z}^n). \quad (7)$$

Proof. Given a clock-control string C^n, if $\hat{Z}^n$ is the combination string produced from (X^{n+1},Y^{n+1}), then $\hat{Z}^n$ is also the combination string produced from $(\bar{X}^{n+1},\bar{Y}^{n+1})$ and $\bar{\hat{Z}}^n$ is the combination string produced from $(\bar{X}^{n+1},Y^{n+1})$. The proposition then directly follows from (1). □

Proposition 3.

$$D(X^{n+1},Y^{n+1};Z^n) = D(\tilde{X}^{n+1},\bar{\tilde{Y}}^{n+1};\tilde{Z}^n). \quad (8)$$

Proof. Given a clock-control string C^n, if $\hat{Z}^n$ is the combination string produced from (X^{n+1},Y^{n+1}), then $\tilde{\hat{Z}}^n$ is the combination string produced from $(\tilde{X}^{n+1},\bar{\tilde{Y}}^{n+1})$. The proposition then directly follows from (1). □

4 Embedding Probabilities

Let $P_n(D|Z^n)$ denote the probability that the edit distance $D(X^{n+1},Y^{n+1};Z^n)$ is equal to D when Z^n is fixed and the pair (X^{n+1},Y^{n+1}) is randomly chosen according to the uniform probability distribution. Also, let $P_n(D)$ denote the expected value of $P_n(D|Z^n)$ over a uniformly distributed Z^n, that is, the probability that the edit distance is equal to D when the triple (X^{n+1},Y^{n+1},Z^n) is randomly chosen according to the uniform probability distribution. The underlying uniform probability distributions over the string pairs and triples mean that the defined probabilities are in fact the fractions of the string pairs and triples such that the edit distance is equal to D, respectively.

In particular, let $P_n(Z^n)$ denote the probability, $P_n(0|Z^n)$, that the edit distance is equal to zero. Since the zero edit distance means that there exists a clock-control sequence such that Z^n is produced from (X^{n+1},Y^{n+1}) by step-then-add alternating stepping, $P_n(Z^n)$ is also called the embedding probability, given an output string Z^n of length n, see [4] and [5]. Further, let $\bar{P}_n$, $P_n^{\max}$, and $P_n^{\min}$ denote the average, the maximum, and the minimum values of $P_n(Z^n)$ over a uniformly distributed Z^n, respectively. All of these embedding probabilities are related to the success of the edit distance correlation attack to be described in the next section. To this end, it is critical that $\bar{P}_n$ decreases with n and it is desirable that this decrease is exponentially fast. Apart from that, it will be nice if $P_n^{\max}$ has a similar behavior itself too.

The evaluation of the embedding probabilities defined above seems to be a difficult combinatorial problem, related to the problems investigated in [13], [4], and [5]. The desired exponential decrease with the output string length n is not apparent at all. The approach taken in this section is essentially to compute the embedding probabilities for smaller values of n by exhaustive counting and to estimate their values for larger n by counting on a random sample of suitable input and output strings.

Our first objective is to estimate the edit distance probability distribution $P_n(D)$, $0 \leq D \leq n$. Tables 1 and 2 show the observed minimum, maximum, mean, and median values along with the standard deviation of D in a random sample of 1000 triples (X^{n+1}, Y^{n+1}, Z^n) for $n = 10, (10), 100$ and $n = 200, (100), 1000$, respectively. It appears that the expected value of D increases approximately linearly with n, around $n/10 + 1$, whereas the standard deviation is small and increases very slowly with n.

Table 1: Statistics of D on 1000 random triples (X^{n+1}, Y^{n+1}, Z^n).

n	10	20	30	40	50	60	70	80	90	100
Min	0	0	0	0	1	2	2	3	4	5
Max	5	6	9	10	11	12	14	15	16	18
Mean	1.611	2.783	3.973	5.125	6.147	7.243	8.293	9.336	10.393	11.542
Median	2	3	4	5	6	7	8	9	10	12
Std Dev	.996	1.194	1.373	1.508	1.570	1.715	1.822	1.859	2.002	1.917

Table 2: Statistics of D on 1000 random triples (X^{n+1}, Y^{n+1}, Z^n).

n	200	300	400	500	600	700	800	900	1000
Min	13	24	32	41	51	59	70	77	81
Max	29	40	56	63	72	86	96	104	117
Mean	21.593	31.918	42.016	52.071	61.932	72.518	81.925	92.035	102.15
Median	22	32	42	52	62	72	82	92	102
Std Dev	2.626	2.823	3.137	3.550	3.707	3.767	3.942	4.304	4.562

Our main objective is to compute or estimate the embedding probabilities $\bar{P}_n$, $P_n^{\max}$, and $P_n^{\min}$. Table 3 displays the computed values of these probabilities obtained by exhaustive counting for $n = 5, (1), 10$. For each such n, $P_n^{\min}$ is achieved if Z^n is the constant zero string and $P_n^{\max}$ is achieved if Z^n is the prefix of the sequence $Z = 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, \ldots$. It is conjectured that this is also the case for every n (a related result is proven in [5]). In fact, the values of $P_n^{\max}$ and $P_n^{\min}$ given in Tables 4 and 5 to follow correspond to such strings, since the exhaustive counting is not feasible for larger n. As well, according to Propositions 2 and 3, if the minimum or the maximum is obtained for a string Z^n, then it is also obtained for $\bar{Z}^n$, $\tilde{Z}^n$, and $\bar{\tilde{Z}}^n$.

Table 3: $\bar{P}_n$, $P_n^{\max}$, and $P_n^{\min}$ determined by exhaustive computation.

n	5	6	7	8	9	10
$\bar{P}_n$	.3165	.2598	.2138	.1763	.1455	.1203
$P_n^{\max}$	.4546	.4136	.3766	.3440	.3137	.2869
$P_n^{\min}$	.1479	.08581	.04880	.02734	.01513	.008300

Table 4 gives the estimated values of $\bar{P}_n$, $P_n^{\max}$, and $P_n^{\min}$ for $n = 11, (1), 20$ on a random sample of 10000 pairs (X^{n+1}, Y^{n+1}) for $P_n^{\max}$ and $P_n^{\min}$ and 10000 triples (X^{n+1}, Y^{n+1}, Z^n) for $\bar{P}_n$, except for $n = 11, 12, 13$ where $P_n^{\max}$ and $P_n^{\min}$ are obtained by exhaustive counting.

Table 4: $\bar{P}_n$, $P_n^{\max}$, and $P_n^{\min}$ determined either by a 10000 points estimation or by exhaustive computation*.

n	11	12	13	14	15	16	17	18	19	20
$\bar{P}_n$	.1026	.0820	.0641	.0587	.0479	.0395	.0349	.0274	.0223	.0188
$P_n^{\max}$	.2621*	.2398*	.2194*	.2045	.1825	.1613	.1532	.1383	.1314	.1176
$P_n^{\min}$	.004516*	.002441*	.001312*	.0007	.0007	.0004	0	0	0	0

Table 5 gives the estimated values of $\bar{P}_n$ and $P_n^{\max}$ for $n = 25, (5), 100$ on a random sample of 30000 triples (X^{n+1}, Y^{n+1}, Z^n) and 30000 pairs (X^{n+1}, Y^{n+1}), respectively. All the obtained estimates of $P_n^{\min}$ are equal to zero, since this probability is then very small.

Table 5a: $\bar{P}_n$ and $P_n^{\max}$ determined by a 30000 points estimation.

n	25	30	35	40	45	50	55	60
$\bar{P}_n$	.0071	.003033	.0009333	.0004	.0001667	.0000667	.0000667	0
$P_n^{\max}$	.0776	.05127	.0335	.02137	.01293	.007867	.005133	.003233

Table 5b: $\bar{P}_n$ and $P_n^{\max}$ determined by a 30000 points estimation.

n	65	70	75	80	85	90	95	100
$\bar{P}_n$	0	0	0	0	0	0	0	0
$P_n^{\max}$	.002467	.001833	.001067	.0006	.0006333	.0003667	.0000667	.0000667

Tables 3-5 are consistent with the exponential decrease with n of all the probabilities $\bar{P}_n$, $P_n^{\max}$, and $P_n^{\min}$. Namely, each of them seems to have the form $a\, b^n$, $b < 1$, for large n. The corresponding estimates of the parameters a and b are presented in Table 6. Since not all of them are equally reliable, especially if a is concerned, the derived estimates are shown for each of the tables separately.

Table 6: Estimation of a and b based on Tables 3*,4**, and 5***.

	a^*	a^{**}	a^{***}	b^*	b^{**}	b^{***}
$\bar{P}_n$	.8303	.7926	-	.8241	.8292	.7357
$P_n^{\max}$	.7186	.6930	-	.9121	.9151	.9064
$P_n^{\min}$	2.695	-	-	.5622	.5305	-

To be on the conservative side, the probabilities $\bar{P}_n$, $P_n^{\max}$, and $P_n^{\min}$ are approximated for large n by using the maximum values for a and b as

$$\bar{P}_n \approx 0.83 \cdot 0.83^n, \quad P_n^{\max} \approx 0.72 \cdot 0.915^n, \quad P_n^{\min} \approx 2.7 \cdot 0.562^n. \qquad (9)$$

5 Correlation Attack

Assume that the feedback polynomials of LFSR_1 and LFSR_2 are known. The objective of the edit distance correlation attack proposed in this section is to reconstruct the secret key dependent initial states of LFSR_1 and LFSR_2 from a known, sufficiently long segment of the output sequence, in the known plaintext scenario. The main point is to measure the statistical dependence or the correlation between the output sequence and the regularly clocked LFSR_1 and LFSR_2 sequences by the edit distance defined in Section 3.

A segment Z^n of the first n successive output bits is produced from the output segments of the regularly clocked LFSR_1 and LFSR_2, called here the input segments, whose lengths are variable depending on the clock-control sequence. If the unknown clock-control sequence is assumed to be purely random, that is, a sequence of independent uniformly distributed binary random variables, then the average length of the input LFSR segments is $n/2 + 1$, whereas their maximum possible length is $n + 1$. Accordingly, for any assumed initial states of LFSR_1 and LFSR_2, one can by their linear recursions generate the two input segments of the same length $n + 1$, X^{n+1} and Y^{n+1}, respectively. The edit distance $D(X^{n+1}, Y^{n+1}; Z^n)$ is then efficiently computed by the recursive algorithm introduced in Section 3. This is repeated for every possible pair of the assumed LFSR_1 and LFSR_2 initial states, altogether $2^{r_1+r_2}$ of them. Since for the correct LFSR initial states the edit distance is clearly equal to zero, all the obtained initial state pairs with the zero edit distance represent the candidates for the correct initial state pairs. The zero edit distance indicates that there exists a clock-control sequence such that Z^n is produced from (X^{n+1}, Y^{n+1}) by step-then-add alternating stepping. In this sense, this edit distance correlation attack can be viewed as a specific embedding attack, just like the Levenshtein-like edit distance attack [2] reduces to the embedding attack [13] in case of a single clock-controlled shift register. Recall that the irregular clocking considered in [2] and [13] is constrained in that the number of clocks per each output bit is an upper-bounded positive integer, whereas the unconstrained clocking is analyzed in [4].

5.1 Theoretical analysis

Ideally, if n is large enough, then there will remain only one candidate for the initial states of LFSR_1 and LFSR_2. This can happen only if the embedding probability $P_n(Z^n)$ defined in Section 4 is sufficiently small. Recall that $P_n(Z^n)$ is the fraction of string pairs (X^{n+1}, Y^{n+1}) for which $D(X^{n+1}, Y^{n+1}; Z^n) = 0$. Namely, the expected number of candidates is clearly $2^{r_1+r_2} P_n(Z_n)$, so that the edit distance correlation attack is deemed successful if and only if, approximately,

$$2^{r_1+r_2} P_n(Z_n) \leq 1, \qquad (10)$$

see [4]. According to Section 4, let $\bar{P}_n$, $P_n^{\max}$, and $P_n^{\min}$ denote the average, the maximum, and the minimum values of $P_n(Z^n)$ over a uniformly distributed Z^n, respectively. Then, by substituting these probabilities for $P_n(Z^n)$ in the condition (10), we obtain the minimum output segment length n required for success for the average, for the worst, and for the best Z^n, respectively. More precisely, if any of these probabilities has the exponential form $a\,b^n$, where $b < 1$, then (10) reduces to

$$n \geq \frac{r_1 + r_2 + \log a}{-\log b} \tag{11}$$

which means that the required output segment length is essentially linear in the total length of LFSR_1 and LFSR_2. In view of (9), the required output segment length is then approximately given as

$$n \geq 3.72\,(r_1 + r_2) - 1 \tag{12}$$

$$n \geq 7.8\,(r_1 + r_2) - 3.7 \tag{13}$$

$$n \geq 1.2\,(r_1 + r_2) + 1.72 \tag{14}$$

in the average, the worst, and the best case, respectively.

The number of the surviving candidate initial state pairs for a chosen n can further be reduced to just one or a very small number by increasing the length n. More precisely, we show that the number of candidate initial state pairs is at least two regardless of how large n is. Moreover, with a certain probability it can also be bigger than two, depending on some linear equations among the candidate pair bits being satisfied or not.

Proposition 4. *The number of candidate initial state pairs for* LFSR_1 *and* LFSR_2 *selected by the zero edit distance criterion is at least equal to two, provided that* $n + 1 > \max(r_1, r_2)$.

Proof. Suppose that the clocking string $C^n = c_1, c_2, \ldots, c_n$, with $c_1 = 1$, is able to transform a pair (X^{n+1}, Y^{n+1}) into Z^n. Suppose that X^{n+1} and Y^{n+1} are generated by LFSR_1 and LFSR_2 from initial states $X^{r_1} = x_1, x_2, \ldots, x_{r_1}$ and $Y^{r_2} = y_1, y_2, \ldots, y_{r_2}$, respectively, where $n+1 > \max(r_1, r_2)$. Then, there always exists another pair $(\hat{X}^{n+1}, \hat{Y}^{n+1})$ produced by LFSR_1 and LFSR_2 from appropriate initial states $\hat{X}^{r_1}$ and $\hat{Y}^{r_2}$, respectively, such that $D(\hat{X}^{n+1}, \hat{Y}^{n+1}; Z^n) = 0$. Namely, it suffices to use $\hat{x}_i = x_{i+1}$, $i = 1, 2, \ldots, r_1$, and $\hat{y}_i = y_{i-1}$, $i = 1, 2, \ldots, r_2$, (y_0 is obtained by backward clocking of LFSR_2) along with the clocking string $\hat{C}^n$ with $\hat{c}_1 = 0$ and $\hat{c}_i = c_i$, $i = 2, 3, \ldots, n$. An analogous proof is readily obtained for the case where $c_1 = 0$. □

For each obtained candidate initial state pair, one can also store the whole matrix of the partial edit distances and then by backtracking recover all possible clock-control strings C^n giving rise to the zero edit distance. The average number of such clock-control strings per candidate pair can be estimated as

$$m_n = \frac{2^n}{2^n \bar{P}_n} = \frac{1}{\bar{P}_n} \approx 1.2 \cdot 2^{0.269\,n}. \tag{15}$$

Note that if n is chosen so that $2^{r_1+r_2}\bar{P}_n \approx 1$, then $m_n \approx 2^{r_1+r_2}$. If the clock-control sequence is generated by another known LFSR, LFSR$_3$, of length r_3 and if $n > r_3$, then the number of C^n can be reduced by checking the LFSR$_3$ recursion which can be performed sequentially, one bit at a time, by backtracking through the matrix of partial edit distances. In fact, starting from m_{r_3} possible strings C_{r_3} for $n = r_3$, each new bit examined is expected to halve the number of the surviving clock-control strings which is therefore reduced to only one if $n - r_3 \geq \log m_n$, that is, if, approximately,

$$n \geq 1.37 r_3. \tag{16}$$

The complexity of this search is upper-bounded by m_n, which is close to $2^{0.37 r_3}$ if $n \approx 1.37 r_3$ and is close to $2^{r_1+r_2}$ if $n \approx 3.72\,(r_1 + r_2)$. In addition, it may happen that some clock-control bits are uniquely determined without exploiting the LFSR$_3$ recursion which can be used to reduce the search effort. So, if apart from (12), the condition (16) is also satisfied, then the described search is likely to reduce the number of candidate initial state pairs for LFSR$_1$ and LFSR$_2$ to only one, correct pair and also to uniquely determine the initial state of LFSR$_3$. The obtained candidate initial state triples for all the LFSRs are then tested for correctness on a longer output sequence. Note that in view of the structure of the ASG generator, one may expect that different LFSR initial state triples necessarily give rise to different output sequences, so that the solution for the LFSR initial states is very likely to be unique.

Finally, one may observe that if one of the initial states, for LFSR$_1$ or LFSR$_2$, is guessed correctly, then, instead of the embedding probability as such, one should in fact consider the fraction of the input and output string triples giving rise to the zero edit distance provided that one of the input strings is guessed correctly. If this, conditional embedding probability was bigger than the embedding probability defined before, then the number of the obtained candidate pairs would very likely be bigger than just two, see Proposition 4. Note that this is not a problem, since the correct individual initial states can then easily be recovered as the ones that appear in most the obtained candidate pairs. Note that the computational complexity of the proposed edit distance correlation attack remains $O(2^{r_1+r_2})$ anyway. However, for the sake of completeness, we have also experimentally obtained an estimate of this conditional embedding probability, $\approx 0.85 \cdot 0.85^n$, which is very close to the estimate of $\bar{P}_n$ given in (9).

5.2 Experimental attacks

A number of computer simulations were conducted to show that the above edit distance correlation attack can work in practice. The clock-control generator was assumed to be another linear feedback shift register, LFSR$_3$, of length r_3. Only primitive feedback polynomials were used for all the LFSRs. The correlation attack was performed in the way explained in Subsection 5.1. The feedback polynomials were assumed to be known and the objective was to reconstruct the initial states of LFSR$_1$ and LFSR$_2$ along with the initial state of LFSR$_3$ from a sufficiently long segment of the ASG output sequence.

Some examples of the experimental results obtained are shown in Table 7, where $N_{1,2}$ denotes the number of candidate initial state pairs for LFSR$_1$ and LFSR$_2$ satisfying the zero edit distance criterion and N_3 denotes the number of candidate initial states for LFSR$_3$ obtained from the found $N_{1,2}$ pairs by examining the corresponding clock-control strings. As different LFSR initial state triples very likely yield different output sequences, then $N_3 = 1$ effectively reduces the number of candidate initial state pairs for LFSR$_1$ and LFSR$_2$ to only one. Observe that the same ASG was considered in the experiments 1 and 1′, the only difference being the keystream sequence length n, which was increased in the experiment 1′ to test the ability of the attack to reduce the number of candidate initial states for LFSR$_3$ when $N_{1,2}$ is already at its minimum. Accordingly, in each of the experiments a unique solution for the LFSR initial states was obtained. Similar results were also obtained in a number of other experiments where the shift register lengths r_1 and r_2 were smaller than those from Table 7.

Notice that multiple candidates for the initial states of LSFR$_1$ and LFSR$_2$ can appear and that their number, $N_{1,2}$, which is expected to be a small positive integer, can not be reduced to one by increasing n, see Proposition 4. In practice, as indicated by (12), it was observed that $N_{1,2}$ is minimized by using $n \approx 4(r_1 + r_2)$. Multiple candidates for the initial states of LSFR$_1$ and LFSR$_2$ can effectively be reduced to only one candidate by reconstructing the LFSR$_3$ initial state from all possible clock-control strings, provided that n is sufficiently long, see (16).

Table 7: Experimental results.

Experiment	1	1′	2	3	4	5
n	180	200	120	150	180	160
r_1, r_2, r_3	14, 14, 64	14, 14, 64	15, 15, 48	15, 15, 64	15, 15, 64	16, 16, 64
$N_{1,2}, N_3$	2, 2	2, 1	6, 1	4, 2	2, 1	3, 1

6 Conclusions

A novel edit distance between two binary input strings and one binary output string of appropriate lengths which reflects the specific stop/go clocking in the ASG generator is introduced. An efficient recursive algorithm for the edit distance computation whose time complexity is quadratic in the output string length is derived. By systematic computer simulations, the underlying embedding probabilities are shown to exponentially decrease with the output string length. The corresponding edit distance correlation attack on the two stop/go clocked shift registers in the ASG generator is then proposed. The attack essentially consists in computing the edit distance for every possible pair of the initial states of the two shift registers and in finding all the pairs with the zero edit distance. By using the evaluated embedding probabilities, it is then established that the minimum output segment length required for a successful attack is linear in the total length of the two stop/go clocked shift registers. More precisely, only about four total lengths on average and about eight total lengths in the worst case are sufficient for the success. The reconstruction of the initial state of the clock-control shift register is also discussed. The theory is illustrated by successful experimental attacks conducted on relatively short shift registers. From the practical cryptographic standpoint, the results show that the total length of the two stop/go clocked shift registers should be sufficiently large in order to prevent the exhaustive search over their initial states. On the other hand, note that the clock-control shift register itself should be long enough to prevent the divide-and-conquer attack from [8].

Finding analytical expressions for the embedding probabilities is an interesting, but difficult combinatorial problem, see [13], [5], and [4] for related problems regarding a single clock-controlled shift register. Defining an appropriate edit probability instead of the edit distance is an approach which may possibly reduce the required output segment length, see [3] and [4] for related previous work in this direction. Finally, it remains to be investigated whether a correlation attack on individual stop/go clocked shift registers in the ASG generator based on another special edit distance or edit probability is also possible.

References

1. T. Beth and F. C. Piper, "The stop-and-go generator," Advances in Cryptology - EUROCRYPT '84, *Lecture Notes in Computer Science*, vol. 209, T. Beth, N. Cot, and I. Ingemarsson eds., Springer-Verlag, pp. 88-92, 1985.
2. J. Dj. Golić and M. Mihaljević, "A generalized correlation attack on a class of stream ciphers based on the Levenshtein distance," *Journal of Cryptology*, vol. 3(3), pp. 201-212, 1991.
3. J. Dj. Golić and S. Petrović, "A generalized correlation attack with a probabilistic constrained edit distance," Advances in Cryptology - EUROCRYPT '92, *Lecture Notes in Computer Science*, vol. 658, R. A. Rueppel ed., Springer-Verlag, pp. 472-476, 1993.
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, A. De Santis ed., Springer-Verlag, pp. 230-243, 1995.
5. J. Dj. Golić, "Constrained embedding probability for two binary strings," *SIAM Journal on Discrete Mathematics*, vol. 9(3), pp. 360-364, 1996.
6. J. Dj. Golić, "Edit distance correlation attacks on clock-controlled combiners with memory," Information Security and Privacy, *Lecture Notes in Computer Science*, vol. 1172, J. Pieprzyk ed., Springer-Verlag, pp. 169-181, 1996.
7. D. Gollmann and W. G. Chambers, "Clock-controlled shift registers: a review," *IEEE Journal on Selected Areas in Communications*, vol. 7, pp. 525-533, May 1989.
8. C. G. Günther, "Alternating step generators controlled by de Bruijn sequences," Advances in Cryptology - EUROCRYPT '87, *Lecture Notes in Computer Science*, vol. 304, D. Chaum and W. L. Price eds., Springer-Verlag, pp. 5-14, 1988.
9. J. L. Massey, "Shift-register synthesis and BCH decoding," *IEEE Trans. Inform. Theory*, vol. IT-15, pp. 122-127, Jan. 1969.

10. R. Menicocci, "Cryptanalysis of a two-stage Gollmann cascade generator," *Proceedings of SPRC* '93, Rome, Italy, pp. 62-69, 1993.
11. 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, G. Brassard ed., Springer-Verlag, pp. 164-174, 1990.
12. K. Zeng, C. H. Yang, and T. R. N. Rao, "An improved linear syndrome algorithm in cryptanalysis with applications," Advances in Cryptology - CRYPTO '90, *Lecture Notes in Computer Science*, vol. 537, A. J. Menezes and S. A. Vanstone eds., Springer-Verlag, pp. 34-47, 1991.
13. M. V. Živković, "An algorithm for the initial state reconstruction of the clock-controlled shift register," *IEEE Trans. Inform. Theory*, vol. IT-37, pp. 1488-1490, Sept. 1991.