Reduced Complexity Attacks on the Alternating Step Generator

Shahram Khazaei[1], Simon Fischer[2], and Willi Meier[2]

[1] EPFL, Lausanne, Switzerland
[2] FHNW, Windisch, Switzerland

Abstract. In this paper, we present some reduced complexity attacks on the Alternating Step Generator (ASG). The attacks are based on a quite general framework and mostly benefit from the low sampling resistance of the ASG, and of an abnormal behavior related to the distribution of the initial states of the stop/go LFSR's which produce a given segment of the output sequence. Our results compare well with previous results as they show a greater flexibility with regard to known output of the ASG, which amounts in reduced complexity. We will also give a closed form for the complexity of attacks on ASG (and SG) as presented in [13].

Key words: Stream Cipher, Clock-Controlled Generator, Alternating Step Generator

1 Introduction

The Alternating Step Generator (ASG), a well-known stream cipher proposed in [11], consists of two stop/go clocked binary LFSR's, $\mathsf{LFSR_X}$ and $\mathsf{LFSR_Y}$, and a regularly clocked binary LFSR, $\mathsf{LFSR_C}$ of which the clock-control sequence is derived. The original description of ASG [11] is as follows. At each time, the clock-control bit determines which of the two stop/go LFSR's is clocked, and the output sequence is obtained as bit-wise sum of the two stop/go clocked LFSR sequences. It is known [13, 8, 12] that instead of working with the original definition of ASG we can consider a slightly different description for which the output is taken from the stop/go LFSR which has been clocked. More precisely, at each step first $\mathsf{LFSR_C}$ is clocked; then if the output bit of $\mathsf{LFSR_C}$ is one, $\mathsf{LFSR_X}$ is clocked and its output bit is considered as the output bit of the generator, otherwise $\mathsf{LFSR_Y}$ is clocked and the output bit of the generator is taken from this LFSR. Since in a cryptanalysis point of view these two generators are equivalent, we use the later one all over this paper and for simplicity we still call it ASG.

Several attacks have been proposed on ASG in the literature. Most of these attacks are applied in a divide-and-conquer based procedure targeting one or two of the involved LFSR's. We will focus on a divide-and-conquer attack which targets one of the two stop/go LFSR's.

A correlation attack on individual $\mathsf{LFSR_X}$ or $\mathsf{LFSR_Y}$ which is based on a specific edit probability has been introduced in [10]. The amount of required keystream is linear in terms of the length of the targeted LFSR and the correct initial state of the targeted LFSR is found through an exhaustive search over all possible initial states. In [13] some reduced complexity attacks on ASG and SG (Shrinking Generator, see [2]) were presented and the effectiveness of the attacks was verified numerically for SG while only few general ideas were proposed for ASG without any numerical or theoretical analysis. These methods avoid exhaustive search over all initial states,

however, the amount of needed keystream is exponential in terms of the length of the targeted LFSR. One of our contributions of this paper is to give a closed form for these reduced complexity attacks.

Our major objective of this paper is to investigate a general method which does not perform an exhaustive search over all possible initial states of the targeted LFSR. We will take advantage of the low *sampling resistance* of ASG. The notion of sampling resistance was first introduced in [1] and it was shown that a low sampling resistance has a big impact on the efficiency of time/memory/data trade-off attacks. Sampling is the capability of efficiently producing all the initial states which generate an output sequence starting with a particular m-bit pattern. Recently it was noticed that sampling may be useful along with other attacks in a unified framework [3]. The results of this paper represent a positive attempt to exploit such a connection for a concrete stream cipher.

For ASG, sampling is easy if the output length m is chosen to be about the total length of the two stop/go LFSR's. Another weakness of ASG which enables us to mount our attack is that different initial states of any of the two stop/go LFSR's have far different probabilities to be accepted as a candidate which can produce a given segment of length m of the output sequence. Systematic computer simulations confirm this striking behavior. The highly non-uniform distribution of different initial states of any of the stop/go LFSR's is valid for any segment of length about m, and the effect is more abnormal for some special outputs which we refer to as weak outputs. Thanks to the low sampling resistance of ASG we first try to find a subset of the most probable initial states which contains the correct one, then using the probabilistic edit distance [10] we distinguish the correct initial state. Our general approach can be faster than exhaustive search even if the amount of keystream is linear in terms of the length of the targeted LFSR, improving the results in [10]. With regard to reduced complexity attacks, our approach does assume less restricted output segments than in [13], a fact that has been confirmed by large-scale experiments. This enables attacks with significantly lower data complexity even for large instances of ASG (whereas asymptotical complexity is shown to be comparable over known methods).

The paper is organized as follows. In section 2 we will give a comprehensive list of the known attacks on ASG along with a short overview of them. A closed form for the reduced complexity attacks of [13] on ASG is given in Sect. 3. In Sect. 4 we present our attack in detail. Experimental results are in Sect. 5, and we finally conclude in Sect. 6.

2   Previous Attacks on ASG

Several attacks have been proposed on the ASG in the literature. This section will provide an overview of the different attacks. We will denote the length of registers LFSR$_\mathsf{C}$, LFSR$_\mathsf{X}$ and LFSR$_\mathsf{Y}$ by L_C, L_X and L_Y, respectively. If we only use parameter L, we apply the simplification $L := L_C = L_X = L_Y$.

2.1 Divide-and-Conquer Linear Consistency Attack

It is shown in [11] that the initial state of $\mathsf{LFSR_C}$ can be recovered by targeting its initial state in a divide-and-conquer based attack based on the fact that the output sequence of the ASG can be split into the regularly clocked $\mathsf{LFSR_X}$ and $\mathsf{LFSR_Y}$ sequences, which are then easily tested for low linear complexity. Hence the complexity of this attack is $\mathcal{O}(\min^2(L_X, L_Y)2^{L_C})$ assuming that only the feedback polynomial of $\mathsf{LFSR_C}$ is available. Under the assumption that the feedback polynomial of all LFSR's are available, which is the basic assumption of all other known attacks (including ours in this paper), the complexity of this attack would be $\mathcal{O}(\min(L_X, L_Y)2^{L_C})$ instead, since a parity check test can be used in place of linear complexity test. In this case the attack is a linear consistency attack [17]. We will use the idea of this attack to sample ASG in Sect. 4.1.

2.2 Edit Distance Correlation Attack

A correlation attack on $\mathsf{LFSR_X}$ and $\mathsf{LFSR_Y}$ combined, which is based on a specific edit distance, was proposed in [8]. If the initial states of $\mathsf{LFSR_X}$ and $\mathsf{LFSR_Y}$ are guessed correctly, the edit distance is equal to zero. If the guess is incorrect, the probability of obtaining the zero edit distance was experimentally shown to exponentially decrease in the length of the output string. Later, a theoretical analysis of this attack was developed in [12, 5]. The minimum length of the output string to be successful for an attack is about four times total lengths of $\mathsf{LFSR_X}$ and $\mathsf{LFSR_Y}$. As the complexity of computing the edit distance is quadratic in the length of the output string, the complexity of this attack is $\mathcal{O}((L_X + L_Y)^2 2^{L_X + L_Y})$. In addition, it was shown that the initial state of $\mathsf{LFSR_C}$ can then be reconstructed with complexity $\mathcal{O}(2^{0.27L_C})$.

2.3 Edit Probability Correlation Attack

A correlation attack on individual $\mathsf{LFSR_X}$ or $\mathsf{LFSR_Y}$ which is based on a specific edit probability was developed in [10]. For a similar approach, see [13]. The edit probability is defined for two binary strings: an input string, produced by the regularly clocked targeted LFSR from an assumed initial state, and a given segment of the ASG output sequence. The edit probability is defined as the probability that the given output string is produced from an assumed input string by the ASG in a probabilistic model, where the LFSR sequences are assumed to be independent and purely random. It turns out that the edit probability tends to be larger when the guess about the LFSR initial state is correct. More precisely, by experimental analysis of the underlying statistical hypothesis testing problem, it was shown that the minimum length of the output string to be successful for an attack is about forty lengths of the targeted LFSR. As the complexity of computing the edit probability is quadratic in the length of the output string, the complexity of reconstructing both LFSR initial states is $\mathcal{O}(\max^2(L_X, L_Y)2^{\max(L_X, L_Y)})$. This yields a considerable improvement over the edit distance correlation attack if L_X and L_Y are approximately equal and relatively large, as is typically suggested (for example, see, [15]).

Remark 1. Note that "edit distance correlation attack" means that the initial states of $\mathsf{LFSR_X}$ and $\mathsf{LFSR_Y}$ can be recovered regardless of the unknown initial state of $\mathsf{LFSR_C}$, whereas "edit probability correlation attack" means that the initial state of $\mathsf{LFSR_X}$ ($\mathsf{LFSR_Y}$) can be recovered regardless of unknown initial states of $\mathsf{LFSR_Y}$ ($\mathsf{LFSR_X}$) and $\mathsf{LFSR_C}$. However, the targeted LFSR initial states should be tested exhaustively. The main motivation for this paper is to investigate if the initial states of $\mathsf{LFSR_X}$ ($\mathsf{LFSR_Y}$) can be reconstructed faster than exhaustive search regardless of unknown initial states of $\mathsf{LFSR_Y}$ ($\mathsf{LFSR_X}$) and $\mathsf{LFSR_C}$.

2.4 Reduced Complexity Attacks

A first step to faster reconstruction of LFSR's initial states was suggested in [13], in which some reduced complexity attacks on ASG and SG are presented. In the next section, we will give a general expression in the parameter L_X, the length of target register $\mathsf{LFSR_X}$ (and in Appendix A, we give general expressions for SG). A second movement to faster reconstruction of LFSR initial states was suggested in [7], using an approach based on computing the posterior probabilities of individual bits of the regularly clocked $\mathsf{LFSR_X}$ and $\mathsf{LFSR_Y}$ sequences, when conditioned on a given segment of the output sequence. It is shown that these probabilities can be efficiently computed and the deviation of posterior probabilities from one half are theoretically analysed. As these probabilities represent soft-valued estimates of the corresponding bits of the considered LFSR sequences when regularly clocked, it is argued that the initial state reconstruction is thus in principle reduced to fast correlation attacks on regularly clocked LFSR's such as the ones based on iterative probabilistic decoding algorithms. Although this valuable work shows some vulnerability of the ASG towards fast correlation attacks, the practical use of these probabilities has not yet been deeply investigated. Nonetheless, these posterior probabilities can certainly be used to mount a distinguisher on ASG. This can be compared with [4], a similar work on SG for which a distinguisher was later developed in [9].

3 Johansson's Reduced Complexity Attacks

In [13] some reduced complexity attacks on the ASG and SG were presented, and the effectiveness of the attacks was verified numerically for the SG (while only few general ideas were proposed for the ASG without any numerical or theoretical analysis). We give a closed form for the reduced complexity attack on ASG, using the approximation $\binom{n}{w} \approx 2^{nh(w/n)}$ where $h(p)$ is the binary entropy function defined as

$$h(p) := -p \log_2(p) - (1-p) \log_2(1-p) . \tag{1}$$

In the first scenario, the attacker waits for a segment of M consecutive zeros (or ones) in the output sequence and assumes that exactly $M/2$ of them are from $\mathsf{LFSR_X}$. This is true with probability $\beta = \binom{M}{M/2} 2^{-M}$. The remaining $L - M/2$ bits of $\mathsf{LFSR_X}$ are then found by exhaustive search. Time and data complexities of this attack are $C_T = L^2 2^{L-M/2} \beta^{-1} = L^2 2^{L+M/2} \binom{M}{M/2}^{-1}$ and $C_D = 2^{M-1} \beta^{-1} = 2^{2M-1} \binom{M}{M/2}^{-1}$ (using overlapping blocks of keystream). Ignoring the polynomial and constant terms and

equaling the time and data complexities, we have $L - M/2 = M$, which shows $M = \frac{2}{3}L$. Thus the optimal complexities of this attack are $C_T = \mathcal{O}(L^2 2^{\frac{2}{3}L})$ and $C_D = \mathcal{O}(2^{\frac{2}{3}L})$. These arguments apply to both LFSR$_X$ and LFSR$_Y$.

Remark 2. The total time of the attack is composed of the time to filter the blocks of data with desired properties, and of the time to further process the filtered blocks. Although the unit of examination time of these two phases are not equal, we ignore this difference to simplify the analysis.

In another scenario in [13], it is suggested to wait for a segment of length M containing at most w ones (zeros) and make the assumption that only half of the zeros (ones) come from the LFSR$_X$. All the ones (zeros) and the remaining zeros (ones) are assumed to come from the LFSR$_Y$. This is true with probability $\beta = 2^{-w} \binom{M-w}{(M-w)/2} 2^{-(M-w)}$. The time and data complexities of this attack are then $C_T = L^2 2^{L-(M-w)/2} \beta^{-1}$ and $C_D = 2^{M-1} \binom{M}{w}^{-1} \beta^{-1}$, respectively. With $w := \alpha M$, ignoring the constant and polynomial terms, and equaling the time and data complexities, we have $L - (1-\alpha)M/2 + \alpha M = M - h(\alpha)M + \alpha M$, which results in $M = L/(3/2 - \alpha/2 - h(\alpha))$. The minimum value of the exponents $M(1 - h(\alpha) + \alpha)$ is $0.6406L$, which is achieved for $\alpha \approx 0.0727$ (and hence $M = 0.9193L$ and $w = 0.0668L$). Therefore, the optimal complexities are $C_T = \mathcal{O}(L^2 2^{0.64L})$ and $C_D = \mathcal{O}(2^{0.64L})$. Note that this complexity is only for reconstruction of the initial state of LFSR$_X$. The complexity for recovering the initial state of LFSR$_Y$ highly depends on the position of ones (zeros) in the block. In the best case, the block starts with w ones (zeros) and the complexity becomes $C_T = L^2 2^{L-(M+w)/2}$. In the worst case, the attacker has to search for the positions of ones (zeros), and the complexity becomes $C_T = \binom{(M+w)/2}{w} L^2 2^{L-(M-w)/2}$. It is difficult to give an average complexity, but we expect that it is close to the worst case complexity. With $M = 0.9193L$ and $w = 0.0668L$, this gives $C_T = \mathcal{O}(L^2 2^{0.69L})$ to recover the initial state of LFSR$_Y$. Consequently, as a distinguishing attack, this scenario operates slightly better than the previous one, but as an initial state recovery it is slightly worse.

4 New Reduced Complexity Attack

Before we describe our attack in detail, let us introduce some notations. Throughout the paper, the symbols Pr and E are respectively used for probability of an event and expectation of a random variable. For simplicity we do not distinguish between random variables and their instances. We use $A := \{a_i\}$ for a general binary sequence, $A_k^m := \{a_i\}_{i=k}^m$ for a segment of it and $A^m := \{a_i\}_{i=1}^m$ for a prefix of length m. The number of 1's in A is denoted by wt(A). We define the first derivative of A as $\{a_i + a_{i+1}\}$ and denote it by $\dot{A}$. Let C, X, Y and Z denote the regular output sequences of LFSR$_C$, LFSR$_X$, LFSR$_Y$ and the output sequence of the ASG itself, respectively. The initial state of the LFSR's can be represented by C^L, X^L and Y^L.

4.1 Sampling Resistance

Any initial state (C^L, X^L, Y^L) of ASG which can produce Z^m, a given prefix of the output sequence of ASG, is called a preimage of Z^m. The sampling resistance is

defined as 2^{-m} where m is the maximum value for which we can efficiently produce all preimages of m-bit outputs. As will be shown in this subsection, the low sampling resistance of ASG is an essential ingredient for our attack. Let $\mathcal{A}(Z^m)$ denote the set of all preimages of Z^m. Based on the divide-and-conquer linear consistency attack, introduced in Sect. 2, we can compute $\mathcal{A}(Z^m)$ as in Alg. 1.

Algorithm 1 Sampling of ASG

Input: Output sequence Z^m of m bits.
Output: Find $\mathcal{A}(Z^m)$ with all preimages of Z^m.
 1: Initially, set $\mathcal{A}(Z^m) = \emptyset$.
 2: **for all** non-zero initial states C^L **do**
 3:     Set $\mathcal{X} = \mathcal{Y} = \emptyset$.
 4:     Compute C^m, a prefix of length m of the output sequence of $\mathsf{LFSR_C}$.
 5:     Based on C^m, split up Z^m into X^w and Y^{m-w}, where $w = \mathrm{wt}(C^m)$.
 6:     Add all (non-zero) X^L to $\mathcal{X}$, if $\mathsf{LFSR_X}$ can generate X^w.
 7:     Add all (non-zero) Y^L to $\mathcal{Y}$, if $\mathsf{LFSR_Y}$ can generate Y^{m-w}.
 8:     For all $X^L \in \mathcal{X}$ and $Y^L \in \mathcal{Y}$, add (C^L, X^L, Y^L) to the set $\mathcal{A}(Z^m)$.
 9: **end for**

Let us discuss the complexity of Alg. 1. If $|\mathcal{A}(Z^m)| \leq 2^L$, then the overall complexity is 2^L, because the complexity of Steps 3 to 8 are $\mathcal{O}(1)$. On the other hand, if $|\mathcal{A}(Z^m)| > 2^L$, then Steps 3 to 8 introduce additional solutions, and overall complexity is about $|\mathcal{A}(Z^m)|$. The following statement is given under the assumption of balancedness, *i.e.* the average number of preimages of ASG for any output Z^m is about 2^{3L-m}, where $m \leq 3L$.

Statement 1 *Time complexity of Alg. 1 is $C_T = \mathcal{O}(\max(2^L, 2^{3L-m}))$.*

With the previous definition of sampling resistance, this algorithm can be considered as an efficient sampling algorithm iff $|\mathcal{A}(Z^m)| \geq \mathcal{O}(2^L)$ or equivalently $m \leq 2L$. That is, the sampling resistance of ASG is about 2^{-k} with $k = 2L$ the total length of the two stop/go LFSR's.

A related problem is how to find a multiset $\mathcal{B}$ with T uniformly random independent elements of $\mathcal{A}(Z^m)$. We suggest to modify Alg. 1 as follows: $\mathcal{A}(Z^m)$ is replaced by $\mathcal{B}$ and T is added as another input parameter. In Step 2, a uniform random (non-zero) initial state C^L is chosen, and Steps 3 to 8 are not modified. The new Steps 2 to 8 are then repeated, until T preimages have been found. This modified algorithm will be referred to as Alg. 1B. We assume correctness of the algorithm, *i.e.* the preimages found with Alg. 1B are uniformly random elements of $\mathcal{A}(Z^m)$ (for which we will give experimental evidence). The following statement is presented under the assumption that the average number of preimages of ASG for any output Z^m, given some fixed initial state of $\mathsf{LFSR_C}$, is about 2^{2L-m}, where $m \leq 2L$.

Statement 2 *Time complexity of Alg. 1B is $C_T = \mathcal{O}(T)$ for $m \leq 2L$, and $C_T = \mathcal{O}(\min(2^L, T2^{m-2L}))$ for $m > 2L$, where $1 \leq T \leq \mathcal{O}(2^{3L-m})$.*

4.2 Conditional Distribution of the Initial States

With the sampling algorithm described in Sect. 4.1, we can find T random preimages of an output sequence Z^m. The natural question which arises is *how large should T*

be so that our subset contains the correct initial state of one of the LFSR*'s, let say* LFSR$_X$? The answer is related to the conditional distribution of different initial states of LFSR$_X$ which can produce a given segment of length m of the output sequence of the ASG. Consider the following two general propositions (with proofs in Appendix B):

Proposition 1. *Let $X_0, \ldots, X_T$ be a sequence of i.i.d. random variables, defined over the finite set $\{s_1, \ldots, s_N\}$ with probability distribution $p := (p_1, \ldots, p_N)$ and $p_i := \Pr(X_j = s_i)$. Then, the probability $P := \Pr(X_0 \in \{X_1, \ldots, X_T\})$ that a realisation of $X_1, \ldots, X_T$ contains a realisation of X_0 is*

$$P = 1 - \sum_{i=1}^{N}(1-p_i)^T p_i \ . \tag{2}$$

Proposition 2. *Let $H := -\sum_{i=1}^{N} p_i \log_2(p_i)$ be the entropy of random variable X_j. With about $T = 2^H$, the probability $\Pr(X_0 \in \{X_1, \ldots, X_T\})$ is significant.*

To apply these propositions to the situation of ASG, let $\mathcal{A}_X(x, Z_m)$ be a subset of $\mathcal{A}(Z_m)$, defined by $\{(u, v, w) \in \mathcal{A}(Z_m) \mid v = x\}$. The conditional probability for a fixed initial state x of LFSR$_X$ is then defined by $p_X(x|Z^m) = |\mathcal{A}_X(x, Z_m)|/|\mathcal{A}(Z_m)|$. Consequently, we need to draw about $T = 2^{H_X}$ uniformly random elements of $\mathcal{A}(Z^m)$ to include the correct initial state of LFSR$_X$ where H_X is the conditional entropy of the initial state of the LFSR$_X$ given Z^m, defined by

$$H_X(Z^m) = -\sum_x p_X(x|Z^m) \cdot \log_2 p_X(x|Z^m) \ . \tag{3}$$

The same argument applies to LFSR$_Y$, and the symmetry of ASG motivates the simplification $H := H_X = H_Y$ (if not mentioned otherwise). Another natural question is the expected number of different elements Q drawn in this sample of size T. This is related to the Coupon-Collector Problem with non-uniform distribution. However, we can assume that $Tp_i \ll 1$, which results in $Q \approx T$.

Remark 3. Any adversary who *would know* the distribution p_X could try to recover the unknown initial state of LFSR$_X$ by considering the most probable initial state first, then the second most probable one and so on. Here, to cope with *unknown* distribution p_X, we simulate it by choosing uniformly random elements of $\mathcal{A}(Z^m)$ (where element x is chosen with probability $p_X(x|Z^m)$). This procedure is similar to [14] in which an equivalent description of the underlying cipher was used, for which the initial states were no longer equiprobable.

Remark 4. As mentioned in Sect. 2.4, it has been suggested in [7] to take advantage of the posterior probabilities of the individual bits of the regularly clocked LFSR$_X$ and LFSR$_Y$ sequences, when conditioned on a given segment of the output sequence for faster reconstruction of LFSR initial states. Our attack can be considered as a generalization of this attack in which we take advantage of the posterior probabilities of the initial states rather than individual bits, when conditioned on a given segment of the output sequence. Although unlike [7] we are able to give an estimation for the time and data complexities of our attack, a theoretical analysis of the conditional entropy of the initial states remains an open problem, see Sect. 5.1.

4.3 Description of the Attack

In the basic edit probability correlation attack on the ASG [6,10], the edit probability is computed for each of the 2^L possible initial states of LFSR$_X$ (given a segment of length $n \approx 40L$ of the output sequence of the ASG) to find the correct initial state. This is repeated also for LFSR$_Y$, and finally the initial state of LFSR$_C$ can be recovered. In our improved attack, we take the output sequence Z^m into account to compute a smaller multiset $\mathcal{B}$ of candidates of initial states, which is of size T and contains the correct initial state of LFSR$_X$ (resp. LFSR$_Y$) with some probability P, see Prop. 1. The multiset $\mathcal{B}$ is constructed with Alg. 1B. In Alg. 2, we give a formalisation of this attack.

Remark 5. One would think that it is better to compute the edit probability between Z^n and only the LFSR output sequence of all *distinct* initial states suggested by multiset $\mathcal{B}$ to avoid processing the same initial state several times. However, this needs memory of $\mathcal{O}(|\mathcal{B}|)$ and extra effort to keep the track of the non-distinct initial states. Since $|\mathcal{B}| \approx T$ the achieved gain is negligible and therefore we alternatively compute the edit probability at the time where a preimage is found.

Algorithm 2 Attack on ASG

Input: Parameters T, m, n, output Z^n.
Output: Recover the initial state of ASG with some success probability δ.
 1: Given the segment Z^m, find T preimages using Alg. 1B.
 2: Compute the edit probability between Z^n and the output sequence for each suggested initial state.
 3: Choose the most probable candidates for LFSR$_X$ resp. LFSR$_Y$.
 4: Recover LFSR$_C$ and verify the validity, see Sect. 2.3.

Parameters for the Entropy The complexity of the attack is related to the conditional entropy H. However, for large instances of ASG, the conditional probabilities and hence H are unknown. To be able to evaluate our attack and give an explicit expression for the data and time complexities, we need to know the relation between conditional entropies H and all parameters which can possibly affect them. The parameters are LFSR's feedback polynomials and the output prefix Z^m, which implicitly include the lengths of LFSR's and output segment length as well. In our simulations we noticed that feedback polynomials have almost no effect and the only important parameters are LFSR lengths L, the size of the output segment m (as larger values of m reduce uncertainty about the correct preimage), and the weight w of the output segment Z^m or the weight w of the first derivative of the output segment Z^m (as will be shown in our simulations). The entropy is significantly reduced if $|\text{wt}(Z^m)/m - 0.5| \gg 0$ (*i.e.* many zeros or ones) or if $\text{wt}(\dot{Z}^m)/m \ll 0.5$ (*i.e.* many runs of zeros or ones). This can be explained by the fact that a biased output segment results in a biased LFSR segment, and we will refer to such outputs as *weak* outputs. In Sect. 5.1, we will predict the average value of H depending on these parameters using some regression analysis, hence $\text{E}(H) = f(L, m, w)$.

Time Complexity Let us discuss time complexity of Alg. 2. According to Prop. 2, we set $T = 2^H$. Complexity of Step 1 is described in Statement 2. Computation of the edit probability distance of a single preimage takes about $\mathcal{O}(L^2)$, hence complexity of Step 2 is at most $\mathcal{O}(L^2 T)$. Finally, the complexity of Step 4 is $\mathcal{O}(2^{0.27L})$, which can be neglected here.

Statement 3 *Time complexity of Alg. 2 is about $C_T = \mathcal{O}(L^2 2^H)$ for $m \leq 2L$, and $C_T = \mathcal{O}(2^{H+m-2L})$ for $m \gg 2L$.*

This should be compared to the attack by Golic *et al.* of complexity $C_T = \mathcal{O}(L^2 2^L)$ using an output sequence of length about $C_D = 40L$ which was described in Sect. 2.3, and Johansson's attack of complexity $C_T = \mathcal{O}(L^2 2^{\frac{2}{3}L})$ using an output sequence of length $C_D = \mathcal{O}(2^{\frac{2}{3}L})$ as described in Sect. 3.

Data Complexity The parameter w has some influence on the data complexity of our attack. Once we know that the weight of Z^m is at most w or at least $m - w$, or that the weight of the first derivative of Z^m is at most w, a prefix of length about $n = 40L$ suffices to recover the initial states, see [10]. However, in order to obtain such an output segment Z_{t+1}^{t+m} for some t, the required amount of keystream bits is $C_D = 2^m (3 \sum_{i=0}^{w} \binom{m}{i})^{-1}$. This can be roughly approximated by $C_D = \mathcal{O}(2^{m(1-h(w/m))})$.

Success Probability The success probability δ of the attack depends on three events: 1) The probability that our multiset $\mathcal{B}$ of size $T = 2^H$ contains the correct initial state. 2) The probability that our prediction of the entropy gives at least H. 3) The success probability of the edit distance correlation attack. The first probability corresponds to P according to Eq. 2. The second probability comes from the fact that we use an estimation of the average value of H instead of the exact value of H.

5 Experimental Results

In this section, we give experimental results on ASG. We estimate the conditional entropy, give a detailed discussion of the complexity for different scenarios and present an example of an attack.

5.1 Distribution of Initial States

For specific instances of ASG, we investigate the distributions of initial states. Here, ASG is small enough such that an exact computation of initial states with Alg. 1 is feasible. We use registers of the same length, but our results do not significantly change if the lengths are pairwise coprime and about the same, as suggested in [15]. The following example has illustrative character: First, we compute the distributions for one fixed output sequence. Second, the block size m is varied for average-weighted output sequences. Third, an output sequences of low weight is investigated.

Example 1. Consider a setup with $L = 20$ and some randomly chosen primitive feedback polynomials. Fix a random output sequence Z^m of $m = 40$ bits according to

$$Z^m = \texttt{1110110110100101010000100100101011000110} \ .$$

The number of preimages is $|\mathcal{A}(Z^m)| = 1\,046\,858 = 2^{20.00}$, and the entropies are $H_C = 17.49$, $H_X = 17.32$, and $H_Y = 17.34$. If this output is padded by the 2 additional bits $\texttt{01}$, then the number of preimages becomes $|\mathcal{A}(Z^m)| = 264265 = 2^{18.01}$ and the entropies are $H_C = 16.26$, $H_X = 16.46$, and $H_Y = 16.45$. On the other hand, consider the following output sequence for $m = 40$ and with weight $w = 7$,

$$Z^m = \texttt{0001010000100000000110000001000100000000} \ .$$

The number of preimages for this low-weight output sequence is $|\mathcal{A}(Z^m)| = 1117725 = 2^{20.09}$, with entropies $H_C = 17.39$, $H_X = 12.24$, and $H_Y = 12.63$. $\qquad\square$

Let us discuss this example. The number of preimages is about 2^{60-m}, as expected. In all three registers, the entropy is not maximal for the random output sequence of size $m = 40$. This may be explained by the fact that sequences are not fully random, as they satisfy a linear recurrence. In the stop/go LFSR's, the entropy is strongly reduced for outputs of low weight, without any losses in the number of preimages. Notice that H_C does not depend on the weight of the output, which is optimal for efficient sampling.

In the following we will focus on the case $m = 2L$. The entropy H of the stop/go LFSR's is exactly determined for different values of L and w, where $L = 5, \ldots, 21$ and $w = 0, \ldots, m$. More precisely, given some L (and randomly chosen primitive feedback polynomials), we determine the average entropy $\mathrm{E}(H)$ using 500 randomly chosen outputs of weight w. The values of $\mathrm{E}(H)/L$ as a function of w/m are shown in Fig. 1. The inner dots in this figure relate to smaller values of L, and the outer dots relate to larger values of L. A convergence behavior of $\mathrm{E}(H)/L$ for increasing L is perceivable from this figure.
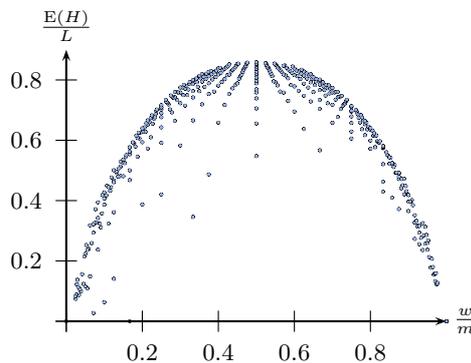

Fig. 1. $\mathrm{E}(H)/L$ versus w/m for all $0 \le w \le m$ and $5 \le L \le 21$.

It turns out that $\mathrm{E}(H)/L$ can be well approximated by a scaled binary entropy function, namely $\mathrm{E}(H)/L \approx \gamma \cdot h(w/m)$ with $0 < \gamma \le 1$ depending on L. Notice that

$\gamma = \max_w(\mathrm{E}(H)/L)$, which can be well approximated by $\gamma \approx 1 - 1/(0.19L + 3.1)$, see Appendix C. Consequently, with this regression analysis, the average value of the entropy is approximated by:

$$\mathrm{E}(H) \approx \gamma(L) \cdot L \cdot h\left(\frac{w}{m}\right) \tag{4}$$

$$\gamma(L) \approx 1 - \frac{1}{0.19L + 3.1} \ . \tag{5}$$

In the case $w = \mathrm{wt}(\dot{Z}^m)$ the shape is not symmetric, however it seems that for $w/m < 0.5$ for a fixed L the figures of $E(H)$ versus w/m are well comparable regardless of what w represents ($w = \mathrm{wt}(\dot{Z}^m)$ or $w = \mathrm{wt}(Z^m)$), see Appendix C. For $m > 2L$, the expected entropy does not correspond to this functional form anymore. The maximum of $H(w/m)$ decreases linearly with m, but the graph of $\mathrm{E}(H)/L$ versus w/m is broader compared to $h(w/m)$, which means that a reduction of the entropy requires an output of very low weight. We do not further investigate this scenario.

5.2 Complexity of the Attack

Our attack allows different time/data trade-offs. We describe the complexity of our attack for $m = 2L$ and different values of parameters L and w. According to Statement 3, time complexity of our attack is $C_T = \mathcal{O}(L^2 2^H)$. Including the approximation for H, we obtain $C_T = \mathcal{O}(L^2 2^{\gamma L h(w/m)})$. Given an random output sequence, the complexity of our attack is $C_T = \mathcal{O}(L^2 2^{\gamma L})$. In this case the data complexity is minimal and our attack should be compared to the attack by Golic et al. [10] which shows an improvement by a factor $2^{(1-\gamma)L}$. In the limit $\gamma \to 1$ (hence for $L \to \infty$), our attack reduces to the previous attack. However for moderate values there is some gain. For example, we expect $\gamma = 0.945$ for $L = 80$, which gives an improvement of a factor $2^{4.4}$.

Reduced complexity attacks can be mounted by using weak outputs. This can be compared to the attack by Johansson [13]. Asymptotical data complexity of our attack becomes $C_D = \mathcal{O}(2^{m(1-h(w/m))})$. Similar to what we do in Sect. 3, the optimised complexity is achieved if time and data complexities are almost equal. Considering only the exponential terms and $\gamma = 1$, this happens when $h(w/m)L = m(1-h(w/m))$, that is $h(w/m) = 2/3$ and hence $w \in \{0.174m, 0.826m\}$. The asymptotical complexities become $C_T = \mathcal{O}(L^2 2^{\frac{2}{3}L})$ and $C_D = \mathcal{O}(2^{\frac{2}{3}L})$, which is identical to the complexities of the attack by Johansson, see Sect. 3. However, compared to the simple attack in [13], it is clear that our attack allows for more flexibility in the structure of the output sequence: the weight can be arbitrary, we can also use outputs of low weight derivative, and we do not need a hypothesis about the origin of the output bits. With a more subtle (non-asymptotical) investigation of the complexities, we show that data (and/or time) complexity can be significantly reduced with our attack. More precisely, we evaluate the exact complexities of our and Johansson's attack for reasonable value of L. Regarding Johansson's attack, consider the special point $M = \frac{2}{3}L$ in the time/data tradeoff curve. For $L = 80$, this gives $C_T = 2^{69.4}$ and $C_D = 2^{55.2}$. If we choose $w = 0.21m$ in our attack, we obtain about the same time complexity and

require only $C_D = 2^{42.3}$ data. This is an improvement of a factor $2^{12.9}$ (notice that a significant reduction can be expected even for $\gamma = 1$).

5.3 Example of an Attack

In this section, we present a large-scale example of a partial attack. We fix a random initial state in all three registers, such that the corresponding output sequence has weight $w = 0.174m$. Then, H is computed according to Eq. 4 and 5. Alg. 1B is used to compute the multiset $\mathcal{B}$ of size $T = 2^H$, and we check if the correct initial state of the $\mathsf{LFSR_X}$ (resp. $\mathsf{LFSR_Y}$) is included in $\mathcal{B}$. This is repeated several times, in order to determine the success probability P. In addition, time complexity of Alg. 1B is measured experimentally: For each choice of C^L, the complexity is increased by the number of preimages found (and by one if no preimage can be found), see Remark 5. For $m = 2L$, this should be compared to $C_T = \mathcal{O}(T)$, see Statement 2. Notice that we do not implement the edit probability correlation attack and rely on the results of [10].

Example 2. Let $L = 42$ and fix a random initial state such that the corresponding output sequence Z^m of $m = 84$ bits has weight $w = 14$. The expected entropy becomes $H = 24.16$, we set $T = 2^H = 18\,782\,717$ and apply Alg. 1B. This is repeated 200 times, and the correct initial state of $\mathsf{LFSR_X}$ is found in 84 cases which shows a success probability of $P = 0.42$ for our algorithm. The average time complexity of the sampling algorithm is $2^{25.35}$. □

6 Conclusions

A reduced complexity attack on the Alternating Step generator (ASG) has been presented, the success of which has been confirmed experimentally. For comparison, the complexity of the best previous attack has been determined and described in closed form. Estimates of the overall complexity of our new attack are shown to improve the complexity of the previous attack. Our attack allows for greater flexibility in known output data constraints, and hence for lower data complexity, for being successful. The attack method demonstrates the usefulness of a quite general attack principle exemplified in the case of ASG: to exploit low sampling resistance and heavily biased inputs for outputs satisfying certain constraints.

Acknowledgments

References

1. A. Biryukov, A. Shamir, and D. Wagner. Real Time Cryptanalysis of A5/1 on a PC. In *FSE 2000: 1–18*.
2. D. Coppersmith, H. Krawczyk, and Y. Mansour. The Shrinking. Generator. In *CRYPTO 1993: 22-39*.
3. S. Fischer and W. Meier. Algebraic Immunity of S-boxes and Augmented Functions. To appear in *FSE 2007*.
4. J. Dj. Golic. Correlation Analysis of the Shrinking Generator. In *CRYPTO 2001: 440–457*.
5. J. Dj. Golic. Embedding probabilities for the Alternating Step Generator. In *IEEE Transactions on Information Theory 51(7): 2543–2553 (2005)*.
6. J. Dj. Golic and R. Menicocci. Edit Probability Correlation Attacks on Stop/Go Clocked Keystream Generators. In *J. Cryptology 16(1): 41–68 (2003)*.
7. J. Dj. Golic and R. Menicocci. Correlation Analysis of the Alternating Step Generator. In *Des. Codes Cryptography 31(1): 51–74 (2004)*.
8. J. Dj. Golic and R. Menicocci. Edit Distance Correlation Attack on the Alternating Step Generator. In *CRYPTO 1997: 499–512*.
9. J. Dj. Golic and R. Menicocci. Statistical Distinguishers for Irregularly Decimated Linear Recurring Sequences. In *IEEE Transactions on Information Theory 52(3): 1153–1159 (2006)*.
10. J. Dj. Golic and R. Menicocci. Edit Probability Correlation Attack on the Alternating Step Generator. In *Sequences and Their Applications - SETA 1998*.
11. C. G. Günther. Alternating Step Generators Controlled by De Bruijn Sequences. In *EUROCRYPT 1987: 5–14*.
12. S. Jiang and G. Gong. On Edit Distance Attack to Alternating Step Generator. In *Other Combinatorial Structures: 85–92 (2003)*.
13. T. Johansson. Reduced Complexity Correlation Attacks on Two Clock-Controlled Generators. In *ASIACRYPT 1998: 342–356*.
14. W. Meier and O. Staffelbach. Analysis of Pseudo Random Sequence Generated by Cellular Automata. In *EUROCRYPT 1991: 186–199*.
15. A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone. Handbook of Applied Cryptography. In *CRC Press 1997*.
16. R. Sundaresan. Guessing Under Source Uncertainty. In *IEEE Transactions on Information Theory 53(1): 269–287 (2007)*.
17. K. Zeng, C. H. Yang, and T. R. N. Rao. On the Linear Consistency Test (LCT) in Cryptanalysis with Applications. In *CRYPTO 1989: 164–174*.

A Johansson's Reduced Complexity Attack on SG

Here, we give a closed form for the reduced complexity attacks on SG in [13]. In the approach B, the attacker waits for an output sequence of length $2M + 1$ of the form $(z_{t-M}, z_{t-M+1}, ..., z_{t-1}, z_t, z_{t+1}, ..., z_{t+M-1}, z_{t+M}) = (0, 0, ..., 0, 1, 0, ..., 0, 0)$. Then, an exhaustive search is performed over all typical initial states $(x_1, x_2, ..., x_L)$ satisfying

$$\Pr(x_i = 0) = \begin{cases} 0 \text{ for } i = \lfloor \frac{L+1}{2} \rfloor \\ 3/4 \text{ for } 1 \leq |i - \lfloor \frac{L+1}{2} \rfloor| \leq 2M \\ 1/2 \text{ for } 1 \leq i < \lfloor \frac{L+1}{2} \rfloor - 2M, \lfloor \frac{L+1}{2} \rfloor + 2M < i \leq L \end{cases} \qquad (6)$$

for the LFSR from which the output sequence of the SG is derived. The time and data complexities are $C_T = L^2 2^{L-4M-1} \binom{4M}{M}$ and $C_D = 2^{2M}$, with the restriction $2M + 1 \leq L$. Assuming $2M = \alpha L$ with $\alpha \leq 1$, again ignoring the polynomial and constant terms and equaling the time and data complexities, we have $L - 2\alpha L + 2\alpha L h(0.25) = \alpha L$ which shows $\alpha = 1/(3 - 2h(0.25)) \approx 0.726$. Thus in the best case, the complexities of this attack are $C_T = \mathcal{O}(L^2 2^{0.726L})$ and $C_D = \mathcal{O}(2^{0.726L})$, where $M = 0.363L$. For approach C, the gain is negligible when L is increased.

B Proofs

B.1 Proof of Prop. 1

The probability P can be expressed as

$$\Pr(X_0 \in \{X_1, \ldots, X_T\}) = 1 - \Pr(X_0 \neq X_j, \ 1 \leq j \leq T)$$

$$= 1 - \sum_{i=1}^{N} \Pr(X_0 \neq X_j, \ 1 \leq j \leq T \mid X_0 = s_i) \cdot \Pr(X_0 = s_i)$$

$$= 1 - \sum_{i=1}^{N} \Pr(s_i \neq X_j, \ 1 \leq j \leq T) \cdot \Pr(X_0 = s_i)$$

$$= 1 - \sum_{i=1}^{N} (1 - p_i)^T p_i \ .$$

$\square$

B.2 Proof of Prop. 2

From Prop. 1 we have $\Pr(X_0 \in \{X_1, \ldots, X_T\}) = 1 - \sum_{i=1}^{N} (1 - p_i)^T p_i$. With the assumption $Tp_i \ll 1$, we obtain $(1 - p_i)^T \approx 1 - Tp_i$, which gives the approximation $\Pr(X_0 \in \{X_1, \ldots, X_T\}) \approx 1 - \sum_{i=1}^{N} (1 - Tp_i)p_i = T\sum_{i=1}^{N} p_i^2$. Assuming $\Pr(X_0 \in \{X_1, \ldots, X_T\}) \approx 1$, we have $T \approx 1/\sum_{i=1}^{N} p_i^2$, or equivalently $T \approx 2^G$ with $G := -\log_2 \sum_{i=1}^{N} p_i^2$. This can be compared with the entropy function H. Both H and Q are approximated with a multivariate Taylor series of order 2 at the point p_0, such that $p_i = p_0 + \varepsilon_i$. If T_2 denotes the second order part, this gives

$$T_2(H) = \frac{Np_0}{\ln 2} - \frac{1}{\ln 2} - \log_2 p_0$$

$$T_2(G) = \frac{2}{\ln 2} - \frac{2}{Np_0 \ln 2} - \log_2 N - \log_2 p_0^2 \ .$$

Now let $p_0 := 1/N$, then we have $T_2(H) = \log_2 N$ and $T_2(G) = -\log_2 N + 2\log_2 N = \log_2 N$. Consequently, the difference becomes $T_2(H) - T_2(G) = 0$, hence $H = G$ of order 2 on the points $p_i = 1/N$. $\square$

Remark 6. The quantity $G := -\log_2 \sum_{i=1}^{N} p_i^2$ is the Rényi entropy of order 2. It is known that guessing a random value, drawn from a *known* nonuniform probability distribution, on average requires the number of steps related to the Rényi entropy of order 2, *e.g.* see [16] or references therein. The Prop. 2 shows that this is still true when the distribution is not directly known but can be simulated. One can directly use this entropy instead of Shannon entropy which is only an approximation in this regard, however, we prefer to use the better known Shannon entropy. For the case $p_i = 1/N$ we have $G = H = \log_2 N$, hence $T = N$ and $P = 1 - \sum_{i=1}^{N} (1 - 1/N)^N (1/N)$. For $N \gg 1$ we have $(1 - 1/N)^N \approx e^{-1}$ which shows $P \approx 1 - e^{-1} \approx 0.63$. We guess that in general we have $P \geq 1 - e^{-1}$. Our extensive simulations for several distributions verifies this conjecture. $\square$

C Additional Figures

Fig. 2 shows some additional figures of the average entropy, together with our approximations using nonlinear regression. Fig. 3 compares the average value of the entropy as a function of the weight of the output sequence and as a function of the weight of the derivative of the output sequence.
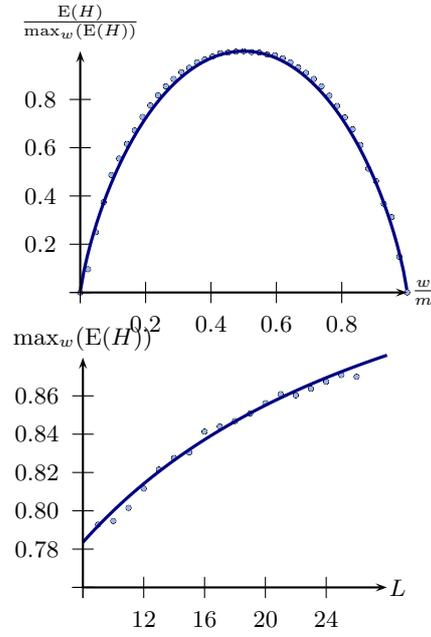

Fig. 2. Top: $\mathrm{E}(H)/(\max_w(\mathrm{E}(H)))$ versus w/m for $L = 21$, approximated by the entropy function. Bottom: $\max_w(\mathrm{E}(H))$ versus L, approximated by $\gamma(L)$.
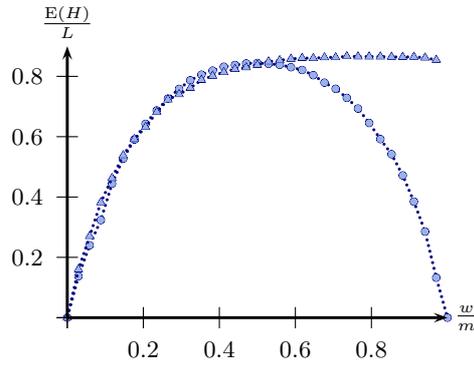

Fig. 3. $\frac{\mathrm{E}(H)}{L}$ versus w/m for $L = 17$ in two cases: $w = \mathrm{wt}(\dot{Z}^m)$ and $w = \mathrm{wt}(Z^m)$.