# The Editing Generator and Its Cryptanalysis

Guang Gong and Shaoquan Jiang
Department of Electrical and Computer Engineering
University of Waterloo
Waterloo, Ontario N2L 3G1, CANADA
E-mail: {ggong,jiangshq}@calliope.uwaterloo.ca

## Abstract

*In this paper*[1]*, we present a new pseudo-random sequence generator, constructed by using two ternary linear feedback shift registers (LFSR). This new generator is called an* editing generator *which is a combined model of the clock-control generator (viewed as insertion) and the shrinking generator (viewed as deletion). It is shown that the editing generator has good properties of randomness, such as large periods, high linear spans, large ratio of linear span per symbol, and small unbias of occurrences of symbols. Given that a portion of a key stream is exposed, two different attacks for recovering the initial states of the two LFSRs are analyzed, which are the parity-check attack, the unconstraint embedding attack (analogous to the binary case). The analysis shows that the edited sequences resist to both the attacks.*

**Index words:** *Pseudo-random sequences, LFSR, cryptanalysis.*

## 1. Introduction

Pseudo-random sequences are commonly used in cryptology. It can be implemented as either key stream generators in stream cipher systems or pseudo-random number generators in session key generators. Pseudo-random sequence generators (PRSG) based on linear feedback shift registers (LFSR) are most common structures in practice due to their efficient hardware implementation. There are two ways to construct LFSR based PRSGs. One way is to apply boolean functions to a set of $n$ LFSRs, the resulting sequences are called *filtering function sequences* if the LFSRs are equal or *combinatorial function sequences* if they are distinct (see [10]). The other way is to use one LFSR to control outputs of another LFSR (may be the two LFSRs are the same) by using two different control models, One is the *clock-control generators* [1][8] (stop/go or Gunther's generalized form [7]), and the other is the *shrinking generators* [2], including self-shrinking generators [9]. There are extensive researches on attacks on the clock-control generators and/or the shrinking generators, here we list some of them [3] [5][12] [4].

In this paper, we will present a new type of pseudo-random sequence generators, constructed by using two ternary LFSRs. The resulting generator is called an *editing generator*. This is a combined model of the clock-control generator (viewed as insertion) and the shrinking generator (viewed as deletion). The resulting sequences are ternary sequences. We will show that the editing generators have good properties of randomness. For example, they have large periods, high linear spans, large ratio of linear span per symbol, and small unbias of occurrences of symbols. Although these properties may not guarantee security of the editing generators for cryptographic applications, it is necessary to resist to the attacks from the application of the Berlekamp-Massey algorithm. Following satisfaction in these results on the new generator, we will look at its security under a randomized model. We will consider two different attacks for recovering the initial states of two LFSRs, given that a portion of a key stream is exposed. The first one is the parity-check attack in search of the initial states of two LFSRs. The complexity of this algorithm will serve as a benchmark in comparison with the other attack. The second attack is analogous to the attack to the clock-control generator proposed in [3]. Our analysis shows that the edited sequences resist to both of these attacks.

This paper is organized as follows. In Section 2, we will introduce the construction of the editing generator and derive some basic properties. In Section 3, we will present the properties of randomness of the editing generator. The security of the editing generator under a randomized model will be discussed in section 4.

We conclude this section with some preliminaries on m-sequences over $Z_3$ which will be used in this paper. For

[1] A slightly extended version of this paper appears at [6]

more results on m-sequences over non-binary fields, the reader is referred to [11].

### A. Some Basic Concepts and Notations

Let $A = \{a_i\}$ be a sequence over $Z_3$, i.e., $a_i \in Z_3$. We say that $r > 0$ is a *period* of $A$ if and only if $a_i = a_{i+r}, i = 0, 1, \cdots$. The smallest positive number $r$ satisfying $a_i = a_{i+r}, i = 0, 1, \cdots$ is called the *least period* of $A$. If $A$ is a periodic sequence of period $r$, we also write $A = (a_0, a_1, \cdots, a_{r-1})$, a vector in the $r$ dimensional linear space $Z_3^{(r)} = \{(a_0, a_1, \cdots, a_{r-1}) | a_i \in Z_3\}$.

Let $f(x) = x^n - c_{n-1}x^{n-1} - \cdots - c_1 x - c_0, c_i \in Z_3$. If the elements of $A$ satisfies the following linear recursive relation:

$$a_{n+k} = c_{n-1}a_{n-1+k} + \cdots + c_1 a_{1+k} + c_0 a_k, k = 0, 1, \cdot \quad (1)$$

then $A$ is said to be an *LFSR sequence* generated by $f(x)$ and $f(x)$ is called a *characteristic polynomial* of $A$. The $n$-tuple $(a_0, a_1, \cdots, a_{n-1})$ is called an *initial state* of $A$. It is clear that $A$ is completely determined by $f(x)$ and the initial state. We say that the first $N$ elements of $A$ are generated by LFSR $f(x)$ if (1) is true for $k = 0, 1, \cdots, N - n - 1$. The characteristic polynomial of $f(x)$ with the smallest degree is called the *minimal polynomial* of $A$. The degree of the minimal polynomial of $A$ is called *linear span or linear complexity* of $A$. If $f(x)$ is a primitive polynomial over $Z_3$ (i.e., $f(x)$ is irreducible and a root of $f(x)$ in the extension of $Z_3$ has order $3^n - 1$), then $A$ is called an $m$-sequence over $Z_3$ of degree $n$.

### B. The Properties of Randomness of Ternary $m$-Sequences

**Property 1** *Let $A$ be a ternary m-sequence of degree $n$.*

*(1) The least period of $A$ is $3^n - 1$.*

*(2) It satisfies the balance property, i.e., each non-zero element in $Z_3$ occurs $3^{n-1}$ times in one period of $A$ and zero element occurs $3^{n-1} - 1$ times in one period.*

*(3) It satisfies the following run property. In each period,*

- *(a) for $1 \le k \le n-2$, the runs of each element in $Z_3$ of length $k$ occur $2^2 3^{n-k-2}$ times,*
- *(b) the runs of each nonzero element of length $n-1$ occur once,*
- *(c) the runs of the zero element occur twice,*
- *(d) the run of each nonzero element of length $n$ occurs once.*

*(4) 2-level autocorrelation.*

*(5) Each nonzero $n$-tuple occurs exactly once.*

*(6) Let $d = (3^n - 1)/2$. We write $A = (A_1, A_2)$ where $A_1 = (a_0, a_1, \cdots, a_{d-1})$ and $A_2 = (a_d, a_{d+1}, \cdots, a_{d+d-1})$. Then $A_2 = 2A_1$. I.e., $a_{i+d} = 2a_i, i = 0, 1, \cdots, d-1$. In particular, if $a_i = 0$, then $a_{i+d} = 0$ for all $\forall i \ge 0$.*

## 2. Construction

In this section, we will introduce the construction of the editing generator and derive some basic properties on randomness. First, we take a look at the operations of the stop-and-go generator, a simple model of clock-control generators, and the shrinking generator.

Let $A = \{a_i\}$ and $B = \{b_j\}$ be two binary sequences generated by LFSR 1 and LFSR 2 respectively.

*The Stop-and-Go generator*

Output sequence: $S = \{s_k\}$ is given by

$$s_{k_j} = a_{i_j}, j = 0, 1, \cdots, \quad (2)$$

where $k_j = j, i_j = \sum_{t=0}^{j} b_t - 1$ and set $a_{-1} = 0$.

In other words, at time $j$, if $b_j = 1$, the generator outputs the current output bit of LFSR 1. Otherwise, the generator outputs the previous output bit (inserting).

*The Shrinking Generator:*

Output sequence: $S = \{s_k\}$, at time $j$, if $b_j = 1$, then the generator outputs the current bit $a_j$ of $A$. Otherwise this bit is discarded. I.e.,

$$s_{k_j} = a_{i_j}, j = 0, 1, \cdots, \text{ where } i_j = j, k_j = \sum_{t=0}^{j} b_t. \quad (3)$$

The new generator, the editing generator, is to combine these two operations together, in order to resist to the known attacks to those two control models when they are used separately.

**Definition 1** *Let $A = \{a_i\}$ and $B = \{b_j\}$ be two ternary sequences generated by LFSRs 1 and 2 respectively. The output sequence $S = \{s_k\}$ is determined by the following rules. At time $j$, if $b_j = 2$, the generator discards the current output element in LFSR 1; else if $b_j = 1$, the generator outputs the current output element in LFSR 1. Otherwise, the generator outputs the previous output element in LFSR 1. We will call it an* editing generator, *the resulting sequence an* edited sequence, *denoted as $Edit(A, B)$, the sequence $A$ a* base sequence, *and the sequence $B$ a* control sequence.

This definition can be written into the following pseudo-code:

*Algorithm 1 (Edit(A, B)): Algorithm for Generating Edited Sequences*

**Input**: $A$ and $B$, two ternary LFSR sequences
**Output**: an edited sequence $S$ of length $m$

1. **Procedure** $Edit(A, B, S, m)$
2. Set $a_{-1} = 0$
3. Set $k = j = i = 0$
4. **while** $(j < m)$ **do**
   (a) **if** $b_j = 2$, **then** set $i = i + 1$
   (b) **else if** $b_j = 1$, **then** set
   $s_k = a_i$; $k = k + 1$; $i = i + 1$
   (c) **else** set $s_k = a_{i-1}$; $k = k + 1$
   (d) $j = j + 1$
5. **return** $S$.

*Note.* The initial value of $a_{-1}$ can be taken as any value in $Z_3$. In this paper, we will use this initial setting throughout of this paper.

In the following, we will derive some basic properties of the edited sequences. From Algorithm $Edit(A, B)$, we know that the index $i$ will be incremented in cases (a) and (b) which correspond to $b_j \in \{1, 2\}$. The index $k$ will be incremented in cases (b) and (c), which correspond to $b_j \in \{0, 1\}$. Thus we have established the following assertion.

**Property 2** *With the notation in Definition 1, let $b'_j = 1$ if $b_j = 2$, otherwise $b'_j = b_j$. Then the elements of the edited sequence $S = \{s_k\}$ can be represented by*

$$s_{k_j} = a_{i_j} \text{ where } i_j = \sum_{t=0}^{j} b'_t - 1 \text{ and } k_j = \sum_{b_t \neq 2, 0 \leq t \leq j} b_t \tag{4}$$

**Definition 2** *With the notation in Property 2, we say that $a_{i_j}$ is* controlled *by $b_j$. In this case, we also call the integer triple $(i_j, k_j, j)$ a* match.

**Remark 1** *If the elements of $A$ and $B$ only take the values 1 and 2, from Property 2, we have $i_j = j$. According to (3), $Edit(A, B)$ is degenerated into a shrunk sequence. Similarly, if the elements of $A$ and $B$ only take the values 0 and 1, we have $k_j = j$. According to (2), $Edit(A, B)$ is degenerated into a clock-controlled sequence.*

**Remark 2** *The elements in an edited sequence are determined by the domain from which the base sequence $A$ takes its elements. The domain from which the control sequence $B$ takes its elements represents a set of strategies used for editing. Thus one may consider a general model for $Edit(A, B)$ where $A$ is a sequence whose elements taken from $Z_t = \{0, 1, \cdots, t-1\}$ and $B$ is a sequence whose elements, taken from $Z_r$. We will continue to investigate the editing generator in this general setting in the future.*

The properties of randomness of the edited sequence $Edit(A, B)$ depend on the base sequence $A$ and the control sequence $B$. In the rest of this paper, we will restrict ourselves into the case that both these two sequences are chosen as modified ternary $m$-sequences of degree $n$. The definition of a modified ternary $m$-sequence of degree $n$ is as follows.

**Definition 3** *If a sequence is constructed by adding one zero into one of two zero runs of length $n-1$ in a ternary $m$-sequence of degree $n$, then we call it a* modified ternary $m$-sequence of degree $n$, mm-sequence *for short.*

From Property 1, we know that a modified ternary $m$-sequence of degree $n$ has period $3^n$, each element in $Z_3$ occurs exactly $3^{n-1}$ times and each $n$-tuple in $Z_3^{(n)}$ occurs exactly once in one period of $3^n$.

**Property 3** *Let $S = Edit(A, B)$ where $A$ and $B$ are two $mm$-sequence of degree $n$. Then we have the following matches.*

$$i_{3^n} = 2 \cdot 3^{n-1} \quad \text{and } k_{3^n} = 2 \cdot 3^{n-1} \tag{5}$$

$$i_{2 \cdot 3^n} = 4 \cdot 3^{n-1} \quad \text{and } k_{2 \cdot 3^n} = 4 \cdot 3^{n-1} \tag{6}$$

$$i_{3 \cdot 3^n} = 2 \cdot 3^n \quad \text{and } k_{3 \cdot 3^n} = 2 \cdot 3^n \tag{7}$$

*Moreover, $2 \cdot 3^n$ is a period of $S$.*

**Proof** Since both of $A$ and $B$ are $mm$-sequences of degree $n$, then each element in $Z_3$ occurs exactly $3^{n-1}$ times in one period of $A$ and $B$, respectively. Notice that in Algorithm 1, $i$ will not be incremented when $b_j = 0$ and neither $k$ will when $b_j = 2$. Therefore we have that $i = 2 \cdot 3^{n-1}$ and $k = 2 \cdot 3^{n-1}$ when $j = 3^n$, which gives the two formulae in (5). The rest of the formulae follows by applying a similar argument. Thus the first assertion is true. From (7), we have $i = k = 2 \cdot 3^n$ when $j = 3^{n+1}$. At this time, $A$ and $B$ go back to the same initial states as $k = j = i = 0$. Thus we have $s_{k+2 \cdot 3^n} = s_k$ for all $k \geq 0$, which establishes the assertion 2. $\diamondsuit$

## 3. The Properties of Randomness

In this section, we will derive the least period of the edited sequence $Edit(A, B)$, a large lower bound for the linear span, ratio of linear span to period, and occurrences of symbols. We will keep the notation in Section 2.

### 3.1. Period and Linear Span

**Theorem 1** *Let $S = \{s_k\}$ be the edited sequence $Edit(A, B)$ where both $A = \{a_i\}$ and $B = \{b_j\}$ are $mm$-sequence of degree $n$. Then $Per(S)$, the least period of $S$, is either $2 \cdot 3^n$ or $3^n$.*

**Proof** From Property 3, we know that $Per(S)|2 \cdot 3^n$. If $Per(S)$ is a factor of $2 \cdot 3^n$ or $3^n$, then $Per(S)|2 \cdot 3^{n-1}$. Since $B$ is a $mm$-sequence of degree $n$, then each $n$-tuple occurs exactly once in one period of $B$. Thus we can suppose that $b_j = b_{j+1} = \cdots = b_{j+n-1} = 1$ for some $j \geq 0$. Since $a_{i_j}$ is controlled by $b_j$, then $a_{i_j+r}$ is controlled by $b_{j+r}$ for $r = 0, 1, \cdots, n-1$. Notice that from Property 3, we know that $(2 \cdot 3^{n-1}, 3^n)$ is a match. Thus $b_{j+3^n+r}$ controls $a_{i_j+2\cdot 3^{n-1}+r}$ for $r = 0, 1, \cdots, n-1$. Notice $b_{j+3^n}$ is the element after one period of length $3^n$ of $B$, which is just $b_j$. Therefore, the elements in a period of $2 \cdot 3^{n-1}$ of $S$ are generated under the control of a shift of $B$: $(b_j, b_{j+1}, \cdots, b_{j+3^n-1})$. Thus we have $a_{i_j+r} = a_{i_j+2\cdot 3^{n-1}+r}$, $r = 0, \cdots, n-1$. Note that $3^{n-1} > n$ when $n > 1$. Hence $2 \cdot 3^{n-1} + n - 1 + 1 < 3^n$. Therefore, all elements in the set of $\{a_{i+r}, a_{i+2\cdot 3^{n-1}+r} | r = 0, \cdots, n-1\}$ are within one period of $A$. Consequently, the two equal $n$-tuples $(a_i, \cdots, a_{i+n-1})$ and $(a_{i+2\cdot 3^{n-1}}, \cdots, a_{i+2\cdot 3^{n-1}+n-1})$ appear in one period of $A$ which is a contradiction. Thus we have that $Per(S) = 2 \cdot 3^n$ or $3^n$. $\diamondsuit$

In the next theorem, we will establish a lower bound for the linear span of the edited sequences.

**Theorem 2** *Let $S = Edit(A, B)$, where $A$ and $B$ are $mm$-sequences of degree $n$. Then $LS(S)$, the linear span of $S$, is lower bounded by*

$$LS(S) > 3^{n-1}.$$

**Proof** From Theorem 1, $f(x) = x^{2\cdot 3^n} - 1$ is a characteristic polynomial of $S$ over $Z_3$. We can write that $f(x) = (x^2-1)^{3^n}$. Let $m(x)$ be the minimal polynomial of $S$. Then $m(x) = (x-1)^a(x+1)^b$ for some integer $a \leq 3^n$ and $b \leq 3^n$. If $deg(m(x)) \leq 3^{n-1}$, then $a \leq 3^{n-1}$ and $b \leq 3^{n-1}$. Consequently, $m(x)|(x^2-1)^{3^{n-1}} = x^{2\cdot 3^{n-1}} - 1$. Thus the least period of $S$ is less than or equal to $2 \cdot 3^{n-1}$, which is a contradiction with Theorem 1. So $LS(S) > 3^{n-1}$. $\diamondsuit$

**Corollary 1** *With the notation in Theorem 2, let $\eta$ be the ratio of the linear span to the least period of the edited sequence (represents* linear span per symbol*). Then $\eta$ satisfies*

$$\eta > \begin{cases} 1/6 & \text{if } Per(S) = 2 \cdot 3^n \\ 1/3 & \text{if } Per(S) = 3^n. \end{cases}$$

### 3.2. Occurrences of Symbols

In the following theorem, we will establish a bound for frequency of each element in $Z_3$ occurs in an edited sequence.

**Theorem 3** *Let $S = Edit(A, B)$ where both $A = (a_0, a_1, \cdots, a_{3^n-1})$ and $B = (b_0, b_1, \cdots, b_{3^n-1})$ are $mm$-sequences of degree $n$. Then each element in $Z_3$ occurs at least $3^{n-1}$ times in one period of length $2 \cdot 3^n$ of $S$.*

**Proof** Let $d = (3^n - 1)/2$. We consider that $S = (s_0, s_1, \cdots, s_{2\cdot 3^n-1})$. Without loss of generality, we may suppose that the zero $n$-tuple appears in the first $d+1$ elements of $B$. Let $\tilde{D} = (\tilde{b}_0, \tilde{b}_1, \cdots, \tilde{b}_{3^{n-1}-1})$ be the resulting sequences from the first $d+1$ elements of $B$ by deleting all 0's, and let $\tilde{B} = (\tilde{b}_0, \tilde{b}_1, \cdots, \tilde{b}_{2\cdot 3^{n-1}-1})$ be the resulting sequence of $B$ by deleting all 0's. Then $\tilde{B}$ has a period of $2 \cdot 3^{n-1}$. Applying Property 1-(6), we have $\tilde{B} = (\tilde{D}, 2\tilde{D})$. We write $A = (A_0, A_1, A_2)$ where $A_i = (a_{i3^{n-1}}, a_{i3^{n-1}+1}, \cdots, a_{i3^{n-1}+3^{n-1}-1})$, $i = 0, 1, 2$. Let $\tilde{S} = Edit(A, \tilde{B})$, an edited sequence from the base sequence $A$ and the control sequence $\tilde{B}$. Then $\tilde{S}$ is the resulting sequence from $S$ by deleting all elements of $S$ which are contributed from the inserting operation. Thus all elements that occur in $\tilde{S}$ will occur in $S$ (in this case, we say that $\tilde{S}$ is a *subsequence* of $S$. In the following, we will show that each elements of $Z_3$ will occur $3^{n-1}$ times in $\tilde{S}$. Note that there is no zeros in $\tilde{B}$, then $i_j = j, j = 0, 1, \cdots$. In other words, $\tilde{b}_j$ controls $a_j$ for all $j \geq 0$. Therefore, $(\tilde{D}, 2\tilde{D}, \tilde{D}, 2\tilde{D}, \tilde{D}, 2\tilde{D})$ controls $(A_0, A_1, A_2, A_0, A_1, A_2)$ term by term. Consequently, each of $\tilde{D}$ and $2\tilde{D}$ controls $A_i, i = 0, 1, 2$ once. From $Edit(A_0, \tilde{D})$, we get the elements $a_j$ for those $j$ such that $\tilde{b}_j = 1, 0 \leq j < 3^{n-1}$. From $Edit(A_0, 2\tilde{D})$, we get the elements $a_j$ for those $j$ such that $\tilde{b}_j = 2, 0 \leq j < 3^{n-1}$. Thus $\tilde{S}$ contains the entire set of the elements of $A_0$. Similarly, from $Edit(A_i, \tilde{D})$ and $Edit(A_i, 2\tilde{D})$, we get all the elements of $A_i$, $i = 1, 2$. Thus the set of the elements of $\tilde{S}$ is equal to the set of the elements of $A$. Since each element in $Z_3$ occurs in $A$ exactly $3^{n-1}$ times and $\tilde{S}$ is a subsequence of $S$, then each element in $Z_3$ occurs in $S$ at least $3^{n-1}$. $\diamondsuit$

## 4. Security Analysis

In this section, we investigate the security of the edited sequence $S = Edit(A, B)$ where $A = \{a_i\}$ and $B = \{b_j\}$ are $mm$-sequences of degree $n$. We model both $A$ and $B$ as independent random sequences. For any sequence $U = \{u_i\}$, We denote $U_k^{m-1+k} = (u_k, u_{k+1}, \cdots, u_{k+m-1})$. We assume that a portion of the key stream sequence $S$, without loss of generality, $S_0^{m-1}$ is known. The primitive polynomials that generate $\{a_i\}$ and $\{b_j\}$ are also assumed to be

known. The objective of attacks is to find out the initial states of $\{a_i\}$ and $\{b_j\}$. In this section, we show that the initial state of $A$ ($B$ resp.) can be determined, given that of $B$ ( $A$ resp.). However, this does not mean this key stream generator is insecure since to find the initial state of $A$ ($B$ resp.) needs a search space of $3^n$.

### 4.1. Parity-check Attack

In this subsection, we show one can recover the initial states of $B$ and $A$ by searching for the initial state of $B$ and then carrying out consistency test. Suppose a segment of key stream $S_0^{m-1}$ is known. The attack is initiated by the following observations. First, we notice the output of $S$ sequentially corresponds to the control bit $b_j \in \{0, 1\}$ (i.e., control symbol 2 does not contribute an element in $S$). Thus if we compute a new sequence $B'$ by removing the symbol 2 from $B$, then we have $s_j$ is controlled by $b'_j$. Therefore, if we remove all the elements $s_j$ which corresponds to $b'_j = 0$, and let the resulting sequence be $S'$, then $S'$ is the subsequence of $S$ whose corrsponding control bit is 1. On the other hand, we notice sequence $A$ will clock if and only if control bit is 1 or 2. Thus the $i$th element in $S'$ is controlled by the $i$th element "1" of sequence $B$. It follows we can recover $a_j$ with probability one half. It is known if $x^t = c_0 + c_1 x + \cdots + c_{n-1}x^{n-1} \pmod{f(x)}$, then $a_t = c_0 a_0 + c_1 a_1 + \cdots + c_{n-1}a_{n-1}$. Thus for each determined bit of $A$ from above, we can write it in terms of its initial state. We then get approximately $n$ equations. If the initial state of $B$ is correctly guessed, then these equations should be consistent. Otherwise, it should be inconsistent. We write all the above idea into the following algorithm.

**Algorithm for Searching Sequence B**

1. Generate candidate sequence $B$ by guessing its initial state and then derive sequence $B'$ from $B$ by removing symbol 2.
2. Compute sequence $S'$ from $S_0^{m-1}$ by removing $s_i$ if $b'_i = 0, i = 0, \cdots, m-1$.
3. Let $t(i)$ be the $i$th index of symbol "1" in sequence $B$. Partially recover $A$ by setting $a_{t(i)} = s'_i$.
4. Write a linear equation in terms of the initial state of $A$ for each determined $a_i$ in Step 3. Check the consistency of the linear system. If it is not consistent, go to step 1. Otherwise, compute the initial state of $A$.

Now let us see the performance of this algorithm. In each round, step 1-3 need only $O(m)$. Step 4 needs $O(mn^3)$. Thus to be successful, it needs $O(3^n mn^3)$. In general, $m = O(n)$, it follows the running time is $O(3^n n^4)$, which is essentially the cost to search for the initial state of $B$.

### 4.2. Unconstraint Embedding Attack

In this subsection, we investigate the possibility to attack the editing generator via searching for the initial state of sequence $A$. This attack is a variant of the unconstraint embedding attack in [3]. According to the definition of the edited sequence, we can write $S = \{a_{i_k}\}$. We define $D = \{d_k\}$ whose elements are given by $d_k = i_k - i_{k-1}, k \geq 0$. $D$ is called a *jump sequence* of $S$. Thus we have $i_k = \sum_{i=0}^{k} d_i + i_{-1}$ where $i_{-1} = -1$. We notice $d_k \geq 0$. In literature, if a generator $S$ can be looked as irregular decimation of sequence $A$, one can recover the initial state of $A$ by generating a candidate sequence $A$ and testing whether it is correct by trying to embedding known key stream, say, $S_0^{N-1}$ into $A_0^{m-1}$ for properly chosen $m$. For attack to be successful, it is required that the embedding probability, denoted as $P_S(N, m)$, is exponentially approaching zero. Golic and L. O'Connor [3] computed $P_S(N, m)$ for the binary sequence case. Notice the above problem considers only the case $d_k > 0$. And out case is $d_k \geq 0$. To apply their technique, we first reduce the case $d_k \geq 0$ to the case $d_k > 0$. We start with a definition for a repeating operation.

**Definition 4** *Let $(i_k, k, j)$ be a match (i.e., $b_j$ controls $a_{i_k}$ to generate $s_k = a_{i_k}$). If $b_{j+1}$ controls $a_{i_{k+1}}$ to generate $s_{k+1} = a_{i_{k+1}}$ and $i_{k+1} = i_k$, then we say that element $s_{k+1}$ is generated by a* repeating operation, *or $s_{k+1}$ is a* repeated element.

**Lemma 1** *In one period of $2 \cdot 3^n$ of the edited sequence $S$, there are $2 \cdot 3^{n-1}$ that are generated by the repeating operation. In other words, one third of elements of $S$ are generated by the repeating operation.*

**Proof** From the definition, we see that each of $b_j$ and $b_{j+1}$ produces one output element. So they are not equal to 2. Since $i_{k+1} = i_k$, we have $b_{j+1} = 0$. Hence $(b_j, b_{j+1})$ can only be (1, 0) or (0, 0). It is easy to check both of these pairs will produce repeated elements. Furthermore, if $s_{k+1}$ is generated by the repeating operation, then the matched control pair $(b_j, b_{j+1})$ for $(s_k, s_{k+1})$ are $(b_j, b_{j+1})$=(0, 0) or (1, 0). Thus in one period of length $2 \times 3^n$ of $S$ , there are $\frac{2}{9} \times 3^{n+1} = 2 \times 3^{n-1}$ elements are generated by the repeating operation.

With the notation above, we will construct two new sequences from $S$. Let $\overline{S} = \{\overline{s}_k\}$ be the resulting sequence obtained from $S = \{s_k\}$ by deleting the elements in $\{s_k\}$ which are generated by the repeating operation. Let $\overline{D} = \{\overline{d}_k\}$ be the jump sequence of $\overline{S}$. Then $\overline{d}_k > 0$ for $k \geq 0$. Let $S^* = \{s^*_k\}$ be the resulting sequence obtained from $\{s_k\}$ by deleting $s_k$ when $s_k = s_{k-1}$. From the definitions, it is immediate that $S^*$ is a subsequence of $\overline{S}$. Note that, unfortunately, we also delete some $s_k$ which does not

come from the inserting operation although it satisfies that $s_k = s_{k-1}$.

The following proposition shows the embedding probability of $S_0^{*N-1}$.

**Proposition 1** *With the notation above, the the unconstraint embedding probability of $S_0^{*N-1}$ is given by*

$$\begin{aligned} P_{S^*}(N,m) &= \sum_{k=0}^{m-N} \left( \begin{array}{c} N-1+k \\ k \end{array} \right) 3^{-N-k} \\ &= 1 - 3^{-m} \sum_{k=0}^{N-1} \left( \begin{array}{c} m \\ k \end{array} \right) 2^{m-k} \end{aligned}$$

The proof of this result is similar to Theorem 3.1 in [3], so we omit it.

If we expect this attack to be successful, then $m$ must be large enough to guarantee that $S_0^{*N-1}$ can be embedded into the original $(a_0, \cdots, a_{m-1})$. This requests that $m \approx N\overline{d}$, where $\overline{d}$ is the average of the jump steps of $S_0^{*N-1}$ when it is considered as a subsequence of $\{a_i\}$. Notice that if $S_0^{*N-1}$ is derived from two periods of $\{a_i\}$ and its length is $l$, then $\overline{d} = \frac{2\times 3^N}{l}$. ($l$ is also a period of $S_0^{*N-1}$). Let $\overline{S}_0^{T-1}$ be constructed from $S_0^{2\cdot 3^n}$. Then $\overline{S}_0^{T-1}$ is a subsequence of $\{a_i\}$. Since $\{a_i\}$ is purely random, then $\overline{S}_0^{T-1}$ is also a purely random sequence. By Lemma 1, $T = (1-\frac{1}{3}) \times 2 \times 3^n = 4 \times 3^{n-1}$. From the construction of $S_0^{*l-1}$, each element, say $u$, in $S_0^{*l-1}$, starts a run of $u$'s in $\overline{S}_0^{T-1}$. Since $\overline{S}_0^{T-1}$ is purely random, from Property 1-(3) in Section 1, it follows that $l \approx 3 \times \sum_{k=3}^{[log_3 T]} \frac{2^2 \times T}{3^k} \approx 8 \times 3^{n-2}$. Therefore, $\overline{d} = \frac{9}{4}$. Thus $P_{S^*} = 3^{-m} \sum_{k=0}^{\frac{5m}{9}} \left( \begin{array}{c} m \\ k \end{array} \right) 2^k$. Notice $\left( \begin{array}{c} m \\ k \end{array} \right) 2^k < \left( \begin{array}{c} m \\ k+1 \end{array} \right) 2^{k+1}$ if $k < \frac{2m-1}{3}$. We have, $\sum_{k=0}^{\frac{5m}{9}} \left( \begin{array}{c} m \\ k \end{array} \right) 2^k \leq m 2^{mH(4/9)} \cdot 2^{5m/9} < m \cdot 3^{0.9760m}$. Thus $P_S^*(N,m)$ is exponentially approaching zero. Thus the attack is asymboliclly successful. However, it requires running time $O(3^n)$. Therefore, it in fact does not affect the security.

## Acknowledgement

Research for this work is supported by NSERC grant RG-PIN 227700-00.

## References

[1] T. Beth and F.Piper, The stop-and-go generator, *Advances in Cryptology, Eurocrypto'94*, vol. 209, Springer-Verlag, 1985, pp. 88-92.

[2] D. Coppersmith, H. Krawczys and Y. Mansour, The shrinking generator, *Advances in Cryptology-Crypt'93*, Lecture Notes in Computer Science, vol. 773, Springer-Verlag, 1994, pp. 22-39.

[3] J.D.Golic and L.O'Connor, Embedding and probability correlation attacks on clock-controlled shift registers, *Advances in Cryptology, Eurocrypto'94*, LNCS 950, A.D.Santis(Ed.), Spring-Verlag, pp.230-243, 1994.

[4] J.D. Golic and M.J. Mihaljevic, A generalized correlation attack on a class of stream ciphers based on the Levenshtein distance, *Journal of Cryptology*, 3(3): pp.201-212, 1991.

[5] J.D. Golic, Correlation analysis of the shrinking generator, Advances in Cryptology, *Advances in Cryptology-Crypt'2001*, Kilian (Ed.), LNCS 2139, pp.440-457, 2001.

[6] G. Gong and S. Jiang, Editting Generators and Their Cryptanalysis, CACR Technical Report, CORR 2002-30, University of Waterloo.

[7] C.G. Gunther, Alternating step generators controlled by de Bruijn sequences, *Advances in Cryptology, Eurocrypto'88*, Lecture Notes in Computer Science, vol. 304, Springer-Verlag, 1988, pp. 88-92.

[8] Dieter Gollman and W.G. Chambers, Clock-controlled shift registers: a review, *IEEE Journal on Selected Areas in Communications*, vol. 7, No. 4, May 1989, pp.525-533.

[9] W. Meier and O. Staffelbach, The self-shrinking generator, *Advances in Cryptology, Eurocrypto'94*, Lecture Notes in Computer Science, Springer-Verlag, 1994, pp. 205-214.

[10] R.A. Rueppel, Stream ciphers, *Contemporary Cryptology, the Science of Inormation*, IEEE Press, 1992, pp.65-134, Gustovos J. Simmons, editor.

[11] M.K. Simon, J.K. Omura, R.A. Scholtz and B.K. Levitt, *Spread Spectrum Communications Handbook*, McGraw-Hill, Inc., Revised version, 1994. Chapter 5.

[12] M.V. Zivkovic, An algorithm for the initial state reconstruction of the clock-controlled shift register, IEEE Trans. on Inform. Theory, vol.37, pp. 1488-1490, Sep. 1991.