LINEAR COMPLEXITY OF TRANSFORMED SEQUENCES

Harriet J. Fell[1]
College of Computer Science, Northeastern University
Boston, Massachusetts 02115 USA

Abstract:

This paper deals with the effect of bit change errors on the linear complexity of finite sequences. Though a change in a single bit can cause a large change in linear complexity, it is shown that on the average the change will be small even when many bits, e.g. 10%, are changed. General bijections on the set of sequences of length n are studied and tight bounds are found on the average difference in linear complexity between a sequence and its image. It is also shown that to change all sequences of length n into sequences with linear complexity less than $c(n)$ where $\lim_{n\to\infty} c(n)/n = 0$, at least $\frac{n-1}{n}2^n$ of the sequences must have close to half of their bits changed.

1 Introduction

The linear complexity of a finite sequence can change drastically if a single bit is changed or deleted. For example, the sequence $0, 0, \ldots, 0, 1$ of length n has maximal linear complexity, n, while deleting the last bit or changing it to a 0 results in a sequence of linear complexity 0. A shift register that generates a given sequence can be found, with Λ^2 operations, after seeing only 2Λ bits where Λ is the linear complexity of the sequence [2], so sequences of low linear complexity are not cryptographically secure. At the *Workshop on Stream Ciphers*, at Karlsruhe, Germany, January 9-12, 1989, W. Diffie suggested that for many finite sequences, it might be possible to find small linear feedback shift registers (LFSRs) that generate nearby sequences. That is, if we can tolerate some errors, we might find it easy to generate a sequence close enough to a given sequence for cryptanalytic purposes.

In this paper, we look at functions that take sequences of length n into sequences of length n and study the average difference in the linear complexity of a sequence and its image. We then apply these results to analyze the average change in linear complexity when errors, caused by changed bits, are introduced into a sequence.

[1]This work was supported in part by The Institut National de Recherche en Informatique et en Automatique, Rocquencourt, France.

2 Notation

We restrict our attention to sequences over the field with two elements, $\mathbf{Z_2}$. Let $\mathbf{S_n} = \{0,1\}^n$ be the set of all sequences of zeros and ones of length n. If $s \epsilon \mathbf{S_n}$ then $s = s_1, s_2, \ldots, s_n$ where each $s_i, \{i = 1, \ldots, n\}$ is either 0 or 1. We assume that $s_i, 1 \leq i \leq n$ are uniformly and independently distributed random variables, so that all $s \epsilon \mathbf{S_n}$ are equiprobable.

An infinite sequence $s = \{s_i\}_{i=1\ldots\infty}$ is said to be generated by an LFSR of length k if there exist constants $a_0, \ldots, a_{k-1}$ such that

$$s_{i+k} = a_{k-1}s_{i+k-1} + \ldots + a_1 s_{i+1} + a_0 s_i \qquad for\ i\ \geq\ 1$$

For $s \epsilon \mathbf{S_n}$ we define the *linear complexity*, $\Lambda(s)$, to be the length of the smallest LFSR that generates a sequence whose first n terms are $s_1, s_2, \ldots, s_n$. If $k \leq n$, we will use $\Lambda_k(s)$ to denote the linear complexity of the sequence, $s_1, s_2, \ldots, s_k$.

If $s, t \epsilon \mathbf{S_n}$, we define $\Delta(s,t) = |\Lambda(s) - \Lambda(t)|$ and for $1 \leq k \leq n, \Delta_k(s,t) = |\Lambda_k(s) - \Lambda_k(t)|$.

3 Changing a Bit and Other Bijections

The drastic increase in linear complexity caused by changing the last bit of a sequence of n zeros brings up the question of the general effect of a single bit change on the linear complexity of a finite sequence.

Fix an integer $k, 1 \leq\ k\ \leq n$ and for each $s \epsilon \mathbf{S_n}$, let $\tilde{s}^k \epsilon \mathbf{S_n}$ be the sequence obtained from s by changing the k^{th} bit, i.e. $\tilde{s}^k_k = 1 - s_k$ and $\tilde{s}^k_i = s_i$, for $i = 1 \ldots n,\ i \neq k$. Define

$$\overset{k}{\Delta}(n) = \frac{1}{2^{n-1}} \sum_{s \epsilon \mathbf{S_n}, s_k = 0} |\ \Delta(s, \tilde{s}^k)\ | \ . \tag{1}$$

This is the average change in linear complexity caused by a change in the k^{th} bit. Although a change in the n^{th} bit can cause a severe change in linear complexity, theorem 3.1 states that on the average, the change is close to one. We first present two lemmas that will be used in in the proof of this theorem and in later sections.

Lemma 3.1 *Let $s \epsilon \mathbf{S_n}$ and let $\tilde{s}^n$ be defined as above, then $\Delta(s, \tilde{s}^n) =| \Lambda(s) - \Lambda(\tilde{s}^n) |$ is given by*

$$\Delta(s, \tilde{s}^n)\ =\ \begin{cases} 0 & if \Lambda_{n-1}(s) = \Lambda_{n-1}(t) \geq \frac{n}{2} \\ n - 2k & if \Lambda_{n-1}(s) = \Lambda_{n-1}(t) = k < \frac{n}{2} \end{cases} \ .$$

Proof: This result is due to Massey, [2].

Lemma 3.2 *The distribution of $\Lambda(s), s \epsilon \mathbf{S_m}$ is given by*

$$card\{s : \Lambda(s) = k\} = \begin{cases} 1 & k = 0 \\ 2 \cdot 4^{k-1} & k \leq m/2 \\ 4^{m-k} & k > m/2 \end{cases} \ .$$

Proof: This result follows, by induction, from lemma 3.1. It appears in a slightly different form in Rueppel, [3, page 36].

Theorem 3.1 *The average change in the linear complexity of a n-bit string caused by a change in the last bit is given by*

$$\overset{n}{\Delta}(n) = \begin{cases} \frac{8}{9} + \frac{3n-4}{9\cdot 2^{n-1}} & \longrightarrow \frac{8}{9} \quad n \text{ even} \\ \\ \frac{10}{9} + \frac{3n-4}{9\cdot 2^{n-1}} & \longrightarrow \frac{10}{9} \quad n \text{ odd} \end{cases} .$$

Proof: From definition (1), we have

$$\overset{n}{\Delta}(n) = \frac{1}{2^{n-1}} \sum_{s\epsilon \mathbf{S_n}, s_n=0} \mid \Delta(s, \tilde{s}^n) \mid .$$

Lemma 3.1 implies that $\Delta(s, \tilde{s}^n)$ depends only on $\Lambda_{n-1}(s)$ so

$$\overset{n}{\Delta}(n) \;=\; \frac{1}{2^{n-1}} \sum_{0\leq k<\frac{n}{2}} (n-2k) card\{s\epsilon \mathbf{S_n}, s_n = 0 \mid \Lambda_{n-1}(s) = k\}$$

and applying lemma 3.2 with $m = n-1$, yields

$$\overset{n}{\Delta}(n) \;=\; \frac{1}{2^{n-1}} \left[n\cdot 1 + \sum_{k=1}^{M} (n-2k)(2\cdot 4^{k-1}) \right]$$

where $M = \frac{n}{2} - 1$ when n is even and $M = \frac{n-1}{2}$ when n is odd. Observing that

$$\sum_{k=1}^{M} 4^{k-1}k = \frac{(3M-1)4^M + 1}{9} \tag{2}$$

we have,

$$\overset{n}{\Delta}(n) \;=\; \frac{1}{2^{n-1}} \left[n \;+\; 2n\left(\frac{4^M-1}{3}\right) - 4\left(\frac{(3M-1)4^M+1}{9}\right) \right].$$

Finally, substituting the appropriate values for M gives the desired results.

□

Having found the average change in linear complexity caused by changing the last bit of a sequence of n zeros, we now look, more generally, at the average change when the k^{th} bit is changed, $1 \leq k \leq n$. Changing the k^{th} bit (or m bits in fixed positions) induces a bijection on $\mathbf{S_n}$. Given a bijection, $\varphi : \mathbf{S_n} \longrightarrow \mathbf{S_n}$, we denote by Δ_φ, the average value of $\mid \Lambda(s) - \Lambda(\varphi(s)) \mid$. We then obtain an upper bound on Δ_φ which serves, also, as an upper bound on the average change in linear complexity caused by flipping the k^{th} bit.

Theorem 3.2 *Let $\varphi : \mathbf{S_n} \longrightarrow \mathbf{S_n}$ be a bijection. Then the average value, Δ_φ, of $\mid \Lambda(s) - \Lambda(\varphi(s)) \mid$ is bounded above by*

$$\begin{cases} \frac{4}{3} - \frac{1}{3(2^{n-2})} & n \text{ even} \\ \\ \frac{5}{3} - \frac{1}{3(2^{n-2})} & n \text{ odd.} \end{cases}$$

For each n, there exists a bijection that attains this bound.

Proof: The average of the absolute value of the differences in linear complexity between s and $\varphi(s)$ is given by

$$\Delta_\varphi = \frac{1}{2^n} \sum_{s \epsilon \mathbf{S_n}} \mid \Lambda(s) - \Lambda(\varphi(s)) \mid$$

$$\leq \frac{1}{2^n} \sum_{s \epsilon \mathbf{S_n}} \left(\mid \Lambda(s) - \frac{n}{2} \mid + \mid \Lambda(\varphi(s)) - \frac{n}{2} \mid \right).$$

$$= \frac{1}{2^{n-1}} \sum_{s \epsilon \mathbf{S_n}} \mid \Lambda(s) - \frac{n}{2} \mid$$

since φ is a bijection. So we evaluate the sum

$$\mathbf{S} \equiv \sum_{s \epsilon \mathbf{S_n}} \mid \Lambda(s) - \frac{n}{2} \mid .$$

From the distribution of linear complexity, (lemma 3.2), we see that if n is even,

$$\mathbf{S} = \frac{n}{2} + \sum_{k=1}^{n/2} 2 \cdot 4^{k-1} (\frac{n}{2} - k) + \sum_{k=1+n/2}^{n} 4^{n-k} (k - \frac{n}{2})$$

and if n is odd,

$$\mathbf{S} = \frac{n}{2} + \sum_{k=1}^{(n-1)/2} 2 \cdot 4^{k-1} (\frac{n}{2} - k) + \sum_{k=(n+1)/2}^{n} 4^{n-k} (k - \frac{n}{2}).$$

Let us first consider n even. Replacing $n-k+1$ with k in the last sum and combining terms yields

$$\mathbf{S} = \frac{n}{2} + \sum_{k=1}^{n/2} 4^{k-1} \left((1 + \frac{3n}{2}) - 3k \right) = \frac{2}{3}(2^n - 1).$$

and finally,

$$\Delta_\varphi \leq \frac{\mathbf{S}}{2^{n-1}} = \frac{4}{3} - \frac{1}{3 \cdot 2^{n-2}}.$$

A similar calculation yields the result for n odd.

To construct a bijection, φ, that obtains the upper bound, start with the sequences of highest and lowest linear complexity, working inward, and always choosing for an image the sequence most distant in linear complexity and not yet used. By the above proof φ attains the maximum value for Δ_φ.

□

4 Other Bit Change Functions

A bijection on $\mathbf{S_n}$ will take some strings to images of lower linear complexity, but others will have images with higher linear complexity. An algorithm which, given a string, $s \epsilon \mathbf{S_n}$, tries to produce a sequence, of "low" linear complexity, that differs from s in only a small percentage of its bits, should not be a bijection. Ideally, bits will only be altered when the change results in a string of lower linear complexity. In general, such an algorithm will induce a function $\varphi : \mathbf{S_n} \rightarrow \mathbf{S_n}$ but φ will not be a bijection. Here, we first consider such functions φ with the restriction that φ transforms only a bounded number of strings to the same image string, e.g. functions that alter no more than k fixed bits. This leads to upper bounds similar to those in the previous section. We then consider functions on strings subject to the condition that the linear complexity of all the image strings be "low". Our final result shows that, under this condition, there must be strings that have "many" bits changed by the function.

Theorem 4.1 *Let* $0 \leq k \leq n$ *and let* φ *be a function,* $\varphi : \mathbf{S_n} \rightarrow \mathbf{S_n}$ *such that* $card\{\varphi^{-1}(s)\} \leq 2^k$ *for all* $s \epsilon \mathbf{S_n}$. *Then an upper bound for* Δ_φ *is given by*

$$\Delta_\varphi \leq \frac{k}{2} - \frac{1 + 2^k}{3 \cdot 2^{n-1}} + \begin{cases} 4/3 & n \text{ even} \quad k \text{ even} \\ 3/2 & n \text{ odd} \quad k \text{ odd} \\ 3/2 & n \text{ even} \quad k \text{ odd} \\ 5/3 & n \text{ odd} \quad k \text{ even} \end{cases} .$$

For each n *and* k, *there exists a function, satisfying the conditions above such that* Δ_φ *attains this upper bound.*

Proof:

As in the analysis of bijections, we have

$$\Delta_\varphi \equiv \frac{1}{2^n} \sum_{s \epsilon \mathbf{S_n}} \mid \Lambda(s) - \Lambda(\varphi(s)) \mid$$

$$\leq \frac{1}{2^n} \sum_{s \epsilon \mathbf{S_n}} \left(\mid \Lambda(s) - \frac{n}{2} \mid + \mid \Lambda(\varphi(s)) - \frac{n}{2} \mid \right)$$

$$= \frac{1}{2^n} \sum_{s \epsilon \mathbf{S_n}} \mid \Lambda(s) - \frac{n}{2} \mid + \frac{1}{2^n} \sum_{s \epsilon \mathbf{S_n}} \mid \Lambda(\varphi(s)) - \frac{n}{2} \mid .$$

The analysis for bijections gives an upper bound on the first of these sums so we have,

$$\Delta_\varphi \leq \frac{1}{2^n} \sum_{s \epsilon \mathbf{S_n}} \mid \Lambda(\varphi(s)) - \frac{n}{2} \mid - \frac{1}{3 \cdot 2^{n-1}} + \begin{cases} 2/3 & n \text{ even} \\ 5/6 & n \text{ odd} \end{cases} . \qquad (3)$$

Now we must find an upper bound for

$$\mathbf{S} \;=\; \sum_{s\epsilon\mathbf{S_n}} \mid \Lambda(\varphi(s)) - \frac{n}{2} \mid$$

$$=\; \sum_{j=0}^{n} \mid \frac{n}{2} - j \mid card\{\varphi(s) \mid \Lambda(\varphi(s) = j\}.$$

This will be maximal when the image values, $\Lambda(\varphi(s))$ are as far as possible from $\frac{n}{2}$. Since each s can have up to 2^k pre-images, we start with the elements of $\mathbf{S_n}$ whose linear complexity is farthest from $\frac{n}{2}$, assuming each has 2^k pre-images until we have enough pre-images to cover the 2^n elements of $\mathbf{S_n}$, so by lemma (3.2) we obtain:

<u>$n-k$ **odd:**</u>

$$\mathbf{S} \;\leq\; 2^k \left[\frac{n}{2} \;+\; \frac{n}{2} \;+\; \sum_{j=1}^{M}(\frac{n}{2} \;-\; j)(2\cdot 4^{j-1} \;+\; 4^j)\right]$$

with $M = \frac{n-k-1}{2}$ as for this value of M,

$$2^k \left[1 \;+\; 1 \;+\; \sum_{j=1}^{M}(2\cdot 4^{j-1} \;+\; 4^j)\right] \;=\; 2^n$$

so that we have used all 2^n sequences in $\mathbf{S_n}$ as pre-images. Regrouping terms, we now obtain

$$\mathbf{S} \;\leq\; 2^k \left[n \;+\; \sum_{j=1}^{M} \frac{3}{2} 4^j (\frac{n}{2} \;-\; j)\right]$$

$$=\; 2^k \left[n \;+\; \frac{3n}{4}\left(\frac{4^{M+1} \;-\; 4}{3}\right) \;-\; 2\left(\frac{(3M-1)4^M + 1}{3}\right)\right].$$

By substituting $M \;=\; \frac{n-k-1}{2}$ and reducing we obtain

$$\mathbf{S} \;\leq\; 2^k \left[n \;+\; n(2^{n-k-1} \;-\; 1) \;-\; \frac{2}{3}\left(\left(\frac{3(n-k-1)}{2} - 1\right) 2^{n-k-1} \;+\; 1\right)\right]$$

$$=\; 2^k \left[2^{n-k-1}(k \;+\; \frac{5}{3}) \;-\; \frac{2}{3}\right]$$

and dividing by 2^n yields

$$\frac{\mathbf{S}}{2^n} \;\leq\; \frac{1}{2}(k \;+\; \frac{5}{3}) \;-\; \frac{2^k}{3\cdot 2^{n-1}} = \frac{k}{2} \;+\; \frac{5}{6} \;-\; \frac{2^k}{3\cdot 2^{n-1}}.$$

$\underline{n-k \textbf{ even:}}$

$$\mathbf{S} \;\leq\; 2^k \left[\frac{n}{2} + \frac{n}{2} \;+\; \left(\sum_{j=1}^{M-1} (\frac{n}{2} - j)(2 \cdot 4^{j-1} + 4^j) \right) \;+\; (\frac{n}{2} - M) \left(2 \cdot 4^{M-1} \right) \right]$$

with $M = \frac{n-k}{2}$ as for this value of M,

$$2^k \left[1 \;+\; 1 \;+\; \left(\sum_{j=1}^{\frac{n-k}{2}-1} (2 \cdot 4^{j-1} \;+\; 4^j) \right) \;+\; 2 \cdot 4^{\frac{n-k}{2}-1} \right] \;=\; 2^n.$$

Regrouping terms, we now obtain

$$\mathbf{S} \;\leq\; 2^k \left[n \;+\; \left(\sum_{j=1}^{M-1} \frac{3}{2} 4^j (\frac{n}{2} \;-\; j) \right) \;+\; 2^{2M-1} (\frac{n}{2} \;-\; M) \right]$$

$$= \; 2^k \left[n \;+\; 2^{2M-1} (\frac{n}{2} - M) \;+\; \frac{3n}{4} \left(\frac{4^M - 4}{3} \right) \;-\; 2 \left(\frac{(3M-4)4^{M-1} + 1}{3} \right) \right].$$

By substituting $M \;=\; \frac{n-k}{2}$ and reducing we obtain

$$\mathbf{S} \;\leq\; 2^k \left[2^{n-k-2} (2k \;+\; \frac{8}{3}) \;-\; \frac{2}{3} \right]$$

and dividing by 2^n yields

$$\frac{\mathbf{S}}{2^n} \;\leq\; \frac{k}{2} \;+\; \frac{2}{3} \;-\; \frac{2^k}{3 \cdot 2^{n-1}}.$$

Finally, substituting these upper bounds into the inequality (3) gives the desired results.

As in the case of a bijection, we can construct a function, $\varphi : \mathbf{S_n} \rightarrow \mathbf{S_n}$ that satisfies the conditions of the theorem and attains the upper bound by starting with those elements of $\mathbf{S_n}$ which have highest and lowest linear complexity, working inward and always choosing for an image the element of $\mathbf{S_n}$ which is still available and maximizes the difference in linear complexity. The only difference is that each element of $\mathbf{S_n}$ can be used as an image 2^k times. The following table shows how to construct this function when $n = 6$ and $k = 2$. Each string can have 4 pre-images. The table says that the unique string of linear complexity 6 and ant three of those with linear complexity 5 should be mapped to the string with linear complexity zero. The average change, $\Delta_\varphi = 2.28125$, as the theorem states.

Λ	$card\{s \mid \Lambda(s) = \Lambda\}$	Λ_*	$card\{s' \mid \Lambda(\varphi(s')) = \Lambda$ and $\Lambda(s') = \Lambda_*\}$
0	1	6 5	1 3
1	2	5 4	1 7
2	8	4 3	9 23
3	32		
4	16		
5	4	3 2	9 7
6	1	2 1 0	1 2 1

□

Theorem 4.2 *Given $c(n), 0 < c(n) < \frac{n}{2}$, there exists a largest $\mu, 0 < \mu < 1$, such that there are at least $\frac{n-1}{n}2^n$ sequences for which at least a fraction of μ bits must be changed to get a sequence with linear complexity less than $c(n)$. The fraction μ is a function of n and $c(n)$. If $\lim_{n\to\infty} \frac{c(n)}{n} = 0$ (e.g. if $c \leq p(\log_2(n))$ for a polynomial, p) then μ converges to $\frac{1}{2}$ as $n \to \infty$.*

Proof:

Let $M(\mu, n)$ be the number of sequences in $\mathbf{S_n}$ within Hamming distance μn of a fixed string, $s\epsilon\mathbf{S_n}$. Then, for $0 < \mu < 1/2$,

$$M(\mu, n) = \sum_{k=0}^{\lfloor \mu n \rfloor} \left(\begin{array}{c} n \\ k \end{array} \right) \ \leq \ 2^{nH_2(\mu)} \tag{4}$$

where $H_2(x) = -x\log_2(x) - (1-x)\log_2(1-x)$ is the *entropy* function, (MacWilliams and Sloane, [1, page 310]).

The number of sequences in $\mathbf{S_n}$ with linear complexity less than $c(n)$ is given by:

$$1 \ + \ \sum_{k=1}^{c(n)} 2 \cdot 4^{k-1} \ = \ 1 \ + \ 2\left(\frac{2^{2c(n)} - 1}{3}\right) \ = \ \frac{1 \ + \ 2^{2c(n)+1}}{3} \ \leq \ 2^{2c(n)+1}.$$

In order to change $\frac{2^n}{n}$ of the sequences in $\mathbf{S_n}$ to sequences of linear complexity less than $c(n)$ by altering at most μn bits of each sequence, we must have:

$$2^{2c(n)+1} M(\mu, n) \ \geq \ \frac{2^n}{n}.$$

Hence $M(\mu, n)$ must satisfy

$$M(\mu, n) \ \geq \ \frac{2^{n-\log_2(n)}}{2^{2c(n)+1}} \ = \ 2^{n \ - \ \log_2(n) \ - \ 2c(n) \ - \ 1}.$$

Substituting the upper bound from (4), we have

$$2^{nH_2(\mu)} \ \geq \ M(\mu, n) \ \geq \ 2^{n \ - \ \log_2(n) \ - \ 2c(n) \ - \ 1}.$$

Comparing the exponents, we see that

$$H_2(\mu) \ \geq \ 1 - \frac{\log_2(n) + 2c(n) + 1}{n}.$$

So $H_2(\mu) \to 1$ and hence, $\mu \to \frac{1}{2}$ as n tends to infinity, (MacWilliams and Sloane [1, page 308]).

□

5 Conclusion and Future Work

The results of this paper tell us two things, of cryptographic importance, about the difference in linear complexity of strings and their neighbors vis-à-vis Hamming distance.

1: For large n there are strings in $\mathbf{S_n}$ which are cryptographically secure in the sense that they are far, in Hamming distance, from any string of "low" linear complexity.

2: There are enough such secure strings that we cannot expect to find an algorithm which for "most" strings produces nearby strings (in Hamming distance) of "low" linear complexity.

This suggests the following paths for future investigation:

1. Classify those sequences which are close (in Hamming distance) to sequences of "low" linear complexity. The results of this paper put bounds on how many such sequences there are but do not indicate what they look like.

2. Study the effect of synchronization errors on the linear complexity of sequences. We say that two sequences are $k - close$ if one can be obtained from the others by a sequence of no more then k errors where we now include added and lost bits as well as changes of bits. A sequence is k-close to many more sequences than it is within k of in Hamming distance. Theorem 6.2 does not immediately generalize in a useful way. As with Hamming distance, we should classify those sequences which are k-close to sequences of "low" linear complexity.

3. Answer these questions for "practical" sequences, e.g. those that can be generated by nonlinear registers of acceptable size.

References

[1] F. J. MacWilliams and N. J. A. Sloane. *The Theory of Error-Correcting Codes.* North-Holland Mathematical Library, Amsterdam, 1977.

[2] J. M. Massey. *Shift-Register Synthesis and BCH Decoding.* IEEE Trans. Information Theory 15, 122-127 (1969).

[3] R. A. Rueppel. *Analysis and Design of Stream Ciphers.* Springer, Berlin,1986.