Stream Ciphers (contd.)

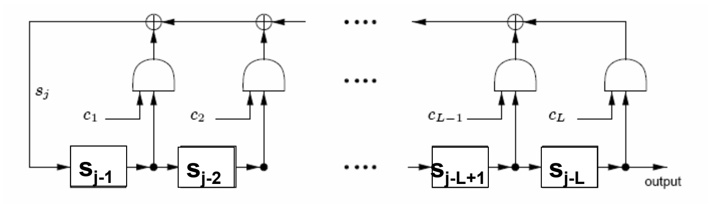
Debdeep Mukhopadhyay

Assistant Professor
Department of Computer Science and Engineering
Indian Institute of Technology Kharagpur
INDIA -721302

Objectives

- **Linear Complexity**
- **Berlekamp Massey Algorithm**

The LFSR Structure



$$s_j = \sum_{i=1}^{L} c_i s_{j-i},\ j = L, L+1, \ldots$$

An LFSR is said to generate a finite sequence $s_0, s_1, \ldots, s_{N-1}$ when this sequence coincides with the first N output digits of the LFSR for some initial loading.

Generation of a sequence

- **If L≥N, the LFSR always generates the sequence.**
- **If L<N, it follows that the LFSR generates the sequence if and only if:**

$$s_j = \sum_{i=1}^{L} c_i s_{j-i},\ j = L, L+1, \ldots, N-1$$

Theorem 1

If some LFSR of length L generates the sequence $s_0, s_1, ..., s_{N-1}$ but not the sequence $s_0, s_1, ..., s_{N-1}, s_N$ then any LFSR that generates the latter sequence has length L', satisfying:

$$L' \geq N + 1 - L$$

Proof

Case 1: L ≥ N, the theorem is trivially true.

Case 2: L<N, let $c_1, c_2, ..., c_L$ and $c'_1, c'_2, ..., c'_{L'}$ denote the connection coefficients of the two LFSRs in question and assume that L' ≤ N-L.

$$\therefore \sum_{i=1}^{L} c_i s_{j-i} = s_j, \; j = L, L+1, ..., N-1$$

$$\neq s_N, \; j = N$$

$$\therefore \sum_{i=1}^{L'} c'_k \, s_{j-k} = s_j, \; j = L', L'+1, ..., N-1, N$$

Proof (contd.)

Consider, $\sum_{i=1}^{L} c_i s_{N-i}$

Note that $\{s_{N-L}, s_{N-L+1}, ..., s_{N-1}\}$ is a subset of $\{s_{L'}, s_{L'+1}, ..., s_{N-1}\}$.

$$\therefore \sum_{i=1}^{L} c_i s_{N-i} = \sum_{i=1}^{L} c_i \sum_{k=1}^{L'} c'_k\, s_{N-i-k}$$

$$= \sum_{k=1}^{L'} c'_k \sum_{i=1}^{L} c_i s_{N-i-k}$$

$$= \sum_{k=1}^{L'} c'_k\, s_{N-k} = s_N$$

Note that $\{s_{N-L'}, s_{N-L'+1}, ..., s_{N-1}\}$ is a subset of $\{s_L, s_{L+1}, ..., s_{N-1}\}$.

Thus we have a contradiction. This proves the result.

Linear Complexity

- **Define $L_N(s)$ as the minimum length of all LFSRs that generate $s_0, s_1, \ldots, s_{N-1}$**
- **Clearly, $L_N(s) \leq N$**
- **Moreover, $L_N(s)$ must be monotonically decreasing with increasing N.**
- **Convention:**
  - **all 0 sequence is generated by the LFSR with L=0**
  - **When $s_0, s_1, \ldots, s_{N-1}$ are all 0's but s_N=1, then L=N+1**

Lemma 1

If some LFSR of length L generates the sequence $s_0, s_1, ..., s_{N-1}$ but not the sequence $s_0, s_1, ..., s_{N-1}, s_N$ then

$$L_{N+1}(s) \geq \max[L_N(s), N+1-L_N(s)]$$

From the monotonicity of $L_{N+1}(s) \geq L_N(s)$.

From Theorem 1, $L_{N+1}(s) \geq N+1-L_N(s)$.

Thus the lemma 1 follows.

Berlekamp Massey's Algorithm

- **A recursive algorithm for producing one of the LFSRs of length $L_N(s)$, which generates $s_0, s_1, \ldots, s_{N-1}$ for N=1, 2, 3, …**
- **$C(D)=1+C_1D+\ldots+C_LD^L$ which has degree at most L in the indeterminate.**
- **Convention: C(D)=1 for the LFSR of length L=0**

Connection Polynomial

For a given s, let

$$C^N(D) = 1 + C_1^{(N)}(D) + \ldots + C_{L_N(S)}^{(N)}(D)^{L_N(s)}$$

denote the connection polynomial of a minimal length $L_N(s)$ LFSR that generates $s_0, s_1, \ldots, s_{N-1}$

Discrepancy

Lemma 1 is actually an equality. We have seen this for the base case.

Assume an induction hypothesis for $L_N(s)$.

The corresponding polynomial is $C^N(D)$.

$$\therefore s_j \oplus \sum_{i=1}^{L_n(s)} c_i^{(n)} s_{j-i} = \begin{cases} 0, \ j = L_{n(s)}, \ldots, n-1 \\ d_n, \ j = n \end{cases}$$

d_n : next discrepancy (between s_n and the (n+1)st bit generated by the minimal length LFSR, which we have found to generate the first n bits of s.

Correcting the discrepancy

$Case 1: d_n = 0$

LFSR also generates the first n+1 bits of s. Thus,

$\mathrm{L}_{n+1}(s) = L_n(s), C^{(n+1)}(D) = C^n(D)$

$Case 1: d_n = 1$

Let m be the sequence length before the last length change in the minimal length register,

i,e

$$\mathrm{L}_m(s) < L_n(s)$$
$$\mathrm{L}_{m+1}(s) = L_n(s)$$

Proving the Induction Hypothesis

Since a length change was required $<\mathrm{L}_m(s), c^m(D)>$ could not generate $\mathrm{s}_0, s_1, \ldots, s_m$

$$\therefore s_j \oplus \sum_{i=1}^{L_n(s)} c_i^{(n)} s_{j-i} = \begin{cases} 0, \ j = L_{m(s)}, \ldots, m-1 \\ d_m, \ j = m \end{cases}$$

By induction hypothesis,

$$\mathrm{L}_{m+1}(s) = \mathrm{L}_n(s) = \max[\mathrm{L}_m(s), m+1-\mathrm{L}_m(s)]$$

$\because \mathrm{L}_m(s) < L_n(s), \mathrm{L}_n(s) = m+1-\mathrm{L}_m(s)$

Recursive construction of polynomial

Claim:

$C(D) = C^n(\mathrm{D}) \oplus \mathrm{D}^{n-m}\mathrm{C}^m(\mathrm{D})$ is a valid next choice for $\mathrm{C}^{n+1}(D)$.

Note: degree of C(D)=max[$\mathrm{L}_n(s), n-m+L_m(s)$]

=max[$\mathrm{L}_n(s), n+1-L_n(s)$]

$\therefore C(D)$ is an allowable connection polynomial

for an LFSR of length L=max[$\mathrm{L}_n(s), n+1-L_n(s)$]

Proof that C(D) generates s^{n+1}

$$\therefore s_j \oplus \sum_{i=1}^{L} c_i s_{j-i} = s_j \oplus \sum_{i=1}^{L_n(s)} c_i^{(n)} s_{j-i} \oplus$$

$$[s_{j-n+m} \oplus \sum_{i=1}^{L_m(s)} c_i^{(m)} s_{j-n+m-i}]$$

$$= \begin{cases} 0, \ j = L, L+1, \ldots, n-1 \\ 1 \oplus 1 = 0, \ j = n \end{cases}$$

Conclusions

- **The LFSR with length L and connection polynomial C(D) generates $s_0,s_1,\ldots,s_n$**
- **Since L satisfies Lemma 1 with equality, the induction is also proved.**

The final Algorithm

Algorithm Berlekamp-Massey algorithm

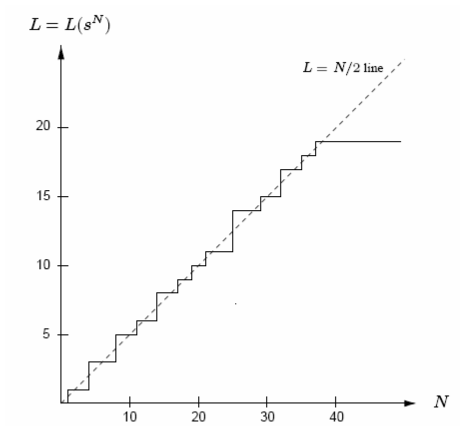
INPUT: a binary sequence $s^n = s_0, s_1, s_2, \ldots, s_{n-1}$ of length n.
OUTPUT: the linear complexity $L(s^n)$ of s^n, $0 \le L(s^n) \le n$.

1. *Initialization.* $C(D) \leftarrow 1,\ L \leftarrow 0,\ m \leftarrow -1,\ B(D) \leftarrow 1,\ N \leftarrow 0.$
2. While $(N < n)$ do the following:
   - 2.1 *Compute the next discrepancy d.* $d \leftarrow (s_N + \sum_{i=1}^{L} c_i s_{N-i}) \bmod 2.$
   - 2.2 If $d = 1$ then do the following:
     $T(D) \leftarrow C(D),\ C(D) \leftarrow C(D) + B(D) \cdot D^{N-m}.$
     If $L \le N/2$ then $L \leftarrow N + 1 - L,\ m \leftarrow N,\ B(D) \leftarrow T(D).$
   - 2.3 $N \leftarrow N + 1.$
3. Return(L).

Example

- **Consider the sequence of periodicity 20:**
  10010011110001001110
- **We plot the variation of the linear complexity with N.**
  - **this is obtained by the Berlekamp Massey Algorithm**
  - **this is called Linear Profile**

Example



Exercise

- **Reconstruct an LFSR (of the shortest length) which generates the sequence 00111011.**

s_n	d	T(D)	C(D)	L	m	B(D)	N
-	-	-	1	0	-1	1	0
0	0	-	1	0	-1	1	1
0	0	-	1	0	-1	1	2
1	1	1	$1+D^3$	3	2	1	3
1	1	$1+D^3$	$1+D+D^3$	3	2	1	4
1	0	$1+D^3$	$1+D+D^3$	3	2	1	5
0	0	$1+D^3$	$1+D+D^3$	3	2	1	6
1	0	$1+D^3$	$1+D+D^3$	3	2	1	7
1	1	$1+D+D^3$	$1+D+D^3 + D^5$	5	7	$1+D+D^3$	8

Further Reading

- James Massey, “Shift-Register Synthesis and BCH Decoding”, IEEE Transactions on Information Theory, 1969
- D. Stinson, Cryptography: Theory and Practice, Chapman & Hall/CRC
- A. Menezes, P. Van Oorschot, Scott Vanstone, “Handbook of Applied Cryptography” (Available online)

Next Days Topic

- Stream Ciphers (contd.)