Berlekamp–Massey Algorithm

input: $s^N = s_0 s_1 \ldots s_{N-1}$ binary seq. of length N

output: lin. complexity profile for s^N:

$L_0, L_1, \ldots, L_N$

$L_i = L(s_0 s_1 \ldots s_{i-1})$, where $s_0 s_1 \ldots s_{i-1} = s^i$

$f_0, f_1, \ldots, f_N$

f_i generating polynomial for s^i of degree L_i
(smallest degree gen. pol. for s^i).

---

s^N is generated by $f_N(x) = x^{L_N} + c_1 x^{L_N - 1} + \ldots + c_{L_N}$

$s^{N+1} = s^N s_N = s_0 s_1 \ldots s_{N-1} s_N$

is generated by $f_N(x) \Longleftrightarrow$

$$d_N = s_N + c_1 \cdot s_{N-1} + \ldots + c_{L_N} \cdot s_{N-L_N} = 0$$

(↑ called N-th discrepancy)

why? $d_N = 0 \Longleftrightarrow s_N = c_1 \cdot s_{N-1} + \ldots + c_{L_N} \cdot s_{N-L_N}$

$\Longleftrightarrow f_N(x)$ gen. s^{N+1}

Formulate the Algorithm as a Theorem.

Theorem (B-M) $s^{N+1} = s_0 \ldots s_{N-1} s_N$. Then

1. $s^{N+1} = 0^{N+1}$ (all zero sequence of length $N+1$)

   by agreement $L_{N+1} = 0$, $f_{N+1} = 1$.

$S^{N+1} = \underline{0^N 1}$

By a Lemma $L_{N+1} = N+1$, f_{N+1} is any polynom. of degree $N+1$.

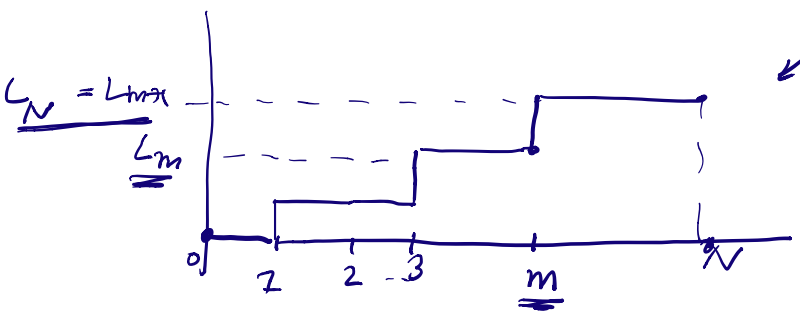
2. $d_N = 0$ (computed with f_N)
then $L_{N+1} = L_N$ and $f_{N+1} = f_N$.

3. $d_N = 1$, then

$$L_{N+1} = \begin{cases} L_N, & L_N > N/2 \\ N+1-L_N, & L_N \le N/2 \end{cases}$$

$$f_{N+1} = \begin{cases} f_N + x^{2L_N - N - 1} \cdot f_m, & L_N > N/2 \\ x^{N+1-2L_N} \cdot f_N + f_m, & L_N \le N/2 \end{cases}$$

m largest integer s.t. $L_m < L_N$.



↙ graph for lin. complexity profile.

Equivalent form

$$d_N = 1$$

$$L_{N+1} = \begin{cases} L_N, & N - L_N \le m - L_m \\ N+1-L_N, & N - L_N > m - L_m \end{cases}$$

$$f_{N+1} = \begin{cases} f_N + x^{(m-L_m)-(N-L_N)} \cdot f_m, & N - L_N \le m - L_m \\ x^{(N-L_N)-(m-L_m)} \cdot f_N + f_m, & N - L_N > m - L_m \end{cases}$$

↗ more conv.

Idea of the proof.

$d_N = 1$ and s^N is generated by $f_N = x^{L_N} + c_1 \cdot x^{L_N - 1} + \dots + c_{L_N}$

$\Rightarrow$

$$(*)\quad \underbrace{s_{N-i} + c_1 \cdot s_{N-i-1} + \dots + c_{L_N} \cdot s_{N-i-L_N}}_{} = \begin{cases} d_N = 1 & i = 0 \\ 0 & 0 < i \le N - L_N \end{cases}$$

↗ f_N gen. pol. for s^N.

$d_m = 1$ as $L_m < L_{m+1} = L_N$

s^m is generated by $f_m = x^{L_m} + b_1 \cdot x^{L_m - 1} + \dots + b_{L_m}$

$$(**)\quad \underbrace{s_{m-i} + b_1 \cdot s_{m-i-1} + \dots + b_{L_m} \cdot s_{m-i-L_m}}_{} = \begin{cases} d_m = 1 & i = 0 \\ 0 & 0 < i < m - L_m \end{cases}$$

↗ f_m gen. pol. for s^m

sum up (*) and (**) and get

$$s_{N-i} + \dots + c_{L_N} s_{N-i-L_N} + s_{m-i} + \dots + b_{L_m} s_{m-i-L_m} = 0$$

$$0 \le i \le \min(N-L_N, m-L_m)$$

recurrence incorporate s_N for $i=0$

1) $N - L_N \le m - L_m \iff L_N > N/2$

2) $N - L_N > m - L_m \iff L_N \le N/2$

$\Rightarrow$ two poss. recurrences $\Rightarrow$ two poss. polynomials.

Why $N - L_N \le m - L_m \iff L_N > N/2$?

$L_m < L_{m+1} = L_N$ by definition of m.

$\Rightarrow L_N = L_{m+1} = m+1-L_m$ (induction)

$\Rightarrow m - L_m = L_N - 1$.

$N - L_N \le m - L_m \iff N - L_N \le L_N - 1$

$\iff N+1 \le 2 \cdot L_N \iff N < 2 \cdot L_N \iff L_N > N/2$.

Back to the example.

$s^{10} = 1001001111$

N	s^N	P_N	L_N
0	$\emptyset$	1	0
3	100	x	1

4	1001	$x^3+1, \dots$	3
⋮			
7	1001001	x^3+1	3
8	10010011	x^5+x^2+x	5
9	100100111	$x^5+x^4+x^2$	5
10	1001001111	$x^5+x^4+x^3+x^2+1$	5

$N=7$

$d_7 = s_7 + 0\cdot s_6 + 0\cdot s_5 + 1\cdot s_4 = 1$, $L_7 = 3 < 7/2$

(1 1 0 0)

$f_8 = x^{7+1-2\cdot 3} f_7 + f_3 = x^2(x^3+1) + x = x^5 + x^2 + x$

$N=8$

$d_8 = s_8 + 0\cdot s_7 + 0\cdot s_6 + 1\cdot s_5 + 1\cdot s_4 + 0\cdot s_3 = 1$

(1 1 1 0 0)

$L_8 = 5 > 8/2 \Rightarrow f_9 = f_8 + x^{2\cdot 5 - 8 - 1} f_7 =$

$= x^5 + x^2 + x + x(x^3+1) = x^5 + x^4 + x^2$

$d_9 = s_9 + 1\cdot s_8 + 0\cdot s_7 + 1\cdot s_6 + 0\cdot\dots = 1$

(1 1 1 1)

$L_9 = 5 > 9/2 \Rightarrow f_{10} = f_9 + x^{2\cdot 5 - 9 - 1} \cdot f_7 =$

$= x^5 + x^4 + x^2 + x^0(x^3+1) = x^5 + x^4 + x^3 + x^2 + 1$

Corollary. $S = s_0 s_1 \dots s_N \dots$ and $L(s) = L$

$s^t = s_0 s_1 \dots s_{t-1}$ and $t \geq 2L-1$

Then $L(s^t) = L$.

Proof. Assume $L(s^t) < L$ construct a contradiction.

$L(s^t) = \dots = L(s^{T-1}) < L(s^T)$

for some $T > t \geq 2L-1$

jump in lin. complexity

$$L(s^T) = (T-1)+1-L(s^{T-1}) = T - L(s^t)$$

$\Rightarrow$

$$2\cdot L > L(s^T) + L(s^t) = T \geq 2\cdot L$$

(with $L(s^T) \geq L$ and $L(s^t) > L$)

contradiction. proves the statement.

Lemma. $s^{2\cdot L-1} = s_0 s_1 \dots s_{2L-2}$

and $L(s^{2L-1}) = L$,

$(0\dots 0 1 a_1 \dots a_t)$

$$S = \begin{pmatrix} s_{L-1} & s_L & \dots & s_{2L-2} \\ & & & \\ s_t & s_{t+1} & \dots & s_{L-t-1} \\ & & & \\ s_1 & s_2 & \dots & s_L \\ s_0 & s_1 & \dots & s_{L-1} \end{pmatrix} = 0$$

square matrix of size $L \times L$

Then S is non-singular (its determinant $\neq 0$)

Proof. ~~$f(x) = x^{L} + c_1 \cdot x^{L-1} + \dots + c_L$~~ gen. polynomial

for s^{2L-1} (smallest degree gen. pol.)

Assume S is singular and construct a contradiction.

$$(*) \quad a \cdot S = 0$$

e.g. $(1 \; \boxed{L-1})$ $t \geq L$

$(0 \; 1 \; \boxed{L-2})$

~~$a = (0 \dots 0 1 a_1 \dots a_t)$~~ non-zero solution to $(*)$.

$t < L$ vector a is of length L.

$\Leftrightarrow$

$$s_t = a_1 s_{t-1} + \dots + a_t s_0 \quad \times c_L$$
$$s_{t+1} = a_1 s_t + \dots + a_t s_1 \quad \times c_{L-1}$$
$$\dots$$
$$s_{t+L-1} = a_1 s_{t+L-2} + \dots + a_t s_{L-1} \quad \times c_1$$

$$s_{t+L} = a_1 s_{t+L-1} + \dots + a_t s_L$$

$\times c_L, \dots, \times c_2, c_1$

$$s_{t+L+1} = a_1 s_{t+L} + \dots + a_t s_{L+1}$$

$$\dots$$

$$s_{2L-2} = a_1 s_{2L-3} + \dots + a_t s_{2L-t-2}$$

$\Rightarrow$ $X^t + a_1 X^{t-1} + \dots + a_t$ generating poz s^{2L-1}

$t < L$ contradiction as $f(x)$ of degree L smallest degree gen. pol.

$\Rightarrow$ S non-singular.

Corollary. $s^{2L} = s_0 s_1 \dots s_{2L-1}$

$L(s^{2L}) = L$

$f(x) = X^L + c_1 X^{L-1} + \dots + c_L$ generating polynomial for s^{2L}.

Then

1) $f(x)$ is unique

2) $(c_1 \dots c_L)$ $\begin{pmatrix} s_{L-1} & s_L & \dots & s_{2L-2} \\ & & & \\ s_1 & s_2 & \dots & s_L \\ s_0 & s_1 & \dots & s_{L-1} \end{pmatrix} = (s_L\ s_{L+1} \dots s_{2L-1})$

$(*)$

Proof. 2) obvious

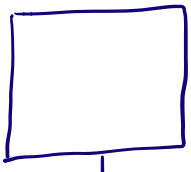
$S^{2L-1} = S_0 \dots S_{2L-2}$ is of lin. complexity L $\Rightarrow$

$\Rightarrow$ matrix in (*) is non-singular

$\Rightarrow$ solution $(c_1 \dots c_L)$ is unique

$\Rightarrow$ polynomial $f(x)$ is unique $\Rightarrow$ 1).

Applications in cryptanalysis of stream ciphers.



$\downarrow$ $2c_i$ key-stream

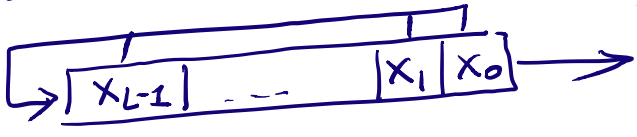
assume lin. complexity of the key-stream is $L \leq A$ some bound.

$\Rightarrow$ $x^{2L}(o2\, x^{2\cdot A})$ defines unique generating polynomial for the whole key-stream.

apply B-M algorithm to $x^{2\cdot A}$ and recover this polynomial.

$$x^L + c_1 x^{L-1} + \dots + c_L$$

construct LFSR



given only $2 \cdot A$ bits of the key-stream

we predict all bits.