T-79.5501
Cryptology

Lecture 5 (Feb 13, 2007):

- Linear complexity
- Linear cryptanalysis Sections 3.1-3.3

Linear complexity

Let $S = z_0, z_1, z_2, z_3, \ldots$ be a finite or infinite sequence. We say that the linear complexity LC(S) of S is the length of the shortest LFSR which generates it.

Linear complexity of a finite sequence does not decrease if new terms are added to the sequence, but it may remain the same.

<u>Examples 5.</u>

a) S = 000…01 (with n - 1 zeroes); LC(S) = n; one feedback polynomial of the LFSR is $1 + x^n$; indeed, any polynomial of degree n can be taken as feedback polynomial.

b) S = 111..10 (with n ones); LC(S) = n; one feedback polynomial of the LFSR is $1 + x + x^n$; indeed, any polynomial of degree n with odd number of terms can be taken as feedback polynomial.

c) By example 3, the linear complexity of 0111001011 is less than or equal to 3, since the polynomial f has degree 3. From b) above it follows that the linear complexity is exactly 3.

Linear complexity

Theorem 4. Let LC(S) = L . Consider the LFSR of length L which generates the sequence S of length n (where n can be infinite). Then

a) the L subsequent states of the the LFSR are linearly independent.

b) the L + 1 subsequent states are linearly dependent.

c) If moreover, at least $2L$ terms of the sequence are given, that is, $n \geq 2L$, then the connection polynomial of the generating LFSR is uniquely determined (see also Stinson: Section 1.2.5).

Proof. Let the connection coefficients be $c_0\, c_1\ c_2\ c_3 \ldots c_{L-1}$. Writing the recursion equation

$$z_{k+L} = c_0 z_k + c_1 z_{k+1} + c_2 z_{k+2} + \ldots + c_{L-1} z_{k+L-1}$$

in vector form we get

$$(z_L\ z_{L+1}\ z_{L+2}\ z_{L+3} \ldots z_{2L-1}) = (c_0\ c_1\ c_2\ c_3 \ldots c_{L-1})\, Z \qquad (*)$$

Linear Complexity

where the rows (and columns) of the matrix Z are vectors $(z_k\ z_{k+1}\ z_{k+2}\ z_{k+3}\ .\ .\ .\ z_{k+L-1})$, for $k = 0,1,\ldots,L-1$. Claim b) follows immediately from this representation. Further, if L subsequent states are linearly dependent, the sequence satisfies a linear recursion relation of length (at most) L -1, and can be generated using a LFSR of length less than L . This gives a).

Finally, if at least $2L$ terms of the sequence are given, then the L vectors

$$(z_k\ z_{k+1}\ z_{k+2}\ z_{k+3}\ .\ .\ .\ z_{k+L-1}),\ \ k = 0,1,\ldots,L$$

that determine the columns of the matrix Z in equation (*) are known. By a), the matrix Z is invertible. This gives a unique solution for the tap constants $(c_0\ c_1\ c_2\ c_3\ .\ .\ .c_{L-1})$. □

Linear Complexity

Now we know:

1. Any finite or periodic sequence has a finite linear complexity. Linear complexity is less than or equal to the length and the period of the sequence.
2. If we know the linear complexity of the sequence we can compute the feedback polynomial. The feedback polynomial is unique if the length of the available sequence is at least twice as much as the linear complexity.

Question:

How can we determine the linear complexity for a sequence?

Answer:

Using Berlekamp-Massey Algorithm

Linear Complexity Change Lemma

Denote: $S = z_0, z_1, z_2, z_3, \ldots$

$S^{(k)} = z_0, z_1, z_2, \ldots, z_{k-1}$

$L_k = \mathrm{LC}(S^{(k)})$

$f^{(k)}(x) =$ polynomial of degree L_k such that $S^{(k)}$ can be generated using an LFSR with feedback polynomial $f^{(k)}(x)$

Lemma. If LFSR with $f^{(k)}(x)$ does not generate $S^{(k+1)}$ then

$$L_{k+1} \geq \max\{L_k, k + 1 - L_k\}$$

Proof. $f^{(k)}(x)$ generates $S^{(k+1)} + \{00\ldots01\}$, that is, $S^{(k+1)}$ with the last bit flipped, hence $\mathrm{LC}(S^{(k+1)} + \underbrace{\{00\ldots01\}}_{k+1}) = L_k$. Then

$$k + 1 = \mathrm{LC}(00\ldots01) = \mathrm{LC}((S^{(k+1)} + 00\ldots01) + S^{(k+1)}) \leq$$

$$\mathrm{LC}(S^{(k+1)} + 00\ldots01) + \mathrm{LC}(S^{(k+1)}) = L_k + L_{k+1},$$

from where the claim follows.

□

Linear Complexity: Berlekamp-Massey

Berlekamp-Massey: If $f^{(k)}(x)$ does not generate $S^{(k+1)}$ then

$$L_{k+1} = \max\{L_k, k+1-L_k\}$$

and

$$\boxed{f^{(k+1)}(x) = x^{L_{k+1}-L_k} f^{(k)}(x) + x^{L_{k+1}-k+m-L_m} f^{(m)}(x)}$$

where m is the largest index such that $L_m < L_k$. That is, m the previous index at which the linear complexity changed.

Comments:

(1) BM algorithm may give feedback polynomials with $c_0 = 0$.

(2) Polynomial $f^{(k)}(x)$ is not unique unless degree of $f^{(k)}(x)$ is $\leq k/2$.

Berlekamp-Massey Algorithm

k = number of terms observed

z_{k-1} = k^{th} term observed

1. Intialize $k = 0$, $L_k = 0$, $f^{(k)}(x) = 1$. If all $z_k = 0$, output $L = 0$, $f(x) = 1$.
2. Else, set r to be the least index such that $z_{r-1} = 1$. Then set $m = r - 1$, $L_m = 0$, $f^{(m)}(x) = 1$, and set $L_r = r$, $f^{(r)}(x) = 1 + x^r$.
3. Set $k = r$.
4. Check if $f^{(k)}(x)$ generates z_k from the preceeding terms of the sequence. If yes, set $f^{(k+1)}(x) = f^{(k)}(x)$ and $L_{k+1} = L_k$.
5. Else use Berlekamp-Massey theorem to compute L_{k+1} and $f^{(k+1)}(x)$. If $L_{k+1} > L_k$ set $m = k$, $L_m = L_k$ and $f^{(m)}(x) = f^{(k)}(x)$.
6. If z_k the last term, output $f(x) = f^{(k+1)}(x)$ and $L = L_{k+1}$.
7. Else set $k = k+1$, and go to 4.

Berlekamp-Massey: Example

k	z_{k-1}	L_k	$f^{(k)}(x)$	m
0		0	1	
1	1	$r=1$	$1+x^r = 1+x$	0
2	1	1	$1+x$	0
3	0	2	$x(1+x)+1 = 1+x+x^2$	2
4	0	2	x^2	2
5	1	3	$x^{3-2}\cdot x^2 + x^{3-4+2-1}\cdot(1+x)$	
			$= 1+x+x^3$	4
6	0	3	$1+x+x^3$	4
7	1	3	$1+x+x^3$	4
8	1	3	$1+x+x^3$	4

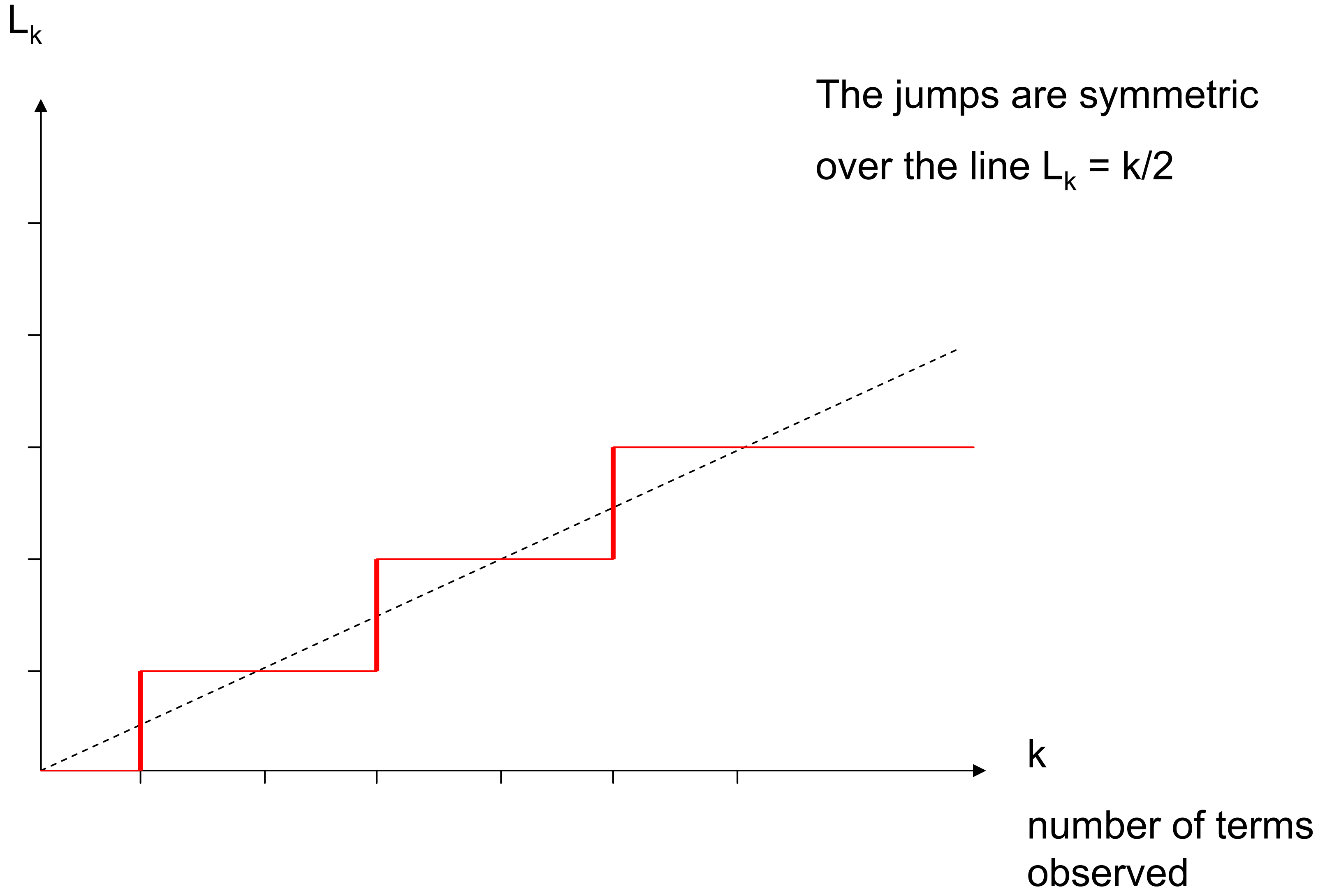
initialisation

the first index such that $z_{r-1}=1$

a jump:
$k=2$, $L_k=1$
$m=0$, $L_m=0$
$k+1=3$, $L_{k+1}=2$

a jump:
$k=4$, $L_k=2$
$m=2$, $L_m=1$
$k+1=5$, $L_{k+1}=3$

LC profile



L_k
The jumps are symmetric
over the line $L_k = k/2$
k
number of terms observed

Linear cryptanlysis

Sections 3.1- 3.3

- Substitution-Permutation Networks
- Piling-up Lemma
- Linear cryptanalysis of SPNs