Lecture 10-11: General attacks on LFSR based stream ciphers

Thomas Johansson

Introduction

- $\mathbf{z} = z_1, z_2, \ldots, z_N$ is a known keystream sequence
- find a distinguishing attack, or
- find a key recovery attack

with complexity lower than exhaustive key search.

Linear complexity and Berlekamp-Massey algorithm

- the keystream sequence $\mathbf{z} = z_1, z_2, \ldots$ will be periodic (after possibly removing some of the first symbols).
- Any such sequence can be generated by an LFSR.
- One possible approach would then be to replace the entire generator with an (in general very long) LFSR.

Definition

The *linear complexity* of a sequence $\mathbf{s}$ (finite or semi-infinite), denoted $L(\mathbf{s})$, is the length of the shortest LFSR that can produce the sequence.

To find the shortest LFSR producing a certain sequence we use the *Berlekamp-Massey algorithm*.

Berlekamp-Massey algorithm

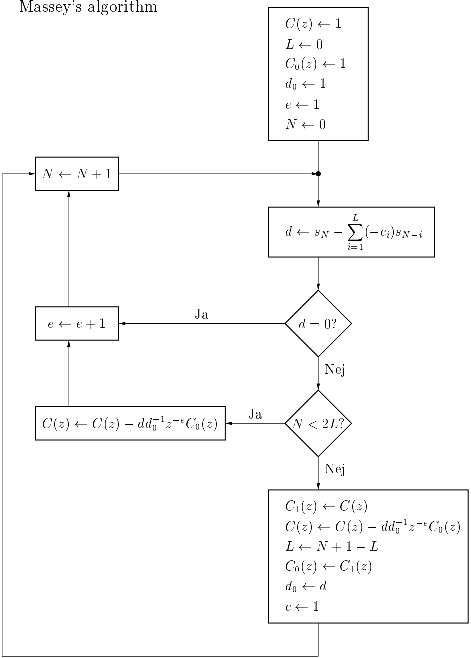
IN: A sequence $\mathbf{s} = (s_0, s_1, \ldots, s_{N-1})$ of length N.

OUT: The shortest LFSR $< C(D), L >$ generating $\mathbf{s}$.

1. *Initilization* $C(D) = 1$, $L = 0$, $C^*(D) = 1$, $d^* = 1$, $m = -1$, $n = 0$.
2. While $(n < N)$ do the following:
   2.1 Compute the discrepancy
   $$d = s_n - \sum_{i=1}^{L} -c_i s_{n-i}.$$
   2.2 If $d \neq 0$ do the following:
   - $T(D) = C(D)$, $C(D) = C(D) - d \cdot (d^*)^{-1} \cdot C^*(D) D^{n-m}$.
   - If $L \leq n/2$ then $L = n + 1 - L$, $C^*(D) = T(D)$, $d^* = d$, $m = n$.

   2.3 $n = n + 1$.
3. Return $< C(D), L >$

Massey's algorithm



Properties of Berlekamp-Massey algorithm

- The running time of the Berlekamp-Massey algorithm for determining the linear complexity of a length n sequence is $O(n^2)$ operations.
- Delivers one connection polynomial and the length L of the LFSR.
- If (and only if) $L \leq N/2$ there is a unique connection polynomial.
- The proof is left out...

Properties of Berlekamp-Massey algorithm

- We want to find the shortest LFSR generating $\mathbf{s}$, a periodic sequence with period T.
- Berlekamp-Massey algorithm can provide the solution if the input is the length $2T$ sequence $(s_0, s_1, \ldots, s_{T-1}, s_0, s_1, \ldots, s_{T-1})$.
- You only need to process the first $T + k$ symbols of the sequence, where k is the first positive integer such that $(s_0, s_1, \ldots, s_{T-1}, s_0, s_1, \ldots, s_{k-1})$ has linear complexity $\leq k$.

Linear complexity

- Let $\mathbf{s^1} = s_0^1, s_1^1, s_2^1, \ldots$ and $\mathbf{s^2} = s_0^2, s_1^2, s_2^2, \ldots$
- $f(x_1, x_2)$ be a function in two variables, $x_1, x_2 \in \mathbb{F}_q$.
- By

$$\mathbf{s} = f(\mathbf{s^1}, \mathbf{s^2})$$

we mean the sequence $\mathbf{s} = f(s_0^1, s_0^2), f(s_1^1, s_1^2), f(s_2^1, s_2^2), \ldots$.

Theorem

Let $\mathbf{s^1}$ *and* $\mathbf{s^2}$ *be two sequences with linear complexity* $L(\mathbf{s^1})$ *and* $L(\mathbf{s^2})$ *respectively. Then*

- *If* $f(x_1, x_2) = x_1 + x_2$ *then* $L(f(\mathbf{s^1}, \mathbf{s^2})) \leq L(\mathbf{s^1}) + L(\mathbf{s^2})$.
- *If* $f(x_1, x_2) = x_1 x_2$ *then* $L(f(\mathbf{s^1}, \mathbf{s^2})) \leq L(\mathbf{s^1}) L(\mathbf{s^2})$.

Example

$$\mathbf{z} = f(\mathbf{s^1}, \mathbf{s^2}, \mathbf{s^3}, \mathbf{s^4}),$$

where $\mathbf{s^i}$, $i = 1, \ldots, 4$ are LFSR sequences with period $2^{L_i} - 1$ (m-sequences). Let

$$f(x_1, x_2, x_3, x_4) = x_1 + x_2x_3 + x_3x_4 + x_2x_4.$$

Find a bound on the linear complexity of the keystream sequence $L(\mathbf{z})$.

Solution: Using Theorem 2 we get

$$L(\mathbf{z}) = L(f(\mathbf{s^1}, \mathbf{s^2}, \mathbf{s^3}, \mathbf{s^4})) \leq L(\mathbf{s^1}) + L(\mathbf{s^2})L(\mathbf{s^3}) + L(\mathbf{s^3})L(\mathbf{s^4}) + L(\mathbf{s^2})L(\mathbf{s^4}) \leq$$

Correlation attacks - idea

- A common method to build a keystream generator is to combine several linear feedback shift registers to get a keystream with desired statistical properties.
- *Correlation attack:* If one can detect a correlation between $\mathbf{z}$ and the output of one individual LFSR, this can be used in a "divide-and-conquer" attack on the individual LFSR.

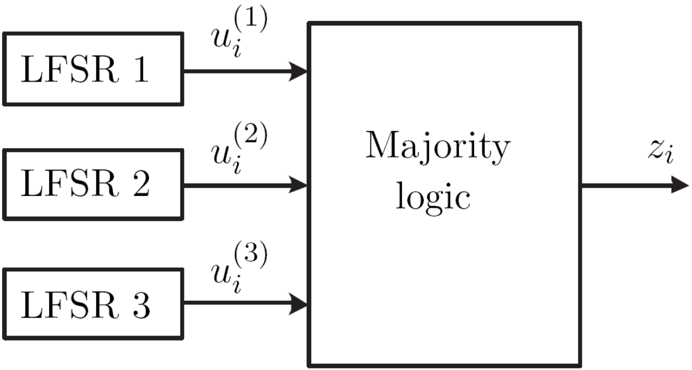


Figure 2: The keystream generator.

Project 3

The values of L_i and $C_i(D)$ for the different LFSRs are,

$$
\begin{aligned}
L_1 &= 13, \quad C_1(D) = 1 + D^1 + D^2 + D^4 + D^6 + D^7 + D^{10} + D^{11} + D^{13}, \\
L_2 &= 15, \quad C_2(D) = 1 + D^2 + D^4 + D^6 + D^7 + D^{10} + D^{11} + D^{13} + D^{15}, \\
L_3 &= 17, \quad C_3(D) = 1 + D^2 + D^4 + D^5 + D^8 + D^{10} + D^{13} + D^{16} + D^{17}.
\end{aligned}
$$

The secret key K is the initial state of the three LFSRs.

$K = (K_1, K_2, K_3)$, where K_i is the initial state of the ith LFSR.

Project 3

Exercise 1:

Each group is given a keystream $z_1, z_2, \ldots, z_N$ of some length N. Find the key K that was used to produce this keystream.

Exercise 2:

Assume that the attack takes T seconds. How long would it take to attack by an exhaustive search over the entire keyspace?

Project 3 - the correlation attack

- a correlation between the output of one of the shift registers and the keystream, i.e., $Pu_i = z_i \neq 0.5$,

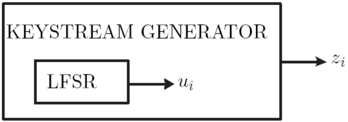


Figure 3: A sufficient requirement for a correlation attack, $\Pr\{u_i = z_i\} \neq 0.5$.

Project 3 - the correlation attack

- Let $u_i^{(j)}$ be the output of the jth LFSR and assume that $Pu_i^{(j)} = z_i = p$, where $p \neq 0.5$.
- What is the exact value of p?
- Guess that the initial state of the jth LFSR is

$$\hat{\mathbf{u}}_{\mathbf{0}} = (\hat{u}_1^{(j)}, \hat{u}_2^{(j)}, \ldots, \hat{u}_{L_j}^{(j)}).$$

We can calculate an LFSR output sequence $\hat{\mathbf{u}} = (\hat{u}_1, \hat{u}_2, \ldots, \hat{u}_N)$, where

$$\begin{aligned} \hat{u}_i =& \hat{u}_i^{(j)}, \quad 0 < i \leq L_j, \\ \hat{u}_i =& \sum_{l=1}^{L_j} c_l \hat{u}_{i-l}, \quad L_j < i \leq N. \end{aligned}$$

Project 3 - the correlation attack

- *The Hamming distance* between $\mathbf{x}$ and $\mathbf{y}$, $d_H(\mathbf{x}, \mathbf{y})$, is defined to be the number of coordinates in which $\mathbf{x}$ and $\mathbf{y}$ differ.
- Estimate the correlation p with p^*, where

$$p^* = 1 - \frac{d_H(\hat{\mathbf{u}}, \mathbf{z})}{N}.$$

- If the guessed initial state, $\hat{\mathbf{u}}_{\mathbf{0}}$, is correct, we get $p^* \approx p$, otherwise $p^* \approx 0.5$.

Algorithm

1. For each possible initial state, calculate p^*;
2. Output the initial state for which p^* is maximal.

Linear distinguishing attacks

- Introduce linear approximations of all nonlinear operations in a specific "path" of the cipher.
- The path should connect some know values, i.e., key stream symbols.
- If the linear approximation is true, this leads to a linear relationship among the known key stream symbols.
- If it is not true, we can think of the error introduced by the linear approximation as truly random noise.
- A linear combination of key stream symbols above can be viewed as a sample from a very noisy (but not uniform) distribution. By collecting many such samples, we can eventually distinguish the distribution they are drawn from, from the uniform distribution.

Example

- A long LFSR with connection polynomial $C(D) = 1 + D^{34} + D^{69}$ is generating a sequence $\mathbf{s} = (s_0, s_1, s_2, \ldots)$ in $\mathbb{F}_2$. The output of the generator is generated as

$$z_i = f(s_i, s_{i+1}, s_{i+2}, s_{i+3}), i = 0, 1, \ldots,$$

  where f is the Boolean function
  $f(x_1, x_2, x_3, x_4) = x_1 + x_2 + x_3 + x_1x_2x_3x_4$. *filter generator*
- Describe a linear distinguishing attack on the generator.

Example

- The LFSR sequence obeys the recursion

$$s_i + s_{i+35} + s_{i+69} = 0, i = 0, 1, \ldots.$$

- Replace f with the linear function $g(x_1, x_2, x_3, x_4) = x_1 + x_2 + x_3$.
- After replacement, we have

$$z_i = g(s_i, s_{i+1}, s_{i+2}, s_{i+3}) = s_i + s_{i+1} + s_{i+2}, i = 0, 1, \ldots.$$

  Now we try to find a set of dependent linear equations.
- As $s_i + s_{i+35} + s_{i+69} = 0$ we also have

$$s_i+s_{i+1}+s_{i+2}+s_{i+35}+s_{i+36}+s_{i+37}+s_{i+69}+s_{i+70}+s_{i+71} = 0, i = 0, 1, \ldots$$

Example cont'

- But $z_i = s_i + s_{i+1} + s_{i+2}$, so we must have $z_i + z_{i+35} + z_{i+69} = 0$ (assuming that g always gives the result of f).
- So in our attack we create a sequence of sample values $Q_i = z_i + z_{i+35} + z_{i+69}, i = 0, 1, \ldots$. Calculating $P(Q_i)$ gives $P(Q_i = 0) = 0.835$.
- The number of zeros in $\mathbf{Q}$ has a binomial distribution with probability 0.835, whereas a random $\mathbf{Q}$ probability 0.5. Apply hypothesis testing!