Linear Cryptanalysis of Stream Ciphers

T-79.514 Special Course on Cryptology

Seminar talk

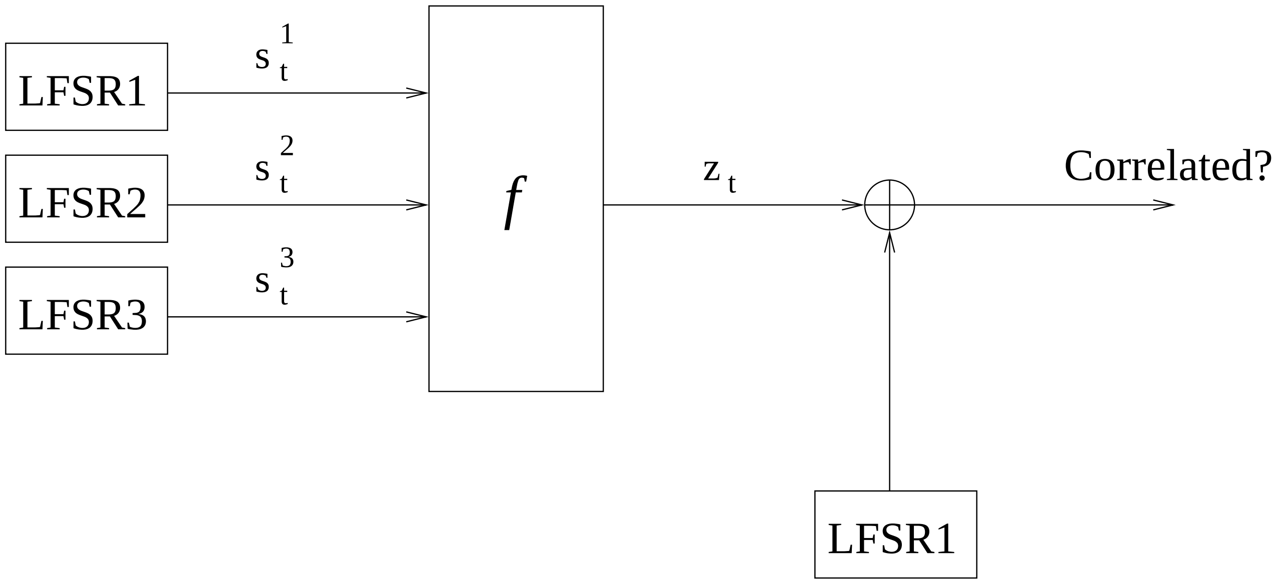
Emilia Käsper

Overview

- Basic concept of correlation attacks on stream ciphers
- A correlation attack on the GSM cipher A5/1
- A correlation attack on the Bluetooth cipher E_0

- Linear cryptanalysis studies the correlation between linear combinations of input and output bits of functions.
- In the usual case of (binary additive) stream ciphers
  - the function under study is a nonlinear combiner function;
  - the input bits to the function are bits from LFSR bitstreams;
  - the output bits are the keystream bits;
  - known plaintext-ciphertext sequences allow us to obtain known keystream.

Principles of the correlation attack



Divide-and-conquer attack

- Assume a nonlinear combining generator with N LFSR-s of lengths $l_1, \ldots, l_N$.
- Exhaustive search then has to be performed over

$$\prod_{i=1}^{N} (2^{l_i} - 1)$$

  initial states.
- If each of the LFSR streams is correlated with the (known) keystream, we can test each of the LFSR-s separately, so the complexity reduces to

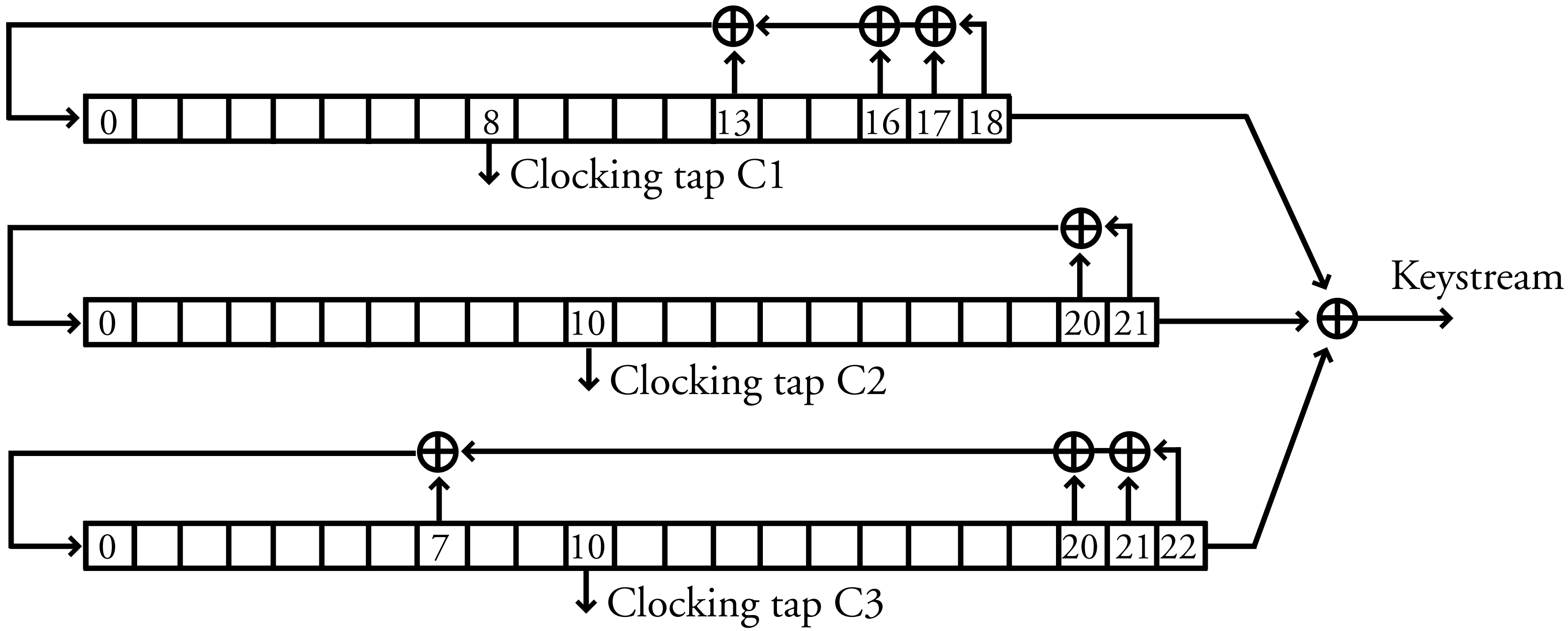
$$\sum_{i=1}^{N} (2^{l_i} - 1).$$

- Example: the Geffe generator (1973) is defined by three maximum-length LFSR-s and a combining function

$$f(x_1, x_2, x_3) = x_1x_2 \oplus x_2x_3 \oplus x_3.$$

- $P(z(t) = x_1(t)) = \frac{3}{4}$, $P(z(t) = x_3(t)) = \frac{3}{4}$
- If the combining function is correlation immune to the 1^{st} order, we need to consider the LFSR-s pairwise, etc.
- If a boolean function f is m^{th} order correlation immune, then the nonlinear order of f is at most $n - m$.
- The correlation immunity-nonlinear order tradeoff can be avoided by e.g.
  - irregular clocking, as in the case of A5/1 or
  - using memory in the function, as in the case of E_0.

The GSM encryption cipher A5/1



A correlation attack on A5/1

- The initial state of the A5/1 generator is a linear function of the key and the frame number (IV).
- Each output bit of an LFSR is a linear combination of key and frame number bits:

$$s_t^R = \sum_{i=1}^{64} c_{it}^R k_i + \sum_{i=1}^{22} d_{it}^R f_i$$

- Separate the key and frame number parts in each of the LFSR-s:

$$s_t^R = \hat{k}_t^R + \hat{f}_t^R.$$

- The sequences $\hat{k}_0^R, \hat{k}_1^R, \ldots$ are unknown, but remain the same for all frames.
- The sequences $\hat{f}_0^R, \hat{f}_1^R, \ldots$ can be derived for each frame.

Basic idea for the attack

- Each of the LFSR-s is clocked on average three times out of four
- *Assume* for a moment that after 101 clockings, each of the LFSR-s has been clocked exactly 76 times. Then

$$s^1_{76} + s^2_{76} + s^3_{76} = z_1,$$

or

$$\hat{k}^1_{76} + \hat{k}^2_{76} + \hat{k}^3_{76} = \hat{f}^1_{76} + \hat{f}^2_{76} + \hat{f}^3_{76} + z_1 \tag{1}$$

- Denote the *known* rhs of (1) for frame j by $O^j_{(76,76,76,1)}$
- Then we obtain a correlation for the key bit combinations:

$$\begin{aligned} & P(\hat{k}^1_{76} + \hat{k}^2_{76} + \hat{k}^3_{76} = O^j_{(76,76,76,1)}) = \\ = \quad & P(\text{assumption correct}) \cdot 1 + P(\text{assumption wrong}) \cdot \frac{1}{2}. \end{aligned}$$

A refinement of the attack

- The probability of the particular clocking $(76, 76, 76, 1)$ is around 10^{-3}.
- The basic attack requires a few million frames (hours of conversation) to determine information about the key.
- Consider now *all* keystream positions where a clocking triple has a non-negligible probability of occuring and take a weighted decision for each frame:

$$\begin{aligned} p^j_{cl_1,cl_2,cl_3} &= P(\hat{k}^1_{cl_1} + \hat{k}^2_{cl_2} + \hat{k}^3_{cl_3} = 0) = \\ = & \sum_{v \in \mathcal{I}} P(cl_1, cl_2, cl_3, v) \cdot [O^j_{cl_1,cl_2,cl_3,v-100} = 0] + \\ + & \frac{1}{2} \cdot (1 - \sum_{v \in \mathcal{I}} P(cl_1, cl_2, cl_3, v)). \end{aligned}$$

- To evaluate clocking probabilities, assume that the clock control bits are uniformly distributed independent bits:

$$P(cl_1, cl_2, cl_3, v) = \frac{\binom{v}{v-cl_1}\binom{v-(v-cl_1)}{v-cl_2}\binom{v-(v-cl_1)-(v-cl_2)}{v-cl_3}}{4^v}.$$

- Use the log-likelihood ratio

$$\Lambda_{(cl_1,cl_2,cl_3)} = \sum_{j=1}^{m} \ln \frac{p^j_{cl_1,cl_2,cl_3}}{1 - p^j_{cl_1,cl_2,cl_3}}$$

to estimate the linear combination $\hat{k}^1_{cl_1} + \hat{k}^2_{cl_2} + \hat{k}^3_{cl_3}$.

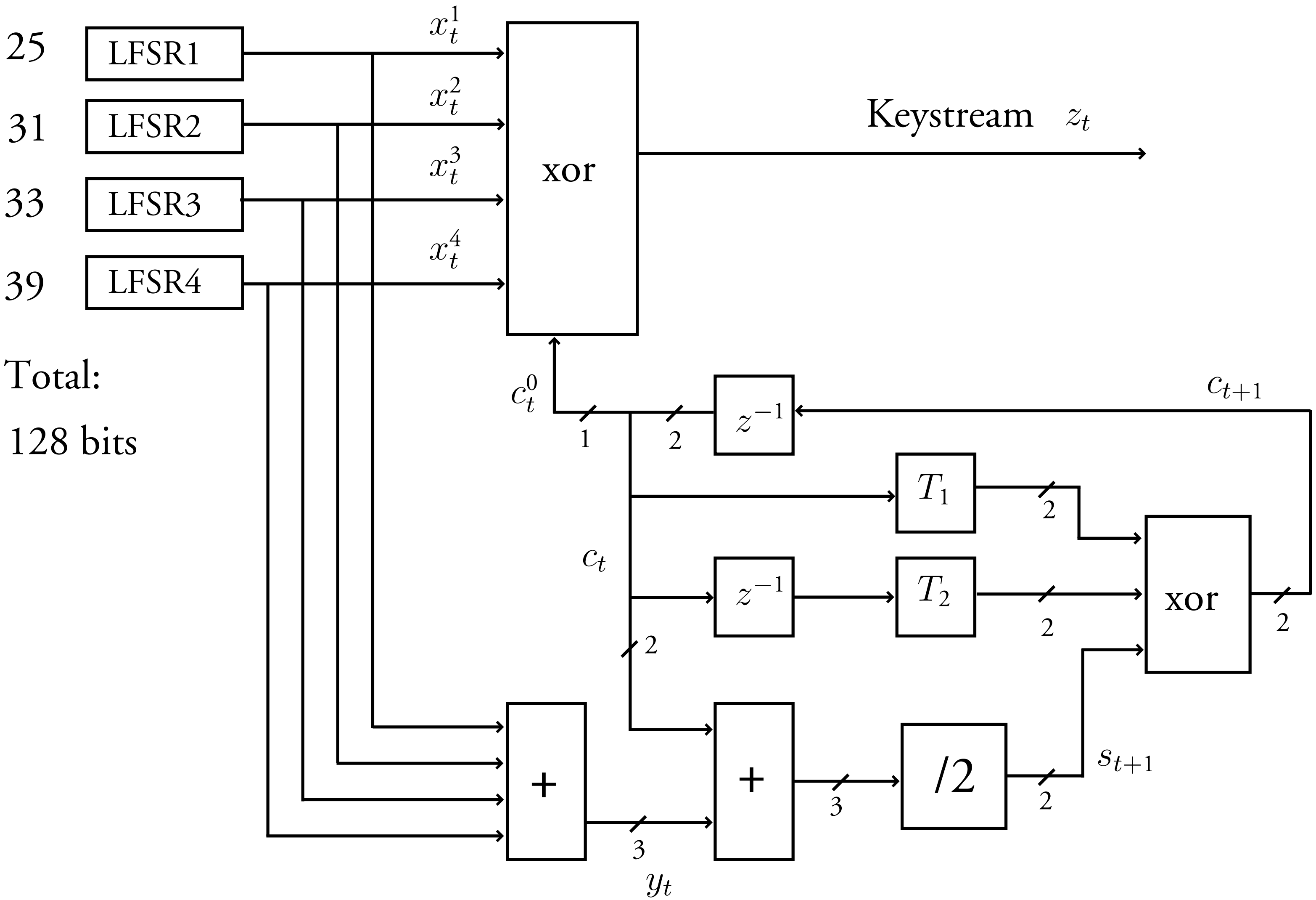
- Recall that the bit $\hat{k}^R_{cl_i}$ is the i^{th} output bit of the LFSR R, when loaded only with key bits.
- If we recover enough (consecutive) bits $\hat{k}^R_{cl_i}$, we can load them into the registers, clock the cipher (regularly) backwards, load a frame number and check against the known keystream.
- If we consider all clocking triples in an interval of length N, we obtain N^3 linear equations with $3N$ variables.
- The problem of finding the variables is equivalent to decoding a linear code.

Divide and conquer

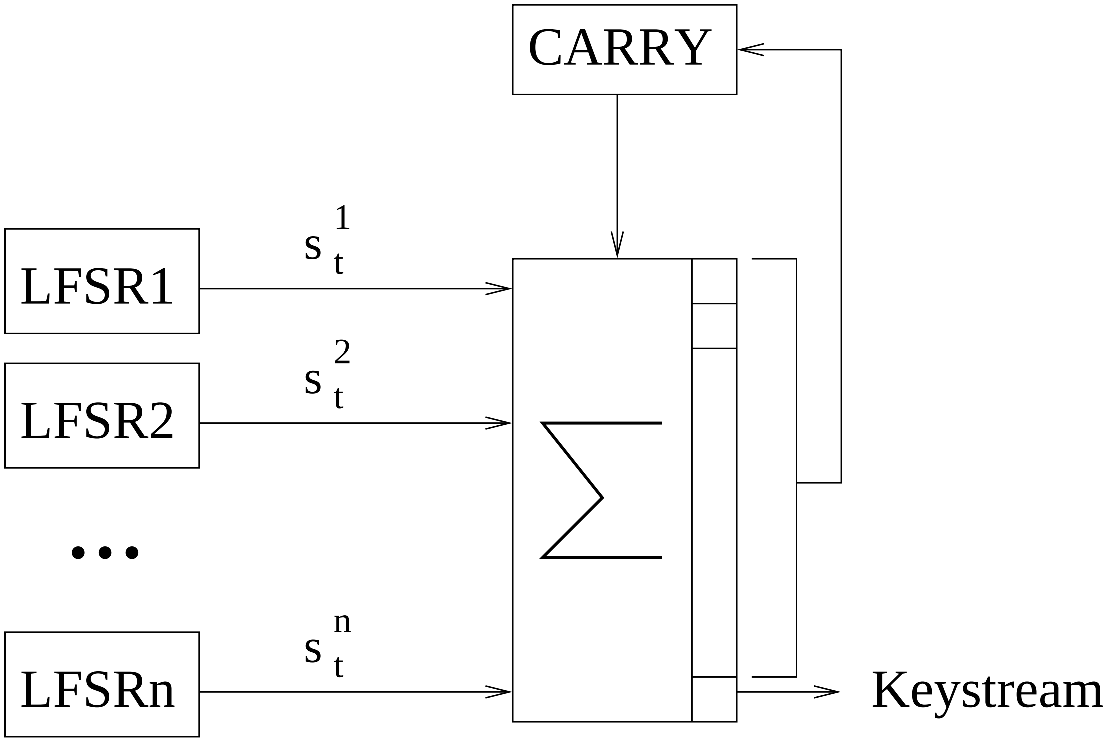
- We need 64 bits of information — exhaustive search over one interval of length at least 22 gives no advantage over brute-force attack.
- Consider instead several shorter intervals, e.g. pick $N = 8$ and intervals $[79, \ldots, 86], [87, \ldots, 94], [95, \ldots, 102]$.
- We now need to perform exhaustive searches over only 24 variables.
- What if the closest solution is erroneous?
- We can either increase the number of received frames...
- ... or check for T closest solutions.

- T solutions from each interval give T^3 combinations of solutions.
- To reduce the number of solutions to be verified, use overlapping intervals and the properties of the feedback polynomials.
- With parameters $N = 9$ and $T = 1000$, the attack has been implemented and gives 75% success probability, using 70000 frames (5 min) of known plaintext.

The Bluetooth encryption cipher E_0



- Integer addition over $\mathbb{Z}_2$ defines a nonlinear function with memory whose correlation immunity is maximum.
- This idea was first employed in the summation generator (1985)



A correlation attack on E_0

- The only nonlinear part of the keystream is the sequence c_t^0.
- Correlations for the sequence have been identified, e.g.

$$P(c_t^0 \oplus c_{t-5}^0 = 0) = \frac{1}{2} + 0.04883.$$

- To mount a correlation attack, we can replace the nonlinear part with a sequence of random variables having certain correlation probability.

Divide and conquer

- Guess the initial state of LFSR1 and denote its output sequence by (x_t).
- Model the other three LFSR-s as a single LFSR and denote its (unknown) output sequence by (u_t).
- Assume that (c_t) is a random noise sequence with the above correlation probability $\frac{1}{2} + \epsilon$.
- Then

$$z_t = x_t \oplus u_t \oplus c_t,$$

or

$$z_t \oplus x_t = u_t \oplus c_t,$$

where the lhs (denote it by v_t) is known.

- We shall now identify a correlation probability for v_t to verify our guess.
- For this, we need to eliminate the influence of the sequence u_t.
- The sequence $\mathbf{u}= (u_0, u_1, \ldots, u_{N-1})$ has generator matrix G such that $\mathbf{u} = \mathbf{u_0 G}$.
- Suppose we are able to find k columns $i_1, \ldots, i_k$ in G that add up to a zero-column.
- Then also $u_{t+i_1} + \ldots + u_{t+i_k} = 0$ for any time index t (since the code is cyclic).

- Now

$$\sum_{i\in\mathcal{I}} v_{t+i} + v_{t+i-5} = \sum_{i\in\mathcal{I}} (c_{t+i} + u_{t+i}) + (c_{t+i-5} + u_{t+i-5}) =$$

$$= \sum_{i\in\mathcal{I}} c_{t+i} + c_{t+i-5}$$

and

$$P\left(\sum_{i\in\mathcal{I}} v_{t+1} + v_{t+i-5} = 0\right) = \frac{1}{2} + 2^{k-1}\epsilon^{k}.$$

- The attack has two parameters that will influence the length of the received keystream:
  - w, the value of the highest index in $\mathcal{I}$ (or, in other words, the number of columns required to find k columns that sum to a zero-column) and
  - m, the number of time samples required to gain statistical significance.
- **Theorem** Assume a cyclic code with a random generator matrix. The total number of columns, w, required to find k columns that add up to the all-zero column is approximately $2^{\frac{l}{k-1}}$, where l is the number of rows in the matrix.
- Hence, w decreases when k increases.

- On the other hand, when k increases, the probability $\frac{1}{2} + 2^{k-1}\epsilon^k$ tends to $\frac{1}{2}$, i.e. the correlation gets weaker.
- Hence, m increases when k increases.
- Recall that the available keystream from one frame is at most 2745 bits.
- The required length of keystream is found to be $> 2^{34}$ bits, thus, the attack cannot be applied on the actual Bluetooth encryption scheme.