

# Cryptanalysis

## Part III: Stream Ciphers

André Schrottenloher  
Stealing slides from Maxime Bombar

Inria Rennes  
Team CAPSULE



1 LFSR

2 Combining LFSRs

3 Cryptanalysis

# Stream ciphers

- Hardware implementations
- Very fast, energy-efficient

- WEP (Wifi, norme IEEE 802.11, 1999): RC4 (broken)
- Bluetooth: E0
- 2G/3G: A5/1 (backdoored)
- 4G/5G: SNOW

# They still exist

- eSTREAM competition (2004-2008)  $\implies$  7 new ciphers, incl. Salsa20
- Chacha20: new variant of Salsa20 with better performance, used in TLS1.3
- GrainAEAD: finalist of the NIST Lightweight standardization (although ASCON won)

LFSR

# Feedback Shift Register

**Objective:** create a pseudorandom sequence that depends only on the initial internal state.

$(s_n)_{n \in \mathbb{N}} \in \mathbb{F}_q^{\mathbb{N}}$  is produced by a feedback function  $F$  if there exists  $\ell \in \mathbb{N}$  and  $F : \mathbb{F}_q^\ell \rightarrow \mathbb{F}_q$  such that:

$$\forall n \in \mathbb{N}, s_{n+\ell} = F(s_{n+\ell-1}, s_{n+\ell-2}, \dots, s_{n+1}, s_n)$$

# Linear Feedback Shift Register

$(s_n)_{n \in \mathbb{N}} \in \mathbb{F}_q^{\mathbb{N}}$  is produced by a **linear feedback function** if there exists  $\ell \in \mathbb{N}$  and  $F : \mathbb{F}_q^\ell \rightarrow \mathbb{F}_q$  a **linear form** such that:

$$\forall n \in \mathbb{N}, s_{n+\ell} = F(s_{n+\ell-1}, s_{n+\ell-2}, \dots, s_{n+1}, s_n)$$

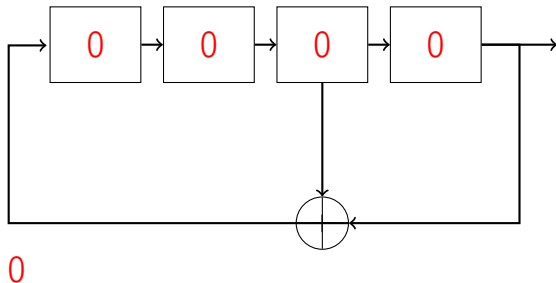
$(s_n)$  is produced by an LFSR if there exists a **constant vector**  $(c_1, \dots, c_\ell) \in \mathbb{F}_q^\ell$  such that:

$$\forall n, s_{n+\ell} = c_1 s_{n+\ell-1} + \dots + c_\ell s_n$$

Every linear recurrent sequence of order  $\ell$  is produced by an LFSR of length  $\ell$ .

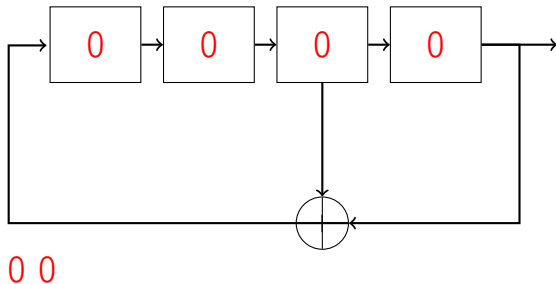
# Example: binary LFSR

$$\begin{cases} s_{n+4} = s_n + s_{n+1} \\ c_1, c_2, c_3, c_4 = 0, 0, 1, 1 \end{cases} \quad (1)$$



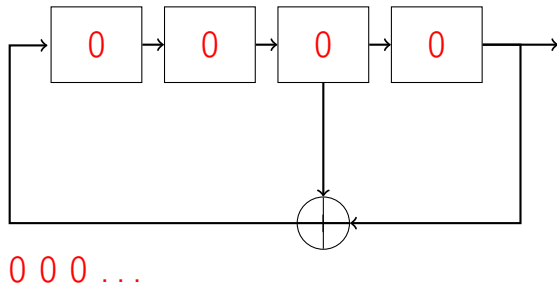
# Example: binary LFSR

$$\begin{cases} s_{n+4} = s_n + s_{n+1} \\ c_1, c_2, c_3, c_4 = 0, 0, 1, 1 \end{cases} \quad (1)$$

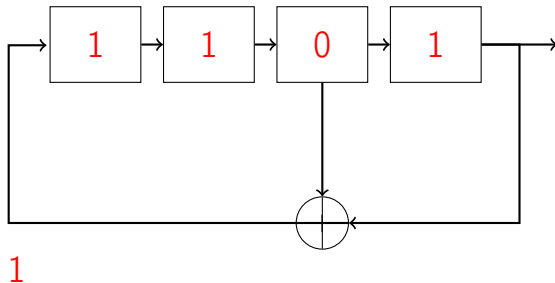


# Example: binary LFSR

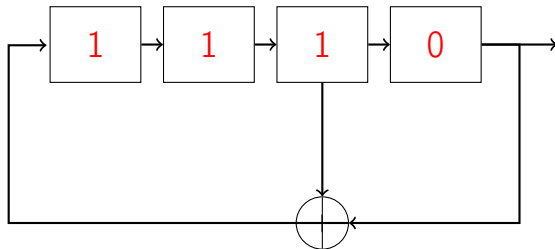
$$\begin{cases} s_{n+4} = s_n + s_{n+1} \\ c_1, c_2, c_3, c_4 = 0, 0, 1, 1 \end{cases} \quad (1)$$



# Example: nontrivial binary LFSR

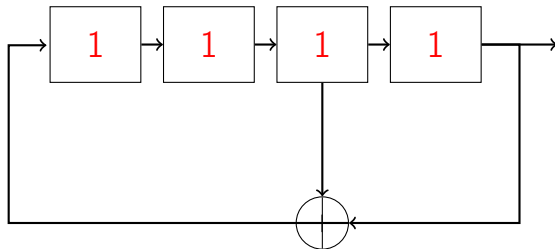


# Example: nontrivial binary LFSR



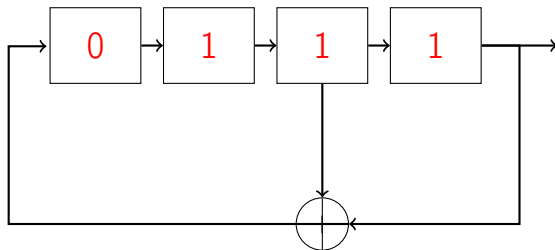
10

# Example: nontrivial binary LFSR



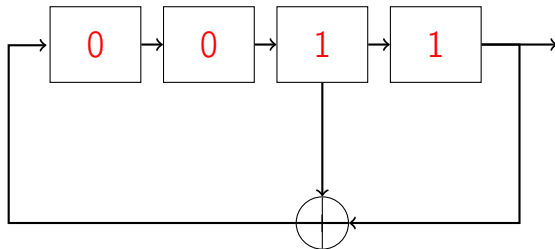
101

# Example: nontrivial binary LFSR



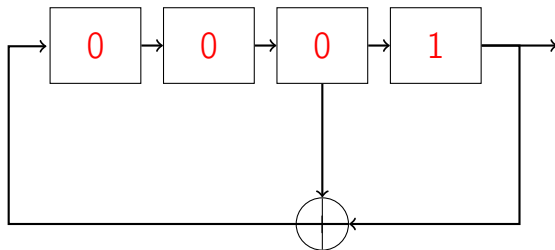
1011

# Example: nontrivial binary LFSR



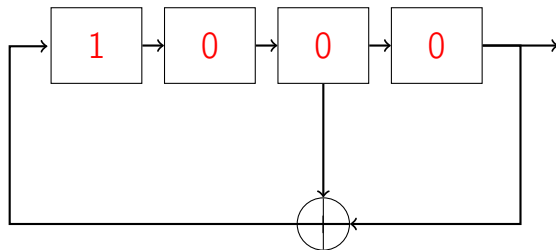
10111

# Example: nontrivial binary LFSR



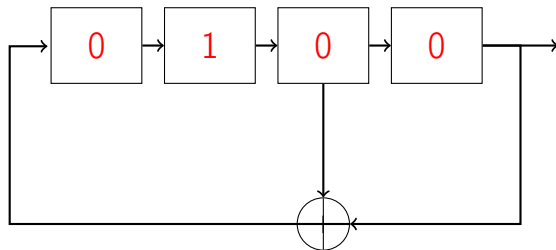
101111

# Example: nontrivial binary LFSR



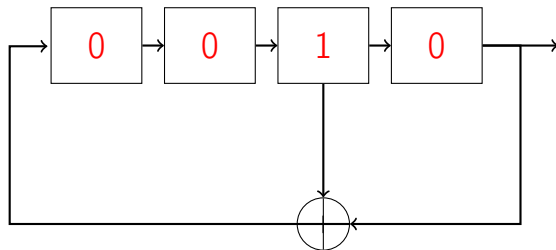
1011110

# Example: nontrivial binary LFSR



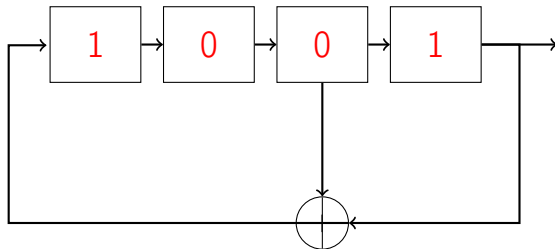
10111100

# Example: nontrivial binary LFSR



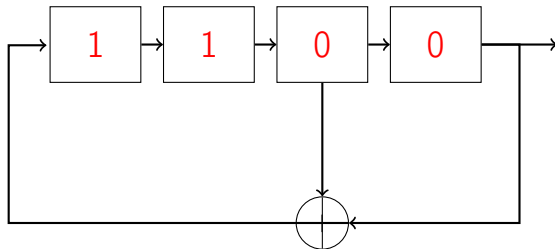
101111000

# Example: nontrivial binary LFSR



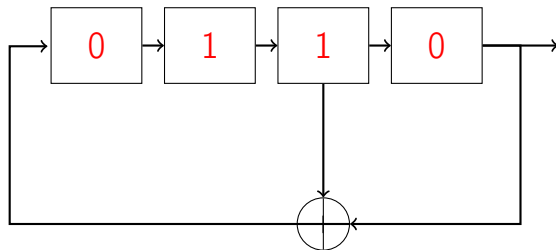
1011110001

# Example: nontrivial binary LFSR



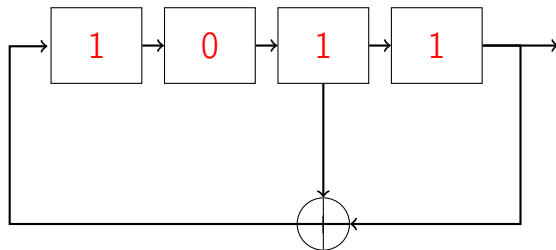
10111100010

# Example: nontrivial binary LFSR



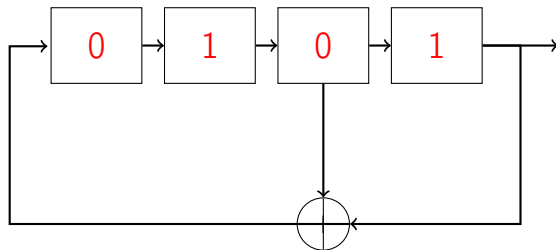
101111000100

# Example: nontrivial binary LFSR



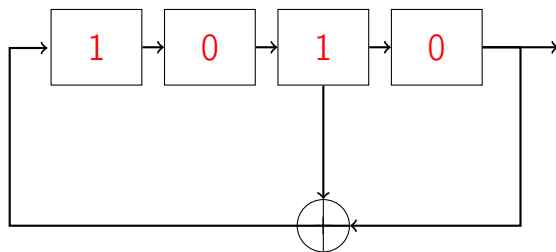
1011110001001

# Example: nontrivial binary LFSR



10111100010011

# Example: nontrivial binary LFSR



101111000100110 ...

- The period is  $15 = 2^4 - 1$
- It is maximal!

# Retroaction polynomial

Consider an LFSR of length  $\ell$  and coefficients  $c_1, \dots, c_\ell \in \mathbb{F}_q^\ell$ . We define the **retroaction polynomial** by:

$$P(X) = 1 - \sum_{i=1}^{\ell} c_i X^i \in \mathbb{F}_q[X]$$

The previous LFSR has the polynomial:

$$P(X) = 1 + X^3 + X^4 \in \mathbb{F}_2[X]$$

---

+ and - in  $\mathbb{F}_2$  are the same thing.

# Sparsification

Any sequence produced by a LFSR of retroaction  $P$  can be produced by any LFSR of retroaction a multiple of  $P$ .

Example: let  $(s_n)$  be a sequence in  $\mathbb{F}_2$  that satisfies:

$$s_{n+6} = s_{n+4} + s_{n+3} + s_{n+1} + s_n, \forall n \geq 6$$

Its retroaction polynomial is  $P(X) = 1 + X^2 + X^3 + X^5 + X^6$ . The sequence also satisfies:  $s_{n+8} = s_{n+7} + s_n$ , since:

$$1 + X + X^8 = (1 + X + X^2)P(X)$$

# Minimisation

Let  $(s_n)_{n \in \mathbb{N}}$  be a linear recurrent sequence.

Among all retroaction polynomials for  $(s_n)$ , there exists one of **minimal degree**.

# Period

Let  $(s_n)_{n \in \mathbb{N}}$  be a linear recurrent sequence and  $P$  its **minimal** retroaction polynomial. Let  $\ell$  be its degree.

The period of  $s$  is  $q^\ell - 1$  iff  $P$  is **primitive**.

**Example:**  $P(X) = 1 + X^3 + X^4$  is a primitive polynomial; so the period is going to be  $2^4 - 1 = 15$ .

A primitive LFSR has good statistical properties, but cannot be used alone to construct a stream cipher: we **combine** multiple LFSRs and use a **filtering function**.

---

Primitive polynomial = monic and one of its roots is a primitive  $q^\ell - 1$ -root of unity.

# The Berlekamp-Massey algorithm

If the minimal polynomial is of degree  $d$ , the Berlekamp-Massey algorithm can find it from  $2d$  terms of the sequence.

$\Rightarrow$  useful if we do not know a retroaction polynomial (e.g., for cryptanalysis).

## Combining LFSRs

# Objective

Combine the outputs of multiple LFSRs:

$$s_t = f(s_t^{(1)}, \dots, s_t^{(n)})$$

where  $f : \mathbb{F}_2^n \rightarrow \mathbb{F}_2$  is a Boolean function.

- Each LFSR has a primitive retroaction polynomial.
- The characteristics (length, polynomials) are public.
- The initial states of each LFSR (formed from the secret key + IV) are secret.

# Interlude: Boolean functions

## Definition

A Boolean function in  $n$  variables is a function  $f : \mathbb{F}_2^n \rightarrow \mathbb{F}_2$ . It can be described by its truth table.

How many  $n$ -variable Boolean functions are there?

# Support and weight

- The support of a Boolean function is:

$$\text{Supp}(f) = \{\mathbf{x} \in \mathbb{F}_2^n, f(\mathbf{x}) \neq 0\}$$

- The weight of  $f$  is  $w(f) = |\text{Supp}(f)|$
- $f$  is balanced if  $w(f) = 2^{n-1}$

We need the Boolean function to be balanced (otherwise the output will be biased).

# Algebraic normal form

There exists a unique multivariate polynomial  $\bar{f}$  such that:

$$\bar{f}(X_1, \dots, X_n) = \sum_{\mathbf{u}=(u_1, \dots, u_n) \in \mathbb{F}_2^n} a_{\mathbf{u}} X_1^{u_1} \cdots X_n^{u_n}$$

such that:  $f(x_1, \dots, x_n) = \bar{f}(x_1, \dots, x_n)$

$\bar{f}$  is the **ANF** of  $f$ , and:

$$a_{\mathbf{u}} = \sum_{x \preceq \mathbf{u}} f(x) \text{ where } x \preceq y \text{ iff } x_i \leq y_i \text{ for all } i$$

---

Remember that this is a sum on  $\mathbb{F}_2$ .

# Algebraic degree

The algebraic complexity of a Boolean function is quantified by the **degree** of its ANF: if

$$f(X_1, \dots, X_n) = \sum_{\mathbf{u} \in \mathbb{F}_2^n} a_{\mathbf{u}} X_1^{u_1} \cdots X_n^{u_n}$$

then

$$\deg(f) = \max\{hw(\mathbf{u}) \mid a_{\mathbf{u}} \neq 0\}$$

where  $hw(\mathbf{u})$  is the Hamming weight of  $\mathbf{u}$  (number of ones).

This is the **maximal number of variables in a monomial** of  $f$ .

A “random” Boolean function has very large degree ( $n - 1$ ). Having small degree is a property that can be used for cryptanalysis.

# Example

Geffe proposed to use the function defined by the following truth table to combine 3 LFSRs.

|                    |  |   |   |   |   |   |   |   |   |
|--------------------|--|---|---|---|---|---|---|---|---|
| $x_1$              |  | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| $x_2$              |  | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| $x_3$              |  | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| <hr/>              |  |   |   |   |   |   |   |   |   |
| $f(x_1, x_2, x_3)$ |  | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |

We can see that:

# Example

Geffe proposed to use the function defined by the following truth table to combine 3 LFSRs.

|                    |  |   |   |   |   |   |   |   |   |
|--------------------|--|---|---|---|---|---|---|---|---|
| $x_1$              |  | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| $x_2$              |  | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| $x_3$              |  | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| <hr/>              |  |   |   |   |   |   |   |   |   |
| $f(x_1, x_2, x_3)$ |  | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |

We can see that:

$$\text{Supp}(f) = \{(0, 0, 1), (1, 0, 1), (1, 1, 0), (1, 1, 1)\}$$

and  $w(f) = 4$ .

# ANF

$$a_{000} = f(0, 0, 0) = 0$$

$$a_{001} = f(0, 0, 0) + f(0, 0, 1) = 1$$

$$a_{010} = f(0, 0, 0) + f(0, 1, 0) = 0$$

$$a_{011} = f(0, 0, 0) + f(0, 1, 0) + f(0, 0, 1) + f(0, 1, 1) = 1$$

$$a_{100} = f(0, 0, 0) + f(1, 0, 0) = 0$$

$$a_{101} = f(0, 0, 0) + f(1, 0, 0) + f(0, 0, 1) + f(1, 0, 1) = 0$$

$$a_{110} = f(0, 0, 0) + f(1, 0, 0) + f(0, 1, 0) + f(1, 1, 0) = 1$$

$$a_{111} = \sum_{\mathbf{u}} f(\mathbf{u}) = w(f) \mod 2 = 0$$

## ANF

$$a_{000} = f(0, 0, 0) = 0$$

$$a_{001} = f(0, 0, 0) + f(0, 0, 1) = 1$$

$$a_{010} = f(0, 0, 0) + f(0, 1, 0) = 0$$

$$a_{011} = f(0, 0, 0) + f(0, 1, 0) + f(0, 0, 1) + f(0, 1, 1) = 1$$

$$a_{100} = f(0, 0, 0) + f(1, 0, 0) = 0$$

$$a_{101} = f(0, 0, 0) + f(1, 0, 0) + f(0, 0, 1) + f(1, 0, 1) = 0$$

$$a_{110} = f(0, 0, 0) + f(1, 0, 0) + f(0, 1, 0) + f(1, 1, 0) = 1$$

$$a_{111} = \sum_{\mathbf{u}} f(\mathbf{u}) = w(f) \pmod{2} = 0$$

$$f(X_1, X_2, X_3) = X_3 + X_2X_3 + X_1X_2 \text{ and } \deg(f) = 2$$

# Linear complexity of the combined sequence

The degree of the minimal polynomial is named **linear complexity** of the sequence and noted  $\Lambda(s)$ .

## Lemma (Rueppel, Staffelbach, 1987)

Let  $s^1$  and  $s^2$  be two linear recurrent sequences, of minimal polynomials  $P^1$  and  $P^2$ . Then:

- $\Lambda(s^1 + s^2) \leq \Lambda(s^1) + \Lambda(s^2)$  with equality iff  $\gcd(P^1, P^2) = 1$
- $\Lambda(s^1 * s^2) \leq \Lambda(s^1)\Lambda(s^2)$ . If  $P^1, P^2$  are primitive, of degrees distinct and bigger than 2, there is an equality.

---

Here  $s^1 * s^2$  means the pointwise product of the sequences.

# Linear complexity of the combined sequence

## Corollary

Let  $s^1, \dots, s^n$  be linear recurrent sequences produced by LFSRs of respective lengths  $\ell^1, \dots, \ell^n$ . Let  $f : \mathbb{F}_2^n \rightarrow \mathbb{F}_2$  be a Boolean function. The combined sequence  $f(s^1, \dots, s^n)$  has linear complexity:

$$\Lambda = f(\ell^1, \dots, \ell^n)$$

obtained by evaluating the ANF of  $f$  as a polynomial in  $\mathbb{Z}$ .

**Example:** for Geffe's cipher:  $f(x_1, x_2, x_3) = x_3 + x_2x_3 + x_1x_2$   
The linear complexity is  $\Lambda = \ell^3 + \ell^2\ell^3 + \ell^1\ell^2$ .

# Filtered LFSRs

- Take several bits of the same LFSR
- Equivalent: Combine  $k$  LFSRs with the same retroaction polynomial, but shifted initial states
- Caution: previous results do not apply

(Edwin L. Key, 1976)

The linear complexity  $\Lambda(s)$  of a sequence  $s$  produced by an LFSR of length  $\ell$  and filtered by a Boolean function of degree  $d$  satisfies:

$$\Lambda(s) \leq \sum_{i=0}^d \binom{\ell}{i}$$

(Rueppel, 1986)

When  $\ell$  is prime and big enough, then  $\Lambda(s) \simeq \binom{\ell}{d}$  for most degree- $d$  Boolean functions.

# Cryptanalysis

# Correlation attacks

- Consider  $n$  LFSRs of lengths  $\ell_1, \dots, \ell_n$  with a post-processing function  $f$ .
- **Goal:** find the internal states.

# Correlation attacks

- Consider  $n$  LFSRs of lengths  $\ell_1, \dots, \ell_n$  with a post-processing function  $f$ .
- **Goal:** find the internal states.

Exhaustive search:  $= \prod_{i=1}^n (2^{\ell_i} - 1)$

# Correlation attack: principle

If  $f$  is **badly chosen**, the output sequence may be correlated to a sequence formed by **less LFSRs**.

- Perform an exhaustive search of the internal states of these LFSRs
- Check if the output sequence is correlated as we expect
- Once the internal state of an LFSR is obtained, continue with the others

e.g.,  $\prod_i (2^{\ell_i} - 1) \rightarrow \sum_i (2^{\ell_i} - 1)$

# Countermeasures

A boolean function  $f$  is **uncorrelated to order  $k$**  if for all independent random variables  $X_1, \dots, X_n$ , the random variable  $f(X_1, \dots, X_n)$  is independent from any  $(X_{i_1}, \dots, X_{i_k})$ . The largest such  $k$  is the **immunity** of  $f$  to correlations.

We need to choose  $f$  with a strong immunity, and a large algebraic degree.