

# Building Stream Ciphers from FCSRs

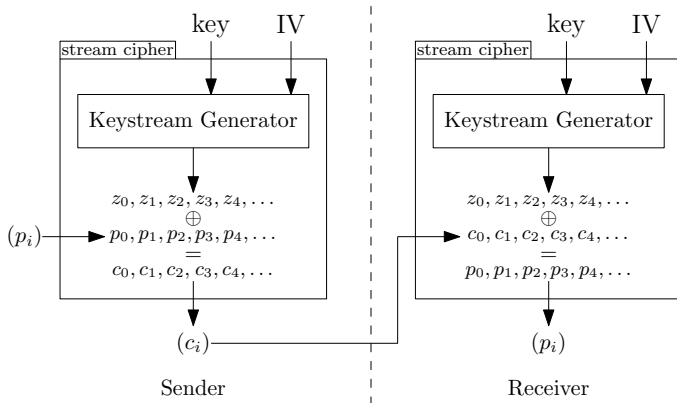
Dirk Stegemann

Theoretical Computer Science  
University of Mannheim  
68161 Mannheim (Germany)

Weekend of Cryptography (Kryptowochenende) 2008  
July 04–06, 2008  
Tabarz, Germany

# (Additive) Stream Ciphers — Basic Structure

Stream Ciphers are used for encrypting/decrypting data streams:



Main Application: hardware-constraint environments such as GSM, Bluetooth, Sensor Networks, ...

# Attack Model

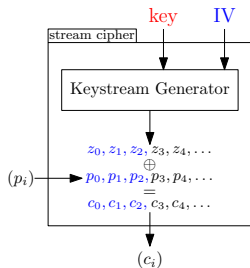
The (passive) attacker **knows**

- the definition of the cipher
- the IV
- the ciphertext stream  $(c_i)$
- some pairs  $(p_i, c_i)$ ,  $0 \leq i \leq n_0$   
 → corresponding keystream bits  $z_i$

and wants to reconstruct the whole keystream  $(z_i)$ .

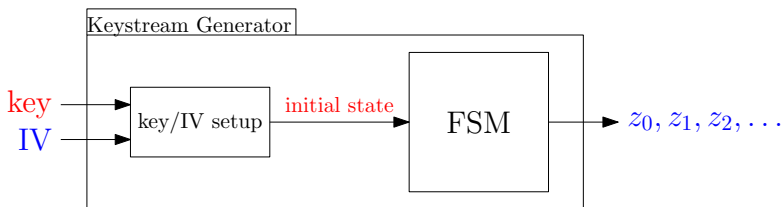
Generic strategy: Recover the secret **key**.

Generic countermeasure: Cipher should behave like a pseudo one-time pad, i.e.,  $(z_i)_{t \geq 0}$  should not be efficiently distinguishable from a truly random sequence.



# The Keystream Generator

Many stream ciphers use an FSM-based keystream generator



to produce the keystream bits  $z_i$ .

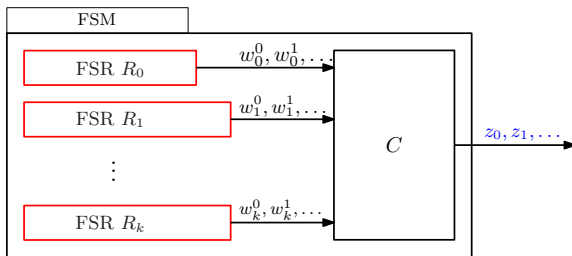
**Observation:** In order to reproduce the keystream, it is sufficient to recover the initial state of the FSM.

Popular hardware-oriented designs for the FSM:

- combination generators
- filter generators

based on Feedback Shift Registers (FSRs)

# Combination Generators



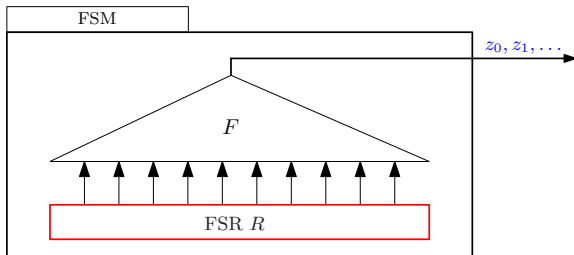
## The FSM

- stores the state in Feedback Shift Registers (FSRs),
- produces keystream by combining the FSRs' output bits.

## Examples:

- Bluetooth keystream generator  $E_0$
- GSM keystream generator A5/1
- eSTREAM-recommended stream cipher Trivium

# Filter Generators



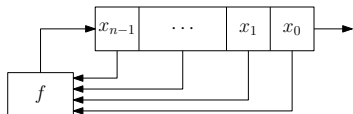
## The FSM

- stores the state in a single FSR,
- produces keystream by filtering the current FSR content.

## Examples:

- eSTREAM-recommended ciphers Grain and F-FCSR

# Feedback Shift Registers (FSRs)



## A Feedback Shift Register (FSR)

- consists of  $n$  register cells,  $x_i \in \{0, 1\}$
- in each clock
  - outputs  $x_0$
  - shifts the content one position to the right
  - updates  $x_{n-1}$  according to  $f(x_0, \dots, x_{n-1})$

$f$  linear  $\Rightarrow$  Call the register Linear Feedback Shift Register (LFSR).

$f$  nonlinear  $\Rightarrow$  Call it Nonlinear Feedback Shift Register (NFSR).

# Cryptanalysis of LFSR-based Keystream Generators

Keystream Generators with LFSR-based FSMs are widely used, but many have been broken using

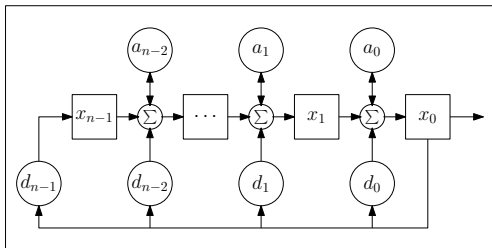
- Correlation Attacks
  - Compute the initial state(s) of the LFSR(s) based on correlations of their output streams with the keystream
- Algebraic Attacks
  - Set up a system of equations (depending on a piece of observed keystream) in the initial state bits of the LFSR(s) and solve it.

Idea: Try to replace the LFSRs by other devices that

- have similar pseudorandomness properties
- are (hopefully) more resistant to known attacks

One possibility: NFSRs with algebraic structure, e.g.,  
Feedback with Carry Shift Registers (FCSRs)

# FCSR in Galois Architecture

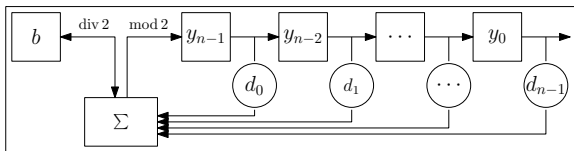


In each clock cycle,  $x_i := \sigma_i \bmod 2$   
 $a_i := \sigma_i \text{ div } 2$   
 with  $\sigma_i := x_{i+1} + (a_i + x_0)d_i$

Identify the state  $(x, a)$  with the integer

$$p := \left( \sum_{i=0}^{n-1} x_i 2^i \right) + 2 \sum_{i=0}^{n-2} a_i 2^i .$$

# FCSR in Fibonacci Architecture



In each clock cycle,  $y_{n-1} := \sigma \bmod 2$   
 $b := \sigma \operatorname{div} 2$   
 with  $\sigma := \sum_{i=0}^{n-1} y_i d_{n-i-1} + b$   
 $y_i := y_{i+1}$  for  $0 \leq i < n-1$

Identify the state  $(y, b)$  with the integer

$$p := b2^n - \sum_{k=0}^{n-1} \sum_{j=0}^k d_{j-1} y_{k-j} 2^k .$$

**Observation:** There is a one-to-one correspondence between periodic Galois and Fibonacci states.

## Algebra behind FCSRs: Ring of 2-adic Integers

2-adic Integer = formal power series  $\alpha = \sum_{i=0}^{\infty} u_i 2^i$

- Addition: addition mod 2 with carry
- Multiplication: multiplication mod 2 with carry
- Odd 2-adic integers have multiplicative inverse  $q^{-1}$ .

→ Ring contains especially rational numbers  $\frac{p}{q}$  with odd  $q$ .

### Observation

*The output of an FCSR with connection integer  $q = 1 - 2 \sum_{i=0}^{n-1} d_i$  and initial state corresponding to  $p$  is the 2-adic expansion of  $\frac{p}{q}$ .*

## Properties of FCSRs and their Output Sequences

For a periodic initial state corresponding to the integer  $p_0$ ,

- the state in time  $t$  corresponds to  $p_t = 2^{-t} p_0 \pmod q$
- the  $t$ -th output bit is  $z_t = p_t \pmod 2$

If the connection integer  $q$  is prime, 2 is a primitive root for  $q$ , and  $0 < p_0 < |q|$ , the FCSR produces so-called  $l$ -sequences with

- maximum period ( $= |q| - 1$ )
- 2-adic complexity ( $=$  size of the smallest FCSR generating the sequence) is low
- otherwise no biases from truly random sequences, especially
  - equally many zeros and ones
  - expected autocorrelation equal to zero
  - high linear complexity

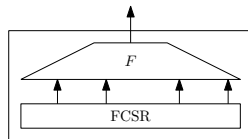
# The F-FCSR Stream Cipher Family (Arnault et al.)

... is the first real-world FCSR-based stream cipher.

The FSM of the F-FCSR instances consists of

- a Galois FCSR with  $n$  register cells
- a linear filter function

$$F : \{0, 1\}^n \rightarrow \{0, 1\}^k$$

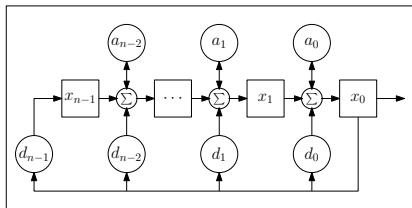


Proposed instances:

	$k$	$n$	keylength [bits]	IV length [bits]
F-FCSR-H	8	160	80	[32, 80]
F-FCSR-16	16	256	128	[0, 128]

We can assume the initial FCSR-state produced by the key/IV-setup to be periodic.

# Structure of the F-FCSR keystream

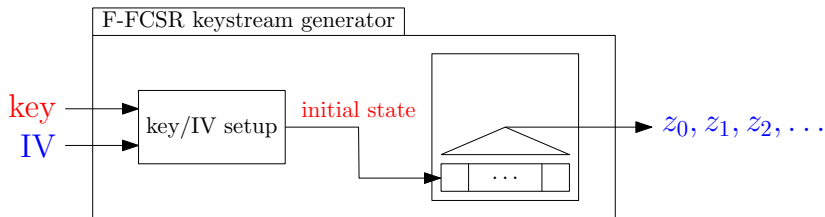


**Observation:**  $(x_{i,t})_{t \geq 0}$  in F-FCSR is given by the output sequence of a Galois FCSR with connection integer  $q$  and initial state  $p_{i,0} = p^{s_i} \bmod q$  for a constant  $s_i$ .

Implications:

- $(x_{i,t})_{t \geq 0}$  is a shifted version of  $(z_t)_{t \geq 0}$ .
- The keystream is obtained by bitwise XOR of several parts of the same  $l$ -sequence.  
→ We can view F-FCSR as a combination generator.
- The distances between the parts are publicly known, but large.

## Security of the F-FCSR Family



- F-FCSR has been designed to resist generic attacks such as time-memory-data tradeoffs, correlation attacks and algebraic attacks.
- Attacks on early versions of F-FCSR exploit weaknesses in the key/IV-setup to recover the secret key.
- No attacks on the FSM itself (i.e., with the key/IV-setup treated as a black box) that are more efficient than exhaustive search over all possible keys are publicly known yet.

# The End.

`dstegema@th.informatik.uni-mannheim.de`