

# Stream Ciphers

Gianluca Dini  
Dept. of Ingegneria dell'Informazione  
University of Pisa  
Email: [gianluca.dini@unipi.it](mailto:gianluca.dini@unipi.it)  
Last version: 2024-02-28

1

Stream Ciphers


# STREAM CIPHERS

Feb-24

Stream Ciphers

2

2

  
UNIVERSITÀ DI PISA

## Making OTP practical (1/3)

- Idea: replace the random key stream by a **pseudo-random** key stream
- Pseudo Random Generator  $G$**  is an efficient and deterministic function

$G: \{0,1\}^s \rightarrow \{0,1\}^n, n \gg s$

Seed space

Key-stream space


The key stream is computed from a seed

Feb-24

Stream Ciphers

3

3

  
UNIVERSITÀ DI PISA

## Making OTP practical (2/3)

Encryption:  $y = G(k) \oplus x$   
Decryption:  $x = G(k) \oplus y$

- Key  $k$  is a small secret (e.g., 100 bits)
- $G$  is pseudo-random so *sndr* & *rcvr* generate the same key stream

$k$

$G$

$G(k)$

$\oplus$

$x$

$-$

$y$

Feb-24

Stream Ciphers

4

4

Foundations of cybersecurity

2

## Making OTP practical (3/3)



UNIVERSITÀ DI PISA

- Is OTP-modified (stream cipher) still perfect?
  - NO!
    - $\#keys < \#msg \rightarrow$  Shannon's theorem is violated
  - We need a new definition of security!
- Security will depend on the specific PRG
  - PRG must **look random**, i.e., indistinguishable from a TRG **for a limited adversary**
    - It must be computationally unfeasible to distinguish PRNG output from a TRG output
  - **Computational security** (a new definition of security)

Feb-24

Stream Ciphers

5

5

## Computational security



UNIVERSITÀ DI PISA

- A new definition of security
- A cipher is **computationally (practically) secure** if the *perceived* level of computation required to defeat it, using the *best attack known*, exceeds, by a comfortable margin, the computation resources of the *hypothesized adversary*
- Now, the adversary is assumed to have a limited computation power

Feb-24


Stream Ciphers

6

6

# Computational security

- What is the *best known attack*?
- Even if a lower bound on the complexity of one attack is known, we don't know whether any other, more powerful attacks, are possible
- The best we can do is to design cryptosystem for which it is *assumed* that they are computationally secure



UNIVERSITÀ DI PISA

Feb-24

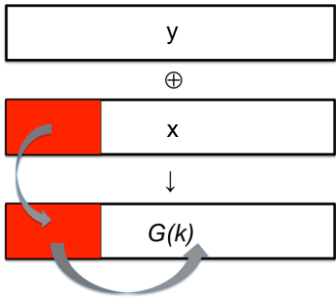
Stream Ciphers


8

8

# PRG must be unpredictable

- Not only a PRG must have good statistics, it must be also **unpredictable**
- If PRG is predictable, a stream cipher is not secure!
  - Assume an adversary can determine a prefix of  $x$  then
  - Then, (s)he can compute a prefix of the key stream
  - If  $G$  is predictable, (s)he can compute the rest of the key stream and thus decrypt  $y$






UNIVERSITÀ DI PISA

Feb-24

Stream Ciphers

9

9



UNIVERSITÀ DI PISA

## Unpredicatability


- Forward unpredicatability
  - If the seed is not known, the next output bit of a sequence must be unpredictable regardless of knowledge of any prefix of the sequence
- Backward unpredicatability
  - It must not be possible to determine the seed from the knowledge of any generated sequence
- If a sequence is/appears random it is not possible to predict either the next bit(s) or the seed

Feb-24

Stream Ciphers

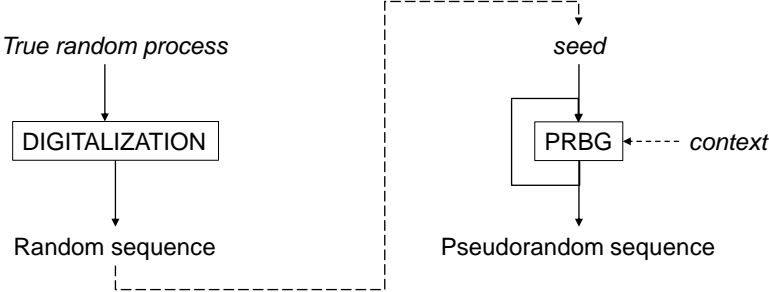
10

10



UNIVERSITÀ DI PISA

## TRBG and PRBG



```
graph TD; TRP[True random process] --> DIG[DIGITALIZATION]; DIG --> RS[Random sequence]; RS -.-> seed[seed]; seed --> PRBG[PRBG]; context[context] -.-> PRBG; PRBG --> PRS[Pseudorandom sequence]; PRS -.-> PRBG;
```

Feb-24

Stream Ciphers

11

11

Stream ciphers

STATE OF THE ART AND CASE STUDIES

Feb-24

Stream Ciphers

12

12

MS-PPTP (Windows NT)

$G(k) \oplus (m1 || m2 || m3 || \dots)$

$G(k) \oplus (s1 || s2 || s3 || \dots)$

Two-time pad!


UNIVERSITÀ DI PISA

Feb-24

Stream Ciphers

13

13

  
UNIVERSITÀ DI PISA

## MS-PPTP (Windows NT)


- The correct way to proceed is  $K = (K_{cs}, K_{sc})$
- $Z_{cs} = G(K_{cs})$ , key stream for encryption client  $\rightarrow$  server
- $Z_{sc} = G(K_{sc})$ , key stream for encryption server  $\rightarrow$  client

Feb-24

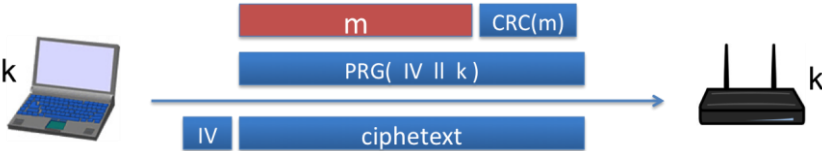
Stream Ciphers

14

14

  
UNIVERSITÀ DI PISA

## 802.11b WEP



- A new IV for each new message
  - Key is fixed (104-bits)
  - IV avoids 2TP
- Length of IV: 24 bits (in the standard!)
  - Repeated IV after  $2^{24} \approx 16M$  frames
  - On some 802.11 cards IV resets to 0 after power cycle

Feb-24

Stream Ciphers

15

15

# 802.11b WEP

Key for frame #1: 1||k  
Key for frame #2: 2||k  
Key for frame #3: 3||k  
...

- Related keys, not random
- FMS 2001 attack can recover  $K$  in  $10^6$  frames (now 40 Kframes)
- **Avoid related keys!**

Feb-24

Stream Ciphers

16

# 802.11b: WEP

Key for frame #1  
Key for frame #2  
Key for frame #3

- A better construction
- Each frame has its own key
- Keys are pseudo-random

Feb-24


Stream Ciphers

17



# RC4

- RC4 (1987)
  - Used in HTTPS and WEP
  - Variable seed; output: 1 byte
- Weaknesses
  - Bias
    - $\text{Pr}[\text{2nd byte} = 0] = 2/256$  (twice as random)
      - Other bytes are biased too (e.g., 1st,3rd)
      - It is recommended that the first 256 bytes are ignored
    - $\text{Pr}[00] = 1/256^2 + 1/256^3$ 
      - Bias starts after several gigabytes but it is still a distinguisher
  - Related keys
- It is recommended not to use RC4 but modern CSPRNG



UNIVERSITÀ DI PISA

Feb-24

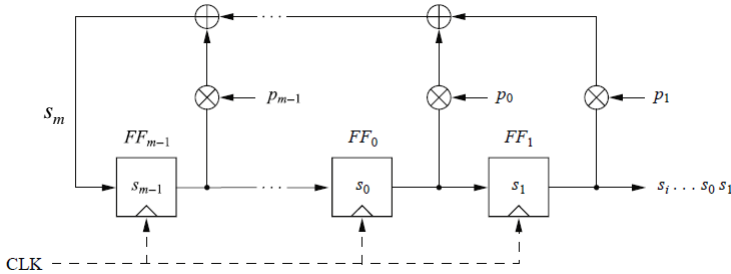
Stream Ciphers

18

18

# Linear Feedback Shift Register

- $p_i$  = feedback coefficient (If  $p_i == 1$ , the feedback is active; otherwise it is not)




$$s_m \equiv p_{m-1}s_{m-1} + \dots + p_1s_1 + p_0s_0 \text{ mod } 2$$
$$s_{m+1} \equiv p_{m-1}s_m + \dots + p_1s_2 + p_0s_1 \text{ mod } 2$$
$$s_{i+m} \equiv \sum_{j=0}^{m-1} p_j \cdot s_{i+j} \text{ mod } 2, s_i, p_j \in \{0,1\}, i = 0, 1, 2, \dots$$

Feb-24

Stream Ciphers

19

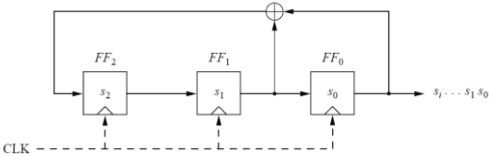
19

  
UNIVERSITÀ DI PISA

# LFSR is periodical

- LFSR
  - Degree: 3
- Sequence of states


clk	$FF_2$	$FF_1$	$FF_0 = s_i$
0	1	0	0
1	0	1	0
2	1	0	1
3	1	1	0
4	1	1	1
5	0	1	1
6	0	0	1
7	1	0	0
8	0	1	0



← The initial state (*seed*)

← The sequence of states is *periodical*

20

  
UNIVERSITÀ DI PISA

# LFSR - Properties

- Properties
  - Seed = initial state of the register
    - All 0's state must be avoided
  - Degree = number of storage units
    - Degree = 8
  - Periodic
- Maximum-length LSFR
  - Theorem
    - The maximum sequence length generated by an LFSR of degree  $m$  is  $2^m - 1$
  - Maximum-length LSFR can be easily found

21

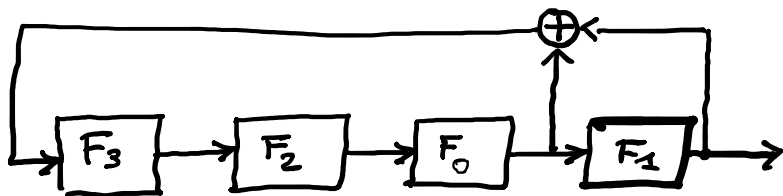
Foundations of cybersecurity

10

# LFSR – example #1



- LFSR with maximum output sequence
  - Degree  $m = 4$
  - Coefficients:  $p_3 = 0, p_2 = 0, p_1 = 1, p_0 = 0$
  - Period  $= 2^m - 1 = 15$

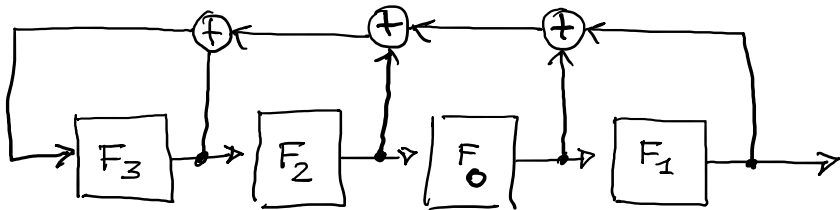


22

# LFSR – example #2



- LFSR with non-maximum output sequence
  - Degree  $m = 4$
  - Coefficients:  $p_3 = 1, p_2 = 1, p_1 = 1, p_0 = 1$
  - Period  $= 5$



23

## LFSRs are not good for crypto



UNIVERSITÀ DI PISA

- Pros:
  - LFSRs have good statistical properties
- Cons
  - Periodical
  - Linear

Feb-24

Stream Ciphers

24

24

## LFSRs are not good for crypto



UNIVERSITÀ DI PISA

- Known-Plaintext attack against LFSR
  1. Given  $2m$  pairs (pt, ct), the adversary determines a prefix of the sequence  $s_i$
  2. Then, the adversary determines *feedback coefficients* by solving a system of  $m$  linear equations in  $m$  unknowns
  3. Finally, the adversary can “build” the LFSR and produce the entire sequence

Feb-24

Stream Ciphers

25

25

## LSFRs are not good for crypto



UNIVERSITÀ DI PISA

- Have LSFRs to be thrown away?
  - Use a **non-linear combination** of several LFSRs to build strong cryptosystems
    - E.g., use AND
  - E.g.: Trivium (2003)

Feb-24

Stream Ciphers

26

26

## State of the art



UNIVERSITÀ DI PISA

- Software-oriented
  - RC4 and SEAL
    - Very well-investigated; secure
- Hardware-oriented
  - LFSR-based
    - Many have been broken
  - GSM A5/1 and A5/2
    - A5/1 used to be secret but was reverse-engineered
    - A5/2 has serious flaws
    - Neither of them is recommended nowadays
    - A5/3 (KASUMI) is used but it is a block cipher


Feb-24

Stream Ciphers

27

27

# State of the art



UNIVERSITÀ DI PISA

- eSTREAM Project
  - ECRYPT Network of Excellence
    - Call for stream ciphers; 34 candidates
  - Profile 1. Stream ciphers for software applications with high throughput requirements
    - HC-128, Rabbit, Salsa20/12, SOSEMANUK
  - Profile 2. Stream ciphers for hardware applications with restricted resources
    - Grain v1, MICKEY v2, Trivium


Feb-24

Stream Ciphers

28

28

# eSTREAM performance



UNIVERSITÀ DI PISA

- RC4                    126 Mb/s (\*)
- Salsa 20/12        643 Mb/s
- Sosemanuk        727 Mb/s
- (\*) AMD Opteron 2.2. GHz (Linux)

Feb-24

Stream Ciphers

29

29

Stream Ciphers

CONTENT SCRAMBLING SYSTEM  
(CSS)


Feb-24

Stream Ciphers

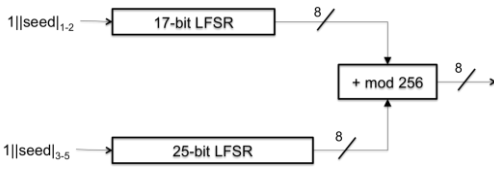
30

30

Content Scrambling System

  
UNIVERSITÀ DI PISA

- Seed (key)
  - initial states of the LFSRs 5 bytes (80 bit)
- Each round
  - 8 CLK cycles
  - Each LFSR produces 8 bits
  - LFSR's outputs are added mod 256<sup>(\*)</sup> so producing the key stream
    - <sup>(\*)</sup> neglect carry bit for simplicity



Feb-24

Stream Ciphers

31

31

## Content Scrambling System



UNIVERSITÀ DI PISA

- Easy to break in  $2^{17}$  steps ( $\ll 2^{40}$ )
- Known-plaintext attack
  - A prefix $|_{1-20}$  of the (cleartext) movie is known  $\Rightarrow$  a prefix of the keystream $|_{1-20}$  can be computed
    - E.g., 20 initial bytes in mpeg
- For details
  - <https://www.cs.cmu.edu/~dst/DeCSS/Kesden/>

Feb-24

Stream Ciphers

32

32

## Content Scrambling System



UNIVERSITÀ DI PISA

- Attack algorithm
  - For all possible initial setting of LFSR-17 ( $2^{17}$ )
    1. Run LFSR-17 to get 20 bytes of output
    2. Subtract LFSR-17 $|_{1-20}$  from keystream $|_{1-20}$  and obtain a candidate output of LFSR-25 $|_{1-20}$
    3. Check whether LFSR-25 $|_{1-20}$  is consistent with LFSR-25
      - a. If it is consistent then we have found correct initial setting of both and the algorithm is finished!
      - b. Otherwise, go to 1 and test the next LFSR-17 initial setting
  - Using key, generate entire CSS output
  - Complexity
    - At most, the attack need to try all the possible initial setting of LFSR-17 ( $2^{17}$ )

Feb-24

Stream Ciphers

33

33