

Stream Ciphers

Gianluca Dini

Dept. of Ingegneria dell'Informazione

University of Pisa

Email: gianluca.dini@unipi.it

Last version: 2021-03-02

1

Stream Ciphers

STREAM CIPHERS

March 22

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

March 22

Stream Ciphers

3

3



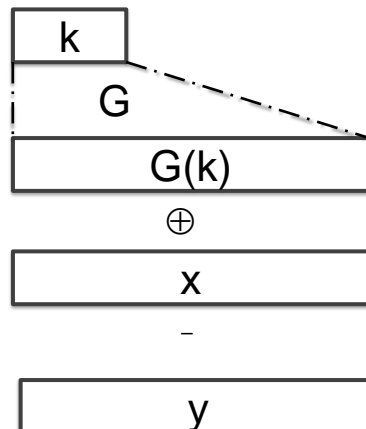
UNIVERSITÀ DI PISA

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



March 22

Stream Ciphers

4

4

Making OTP practical (3/3)



UNIVERSITÀ DI PISA

- Is OTP-modified (stream cipher) still perfect?
 - NO! $\#keys < \#msg \Rightarrow$ Shannon's theorem 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
 - A new definition of security is necessary: **computational security**

March 22

Stream Ciphers

5

5

Computational security



UNIVERSITÀ DI PISA

- Definition
 - A cryptosystem is computationally secure if the **best known algorithm** for breaking it requires at least t operations
 - Cons
 - What is the best known attack?
 - The best we can do it to design cryptosystem for which it is *assumed* that they are computationally secure

March 22

Stream Ciphers

6

6

Computational security



UNIVERSITÀ DI PISA

- Cons
 - A. What is the best known attack?
 - B. 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 it to design cryptosystem for which it is *assumed* that they are computationally secure

March 22

Stream Ciphers

7

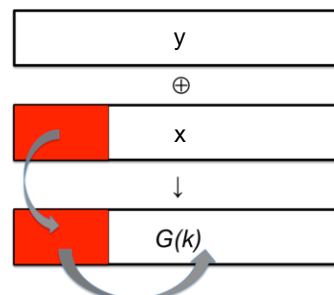
7

Why we need predictability



UNIVERSITÀ DI PISA

- If PRG is predictable, a stream cipher is not secure!
 - Assume an adversary is able to 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



March 22

Stream Ciphers

8

8

Stream ciphers

STATE OF THE ART AND CASE STUDIES


March 22

Stream Ciphers


9

9

MS-PPTP (Windows NT)


UNIVERSITÀ DI PISA

K



m1

m2


m3

s1

s2

s3

K



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

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

Two-time pad!

March 22

Stream Ciphers

10

10

MS-PPTP (Windows NT)



UNIVERSITÀ DI PISA

- 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

March 22

Stream Ciphers

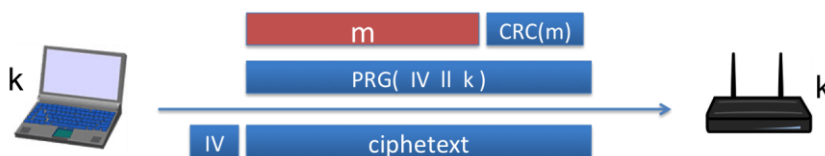
11

11

802.11b WEP



UNIVERSITÀ DI PISA



- 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

March 22

Stream Ciphers

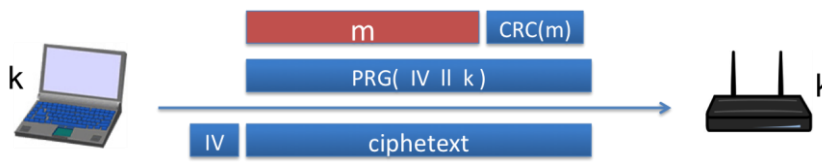
12

12

802.11b WEP



UNIVERSITÀ DI PISA



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!**

March 22

Stream Ciphers

13

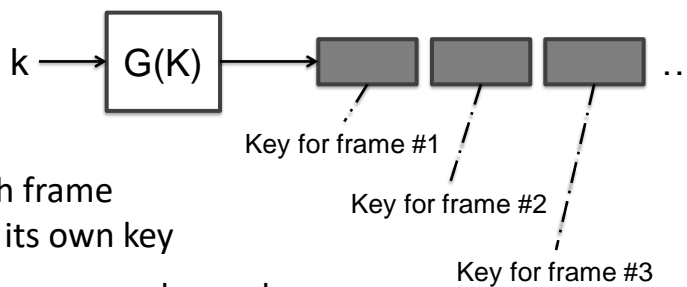
13

802.11b: WEP



UNIVERSITÀ DI PISA

- A better construction



- Each frame has its own key
- Keys are pseudo-random

March 22

Stream Ciphers

14

14



UNIVERSITÀ DI PISA

RC4

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

March 22

Stream Ciphers

15

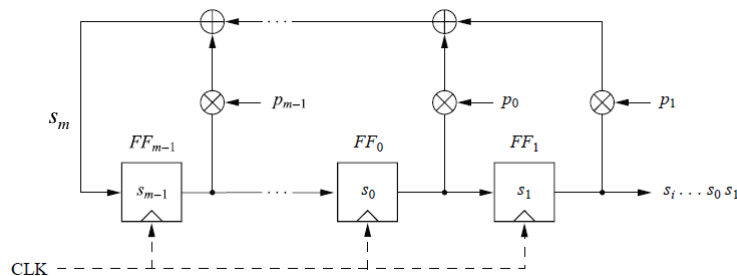
15



UNIVERSITÀ DI PISA

Linear Feedback Shift Register

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



$$s_m \equiv p_{m-1}s_{m-1} + \dots + p_1s_1 + p_0s_0 \pmod{2}$$

$$s_{m+1} \equiv p_{m-1}s_m + \dots + p_1s_2 + p_0s_1 \pmod{2}$$

$$s_{i+m} \equiv \sum_{j=0}^{m-1} p_j \cdot s_{i+j} \pmod{2}, s_i, p_j \in \{0,1\}, i = 0, 1, 2, \dots$$

March 22

Stream Ciphers

16

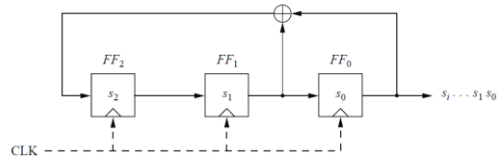
16

LFSR is periodical



UNIVERSITÀ DI PISA

- 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*

March 22

Stream Ciphers

17

17

LFSR - Properties



UNIVERSITÀ DI PISA

- 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

March 22

Stream Ciphers

18

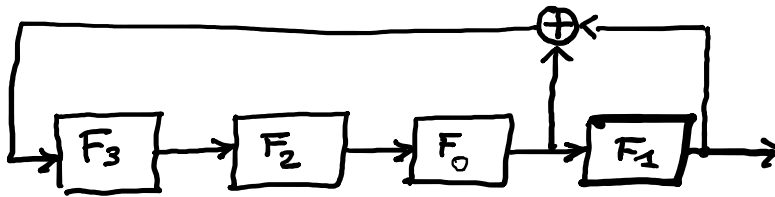
18

LFSR – example #1



UNIVERSITÀ DI PISA

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



March 22

Stream Ciphers

19

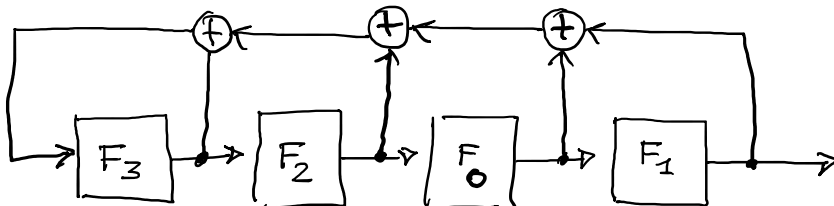
19

LFSR – example #2



UNIVERSITÀ DI PISA

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



March 22

Stream Ciphers

20

20

LFSRs are not good for crypto



UNIVERSITÀ DI PISA

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

March 22

Stream Ciphers

21

21

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

March 22

Stream Ciphers

22

22

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)

March 22

Stream Ciphers

23

23

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

March 22

Stream Ciphers

24

24

State of the art



UNIVERSITÀ DI PISA

- eSTREAM Project
 - ECRYPT NoE
 - 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

March 22

Stream Ciphers

25

25

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)

March 22

Stream Ciphers

26

26

Stream Ciphers

CONTENT SCRAMBLING SYSTEM (CSS)

March 22

Stream Ciphers

27

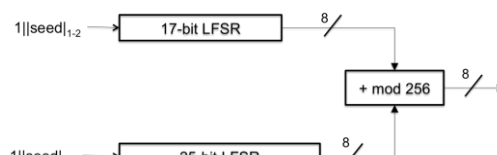
27

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^(*) so producing the key stream
 - ^(*) neglect carry bit for simplicity



March 22

Stream Ciphers

28

28

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/>

March 22

Stream Ciphers

29

29

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 LSFR-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})

March 22

Stream Ciphers

30

30