

# Modern Symmetric Cryptography



© André Zúquete /  
Tomás Oliveira e Silva

Applied Cryptography

1

## Modern ciphers: types

- ▷ Concerning operation
  - ♦ Block ciphers (mono-alphabetic)
  - ♦ Stream ciphers (poly-alphabetic)
- ▷ Concerning their key
  - ♦ Symmetric ciphers (secret key or shared key ciphers)
  - ♦ Asymmetric ciphers (or public key ciphers)
- ▷ Arrangements

	Block ciphers	Stream ciphers
Symmetric ciphers		
Asymmetric ciphers		



© André Zúquete /  
Tomás Oliveira e Silva

Applied Cryptography

2

# Symmetric ciphers

- ▷ Secret key
  - Shared by 2 or more peers
- ▷ Allow
  - Confidentiality among the key holders
  - Limited authentication of messages
    - When block ciphers are used
- ▷ Advantages
  - Performance (usually very efficient)
- ▷ Disadvantages
  - $N$  interacting peers, pairwise secrecy  $\Rightarrow N \times (N-1)/2$  keys
- ▷ Problems
  - Key distribution



# Symmetric block ciphers

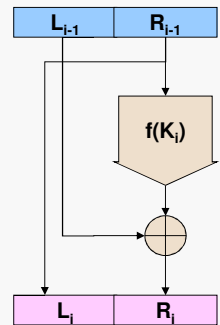
- ▷ Usual approaches
  - Large bit blocks for input, output and key
    - 64, 128, 256, etc.
  - Diffusion & confusion
    - Permutation, substitution, expansion, compression
    - Feistel networks, substitution-permutation networks
    - Iterations
    - Sub-keys (key schedules, round keys, etc.)
- ▷ Most common algorithms
  - DES (Data Enc. Stand.), D=64 K=56
  - IDEA (Int. Data Enc. Alg.), D=64 K=128
  - AES (Adv. Enc. Stand., aka Rijndael) D=128 K=128, 192, 256
  - Other (Blowfish, CAST, RC5, etc.)



## Feistel networks

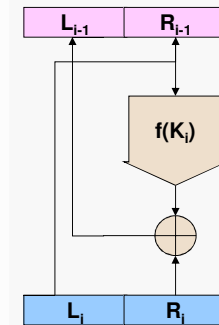
$$L_i = R_{i-1}$$

$$R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$$



$$R_{i-1} = L_i$$

$$L_{i-1} = R_i \oplus f(L_i, K_i)$$



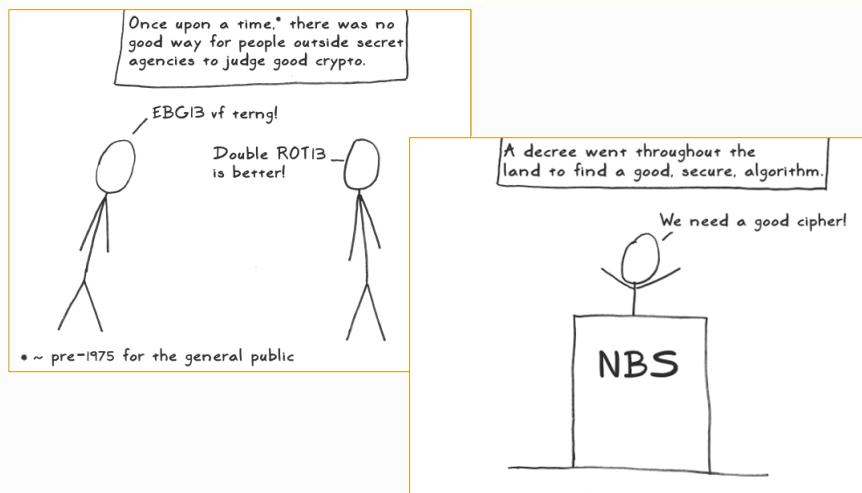
© André Zúquete /  
Tomás Oliveira e Silva

Applied Cryptography

5

## We need a good cipher!

<http://www.moserware.com/2009/09/stick-figure-guide-to-advanced.html>



© André Zúquete /  
Tomás Oliveira e Silva

Applied Cryptography

6

# DES (Data Encryption Standard)

- ▷ 1970: the need of a standard cipher for civilians was identified
- ▷ 1972: NBS opens a contest for a new cipher, requiring:
  - The cryptographic algorithm must be secure to a high degree
  - Algorithm details described in an easy-to-understand language
  - The details of the algorithm must be publicly available
    - So that anyone could implement it in software or hardware
  - The security of the algorithm must depend on the key
    - Not on keeping the method itself (or part of it) secret
  - The method must be adaptable for use in many applications
  - Hardware implementations of the algorithm must be practical
    - i.e. not prohibitively expensive or extremely slow
  - The method must be efficient
  - Test and validation under real-life conditions
  - The algorithm should be exportable



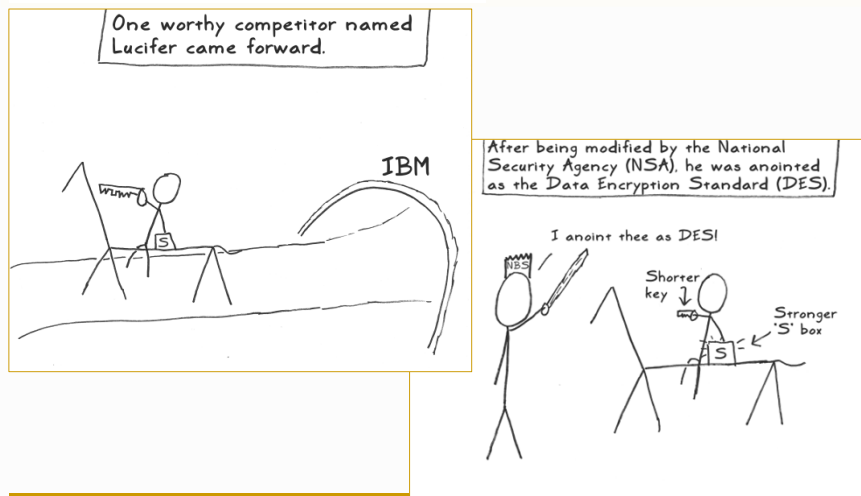
© André Zúquete /  
Tomás Oliveira e Silva

Applied Cryptography

7

# Lucifer and DES

<http://www.moserware.com/2009/09/stick-figure-guide-to-advanced.html>



© André Zúquete /  
Tomás Oliveira e Silva

Applied Cryptography

8

## DES: proposal and adoption

- ▷ 1974: new contest
  - Proposal based on Lucifer from IBM
  - 64-bit blocks
  - 56-bit keys
    - 48-bit subkeys (key schedules)
  - Diffusion & confusion
    - Feistel networks
    - Permutations, substitutions, expansions, compressions
    - 16 iterations
  - Several modes of operation
    - **ECB** (Electronic Code Book), **CBC** (Cypher Block Chaining)
    - **OFB** (Output Feedback), **CFB** (Cypher Feedback)
- ▷ 1976: adopted at US as a federal standard

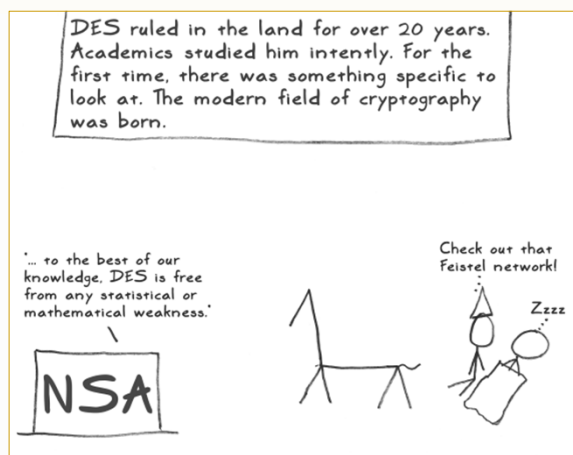


© André Zúquete /  
Tomás Oliveira e Silva

Applied Cryptography

9

## DES as a milestone

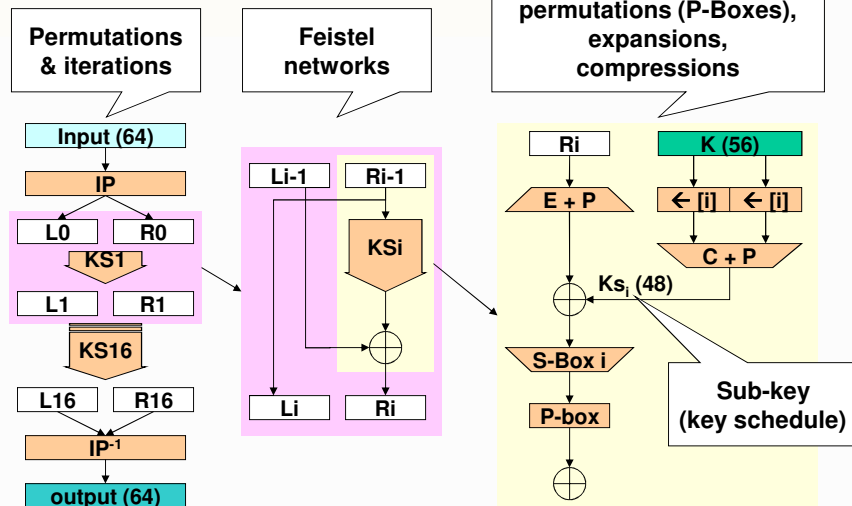


© André Zúquete /  
Tomás Oliveira e Silva

Applied Cryptography

10

## DES: architecture



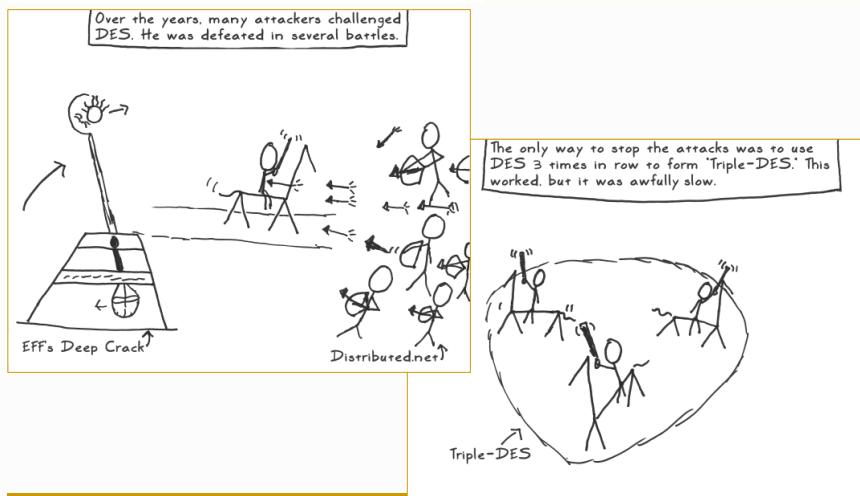
© André Zúquete /  
Tomás Oliveira e Silva

Applied Cryptography

11

## DES security

<http://www.moserware.com/2009/09/stick-figure-guide-to-advanced.html>



© André Zúquete /  
Tomás Oliveira e Silva

Applied Cryptography

12

## DES: offered security

### ▷ Key selection

- Most 56-bit values are suitable
- 4 weak, 12 semi-weak keys, 48 possibly weak keys
  - Equal key schedules (1, 2 or 4)
  - Easy to spot and avoid

### ▷ Key length

- 56 bits are actually too few
- Exhaustive search is technically possible and economically interesting

### ▷ Known attacks

- Exhaustive key space search

### ▷ Multiple encryption

- Double encryption
  - Theoretically not more secure
- Triple DES (3DES)
  - With 2 or 3 keys
  - Equivalent key length of 112 or 168 bits
  - Slow ...but secure!



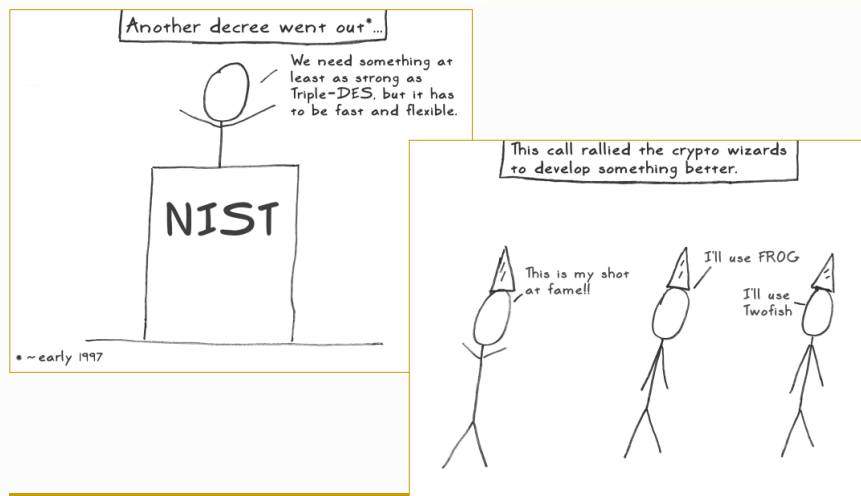
© André Zúquete /  
Tomás Oliveira e Silva

Applied Cryptography

13

## Replacement of DES (and DES variants)

<http://www.moserware.com/2009/09/stick-figure-guide-to-advanced.html>



© André Zúquete /  
Tomás Oliveira e Silva

Applied Cryptography

14

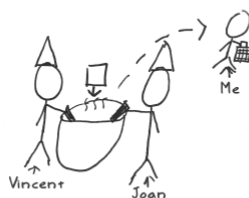
# AES (Advanced Encryption Standard)

- ▷ 2/Jan/1997: Call for evaluation criteria
  - NIST publicly asked interested parties to propose a criteria to choose a DES successor
  - Many submissions received during 3 months
- ▷ 12/Sep/1997: Call for new algorithms
  - Block ciphers
  - 128-bit blocks
  - 128, 192, and 256-bit keys
  - Such ciphers were rare at the time of the call



# Rijndael

My creators, Vincent Rijmen and Joan Daemen, were among these crypto wizards. They combined their last names to give me my birth name: Rijndael.\*



\* That's pronounced 'Rhine Dahl' for the non-Belgians out there.





## AES: evaluation rounds

### ▷ 1<sup>st</sup> round

- 15 candidate algorithms were evaluated by the community
- Conferences were organized for the evaluation
- Cryptographic weakness were found
- Performance issues were identified
  - In a variety of hardware
  - PCs, smart cards, hardware implementations
- Constrained environment were evaluated
  - Limited memory smart cards, low gate count circuits, FPGAs

### ▷ Aug/1999: AES finalists announced

- MARS, RC6, Rijndael, Serpent, and Twofish



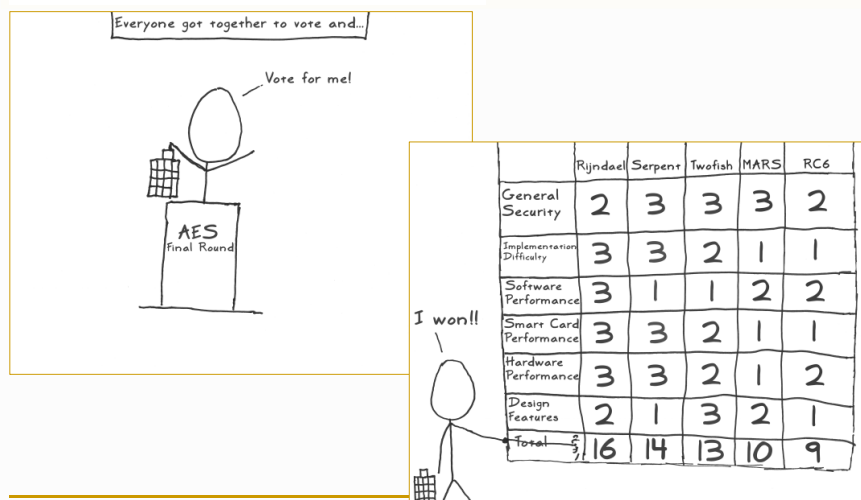
© André Zúquete /  
Tomás Oliveira e Silva

Applied Cryptography

17

## Rijndael selection as AES

<http://www.moserware.com/2009/09/stick-figure-guide-to-advanced.html>



© André Zúquete /  
Tomás Oliveira e Silva

Applied Cryptography

18

## AES: evaluation rounds

- ▷ 2<sup>nd</sup> round
  - The 5 finalists continued to be evaluated
  - In a final conference the proposal of each algorithm presented their advantage against the other
- ▷ 2/Oct/2000: AES algorithm was announced
  - Rijndael was selected
  - Proposed by Vincent Rijmen and Joan Daemen
  - Family of ciphers with different key and block sizes
- ▷ 26/Nov/2001: AES was approved by NIST
  - FIPS PUB 197
  - Subset of Rijndael (3 family members)
- ▷ Now part of the ISO/IEC 18033-3 standard

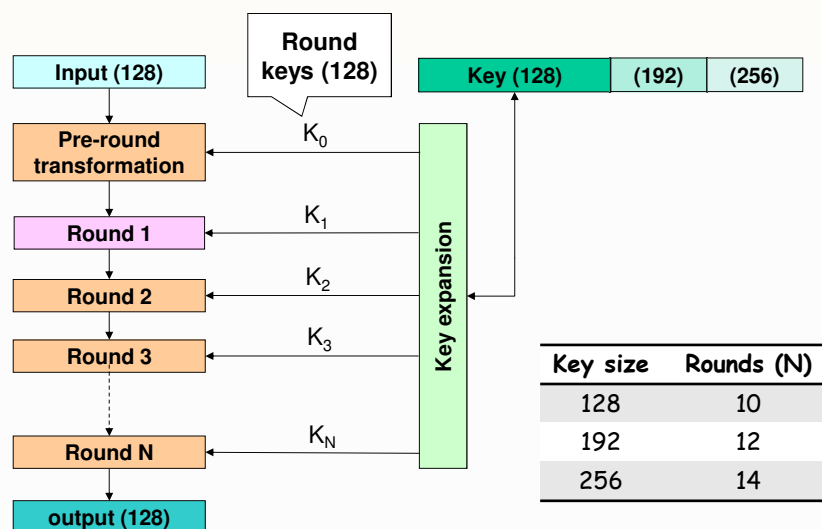


© André Zúquete /  
Tomás Oliveira e Silva

Applied Cryptography

19

## AES: architecture

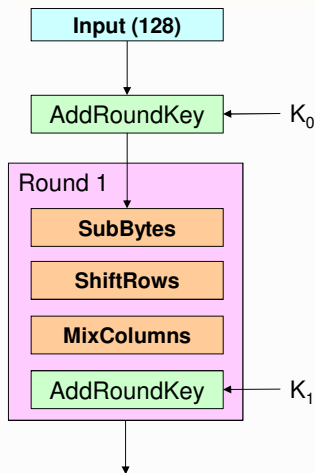


© André Zúquete /  
Tomás Oliveira e Silva

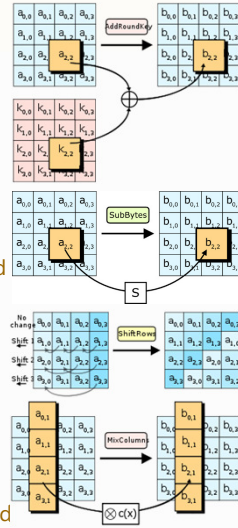
Applied Cryptography

20

# AES: architecture



- ▶ **AddRoundKey:**
  - 128-bit XOR
  - Output is a 4x4 byte matrix
- ▶ **SubBytes:**
  - 256-element S-box
  - Each matrix bytes is substituted
- ▶ **ShiftRows**
  - Rows are rotated left
  - Byte shifts vary (0, 1, 2 & 3)
- ▶ **MixColumns**
  - Each column is transformed
  - Not performed in the last round



© André Zúquete /  
Tomás Oliveira e Silva

Applied Cryptography

<https://aescryptography.blogspot.com>

21

# AES complexity and speed-up

<http://www.moscowware.com/2009/09/stick-figure-guide-to-advanced.html>

Mix Columns is the hardest. I treat each column as a polynomial. I then use our new multiply method to multiply it by a specially crafted polynomial and then take the remainder after dividing by  $x^4+1$ . This all simplifies to a matrix multiply:

$$b(x) = c(x) \cdot a(x) \mod x^4+1$$

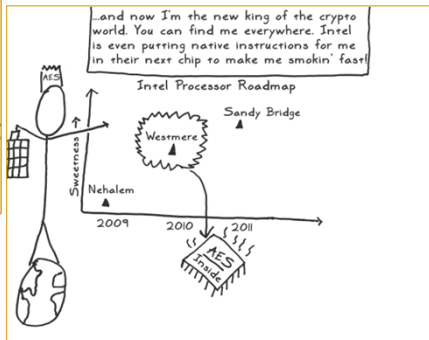
$$= (03x^3+01x^2+01x+02) \cdot (a_3x^3+a_2x^2+a_1x+a_0) \mod x^4+1$$

$$= 03a_3x^6 + 03a_2x^5 + 03a_1x^4 + 03a_0x^3 + 01a_3x^5 + 01a_2x^4 + 01a_1x^3 + 01a_0x^2 + 02a_3x^4 + 02a_2x^3 + 02a_1x^2 + 02a_0x \mod x^4+1$$

$$= 03a_3x^2 + 03a_2x + 03a_1 + 03a_0x^3 + 01a_3x + 01a_2 + 01a_1x^3 + 01a_0x^2 + 02a_3x + 02a_2 + 02a_1x^3 + 02a_0x^2 \mod x^4+1$$

$$= 03a_3x^2 + 03a_2x + 03a_1 + 03a_0x^3 + 01a_3x + 01a_2 + 01a_1x^3 + 01a_0x^2 + 02a_3x + 02a_2 + 02a_1x^3 + 02a_0x^2 \mod x^4+1$$

$$= (03a_3+01a_2+02a_1+02a_0)x^3 + (03a_2+01a_1+02a_0)x^2 + (03a_1+01a_0)x + (03a_0) \mod x^4+1$$

$$= \begin{bmatrix} 03 & 01 & 02 & 02 \\ 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 03 \\ 02 & 01 & 01 & 03 \end{bmatrix} \begin{bmatrix} a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} b_3 \\ b_2 \\ b_1 \\ b_0 \end{bmatrix}$$


© André Zúquete /  
Tomás Oliveira e Silva

Applied Cryptography

22

## AES in CPU instruction's set

### ▷ Intel AES New Instructions (AES-NI)

AESENC	Perform one round of an AES encryption flow
AESENCLAST	Perform the last round of an AES encryption flow
AESDEC	Perform one round of an AES decryption flow
AESDECLAST	Perform the last round of an AES decryption flow
AESKEYGENASSIST	Assist in AES round key generation
AESIMC	Assist in AES Inverse Mix Columns

### ▷ ARMv8 Cryptographic Extension

### ▷ ... and other



© André Zúquete /  
Tomás Oliveira e Silva

Applied Cryptography

23

## Stream ciphers

### ▷ Approaches

- ♦ Cryptographically secure pseudo-random generators (PRNG)
  - Using linear feedback shift registers (LFSR)
  - Using block ciphers
  - Other (families of functions, etc.)
- ♦ Usually not self-synchronized
- ♦ Usually without uniform random access
  - No immediate setup of generator's state for a given plaintext/ciphertext offset

### ▷ Most common algorithms

- ♦ A5/1 (US, Europe), A5/2 (GSM)
- ♦ RC4 (802.11 WEP/TKIP, etc.)
- ♦ E0 (Bluetooth BR/EDR)
- ♦ SEAL (w/ uniform random access)



© André Zúquete /  
Tomás Oliveira e Silva

Applied Cryptography

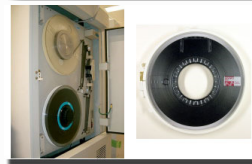
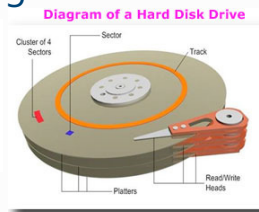
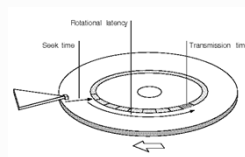
24

## Uniform random access

- ▷ Same time to reach and process any piece of information regardless of its storage location

- ▷ Uniform

- Memory
- Disks (magnetic, optical)
  - Average  $T_{\text{access}} = T_{\text{seek}} + \frac{1}{2} T_{\text{revolution}}$



- ▷ Non-uniform

- Tapes (audio, video, computer)



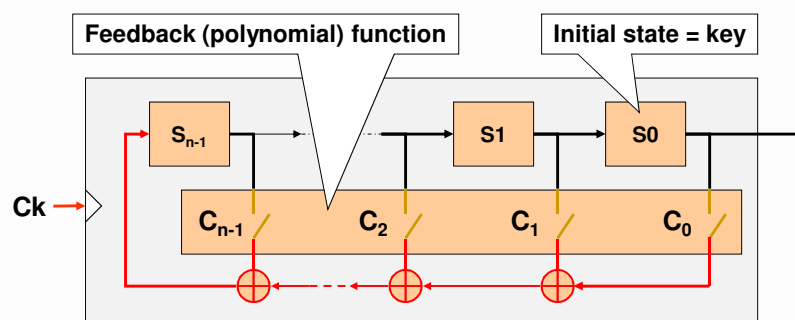
© André Zúquete /  
Tomás Oliveira e Silva

Applied Cryptography

[http://csr507doc.sco.com/en/PERFORM/disk\\_IO\\_mech.html](http://csr507doc.sco.com/en/PERFORM/disk_IO_mech.html)  
<https://www.ict4u.net/components/backing-storage.php>

25

## Linear Feedback Shift Register (LFSR)



- ▷  $2^n - 1$  non-null sequences
  - If one of them has a  $2^n - 1$  period length, then all have it
- ▷ Primitive feedback functions (primitive polynomials)
  - All non-null sequences have a  $2^n - 1$  period length

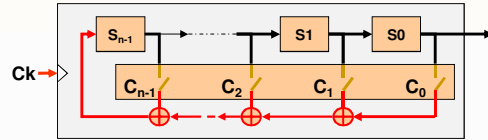


© André Zúquete /  
Tomás Oliveira e Silva

Applied Cryptography

26

## Linear Feedback Shift Register (LFSR)



### Issue

- ♦ If you know N consecutive bits of the output, you know the entire sequence ahead

$$O_0 \quad O_1 \quad O_2 \quad O_3 \quad \dots \quad O_n \quad O_n = C_0 O_0 + C_1 O_1 + \dots + C_{n-1} O_{n-1}$$

- ♦ The output must be mixed with something else ...

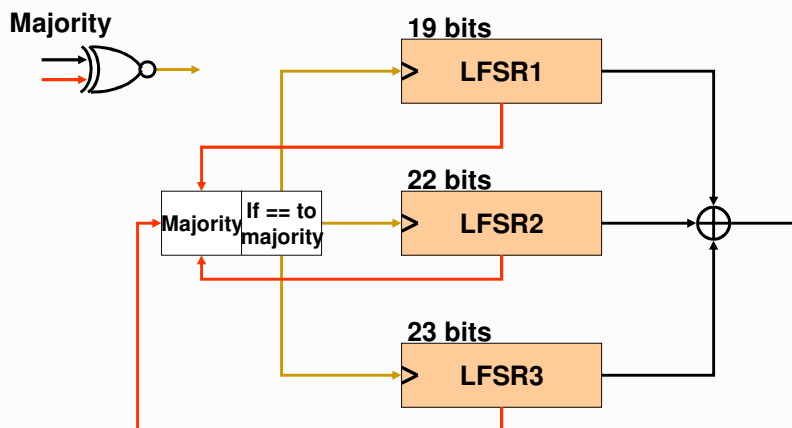


© André Zúquete /  
Tomás Oliveira e Silva

Applied Cryptography

27

## Generators using many LFSR: A5/1 (GSM)



© André Zúquete /  
Tomás Oliveira e Silva

Applied Cryptography

28