# Modern Symmetric Cryptography



# Modern ciphers: types

- ▷ Concerning operation
  - Block ciphers (mono-alphabetic)
  - Stream ciphers (poli-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 | | |

# 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 ⇒ N x (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

Encrypt

$L_i = R_{i-1}$

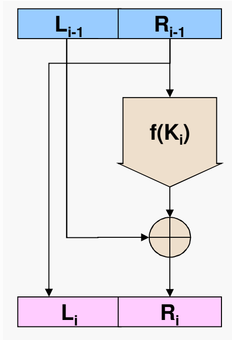
$R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$

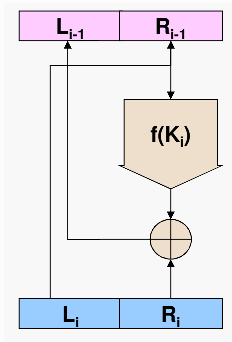
Decrypt

$R_{i-1} = L_i$

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

reverse

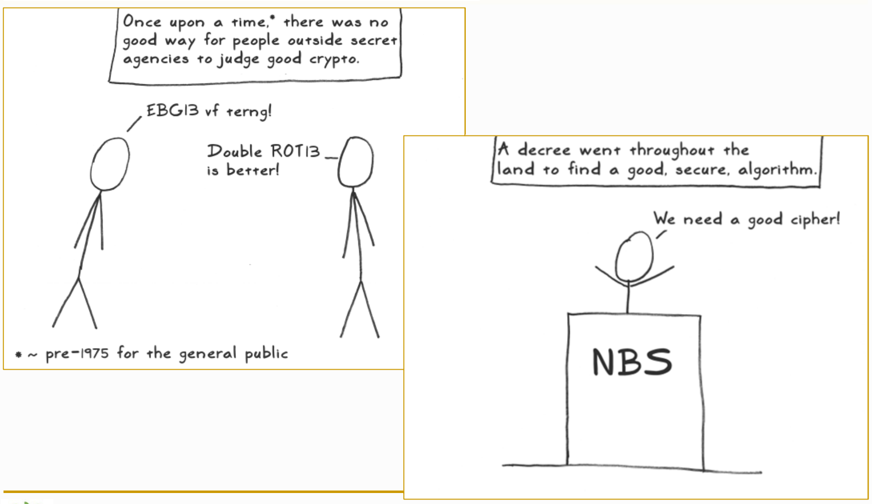





# We need a good cipher!

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



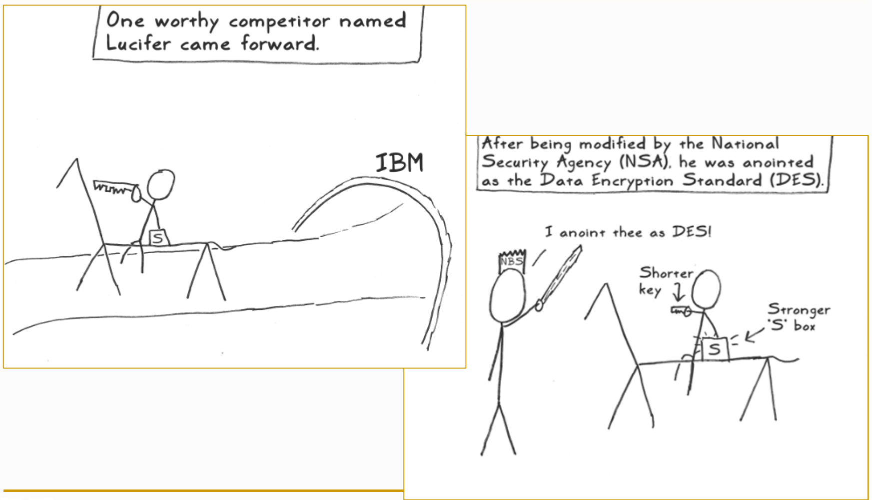
# DES (Data Encryption Standard)

- 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



# Lucipher and DES

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

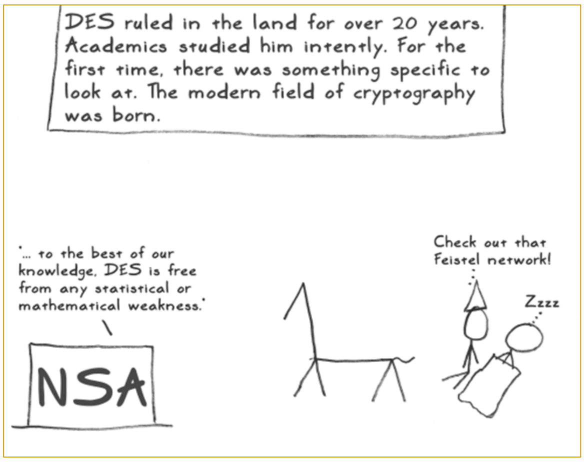


# DES: proposal and adoption

- 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

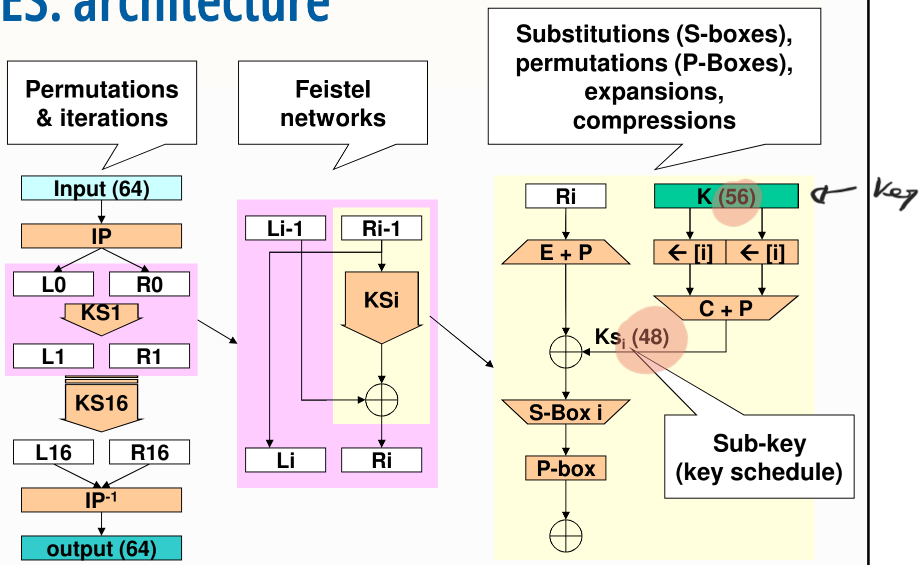


# DES as a milestone




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

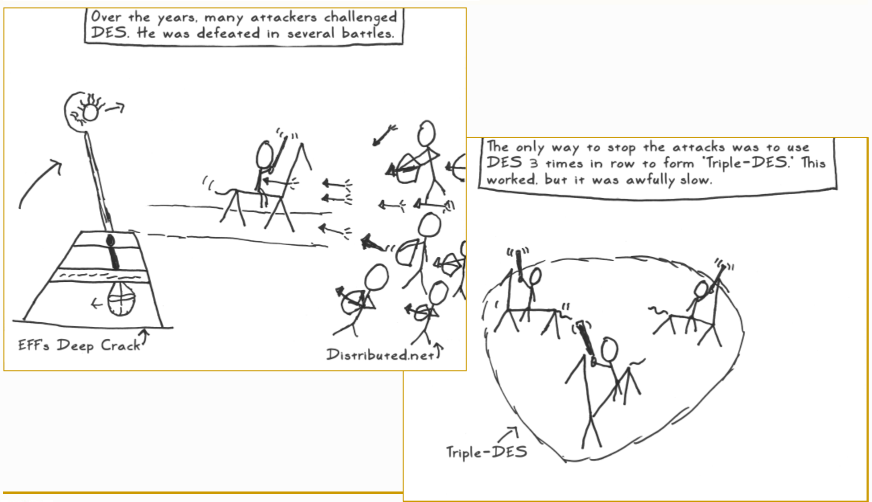
# DES: architecture






# DES security

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

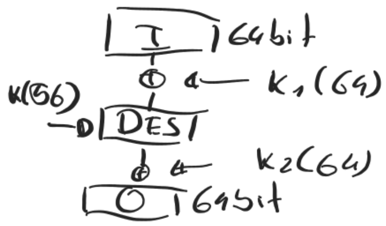


DES Issues → Slow on software
→ short key

# DES: offered security

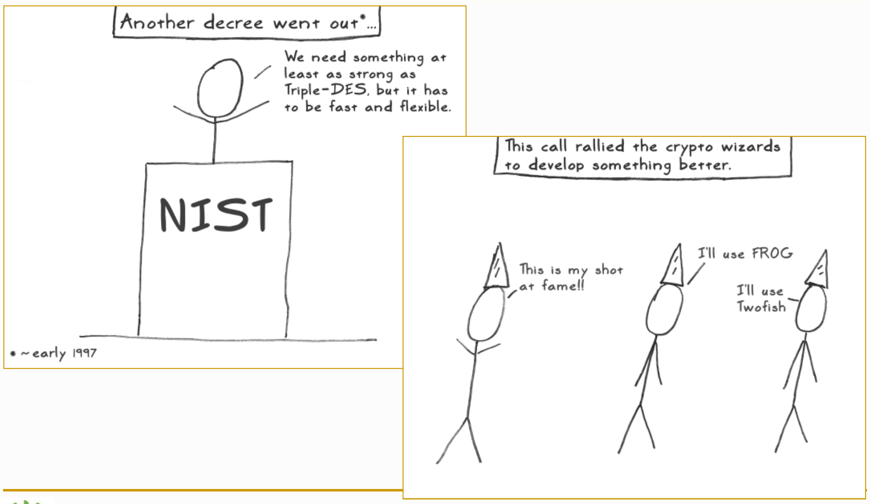
- Key selection
  - Most 56-bit values are suitable
  - 4 weak, 12 semi-weak keys, 48 possibly weak keys
    - Equal key schedules (1, 2 or 4)
    - Easy to spot and avoid
- Known attacks
  - Exhaustive key space search
  ↳ Brute Force
- Key length
  - 56 bits are actually too few
  - Exhaustive search is technically possible and economically interesting
- Multiple encryption
  - Double encryption
    - Theoretically not more secure
  - Triple DES (3DES)
    - With 2 or 3 keys
    - Equivalent key length of 112 or 168 bits
    - Secure but …slow!
  - DES-X
    - $K_1 \oplus DES(K_2) \oplus K_3$
    - Total key length = 64 + 56 + 64 = 184 bits






# Replacement of DES (and DES variants)

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

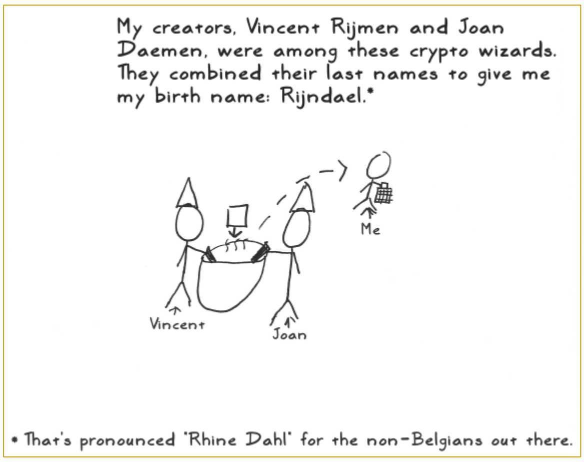


# AES (Advanced Encryption Standard)

- 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




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

# AES: evaluation rounds

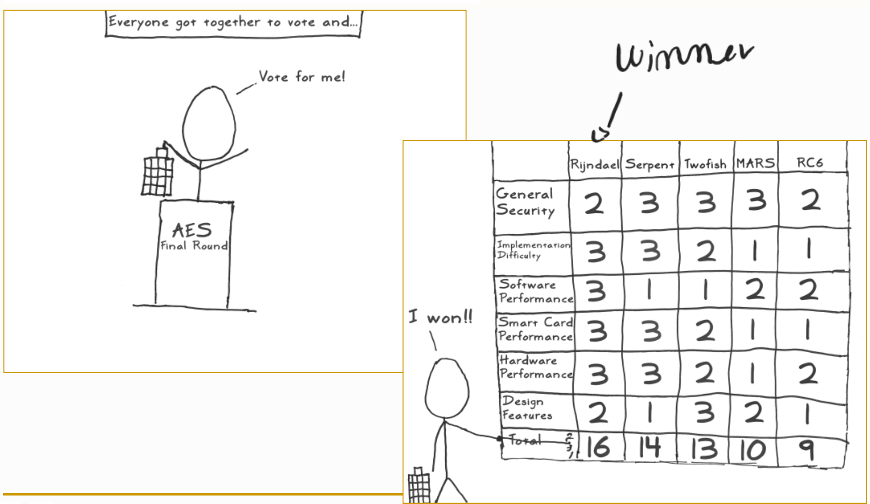
- 1st round
  - 15 candidate algorithms were evaluated by the community
  - Conferences were organized for the evaluation
  - Cryptographic weakness were found
  - Performance issues were identified
    - In a variety of hardware
    - PCs, smart cards, hardware implementations
  - Constrained environment were evaluated
    - Limited memory smart cards, low gate count circuits, FPGAs
- Aug/1999: AES finalists announced
  - MARS, RC6, Rijndael, Serpent, and Twofish



# Rijndael selection as AES

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

Winner




| | Rijndael | Serpent | Twofish | MARS | RC6 |
|---|---|---|---|---|---|
| General Security | 2 | 3 | 3 | 3 | 2 |
| Implementation Difficulty | 3 | 3 | 2 | 1 | 1 |
| Software Performance | 3 | 1 | 1 | 2 | 2 |
| Smart Card Performance | 3 | 3 | 2 | 1 | 1 |
| Hardware Performance | 3 | 3 | 2 | 1 | 2 |
| Design Features | 2 | 1 | 3 | 2 | 1 |
| Total | 16 | 14 | 13 | 10 | 9 |



Block : |128| . . . . . |256|
+16

Key : $\boxed{128}$ . . . . . $\boxed{256}$
+16

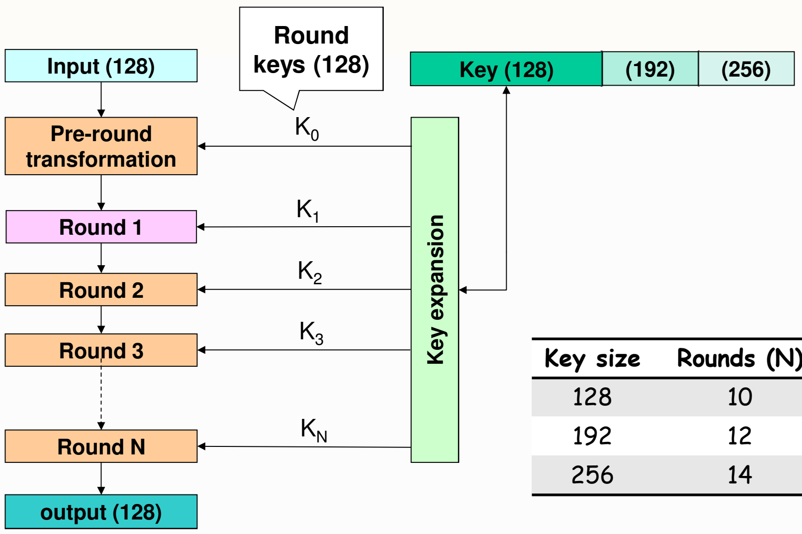
Aes is a variation o Rijndael
it limited the sizes of the keys or Blocks
by 3 values (128, 192, 256)

# AES: evaluation rounds

- 2nd 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

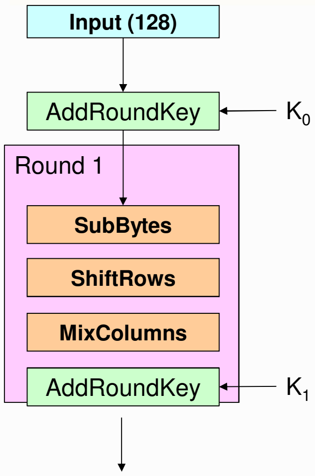


# AES: architecture

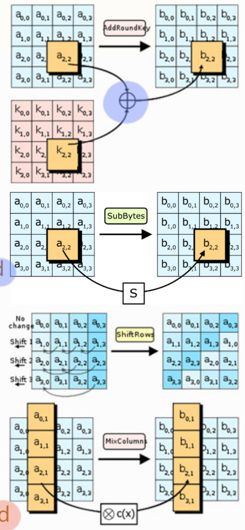



| Key size | Rounds (N) |
|---|---|
| 128 | 10 |
| 192 | 12 |
| 256 | 14 |

# AES: architecture




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

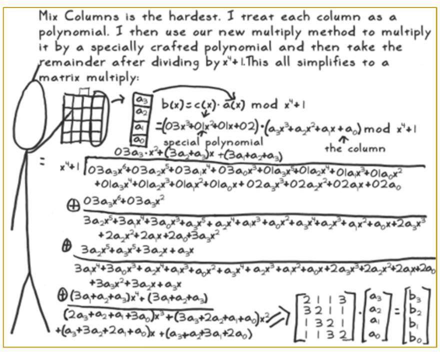


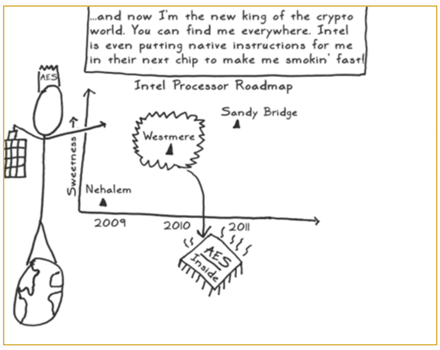

https://aescryptography.blogspot.com



# AES complexity and speed-up

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



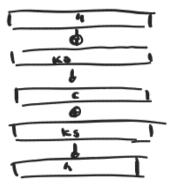
# AES in CPU instruction sets

- ▷ 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





# Stream ciphers

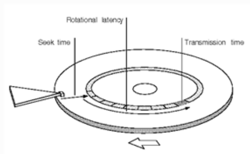
Avoid to guess num before or after a certain sample.

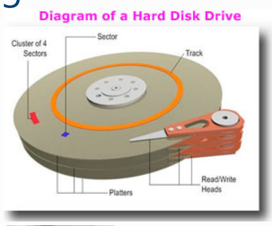
- ▷ Approaches
  - Cryptographically secure pseudo-random generators (PRNG)
    - Using linear feedback shift registers (LFSR)
    - Using block ciphers
    - Other (families of functions, etc.)
  - Usually not self-synchronized
  - Usually without uniform random access
    - No immediate setup of generator's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

chacha 20, salsa 20

# Uniform random access

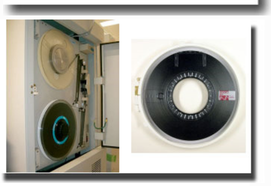
▷ Same time to reach and process any piece of information regardless of its storage location




▷ Uniform
- Memory
- Disks (magnetic, optical)
  - Average $T_{access} = T_{seek} + ½\ T_{revolution}$

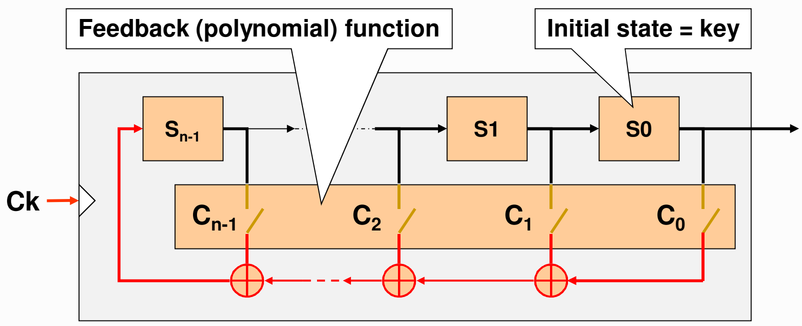
▷ Non-uniform
- Tapes (audio, video, computer)



http://osr507doc.sco.com/en/PERFORM/disk_IO_mech.html
https://www.ict4u.net/components/backing-storage.php



# Linear Feedback Shift Register (LFSR)

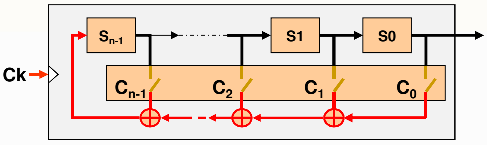



▷ $2^n$-1 non-null sequences
- If one of them has a $2^n$-1 period length, then all have it

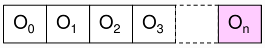
▷ Primitive feedback functions (primitive polynomials)
- All non-null sequences have a $2^n$–1 period length

it passes through all states except the null one

# Linear Feedback Shift Register (LFSR)




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




$O_n = C_0O_0 + C_1O_1 + \ldots + C_{n-1}O_{n-1}$

  - The output must be mixed with something else …



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