# Real World
# Stream Ciphers

*MSc in Information Security & Digital Forensics*.

Mark Cummins
**Institute of Technology Blanchardstwon**
**Phone:** +353 1-885-1156
**Email**: mark.cummins@itb.ie

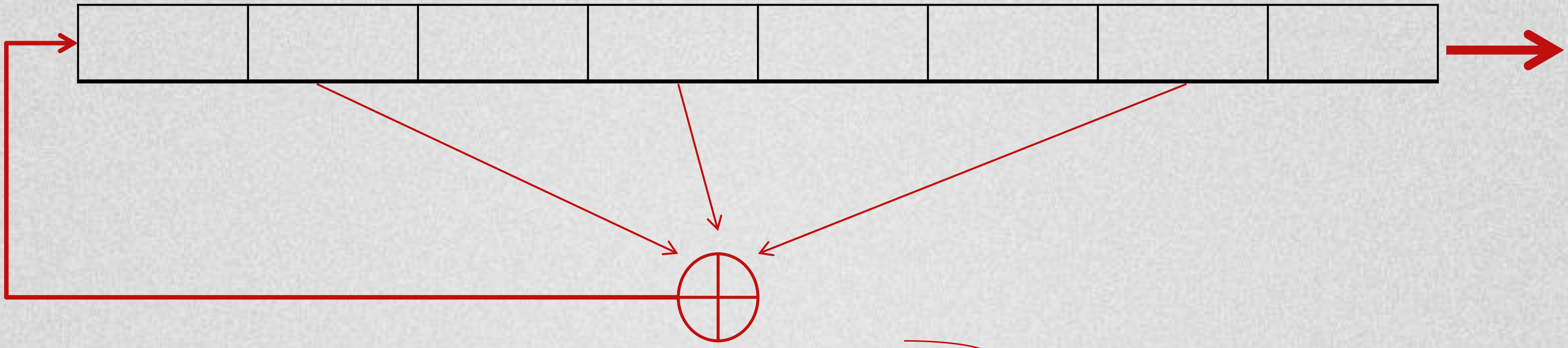
# RC4 *(Old software example, 1987)*

- Used in HTTPS and WEP
- Weaknesses:
  1. Bias in initial output: Pr[ 2nd byte = 0 ] = 2/256
  2. Prob. of (0,0) is $1/256^2 + 1/256^3$
  3. Related key attacks

Mark Cummins
**Institute of Technology Blanchardstwon**
**Phone:** +353 1-885-1156
**Email**: mark.cummins@itb.ie

# CSS *(Old hardware example, badly broken)*

Linear feedback shift register (LFSR):



DVD encryption (CSS): 2 LFSRs

GSM encryption (A5/1,2): 3 LFSRs

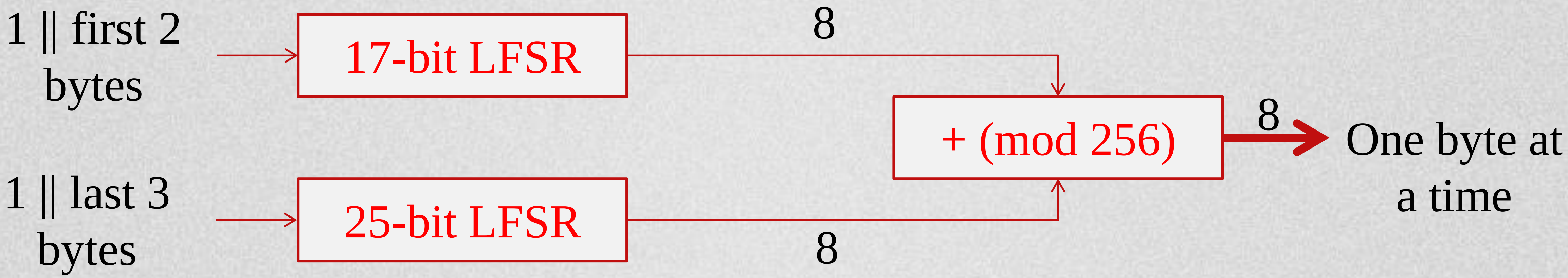
Bluetooth (E0): 4 LFSRs

All Broken

Mark Cummins
**Institute of Technology Blanchardstwon**
**Phone:** +353 1-885-1156
**Email**: mark.cummins@itb.ie

# CSS *(Old hardware example, badly broken)*

CSS: seed = 5 bytes = 40 bits



1 || first 2 bytes

17-bit LFSR

8

1 || last 3 bytes

25-bit LFSR

8

+ (mod 256)

8

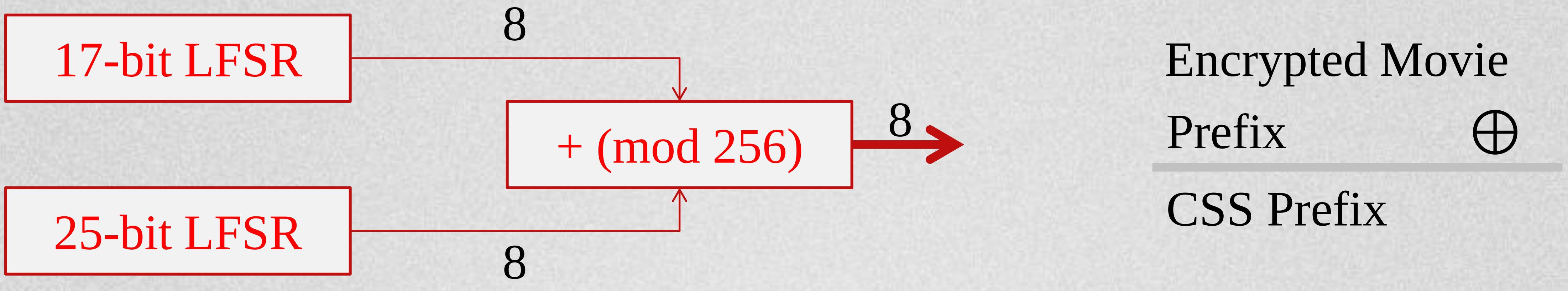
One byte at a time

Easy to break in $2^{17}$

Mark Cummins
**Institute of Technology Blanchardstwon**
**Phone:** +353 1-885-1156
**Email**: mark.cummins@itb.ie

# Cryptananalysis *of CSS*

Linear feedback shift register  (LFSR):



For all possible initial settings of 17-bit LFSR do:

- Run 17-bit LFSR to get 20 bytes of output
- Subtract from CSS prefix   ⇒   candidate 20 bytes output of 25-bit LFSR
- If consistent with 25-bit LFSR, found correct initial settings of both !!

Using key, generate entire CSS output

Mark Cummins
**Institute of Technology Blanchardstwon**
**Phone:** +353 1-885-1156
**Email**: mark.cummins@itb.ie

# Modern Stream Ciphers - EStream

$$\text{PRG:}\quad \underbrace{\{0,1\}^s}_{\text{Seed}} \times \underbrace{R}_{\text{Nonce}} \longrightarrow \{0,1\}^n$$

Nonce: a non-repeating value for a given key.

$$E(k, m\,;\, r) = m \oplus \text{PRG}(k\,;\, r)$$

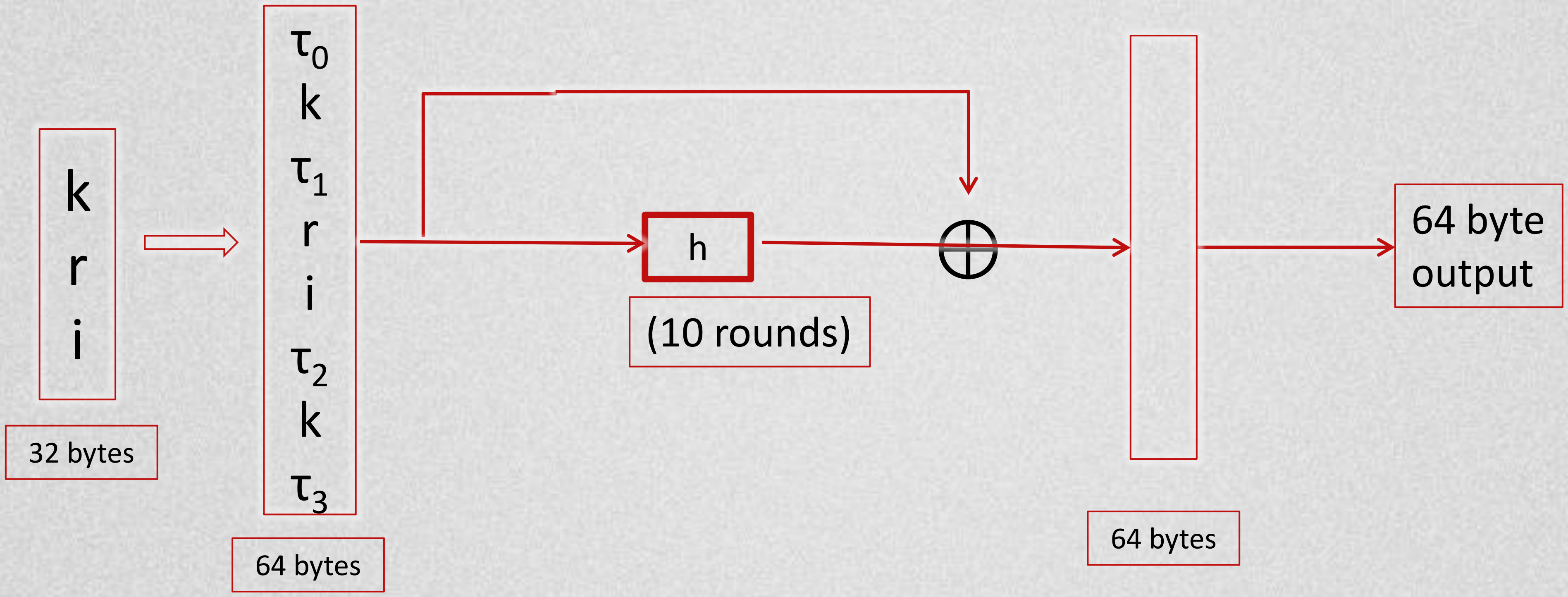
The pair (k,r) is never used more than once.

Mark Cummins
**Institute of Technology Blanchardstwon**
**Phone:** +353 1-885-1156
**Email**: mark.cummins@itb.ie

# *EStream – Salsa20*

Salsa20: $\{0,1\}^{128 \text{ or } 256} \times \underbrace{\{0,1\}^{64}}_{\text{Nonce}} \longrightarrow \{0,1\}^{n}$ (max n = $2^{73}$ bits)

Salsa20( k ; r) := H( k , (r, 0)) ll H( k , (r, 1)) ll …



h: invertible function. Designed to be fast on x86 (SSE2)

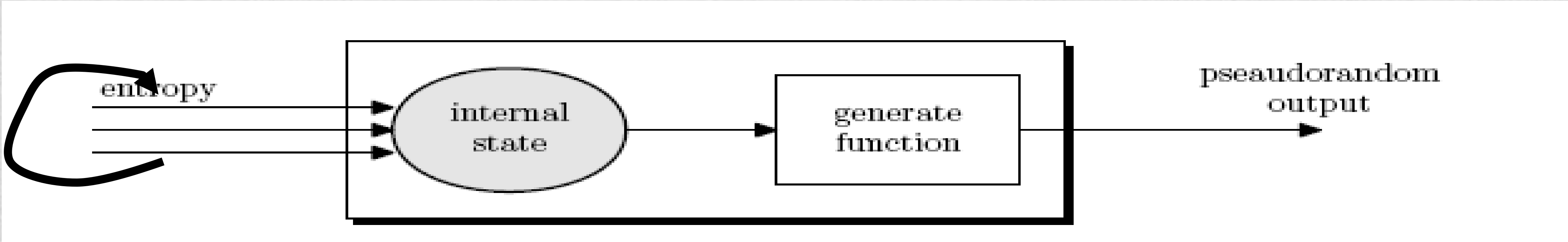
Mark Cummins
Institute of Technology Blanchardstwon
**Phone:** +353 1-885-1156
**Email**: mark.cummins@itb.ie

# Is Salsa20 Secure?

- Unknown: no known **provably** secure PRGs
- In reality: no known attacks better than exhaustive search

Mark Cummins
**Institute of Technology Blanchardstwon**
**Phone:** +353 1-885-1156
**Email**: mark.cummins@itb.ie

# Generating Randomness (keys, IV)



Pseudo random generators in practice: (e.g. /dev/random)

- Continuously add entropy to internal state
- Entropy sources:
  - Hardware RNG: Intel **RdRand** inst. (Ivy Bridge). 3Gb/sec.
  - Timing: hardware interrupts (keyboard, mouse)

NIST SP 800-90: NIST approved generators

Mark Cummins
**Institute of Technology Blanchardstwon**
**Phone:** +353 1-885-1156
**Email**: mark.cummins@itb.ie

# Thank You !

## End of Section

Mark Cummins
**Institute of Technology Blanchardstwon**
**Phone:** +353 1-885-1156
**Email**: mark.cummins@itb.ie