Stream ciphers

Stream Cipher

Suppose you want to encrypt a *stream* of data, such as:

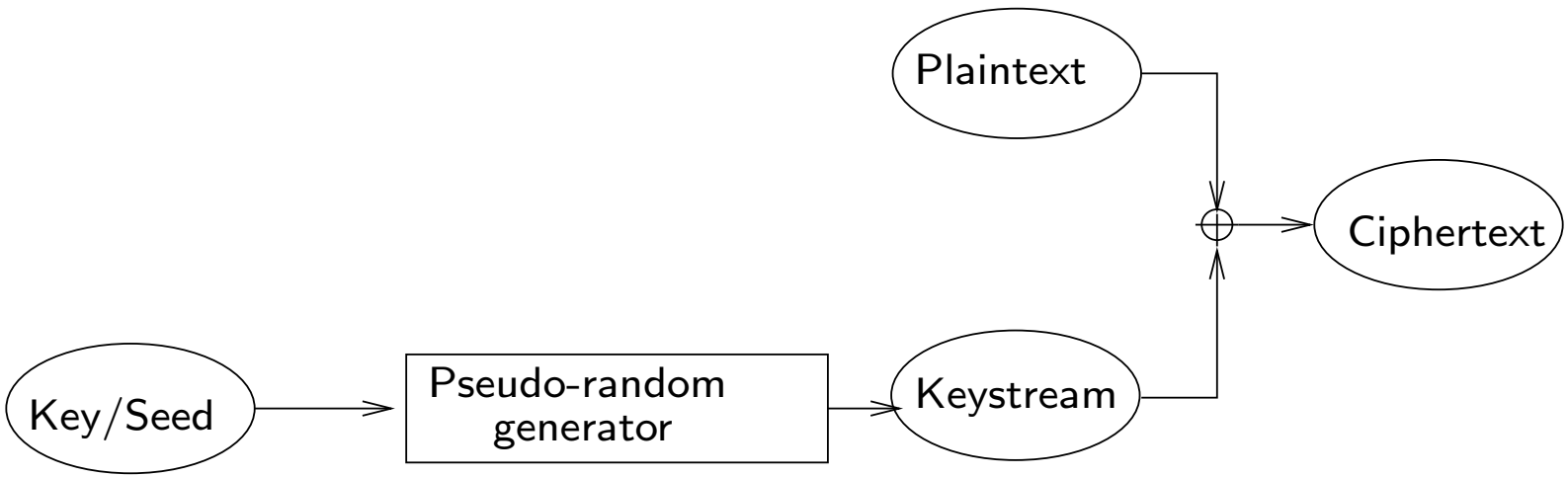
- the data from a keyboard
- the data from a sensor

Block ciphers might be awkward/inefficient (consider each keystroke as a block??).

Stream cipher

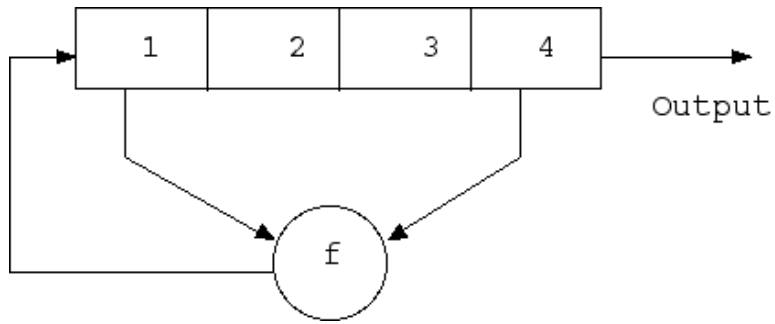
The way we use a stream cipher is a bit like a the way we use a one-time pad (OTP): we generate a key that's as long as the data. However, we generate it from a short random seed, so it is not random like the key used in the OTP.

The key stream is not random, but pseudorandom.



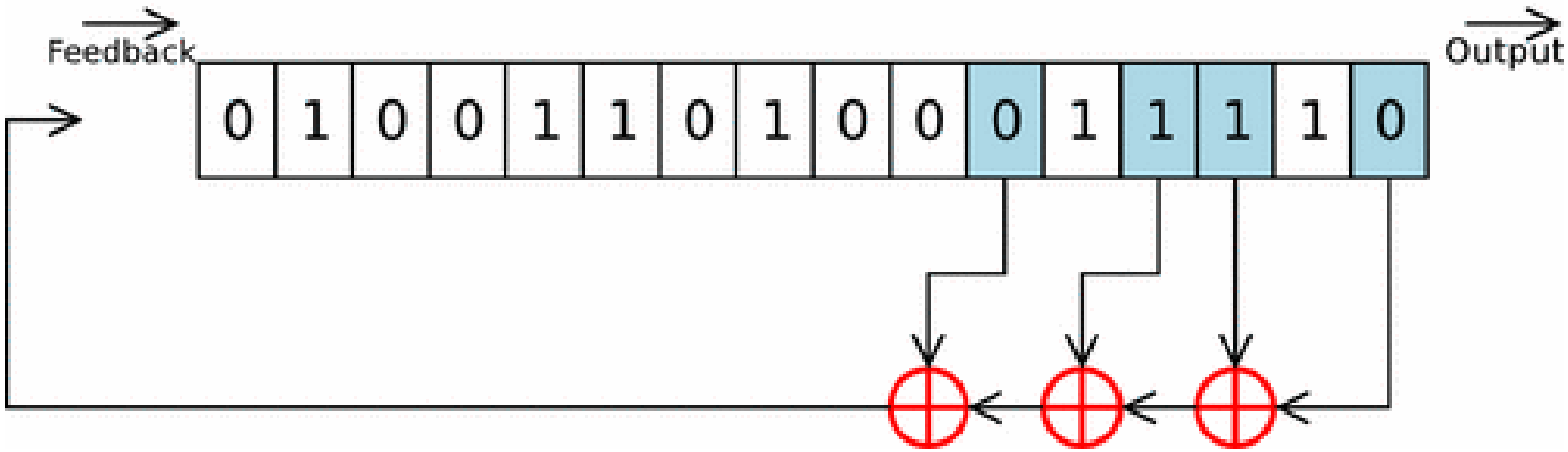
Example 1: LFSR

Linear Feedback Shift Register:
Building block for many stream ciphers
Can be implemented very efficiently
Key idea: have register of single bit cells shifted by one at every clock cycle together with feedback function



Source: Wikipedia

Example:



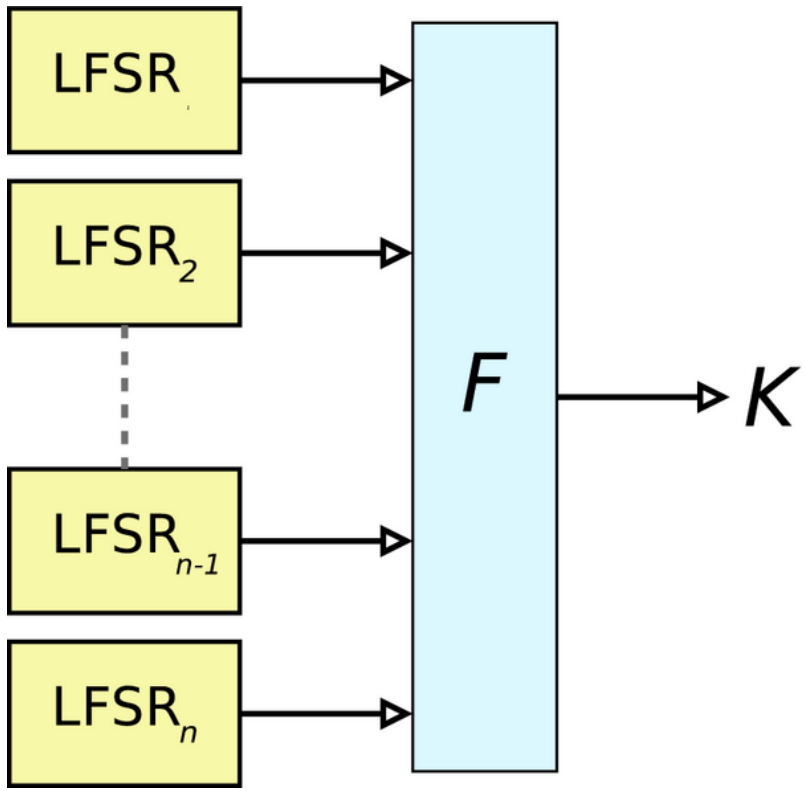
Source: Wikipedia

Reasoning about LFSRs

Interesting property: Length of keystream period
If the LFSR has n bits, the keystream period will be at most 2^n, but could be much less. LFSRs are not very secure.

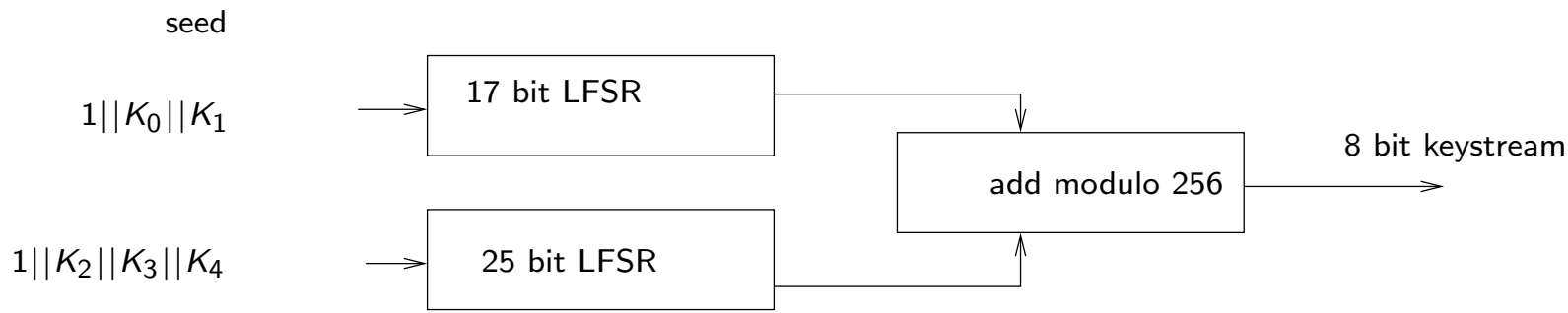
Combining LFSRs

LFSRs are insecure in practice (given a lot of output, the tap positions can be computed fairly efficiently)
Hence multiple LFSRs are combined in non-linear fashion



Source: Wikipedia

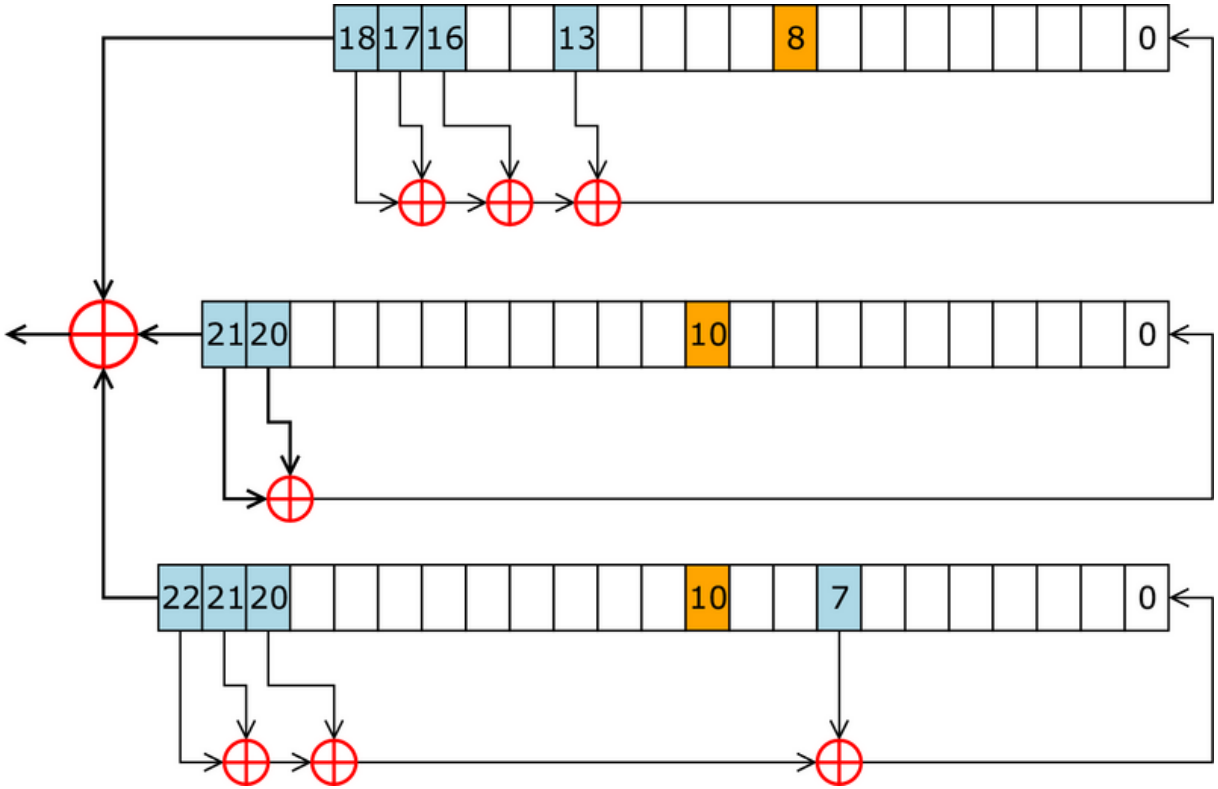
Content Scrambling System (CSS) is an encryption system used on DVDs. Sector encryption is combination of two LFSR's added modulo 256 (observing carry bit from previous addition):



However, analysis showed that this can be broken in time 2^{17} (which is much less than the "expected" 2^{42}) [see the wikipedia page on CSS].

A5/1

- Stream cipher used in GSM mobile phone communication
- Became public knowledge through leaks and reverse engineering
- Built from three LFSRs with irregular clock cycle
  - 54 bit secret key and 22 bit initialisation vector
- A register is shifted only if its clock bit is the same as majority of the three clock bits



Source: Wikipedia

Security of A5/1

Better design: Clock shift make cryptanalysis much harder
However, advanced techniques means mainstream PC with terabytes of flash memory (to store pre-processed tables) can break A5/1 with probability $\geq 90\%$ in a few seconds

Example 2: RC4

Stream cipher invented 1987 by Ron Rivest
Main datastructure is array S of 256 bytes.

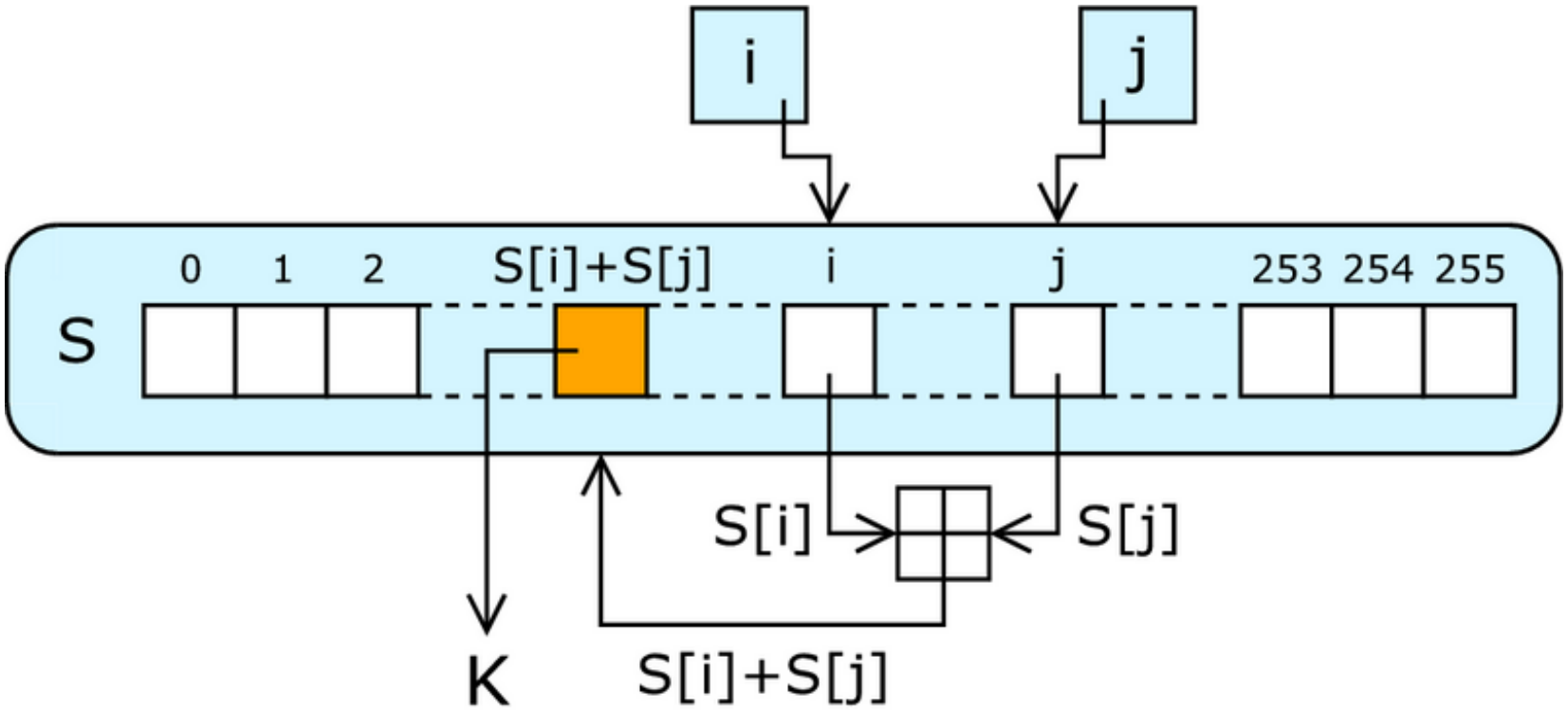
Consists of two phases:

- Initalisation of S phase ("key schedule")
- Keystream generation phase

Code for RC4

```
for i := 0 to 255 do
    S[i] := i
end
j := 0
for i := 0 to 255 do
    j := (j + S[i] + K[i mod keylength]) mod 256
    swap(S[i],S[j])
end
i := 0
j := 0
while GeneratingOutput:
        i := (i + 1) mod 256                    every element will get swapped eventually
        j := (j + S[i]) mod 256                 chose the swapped element randomly
        swap(S[i],S[j])                         evolve the array
        output S[(S[i] + S[j]) mod 256]         don't reveal the internal state
end
```

Graphical representation



Source: Wikipedia

Properties of RC4

- Extremely compact, and beautiful
- Exhaustively studied
  - Two books, and hundreds of research papers
- Pretty good!
- But not good enough by modern standards:
  - Numerous "attacks" (biases in the stream, especially the first few bytes)
  - Led to real attacks on WEP, and modes of TLS that use it

WEP

Old standard for encryption on wireless networks
based on RC4:

- RC4 is run on a seed consisting of the pre-shared WEP key (128 bits) and an initialisation vector of 24 bits.
- Initialisation vector only 24 bits, hence key streams repeat after at most 2^{24} frames
- First bytes of key stream can be known by adversary because standard headers are always sent.

These two facts have led to the development of a method which can crack the key in minutes on modern PC hardware.
Thus, WEP is is seriously broken - don't use it.

Example 3

A block cipher like AES, using counter mode.
This is certainly the best of the stream ciphers seen so far!