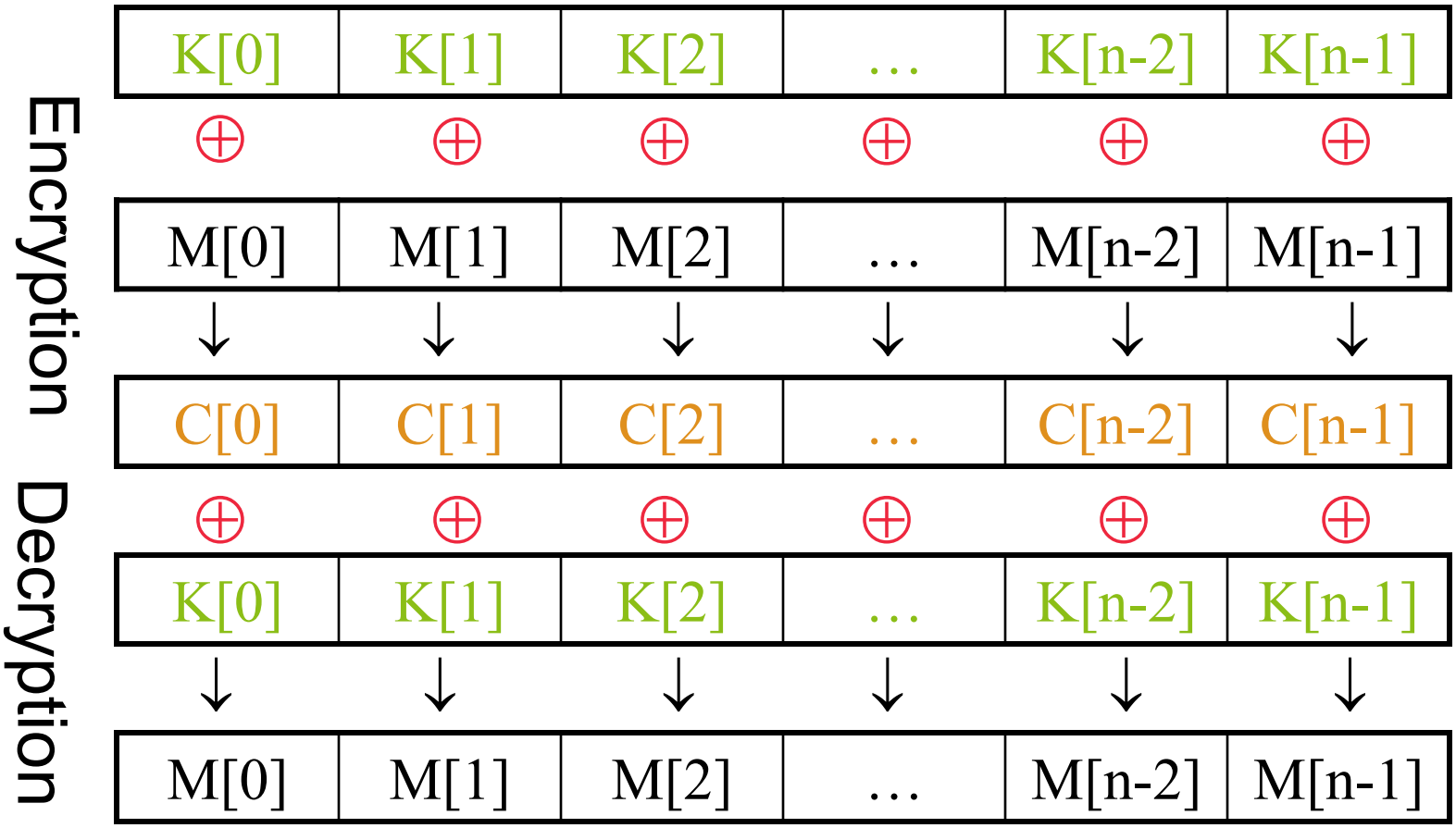
Stream Ciphers

- Readings
  - Section 3.6
  - A5/1
  - WEP

Binary One-Time Pad



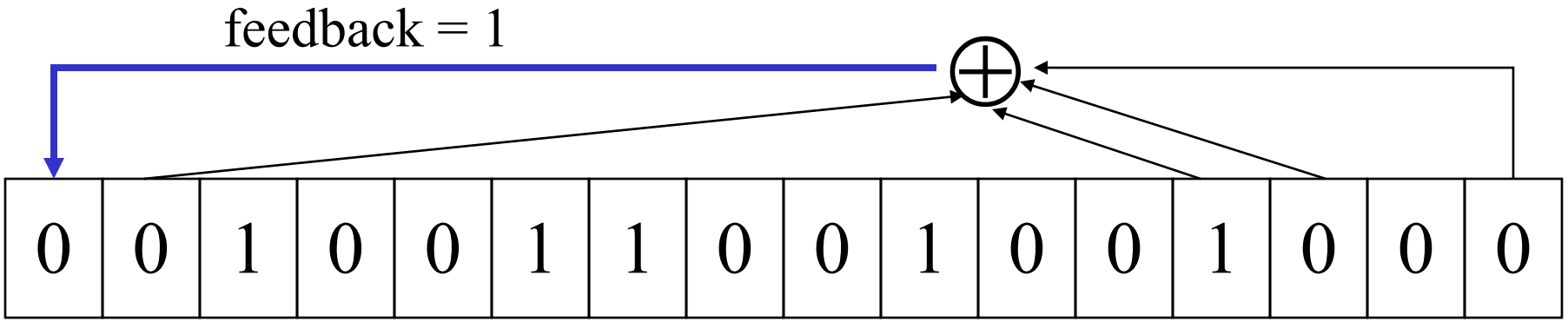
Idea behind Stream Ciphers (approximate a one-time pad)

- Start with a fixed-length, shared secret. This is generally called the **seed** *s.*
- Use a procedure that, with the seed as input, generates a stream of bits that seems random, but which is in fact deterministically computable (from *s*).
- Use this stream (**keystream**) as the one-time pad: XOR it with the plaintext.
- Note: we previously showed how block ciphers could be used to generate stream ciphers
  - OFB
  - CFB

Types of Stream Ciphers

- A stream cipher is a finite state machine (finite input, fixed memory size, deterministic). Two main types:
  - **Key-auto-Key** (KAK, synchronous) -- state determined by last bits of keystream
  - **Ciphertext-auto-key** (CTAK, self-synchronizing) -- state determined by last bits of ciphertext

Linear Feedback Shift Registers



- Compute the parity of "tap" entries in a register [16,14,13,2]
  - Note that position are numbered 1-16 from left to right
- Shift the register entries right
- Enter the new bit (output: XOR of tap bits, with value 1) in position 1 (leftmost in picture)
- Keystream is the sequence of output values of shift register
  - normally rightmost bit

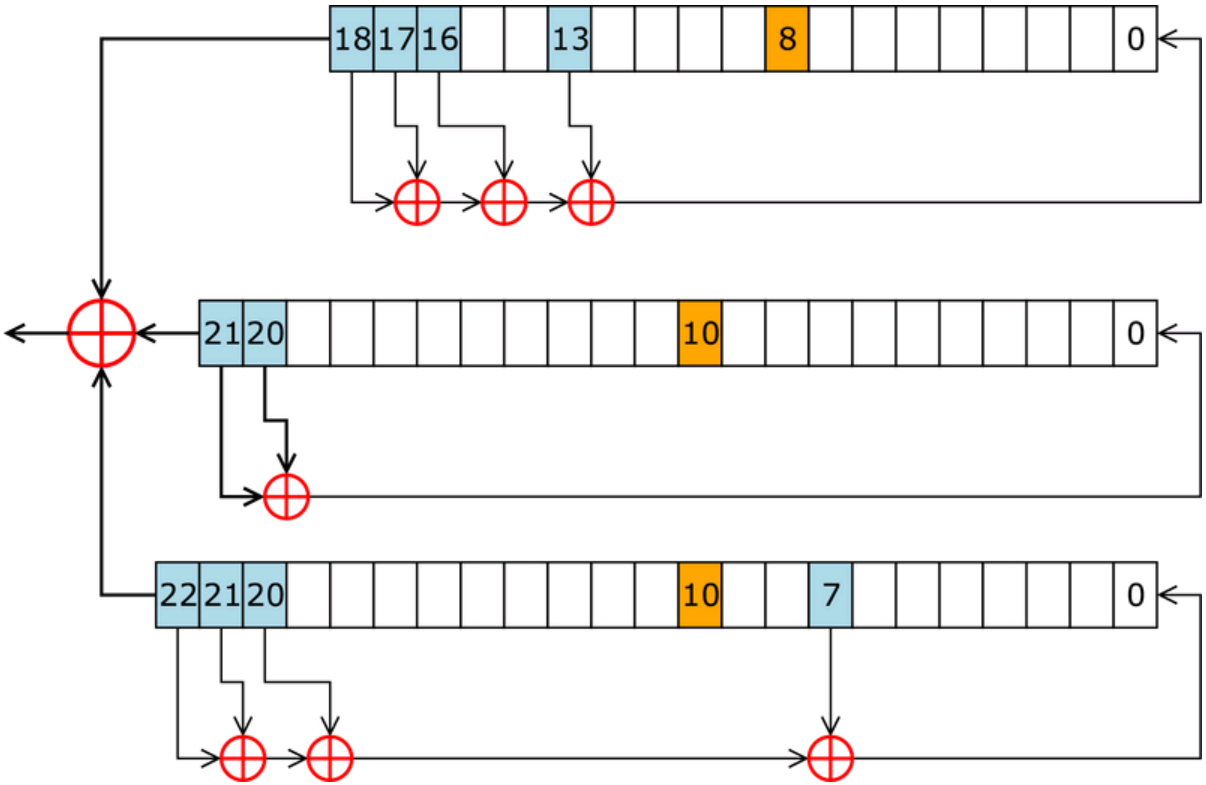
Properties of Shift Registers

- Very efficient generators of pseudo-random sequences. Think of the newly computed bits as the output
- Generate provably long sequences before cycling
- Shift registers used in crypto have often resulted in weak ciphers, such as A5/x which has been attacked, even though the shrinking generator and extensions can be fairly good.

Shrinking Generator

- Two LFSR generate keystreams s_k and t_k
- If $s_k = 1$, output t_k
- If $s_k = 0$, output nothing; increase k
- Buffer output in order to disguise timing delays which reveal information about the state of keystream s.

A5/1



- Used in GSM cellular phone system
- Read more about A5/1

RC4 Stream Cipher

- Generates a one time pad (string of bits)
- Designed by Ron Rivest
- Algorithm is simple and fast
  - Ideal for software implementation
- Seed can be from 1 to 256 octets long
- Also makes a very good pseudo random number generator
  - http://en.wikipedia.org/wiki/RC4

RC4

- A state array of 256 bytes: S[0, …, 255]
- A seed, from 1 to 256 bytes: K[0, …, n]
- Initialize state as S[i] = i;
  - j = 0
  - FOR i = 0 … 255
    - j = j + S[i] + K[i mod n] mod 256
    - SWAP(S[i], S[j])

RC4 Stream Generator

- Procedure to output next octet
- i = 0; j= 0;
- WHILE(TRUE)
  - i = i + 1 mod 256;
  - j = j + S[i] mod 256;
  - SWAP(S[i] , S[j]);
  - OUTPUT(S[(S[i] + S[j]) mod 256]);

RC4 Weaknesses

- Initialization starts from a known state.
  - The first bytes of the RC4 key-stream are distinguishable from random output, and reveal information about the seed.
- Recommendations:
  - Use random seeds, without ever re-using.
  - Generate strong pseudo-random seeds
  - Drop initial bytes of key-stream
    - practical: drop 768; paranoid: drop 3072

WEP (Wired Equivalent Privacy)

- Designed to protect eavesdropping on wireless 802.11 networks between a client station and an access point
- Uses the RC4 stream cipher
- In WEP, a 40 bit or 104 bit shared secret key K is augmented by a 24 bit IV to form the seed S = (IV, K).
- Many attacks were found possible against WEP with the result that its use has been deprecated and tools exist that can easily crack the key.

Is RC4 Insecure?

- RC4 has been well-studied.
  - No known methods to break RC4 (except brute-forcing the key) when used according to recommendations.
  - E.g.: Robust implementation of RC4 within SSL.
- RC4 was used in a very insecure way in the WEP protocol:
  - No method to distribute initial keys
  - Poor handling of IVs
  - No dropping of the first key-stream bytes
- 802.11.x took unnecessary risks.

Stream Cipher Summary

- Stream ciphers are efficient
  - Useful for secure communication with constrained devices (cell phones, smartcards)
- Good stream ciphers available (?)
  - Even as theoretical framework still in development
- Never re-use key-streams: must provide mechanism to change IV EVERYTIME.

Reading Assignments

- Sections 2.6, 5.1, 5.2, 5.4, 5.6, 5.7