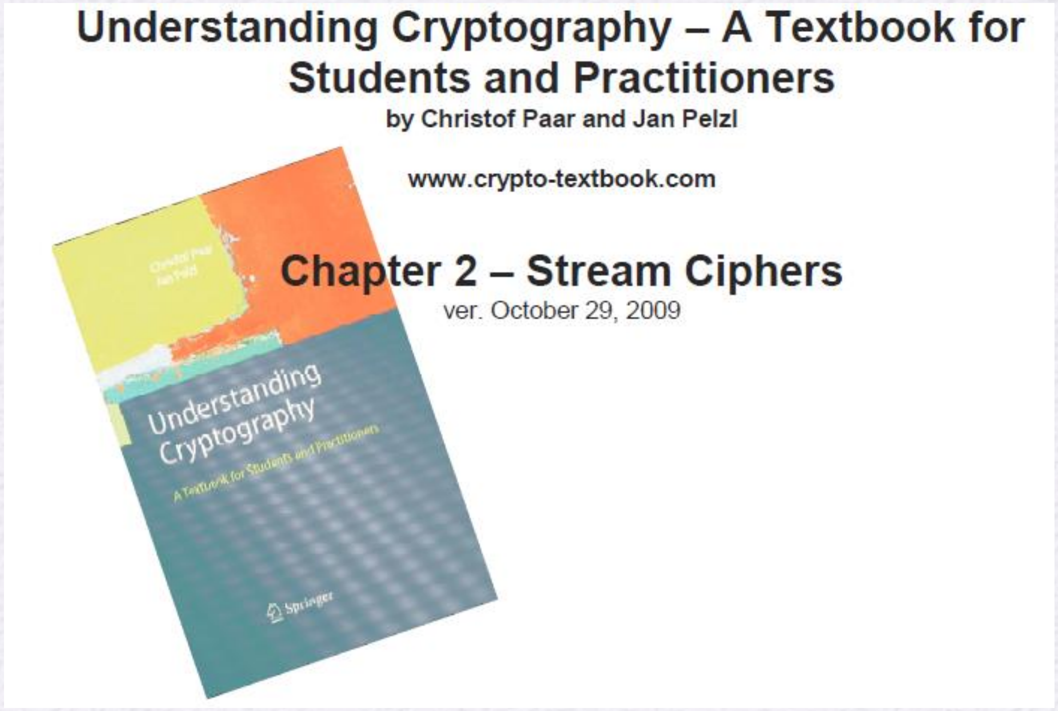
Chapter 4

Steam Ciphers

Acknowledgement

- Most of the slides are from this book.
- I highly recommend. Excellent book!


Understanding Cryptography – A Textbook for Students and Practitioners
by Christof Paar and Jan Pelzl

www.crypto-textbook.com

Chapter 2 – Stream Ciphers
ver. October 29, 2009

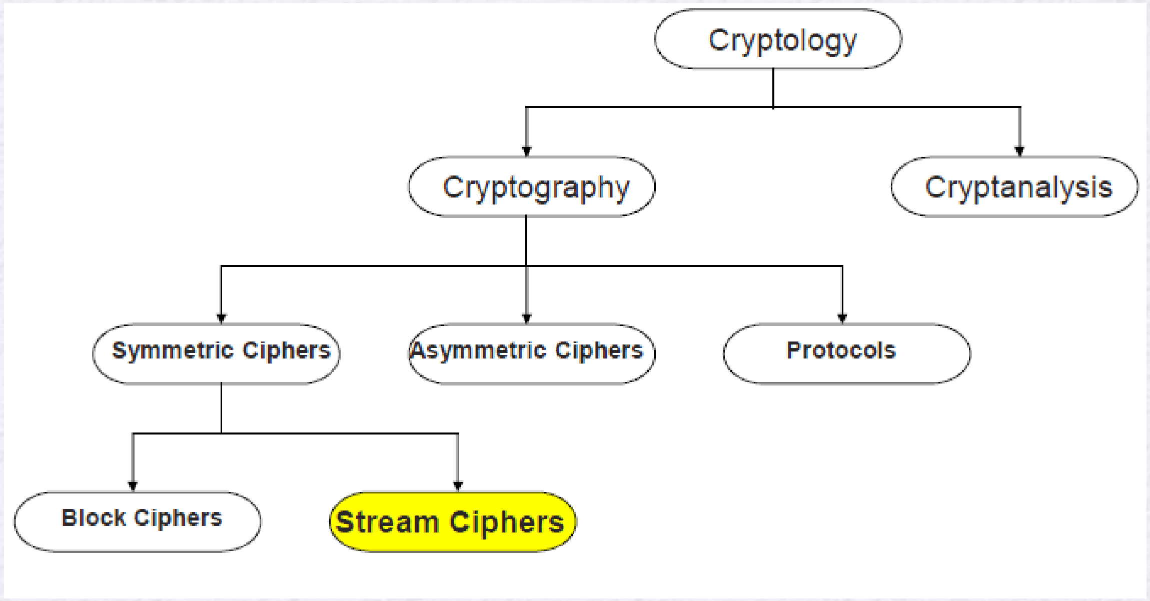
Understanding Cryptography
A Textbook for Students and Practitioners
Springer

Outline

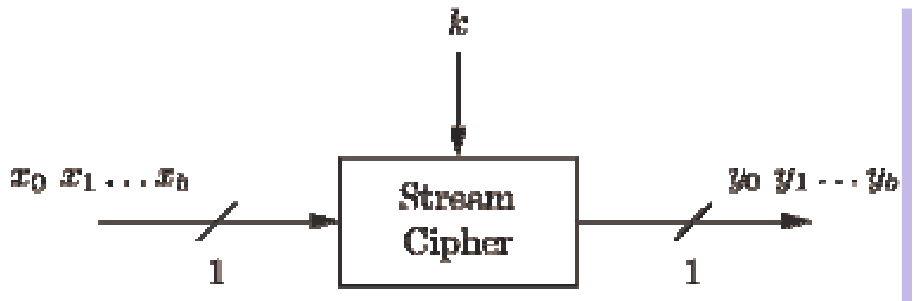
- Intro to stream ciphers
- Random number generators (RNGs)
- One-Time Pad (OTP)
- Linear feedback shift registers (LFSRs)
- Trivium: a modern stream cipher

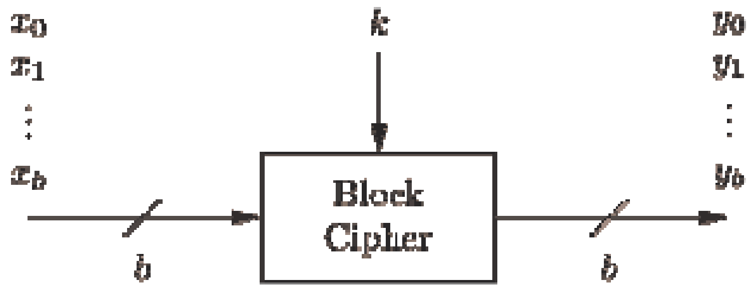
Stream Ciphers in Cryptography

- Both stream ciphers and block ciphers are symmetric encryption algorithms.
  - Cipher uses one secret key for encryption and decryption.
- Stream Ciphers encrypt bits individually.
- Block ciphers encrypt a full block (several bits.)



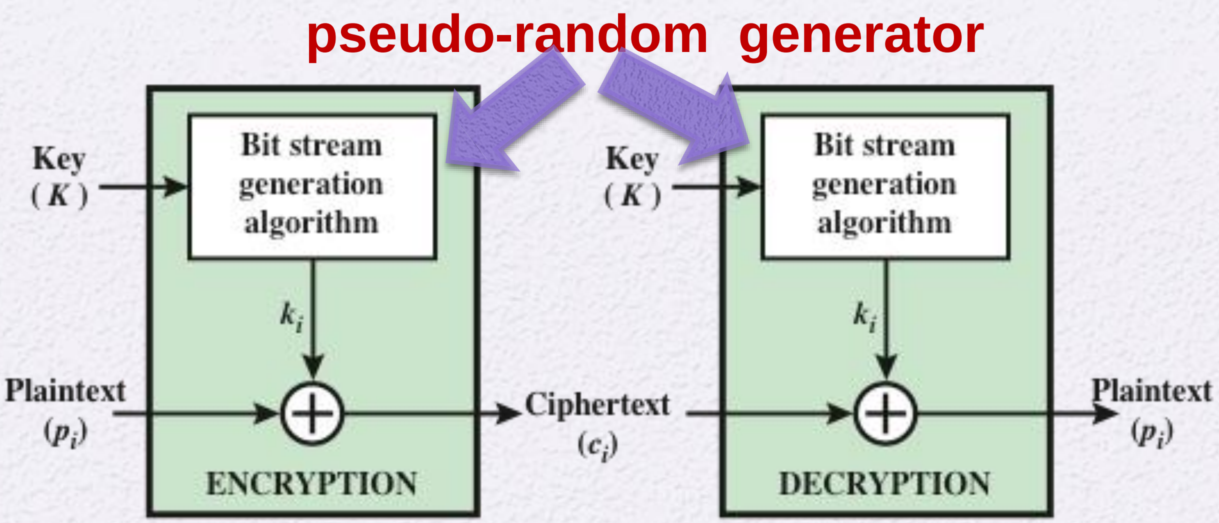
Stream Cipher vs. Block Cipher



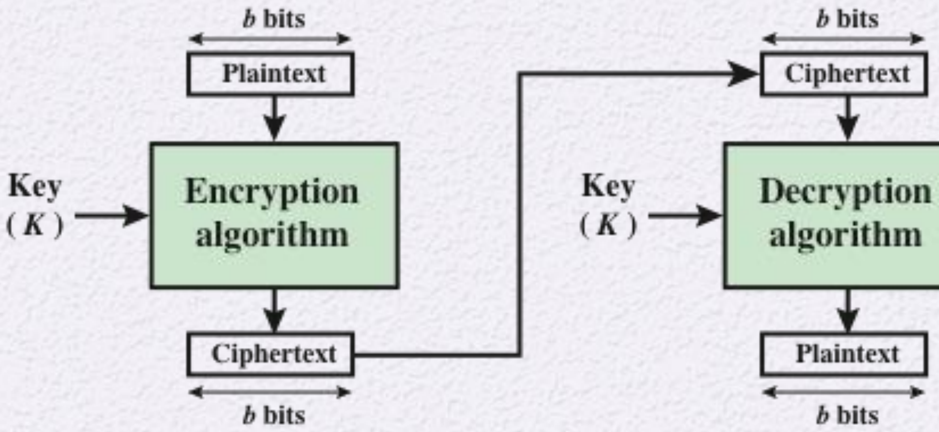


- **Stream Ciphers**
  - Encrypt bits individually
  - Usually small and fast → common in embedded devices (e.g., A5/1 for GSM phones)

- **Block Ciphers:**
  - Always encrypt a full block (several bits)
  - Are common for Internet applications



(a) Stream Cipher Using Algorithmic Bit Stream Generator

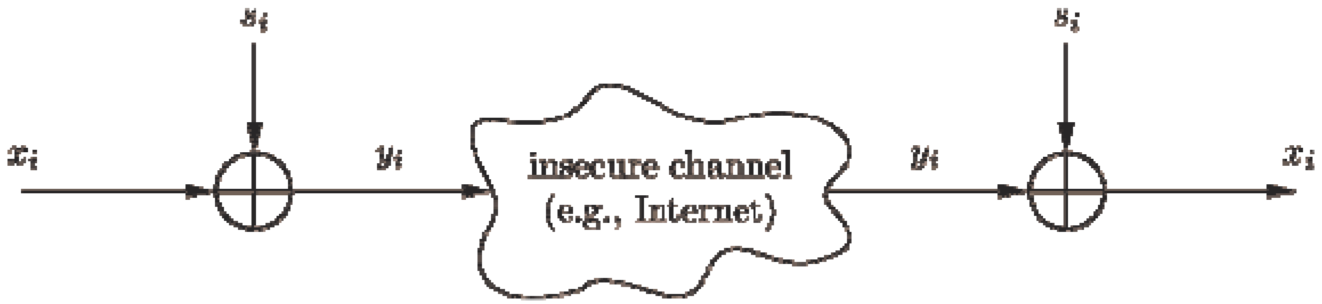


(b) Block Cipher

Figure 4.1 Stream Cipher and Block Cipher

Encryption and Decryption with Stream Ciphers

Plaintext x_i, ciphertext y_i and key stream s_i consist of individual bits

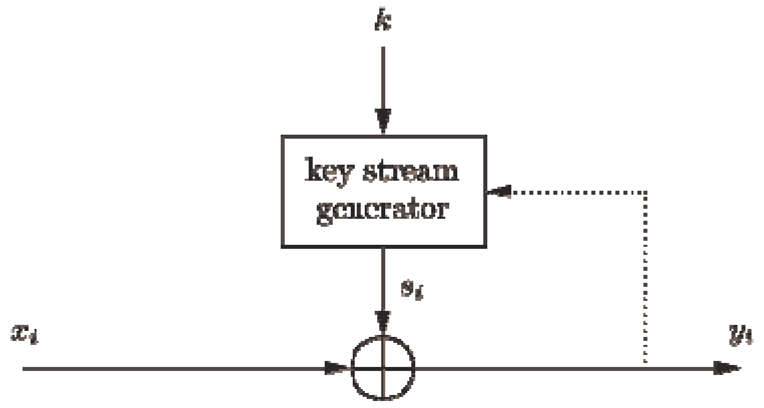


- Encryption and decryption are simple additions modulo 2 (aka XOR)
- Encryption and decryption are the same functions

- **Encryption:** $y_i = e_{si}(x_i) = x_i + s_i \bmod 2$ $\qquad x_i, y_i, s_i \in \{0,1\}$
- **Decryption:** $x_i = e_{si}(y_i) = y_i + s_i \bmod 2$

- Note the notation $y_i = e_{si}(x_i)$ means: $y_i =$ E(Key=s_i, $\mathrm{P} = x_i$)
- Addition on (mod 2) is the same as binary XOR operation (more discussion later).

Stream Cipher: Synchronous vs. Asynchronous

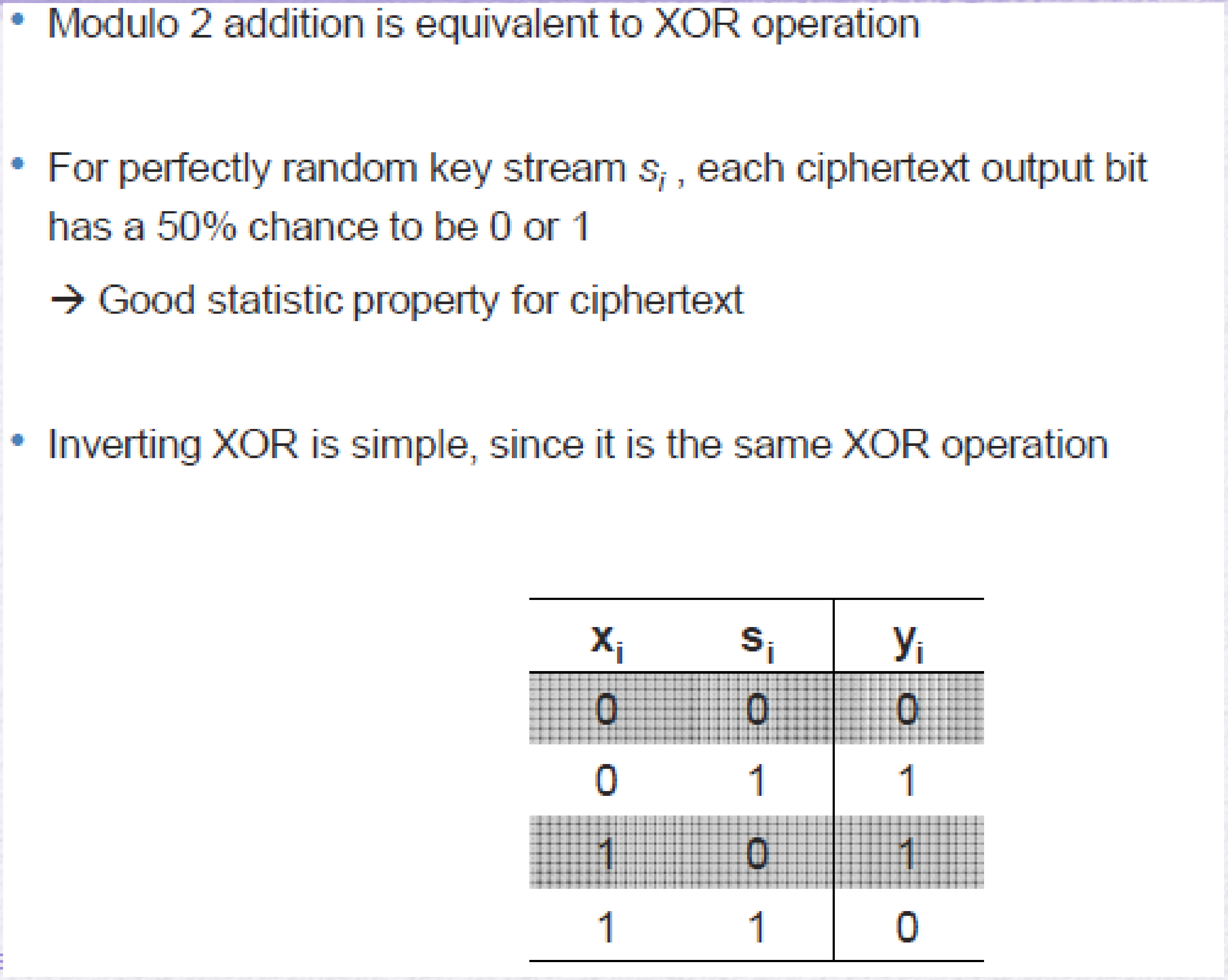


- Security of stream cipher depends entirely on the key stream s_i :
  - Should be **random** , i.e., Pr(s_i = 0) = Pr(s_i = 1) = 0.5
  - Must be **reproducible** by sender and receiver

- **Synchronous Stream Cipher**
  - Key stream depend only on the key (and possibly an initialization vector IV)

- **Asynchronous Stream Ciphers**
  - Key stream depends also on the ciphertext (dotted feedback enabled)

Module 2 Addition As Encryption Function



- Modulo 2 addition is equivalent to XOR operation

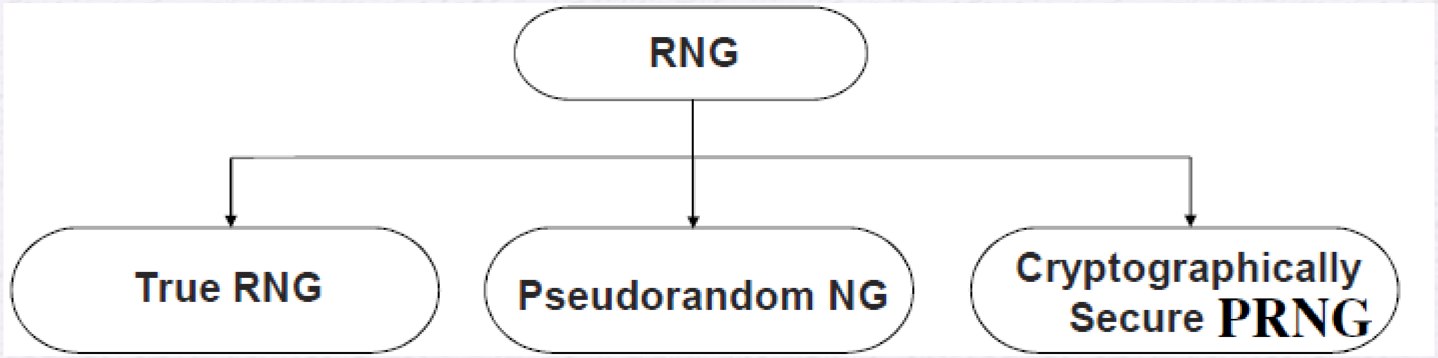
- For perfectly random key stream s_i , each ciphertext output bit has a 50% chance to be 0 or 1

  → Good statistic property for ciphertext

- Inverting XOR is simple, since it is the same XOR operation

x_i	s_i	y_i
0	0	0
0	1	1
1	0	1
1	1	0

Random Number Generators (RNGs)

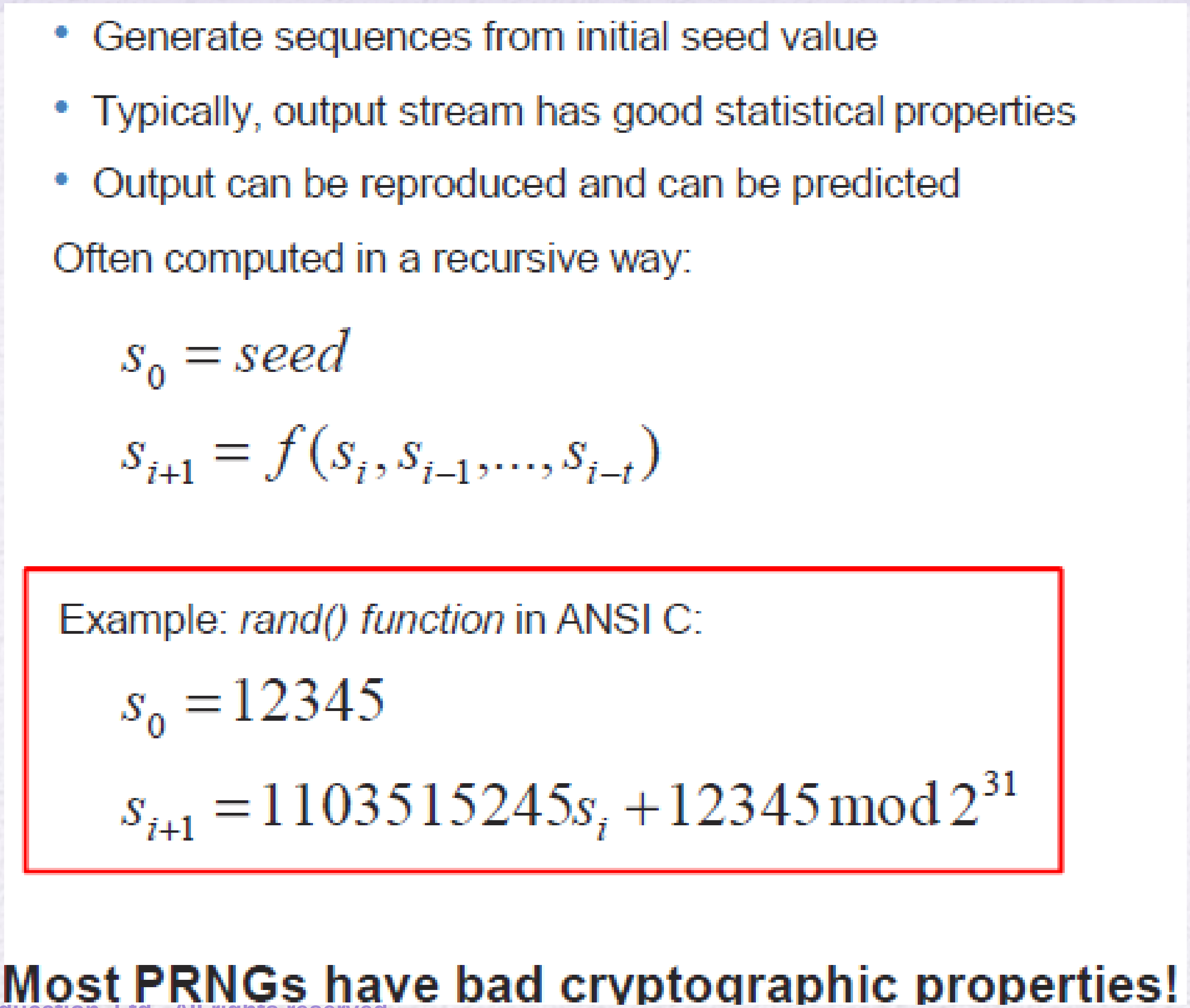


- True random number generators (TRNGs) are characterized by the fact that their output cannot be reproduced.
- Pseudorandom number generators (PRNGs) generate sequences which are computed from an initial seed value.
- Cryptographically secure pseudorandom number generators (CSPRNGs) are a special type of PRNG, and is unpredictable.

True Random Number Generators (TRNG)

- True random number generators (TRNGs) are characterized by the fact that their output cannot be reproduced.
- For instance, if we flip a coin 100 times and record the resulting sequence of 100 bits, it will be virtually impossible for anyone on Earth to generate the same 100 bit sequence. The chance of success is $1/2^{100}$, which is an extremely small probability.
- TRNGs are based on physical processes. Examples include
  - coin flipping,
  - rolling of dice,
  - semiconductor noise,
  - clock jitter in digital circuits
  - radioactive decay.
- In cryptography, TRNGs are often needed for generating session keys, which are then distributed between Alice and Bob, and for other purposes.

Pseudorandom Number Generator (PRNG)



- Generate sequences from initial seed value
- Typically, output stream has good statistical properties
- Output can be reproduced and can be predicted

Often computed in a recursive way:

$$s_0 = seed$$

$$s_{i+1} = f(s_i, s_{i-1}, \ldots, s_{i-t})$$

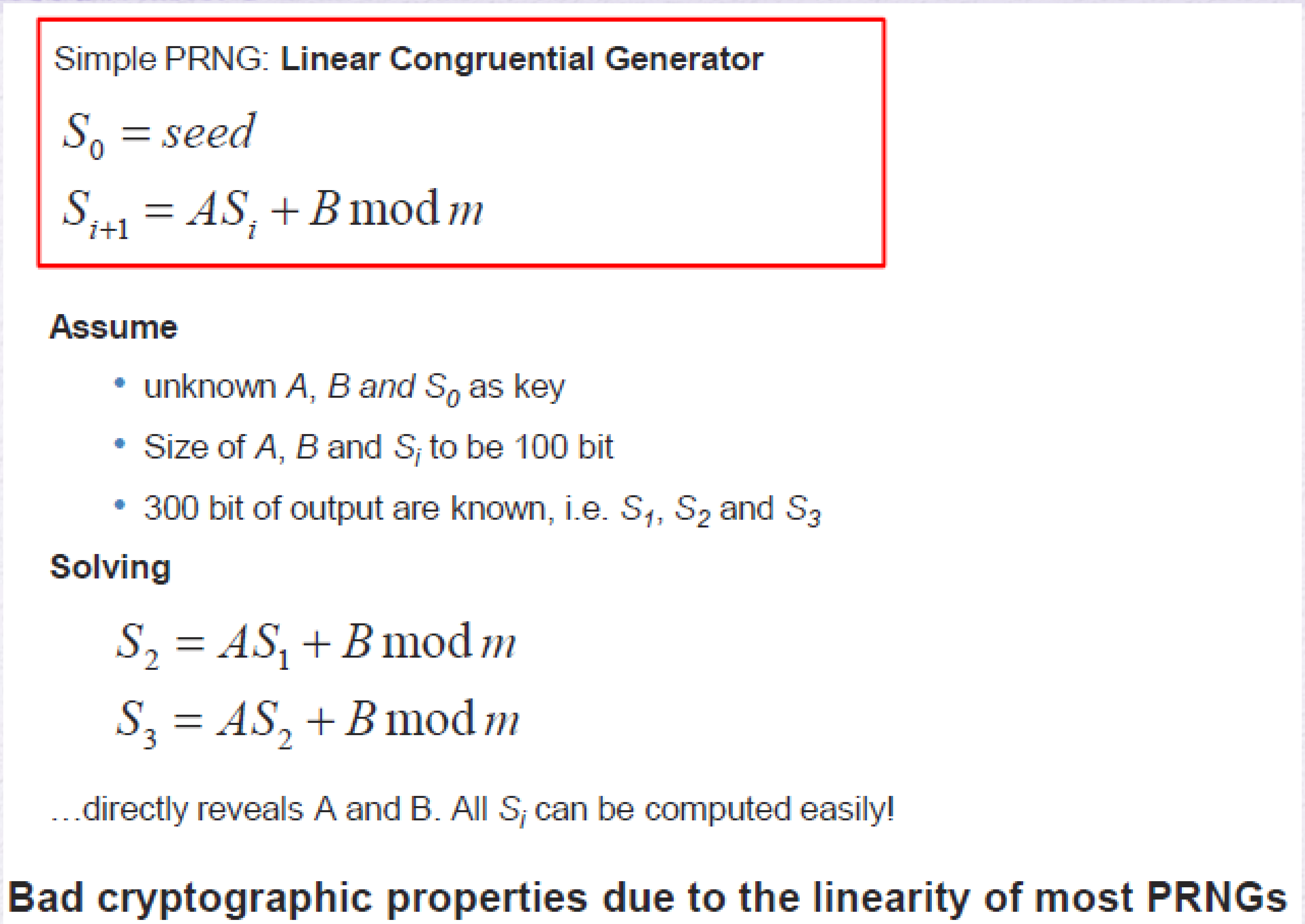
Example: *rand() function* in ANSI C:

$$s_0 = 12345$$

$$s_{i+1} = 1103515245 s_i + 12345 \bmod 2^{31}$$

Most PRNGs have bad cryptographic properties!

Cryptanalyzing a Simple PRNG



Simple PRNG: **Linear Congruential Generator**

$$S_0 = seed$$

$$S_{i+1} = AS_i + B \bmod m$$

Assume

- unknown A, B *and* S_0 as key
- Size of A, B and S_i to be 100 bit
- 300 bit of output are known, i.e. S_1, S_2 and S_3

Solving

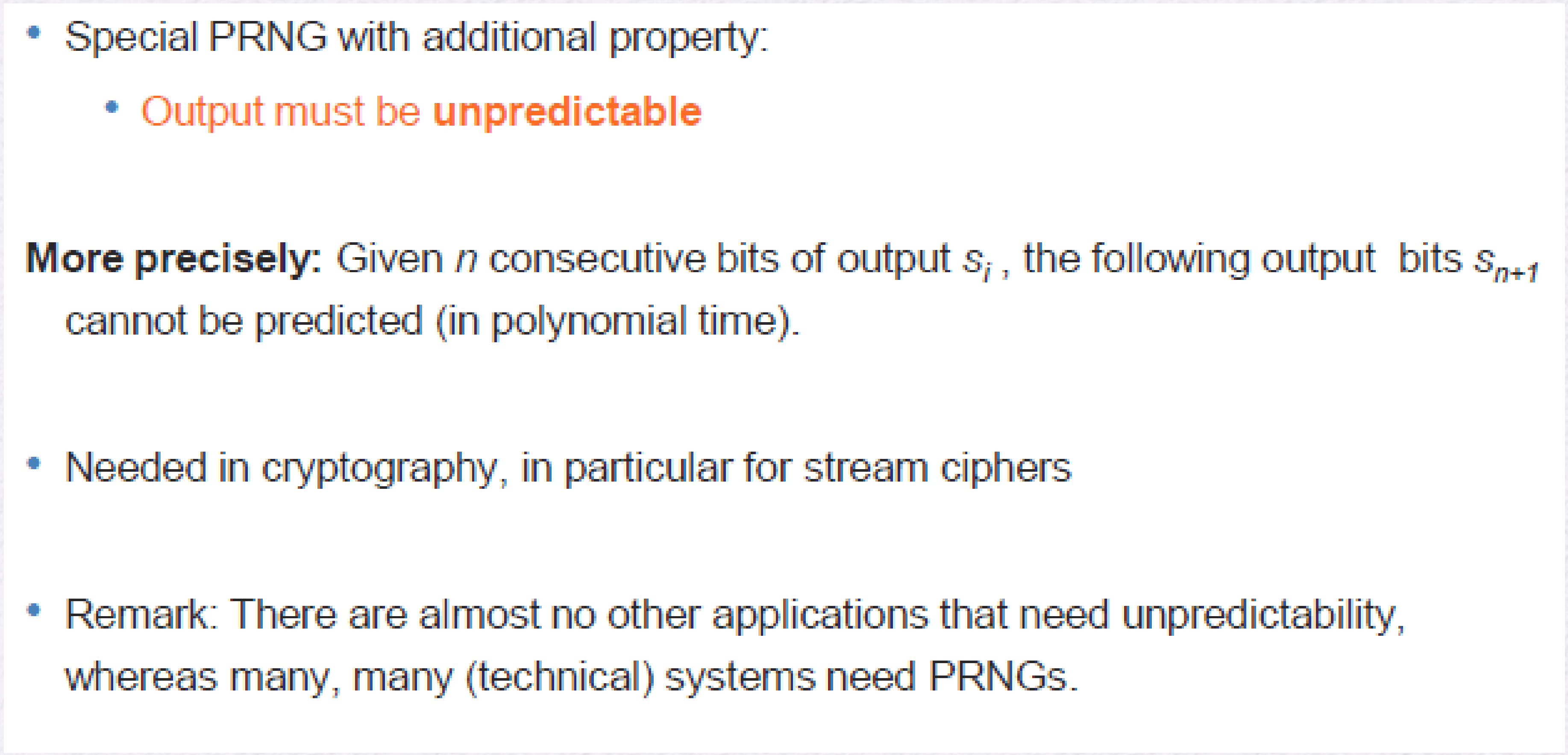
$$S_2 = AS_1 + B \bmod m$$

$$S_3 = AS_2 + B \bmod m$$

…directly reveals A and B. All S_i can be computed easily!

Bad cryptographic properties due to the linearity of most PRNGs

Cryptographically Secure Pseudorandom Number Generator (CSPRNG)



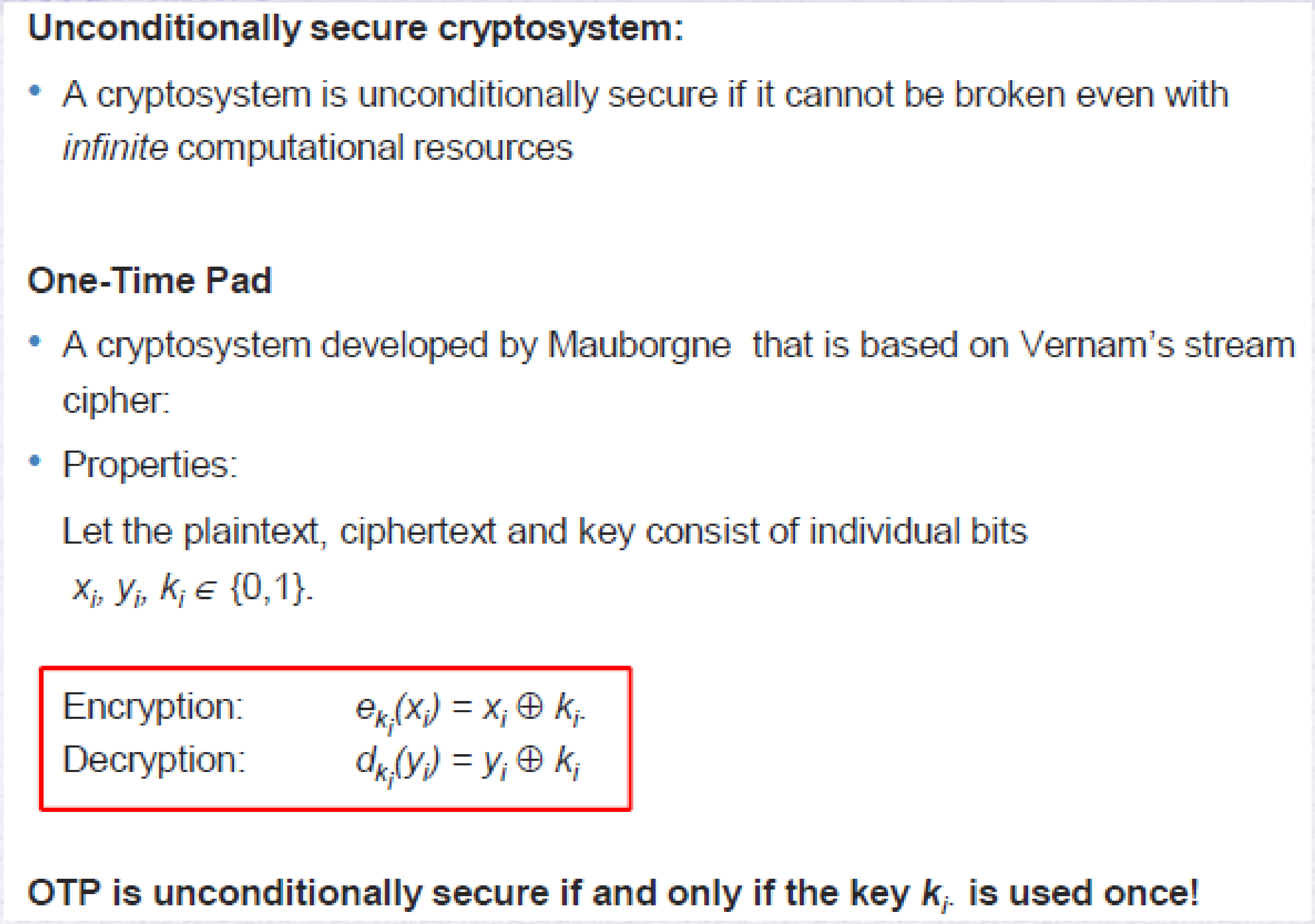
- Special PRNG with additional property:
  - Output must be **unpredictable**

More precisely: Given n consecutive bits of output s_i , the following output bits s_{n+1} cannot be predicted (in polynomial time).

- Needed in cryptography, in particular for stream ciphers

- Remark: There are almost no other applications that need unpredictability, whereas many, many (technical) systems need PRNGs.

One-Time Pad (OTP)



Unconditionally secure cryptosystem:

- A cryptosystem is unconditionally secure if it cannot be broken even with *infinite* computational resources

One-Time Pad

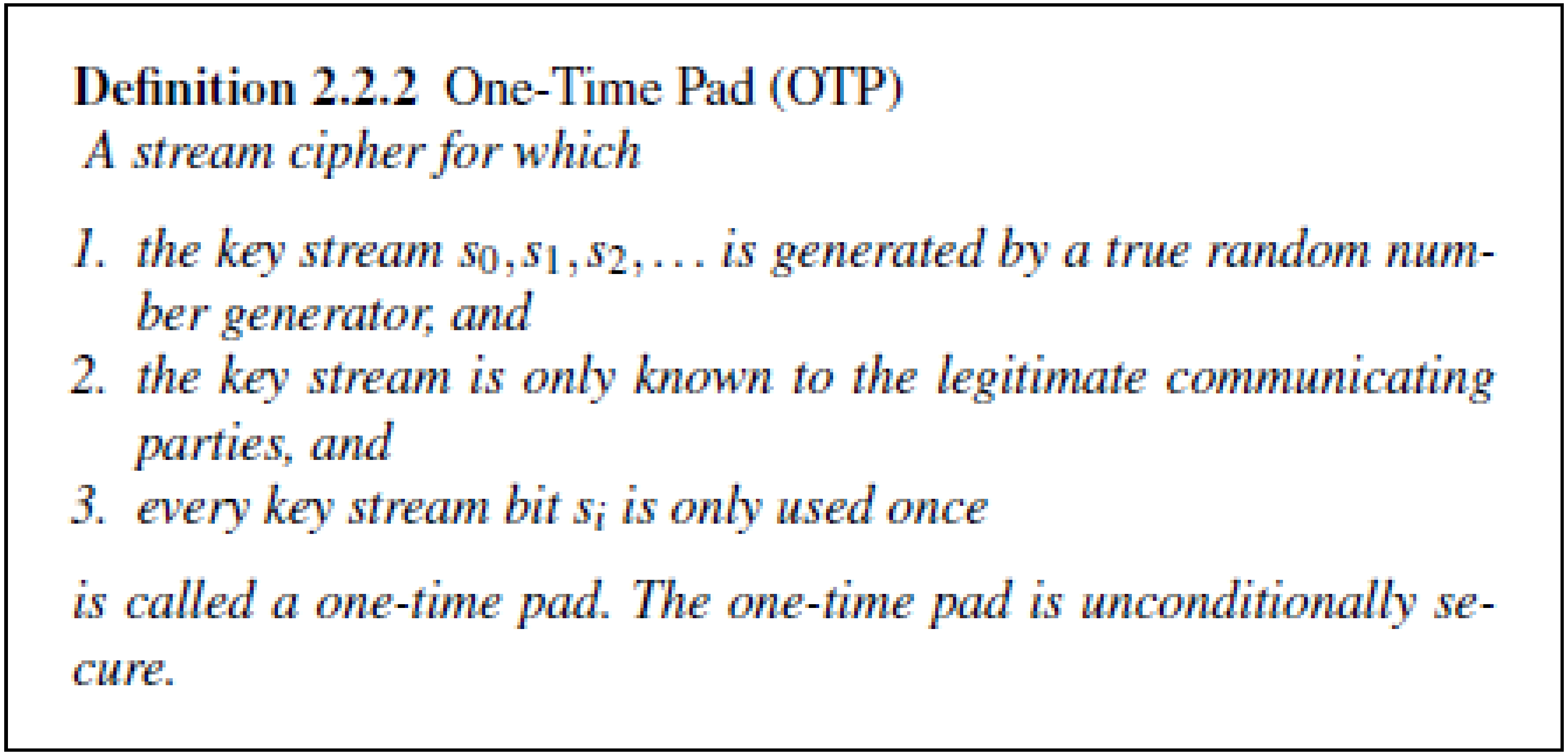
- A cryptosystem developed by Mauborgne that is based on Vernam's stream cipher:
- Properties:

  Let the plaintext, ciphertext and key consist of individual bits $x_i, y_i, k_i \in \{0,1\}$.

Encryption:	$e_{k_i}(x_i) = x_i \oplus k_i.$
Decryption:	$d_{k_i}(y_i) = y_i \oplus k_i$

OTP is unconditionally secure if and only if the key k_i. is used once!

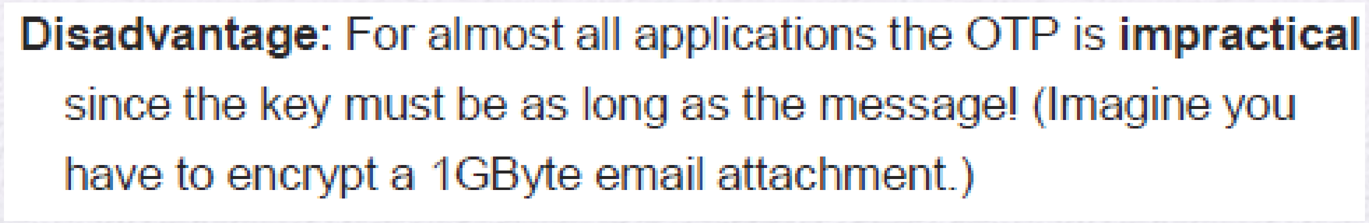
One-Time Pad: Formal Definition



Definition 2.2.2 One-Time Pad (OTP)
A stream cipher for which

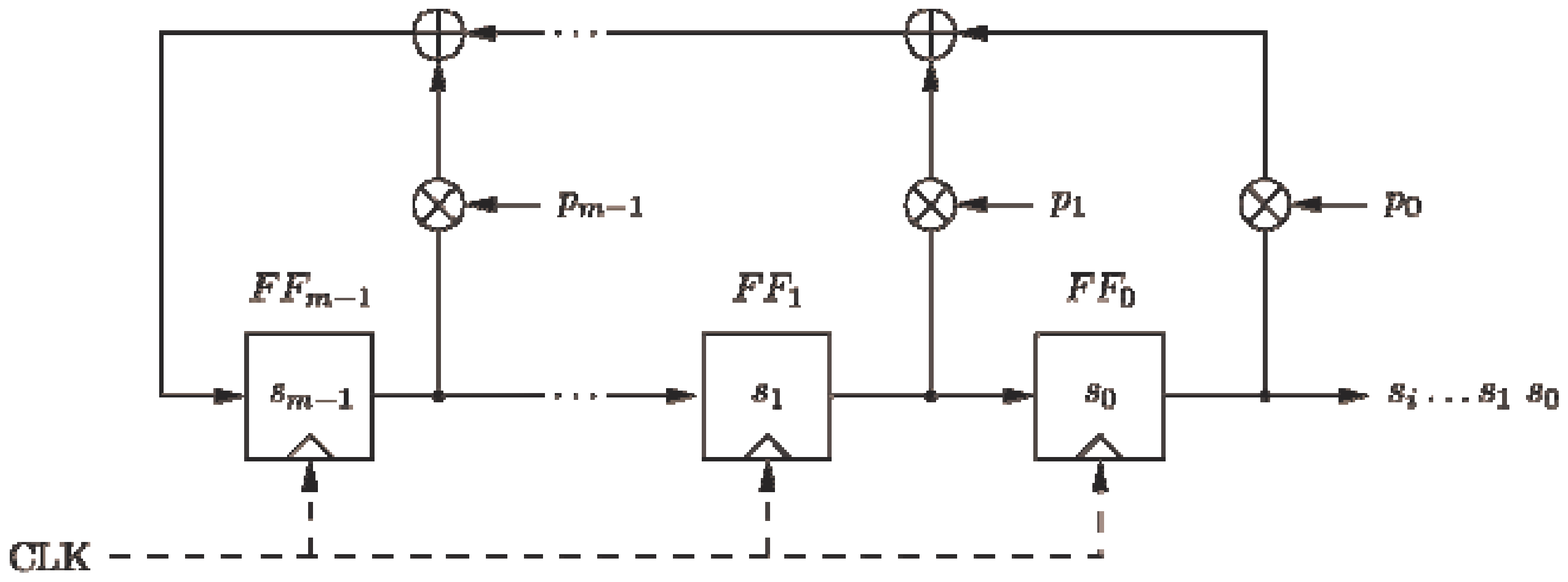
1. *the key stream $s_0, s_1, s_2, \ldots$ is generated by a true random number generator, and*
2. *the key stream is only known to the legitimate communicating parties, and*
3. *every key stream bit s_i is only used once*

is called a one-time pad. The one-time pad is unconditionally secure.



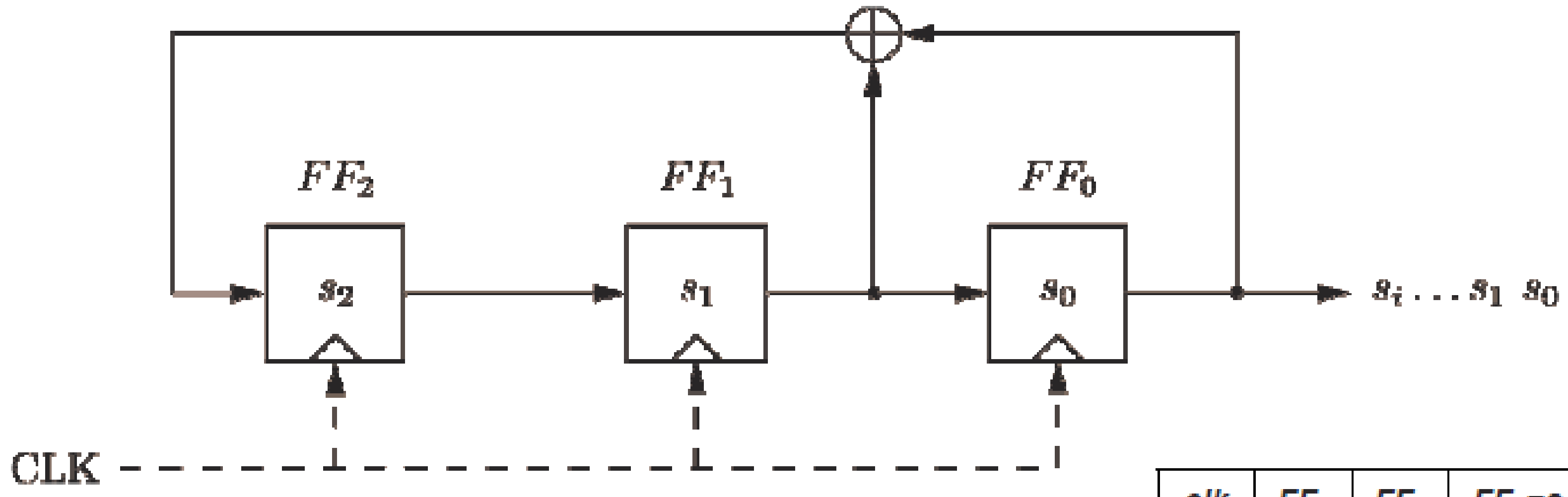
Disadvantage: For almost all applications the OTP is **impractical** since the key must be as long as the message! (Imagine you have to encrypt a 1GByte email attachment.)

Linear Feedback Shift Registers (LFSRs)



- Concatenated *flip-flops (FF)*, i.e., a shift register together with a feedback path
- Feedback computes fresh input by XOR of certain state bits
- *Degree m* given by number of storage elements
- If $p_i = 1$, the feedback connection is present ("closed switch), otherwise there is not feedback from this flip-flop ("open switch")
- Output sequence repeats periodically
- Maximum output length: 2^m-1

LFSR: m=3



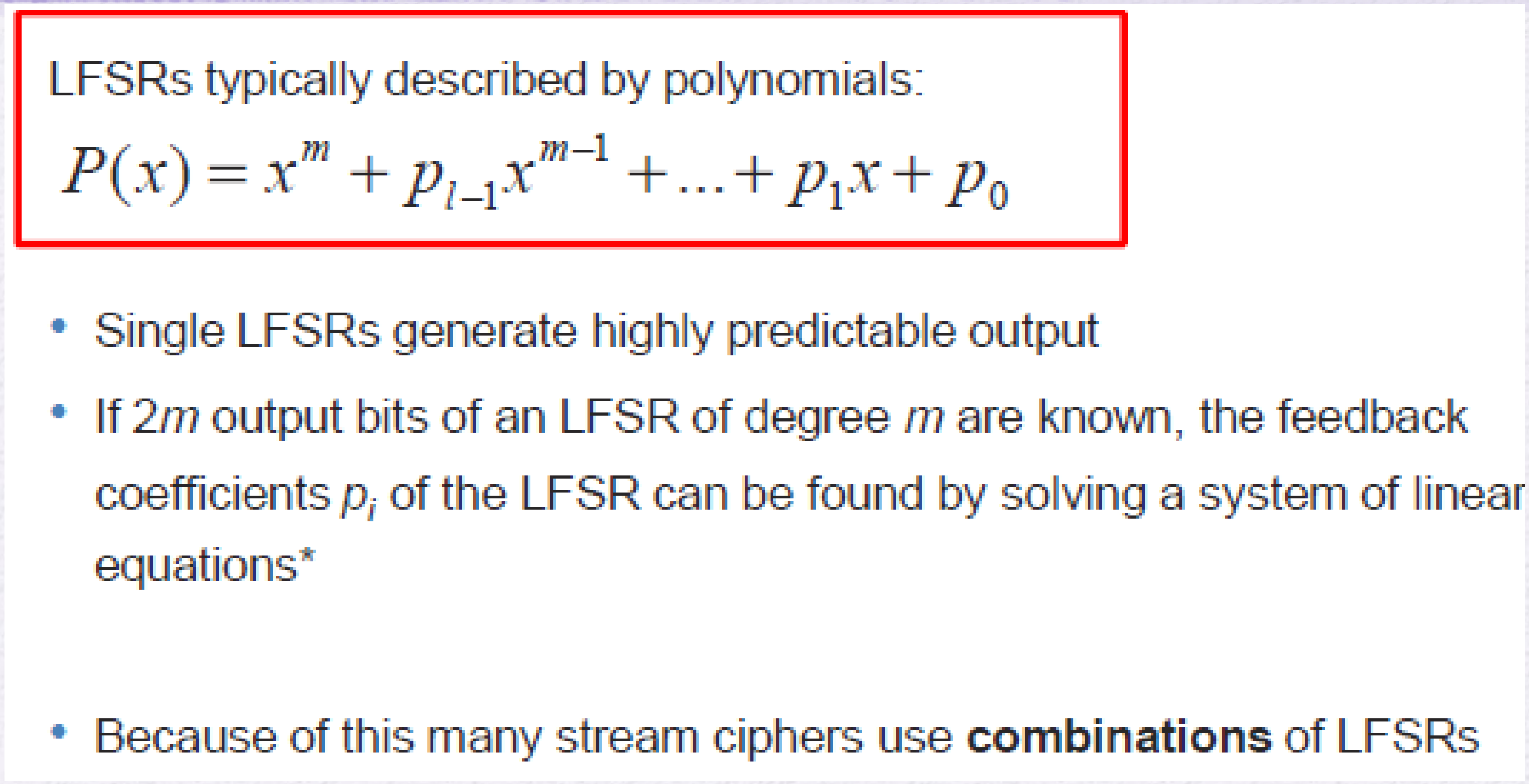
- LFSR output described by recursive equation:

$$s_{i+3} = s_{i+1} + s_i \bmod 2$$

- Maximum output length (of 2^3-1=7) achieved only for certain feedback configurations, .e.g., the one shown here.

clk	*FF_2*	*FF_1*	*FF_0=s_i*
0	1	0	0
1	0	1	0
2	1	0	1
3	1	1	0
4	1	1	1
5	0	1	1
6	0	0	1
7	1	0	0
8	0	1	0

Security of LFSR



LFSRs typically described by polynomials:

$$P(x) = x^m + p_{l-1}x^{m-1} + \ldots + p_1 x + p_0$$

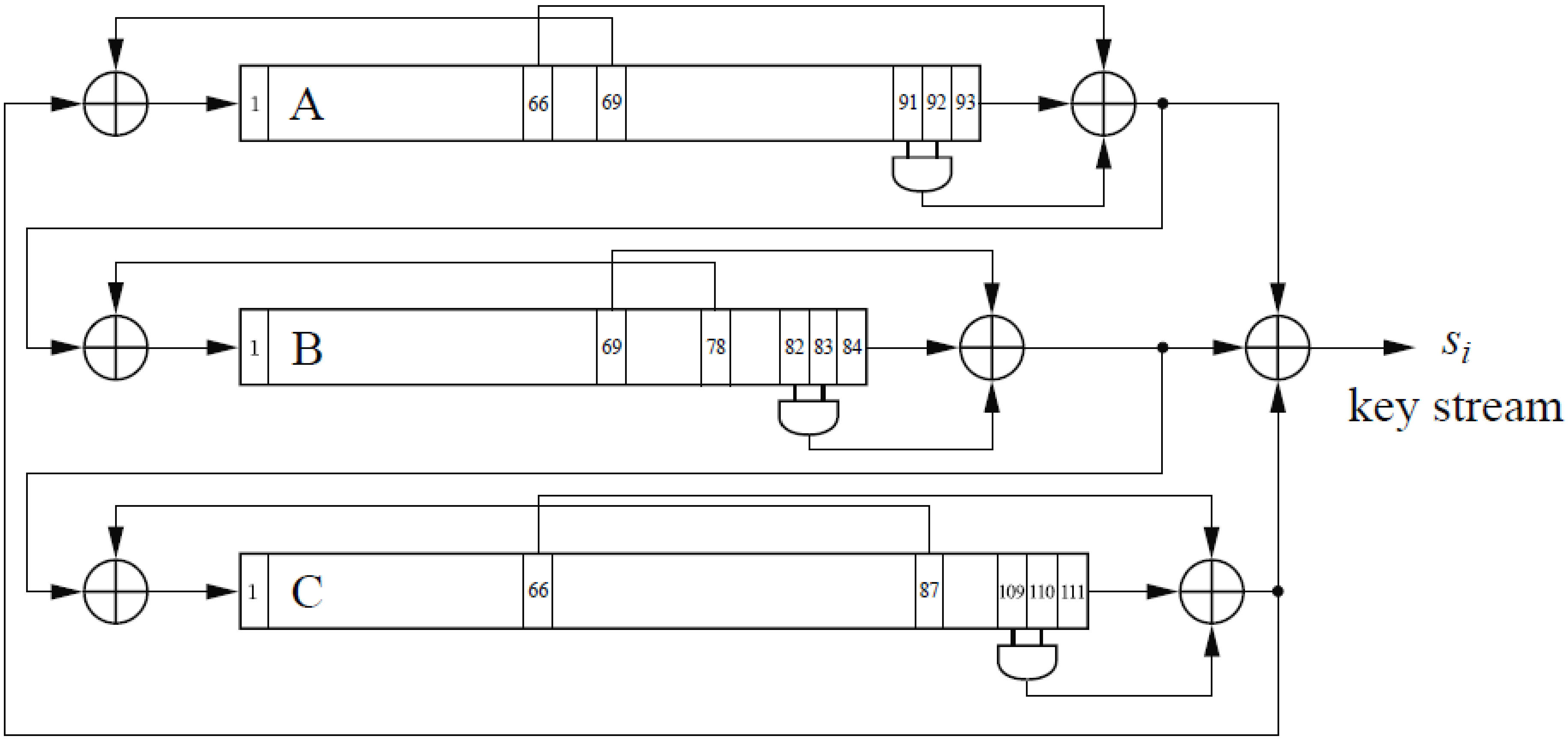
- Single LFSRs generate highly predictable output
- If 2*m* output bits of an LFSR of degree *m* are known, the feedback coefficients p_i of the LFSR can be found by solving a system of linear equations*
- Because of this many stream ciphers use **combinations** of LFSRs

Stream Cipher Example: Trivium

- Trivium is a stream cipher which uses an 80-bit key
- It is based on a combination of three shift registers.
- Even though these are feedback shift registers, there are nonlinear components used to derive the output of each register, unlike the LFSRs.

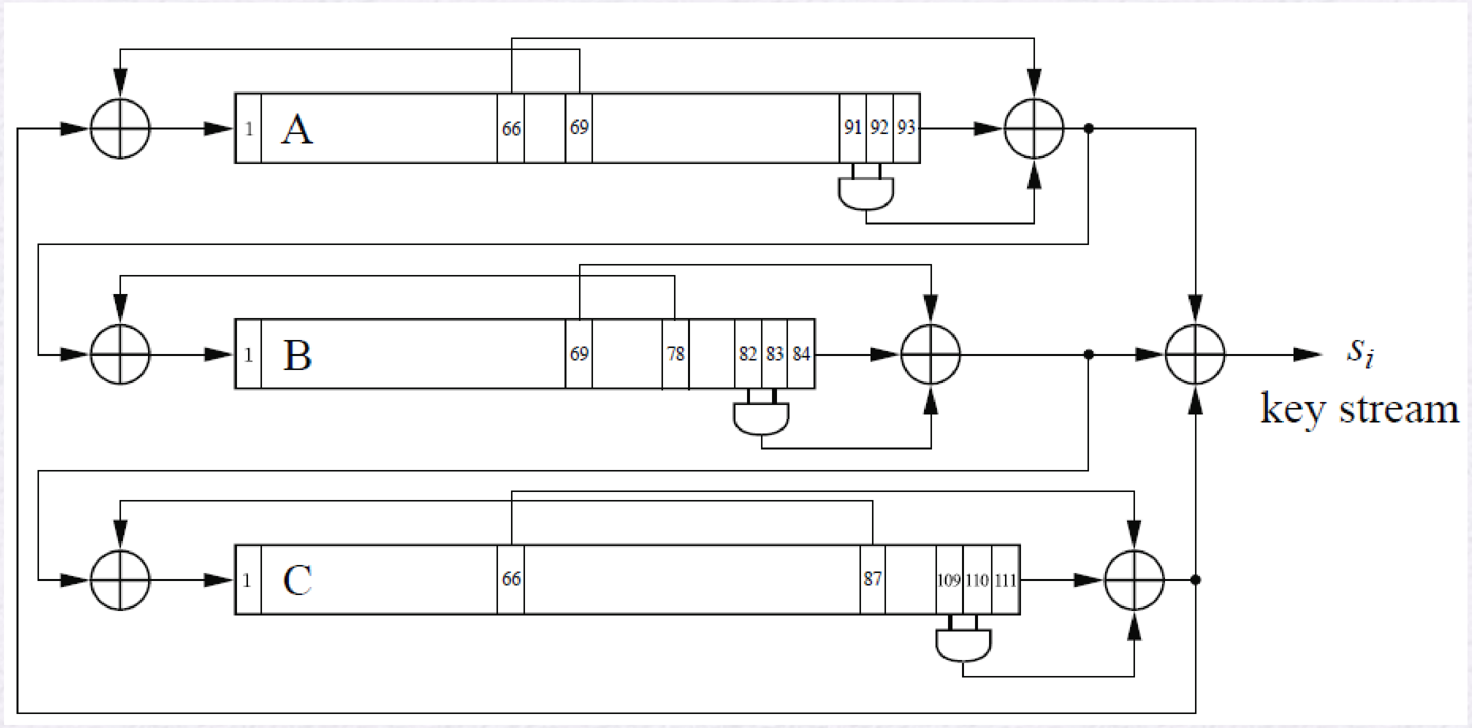
Trivium Cipher

- 3 shift registers: A (93-bit), B(84-bit) and C (111-bit)
- S_i is the XOR-SUM of all three registers output
- The output of each register is connect to the input of another register



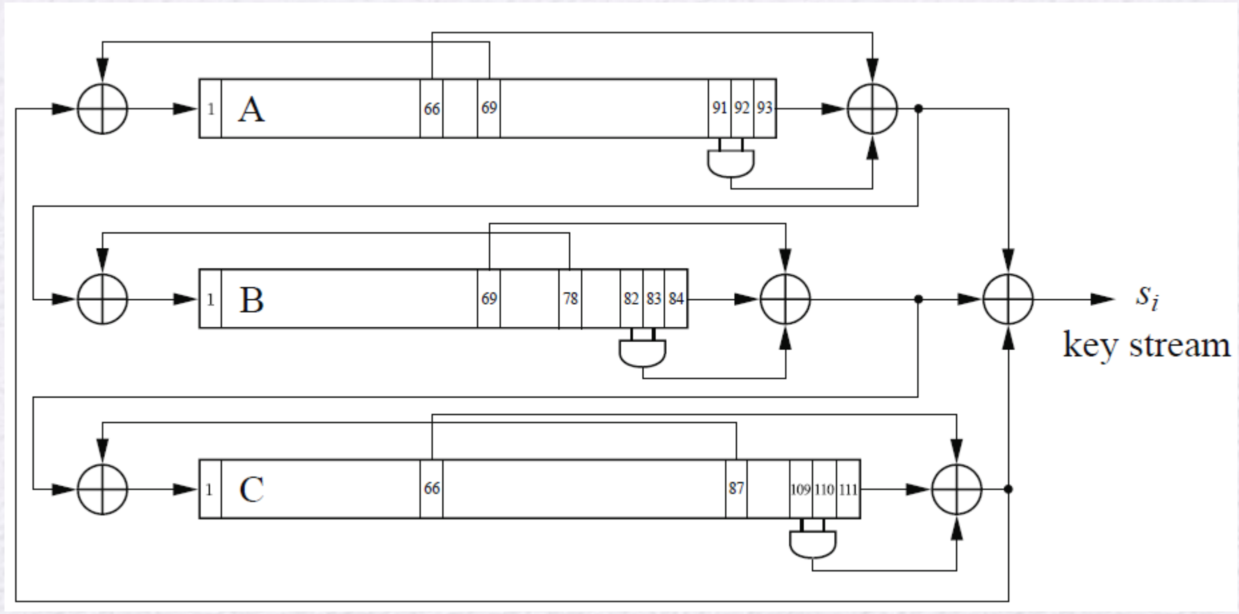
Trivium Cipher

- The output of each register is connect to the input of another register
- The cipher can be viewed as circular register with a length of 288 bits
- The input of each register is computed as the XOR-sum of two bits: the output of another register and one bit in specific location



Trivium Cipher: Encryption

- Initialization:
  - an 80-bit IV (initialization vector) is loaded into the 80 leftmost locations of registerA; and an 80-bit key is loaded in the 80 leftmost locations of register B.
  - All other register bits are set to zero with the exception of the three rightmost bits of register C, i.e., bits c_{109}, c_{110} and c_{111}, which are set to 1.
- Warm-up: he cipher is clocked 4×288 = 1152 times. No cipher output is generated.
- Encryption phase starts:
  - At cycle 1153



More Info Slides

Stream Cipher

Encrypts a digital data stream one bit or one byte at a time

Examples:

- Autokeyed Vigenère cipher
- Vernam cipher

In the ideal case, a one-time pad version of the Vernam cipher would be used, in which the keystream is as long as the plaintext bit stream

If the cryptographic keystream is random, then this cipher is unbreakable by any means other than acquiring the keystream

- **Keystream** must be provided to both users in advance via some independent and secure channel
- **This introduces insurmountable** logistical problems **if the intended data traffic is very large**

For **practical reasons the bit-stream generator** must be implemented as an algorithmic procedure so that the cryptographic bit stream can be produced by both users

It must be computationally impractical to predict future portions of the bit stream based on previous portions of the bit stream

The two users need only share the generating key and each can produce the keystream

Block Cipher

A block of plaintext is treated as a whole and used to produce a ciphertext block of equal length

Typically a block size of 64 or 128 bits is used

As with a stream cipher, the two users share a symmetric encryption key

The majority of network-based symmetric cryptographic applications make use of block ciphers