Stream encryption
Hash function

Cryptography: course for master's degree in EDGE COMPUTING

Michał Melosik, PhD

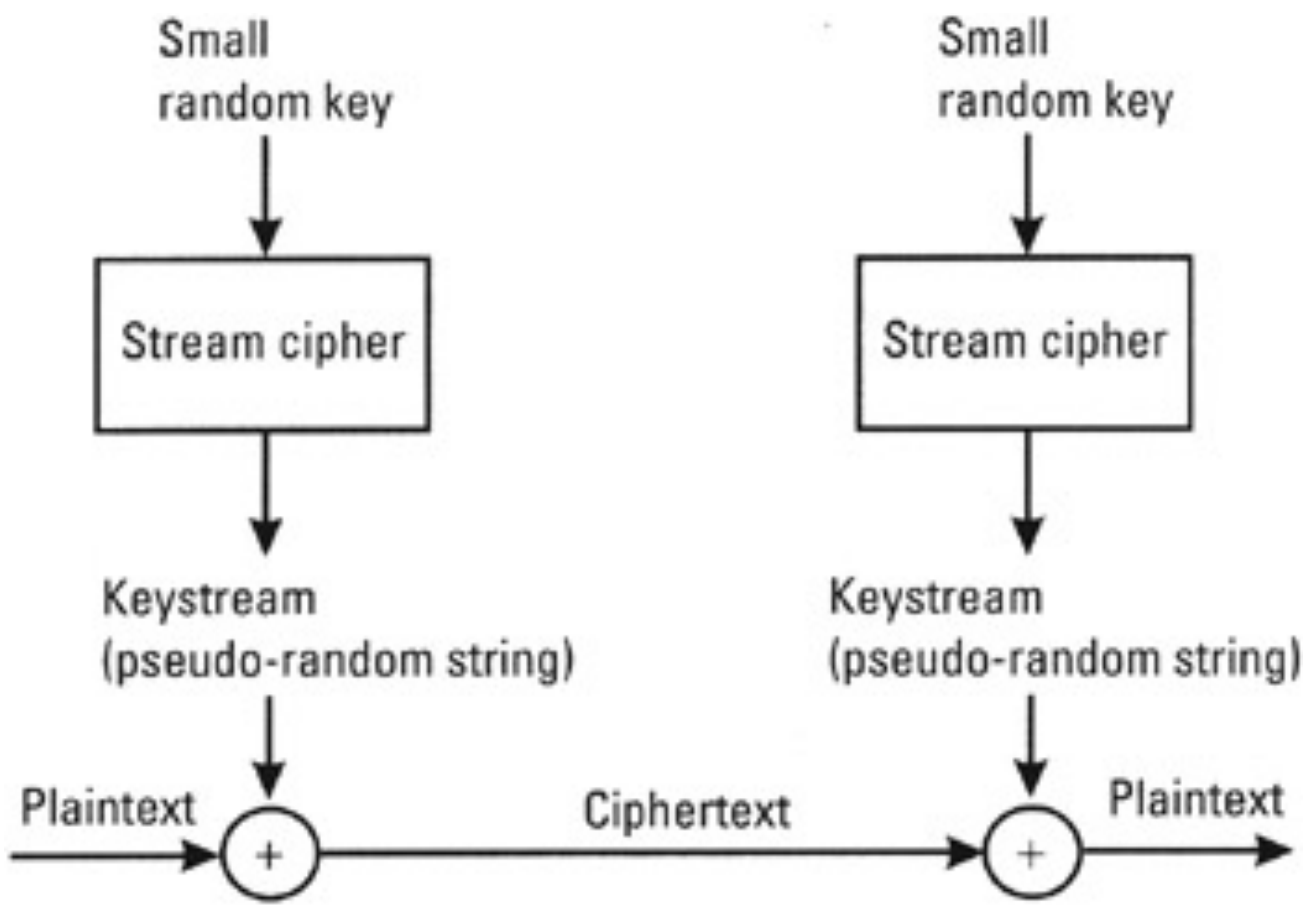
Lecture outline

1. **Stream cipher**
2. **Block cipher**
3. **OTP**
4. **Stream ciphers built on chaotic generators - some practical aspects of cryptography**
5. **Hash**
6. **Discussion**

Stream vs block encryption

Stream cipher

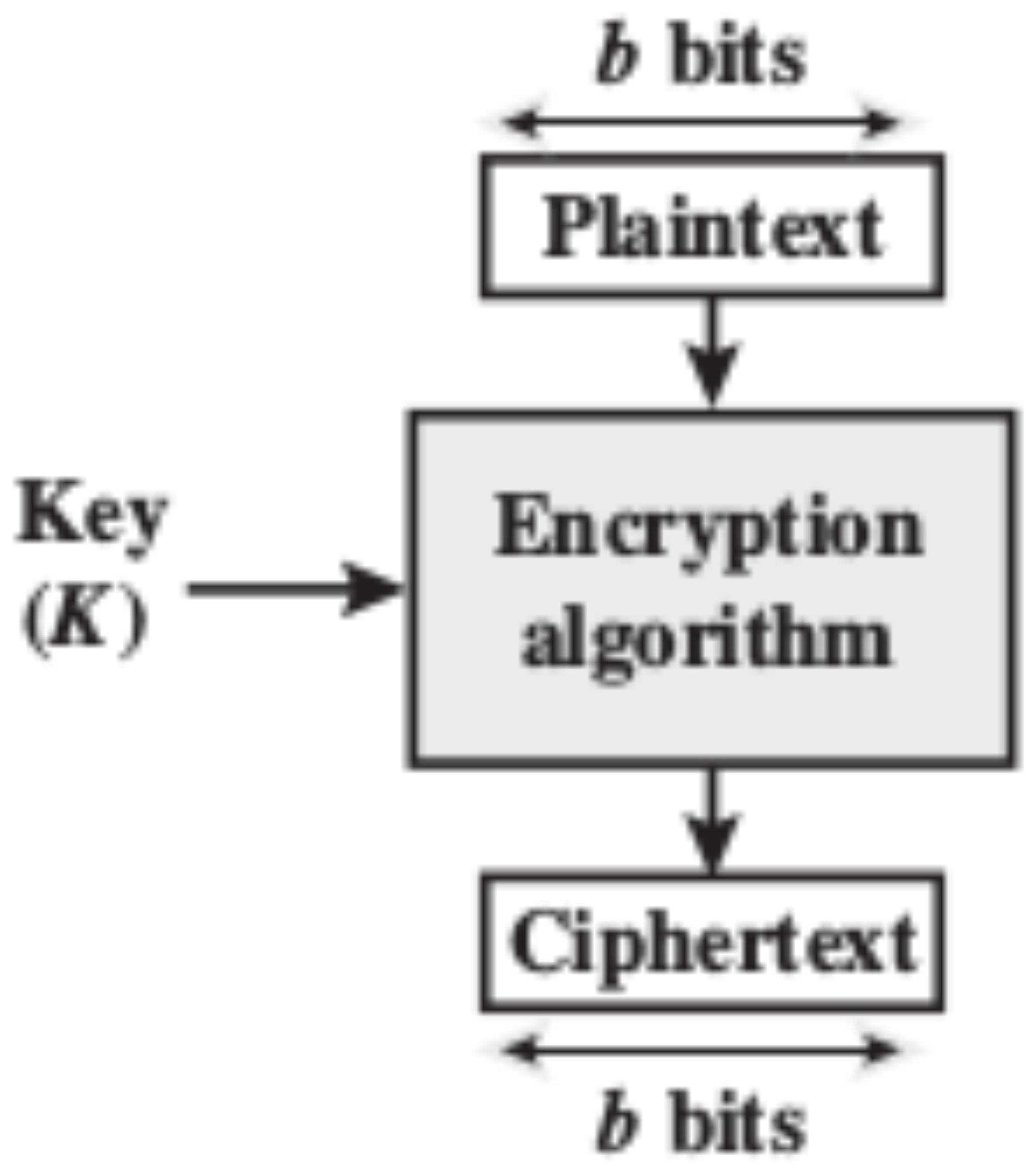
Encryption and decryption process



Source (12.10.22): https://crypto.stackexchange.com/questions/33100/does-a-stream-cipher-provide-perfect-secrecy

Block cipher

Encryption and decryption process



Question to discuss with students at lecture:

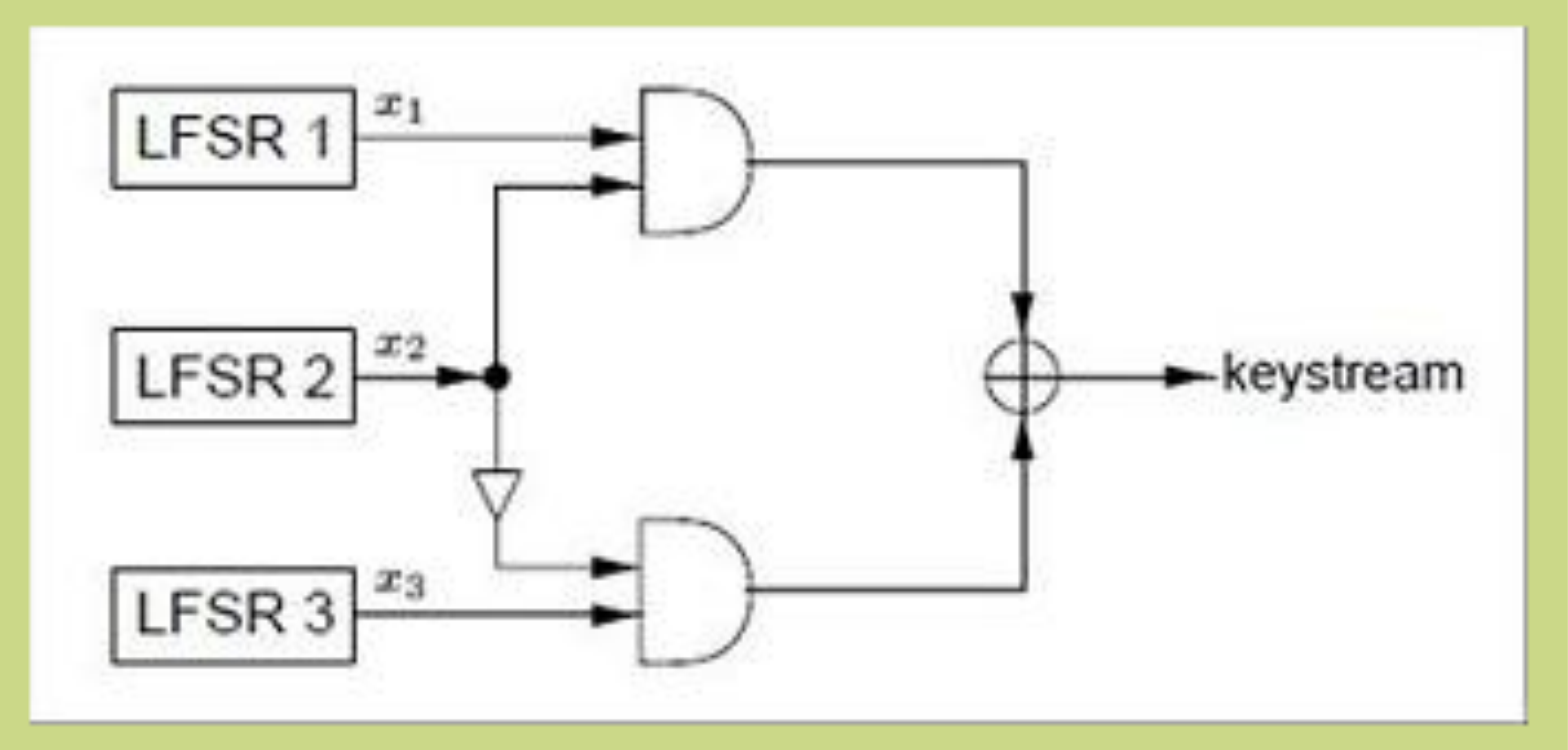
When considering these two types of encryption, how to approach:

1. Implementation?
2. Testing?
3. Application usability?
4. Security?

Source (12.10.22): https://techblogmu.blogspot.com/2018/05/what-are-block-ciphers-explain-with-examples-the-ecb-and-cbc-modes-of-block-ciphers.html

Selected examples of stream ciphers

Geffe

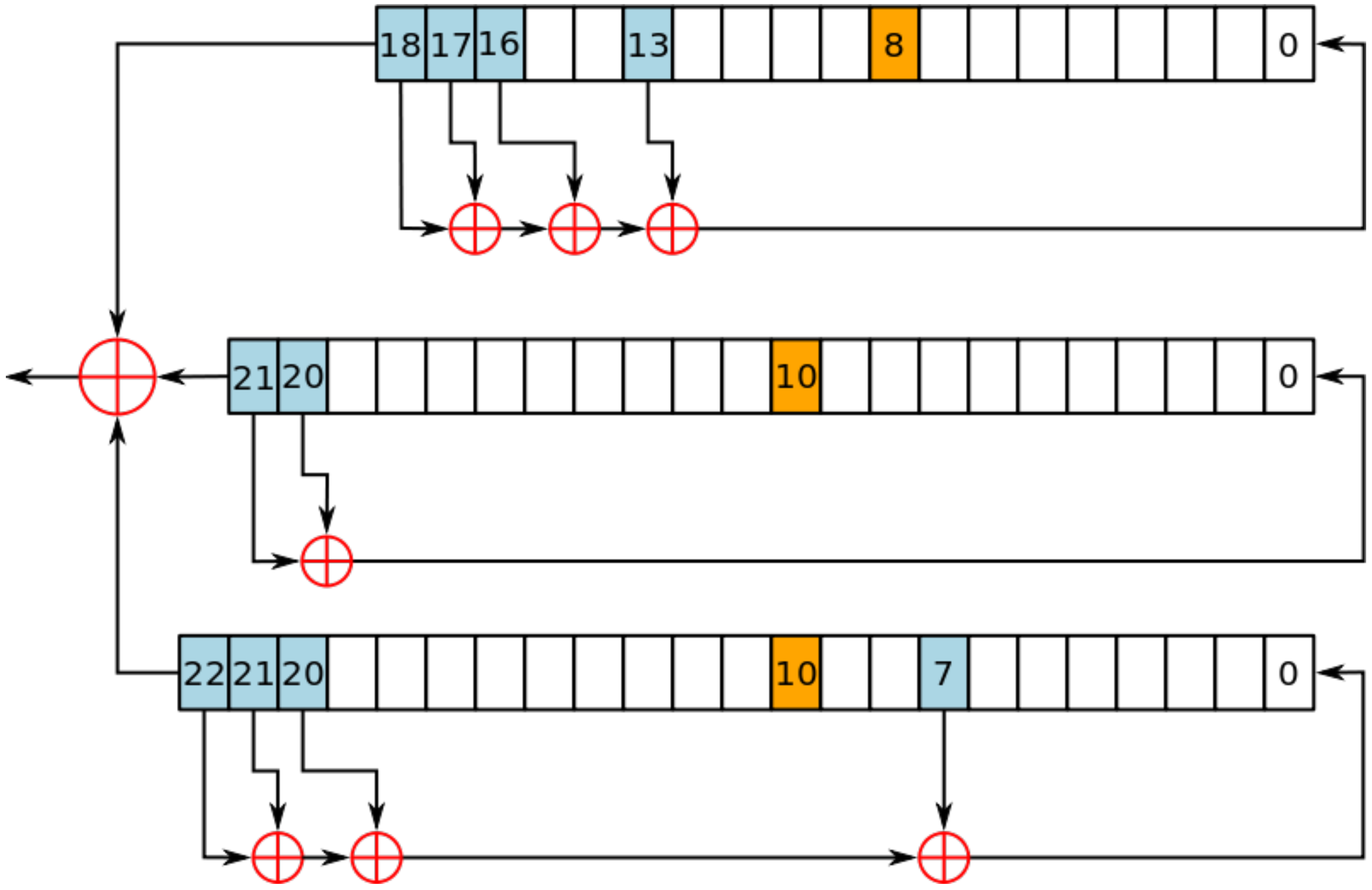


Source (12.10.22): http://emmanuel.pouly.free.fr/cipher1.html

The critical for security is to correctly seed and re-seed LFSR modules.

Selected examples of stream ciphers

A5/1



(Source 13.10.22): https://en.wikipedia.org/wiki/A5/1

The critical for security is to correctly seed and re-seed of all register.

One Time Pad

One Time Pad

The most secure cryptographic approach

Vernam (1917)

Key:	0	1	0	1	1	1	0	0	1	0
Plaintext:	1	1	0	0	0	1	1	0	0	0
$\oplus$										
Ciphertext:	1	0	0	1	1	0	1	0	1	0

Discussion with students at lecture:

What can be stated about:

1. Advantages?
2. Disadvantages?
3. Security?

More precisely: $m, c, k \in \{0,1\}^n$

$$c = E(k, m) = k \oplus m \quad , \quad D(k, c) = k \oplus c$$

Indeed, for all k, m: $D(k, E(k, m)) = k \oplus (k \oplus m) = m$

Source (12.10.22): https://www.warrencodes.com/ciphers

One Time Pad

The most secure cryptographic approach

Vernam (1917)

Key:	0	1	0	1	1	1	0	0	1	0	$\oplus$
Plaintext:	1	1	0	0	0	1	1	0	0	0	
Ciphertext:	1	0	0	1	1	0	1	0	1	0	

No Force to Brute Force

More precisely: $m, c, k \in \{0,1\}^n$

$$c = E(k, m) = k \oplus m \quad , \quad D(k, c) = k \oplus c$$

Indeed, for all k, m: $D(k, E(k, m)) = k \oplus (k \oplus m) = m$

Source (12.10.22): https://www.warrencodes.com/ciphers

Stream ciphers based on chaotic systems

Chaos vs deterministic chaos

What does wikipedia have to say about it?

Chaos theory

From Wikipedia, the free encyclopedia

For other uses, see Chaos theory (disambiguation) and Chaos (disambiguation).

Chaos theory is an interdisciplinary area of scientific study and branch of mathematics focused on underlying patterns and deterministic laws of dynamical systems that are highly sensitive to initial conditions, and were once thought to have completely random states of disorder and irregularities.[1] Chaos theory states that within the apparent randomness of chaotic complex systems, there are underlying patterns, interconnection, constant feedback loops, repetition, self-similarity, fractals, and self-organization.[2] The butterfly effect, an underlying principle of chaos, describes how a small change in one state of a deterministic nonlinear system can result in large differences in a later state (meaning that there is sensitive dependence on initial conditions).[3] A metaphor for this behavior is that a butterfly flapping its wings in Brazil can cause a tornado in Texas.[4][5][6]

Small differences in initial conditions, such as those due to errors in measurements or due to rounding errors in numerical computation, can yield widely diverging outcomes for such dynamical systems, rendering long-term prediction of their behavior impossible in general.[7] This can happen even though these systems are deterministic, meaning that their future behavior follows a unique evolution[8] and is fully determined by their initial conditions, with no random elements involved.[9] In other words, the deterministic nature of these systems does not make them predictable.[10][11] This behavior is known as **deterministic chaos**, or simply **chaos**. The theory was summarized by Edward Lorenz as:[12]

> Chaos: When the present determines the future, but the approximate present does not approximately determine the future.

Butterfly effect

What does wikipedia have to say about it?

Butterfly effect

From Wikipedia, the free encyclopedia

For other uses, see Butterfly effect (disambiguation).

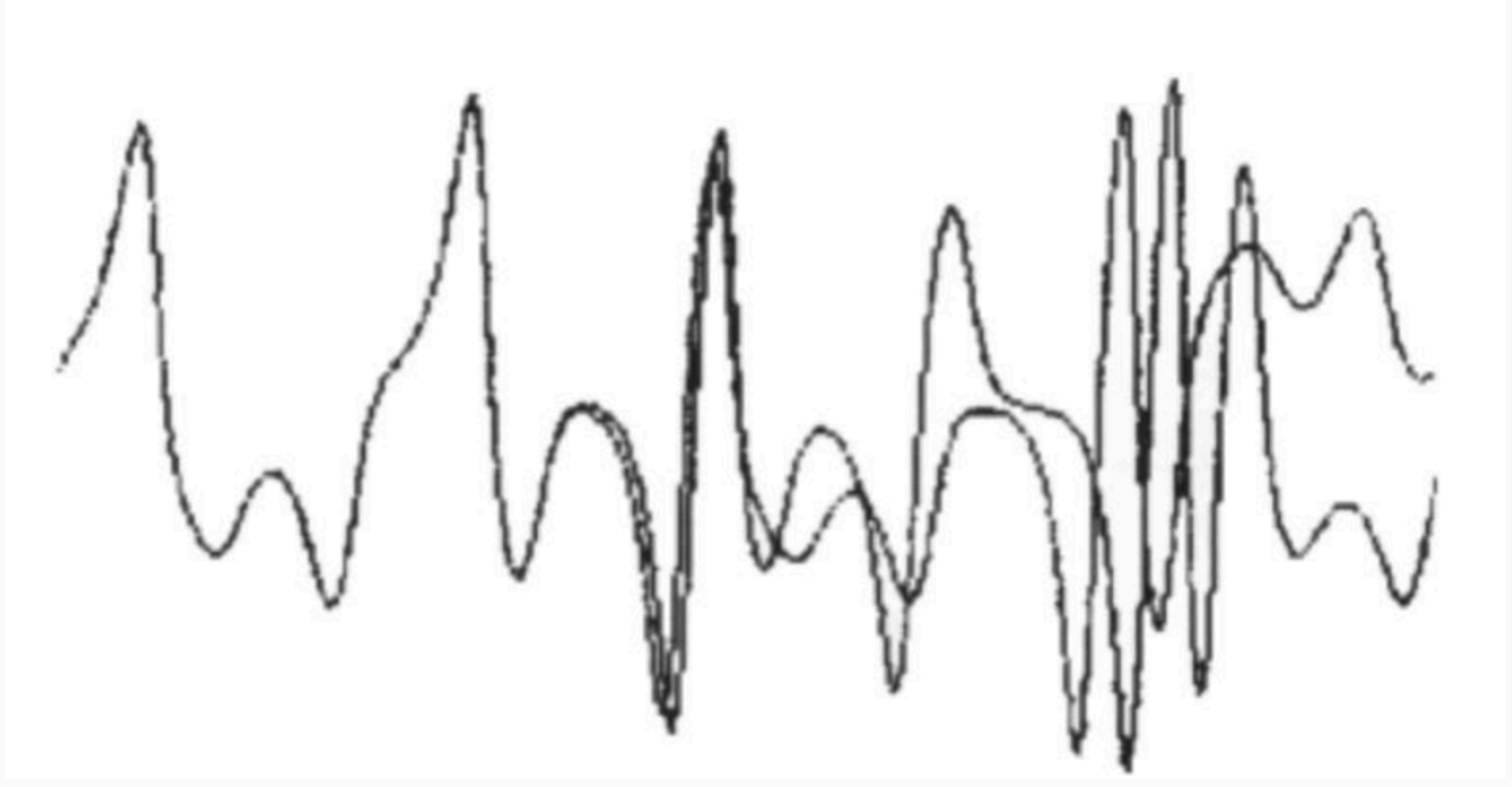
In chaos theory, the **butterfly effect** is the sensitive dependence on initial conditions in which a small change in one state of a deterministic nonlinear system can result in large differences in a later state.

The term is closely associated with the work of mathematician and meteorologist Edward Norton Lorenz. He noted that the butterfly effect is derived from the metaphorical example of the details of a tornado (the exact time of formation, the exact path taken) being influenced by minor perturbations such as a distant butterfly flapping its wings several weeks earlier. Lorenz originally used a seagull causing a storm but was persuaded to make it more poetic with the use of butterfly and tornado by 1972.[1][2] He discovered the effect when he observed runs of his weather model with initial condition data that were rounded in a seemingly inconsequential manner. He noted that the weather model would fail to reproduce the results of runs with the unrounded initial condition data. A very small change in initial conditions had created a significantly different outcome.[3]

The application of chaotic systems in cryptography

In stream ciphers

The Butterfly Effect, also known as deterministic chaos, is a phenomenon where equations with no... [+] PUBLIC DOMAIN



Two systems starting from an identical configuration, but with imperceptibly small differences in... [+] LARRY BRADLEY

Source (12.10.22): https://www.forbes.com/sites/startswithabang/2018/02/13/chaos-theory-the-butterfly-effect-and-the-computer-glitch-that-started-it-all/?sh=5c26d47569f6

The application of chaotic systems in cryptography

Continuous-time chaotic system - Lorenz system

$$\frac{\mathrm{d}x}{\mathrm{d}t} = \sigma(y - x),$$

$$\frac{\mathrm{d}y}{\mathrm{d}t} = x(\rho - z) - y,$$

$$\frac{\mathrm{d}z}{\mathrm{d}t} = xy - \beta z.$$

Analogy to a pseudo-random generator.

Security-critical is the variety of seed=initial conditions.

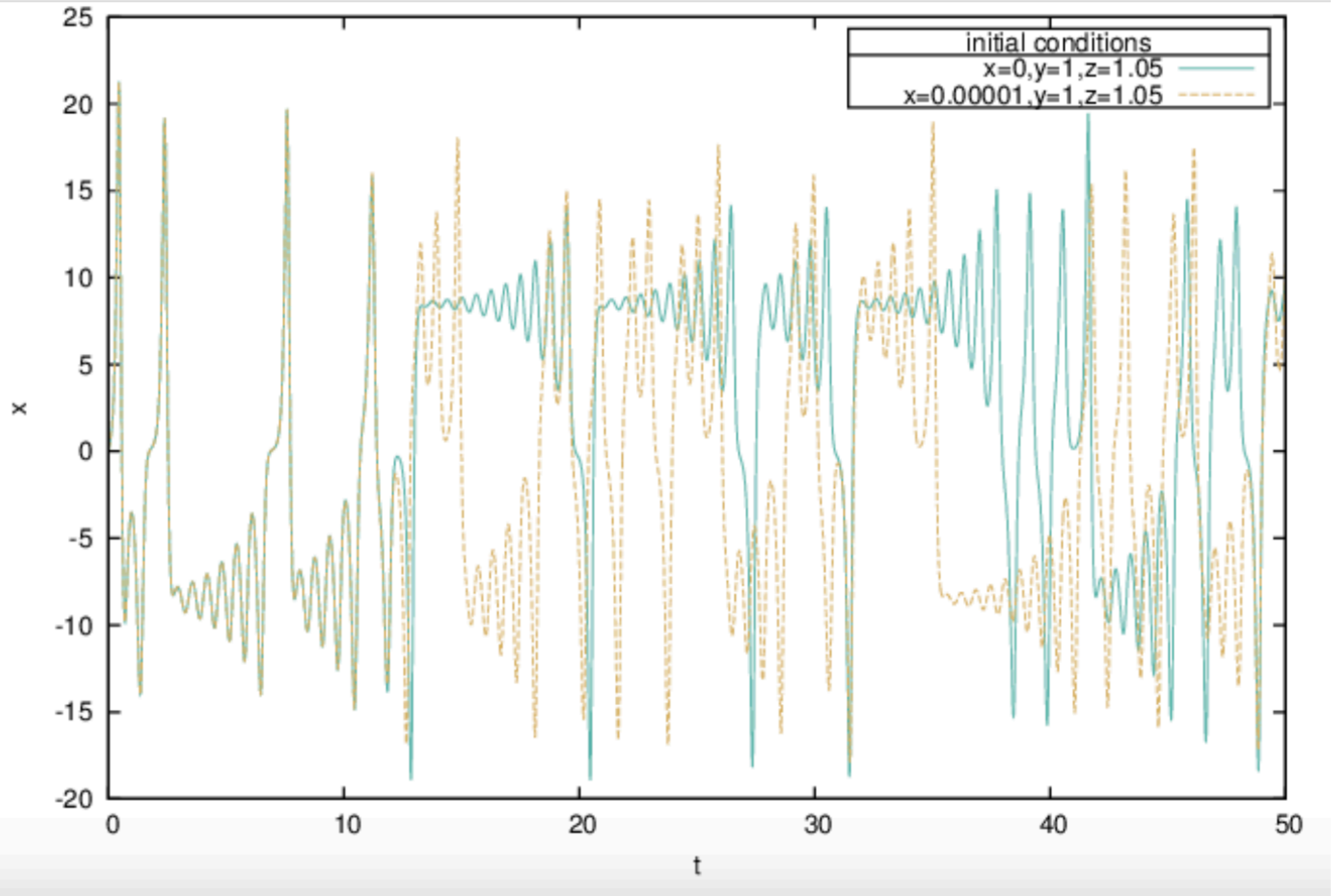
The application of chaotic systems in cryptography

Continuous-time chaotic system - Lorenz system

$$\frac{\mathrm{d}x}{\mathrm{d}t} = \sigma(y - x),$$

$$\frac{\mathrm{d}y}{\mathrm{d}t} = x(\rho - z) - y,$$

$$\frac{\mathrm{d}z}{\mathrm{d}t} = xy - \beta z.$$



Source: Gutierrez, Tomas Navarrete. A control architecture for complex systems, based on multi-agent simulation. Diss. Université de Lorraine, 2012.

Chaotic stream cipher - real case

Chaotic stream cipher

Logistic map as a PRBG

Hindawi
Complexity
Volume 2017, Article ID 8692046, 21 pages
https://doi.org/10.1155/2017/8692046

WILEY | Hindawi

Research Article

Finite Precision Logistic Map between Computational Efficiency and Accuracy with Encryption Applications

Wafaa S. Sayed,[1] Ahmed G. Radwan,[1,2] Ahmed A. Rezk,[2] and Hossam A. H. Fahmy[3]

[1] *Engineering Mathematics and Physics Department, Faculty of Engineering, Cairo University, Giza 12613, Egypt*
[2] *Nanoelectronics Integrated Systems Center, Nile University, Cairo 12588, Egypt*
[3] *Electronics and Communication Engineering Department, Faculty of Engineering, Cairo University, Giza 12613, Egypt*

Chaotic stream cipher

Logistic map as a PRBG - implementation

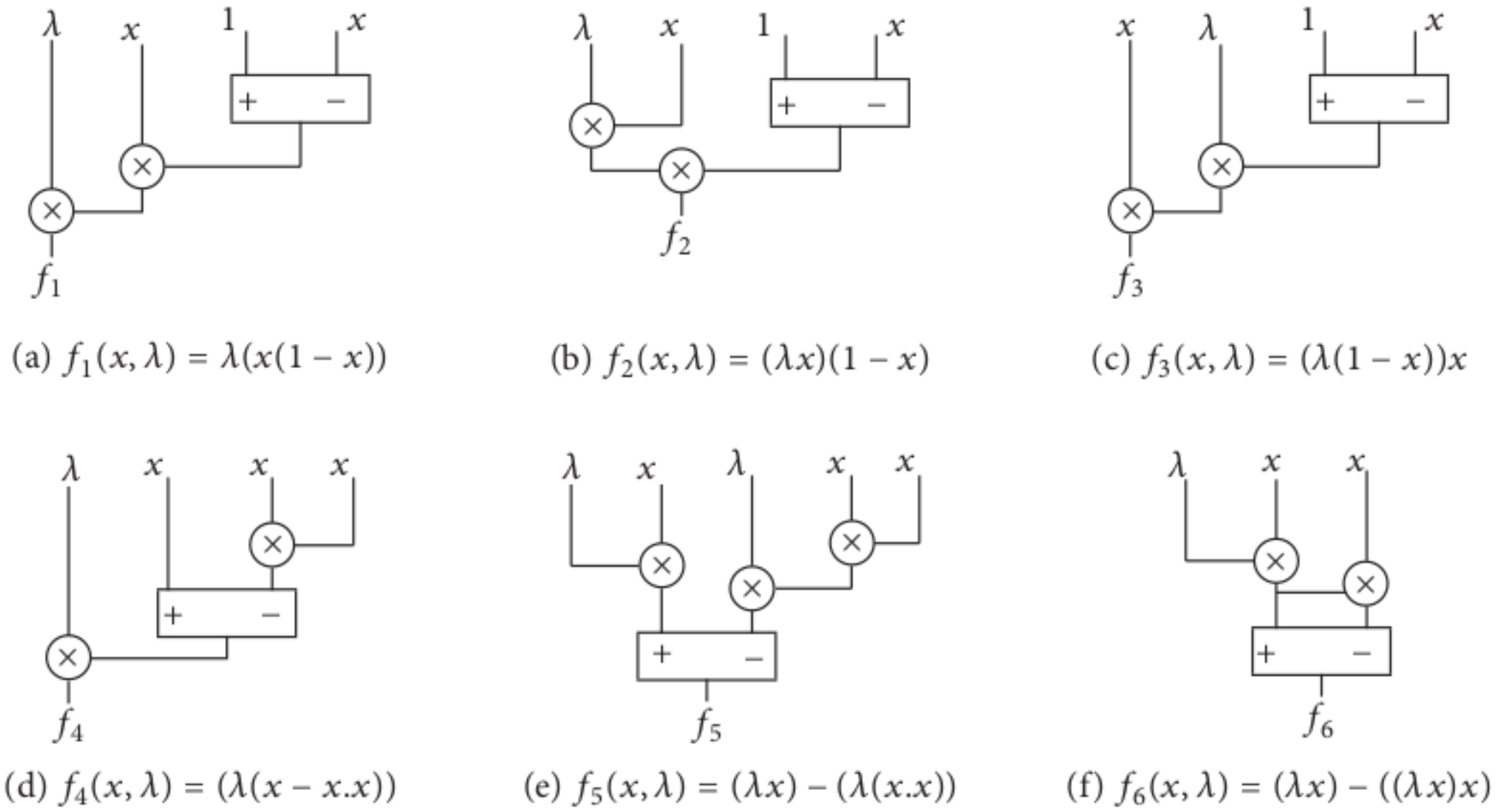


FIGURE 2: Six different maps in fixed-point arithmetic.

Chaotic stream cipher

Logistic map as a stream cipher module

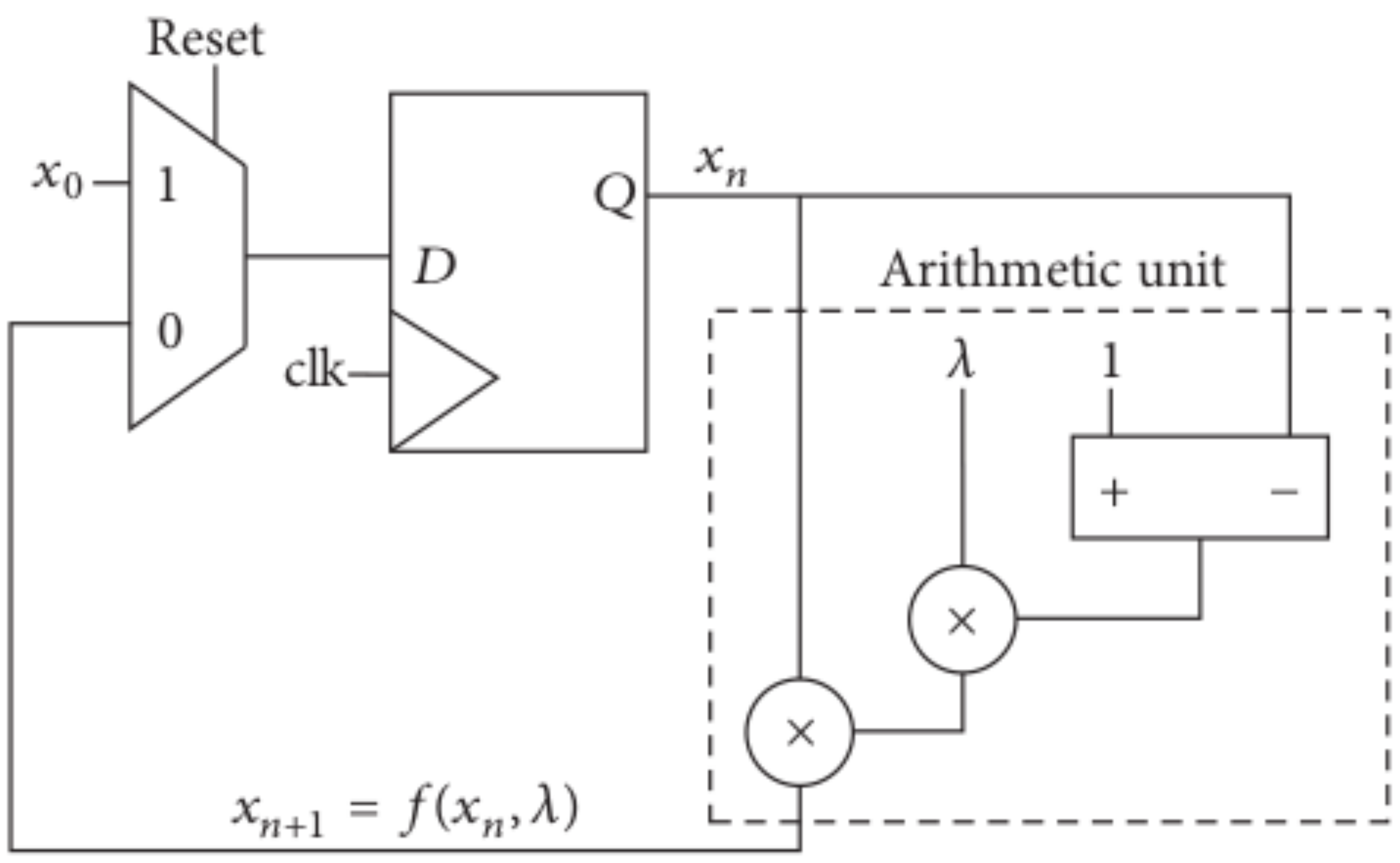


FIGURE 18: Hardware realization of Pseudo-Random Number Generator.

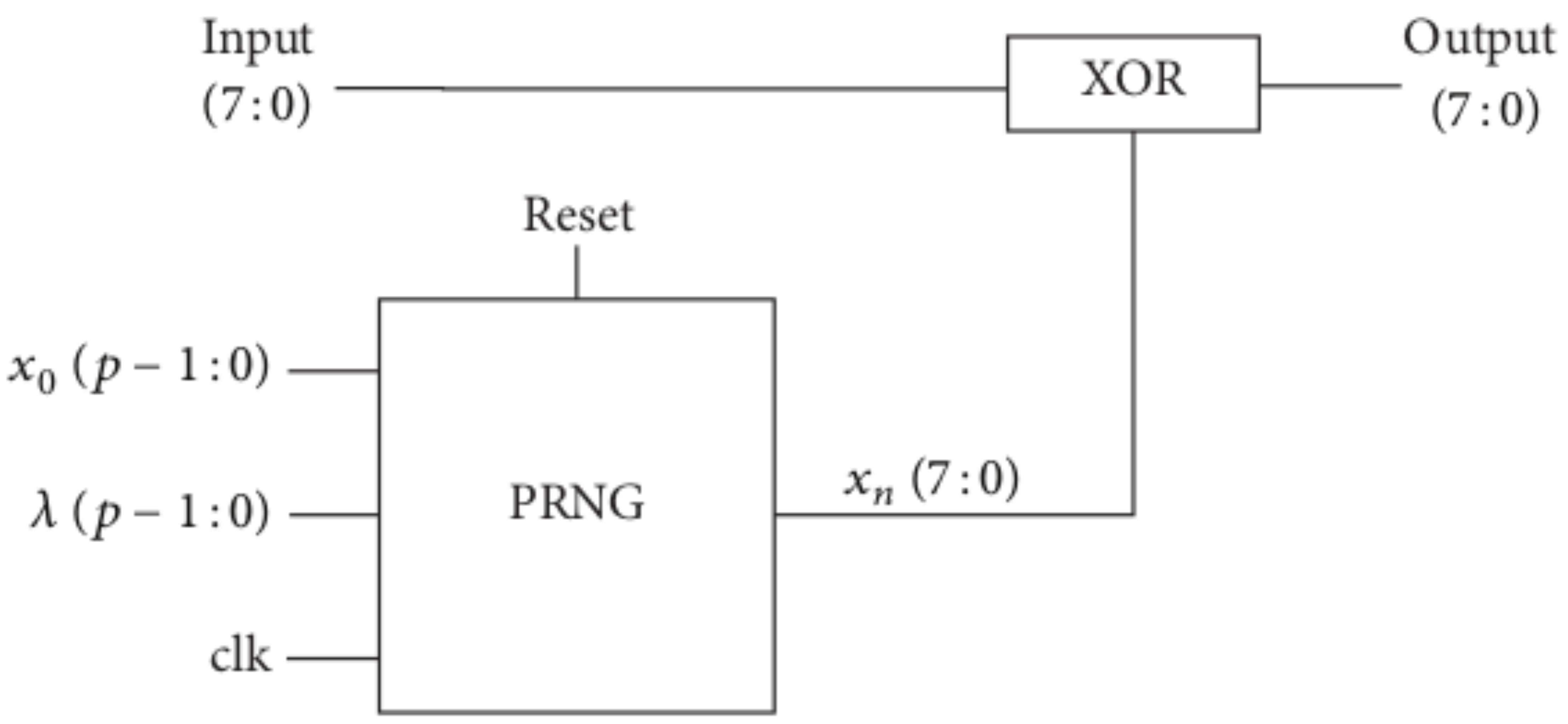
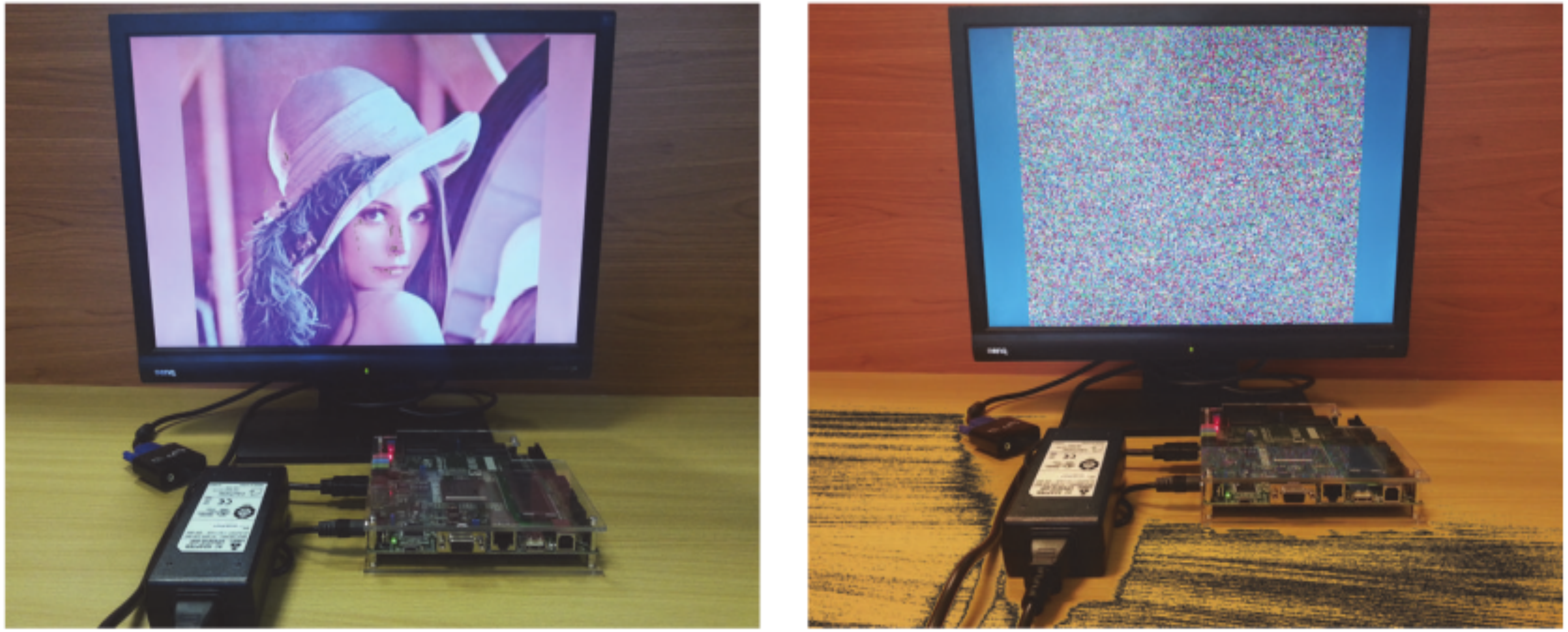


FIGURE 19: Stream cipher system for encryption applications.

Chaotic stream cipher

Logistic map as a stream cipher module - real case



(a) (b)

FIGURE 21: Standalone image encryption system. (a) Decrypted image. (b) Encrypted image.

Chaotic stream cipher

Logistic map as a stream cipher module - security assessment

Table 4: NIST results for different bus sizes.

NIST test (sample 1000000 bits in length)	8 bits	27 bits	34 bits	36 bits	38 bits	40 bits	42 bits	45 bits' P value
System parameters (λ, x_0)	$(4-2^{-4}, 0.5)$	$(4-2^{-23}, 0.5+2^{-15})$	$(4-2^{-30}, 0.5+2^{-15})$	$(4-2^{-32}, 0.5+2^{-15})$	$(4-2^{-34}, 0.5+2^{-15})$	$(4-2^{-36}, 0.5+2^{-15})$	$(4-2^{-38}, 0.5+2^{-15})$	$(4-2^{-41}, 0.5+2^{-15})$
Frequency	✗	✗	✗	√	√	√	√	0.788699 √
Block frequency ($m = 128$)	✗	✗	√	√	√	√	√	0.880935 √
Cusum-Forward	✗	✗	✗	√	√	√	√	0.321183 √
Cusum-Reverse	✗	✗	✗	√	√	√	√	0.511427 √
Runs	✗	✗	√	√	√	√	√	0.950620 √
Long runs of one	✗	✗	✗	√	√	√	√	0.301448 √
Rank	✗	✗	√	√	√	√	√	0.178158 √
Spectral DFT	✗	✗	✗	✗	✗	√	√	0.581909 √
Nonoverlapping templates	✗	✗	✗	✗	✗	✗	✗	0.645372 √
Overlapping templates ($m = 9$)	✗	✗	✗	√	√	√	√	0.566886 √
Universal	✗	√	√	√	√	√	√	0.725132 √
Approximate entropy ($m = 10$)	✗	✗	✗	✗	✗	√	√	0.877618 √
Random excursions	✗	✗	✗	✗	✗	✗	✗	0.970335 √
Random excursions variant	✗	✗	✗	✗	✗	✗	√	0.125786 √
Linear complexity ($M = 500$)	✗	√	√	√	√	√	√	0.113062 √
Serial ($m = 16$)	✗	✗	✗	✗	✗	√	√	0.115512 √

Chaotic stream cipher

Implementation issue

entropy

MDPI

Article

Complexity of Simple, Switched and Skipped Chaotic Maps in Finite Precision

Maximiliano Antonelli [1,2,*], Luciana De M
and Osvaldo Anibal Rosso [3,4,5,6]

Logistic map is interesting because it is representative of the very large family of quadratic maps. Its expression is:

$$x_{n+1} = 4\,x_n\,(1 - x_n) \tag{10}$$

with $x_n \in \mathbb{R}$.

Note that to effectively work in a given representation it is necessary to change the expression of the map in order to make all the operations in the chosen representation numbers. For example, in the case of LOG the expression in binary fixed-point numbers is:

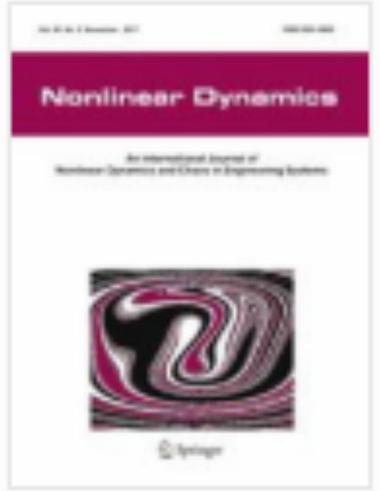
$$x_{n+1} = 4\,\epsilon\ \text{floor}\left\{\frac{x_n(1 - x_n)}{\epsilon}\right\} \tag{11}$$

with $\epsilon = 2^{-B}$ where B is the number of bits that represents the fractional part.

Chaotic stream cipher

PRBG example with sources

SpringerLink



Nonlinear Dynamics
November 2017, Volume 90, Issue 3, pp 1661–1670 | Cite as

Hardware implementation of pseudo-random number generators based on chaotic maps

Authors   Authors and affiliations

Luis Gerardo de la Fraga, Esteban Torres-Pérez, Esteban Tlelo-Cuautle, Cuauhtemoc Mancillas-López

Original Paper
First Online: 30 August 2017

Źródło pomocnicze i jego analiza: https://link.springer.com/article/10.1007/s11071-017-3755-z

Whiteboard exercise with students in the classroom

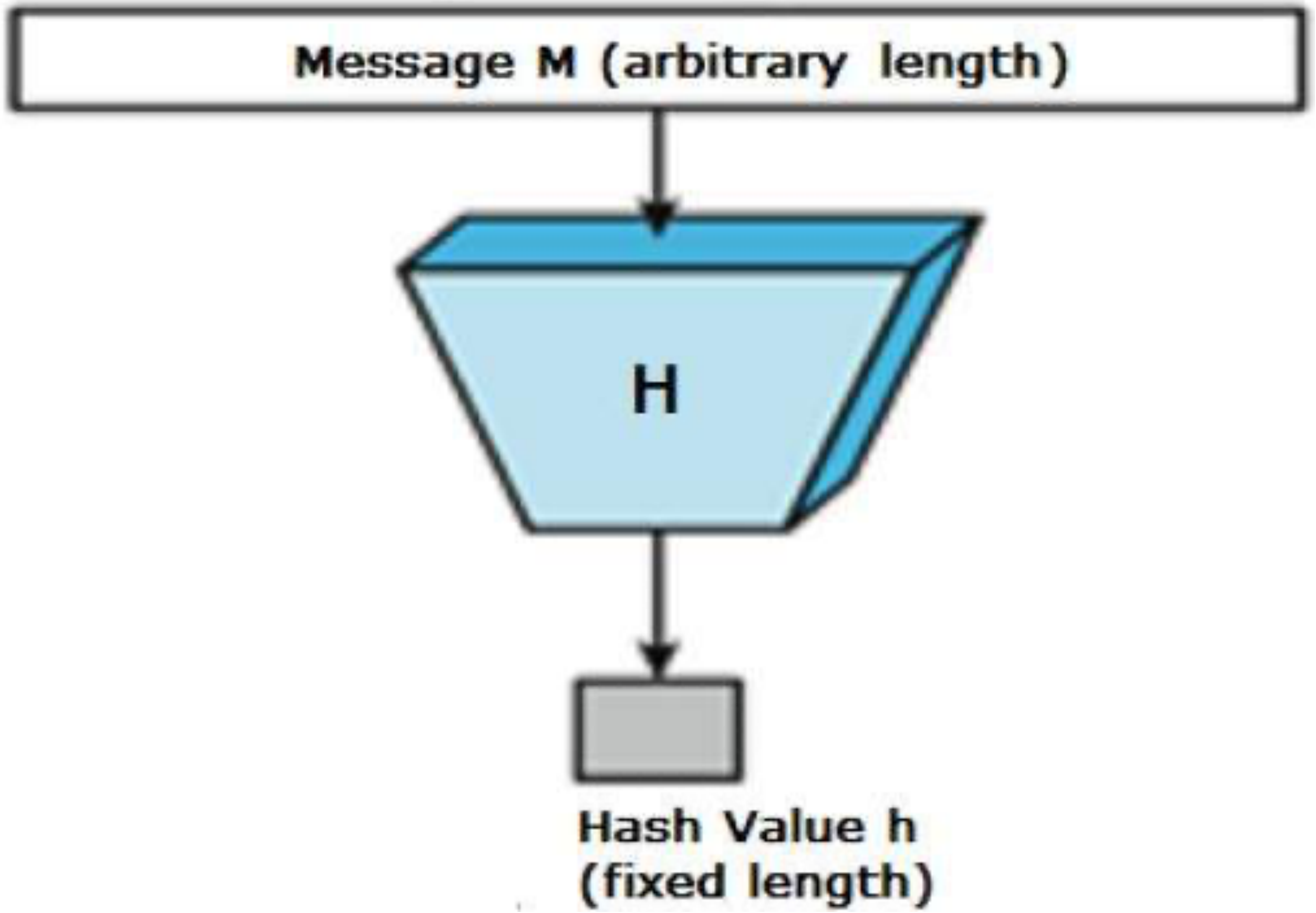
What is the most important security issue of encrypted data using stream ciphers?

Hash

Hash

General idea nad SHA-2 family

- SHA-224
- SHA-256
- SHA-384
- SHA-512
- SHA-512/224
- SHA-512/256



Source (23.11.22): https://www.tutorialspoint.com/cryptography/cryptography_hash_functions.htm

Hash

Selected uses

- store passwords
- ensure data integrity
- secure authentication

SHA256

Teaching example - sha256algorithm.com