OPEN ACCESS

2nd International Electronic Conference on Entropy and Its Applications
15-30 November 2015

www.sciforum.net/conference/ecea-2

*Conference Proceedings Paper – Entropy*

# Chaos Cryptography with prescribed Entropy Production

**Georgios Makris * and Ioannis Antoniou**

Complex Systems Analysis Laboratory, School of Mathematics, Aristotle University of Thessaloniki, Greece, 54124; E-mail: iantonio@math.auth.gr

* Author to whom correspondence should be addressed; E-Mail: georgmak@gmail.com; Tel: +302310997971; Fax: +302310997929.

*Published: 13 November 2015*

---

**Abstract:** Chaos was used as a mechanism for Cryptography by Shannon in his classic 1949 paper, without mentioning of course the word chaos. This idea has been extended and realized by Chaotic *i.e.* Entropy producing Torus Automorphisms. The corresponding algorithms and the software have been developed for any Torus Automorphism, adapted to be applicable for encryption in real time. We may select and combine in an arbitrary way several chaotic maps, creating in this way an infinite number of keys. Decryption is simply the reverse application of the selected maps. Therefore, the cryptography is effectively symmetric. The novelties of our work are summarized as follows: i) The possibility to design classes of chaotic Torus maps with any desirable Entropy Production. The Torus parameters are computed as functions of the selected Entropy Production. Moreover, we design Torus Automorphisms with integer parameters determined from the Entropy Production, in order to apply the designed Torus Automorphisms to cryptography of images or texts. The general construction is also applied to Torus Automorphisms constructed from the Fibonacci sequence. ii) The Encryption Mechanisms are designed and implemented by selecting and combining the Torus Maps in an arbitrary way. These Encryptions are in fact new classes and examples of MonoBlock Ciphers. iii) The application to content involving text and images.

**Keywords:** Entropy Production; Chaos; Cryptography; Torus Automorphisms; Fibonacci Sequence.

**PACS Codes:** 89.70.Cf; 87.19.ls; 05.45.Vx; 87.19.lo; 05.45.Ac; 05.10.-a.

---

## 1. Introduction

Chaotic maps are the simplest dynamical systems with high sensitivity to initial conditions [1–3]. Small deviations of the initial conditions lead to enormous differences of the corresponding orbits. Initial deviations may be due to accuracy errors, approximations or numerical estimations, rendering the long-term forecast for the chaotic systems effectively unattainable. This unstable dynamical behavior is equivalently a local mechanism for Entropy Production. Kolmogorov characterized statistically what we now call Chaotic systems, as Entropy producing dynamical systems [3–8]. After a (small) number of steps, the required information to preserve the initial accuracy for predictions, may exceed the available memory and the computation time and cost may grow superexponentially. This initial duration beyond which prediction and control are operationally unattainalble is called the horizon of predictability [9] of the system and depends on the dynamical system and the simulation program.

Chaos was used, for the first time to our knowledge, as a mechanism for Cryptography by Shannon in his classic mathematical 1949 papers [10–11]. More specifically Shannon used the Baker's map, introduced earlier by Hopf (1934) [12], as a simple deterministic model for mixing and statistical regularity. Of course neither Shannon, nor Hopf used the term Chaos which emerged in the 1970s [1–3].

The Entropy theory of Chaotic maps was developed later by Kolmogorov and his group [4,7–8]. Baker's map is the simplest case of (2-dimensional) chaotic Automorphisms with Entropy production equal to one bit per step. As monotonic Entropy increase is the typical property of Systems satisfying the Second Principle of Thermodynamics, Baker's Map has also served as toy model for understanding the problem of Irreversibility in Statistical Mechanics [6]. Shannon observed that using the Baker's maps, encryption is achieved via successive mixing of the initial information which is "spread" fast over the available state space. Therefore, it is exponentially hard to recover the initial message from the uniform mixture (cryptogram) if the reverse transformation is not known.

The Horseshoe Map [13–14] is a variation of Baker's Map with the Entropy production one bit per iteration also. Both Baker's Map and the Horseshoe Map are Torus Automorphisms. The so-called Cat Map introduced by Arnold in 1968 is a Torus automorphism with stronger mixing than the two previous ones, due to the higher Entropy Production (1,39 bits/iteration).

The numerical analysis of the Cat Map reveals interesting periodicities [15–16]. Although the Cat Map and the Torus Automorphisms admit analytical solutions, computability does seem not increase significantly [17]. Although spectral analysis provides statistical estimates for the Baker's Map and the Cat Map [18–20], this is not obviously applicable to cryptography. The reason is that the local information encoded on points is required, and this information is lost in the statistical estimates.

Most applications of Cryptography with Chaos involvng 2-dimensional maps deal with image encryption [21–22]. We found some results on text encryption [23–26].

The goal of this work is to present a new method for cryptography based on Chaotic Torus Automorphisms, applicable for both image and text encryption in real time. The method allows to design encoding transformations with any desired Entropy Production. After expressing the Entropy Production of Torus Automorphisms in terms of the elements of the corresponding matrix in section 2, we construct Torus Automorphisms with desired Entropy Production in section 3. Cryptographic applications impose the additional requirement for Integer Torus Automorphisms constructed in section 4 and compatible grids specified in section 5. We present the Encryption and Decryption Algorithms in section 6, the

Algorithms for the Insertion of Messages for Encryption in section 7, the Implementation software in section 8 and our conclusions in section 9.

## 2. The Entropy Production of Torus Automorphisms

We consider the Automorphisms of the 2-Torus $Y=[0,1)\times[0,1)$ defined by:

$$S: Y \to Y: \begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = A \begin{bmatrix} x_n \\ y_n \end{bmatrix} (mod\ ?)\ n \in \mathbb{N} \tag{1}$$

$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ is a real matrix with $|\det(A)|=1$ or $ad-bc=\pm 1$?

The Torus Automorphism $S$ has positive Entropy Production if and only if one eigenvalue λ of the Matrix A is greater than 1. This follows from Pesin's 1977 Formula [27]:

$$h? \log_2 \lambda \tag{2}$$

For the famous Cat Map [4]:

$$A = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \text{ and } h? \log_2 \frac{3+\sqrt{5}}{2} \simeq 1,39 \tag{3}$$

The Entropy Production h of the general Torus automorphism can be expressed in terms of the elements c,d of the matrix A, according to the following:

**Theorem 1.** For chaotic Torus Automorphisms with matrix $A$, $|\det(A)|=1$, $a \in \mathbb{R}$:

$$h = \log_2 \frac{(a+d)+\sqrt{(a+d)^2-4(ad-bc)}}{2} = \log_2 \frac{\mathrm{tr}(A)+\sqrt{\mathrm{tr}(A)^2-4\det(A)}}{2} \tag{4}$$

$$\text{with } b \in \mathbb{R}, d > (1+\det(A))-a$$

The proof of formula (4) is straightforward and is given in Appendix A.

We illustrate formula (4), in two examples, namely Arnold's Cat Map and Fibonacci Torus Automorphisms.

**Example 1.** Arnold's Cat Map (1): a = 1, b = 1, c = 1, d = 2, $\det(A)=1$

$$h = \log_2 \frac{(1+2)+\sqrt{(1+2)^2-4}}{2} = \log_2 \frac{3+\sqrt{5}}{2} \simeq 1,39$$

**Example 2.** Fibonacci Torus Automorphisms have integer matrices (1) with elements the successive terms of Fibonacci sequence $F_n$:

$$F_n = F_{n-1} + F_{n-2} \text{ with } F_0 = 0 \text{ and } F_1 = 1$$

For each n = 0, 1, 2, 3, … we obtain the Fibonacci Matrices, each defining a different Torus Automorphism [28]:

$$A(n) = \begin{bmatrix} F_n & F_{n+1} \\ F_{n+2} & F_{n+3} \end{bmatrix} \tag{5}$$

The Entropy Production of each Fibonacci matrix $A(n)$ is computed applying formula (4):

$$\mathrm{h}(n)=\log_2\left(F_{n+2}+\sqrt{(F_{n+2})^2+(-1)^n}\right) \tag{6}$$

The proof of formulas (6) is given in Appendix B.

As the Fibonacci sequence is strictly increasing: $F_{n+1}>F_n$ , we see from formula (6) that the Entropy Production of the family of Fibonacci Automorphisms with matrices A(n) is a strictly increasing function of n:

$$\mathrm{h}(n+1)>\mathrm{h}(n) \tag{7}$$

## 3. Torus Automorphisms with Desired Entropy Production

Based on the analysis of the previous section we shall now construct Torus Automorphisms with prescribed Entropy Production.

**Theorem 2.** There are two families of Torus Automorphisms with Matrices with Entropy Production $\mathrm{h}\geq 0$ with matrices A given by the formulae:

$$A=\begin{bmatrix} \mathrm{a} & \mathrm{b} \\ \dfrac{\mathrm{a}\cdot\left(2^{\mathrm{h}}+2^{-\mathrm{h}}-\mathrm{a}\right)-1}{b} & 2^{\mathrm{h}}+2^{-\mathrm{h}}-\mathrm{a} \end{bmatrix} \tag{8}$$

$$A=\begin{bmatrix} \mathrm{a} & \mathrm{b} \\ \dfrac{\mathrm{a}\cdot\left(2^{\mathrm{h}}-2^{-\mathrm{h}}-\mathrm{a}\right)+1}{b} & 2^{\mathrm{h}}-2^{-\mathrm{h}}-\mathrm{a} \end{bmatrix} \tag{9}$$

for any real value of the parameters a and b, with $b\neq 0$ The families (8),(9) define proper rotations ( $\det(A)=1$) and improper rotations ( $\det(A)=-1$) correspondingly.

The proof of formulae (8), (9) is given in Appendix C.

**Example 3.** Arnold's Cat Map (1): a=1, b=1,c=1,d=2, $\det(A)=1$

$$A=\begin{bmatrix} \mathrm{a} & \mathrm{b} \\ \dfrac{\mathrm{a}\cdot\left(2^{\mathrm{h}}+2^{-\mathrm{h}}\cdot\det(A)-\mathrm{a}\right)-\det(A)}{b} & 2^{\mathrm{h}}+2^{-\mathrm{h}}\cdot\det(A)-\mathrm{a} \end{bmatrix}=$$

$$\begin{bmatrix} 1 & 1 \\ \dfrac{1\cdot\left(2^{\log_2\frac{3+\sqrt{5}}{2}}+2^{-\log_2\frac{3+\sqrt{5}}{2}}\cdot 1-1\right)-1}{1} & 2^{\log_2\frac{3+\sqrt{5}}{2}}+2^{-\log_2\frac{3+\sqrt{5}}{2}}\cdot 1-1 \end{bmatrix}=\begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}$$

**Example 4.** Fibonacci Automorphisms:

The Matrix A($n$) (5) can be expressed in terms of any given Entropy Production $\mathrm{h}$ either using formula (4) or directly by expressing the Fibonacci numbers as a function of h from formula (6):

$$\begin{aligned} F_{n+2}&=2^{\mathrm{h}-1}-2^{-\mathrm{h}-1},\ \ for\ n=0,2,4,... \\ F_{n+2}&=2^{\mathrm{h}-1}+2^{-\mathrm{h}-1},\ \ for\ n=1,3,5,... \end{aligned} \tag{10}$$

As the Fibonacci numbers are integers, formula (10) should be corrected to take integer values. Therefore, we cannot have Fibonacci Automorphisms with any desired Entropy production $\mathrm{h}$. The

Fibonacci Automorphisms with Entropy Production closer to $\mathrm{h}$ from above are obtained according to the following:

**Lemma 1.** The Fibonacci automorphism with Entropy Production nearest to $\mathrm{h}$ , $\mathrm{h}(n) \geq \mathrm{h}$ has matrix (5) with n given by the formula:

$$n = \min\{n_{even}, n_{odd}\} \tag{11}$$

Where the even and odd values of n are given by the formulae:

$$\begin{aligned} n_{even} &= \left\lfloor \log_{\varphi}\left(\sqrt{5}\cdot\left\lceil 2^{\mathrm{h}-1} - 2^{-\mathrm{h}-1}\right\rceil + \frac{1}{2}\right)\right\rfloor - 2 \\ n_{odd} &= \left\lfloor \log_{\varphi}\left(\sqrt{5}\cdot\left\lceil 2^{\mathrm{h}-1} + 2^{-\mathrm{h}-1}\right\rceil + \frac{1}{2}\right)\right\rfloor - 2 \end{aligned} \tag{12}$$

$\varphi = \frac{1+\sqrt{5}}{2}$ the golden number and $\lfloor x \rfloor, \lceil x \rceil$ are the floor and ceiling functions of x [29].

The proof of formula (12) is given in Appendix D. As Entropy Production is a strictly increasing function of n (7) we have to take the smaller of the possible values $n_{even}, n_{odd}$ . From Table 1 we observe that there is no obvious relation between the desired Entropy Production and smallest number n.

**Table 1.**The Matrices of the Fibonacci Automorphisms with Entropy Production nearest (in gray) to a desired Entropy Production $\mathrm{h}$ .

| $\mathrm{h}$ | $n_{odd}(h)$ | $n_{even}(h)$ | $A(n_{odd})$ | $A(n_{even})$ | $\mathrm{h}(n_{odd})$ | $\mathrm{h}(n_{even})$ |
|---|---|---|---|---|---|---|
| 1 | 1 | 0 | $\begin{bmatrix}1 & 1\\ 2 & 3\end{bmatrix}$ | $\begin{bmatrix}0 & 1\\ 1 & 2\end{bmatrix}$ | 1.90 | 1.27 |
| 1.2 | 1 | 0 | $\begin{bmatrix}1 & 1\\ 2 & 3\end{bmatrix}$ | $\begin{bmatrix}0 & 1\\ 1 & 2\end{bmatrix}$ | 1.90 | 1.27 |
| 1.8 | 1 | 2 | $\begin{bmatrix}1 & 1\\ 2 & 3\end{bmatrix}$ | $\begin{bmatrix}1 & 2\\ 3 & 5\end{bmatrix}$ | 1.90 | 2.62 |
| 2 | 3 | 2 | $\begin{bmatrix}2 & 3\\ 5 & 8\end{bmatrix}$ | $\begin{bmatrix}1 & 2\\ 3 & 5\end{bmatrix}$ | 3.31 | 2.62 |
| 2.6 | 3 | 2 | $\begin{bmatrix}2 & 3\\ 5 & 8\end{bmatrix}$ | $\begin{bmatrix}1 & 2\\ 3 & 5\end{bmatrix}$ | 3.31 | 2.62 |
| 3 | 3 | 4 | $\begin{bmatrix}2 & 3\\ 5 & 8\end{bmatrix}$ | $\begin{bmatrix}3 & 5\\ 8 & 13\end{bmatrix}$ | 3.31 | 4.01 |
| 4 | 5 | 4 | $\begin{bmatrix}5 & 8\\ 13 & 21\end{bmatrix}$ | $\begin{bmatrix}3 & 5\\ 8 & 13\end{bmatrix}$ | 4.70 | 4.01 |
| 6 | 7 | 8 | $\begin{bmatrix}13 & 21\\ 34 & 55\end{bmatrix}$ | $\begin{bmatrix}21 & 34\\ 55 & 89\end{bmatrix}$ | 6.09 | 6.78 |
| 9 | 13 | 12 | $\begin{bmatrix}233 & 377\\ 610 & 987\end{bmatrix}$ | $\begin{bmatrix}144 & 233\\ 377 & 610\end{bmatrix}$ | 10.25 | 9.56 |

| h | $n_{odd}(h)$ | $n_{even}(h)$ | $A(n_{odd})$ | $A(n_{even})$ | $h(n_{odd})$ | $h(n_{even})$ |
|---|---|---|---|---|---|---|
| 11 | 15 | 16 | $\begin{bmatrix} 610 & 987 \\ 1597 & 2584 \end{bmatrix}$ | $\begin{bmatrix} 987 & 1597 \\ 2584 & 4181 \end{bmatrix}$ | 11.64 | 12.34 |
| 100 | 143 | 144 | A(143) | A(144) | 100.50 | 101.20 |
| $A(143)=\begin{bmatrix} 343358302784187294870275058337 & 555565404224292694404015791808 \\ 898923707008479989274290850145 & 145448911123277268367830664195 3 \end{bmatrix}$ | | | | | | |
| $A(144)=\begin{bmatrix} 555565404224292694404015791808 & 898923707008479989274290850145 \\ 145448911123277268367830664195 3 & 235341281824125267295259749209 8 \end{bmatrix}$ | | | | | | |

## 4. Integer Torus Automorphisms with Given Entropy Production

The implementation of cryptographic algorithms by chaotic tansformations is realized by restricting the action of the chaotic maps onto some appropriately selected grid of size $N \times N$ represented as the set $\mathbb{Z}_N \times \mathbb{Z}_N$, $\mathbb{Z}_N = \{0,1,2,...,N-1\}$. In order to preserve the grid structure the Torus Automorphisms should have integer matrix elements.

Based on Theorem 2 we shall construct Integer Torus Automorphisms with Entropy Production nearest to a prescribed Entropy Production $h > 0$. In fact, there are two families of Torus Automorphisms with Matrices $A_+ = A_+(h, a, \boldsymbol{b})$ and $A_- = A_-(h, a, \boldsymbol{b})$ with $\det(A_+)=1$ and $\det(A_-)=-1$ correspondingly with arbitrary integer values of a, b, constructed by the following algorithms

The Algorithms for the computation of $A_+$ and $A_-$ with arbitrary input (h, a, b), $h \in (0,\infty)$, $a, \boldsymbol{b} \in \mathbb{Z}$ are presented below in Matlab:

**MatLab Function for the computation of** $A_+$

```
function [A] = entropy_with_ab1(h,a,b)
    x=2^(-h)+2^(h)
    x1=ceil(x);
    d=x1-a;
    while d<=2-a || (mod(a*d,b)~=1 && b~=1)
            if(mod(a,b)==0 || mod(b,a)==0 )
                    a=a+1;
                    d=x1-a;
            else
                    x1=x1+1;
                    d=x1-a;
            end
    end;
    A=[a b;(a*d-1)/b d]
    E=eig(A)
    L1=max(E)
    h1=log2(L1)
end
```

**MatLab Function for the computation of** $A_-$

```
function [A] = entropy_with_ab2(h,a,b)
    x=-2^(-h)+2^(h)
    x1=ceil(x);
```

```
    d=x1-a;
    while d<=-a || (mod(a*d,b)~=n-1 && b~=1)
        if(mod(a,b)==0 || mod(b,a)==0 )
            a=a+1;
            d=x1-a;
        else
            x1=x1+1;
            d=x1-a;
        end
    end;
    A=[a b;(a*d+1)/b d]
    E=eig(A)
    L1=max(E)
    h1=log2(L1)
end
```

Some examples of Integer Torus Automorphisms with the corresponding Entropy Production computated with Matlab functions are presented in Table 2.

**Table 2.** Examples of Integer Torus Automorphisms with given Entropy Production.

| **Input** | | | **Output** | | | |
|---|---|---|---|---|---|---|
| h | a | b | $A_+ = \begin{bmatrix} a & b \\ \frac{ad-1}{b} & d \end{bmatrix}$ | $A_- = \begin{bmatrix} a & b \\ \frac{ad+1}{b} & d \end{bmatrix}$ | h(+) | h(−) |
| 1.2 | 1 | 1 | $\begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}$ *Arnold's Cat Map* | $\begin{bmatrix} 1 & 1 \\ 2 & 1 \end{bmatrix}$ | 1.39 | 1.27 |
| 1.2 | 2 | 3 | $\begin{bmatrix} 2 & 3 \\ 1 & 2 \end{bmatrix}$ | $\begin{bmatrix} 2 & 3 \\ 1 & 1 \end{bmatrix}$ | 1.90 | 1.72 |
| 3.5 | 1 | 1 | $\begin{bmatrix} 1 & 1 \\ 10 & 11 \end{bmatrix}$ | $\begin{bmatrix} 1 & 1 \\ 12 & 11 \end{bmatrix}$ | 3.57 | 3.59 |
| 3.5 | 5 | 1 | $\begin{bmatrix} 5 & 1 \\ 34 & 7 \end{bmatrix}$ | $\begin{bmatrix} 5 & 1 \\ 36 & 7 \end{bmatrix}$ | 3.57 | 3.59 |
| 3.5 | 5 | 3 | $\begin{bmatrix} 5 & 3 \\ 13 & 8 \end{bmatrix}$ | $\begin{bmatrix} 5 & 3 \\ 12 & 7 \end{bmatrix}$ | 3.69 | 3.59 |
| 11 | 2 | 1 | $\begin{bmatrix} 2 & 1 \\ 4093 & 2047 \end{bmatrix}$ | $\begin{bmatrix} 2 & 1 \\ 4093 & 2046 \end{bmatrix}$ | 11.00 | 11.00 |
| 15 | 5 | 4 | $\begin{bmatrix} 5 & 4 \\ 40956 & 32765 \end{bmatrix}$ | $\begin{bmatrix} 5 & 4 \\ 40954 & 32763 \end{bmatrix}$ | 15.00 | 15.00 |

## 5. Grids Compatible with the desired Entropy Production

The restriction of an integer Torus automorphism to the grid $\mathbb{Z}_N \times \mathbb{Z}_N$ (mod N):

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = A \begin{bmatrix} x \\ y \end{bmatrix} (\mathrm{mod}\, N) = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} (\mathrm{mod}\, N) \tag{13}$$

is a periodic transformation, called the $N \times N$ discretization of the integer automorphism.

Integer Torus Automorphisms have been implemented on $N \times N$ grids and the corresponding periods have been related to the grid size $N$ [15–17,20,22].

Before selecting the grid size N we should verify that the discretization of the integer automorphism does not reduce entropy. The following example shows that Entropy production depends significantly on the grid size.

Given a desired Entropy Production $h = 15.00$ , a=5 and b=4 we construct the integer Torus Automorphisms $A_1$ using algorithm (1+) and then compute the discretizations $A_{100}$ and $A_{50000}$ on the grids with sizes 100x100 and 50000x50000 correspondingly. Using formula (4) we find the corresponding entropies: $h_1 = 15.00$, $h_{100} = 6.49$, $h_{50000} = 15.00$

$$A_1 = \begin{bmatrix} 5 & 4 \\ 40956 & 32765 \end{bmatrix},\ A_{100} = A_1 \bmod 100 = \begin{bmatrix} 5 & 4 \\ 56 & 65 \end{bmatrix} \bmod 100,$$

$$A_{50000} = A_1 \bmod 50000 = \begin{bmatrix} 5 & 4 \\ 40956 & 32765 \end{bmatrix} \bmod 50000$$

Therefore, we should identify the admissible grids which do not reduce the Entropy Production. The admissible grid sizes are given by the following theorem

**Theorem 3.** The grid discretizations of integer Torus Automorphisms with the desired Entropy Production satisfy the condition:

$$N > \max\{a, b, c, d\} \tag{14}$$

Condition (14) is equivalently expressed in terms of Entropy Production, using (8) and (9) as:

$$N > \max\left\{ a(\mathrm{mod}\, N), b(\mathrm{mod}\, N), 2^h + 2^{-h} - a(\mathrm{mod}\, N), \frac{a(\mathrm{mod}\, N) \cdot \left(2^h + 2^{-h} - a(\mathrm{mod}\, N)\right) - 1}{b(\mathrm{mod}\, N)} \right\} \tag{15}$$

$$\text{for } \det(A) = 1$$

$$N > \max\left\{ a(\mathrm{mod}\, N), b(\mathrm{mod}\, N), 2^h - 2^{-h} - a(\mathrm{mod}\, N), \frac{a(\mathrm{mod}\, N) \cdot \left(2^h - 2^{-h} - a(\mathrm{mod}\, N)\right) + 1}{b(\mathrm{mod}\, N)} \right\} \tag{16}$$

$$\text{for } \det(A) = -1$$

**Proof.**

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} (\mathrm{modN}) = \begin{bmatrix} a(\mathrm{modN}) & b(\mathrm{modN}) \\ c(\mathrm{modN}) & d(\mathrm{modN}) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} (\mathrm{modN})$$

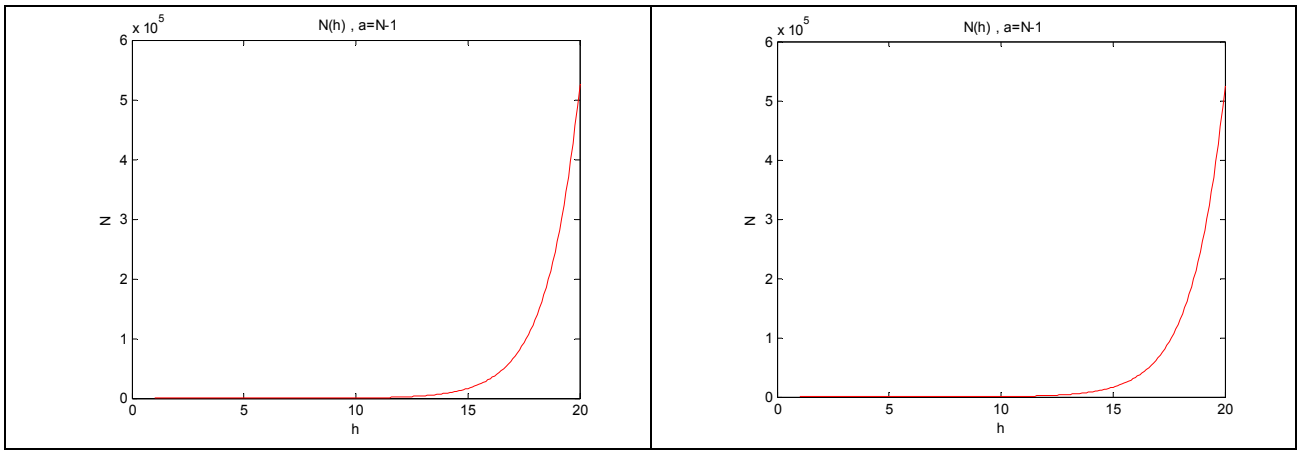
As the remainders $a(\mathrm{modN}), b(\mathrm{modN}), c(\mathrm{modN}), d(\mathrm{modN})$ are not greater than a,b,c,d correspondingly, we have: $\mathrm{tr}\begin{bmatrix} a & b \\ c & d \end{bmatrix} = a+d \le a(\mathrm{modN}) + d(\mathrm{modN}) = \mathrm{tr}\begin{bmatrix} a(\mathrm{modN}) & b(\mathrm{modN}) \\ c(\mathrm{modN}) & d(\mathrm{modN}) \end{bmatrix}$

Therefore, from (4), (8) and (9) we have: $h\begin{bmatrix} a & b \\ c & d \end{bmatrix} \ge h\begin{bmatrix} a(\mathrm{modN}) & b(\mathrm{modN}) \\ c(\mathrm{modN}) & d(\mathrm{modN}) \end{bmatrix}$

The equality: $h\begin{bmatrix} a & b \\ c & d \end{bmatrix} = h\begin{bmatrix} a(\mathrm{mod}N) & b(\mathrm{mod}N) \\ c(\mathrm{mod}N) & d(\mathrm{mod}N) \end{bmatrix}$ is true if and only if each matrix element of the matrix A is less than the remainder N which is the grid size N, which is the desired result. ■

The minimal grid size $N$ as a function of the value of the desired Entropy Production is plotted in Figure 1

**Figure 1. Minimal Grid Size for given Entropy Production for Automorphisms with (a) detA=1. (b) detA=-1**




## 6. Encryption and Decryption

The message (image or text) is inserted on the $N \times N$ grid as the initial data set which is transformed by the selected integer Torus Automorphisms acting as encryption. Decryption is achieved by the application of the inverse automorphism to the transformed (encrypted) data set. The Encryption involves 7 steps, presented below:

**Algorithm 2: Encryption**

Step 0: inputs: content (Text or/and Image), $h \in (0,\infty)$, $a, b \in \mathbb{Z}$, n=1,2,3,…

Step 1: if the content is image with equal height and width, then go to step 4.
Else add pixels so that the image has equal height and width and go to step 4

Step 2: if content is text, place the text in a 2-dimensional grid so that each array element is a character (using Algorithm 3)

Step 3: if the content is text and image apply step 1 to each image and step2 to the text.

Step 4: Apply the selected transformations (Algorithm 1, Step 9) for an number of iterations n on the table

for iterations=0 to n-1
for i=0 to N-1
for j=0 to N-1

$$\begin{bmatrix} x'_i \\ y'_j \end{bmatrix} = A\begin{bmatrix} x_i \\ y_j \end{bmatrix} (\mathrm{mod}\ N)$$

Step 5: if content is image then go to step 7.

Step 6: convert the modified grid from of step 2 to text.

Step 7: return the transformed table

The Encoding Key is the selected sequence of transformations:

$$Encoding\ Key = (h_1,\ a_1,\ b_1,\ n_1),(h_2,\ a_2,\ b_2,\ n_2),...,(h_k,\ a_k,\ b_k,\ n_k), k = 1,2,3,... \quad (17)$$

Decryption is simple for those who hold the Encoding Key (17). Decryption is realized by applying algorithm 2 for the encrypted content with the sequence of inverse transformations in the reverse order. We replace the matrix A of step 4 in Algorithm 2 with the inverse matrix $A^{-1}$. The Decoding Key is simply the reverse of the Encoding Key:

$$Decoding\ Key = (h_k,\ a_k,\ b_k,\ n_k),...,(h_2,\ a_2,\ b_2,\ n_2),(h_1,\ a_1,\ b_1,\ n_1), k = 1,2,3,... \quad (18)$$

## 7. Insertion of Messages for Encryption

The insertion of images is straightforward. If the image has $N \times N$ pixels, then it ready for encryption on the $N \times N$ grid. If the image has $K \times M$ pixels, then we simply add arbitrary pixels to obtain an image with $N \times N$ pixels ( $N = \max\{K, M\}$ ).

The insertion of texts is described in the following algorithm:

**Algorithm 3:**

Step 0: Input: Text

Step 1: Set m=length(Text)

Step 2: Set $N = \lceil \sqrt{m} \rceil$

Step 3: Fill character matrix P[N,N] with each character of the Text or a space

```
    char=1
for i=0 to N-1
            for j=0 to N-1
            if char <=m then
            set P[i.j]=Text(char)
            else
                    set P[i.j]=' ' (space character)
            set char=char + 1
```

Step 4: Output: Matrix of characters P[N,N]

So we create a $N \times N$ matrix of characters with the properties:

1) The size of the matrix is the minimal positive integer N, such that the length m of the text fits the matrix: $N \times N \geq m$ .

2) The number of characters in each line may change during encryption, because all special characters like "enter" are also involved in encryption.

Example of Text Insertion:

| |
|---|
| CHAOS CRYPTOGRAPHY WITH PRESCRIBED ENTROPY PRODUCTION<br>GEORGE MAKRIS, IOANNIS ANTONIOU |

As the length of the text is m=85, we need a matrix of size N=10 because the 9x9 matrix does not fit the text. In other words 100 is the minimal encoding length.

**Table 3.** Example of text insertion

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| C | H | A | O | S | | C | R | Y | P |
| T | O | G | R | A | P | H | Y | | W |
| I | T | H | | P | R | E | S | C | R |
| I | B | E | D | | E | N | T | R | O |

| P | Y | | P | R | O | D | U | C | T |
|---|---|---|---|---|---|---|---|---|---|
| I | O | N | \n | G | E | O | R | G | E |
| | M | A | K | R | I | S | , | | I |
| O | A | N | N | I | S | | A | N | T |
| O | N | I | O | U | | | | | |
| | | | | | | | | | |

## 8. Implementation of "Cryptography With Chaos"

We developed the software “chaos_cryptography” with Java for four reasons:

(i) The language Java is independent of the operating system and platform.
(ii) The Java programs run on Windows, Linux, Unix and MacOS, mobile phones, Ipads, Playstations and other game consoles without any modification like compilation or changing the source code for each different operating system.
(iii) The Java software has a graphical user interface, is very simple and user friendly.
(iv) The libraries (classes) developed in Java are compatible with any other software and application.

The user may encrypt or decrypt images and texts selecting any combination of the maps (5), (8), (9) and (13). Window dialogs alert the user in case of procedural errors.

We show examples of Image encryption and Text encryption in Tables 4 and 5 correspondingly

**Table 4.** Image Encryption

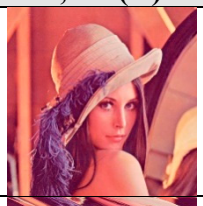
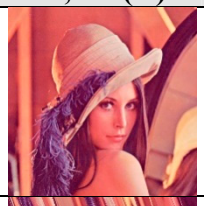
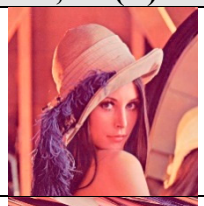
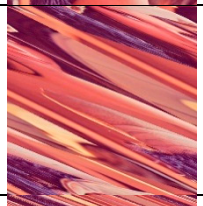
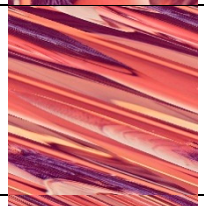
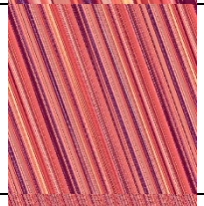
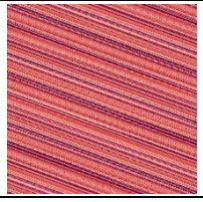
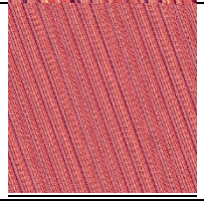
| **Torus Automorphisms with given Entropy Production** h | | | |
|---|---|---|---|
| **Iterations** | **h=1.39, a=1, b=1, det(A)=1** | **h=1.39, a=3, b=5, det(A)=1** | **h=1.90, a=1, b=1, det(A)=-1** |
| t=0 |  |  |  |
| t=1 |  |  |  |
| t=2 |  |  |  |
| t=3 |  |  |  |

**Table 5.** Text Encryption

| **Torus Automorphisms with given Entropy Production** h | | | |
|---|---|---|---|
| **Iterations** | **h=1.39, a=1, b=1, det(A)=1** | **h=1.39, a=3, b=5, det(A)=1** | **h=1.39, a=1, b=1, det(A)=-1** |
| t=0 | CHAOS CRYPTOGRAPHY WITH PRESCRIBED ENTROPY PRODUCTION GEORGE MAKRIS, IOANNIS ANTONIOU | CHAOS CRYPTOGRAPHY WITH PRESCRIBED ENTROPY PRODUCTION GEORGE MAKRIS, IOANNIS ANTONIOU | CHAOS CRYPTOGRAPHY WITH PRESCRIBED ENTROPY PRODUCTION GEORGE MAKRIS, IOANNIS ANTONIOU |
| t=1 | C INREDTCWRRTH OIIOUSRCOIOA US ,GTITGOHRS A EPBIYE A NI T O DPPC HR OMNP RREEYY OAA NKGONS P N | C CEAEYOSRRNRNTIS O C R P D ,TKHIGI MA OIE BUTOD GHGRIOOPNPAINET HSCAP T O Y U AP ORY RSWN ION E R | CWCTDERNI O HTRRUOIISSU AOIOCRTG, OGTIHBPE A SR A EYIIN CPPD O TMO RHR PNR AAO YYEESNOGKNN P |
| t=2 | CKE TUUGSR GYOD EP POSRIOYMN NPNB RTNCHRSOPCIH IR O E A AR YS,IODPIT HTE GO TAICNRR OAAI OWNE | CGYNCISGAO T U O P YO E IURINSRNSART AT A GSTR AHEP RO EORHP C D R E D B OOTNIICNEOPNH PIR,OIWKYM | CICG IP ANAOWORT IDR HOGC SISNR OR KTHSTUGH YNDTER ,B TPYNE RRA PA N MEPNUO E ECOR IOI OOAYP S I |
| t=3 | COA SECRYETMGKAIH, IIAHNPSEACTINEO N R P R D C IHNOG ORGP OARRPSY WOTN IR SNROBIDUE T O Y P O U T | CRSHY APCONDIT BREEOAISK ,RM I PTCRHRE ISIRGOGENEO OO UN I GWHRTYAO PCO TDPPURYIANSNT NOA | CRSHY APCOOAIANSNT NPURYCO TDPTYAO PGWHRO UN I OIRGOGENEO ISPTCRHRE ,RM IAISKIT BREEOND |

## 9. Conclusions

The novelties of our work are summarized as follows: i) The possibility to design classes of chaotic Torus maps with any desirable Entropy Production. The Torus parameters are computed as functions of the selected Entropy Production. Moreover, we design Torus Automorphisms with integer parameters determined from the Entropy Production, in order to apply the designed Torus Automorphisms to cryptography of images or texts. The general construction is also applied to Torus Automorphisms constructed from the Fibonacci sequence. ii) The Encryption Mechanisms are designed and implemented by selecting and combining the Torus Maps in an arbitrary way. The proposed encryption is permutation based and symmetric, but is neither stream nor block cipher [29]. We propose the name “MonoBlock” ciphers for this new class of encryptions. iii) The application to content involving text and images.

We may select and combine easily in an arbitrary way several chaotic maps, creating in this way an unlimited number of keys, as large or as small as desired. The Key (17) does not depend on the message size, in contradistinction to classical permutation algorithms, where secure encryption requires large

keys [30]. As a result we have not only flexibility, but also high security. The quantitative evaluation of the indices assessing the cryptographic security will be presented elsewhere.

As our libraries (classes) are compatible with any other software and application, the method is easily adapted to any real datasets like: websites, e-mails, smart phones, sound and video.

**Acknowledgments**

We thank professors Poulakis D. and Farmakis N. of the School of Mathematics of Aristotle University of Thessaloniki for useful remarks.

**Conflicts of Interest**

The authors declare no conflict of interest.

**Appendix A: Proof of formula (4)**

The eigenvalues of matrix $A$ with $\mathrm{ad}-\mathrm{bc}=1$ are:

$$\lambda_1=\frac{(\mathrm{a}+\mathrm{d})+\sqrt{(\mathrm{a}+\mathrm{d})^2-4}}{2},\ \lambda_2=\frac{(\mathrm{a}+\mathrm{d})-\sqrt{(\mathrm{a}+\mathrm{d})^2-4}}{2} \tag{A1}$$

$$\text{with } d\in(-\infty,-2-\mathrm{a})\cup(2-\mathrm{a},\infty)$$

$$\lambda_1>1\Rightarrow\frac{(\mathrm{a}+\mathrm{d})+\sqrt{(\mathrm{a}+\mathrm{d})^2-4}}{2}>1\Rightarrow(\mathrm{a}+\mathrm{d})+\sqrt{(\mathrm{a}+\mathrm{d})^2-4}>2\Rightarrow$$

$$\sqrt{(\mathrm{a}+\mathrm{d})^2-4}>2-(\mathrm{a}+\mathrm{d}) \tag{A2}$$

We shall identify the range of the parameters a and d so that (A2) is true.

- For $d>2-a$, $2-(a+d)<0$ therefore (A2) is always true.
- For $2-(\mathrm{a}+\mathrm{d})\geq 0$ (A2) is not valid. Indeed :

$$\left(\sqrt{(\mathrm{a}+\mathrm{d})^2-4}\right)^2>(2-(\mathrm{a}+\mathrm{d}))^2\Rightarrow(\mathrm{a}+\mathrm{d})^2-4>4+(\mathrm{a}+\mathrm{d})^2-4(\mathrm{a}+\mathrm{d})\Rightarrow$$

$$4(\mathrm{a}+\mathrm{d})>8\Rightarrow(\mathrm{a}+\mathrm{d})>2\Rightarrow 2-(\mathrm{a}+\mathrm{d})<0$$

$$\text{Therefore } \lambda_1>1 \text{ if and only if } a+d>2 \tag{A3}$$

Since $\mathrm{ad}-\mathrm{bc}=1$, we have:

$$A=\begin{bmatrix}\mathrm{a} & \mathrm{b}\\ \dfrac{\mathrm{a}d-1}{b} & \mathrm{d}\end{bmatrix},\ \text{for } d\in(2-\mathrm{a},\infty) \tag{A4}$$

Inserting the value of $\lambda_1$ in the Entropy formula $\mathrm{h}=\log_2\lambda_1$ we obtain:

$$\mathrm{h}=\log_2\frac{(\mathrm{a}+\mathrm{d})+\sqrt{(\mathrm{a}+\mathrm{d})^2-4}}{2},\ \mathrm{a}\in\mathbb{R},b\in\mathbb{R},d>2-\mathrm{a} \tag{A5}$$

Repeating the same steps for the case $\det(A)=-1$ :

$$\lambda_1 = \frac{(\mathrm{a}+\mathrm{d})+\sqrt{(\mathrm{a}+\mathrm{d})^2+4}}{2},\ \lambda_2 = \frac{(\mathrm{a}+\mathrm{d})-\sqrt{(\mathrm{a}+\mathrm{d})^2+4}}{2} \tag{A6}$$

$$\text{with } d \in (-\infty, -2-\mathrm{a}) \cup (2-\mathrm{a}, \infty)$$

$$A = \begin{bmatrix} \mathrm{a} & \mathrm{b} \\ \dfrac{\mathrm{a}d+1}{b} & \mathrm{d} \end{bmatrix}, \text{燼} \in \mathbb{R}, b \in \mathbb{R}-\{0\}, \text{燼} > - \tag{A7}$$

$$\mathrm{h} = \log_2 \frac{(\mathrm{a}+\mathrm{d})+\sqrt{(\mathrm{a}+\mathrm{d})^2+4}}{2},\ \mathrm{a} \in \mathbb{R}, b \in \mathbb{R}, d > -\mathrm{a} \tag{A8}$$

From (A5) and (A8) we obtain formula (4).

## Appendix B: Proof of formula (6)

For the matrix $\mathrm{A}(n)$:

$$\det(\mathrm{A}(n)) = \det\left(\begin{bmatrix} F_n & F_{n+1} \\ F_{n+2} & F_{n+3} \end{bmatrix}\right) = F_n \cdot F_{n+3} - F_{n+1} \cdot F_{n+2} = F_n \cdot (F_{n+1} + F_{n+2}) - F_{n+1} \cdot F_{n+2} =$$

$$= F_n \cdot (2F_{n+1} + F_n) - F_{n+1} \cdot (F_{n+1} + F_n) = 2F_n \cdot F_{n+1} + F_n^{\,2} - F_{n+1}^2 - F_n \cdot F_{n+1} = F_n^{\,2} - F_{n+1}^2 + F_n \cdot F_{n+1} =$$

$$= F_n^{\,2} + F_{n+1} \cdot (F_n - F_{n+1}) = F_n^{\,2} + F_{n+1} \cdot (-F_{n-1}) = F_n^{\,2} - F_{n+1} \cdot F_{n-1} \Rightarrow$$

$$\det(\mathrm{A}(n)) = F_n^{\,2} - F_{n+1} \cdot F_{n-1} \tag{B1}$$

From Cassini formula [31] we have:

$$F_{n+1} \cdot F_{n-1} - F_n^{\,2} = (-1)^n \ \Rightarrow$$

$$F_n^{\,2} - F_{n+1} \cdot F_{n-1} = (-1)^{n-1} \tag{B2}$$

From (B1) and (B2):

$$\det(\mathrm{A}(n)) = \det\left(\begin{bmatrix} F_n & F_{n+1} \\ F_{n+2} & F_{n+3} \end{bmatrix}\right) = (-1)^{n-1} \tag{B3}$$

**For the case:** $\det(\mathrm{A}(n)) = -1$

The eigenvalues are computed from formula (A8):

$$\lambda_1 = \frac{(F_n + F_{n+3})+\sqrt{(F_n + F_{n+3})^2+4}}{2},\ \lambda_2 = \frac{(F_n + F_{n+3})-\sqrt{(F_n + F_{n+3})^2+4}}{2} \tag{B4}$$

From the definition of the Fibonacci sequence we have:

$$F_n + F_{n+3} = F_n + F_{n+1} + F_{n+2} = F_{n+2} + F_{n+2} = 2 \cdot F_{n+2} \tag{B5}$$

Inserting (B5) into (B4) we have:

$$\lambda_1 = \frac{(2 \cdot F_{n+2})+\sqrt{(2 \cdot F_{n+2})^2+4}}{2},\ \lambda_2 = \frac{(2 \cdot F_{n+2})-\sqrt{(2 \cdot F_{n+2})^2+4}}{2} \Rightarrow$$

$$\lambda_1 = F_{n+2} + \sqrt{\left(F_{n+2}\right)^2 + 1} \quad , \quad \lambda_2 = F_{n+2} - \sqrt{\left(F_{n+2}\right)^2 + 1} \tag{B6}$$

From formula (2) and (B6) the Entropy Production $\mathrm{h}$ is:

$$\mathrm{h}\,?\ \log_2\left(F_{n+2} + \sqrt{\left(F_{n+2}\right)^2 + 1}\right) \tag{B7}$$

**For the case:** $\det\left(\mathrm{A}(n)\right) = 1$

From formula (A1) we have:

$$\lambda_1 = \frac{\left(F_n + F_{n+3}\right) + \sqrt{\left(F_n + F_{n+3}\right)^2 - 4}}{2},\ \lambda_2 = \frac{\left(F_n + F_{n+3}\right) - \sqrt{\left(F_n + F_{n+3}\right)^2 - 4}}{2} \Rightarrow$$

$$\lambda_1 = \frac{\left(2 \cdot F_{n+2}\right) + \sqrt{\left(2 \cdot F_{n+2}\right)^2 - 4}}{2},\ \lambda_2 = \frac{\left(2 \cdot F_{n+2}\right) - \sqrt{\left(2 \cdot F_{n+2}\right)^2 - 4}}{2} \Rightarrow$$

$$\lambda_1 = F_{n+2} + \sqrt{\left(F_{n+2}\right)^2 - 1} \quad , \quad \lambda_2 = F_{n+2} - \sqrt{\left(F_{n+2}\right)^2 - 1} \tag{B8}$$

From formula (2) and (B8) the Entropy Production $\mathrm{h}$ is:

$$\mathrm{h} = \log_2\left(F_{n+2} + \sqrt{\left(F_{n+2}\right)^2 - 1}\right) \tag{B9}$$

From (B7) and (B9) we obtain formula (7).

## Appendix C: Proof of formulas (8) and (9)

For the case $\det(\mathrm{A})=1$ we can express the trace a+d in terms of $\mathrm{h}$ using (A5):

$$\mathrm{h} = \log_2 \frac{\left(\mathrm{a}+\mathrm{d}\right) + \sqrt{\left(\mathrm{a}+\mathrm{d}\right)^2 - 4}}{2} \Rightarrow \frac{\left(\mathrm{a}+\mathrm{d}\right) + \sqrt{\left(\mathrm{a}+\mathrm{d}\right)^2 - 4}}{2} = 2^{\mathrm{h}} \Rightarrow \left(\mathrm{a}+\mathrm{d}\right) + \sqrt{\left(\mathrm{a}+\mathrm{d}\right)^2 - 4} = 2 \cdot 2^{\mathrm{h}} \Rightarrow$$

$$\sqrt{\left(\mathrm{a}+\mathrm{d}\right)^2 - 4} = 2 \cdot 2^{\mathrm{h}} - \left(\mathrm{a}+\mathrm{d}\right) \Rightarrow \left(\sqrt{\left(\mathrm{a}+\mathrm{d}\right)^2 - 4}\right)^2 = \left(2 \cdot 2^{\mathrm{h}} - \left(\mathrm{a}+\mathrm{d}\right)\right)^2 \Rightarrow$$

$$\left(\mathrm{a}+\mathrm{d}\right)^2 - 4 = \left(2 \cdot 2^{\mathrm{h}}\right)^2 + \left(\mathrm{a}+\mathrm{d}\right)^2 - 2 \cdot 2 \cdot 2^{\mathrm{h}} \cdot \left(\mathrm{a}+\mathrm{d}\right) \Rightarrow$$

$$\left(\mathrm{a}+\mathrm{d}\right) = \frac{\left(2 \cdot 2^{\mathrm{h}}\right)^2 + 4}{2 \cdot 2 \cdot 2^{\mathrm{h}}} = 2^{\mathrm{h}} + 2^{-\mathrm{h}} \tag{C1}$$

Inserting (C1) to (A4) we obtain formula (8).
Repeating the same steps for the case $\det(\mathrm{A})=-1$:

$$\left(\mathrm{a}+\mathrm{d}\right) = \frac{\left(2 \cdot 2^{\mathrm{h}}\right)^2 - 4}{2 \cdot 2 \cdot 2^{\mathrm{h}}} = 2^{\mathrm{h}} - 2^{-\mathrm{h}} \tag{C2}$$

Inserting (C2) to (A7) we obtain formula (9).

## Appendix D: Proof of formula (12)

For even n:

$$\mathrm{h}(n)=\log_2\left(F_{n+2}+\sqrt{\left(F_{n+2}\right)^2+1}\right)\Rightarrow 2^{\mathrm{h}(n)}=F_{n+2}+\sqrt{\left(F_{n+2}\right)^2+1}\Rightarrow$$

$$\sqrt{\left(F_{n+2}\right)^2+1}=2^{\mathrm{h}(n)}-F_{n+2}\Rightarrow\left(F_{n+2}\right)^2+1=2^{2\cdot\mathrm{h}(n)}-2\cdot 2^{\mathrm{h}(n)}F_{n+2}+\left(F_{n+2}\right)^2\Rightarrow$$

$$2\cdot 2^{\mathrm{h}(n)}F_{n+2}=2^{2\cdot\mathrm{h}(n)}-1\Rightarrow F_{n+2}=2^{\mathrm{h}(n)-1}-2^{-\mathrm{h}(n)-1}$$

We take as $F_{n+2}$ the ceiling function: $F_{n+2}=\left\lceil 2^{\mathrm{h}(n)-1}-2^{-\mathrm{h}(n)-1}\right\rceil$

$$F_n=\left\lfloor\frac{\varphi^n}{\sqrt{5}}+\frac{1}{2}\right\rfloor\Rightarrow F_{n+2}=\left\lfloor\frac{\varphi^{n+2}}{\sqrt{5}}+\frac{1}{2}\right\rfloor\Rightarrow n+2=\left\lfloor\log_\varphi\left(\sqrt{5}\cdot F_{n+2}+\frac{1}{2}\right)\right\rfloor\Rightarrow$$

$$n=\left\lfloor\log_\varphi\left(\sqrt{5}\cdot F_{n+2}+\frac{1}{2}\right)\right\rfloor-2$$

Therefore:

$$n=\left\lfloor\log_\varphi\left(\sqrt{5}\cdot\left\lceil 2^{\mathrm{h}-1}-2^{-\mathrm{h}-1}\right\rceil+\frac{1}{2}\right)\right\rfloor-2\ \text{with}\ \varphi=\frac{1+\sqrt{5}}{2}\qquad\text{(D1)}$$

Repeating the same steps for odd n we obtain:

$$n=\left\lfloor\log_\varphi\left(\sqrt{5}\cdot\left\lceil 2^{\mathrm{h}-1}+2^{-\mathrm{h}-1}\right\rceil+\frac{1}{2}\right)\right\rfloor-2\ \text{with}\ \varphi=\frac{1+\sqrt{5}}{2}\qquad\text{(D2)}$$

## References


1. Devaney, R. *A First Course in Chaotic Dynamical Systems*; Publisher: The Perseus Books Group, Reading, MAUnited States, 1992.
2. Strogatz, S. *Non-Linear Dynamics and Chaos*. Publisher: The Perseus Books Group, MAUnited States, 1994.
3. Meyers, R.A. *Encyclopedia of Complexity and Systems Science*, Publisher: Springer, New York, 2009.
4. Arnold, V. I.; Avez, A. *Ergodic Problems of Classical Mechanics*; Publisher: Benjamin, New York , 1968.
5. Cornfeld, I.; Fomin, S.; Sinai, Y. *Ergodic Theory;* Publisher: Springer-Verlag, 1982.
6. Prigogine, I. *From Being to Becoming*; Publisher: Freeman, New York, 1980.
7. Katok, A.; Hasselblatt, B. *Introduction to the Modern Theory of Dynamical Systems*; Publisher: Cambridge University Press, Cambridge, UK , 1995.
8. Lasota, A; Mackey, M. *Chaos, Fractals, and Noise*, Publisher: Springer-Verlag, New York, 1994.
9. Lighthill, J. The recently recognized failure of predictability in Newtonian dynamics, *Proc. Roy. Soc. London,*1986, A407, 35–50.
10. Shannon, C. Communication Theory of Secrecy Systems. *Bell System Technical Journal*, 1949, 28, Issue 4, 656–715.
11. Shannon, C.; Weaver W. The Mathematical Theory of Communication, *University of Illinois Press Urbana*, 1949, Ill.
12. Hopf, E. On Causality, Statistics and Probability, *J. Math. and Phys.*, **1934**, *13*, 51–102.

13. Smale, S. Differentiable dynamical systems, *Bulletin of the American Mathematical Society,* **1967**, *73*, 747–817.
14. Smale, S. Finding a horseshoe on the beaches of Rio, *Mathematical Intelligencer 20*, 1998, 39–44.
15. Vivaldi, F. The arithmetic of Chaos. *Chaos, Noise and Fractals* , **1989**, *3*, 187–199.
16. Dyson, F.J.; Falk, H. Period of a discrete cat mapping. *Am Math Monthly,* 1992, 2(99), 603–14.
17. Akritas, P.; Antoniou, I.; Pronko, G. On the Torus Automorphisms: Analytic Solution, Computability and Quantization, *Chaos, Solitons and Fractals*, 2001, 12, 2805–2814.
18. Antoniou, I.; Tasaki, S. Generalized spectral decomposition of the ß-adic baker's transformation and intrinsic irreversibility, *Physica A*, **1992**, *190*, 303–329.
19. Antoniou, I.; Tasaki, S. Generalized spectral decomposition of mixing dynamical systems, *Int. J. Quantum Chemistry*,**1993**, *46*, 425–474.
20. Antoniou, I.; Qiao, B.; Suchanecki, Z. Generalized Spectral Decomposition and Intrinsic Irreversibility of the Arnold Cat Map, *Chaos, Solitons and Fractals*,**1997**, *8*, 77 – 90.
21. Guan, Z.H.; Huang, F.; Guan, W. Chaos-based image encryption algorithm, *Physics Letters A*, 2005, 346, Issues 1–3, 153–157.
22. Xiao, D.; Liao, X.; Wei, P. Analysis and improvement of a chaos-based image encryption algorithm. *Chaos, Solitons & Fractals*,2009, 40, Issue 5, 2191–2199.
23. Kocarev, L.; Lian, S. *Chaos-Based Cryptography. Theory, Algorithms and Applications,Studies in Computational Intelligence*, Publisher: Springer, Berlin , 2011.
24. Kocarev, L.; Sterjev, M.; Amato P. RSA Encryption Algorithm based on Torus Automorphisms, *IEEE, ISCAS*, 2004, IV, 577–580.
25. Kocarev, L.; Tasev, Z.; Makraduli, J. Public-Key Encryption and Digital-Signature Schemes Using Chaotic Maps, 16th European Conference on Circuits Theory and Design, September 1 – September 4, 2003, Krakow, Poland, ECCTD 2003.
26. Li, S. Analyses and New Designs of Digital Chaotic Ciphers, Ph.D. thesis, School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an, China, 2003.
27. Pesin, Ya.B. Characteristic Lyapunov exponents and smooth ergodic theory, *Russ. Math. Surv.*, 1977, 32:4, 55–112.
28. Jiancheng, Z.; Rabab, K.W.; Dongxu, Q. A new digital image scrambling method based on Fibonacci numbers*, IEEE*, 2004,VIII, 965.
29. Graham, R.L.; Knuth, D.E.; Patashnik, O. *Concrete Mathematics;* Publisher: Addison-Wesley, Reading , MAUnited States, 1994.
30. Delfs, H.; Knebl, H. *Introduction to Cryptography Principles and Applications*, 2nd ed; Publisher: Springer, Berlin, 2007.
31. Bergum, G.E.; Philippou, A.; Horadam; Alwyn, F; Volume 4 Proceedings of 'The Fourth International Conference on Fibonacci Numbers and Their Applications', Wake Forest University, N.C., U.S.A., July 30–August 3, 1990