Cryptography with Chaos

George Makris, Ioannis Antoniou

Mathematics Department, Aristotle University, 54124, Thessaloniki, Greece
E-mail: geormak@hotmail.com
Mathematics Department, Aristotle University, 54124, Thessaloniki, Greece
E-mail: iantonio@math.auth.gr

Abstract: We implement Cryptography with Chaos following and extending the original program of Shannon with 3 selected Torus Automorphisms, namely the Baker Map, the Horseshoe Map and the Cat Map. The corresponding algorithms and the software (chaos_cryptography) were developed and applied to the encryption of picture as well as text in real time. The maps and algorithms may be combined as desired, creating keys as complicated as desired. Decryption requires the reverse application of the algorithms.

Keywords: Cryptography, Chaos, image encryption, text encryption, Cryptography with Chaos.

1. Chaotic Maps in Cryptography

Chaotic maps are simple unstable dynamical systems with high sensitivity to initial conditions [Devaney 1992]. Small deviations in the initial conditions (due to approximations or numerical calculations) lead to large deviations of the corresponding orbits, rendering the long-term forecast for the chaotic systems intractable [Lighthill 1986]. This deterministic in principle, but not determinable in practice dynamical behavior is a local mechanism for entropy production. In fact Chaotic systems are distinguished as Entropy producing deterministic systems. In practice the required information for predictions after a (small) number of steps, called horizon of predictability, exceeds the available memory and the computation time grows superexponentially. [Prigogine 1980, Strogatz 1994, Katok, ea 1995, Lasota, ea 1994, Meyers 2009].

Shannon in his classic 1949 first mathematical paper on Cryptography proposed chaotic maps as models - mechanisms for symmetric key encryption, before the development of Chaos Theory. This remarkable intuition was based on the use of the Baker’s map by Hopf in 1934 as a simple deterministic mixing model with statistical regularity. The Baker’s Map is defined below and the mixing character is presented in figure 1:

\[ B : [0,1) \times [0,1) \to [0,1) \times [0,1): \begin{cases} x' = \begin{cases} \frac{2x}{2} \quad x \in \left[0, \frac{1}{2}\right) \\ \frac{2x - 1}{2} \quad x \in \left[\frac{1}{2}, 1\right) \end{cases} \\ y' = \begin{cases} \frac{y}{2} \quad y \in \left[0, \frac{1}{2}\right) \\ \frac{y + 1}{2} \quad y \in \left[\frac{1}{2}, 1\right) \end{cases} \end{cases} \]
The reverse transformation:

\[ B^{-1} : [0,1] \times [0,1] \rightarrow [0,1] \times [0,1] : \begin{cases} x \rightarrow \frac{x + 1}{2} \mod 1 \\ y \rightarrow \frac{2y - 1}{2} \end{cases} \]

Fig. 1: Baker Map

The Entropy production theory of Chaotic maps was developed later by Kolmogorov and his group [Arnold, Avez 1968, Katok, ea 1995, Lasota, ea 1994]. Baker’s map is the simplest example of chaotic automorphisms with constant Entropy production equal to one bit at every step and has served as toy model for understanding the problem of Irreversibility in Statistical Mechanics [Prigogine 1980]. Shannon observed that using chaotic maps, encryption is achieved via successive mixing of the initial information which is “spread” all over the available state space. In this way it is becoming exponentially hard to recover the initial message without knowing the reverse transformation.

A variation of the transformation of Baker Map is the Horseshoe Map [Smale 1967, Smale 1998], with the same Entropy production defined below and the mixing character presented in figure 2:

\[ H : [0,1] \times [0,1] \rightarrow [0,1] \times [0,1] : \]

\[ H(x, y) = \begin{cases} 2x, y & x \in [0, \frac{1}{2}] \\ 2 - 2x, \frac{2 - y}{2} & x \in [\frac{1}{2}, 1] \end{cases} \]

The reverse transformation:

\[ H^{-1}(x, y) = \begin{cases} \frac{x}{2}, 2y & y \in [0, \frac{1}{2}] \\ \frac{2 - x}{2}, 2(1 - y) & y \in [\frac{1}{2}, 1] \end{cases} \]
Both Baker’s Map and the Horseshoe Map belong to the general class of torus automorphisms. The well-known Cat Map introduced by Arnold in 1968 which is a torus automorphism a much stronger mix than two previous ones. The Cat Map is defined below and the mixing character is presented in figure 3:

\[
\begin{bmatrix}
    x' \\
    y'
\end{bmatrix} = A \begin{bmatrix}
    x \\
    y
\end{bmatrix} (\text{mod } N) = \begin{bmatrix}
    1 & p \\
    q & pq + 1
\end{bmatrix} \begin{bmatrix}
    x \\
    y
\end{bmatrix} (\text{mod } N)
\]

The numerical analysis of the Cat Map shows interesting periodicity in the state space discretization [Vivaldi 1989]. Although the Cat Map and the torus automorphisms admit analytical solution, computability does not increase significantly. [Akritas, ea 2001]. Statistical estimates for the transformation of Baker Map and the Cat Map are possible through the spectral analysis [Antoniou and Tasaki 1992, Antoniou, ea 1997, Antoniou and Tasaki 1993].

From Pesin’s 1977 Formula, the entropy of the Cat Map is:

\[
\log_2 \left( \frac{3 + \sqrt{5}}{2} \right) \square 1.39
\]

Following Shannon’s idea, encryption is achieved by entropy producing (chaotic) maps like the torus automorphisms, via successive mixing of the initial information which is “spread” all over the available state space. In this way it is becoming exponentially hard to recover the initial message without knowing the
reverse transformation. Most applications of Chaos cryptography with 2-dimensional maps deal with image encryption [Guan D. ea, 2005, Xiao G. ea 2009]. We also found results on text encryption [Kocarev, ea 2003, Kocarev, ea 2004, Kocarev and Lian 2011, Li 2003]. We shall show how encryption of texts can also be achieved with chaotic maps.

2. Text Encryption and Decryption by Torus Automorphisms

The text Cryptography by Torus Automorphisms involves 3 steps:

Step 1: Place the text in a 2-dimensional table so that each array element is a character.

Step 2: Apply the selected transformations on the table for a number of steps specified by the key.

Step 3: Convert the modified table from step 2 in the text.

The decryption process is equally simple for anyone who holds the key. Simply follow the steps backwards and use inverse transformations to the same number of steps.

We propose 2 algorithms for the implementation of the text cryptography:

Algorithm 1:

Step 1: Count all characters of text including line breaks (=N_i)

Step 2: If N_i is not a perfect square of an integer, then find the smallest integer M > N_i so that M is a perfect square. If the N_i is a perfect square integer number then set M=N_i.

Step 3: Set N = \sqrt{M}

Step 4: Create a character table (N\times N) and place the characters of the text inside the table, putting also the special characters newline (enter) in a position in the table.

Step 5: If there are empty cells at the end of the table place the spaces in these (cells).

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

1) The number of rows and columns of the table depends on the length of the text only.

2) The number of lines of characters changes during the encryption because all the special characters like “enter” are involved in encryption.

Example:

Cryptography with chaos
George Makris, Ioannis Antoniou
Thessaloniki 54124
Greece.

The above text has 82 characters. We need a 10\times 10 table to fit the text in table (100 is the minimal encoding length).
Algorithm 2:
Step 1: Count the number of lines (NL) of the text.
Step 2: Count the number of letters of each line.
Step 3: Find the $M_1 = \max \{\text{the number of letters of each line}\}$.
Step 4: Set $N = \max \{\text{NL, } M_1\}$
Step 5: Create a character table (NxN)
Step 6: Place each character in text in the table so that it corresponds to each line of text in the corresponding row of the table. Put the special character space (‘ ’) in all the blank cells.

So we create a NxN table of characters with the properties:
1) The number of rows and columns of the table defined by the structure and the length of the text.
2) The number of lines of characters does not change in encryption because gaps were placed on each line so that all lines have the same number of characters.

For the same example we have:

<table>
<thead>
<tr>
<th>Cryptography with chaos</th>
<th>23 characters</th>
</tr>
</thead>
<tbody>
<tr>
<td>George Makris, Ioannis Antoniou</td>
<td>31 characters</td>
</tr>
<tr>
<td>Thessaloniki 54124</td>
<td>18 characters</td>
</tr>
<tr>
<td>Greece.</td>
<td>07 characters</td>
</tr>
</tbody>
</table>

Lines $NL = 4$
$M_1 = \max\{23,31,18,07\} = 31$
$N = \max\{4,31\} = 31$
Examples of text and image encryption are presented in the appendices.

3. Software Development for the implementation of "Cryptography with Chaos"

The software for the algorithms was developed with Java, as this language is independent of the operating system and platform. Moreover the Java programs run on Windows, Linux, Unix and Macintosh, mobile phones, Ipads, Playstations and other game consoles without any modification like compilation or changing the source code for each different operating system.

The software developed (chaos_cryptography) has a graphical user interface and is very simple and user friendly (figure 4).

The user may encrypt / decrypt images and texts. The user may use any of the above chaotic maps with one or the other algorithm or any combination for more difficult deciphering.

Window dialogs alert the user in case of any errors in the procedure.

The developed libraries (classes) can be used by any other software and application.

Fig. 4: chaos_cryptography application (main window)
4. Concluding Remarks

Shannon Cryptography indices for chaos cryptography are summarized in the table below.

<table>
<thead>
<tr>
<th>Shannon Cryptography indices</th>
<th>Cryptography with Chaos</th>
</tr>
</thead>
<tbody>
<tr>
<td>Required degree of cryptographic security</td>
<td>High</td>
</tr>
</tbody>
</table>
| Key Length | Small  
The key is the selected transformations and the number of iterations that apply each transformation. |
| Practical implementation of the encryption/decryption | Depends on the size of the text. Generally, permutation is a faster method than the replacement. |
| Growth of the encrypted text | No growth in the case of images. Small growth in the case of texts, due to “spaces” only |
| Error Propagation | In case of images, pixel errors propagate, are preserved in the reconstructed images without influencing the decryption. In case of text, errors may rendering text decryption practically impossible. |

The key length includes the map definition, the number of iterations and the parameters of the specific map. The proposed encryption algorithms are “MonoBlock” ciphers based on permutations, however they are neither steams nor block ciphers. The Key is very small and does not depend on the size of text to be encrypted (block).

For example, the specific key for encryption algorithm (Baker, Cat, Horseshoe) has a size 4 (Table 2x2). In classical permutation algorithms to encrypt a text with N characters (MonoBlock, size of the block = N) a key size N is required which is the size of the Block.

The innovations of this work are summarized as follows:

a) The application of Cryptography with Chaos to content with texts and images.

b) The construction of examples of a new class on ciphers, namely the Mono–Block Ciphers as a third class beyond the Block Ciphers and the Stream Ciphers.

c) The key is completely independent from the length of the block that is encrypted and it is very small compared to the key of the classic permutation algorithms which is equal to the length of the block.

d) in the developed algorithms the key cannot operate if some small part of the document is lost.

Chaos Cryptography has only the disadvantage of all systems of symmetric cryptography, namely the safe transport of the key.

In this paper three of the most famous chaotic maps were investigated. The proposed algorithms can be adapted to other chaotic maps.
References


**Appendix A : Text encryption**

<table>
<thead>
<tr>
<th>Iterations</th>
<th>Baker Map</th>
<th>Horseshoe Map</th>
<th>Cat Map</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>oghr acpheiosr.g els TAhnetsiGkri e e5c Chryy pwtia oMsa kGroomanninoi us4a112o4ne.</td>
<td>c5e eirkGistenhA T sle g,rsoieh pca rhtgoChryy pwtia oMsa kGroomanninoi us4a112o4ne.</td>
<td>C 4n _ghehr iTA nlayy Gkhiet o p r e soMswt o e50aa iGg csnmkrmehr 4ailoiio a e1.2uousrcp</td>
</tr>
<tr>
<td>t=1</td>
<td>n4 o 2uriGokn iaiottwgpu resi o sTArAi G k r i c 5 e esltee ngh,hCphcreay yao ooaMsnse4 a.1 l</td>
<td>n4 o 2uriGokn iaiottwgpu resi o sTArAi G k r i c 5 e esltee ngh,hCphcreay yao ooaMsnse4 a.1 l</td>
<td>C2ih os k he oircaw trnh uo san ilr4s a4ni ea ns r akoety y ceG so tp1nt geo Tg iroc 5MpGAh.</td>
</tr>
<tr>
<td>t=2</td>
<td>epte5hscwo Gt riouh s r rktT ryA iya,M1n sln h,eang ee a i pn k2o eot i4gs iC Gh aseo 4o oa</td>
<td>epte5hscwo Gt riouh s r rktT ryA iya,M1n sln h,eang ee a i pn k2o eot i4gs iC Gh aseo 4o oa</td>
<td>epte5hscwo Gt riouh s r rktT ryA iya,M1n sln h,eang ee a i pn k2o eot i4gs iC Gh aseo 4o oa</td>
</tr>
<tr>
<td>t=3</td>
<td>eerp5hscwo Gt riouh s r rktT ryA iya,M1n sln h,eang ee a i pn k2o eot i4gs iC Gh aseo 4o oa</td>
<td>eerp5hscwo Gt riouh s r rktT ryA iya,M1n sln h,eang ee a i pn k2o eot i4gs iC Gh aseo 4o oa</td>
<td>eerp5hscwo Gt riouh s r rktT ryA iya,M1n sln h,eang ee a i pn k2o eot i4gs iC Gh aseo 4o oa</td>
</tr>
</tbody>
</table>
### Appendix B: Image Encryption

<table>
<thead>
<tr>
<th>Iterations</th>
<th>Baker Map</th>
<th>Horseshoe Map</th>
<th>Cat Map</th>
</tr>
</thead>
<tbody>
<tr>
<td>t=0</td>
<td><img src="image1.png" alt="Image" /></td>
<td><img src="image2.png" alt="Image" /></td>
<td><img src="image3.png" alt="Image" /></td>
</tr>
<tr>
<td>t=1</td>
<td><img src="image4.png" alt="Image" /></td>
<td><img src="image5.png" alt="Image" /></td>
<td><img src="image6.png" alt="Image" /></td>
</tr>
<tr>
<td>t=2</td>
<td><img src="image7.png" alt="Image" /></td>
<td><img src="image8.png" alt="Image" /></td>
<td><img src="image9.png" alt="Image" /></td>
</tr>
<tr>
<td>t=3</td>
<td><img src="image10.png" alt="Image" /></td>
<td><img src="image11.png" alt="Image" /></td>
<td><img src="image12.png" alt="Image" /></td>
</tr>
<tr>
<td>t=4</td>
<td><img src="image13.png" alt="Image" /></td>
<td><img src="image14.png" alt="Image" /></td>
<td><img src="image15.png" alt="Image" /></td>
</tr>
<tr>
<td>t=5</td>
<td><img src="image16.png" alt="Image" /></td>
<td><img src="image17.png" alt="Image" /></td>
<td><img src="image18.png" alt="Image" /></td>
</tr>
<tr>
<td>t=6</td>
<td><img src="image19.png" alt="Image" /></td>
<td><img src="image20.png" alt="Image" /></td>
<td><img src="image21.png" alt="Image" /></td>
</tr>
</tbody>
</table>