International Journal of Scientific and Innovative Mathematical Research (IJSIMR)
Volume 3, Issue 9, September 2015, PP 49-59
ISSN 2347-307X (Print) & ISSN 2347-3142 (Online)
www.arcjournals.org

Steganography Based on Chaotic Torus Automorphisms

G. Makris
Complex Systems Analysis Laboratory
Mathematics Department
Aristotle University of Thessaloniki
Thessaloniki, Greece
geormak@gmail.com

I. Antoniou
Complex Systems Analysis Laboratory
Mathematics Department
Aristotle University of Thessaloniki
Thessaloniki, Greece
iantonio@math.auth.gr

Abstract: *Secure transmission of information is one of the big challenges of knowledge society. Steganography is the transmission of secret messages embedded within other covering messages which conceal the existence of underlying messages (steganosis). In cryptography however, the presence of secret messages is known. We develop a new Steganography method based on Chaos, for concealing texts or images embedded within other transmitted images. Chaotic Torus Automorphisms are employed as mathematical randomization mechanisms for: i) the selection of the cover image pixels where the initial message (text or image) is inserted, ii) the encryption of the initial message (text or image). We have developed the software for efficient implementation of the algorithms in real-time.*

Keywords: *Steganography, Cryptography, Chaos, Torus Automorphisms.*

1. INTRODUCTION

Information hiding techniques are receiving much attention in the present Knowledge Society [1]. Cryptography and Steganography are the main ways for secure Communication of sensitive data. The word Cryptography, from the Greek words κρυπτός (hidden, secret) and γράφειν (writing), means the conversion of Messages to apparent nonsense, so that no one, apart from the intended recipient who holds the decoding technique, may recover the original message. The word Steganography, from the Greek words στεγανός (covered, concealed, protected) and γράφειν (writing), means hiding Messages within other Cover Messages, so that no one, apart from the intended recipient, may suspect the existence of the hidden message. The embedded message within the Cover message is called Steganogram and the construction of the Steganogram is called Steganosis. The recovery of the original message from the received Steganogram follows the reverse procedure. The term Message means according to information theory any digital content-dataset (text, image, sound, video and program).

The differences of Steganography from Cryptography are summarized in Table 1.

Table 1. *Summary of the differences between Steganography and Cryptography.*

Steganography	Cryptography
The observer is not aware that a message passing	The observer is aware that a message is passing but unauthorized access is prevented
Less known technology	Common technology
Once detected, the message is known	The More resistant the algorithms to attacks, The higher the computational cost for cracking
The secret message is not altered	The secret message is significantly altered

Steganography may include Cryptography by inserting an encrypted message into the Cover Message, thus using the advantages of both ways of securing Communication. Steganography involves three Stages, illustrated in Figure 1:

Stage I: **Steganosis:** Preparation of the Steganogram by Embedding the Message into the Cover Message.

Stage II: **Transmission** of the Steganogram through the Communication Channel.

Stage III: **Recovery** of the Message from the transmitted Steganogram by the Recipient.

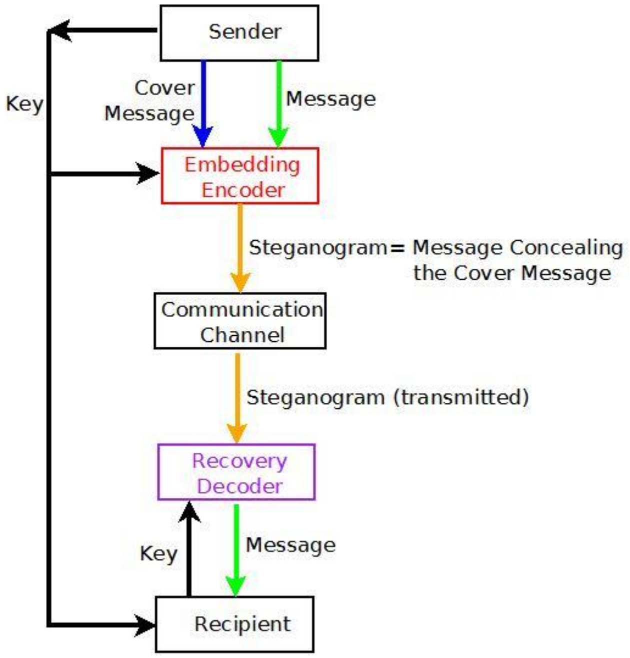


Figure 1. *Steganography Diagram*

Stage II is analysed in the context of standard Information Theory [2-4] and is not specific to Steganography. Steganograms may be transmitted as postcards, digital images in emails, websites, smartphones. Stages I and III are realized by several techniques of Steganography. There are three main methods in the present literature [1, 5-10], namely:

- Least Significant Bit replacement (LSB Stegranography)
- Masking and Filtering
- Algorithms and Transformations

The idea of the Least Significant Bit Steganography is to embed the bits of the Secret Message directly into the least significant bit of the Cover, so that the image is minimally distorted (the original pixel values are not greatly modified) and the modified locations is not easy to identify (irregular placement). In the Algorithms and Transformations methods a transformed cover message (compressed) is modified. The Masking and Filtering methods insert the Message without modifying the Cover, as in the case of Watermarks. Several Steganography Softwares are available in: http://www.jjtc.com/Steganography/tools.html.

The goal of this work is to employ Chaotic Torus Automorphisms, as effective and efficient mathematical randomization mechanisms for Messages represented as 2-dimensional Grids, for the realization of LSB Steganography with Message Encryption. After presenting the 5 parts of the Method in section 2, we present the Results with a simple example with Algorithms implementated in Java, in section 3.

2. Method

The method of constructing Steganography based on Chaotic Torus Automorphisms is presented in 5 parts: 1) The framework of Algorithm of LSB Steganography, 2) The Chaotic Torus Automorphisms, being the underlying Mechamism,3) The Locations Specification with Chaotic Torus Automorphisms, 4) The Encryption and Decryption with Chaotic Torus Automorphisms 5) The Embedding and Recovery of Messages. The 5 parts are discribed below.

2. 1 The Least Significant Bit Steganography

We present the details of Stages I and III of LSB Steganography in an algorithmic way:

Stage I: Steganosis (LSB Steganography)

Step I.0: **Input**

Step I.0.1: Specify the Message to be transmitted.

Step I.0.2: Select the Cover Message

Step I.0.3: Select the Key

Step I.1: **Encryption:** Encrypt the Message

Step I.2: **Locations:** Specify the Locations of the Cover Message, where the Message is to be inserted.

Step I.3: **Insertion:** Insert the (Encrypted) Message (Step I.1) into the Specified Locations (Step I.2) of the Cover Message.

Step I.4: **Output:** The embedded Message within the Cover Message (Steganogram).

Stage III: Recovery (LSB Steganography)

Step III.0: **Input**

Step III.0.1: The Steganogram.

Step III.0.2: The Key (selected in Step I.0.3)

Step III.1: **Locations:** Specify the Locations of the Steganogram, where the Message has been embedded.

Step III.2: **Extraction:** Extract the (Encrypted) Message from the Specified Locations (Step III.1) of the Steganogram.

Step III.3: **Decryption:** Decrypt the Extracted Message.

Step III.4: **Output:** The Message.

The Specification of the Locations of the Cover Message, where the Message is to be inserted (Step I.2 and III.1) should be irregular for security. The usual location specification mechanisms of LSB Steganography are Random Number Generators (RNG) providing random sequences for any selected initial seed [5-10]. The Encryption and Decryption of the Message (Steps I.1 and III.3), are realized with Cryptographic methods [11-17]. If Encryption of the Message is not required, Steps I.1 and III.3 are omitted. Chaotic Torus Automorphisms are effective and efficient mathematical randomization mechanisms for the realization of the two main steps of LSB Steganography, namely the specification of the locations of the Cover image (Steps I.2 and III.1) and the Encryption and Decryption (Steps I.1 and III.3). The Encoder for embedding of the Message into the Cover Message and the Decoder for recovery of the concealed Message are analyzed in Fig. 2 and Fig. 3 correspondingly.

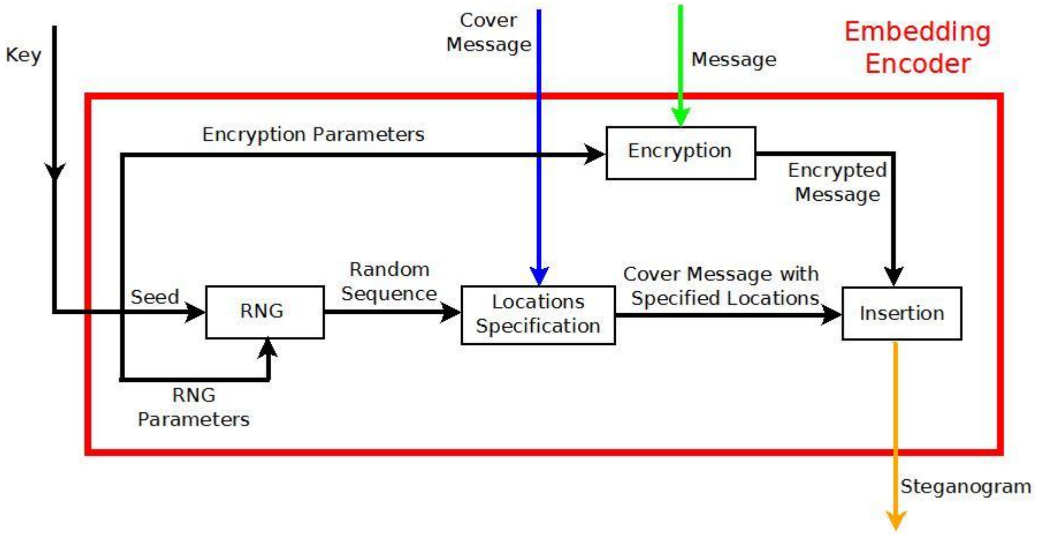


Figure 2. *The Encoder for embedding the Message into the Cover Message in LSB Steganography*.

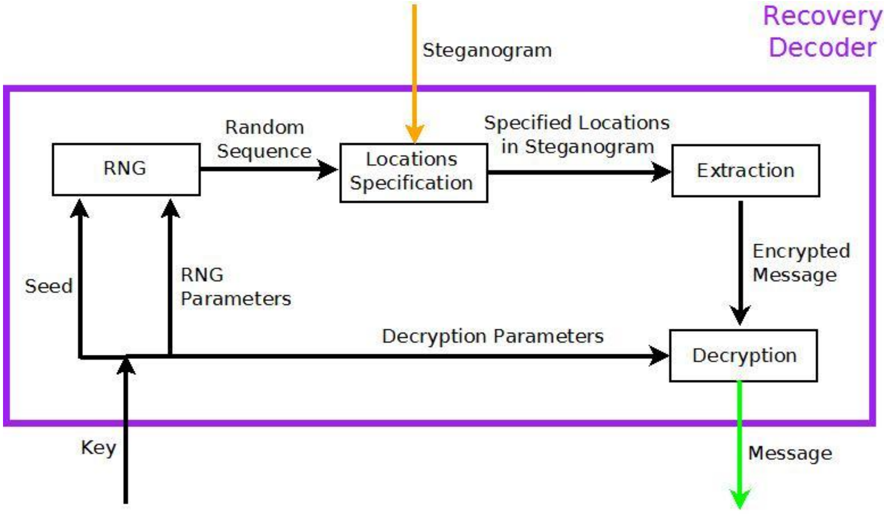


Figure 3. *The Decoder for recovery of the concealed Message received as a Steganogram in LSB*

2.2 Chaotic Torus Automorphisms

Chaotic maps are simple unstable dynamical systems with high sensitivity to initial conditions [18-20]. 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. This deterministic in principle, but not determinable in practice, dynamical behavior is in fact a local mechanism for Entropy Production. Chaotic systems are statistically characterized as Entropy producing deterministic systems [20-25]. In practice the required information for predictions after a (small) number of steps, called horizon of predictability [26], exceeds the available memory. As result, the computation time grows superexponentially.

Shannon in his classic 1949 mathematical paper [2, 27] on Cryptography proposed chaotic maps as models - mechanisms for symmetric key encryption. More specifically he used the Baker's map, introduced earlier (1934) by Hopf [28], as a simple deterministic mixing model with statistical regularity producing 1bit per iteration. Of course neither Shannon, nor Hopf used the term Chaos which emerged forty years later [18-20].

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

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

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

The matrix of Baker's Map is:

$$A = \begin{bmatrix} 2 & 0 \\ 0 & 1/2 \end{bmatrix} \tag{2}$$

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

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = A \begin{bmatrix} x \\ y \end{bmatrix} \bmod N \; = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \bmod N \tag{3}$$

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 periods have been related to the grid size N [29-37].

We shall apply Chaotic Torus Automorphisms to the specification of the locations of the Cover image (Steps I.2 and III.1) and for Encryption and Decryption (Steps I.1 and III.3).

We select the Torus Automorphisms (3) with $\mathrm{ad}-\mathrm{bc}=1$, ie with matrix [38-39]:

$$A=\begin{bmatrix} a & 1 \\ ad-1 & d \end{bmatrix} \qquad (4)$$

The other class of Torus Automorphisms (3) with $\mathrm{ad}-\mathrm{bc}=-1$ can also be applied in the same way with similar results.

2.3 Locations Specification with Chaotic Torus Automorphisms

The Specification of the Locations of the Cover Message, where the Message is to be inserted (Step I.2 and III.1) should be irregular for security. The usual location specification mechanisms of LSB Steganography are Random Number Generators (RNG) providing random sequences for any selected initial seed [40-44]

The usual mathematical Random Number Generators are simple chaotic (entropy producing) maps on real intervals. The simplest non-linear chaotic maps are constructed by applying the modulo operation on linear maps (progressions) [45]:

$$x_{t+1} = \mathrm{a}x_t + c \ \ \mathrm{modN},\ 0 \le x_t < N \qquad (5)$$

Every selection of the initial value (seed) x_0 produces a random sequence iteratively.

We construct random sequences using 2 dimensional Chaotic Automorphisms with matrix (4), instead of the conventional RNG. The only difference is that the initial condition (Seed) is 2-dimensional: (x_0, y_0). The realization of Random Number Generators from Chaotic Torus Automorphisms, is summarized as follows:

- The user is selecting the desired length ℓ of the random sequence.
- We specify the parameters a, d of (4) so that the Chaotic the printed sequences have period T>ℓ [38].
- The produced sequences of length ℓ are aperiodic with uniform distribution

The user may also specify the parameters a, d of (4), so that the Automorphisms have moreover any desired Entropy Production [39].

2.4 Encryption and Decryption with Chaotic Torus Automorphisms

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 process involves 7 steps, summarized below:

Algorithm A: Encryption

```
Step A.0: Input
    Step A.0.1: Specify the Message (Text or/and Image)
    Step A.0.2: Specify the parameters $a,d \in \mathbb{Z}$, of (4)
    Step A.0.3: Specify the number of iterations n=1,2,3,… of (4)
Step A.1: If the Message is image with equal height and width, then go to step A.4.
          Else add pixels so that the image has equal height and width and go to
          step A.4
Step A.2: If the Message is text goto to Step A.3
          Else goto step A.4.
```

```
Step A.3: Text Placement
    Step A.3.0: Place the text in a 2-dimensional grid so that each
        array element is a character
    Step A.3.1: Count all characters of text including line breaks (=$N_1$)
    Step A.3.2: If $N_1$ is a perfect square integer number then set M=$N_1$.
        If $N_1$ is not a perfect square of an integer, then find the smallest integer
        M > $N_1$ so that M is a perfect square.
    Step A.3.3: Set $N = \sqrt{M}$
    Step A.3.4: Create a character grid (NxN) and
        place the characters of the text inside the grid
Step A.4: Apply the selected transformations for a number n of iterations on the grid
    for iterations=0 to n-1
        for i=0 to N-1
            for j=0 to N-1
                $\begin{bmatrix} x'_i \\ y'_j \end{bmatrix} = \begin{bmatrix} \mathrm{a} & 1 \\ \mathrm{ad}-1 & \mathrm{d} \end{bmatrix} \cdot \begin{bmatrix} x_i \\ y_j \end{bmatrix} \; mod\ N$
Step A.5: If the Message is image then go to step A.7.
Step A.6: Convert the modified table from of step A.2 to text.
Step A.7: Return the transformed grid
```

We remark that for text placement in Step A.3, we create a NxN grid of characters depending on the length of the text only.

The Encryption Key is the selected sequence of transformations:

$$Encryption\,Key = \ \mathrm{a}_1,\ \mathrm{d}_1,\ n_1\ ,\ \mathrm{a}_2,\ \mathrm{d}_2,\ n_2\ ,...,\ \mathrm{a}_k,\ \mathrm{d}_k,\ n_k\ , k = 1,2,3,... \qquad (6)$$

Applying algorithm A for the encrypted Message with the sequence of inverse transformations in the reverse order we obtain the original message. We replace the matrix $\begin{bmatrix} \mathrm{a} & 1 \\ \mathrm{ad}-1 & \mathrm{d} \end{bmatrix}$ of step A.4 with the inverse matrix $\begin{bmatrix} \mathrm{a} & 1 \\ \mathrm{ad}-1 & \mathrm{d} \end{bmatrix}^{-1} = \begin{bmatrix} \mathrm{d} & -1 \\ 1-\mathrm{ad} & \mathrm{a} \end{bmatrix}$. The Decryption Key is simply the reverse of the Encryption Key:

$$Decryption\,Key = \ \mathrm{a}_k,\ \mathrm{d}_k,\ n_k\ ,...,\ \mathrm{a}_2,\ \mathrm{d}_2,\ n_2\ ,\ \mathrm{a}_1,\ \mathrm{d}_1,\ n_1\ , k = 1,2,3,... \qquad (7)$$

Therefore, the Decryption process is simple for those who hold the Encryption Key (6), but exponentially hard otherwise.

2.5 Embedding and Recovery

We denote by H,W the Height and the Width of the grid representing the Cover Message and by m the length of the Message. As each Symbol (character or color or pixel or vector) is represented by a given number ν of bits in the selected Symbol Encoding, the Cover Message and the Message are digitally represented by $\nu \cdot H \cdot W$ and $\nu \cdot m$ bits correspondingly.

According to the LSB Steganography, at most one (the Least Significant) digit of the Digital Representation of each symbol in the Cover Message may change. Therefore, Embedding the Message into the Cover message is possible, if and only if:

$$H \cdot W \geq \nu \cdot m \tag{8}$$

If there are several layers of data upon the same H x W grid, formula (8) should be:

$$\textit{number of layers} \ \cdot H \cdot W \geq \nu \cdot m$$

The theoretical maximum distortion of the Cover Message after embedding the secret Message is:

$$\frac{\nu \cdot m}{\textit{number of layers} \ \cdot H \cdot W} \cdot 100\% \tag{9}$$

The number ν of bits per symbol in text encoding in the ASCII representation and images is 8. The number of layers is in RGB representation of image encoding is 3 (one for each color), in CMYK representation of image is 4 (4 colors). Therefore the theoretical maximum distortion is:

$$\frac{8m}{H \cdot W} \cdot 100\% \ \text{ for gray images and texts}$$

$$\frac{8m}{3 \cdot H \cdot W} \cdot 100\% \ \text{ for RGB images}$$

$$\frac{8m}{4 \cdot H \cdot W} \cdot 100\% \ \text{ for CMYK images.}$$

For example, the maximum distortion of a Cover RGB Image 400x600, after embedding a Message of length m=50 by changing only one of the 3 RGB colors is: $\frac{8 \cdot 50}{400 \cdot 600} \cdot 100\% = 0{,}1041\%$. However, if we change all 3 RGB colors, the distortion is one order of magnitude less:

$\frac{8 \cdot 50}{3 \cdot 400 \cdot 600} \cdot 100\% = 0{,}0347\%$.

In practice however, not all bits of the Digital representation of the Cover Message change, because 50% of the bits of the possible locations in the Cover Message are expected to coincide with the Message bits on the average.

Algorithm B: Embedding

```
Step B.0: Input
        Step B.0.0: Specify the number ν of bits per symbol
        Step B.0.1: Specify the Message (Text or/and Image)
        Step B.0.2: Specify the Cover Message
        Step B.0.3: Specify the parameters $a, d \in \mathbb{Z}$, of (4)
        Step B.0.4: Specify the number of iterations n=1,2,3,… of (4)
        Step B.0.5: Specify the Seed (initial condition) $(x_0, y_0)$
Step B.1: Set m=the number of symbols of the Message
Step B.2: Set N=min{ Height of the Cover Message, Width of the Cover Message}
Step B.3: If $N \cdot N \geq \nu \cdot m$ go to Step B.4,
          else go to Step B.0.2 (to specify a Larger Cover Message)
Step B.4: If encryption is desired, then Encrypt the Message using Algorithm A.
Step B.5: Convert each symbol of the Message (Encrypted or not) to the ν-bits digital
           representation.
Step B.6: Compute $\nu \cdot m$ locations of the Cover Message using the map (4) on the grid NxN.
```

Step B.7: Change the least significant digit of the each selected location by inserting a corresponding digit of the Message

Step B.8: Output: Steganogram.

Algorithm C: Extracting

```
Step C.0: Input
    Step C.0.0: Specify the number ν of bits per symbol
    Step C.0.1: Specify the Steganogram
    Step C.0.2: Specify the parameters $a, d \in \mathbb{Z}$, of (4)
    Step C.0.3: Specify the number of iterations n=1,2,3,… of (4)
    Step C.0.4: Specify the Seed (initial condition) $(x_0, y_0)$
Step C.1: Set N=min{ Height of the Steganogram, Width of the Steganogram }
Step C.2: Compute the locations of the Message within the Steganogram using the map (4)
          on the grid NxN
Step C.3: Extract the digits of the computed locations from step C.2
Step C.4: Retrieve the Message
Step C.5: If decryption is desired, then Decrypt the Message using Algorithm A with the
          decryption key (7).
Step C.6: Output: the Message
```

The Embedding Key consists of the parameters $a, d \in \mathbb{Z}$, of (4), the number n of iterations of (4), and the Seed (initial condition) (x_0, y_0). The extracting Key is identical to the Embedding Key for the identifications of the Locations:

$$\textit{Embedding Key} \equiv \textit{Extracting Key} = \ \mathrm{a},\ \mathrm{d},\ n,\ \mathrm{x}_0,\ \mathrm{y}_0 \tag{10}$$

The Steganography Key consists of the Embedding Key (10) and the Encryption Key (6):

$$\begin{gathered}\textit{Steganography Key} = \ \textit{Embedding Key}, \textit{Encryption Key} \\ \textit{Steganography Key} = \ \mathrm{a},\ \mathrm{d},\ n,\ \mathrm{x}_0,\ \mathrm{y}_0,\ \mathrm{a}_1,\ \mathrm{d}_1,\ n_1\ ,\ \mathrm{a}_2,\ \mathrm{d}_2,\ n_2\ ,...,\ \mathrm{a}_k,\ \mathrm{d}_k,\ n_k\end{gathered} \tag{11}$$

3. Results and Discussion

We demonstrate the method in one simple case. The secret Message of n=212 characters is:

“Mathematical Week Thessaloniki March 2012 Ioannis Antoniou Nikolaos Farmakis Georgios Makris & Students of the Mathematics stand Annex mathematical Society Central Macedonia (C) George Makris.!”

The Cover Message has Width W=640 and Height H=480 and is presented in left part of Fig. 4.

The message was encrypted with one Torus Automorphism applying Algorithm A, with parametres a=1, d=4, iterated n=10 number of times. The Encryption Key (6) is: $\textit{Encryption Key} = \ 1,\ 4,\ 10$

After encryption the Message was embedded in the Cover Message using the Torus Automorphism applying Algorithm B with parametres a=5, d=1 and Seed=(x_0,y_0)=(112,13), iterated n=5 number of times. The Embedding Key (10) is: $\textit{Embedding Key} = \ 5,\ 1,\ 5, 112, 13$

Collecting the above together we obtain the Steganography Key (11):

$\textit{Steganography Key} = \ 5,\ 1,\ 5, 112, 13, 1,\ 4,\ 10$

The theoretical distortion of the Cover Message after embedding the secret Message is: $\frac{212 \cdot 8}{640 \cdot 480} \cdot 100\% = 0{,}5521\%$.

The Real distortion is obtained by counting the number of bits actually modified. 911 bits were modified from the 1696 locations. Therefore, the percentage of bits in the Cover image which were actually modified is 0,2965%. In other words the real distortion is 53,70% of the theoretical distortion.

This difference of the real distortion from the theoretical distortion is expected because not all possible locations of Cover Message were modified by the insertion of the Message, as discussed in section 2.5.

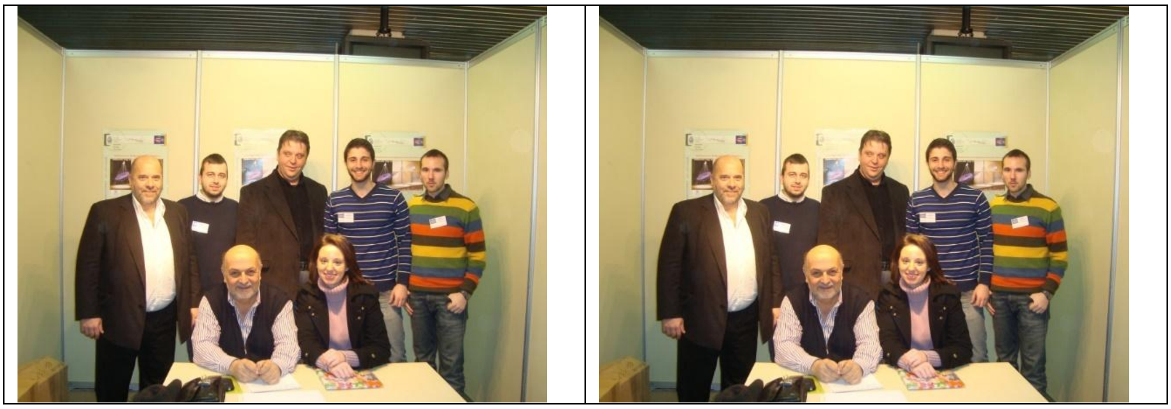


Figure 4. *Cover Image (Left) and Steganogram (Right)*

The computations were implemented by the software "STEGANOGRAPHY-X" which we developed in Java, as this language is independent of the operating system and platform. Moreover 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.

The software has a graphical user interface, it is simple and user friendly. The main window is presented in Fig. 5.

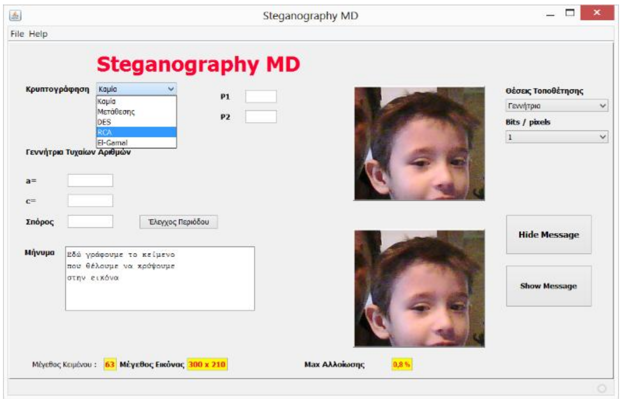
The developed software includes 3 packages (classes), namely:

1) Crypto, implementing the Cryptography Algorithm A.

2) TorusRandomGen, implementing the Random Number Generator with Chaos, section 2.4.

3) Stegano, implementing the Embedding and the Extraction Algorithms B and C.

These packages may be used jointly or separately by any other Java Application.



Κρυπτογράφηση: Καμία

Θέσεις Τοποθέτησης: Γεννήτρια

Bits / pixels: 1

Μήνυμα: Εδώ γράφουμε το κείμενο που θέλουμε να κρύψουμε στην εικόνα

Μέγεθος Κειμένου : 63 Μέγεθος Εικόνας 300 x 210 Max Αλλοίωσης 0,8 %

Figure 5. *The main window of STEGANOGRAPHY-X*

4. Conclusion

The constructed Steganography Algorithm, for secure message transfer, is ready for development for sending messages (e-mails, cellphones, pcs, social networks) and digital authentication (hide copyright details in images, sound, video).

LSB Steganography with Chaotic Automorphisms is also applicable to Algorithms and Transformations methods with minor adaptations, because both Randomization and Cryptography are involved as well as to Masking and Filtering methods for the Cryptography part. We briefly mention the two main advantages and disadvantages of LSB Steganography [1, 5-10]:

Advantage 1: Less chance for degradation of the original image.

Advantage 2: The Hiding capacity is higher, i.e. more information can be stored in an image.

Disadvantage 1: Less robust, the hidden data may be lost with image manipulation.

Disadvantage 2: Hidden data can be easily destroyed by simple attacks on the Steganogram.

Although Histogram Analysis [1, 7] can be employed to detect the possible existence of a hidden message in the Steganogram Image, it is practically impossible to extract the secret message (encrypted or not) for two reasons discussed in section 2.5:

1) about 50% of the bits are modified.

2) the embedding locations are randomly selected, therefore $\nu \cdot m\ !$ attempts are required to extract the Message, if all the bits are modified.

ACKNOWLEDGEMENTS

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

REFERENCES

[1]. J. Fridrich, *Steganography in Digital Media*, Cambridge, U.K.: Cambridge University Press, 2010.

[2]. C. Shannon and W. Weaver, *The Mathematical Theory of Communication*, University of Illinois, U.S.: Press Urbana, 1949.

[3]. T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. Wiley Interscience, 2012.

[4]. D. MacKay, *Information Theory, Inference and Learning Algorithms*, Cambridge, U.K.: Cambridge University Press, 2005.

[5]. Artz D., Digital Steganography: Hiding data within data. IEEE Internet Computing. 5(3), 75-80 (2001).

[6]. F.L. Bauer, *Decrypted Secrets: Methods and Maxims of Cryptology*, 3rd ed. New York, U.S.: Springer-Verlag, 2002.

[7]. R. Gonzales and R. Woods, *Digital Image Processing*, Addison Wesley, 1993.

[8]. Johnson N.F. and Jajodia S., Exploring steganography: Seeing the unseen, Computer. 31(2), 26-34 (1998).

[9]. Kolata G., "A mystery unraveled, twice", The New York Times. F1-F6 (April 1998).

[10]. P. Wayner, *Disappearing Cryptography: Information Hiding: Steganography & Watermarking*, 2nd ed. San Francisco, California, U.S.: Morgan Kaufmann, 2002.

[11]. D. Kahn, *Codebreakers: The Story of Secret Writing*. Revised ed., New York, U.S.: Scribner, 1996.

[12]. A. Menezes, P. Oorschot and S. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1996.

[13]. B. Schneier, *Applied Cryptography*, 2nd ed. New York, U.S.: John Wiley & Sons, 1996.

[14]. B. Schneier, *Secrets & Lies: Digital Security in a Networked World*. New York, U.S.: John Wiley & Sons, 2000.

[15]. W. Stallings, *Cryptography and Network Security: Principles and Practice*, 4th ed. Englewood Cliffs, NJ: Prentice Hall, 2006.

[16]. W. Trappe and L.C. Washington, *Introduction to Cryptography with Coding Theory*, 2nd ed, Upper Saddle River, NJ: Pearson Prentice Hall, 2006.

[17]. A. Young and M. Yung, *Malicious Cryptography: Exposing Cryptovirology*. New York, U.S.: John Wiley & Sons, 2004.

[18]. R. Devaney, *A First Course in Chaotic Dynamical Systems*, Reading, MAUnited States: The Perseus Books Group, 1992.

[19]. S. Strogatz, *Non-Linear Dynamics and Chaos*, Reading, MAUnited States: The Perseus Books Group, 1994.

[20]. R.A. Meyers, *Encyclopedia of Complexity and Systems Science*, New York,U.S.: Springer, 2009.

[21]. V. I. Arnold and A. Avez, *Ergodic Problems of Classical Mechanics*, New York, U.S.: Benjamin, 1968.

[22]. I. Cornfeld, S. Fomin and Y. Sinai, *Ergodic Theory*, Springer-Verlag, 1982.

[23]. I. Prigogine, *From Being to Becoming*, New York, U.S.: : Freeman, 1980.

[24]. A. Katok and B. Hasselblatt, *Introduction to the Modern Theory of Dynamical Systems*, Cambridge, U.K.: Cambridge University Press, 1995.

[25]. A. Lasota and M. Mackey, *Chaos, Fractals, and Noise*, New York, U.S.: : Springer-Verlag, 1994.

[26]. Lighthill J., The recently recognized failure of predictability in Newtonian dynamics, Proc. Roy. Soc. London. A407, 35-50 (1986).

[27]. Shannon C., Communication Theory of Secrecy Systems. Bell System Technical Journal. 28(4), 656–715 (1942).

[28]. Hopf E., On Causality, Statistics and Probability, J. Math. and Phys. 13, 51-102 (1934).

[29]. Vivaldi F., The arithmetic of Chaos. Chaos, Noise and Fractals. 3, 187–199 (1989).

[30]. Dyson F.J. and Falk H., Period of a discrete cat mapping. Am Math Monthly. 2(99), 603-14 (1992).

[31]. Akritas P., Antoniou I. and Pronko G., On the Torus Automorphisms: Analytic Solution, Computability and Quantization, Chaos, Solitons and Fractals. 12, 2805-2814 (2001).

[32]. Antoniou I. and Tasaki S., Generalized spectral decomposition of the ß-adic baker's transformation and intrinsic irreversibility, Physica A. 190, 303-329 (1992).

[33]. Antoniou I. and Tasaki S., Generalized spectral decomposition of mixing dynamical systems, Int. J. Quantum Chemistry. 46, 425-474 (1993).

[34]. Antoniou I., Qiao B. and Suchanecki Z., Generalized Spectral Decomposition and Intrinsic Irreversibility of the Arnold Cat Map, Chaos, Solitons and Fractals. 8, 77 – 90 (1997).

[35]. Guan Z.H., Huang F. and Guan W., Chaos-based image encryption algorithm, Physics Letters A. 346(1-3), 153-157 (2005).

[36]. Xiao D., Liao X. and Wei P., Analysis and improvement of a chaos-based image encryption algorithm. Chaos, Solitons & Fractals. 40(5), 2191-2199 (2009).

[37]. Makris G. and Antoniou I., Cryptography with Chaos. Chaotic Modeling and Simulation (CMSIM). 1, 169-178 (2013).

[38]. Makris G. and Antoniou I., Chaos Cryptography: Ralations of Entropy with message length and Period. Chaotic Modeling and Simulation (CMSIM). 4, 571-581 (2013).

[39]. Makris G. and Antoniou I., Chaos Cryptography with prescribed Entropy Production, Conference Proceeding, 2nd International Electronic Conference on Entropy and Its Applications (accepted), 2015.

[40]. Atkinson A.C., "A Family of Switching Algorithms for the Computer Generation of Beta Random Variables.", Biometrika, 66 (1), 141-145 (1979).

[41]. Bailey R.W., "Polar Generation of Random Variates with the t-Distribution.", Mathematics of Computation. 62 (206), 779-781 (1994).

[42]. Bassein S., "A Sampler of Randomness.", Amer. Math. Monthly. 103, 483-490 (1996).

[43]. Bennett D. J. 1998, Randomness. Cambridge, MA: Harvard University Press,.

[44]. I. Deak, *Random Number Generators and Simulation*. New York, U.S.: State Mutual Book & Periodical Service, 1990.

[45]. D. Knuth, *The Art of Computer Programing Volume 2: Seminumerical Algorithms,* 3rd ed., Addison-Wesley, 1998.