A Robust Data Hiding Process Contributing to the Development of a Semantic Web

Jacques M. Bahi, Jean-François Couchot, Nicolas Friot, and Christophe Guyeux*
FEMTO-ST Institute, UMR 6174 CNRS
Computer Science Laboratory DISC
University of Franche-Comté
Besançon, France
{jacques.bahi, jean-francois.couchot, nicolas.friot, christophe.guyeux}@femto-st.fr

** Authors in alphabetic order*

***Abstract*—In this paper, a novel steganographic scheme based on chaotic iterations is proposed. This research work takes place into the information hiding framework, and focus more specifically on robust steganography. Steganographic algorithms can participate in the development of a semantic web: medias being on the Internet can be enriched by information related to their contents, authors, etc., leading to better results for the search engines that can deal with such tags. As media can be modified by users for various reasons, it is preferable that these embedding tags can resist to changes resulting from some classical transformations as for example cropping, rotation, image conversion, and so on. This is why a new robust watermarking scheme for semantic search engines is proposed in this document. For the sake of completeness, the robustness of this scheme is finally compared to existing established algorithms.**

***Keywords*-Semantic Web; Information Hiding; Steganography; Robustness; Chaotic Iterations.**

I. Introduction

Social search engines are frequently presented as a next generation approach to query the world wide web. In this conception, contents like pictures or movies are tagged with descriptive labels by contributors, and search results are enriched with these descriptions. These collaborative taggings, used for example in Flickr [2] and Delicious [1] websites, can participate to the development of a Semantic Web, in which every Web page contains machine-readable metadata that describe its content. To achieve this goal by embedding such metadata, information hiding technologies can be useful. Indeed, the interest to use such technologies lays on the possibility to realize social search without websites and databases: descriptions are directly embedded into media, whatever their formats.

In the context of this article, the problem consists in embedding tags into internet medias, such that these tags persist even after user transformations. Robustness of the chosen watermarking scheme is thus required in this situation, as descriptions should resist to user modifications like resizing, compression, and format conversion or other classical user transformations in the field. Indeed, quoting Kalker in [11], "Robust watermarking is a mechanism to create a communication channel that is multiplexed into original content [...] It is required that, firstly, the perceptual degradation of the marked content [...] is minimal and, secondly, that the capacity of the watermark channel degrades as a smooth function of the degradation of the marked content". The development of social web search engines can thus be strengthened by the design of robust information hiding schemes. Having this goal in mind, we explain in this article how to set up a secret communication channel using a new robust steganographic process called $\mathcal{DI}_3$. This new scheme has been theoretically presented in [5] with an evaluation of its security. So, the main objective of this work is to focus on robustness aspects presenting firstly other known schemes in the literature, and presenting secondly this new scheme and and evaluate its robustness. This article is thus a first work on the subject, and the comparison with other schemes concerning the robustness will be realized in future work.

The remainder of this document is organized as follows. In Section II, some basic reminders concerning the notion of Most and Least Significant Coefficients are given. In Section III, some well-known steganographic schemes are recalled, namely the YASS [17], nsF5 [8], MMx [12], and HUGO [15] algorithms. In the next section the implementation of the steganographic process $\mathcal{DI}_3$ is detailed, and its robustness study is exposed in Section V. This research work ends by a conclusion section, where our contribution is summarized and intended future researches are presented.

II. Most and Least Significant Coefficients

We first notice that terms of the original content x that may be replaced by terms issued from the watermark y are less important than others: they could be changed without be perceived as such. More generally, a *signification function* attaches a weight to each term defining a digital media, depending on its position t.

Definition 1: *A* signification function *is a real sequence* $(u^k)^{k\in\mathbb{N}}$. □

Example 1: *Let us consider a set of grayscale images stored into portable graymap format (P3-PGM): each pixel* *ranges between 256 gray levels, i.e., is memorized with eight bits. In that context, we consider* $u^k = 8-(k \mod 8)$ *to be the* k*-th term of a signification function* $(u^k)^{k\in\mathbb{N}}$*. Intuitively, in each group of eight bits (i.e., for each pixel) the first bit has an importance equal to 8, whereas the last bit has an importance equal to 1. This is compliant with the idea that changing the first bit affects more the image than changing the last one.* □

Definition 2: *Let* $(u^k)^{k\in\mathbb{N}}$ *be a signification function,* m *and* M *be two reals s.t.* $m < M$.

- *The* most significant coefficients (MSCs) *of* x *is the finite vector*

$$u_M = \left(k \;\middle|\; k \in \mathbb{N} \textit{ and } u^k \geqslant M \textit{ and } k \leq\mid x \mid\right);$$

- *The* least significant coefficients (LSCs) *of* x *is the finite vector*

$$u_m = \left(k \;\middle|\; k \in \mathbb{N} \textit{ and } u^k \leq m \textit{ and } k \leq\mid x \mid\right);$$

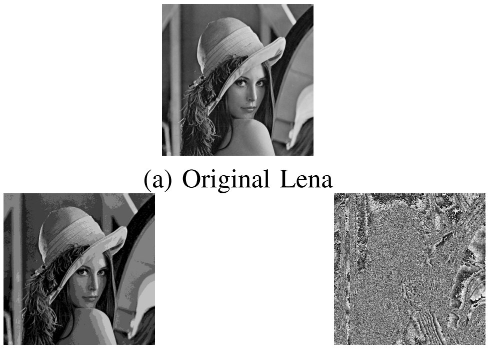
- *The* passive coefficients *of* x *is the finite vector*

$$u_p = \left(k \;\middle|\; k \in \mathbb{N} \textit{ and } u^k \in]m;M[\textit{ and } k \leq\mid x \mid\right).$$

For a given host content x, MSCs are then ranks of x that describe the relevant part of the image, whereas LSCs translate its less significant parts.

Remark 1: *When MSCs and LSCs represent a sequence of bits, they are also called Most Significant Bits (MSBs) and Least Significant Bits (LSBs). In the rest of this article, the two notations will be used depending on the context.* □

Example 2: *These two definitions are illustrated on Figure 1, where the significance function* (u^k) *is defined as in Example 1,* $m = 5$*, and* $M = 6$.



(a) Original Lena

(b) MSCs of Lena

(c) LSCs of Lena (×17)

Figure 1. Most and least significant coefficients of Lena

III. Steganographic schemes

To compare the approach with other schemes, we now present recent steganographic approaches, namely YASS (Cf setc. III-A), nsF5 (Cf setc. III-B), MMx (Cf setc. III-C), and HUGO (Cf setc. III-D). One should find more details in [7].

A. YASS

YASS (*Yet Another Steganographic Scheme*) [17] is a steganographic approach dedicated to JPEG cover. The main idea of this algorithm is to hide data into 8×8 randomly chosen inside $B \times B$ blocks (where B is greater than 8) instead of choosing standard 8×8 grids used by JPEG compression. The self-calibration process commonly embedded into blind steganalysis schemes is then confused by the approach. In the paper [16], further variants of YASS have been proposed simultaneously to enlarge the embedding rate and to improve the randomization step of block selecting. More precisely let be given a message m to hide, a size B, $B \geq 8$, of blocks. The YASS algorithm follows.

1) Computation of m', which is the Repeat-Accumulate error correction code of m.
2) In each big block of size $B \times B$ of cover, successively do:
    a) Random selection of an 8×8 block b using w.r.t. a secret key.
    b) Two-dimensional DCT transformation of b and normalisation of coefficient w.r.t a predefined quantization table. Matrix is further referred to as b'.
    c) A fragment of m' is embedded into some LSB of b'. Let b'' be the resulting matrix.
    d) The matrix b'' is decompressed back to the spatial domain leading to a new $B \times B$ block.

B. nsF5

The nsF5 algorithm [8] extends the F5 algorithm [18]. Let us first have a closer look on this latter.

First of all, as far as we know, F5 is the first steganographic approach that solves the problem of remaining unchanged a part (often the end) of the file. To achieve this, a subset of all the LSB is computed thanks to a pseudo random number generator seeded with a user defined key. Next, this subset is split into blocks of x bits. The algorithm takes benefit of binary matrix embedding to increase it efficiency. Let us explain this embedding on a small illustrative example where a part m of the message has to be embedded into this x LSB of pixels which are respectively a 3 bits column vector and a 7 bits column vector. Let then H be the binary Hamming matrix

$$H = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

The objective is to modify x to get y s.t. $m = Hy$. In this algebra, the sum and the product respectively correspond to the exclusive *or* and to the *and* Boolean operators. If Hx is already equal to m, nothing has to be changed and x can be sent. Otherwise we consider the difference $\delta = d(m, Hx)$

which is expressed as a vector :

$$\delta = \left(\begin{array}{c} \delta_1 \\ \delta_2 \\ \delta_3 \end{array} \right) \text{ where } \delta_i \text{ is 0 if } m_i = Hx_i \text{ and 1 otherwise.}$$

Let us thus consider the jth column of H which is equal to δ. We denote by $\overline{x}^j$ the vector we obtain by switching the jth component of x, that is, $\overline{x}^j = (x_1, \ldots, \overline{x_j}, \ldots, x_n)$. It is not hard to see that if y is $\overline{x}^j$, then $m = Hy$. It is then possible to embed 3 bits in only 7 LSB of pixels by modifying on average $1 - 2^3$ changes. More generally, the F5 embedding efficiency should theoretically be $\frac{p}{1-2^p}$.

However, the event when the coefficient resulting from this LSB switch becomes zero (usually referred to as *shrinkage*) may occur. In that case, the recipient cannot determine whether the coefficient was -1, +1 and has changed to 0 due to the algorithm or was initially 0. The F5 scheme solves this problem first by defining a LSB with the following (not even) function:

$$LSB(x) = \left\{ \begin{array}{l} 1 - x \mod 2 \text{ if } x < 0 \\ x \mod 2 \text{ otherwise.} \end{array} \right. .$$

Next, if the coefficient has to be changed to 0, the same bit message is re-embedded in the next group of x coefficient LSB.

The scheme nsF5 focuses on steps of Hamming coding and ad'hoc shrinkage removing. It replaces them with a *wet paper code* approach that is based on a random binary matrix. More precisely, let D be a random binary matrix of size $x \times n$ without replicate nor null columns: consider for instance a subset of $\{1, 2^x\}$ of cardinality n and write them as binary numbers. The subset is generated thanks to a PRNG seeded with a shared key. In this block of size x, one choose to embed only k elements of the message m. By abuse, the restriction of the message is again called m. It thus remains $x - k$ (wet) indexes/places where the information shouldn't be stored. Such indexes are generated too with the keyed PRNG. Let v be defined by the following equation:

$$Dv = \delta(m, Dx). \tag{1}$$

This equation may be solved by Gaussian reduction or other more efficient algorithms. If there is a solution, one have the list of indexes to modify into the cover. The nsF5 scheme implements such a optimized algorithm that is to say the LT codes.

C. MMx

Basically, the MMx algorithm [12] embeds message in a selected set of LSB cover coefficients using Hamming codes as the F5 scheme. However, instead of reducing as many as possible the number of modified elements, this scheme aims at reducing the embedding impact. To achieve this it allows to modify more than one element if this leads to decrease distortion.

Let us start again with an example with a $[7, 4]$ Hamming codes, *i.e*, let us embed 3 bits into 7 DCT coefficients, $D_1, \ldots, D_7$. Without details, let $\rho_1, \ldots, \rho_7$ be the embedding impact whilst modifying coefficients $D_1, \ldots, D_7$ (see [12] for a formal definition of ρ). Modifying element at index j leads to a distortion equal to ρ_j. However, instead of switching the value at index j, one should consider to find all other columns of H, j_1, j_2 for instances, s.t. the sum of them is equal to the jth column and to compare ρ_j with $\rho_{j_1} + \rho_{j_2}$. If one of these sums is less than ρ_j, the sender has to change these coefficients instead of the j one. The number of searched indexes (2 for the previous example) gives the name of the algorithm. For instance in MM3, one check whether the message can be embedded by modifying 3 pixel or less each time.

D. HUGO

The HUGO [15] steganographic scheme is mainly designed to minimize distortion caused by embedding. To achieve this, it is firstly based on an image model given as SPAM [14] features and next integrates image correction to reduce much more distortion. What follows discuss on these two steps.

The former first computes the SPAM features. Such calculi synthesize the probabilities that the difference between consecutive horizontal (resp. vertical, diagonal) pixels belongs in a set of pixel values which are closed to the current pixel value and whose radius is a parameter of the approach. Thus a fisher linear discriminant method defines the radius and chooses between directions (horizontal, vertical...) of analyzed pixels that gives the best separator for detecting embedding changes. With such instantiated coefficients, HUGO can synthesize the embedding cost as a function $D(X, Y)$ that evaluates distortions between X and Y. Then HUGO computes the matrices of $\rho_{i,j} = \max(D(X, X^{(i,j)+})_{i,j}, D(X, X^{(i,j)-})_{i,j})$ such that $X^{(i,j)+}$ (resp. $X^{(i,j)-}$) is the cover image X where the the (i, j)th pixel has been increased (resp. has been decreased) of 1.

The order of modifying pixel is critical: HUGO surprisingly modifies pixels in decreasing order of $\rho_{i,j}$. Starting with $Y = X$, it increases or decreases its (i, j)th pixel to get the minimal value of $D(Y, Y^{(i,j)+})_{i,j}$ and $D(Y, Y^{(i,j)-})_{i,j}$. The matrix Y is thus updated at each round.

IV. The new steganographic process $\mathcal{DI}_3$

A. Implementation

In this section, a new algorithm which is inspired from the schemes $\mathcal{CIW}_1$ and $\mathcal{CIS}_2$ respectively described in [9] and [10] is presented. Compare to the first one, it is a steganographic scheme, not just a watermarking technique. Unlike $\mathcal{CIS}_2$ which require embedding keys with three strategies, only one is required for $\mathcal{DI}_3$. So compare to

$\mathcal{CIS}_2$ which is also a steganographic process, it is easier to implement for Internet applications especially in order to contribute to a semantic web. Moreover, since $\mathcal{DI}_3$ is a particular instance of $\mathcal{CIS}_2$, it is clearly faster than this one because in $\mathcal{DI}_3$ there is no operation to mix the message on the contrary on the initial scheme. The fast execution of such an algorithm is critical for internet applications.

In the following algorithms, the following notations are used:

Notation 1: *S denotes the embedding and extraction strategy, H the host content or the stego-content depending of the context. LSC denotes the old or new LSCs of the host or stego-content H depending of the context too. N denotes the number of LSCs, λ the number of iterations to realize, M the secret message, and P the width of the message (number of bits).* □

Our new scheme theoretically presented in [5] is here described by three main algorithms:

1) The first one, detailed in Algorithm 1 allows to generate the embedding strategy of the system which is a part of the embedding key in addition with the choice of the LSCs and the number of iterations to realize.
2) The second one, detailed in Algorithm 2 allows to embed the message into the LSCs of the cover media using the strategy. The strategy has been generated by the first algorithm and the same number of iterations is used.
3) The last one, detailed in Algorithm 3 allows to extract the secret message from the LSCs of the media (the stego-content) using the strategy wich is a part of the extraction key in addition with the width of the message.

In adjunction of these three functions, two other complementary functions have to be used:

1) The first one, detailed in Algorithm 4, allow to extract MSCs, LSCs, and passive coefficients from the host content. Its implementation is based on the concept of signification function described in Definition 2.
2) The last one, detailed in Algorithm 5, allow to rebuild the new host content (the stego-content) from the corresponding MSCs, LSCs, and passive coefficients. Its implementation is also based on the concept of signification function described in Definition 2. This function realize the invert operation of the previous one.

Remark 2: *The two previous algorithms have to be implemented by the user depending on each application context should be adjusted accordingly: either in spatial description, in frequency description, or in other description. They correspond to the theoretical concept described in Definition 2. Their implementation depends on the application context.* □

Example 3: *For example the algorithm 4 in spatial domain can correspond to the extraction of the 3 last bits of each pixel as LSCs, the 3 first bits as MSCs, and the 2 center bits as passive coefficients.* □

Algorithm 1: $strategy(N, P, \lambda)$

```
/* $S$ is a sequence of integers into
   $[\![0, P-1]\!]$, such that $(S_{n_0}, \ldots, S_{n_0+P-1})$
   is injective on $[\![0, P-1]\!]$.              */
Result: $S$: The strategy, integer sequence $(S_0, S_1, \ldots)$.
begin
    $n_0 \longleftarrow L - P + 1$;
    if $P > N$ OR $n_0 < 0$ then
        return ERROR
    $S \longleftarrow$ Array of width $\lambda$, all values initialized to 0;
    $cpt \longleftarrow 0$;
    while $cpt < n_0$ do
        $S_{cpt} \longleftarrow$Random integer in $[\![0, P-1]\!]$.;
        $cpt \longleftarrow cpt + 1$;
    $A \longleftarrow$ We generate an arrangement of $[\![0, P-1]\!]$;
    for $k \in [\![0, P-1]\!]$ do
        $S_{n_0+k} \longleftarrow A_k$;
    return $S$
end
```

Algorithm 2: $embed(LSC, M, S, \lambda)$

```
Result: New LSCs with embedded message.
begin
    $N \longleftarrow$ Number of LSCs in $LSC$;
    $P \longleftarrow$ Width of the message $M$;
    for $k \in [\![0, \lambda]\!]$ do
        $i \longleftarrow S_k$;
        $LSC_i \longleftarrow M_i$;
    return $LSC$
end
```

Algorithm 3: $extract(LSC, S, \lambda, P)$

```
Result: The message to extract from $LSC$.
begin
    $RS \longleftarrow$ The strategy $S$ written in reverse order.;
    $M \longleftarrow$ Array of width $P$, all values initialized to 0;
    for $k \in [\![0, \lambda]\!]$ do
        $i \longleftarrow RS_k$;
        $M_i \longleftarrow LSC_i$;
    return $M$
end
```

B. Discussion

We first notice that our $\mathcal{DI}_3$ scheme embeds the message in LSB as all the other approaches. Furthermore, among all

Algorithm 4: $significationFunction(H)$

```
Data: $H$: The original host content.
Result: $MSC$: MSCs of the host content $H$.
Result: $PC$: Passive coefficients of the host content $H$.
Result: $LSC$: LSCs of the host content $H$.
begin
    /* Implemented by the user.        */
    return $(MSC, PC, LSC)$
end
```

Algorithm 5: $buildFunction(MSC, PC, LSC)$)

```
Result: $H$: The new rebuilt host content.
begin
    /* Implemented by the user.        */
    return $(MSC, PC, LSC)$
end
```

the LSB, the choice of those which are modified according to the message is based on a secured PRNG whereas F5, and thus nsF5 only require a PRNG. Finally in this scheme, we have postponed the optimization of considering again a subset of them according to the distortion their modification may induce. According to us, further theoretical study are necessary to take this feature into consideration. In future work, it is planed to compare the robustness and efficiency of all the schemes in the context of semantic web. To initiate this study in this first article, the robustness of $\mathcal{DI}_3$ is detailled in the next section.

V. Robustness Study

This section evaluates the robustness of our approach [3].

Each experiment is build on a set of 50 images which are randomly selected among database taken from the BOSS contest [6]. Each cover is a 512×512 greyscale digital image. The relative payload is always set with 0.1 bit per pixel. Under that constrain, the embedded message m is a sequence of 26214 randomly generated bits.

Following the same model of robustness studies in previous similar work in the field of information hiding, we choose some classical attacks like cropping, compression, and rotation studied in this research work. Other attacks and geometric transformations will be explore in a complementary study. Testing the robustness of the approach is achieved by successively applying on stego content images attacks. Differences between the message that is extracted from the attacked image and the original one are computed and expressed as percentage.

To deal with cropping attack, different percentage of cropping (from 1% to 81%) are applied on the stego content image. Fig. 2 (c) presents effects of such an attack.

We address robustness against JPEG an JPEG 2000 compression. Results are respectively presented in Fig. 2 (a) and in Fig. 2 (b).

Attacked based on geometric transformations are addressed through rotation attacks: two opposite rotations of angle θ are successively applied around the center of the image. In these geometric transformations, angles range from 2 to 20 degrees. Results effects of such an attack are also presented in Fig. 2 (d).

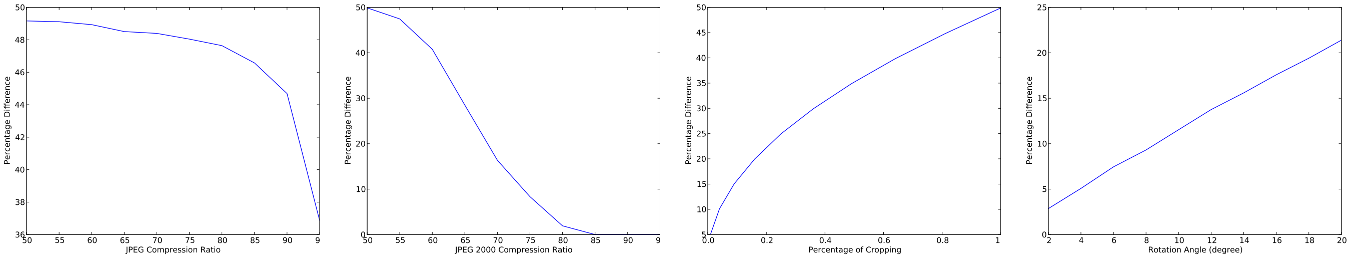
From all these experiments, one firstly can conclude that the steganographic scheme does not present obvious drawback and resists to all the attacks: all the percentage differences are so far less than 50%.

The comparison with robustness of other steganographic schemes exposed in the work will be realize in a complementary study, and the best utilization of each one in several context will be discuss.

VI. Conclusion and future work

In this research work, a new information hiding algorithm has been introduced to contribute to the semantic web. We have focused our work on the robustness aspect. The security has been studied in an other work [5]. Even if this new scheme $\mathcal{DI}_3$ does not possess topological properties (unlike the $\mathcal{CIS}_2$ [9]), its level of security seems to be sufficient for Internet applications. Particularly in the framework of the semantic web it is required to have robust steganographic processes. The security aspects is less important in this context. Indeed, it is important that the enrichment information persist after an attack. Especially for JPEG 2000 attacks, which are the two major attacks used in an internet framework. Additionally, this new scheme is faster than $\mathcal{CIS}_2$. This is a major advantage for an utilization through the Internet, to respect response times of web sites.

In a future work we intend to prove rigorously that $\mathcal{DI}_3$ is not topologically secure. The tests of robustness will be realized on a larger set of images of different types and sizes, using resources of the *Mésocentre de calcul de Franche-Comté [13] (an High-Performance Computing (HPC) center)* and using Jace environment [4], to take benefits of parallelism. So, the robustness and efficiency of our scheme $\mathcal{DI}_3$ will be compared to other schemes in order to show the best utilization in several contexts. Other kinds of attacks will be explored to evaluate more completely the robustness of the proposed scheme. For instance, robustness of the $\mathcal{DI}_3$ against Gaussian blur, rotation, contrast, and zeroing attacks will be regarded, and compared with a larger set of existing steganographic schemes as those described in this article. Unfortunately these academic algorithms are mainly designed to show their ability in embedding. Decoding aspect is rarely treated, and rarely implemented at all. Finally, a first web search engine compatible with the proposed robust watermarking scheme will be written, and



(a) JPEG effect. (b) JPEG 2000 effect. (c) Cropping attack. (d) Rotation attack.

Figure 2. Robustness of $\mathcal{DI}_3$ scheme facing several attacks (50 images from the BOSS repository)

automatic tagging of materials found on the Internet will be realized, to show the effectiveness of the approach.

References

[1] Delicious social bookmarking, http://delicious.com/.

[2] The frick collection, http://www.frick.org/.

[3] Jacques Bahi, Jean-François Couchot, and Christophe Guyeux. Steganography: a class of secure and robust algorithms. *The Computer Journal*, pages ***–***, 2011. Available online. Paper version to appear.

[4] Jacques Bahi, Mourad Hakem, and Kamel Mazouzi. Reliable parallel programming model for distributed computing environments. In *HeteroPar'09*, volume 6043 of *LNCS*, pages 162–171, Delft, Netherlands, 2009. Springer.

[5] Jacques M. Bahi, François Couchot, Nicolas Friot, and Christophe Guyeux. Application of steganography for anonymity through the internet. In *IHTIAP'2012, The First Workshop on Information Hiding Techniques for Internet Anonymity and Privacy*, pages ***–***, Venice, Italy, June 2012. To appear.

[6] P. Bas, T. Filler, and T. Pevný. Break our steganographic system — the ins and outs of organizing boss. In T. Filler, editor, *Information Hiding, 13th International Workshop*, Lecture Notes in Computer Science, Prague, Czech Republic, May 18–20, 2011. Springer-Verlag, New York.

[7] Jessica Fridrich. *Steganography in Digital Media: Principles, Algorithms, and Applications*. Cambridge University Press, New York, NY, USA, 1st edition, 2009.

[8] Jessica J. Fridrich, Tomás Pevný, and Jan Kodovský. Statistically undetectable jpeg steganography: dead ends challenges, and opportunities. In Deepa Kundur, Balakrishnan Prabhakaran, Jana Dittmann, and Jessica J. Fridrich, editors, *MM&Sec*, pages 3–14. ACM, 2007.

[9] Nicolas Friot, Christophe Guyeux, and Jacques M. Bahi. Chaotic iterations for steganography - stego-security and chaos-security. In Javier Lopez and Pierangela Samarati, editors, *SECRYPT*, pages 218–227. SciTePress, 2011.

[10] Christophe Guyeux, Nicolas Friot, and Jacques Bahi. Chaotic iterations versus spread-spectrum: chaos and stego security. In *IIH-MSP'10, 6-th Int. Conf. on Intelligent Information Hiding and Multimedia Signal Processing*, pages 208–211, Darmstadt, Germany, October 2010.

[11] T. Kalker. Considerations on watermarking security. In *Multimedia Signal Processing, 2001 IEEE Fourth Workshop on*, pages 201–206, 2001.

[12] Younhee Kim, Zoran Duric, and Dana Richards. Modified matrix encoding technique for minimal distortion steganography. In Jan Camenisch, Christian S. Collberg, Neil F. Johnson, and Phil Sallee, editors, *Information Hiding*, volume 4437 of *Lecture Notes in Computer Science*, pages 314–327. Springer, 2006.

[13] University of Franche-Comté. Le mésocentre de calcul de franche-comté, an high-performance computing (hpc) center, 2012. On line the 2012.02.23.

[14] Tomás Pevný, Patrick Bas, and Jessica J. Fridrich. Steganalysis by subtractive pixel adjacency matrix. *IEEE Transactions on Information Forensics and Security*, 5(2):215–224, 2010.

[15] Tomás Pevný, Tomás Filler, and Patrick Bas. Using high-dimensional image models to perform highly undetectable steganography. In Rainer Böhme, Philip W. L. Fong, and Reihaneh Safavi-Naini, editors, *Information Hiding*, volume 6387 of *Lecture Notes in Computer Science*, pages 161–177. Springer, 2010.

[16] Anindya Sarkar, Kaushal Solanki, and B. S. Manjunath. Further study on yass: Steganography based on randomized embedding to resist blind steganalysis. In *Security, forensics, steganography, and watermarking of multimedia contents X, San Jose CA , ETATS-UNIS*, pages 1–11, 2008.

[17] Kaushal Solanki, Anindya Sarkar, and B. S. Manjunath. Yass: Yet another steganographic scheme that resists blind steganalysis. In Teddy Furon, François Cayre, Gwenaël J. Doërr, and Patrick Bas, editors, *Information Hiding*, volume 4567 of *Lecture Notes in Computer Science*, pages 16–31. Springer, 2007.

[18] Andreas Westfeld. F5-a steganographic algorithm. In Ira S. Moskowitz, editor, *Information Hiding*, volume 2137 of *Lecture Notes in Computer Science*, pages 289–302. Springer, 2001.