# Two high capacity text steganography schemes based on color coding

**Juvet Karnel SADIE**[1,2,3], **Leonel MOYOU METCHEKA**[1,2,3], **René NDOUNDAM***[1,2,3]

[1]University of Yaounde I, LIA, Team GRIMCAPE, P.O. Box 812,Yaounde, Cameroon
[2]CETIC, Yaounde, Cameroon
[3]IRD, UMI 209, UMMISCO, IRD France Nord, F-93143, Bondy, Sorbonne Université, France

*E-mail : René NDOUNDAM ndoundam@yahoo.com

*DOI : 10.46298/arima.13273*
*Submitted on March 24, 2024 - Published on July 3, 2024*
Volume : **42 - CRI 2023 - 2024** - Year : **2024**
Special Issue : **42 - CRI 2023 - 2024**
Editors : *Paulin Melatagia, René Ndoundam, Kamel Barkaoui, Blaise Omer Yenke*

**Abstract**

Text steganography is a mechanism of hiding secret text message inside another text as a covering message. In this paper, we propose a text steganographic scheme based on color coding. This includes two different methods: the first based on permutation, and the second based on numeration systems. Given a secret message and a cover text, the proposed schemes embed the secret message in the cover text by making it colored. The stego-text is then send to the receiver by mail. After experiments, the results obtained show that our models perform a better hiding process in terms of hiding capacity as compared to the scheme of Aruna Malik et al. on which our idea is based.

**Keywords**

steganography; permutation; embedding capacity; numeration systems

## I INTRODUCTION

The word steganography is of Greek origin and means covered writing. It is the hiding of a message within another (cover medium) such as web pages, images or text, so that the presence of the hidden message is indiscernible. The key concept behind steganography is that the message to be transmitted should not be detectable with bare eyes. From the definition, steganography is used to ensure data confidentiality, like encryption. However, the main difference between the two methods is that with encryption, anybody can see that both parties are communicating in secret. Steganography hides the existence of a secret message and in the best case nobody can detect the presence of the message. When combined, steganography and encryption can provide more security. A number of steganographic methods have been introduced on different cover media such as images [18], video files [19] and audio files [12]. In text based steganographic methods, text is used as a cover media for hiding the secret data. Due to the lack of large scale redundancy of information in a text file, the human eye is very susceptible to any change between the original and the modified texts. Therefore, text steganography seems to be the most difficult kind of steganography [17].

Aruna Malik et al. [16] proposed a high capacity text steganography scheme based on LZW compression and color coding. Their scheme uses the forward mail platform to hide secret data. The algorithm first compresses secret data and then hide the compressed data into the email addresses and also, in the cover message of email. The secret data bits are embedded in the cover text by making it colored using a color table.

In this paper, we first present some limits of the scheme of Aruna Malik et al. [16] and then propose a text steganographic scheme based on color coding, permutation and numeration systems, which improves the embedding capacity of the scheme of Aruna Malik et al. [16]. Given a secret message and a cover text, the proposed scheme embed the secret message in the cover text by making it colored, using a permutation algorithm for the first method and numeration systems for the second one.

Section 2 presents some preliminaries and related works. Section 3 concerns the presentation of the first method of our scheme. Section 4 labels the second approach of our scheme, section 5 presents experimental results. Discussion and conclusion are stated in section 6 and 7 respectively.

## II PRELIMINARIES AND RELATED WORKS

In this section, the focus is to present some preliminaries that lead us to the comprehension of our scheme.

### 2.1 Text Steganography

Several works have been proposed in the field of text steganography [8, 13–16]. Ekodeck and Ndoundam [13] proposed different approaches of PDF file based steganography, essentially based on the Chinese Remainder Theorem. Here, after a cover PDF document has been released from unnecessary characters of ASCII code A0, a secret message is hidden in it using one of the proposed approaches, making it invisible to common PDF readers, and the file is then transmitted through a non-secure communication channel.

Rajeev et al. [15] proposed an email-based steganography method using a combination of compression. The method uses the email forwarding platform to hide secret data in email addresses and the combination of BWT, MTF and LZW compression algorithms to increase the embedding capacity. Rajeev et al. [14] proposed a high-capacity email-based text steganography method using Huffman compression. This method uses the message forwarding platform to hide secret data in email addresses. To increase the embedding capacity, the number of characters in the email address is used to reference the secret bits. Additionally, the method adds random characters just before the @ symbol to increase randomness.

Aruna Malik et al [16] proposed a steganographic scheme based on LZW compression and coloring. Their scheme uses the email transfer platform to hide the secret. The algorithm first compresses the secret, then hides this compressed secret in the email addresses and in the body of the email. This concealment proceeds by coloring the text using a color coding table. The Table 1 gives the embedding capacity (CE) of the steganographic schemes presented above. We present below some limits of the scheme of Aruna Malik et al. [16].

### 2.2 Critic and limits

LZW is a lossless compression technique that performs high compression ratio when the source contains repetition pattern. In the LZW based steganographic scheme proposed by Aruna Malik [16], they apply this lossless compression on the secret message to increase the embedding capacity. But in the example proposed, there is no compression. In other words, the size of the

Table 1: presentation of the embedding capacity (CE) of some steganographic techniques based on coloring and mail transfer

| Techniques | CE in percentage |
|---|---|
| Desoky [7] | 3.87 % |
| Rajeev et al [15] | 7.03 % |
| Rajeev et al [14] | 7.21 % |
| Aruna et al [16] | 13.43% |

compressed text is much greater than the size of the secret. To show this, we will give three different implementations of LZW algorithm applied to the secret message.

### *2.2.1 The LZW Algorithm with initial dictionary fixed and known*

This algorithm [11] starts by initializing the dictionary with the 256 characters of the ASCII code from 0 to 255. The output codes start at a minimum bit size equal to 9 and in general, as long as the indexes considered are strictly inferior to n = $2^k$ - 1, we can represent them on k bits. Applying this method to the following secret message: **underlying physiological mechanisms**, we obtain the outputs presented in Table 2. The binary compressed text is obtained by converting the indexes of the output column of the array to 9 bits :
001110101 001101110 001100100 001100101 001110010 001101100 001111001 001101001 001101110 001100111 000100000 001110000 001101000 001111001 001110011 001101001 001101111 001101100 001101111 001100111 001101001 001100011 001100001 001101100 000100000 001101101 001100101 001100011 001101000 001100001 001101110 001101001 001110011 001101101 001110011. Hence, the size of the output is 35*9 = 315 bits.

### *2.2.2 The LZW algorithm with sharing of the initial dictionary*

In this version [11], the initial dictionary contains only the character of the secret message. The output code is represented on height bits. The particularity of this implementation is from the initial dictionary which must be shared between the two parties in order to be able to decompress the binary code. Table 3 presents the initial dictionary for the same secret message. Here are the output code : 1 2 3 4 5 6 7 8 2 9 10 11 12 7 13 8 14 6 14 9 8 15 16 6 10 17 4 15 12 16 2 8 13 17 13 and in binary we have :
00000001 00000010 00000011 00000100 00000101 00000110 00000111 00001000 00000010 00001001 00001010 00001011 00001100 00000111 00001101 00001000 00001110 00000110 00001110 00001001 00001000 00001111 00010000 00000110 00001010 00010001 00000100 00001111 00001100 00010000 00000010 00001000 00001101 00010001 00001101. Hence, the size of the output is the sum of the size of initial dictionary and the output code: 17+35 = 52 bytes = 416 bits.

### *2.2.3 The Unix compress command*

The ncompress package [21] is a compression utility available on Linux which contains the compress command for fast compression and decompression using LZW algorithm. The algorithm behind this command is explained at page 153 of the data compression book [6]. The initial dictionary size is 512 and the minimum output code size is 9 bits. This package can be installed by using the command "sudo apt-get install ncompress". Based on the Ubuntu 16.04 platform, this command produces a file with .Z extension as compress file. By Applying this command "compress -v source.txt" to the secret message contained in text source file with the -v option, the given output indicates that there is no compression and the .Z file size is 44 bytes

Table 2: LZW Algorithm output with initial dictionary fixed and known

| Buffer | input-char | Output | New Item | Buffer | input-char | Output | New Item |
|---|---|---|---|---|---|---|---|
| u | n | 117 | 256=un | l | o | 108 | 273=lo |
| n | d | 110 | 257=nd | o | g | 111 | 274=og |
| d | e | 100 | 258=de | g | i | 103 | 275=gi |
| e | r | 101 | 259=er | i | c | 105 | 276=ic |
| r | l | 114 | 260=rl | c | a | 99 | 277=ca |
| l | y | 108 | 261=ly | a | l | 97 | 278=al |
| y | i | 121 | 262=yi | l |  | 108 | 279=l |
| i | n | 105 | 263=in |  | m | 32 | 280= m |
| n | g | 110 | 264=ng | m | e | 109 | 281=me |
| g |  | 103 | 265=g | e | c | 101 | 282=ec |
|  | p | 32 | 266= p | c | h | 99 | 283=ch |
| p | h | 112 | 267=ph | h | a | 104 | 284=ha |
| h | y | 104 | 268=hy | a | n | 97 | 285=an |
| y | s | 121 | 269=ys | n | i | 110 | 286=ni |
| s | i | 115 | 270=si | i | s | 105 | 287=is |
| i | o | 105 | 271=io | s | m | 115 | 288=sm |
| o | l | 111 | 272=ol | m | s | 109 | 289=ms |
|  |  |  |  | s |  | 115 |  |

= 352 bits.
Finally the Table 4 shows the comparison in bit between the original text size and the output size after the compression using the three different approaches of LZW implementation: the LZW algorithm with initial dictionary fixed and known, the LZW algorithm with sharing of the initial dictionary and the Unix compress command.

From Aruna et al. paper [16], the size obtained was 264 bits, but we have proven above that there is no compression for this example. This is the principal limit of this steganographic scheme, where for some messages the reduction of the message size will not be possible.

Furthermore, another limit to the scheme of Aruna et al. paper [16] is that: **for any integer $n$, there are $2^n$ different binary words of length $n$, but only $\sum_{i=0}^{n-1} 2^i = 2^n - 1$ shorter descriptions. For all $n$, there therefore exists at least one binary word of length n which cannot be compressed**[6]. So there will be cases where the secret cannot be compressed. Our paper uses:

- the idea of color coding contained in the paper of Aruna Malik et al. [16];
- the permutation generation method of W. Myrvold and F. Ruskey [4];
- the numeration systems;

to present a new scheme where the secret message embedding capacity is better than the scheme of Aruna Malik et al [16].

Table 3: Initial Dictionary

| Index | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| char | u | n | d | e | r | l | y | i | g | space | p | h | s | o | c | a | m |

Table 4: Secret message size comparison

| Secret message size | Output size 1 | Output size 2 | Output size 3 |
|---|---|---|---|
| 280 | 315 | 416 | 352 |

## 2.3 Permutation Generation Methods

Permutation is one of the most important combinatorial object in computing, and can be applied in various applications, for example, the scheduling problems. Permutation generation has a long history. Surveys in the field have been published in 1960 by D.H. Lehmer [1]. Several authors [2–4] have since developed many methods to generate all the possible permutations of n elements. Also, several works [10, 20] in steganography taking advantage of permutations have been done. W. Myrvold and F. Ruskey [4] proposed a ranking function for the permutations on $n$ symbols which assigns a unique integer in the range $[0, n! - 1]$ to each of the $n!$ permutations. Also, they proposed an unranking function for which, given an integer $r$ between 0 and $n!$ - 1, the value of the function is the permutation of rank $r$.

### *2.3.1 Unranking function*

First of all, recall that a permutation of order $n$ is an arrangement of $n$ symbols. An array $\pi[0 \cdots n-1]$ is initialized to the identity permutation $\pi[i] = i$, for $i = 0, 1, \cdots n-1$.

```
Procedure $unrank(n, r, \pi)$[4]
    begin
      if $n > 0$ then
        $swap(\pi[n-1], \pi[r \bmod n])$;
        $unrank(n-1, \lfloor r/n \rfloor, \pi)$;
      end;
   end;
```

**Note**: $swap(a, b)$ exchanges the values of variables $a$ and $b$.

### *2.3.2 Ranking function*

To rank, first compute $\pi^{-1}$. This can be done by iterating
$\pi^{-1}[\pi[i]] = i$, for $i = 0, 1, \cdots, n-1$.
In the algorithm below, both $\pi$ and $\pi^{-1}$ are modified.

```
function $rank(n, \pi, \pi^{-1})$:integer[4]
    begin
      if $n = 1$ then return(0) end;
        $s := \pi[n-1]$ ;
        $swap(\pi[n-1], \pi[\pi^{-1}[n-1]])$ ;
        $swap(\pi^{-1}[s], \pi^{-1}[n-1])$ ;
        return$(s + n.rank(n-1, \pi, \pi^{-1}))$ ;
   end;
```

## III SCHEME DESIGN BASED ON PERMUTATION

In this section, we present the first method of our scheme.

### 3.1 Embedding Algorithm

The proposed algorithm takes the cover text $C$, the secret message M, the e-mail address of the receiver and the initial permutation. First, it computes the binary representation of the secret and divides that representation into blocks of $t$ bits, $t = \lfloor log_2(n!) \rfloor$. It also divides the cover-text into blocks of $n$ characters. For each block of the secret stream, computes its decimal representation (the rank) and the permutation related to that rank, using the initial permutation. Colors the corresponding block of the cover text according the permutation obtained. It finally concatenates all the stego-blocks and send them to the receiver by e-mail.

```
Input:
        $C$: the cover text;
        $M$: the secret message to embed;
        The key $\pi$: the initial permutation of $n$ colors;
        $e$: the e-mail address of the receiver;
Output:
        $C'$: the stego-message;
Begin:
       1. Compute $m$, the binary representation of $M$;
       2. Compute $t = \lfloor log_2(n!) \rfloor$;
       3. Divide $m$ into $p$ blocks of $t$ bits each, $b_1, b_2, \cdots, b_p$;
       4. Divide $C$ into $k$ blocks of $n$ characters each $c_1, c_2, \cdots, c_k$;
       5. For each block $b_i, 1 \leq i \leq p$:
              a. compute $Nperm = (b_i)_{10}$, the decimal representation of $b_i$;
              b. compute $\pi' = unrank(n, Nperm, \pi)$, the permutation corresponding
                  to the number $Nperm$. $\pi'$ can be considered as $\pi'(1), \pi'(2), \cdots, \pi'(n)$;
              c. color each character of $c_i$ by the corresponding color given
                  by the permutation $\pi'$ and obtain the string $c'_i$;
              d. compute $C' \leftarrow C'||c'_i$; where a||b is the concatenation of a and b.
       6. If the next character is EOF (End of File) then
             begin
              a. Use e to send C' by mail to the receiver;
             end
            else
             begin
              a. Coloured the next character with a color different of permutation colors.
                  This color is shared by the sender and the receiver.
                  However, this color will not be very distant from the others;
              b. Randomly color the rest of characters of C by the colors of colors table,
                  until obtain the EOF character;
              c. Use e to send C' by mail to the receiver;
             end;
   End;
```

### 3.2 Retrieval Algorithm

Given the stego-text $C'$, the algorithm extract the secret message M, using the initial permutation. First the stego-text is divided into blocks of $n$ characters. For each block, the algorithm extract the rank related to the permutation according to the order of appearance of the colors.

Finally the algorithm concatenates the binary representation of the rank and obtains the secret.

```
Input:
        $C'$: the stego-text;
        The key $\pi$: the initial permutation of $n$ colors;
Output:
        $M$: the secret message;
Begin:
        1. Retrieve all characters coloured by the permutation colors, until a color different
           from the colors in the colors table, or the EOF character is obtained.
           Lets call them $C''$;
        2. Divide $C''$ into $p$ blocks of $n$ characters each $c_1, c_2, \cdots, c_p$;
        3. For each block $c_k, 1 \leq k \leq p$:
              a. use the color order of characters to compute the related permutation,
                 that we call $\pi'$. $\pi'$ can be considered as $\pi'(1), \pi'(2), \cdots, \pi'(n)$;
              b. compute the number $Nperm = rank(n, \pi', \pi'^{-1})$;
              c. compute $m' = (Nperm)_2$, the binary representation of $Nperm$;
              d. compute $M \leftarrow M||m'$;
    End;
```

## IV SCHEME DESIGN BASED ON NUMERATION SYSTEMS

In this new approach, we improve the method of the first scheme with the assertion that each color can be repeated as many times on some positions of a given group of characters. Unlike the previous scheme in which each color could only appear once in a group of precise characters.

### 4.1 The Scheme description

we give a brief description of how this new scheme works by following these steps:

1. Choose a base $B$ such that $2 \leq B \leq 2^{24}$, where $2^{24}$ is the number of existing colors ;
2. choose B colors from the set of $2^{24}$ colors number from 0 to $B-1$;
3. convert the secret m to base B such that : $m = (m_{q-1}...m_1m_0)_B$,
   where $0 \leq m_i \leq B-1$; We assume that the number of characters of the covert text is $n$ and $q \leq n$;
4. For $i = 0$ to $q-1$ do
   The character $c_i$ is coloured with the color relative to $m_i$
5. The text coloured is then send to the receiver.

The reverse procedure consists to extract the secret conceal in the colors distribution. These steps must be performed by the receiver of the stego-text :

1. Take the text with the first $q$ characters which has been coloured;
2. For $i = 0$ to $q-1$ do

   Find the color number $z_i$ associated to the character $c_i$ by using the reference color table shared between the sender and the receiver;
3. Convert $z = (z_{q-1}...z_1z_0)_B$ to binary and get the secret message.

### 4.2 Embedding Algorithm

Here, we present the embedding algorithm.

```
Input
        C: the cover text;
        M: the secret message to embed;
        B : The base;
        T : The table of B color;
        e : the e-mail address of the receiver;

Output
        C': the stego-message;
Begin
         1. Convert the secret M to base B such that : $m = (m_{n-1}...m_1m_0)_B$,
              where $0 \leq m_i \leq B - 1$;
         2. For $i = n - 1$ to 0 do
            a. Find in the color table, the color $a_i$ associated to the value $m_i$;
            b. Coloured the character $c_i$ of $C$ with the color $a_i$ and obtain $c'_i$ ;
            c. Compute $C' \longleftarrow C' \mid\mid c'_i$; where $a \mid\mid b$ is the concatenation of a and b;
         3. If the next character is not EOF (End of File) then
            a. Coloured the next character with a color different from the colors table T.
              This color is shared by the sender and the receiver. However, this color will
               not be very distant from the others;
            b. Randomly color the rest of characters of C by the colors from the colors table,
               until obtain the EOF character;
            c. Compute $C' \longleftarrow C' \mid\mid c'_j : n \leq j \leq l$,
               where $l$ is the position of the last character of $C$;
         3. Use e to send $C'$ by mail to the receiver;
End.
```

### 4.3 Retrieving Algorithm

In this subsection, we present the retrieval algorithm.

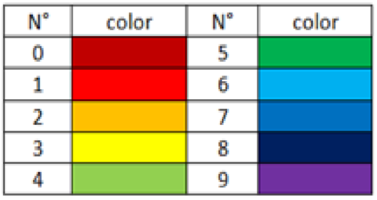
```
Input
        C': the stego-text;
        T : a table of B color;
        B : The base;
Output
        M: the secret message;
Begin
         1. Retrieve all characters coloured with the table colors, until obtain a color different
            from those of the colors table, or obtain the EOF character;
            Lets call them $C''$ and $\mid C'' \mid= n$; $(C'' = c_{n-1}c_{n-2}...c_1c_0)$;
         2. For $i = n - 1$ to 0 do
            a. Get the color $a_i$ associated to the color of the character $c_i$ of $C''$;
            b. Find in the color table, the value $m_i$ associated to the color $a_i$;
            c. compute $M \longleftarrow M \mid\mid m_i$;
         3. Compute $M_2$, the binary representation of the secret $M$;
End.
```

# V EXPERIMENTAL RESULTS

In this section, we present some experimentations related to our scheme.

## 5.1 First Method

In this subsection, we first propose a theoretical estimations of our embedding capacity for $n$ colors. Secondly, we present practical experimentation results in the case of 10, 16, 32 and 64 colors, based on example 1 of [16]. The table of 10 colors is given in figure 1.



| N° | color | N° | color |
|---|---|---|---|
| 0 | 🟥 | 5 | 🟩 |
| 1 | 🟥 | 6 | 🟦 |
| 2 | 🟧 | 7 | 🟦 |
| 3 | 🟨 | 8 | 🟦 |
| 4 | 🟩 | 9 | 🟪 |

Figure 1: The table of 10 colors

### *5.1.1 Theoretical estimations*

The Table 5 presents the embedding capacity of our scheme for some different values of n: 10, 16, 32,64. This theoretical estimation is based on our embedding algorithm.
More generally, in a set of n colors, the number of permutation of n distinct colors is n !. According to the stirling formula [5] we have:

$$n! \sim \left(\frac{n}{e}\right)^n \times \sqrt{2\pi n} \tag{1}$$

Where $\pi = 3.14$ is the area of the circle with unit radius, $e = 2.718$ is the base of the natural logarithm, and $\sim$ means approximate equality.

we know that:

$$n = 2^{log_2(n)} \tag{2}$$

By replacing the value of n in equation 1 we have:

$$\begin{aligned} n! \quad & \sim \left(\frac{2^{log_2(n)}}{2^{1.442695}}\right)^n \times \sqrt{2\pi n} & (3) \\ & \sim \left(2^{log_2(n)-1.442695}\right)^n \times \sqrt{2\pi n} & (4) \\ & \sim \left(2^{nlog_2(n)-1.442695n}\right) \times \sqrt{2\pi n} & (5) \\ & \sim \left(2^{nlog_2(n)-1.442695n}\right) \times 2^{log_2(\sqrt{2\pi n})} & (6) \\ & \sim \left(2^{nlog_2(n)-1.442695n}\right) \times 2^{\frac{1}{2}log_2(2\pi n)} & (7) \\ & \sim \left(2^{nlog_2(n)-1.442695n+\frac{1}{2}log_2(2\pi n)}\right) & (8) \end{aligned}$$

**Proposition :** the embedding capacity (E) using n colors to hide a secret is :

Table 5: Theoretical estimations of the proposed scheme

| n | M= $\lfloor log(n!) \rfloor$ | P=M/8 | 100*(P/n),(embedding capacity) |
|---|---|---|---|
| 10 | 21 | 2.6 | 26.25% |
| 16 | 44 | 5.5 | 34.37% |
| 32 | 117 | 14.6 | 45.63% |
| 64 | 295 | 36.9 | 57.66% |

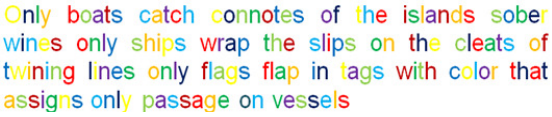
$$E = \frac{M \times 100}{n \times 8}$$

where $M = n(log_2(n) - 1.442695) + \frac{1}{2}log_2(2\pi n)$, and $n$ the number of colors.

**Remark:** As far as the space characters of the stego-text are not coloured, the embedding capacity can decrease in the experimentations.

#### *5.1.2 Experimentation 1*

Here, the secret message is : **underlying physiological mechanisms**
and the cover text is:
**Only boats catch connotes of the islands sober wines only ships wrap the slips on the cleats of twining lines only flags flap in tags with color that assigns only passage on vessels**
We apply our embedding algorithm and obtain the stego-text given by the figure 2. That stego-text is then send by mail to the receiver.



Only boats catch connotes of the islands sober
wines only ships wrap the slips on the cleats of
twining lines only flags flap in tags with color that
assigns only passage on vessels

Figure 2: The stego-text

With this example:

- in the case of 10 colors, the embedding capacity is 20.58 %;
- with 16 colors, the embedding capacity is 25.5 %;
- with 32 colors, the embedding capacity is 29.5 %;

#### *5.1.3 Experimentation 2*

In the example of figure 5 proposed by Aruna Malik [16], the secret message is : **behind using a cover text is to hide the presence of secret messages the presence of embedded messages in the resulting stego-text cannot be easily discovered by anyone except the intended recipient.**
and the cover-text is:
**in the research area of text steganography, algorithms based on font format have advantages of great capacity, good imperceptibility and wide application range. However, little work on steganalysis for such algorithms has been reported in the literature. based on the fact that the statistic features of font format will be changed after using font-format-based steganographic algorithms, we present a novel support vector machine-based steganalysis**

**algorithm to detect whether hidden information exists or not. this algorithm can not only effectively detect the existence of hidden information, but also estimate the hidden information length according to variations of font attribute value. as shown by experimental results, the detection accuracy of our algorithm reaches as high as 99.3 % when the hidden information length is at least 16 bits.**

We apply our embedding algorithm and obtain results.

- in the case of 10 colors, the embedding capacity is 22.32 %;
- with 16 colors, the embedding capacity is 29.64 %;
- with 32 colors, the embedding capacity is 38 %;

## 5.2 Method 2

### *5.2.1 Theoretical Estimation*

We want to color a block of text with $\eta$ characters. Each character is coloured with a single color. The number of colors used is $B$. Knowing that a color can appear as many times on some positions, the total number of colouring possibilities for each character is : $B$. For the $\eta$ characters, the total number of colouring possibilities is : $B^\eta$. The number of bits used to color the $\eta$ characters is : $log_2(B^\eta)$. The embedding capacity [9] is define as the ratio of the secret bits message by the stego cover bits :

$$Capacity \quad = \frac{log_2(B^\eta)}{\eta \times 8} = \frac{log_2(B)}{8} \tag{9}$$

The Table 6 gives a theoretical estimations of the capacity as a function of the base $B$ used.

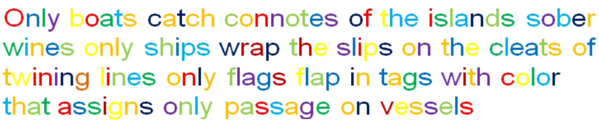
Table 6: Theoretical estimations of the proposed scheme

| $B$ | Capacity $\times 100$ |
|---|---|
| 10 | 41.5% |
| 16 | 50% |
| 32 | 62.5% |
| 64 | 75% |

**Remark:** As far as the space characters of the stego-text are not coloured, the embedding capacity can decrease in the experimentations.

### *5.2.2 Experimentation 1*

This experimentation is based on example 1 of [16], where the number of color $B$ is equal to 10. The figure 3 presents the results of the embedding process based on this second method for 10 colors.



Only boats catch connotes of the islands sober
wines only ships wrap the slips on the cleats of
twining lines only flags flap in tags with color
that assigns only passage on vessels

Figure 3: The stego-text for a table of 10 colors

With this example :

Table 7: Comparison between our scheme and the scheme of Aruna et al [16], in terms of embedding capacity, for 10 colors

| | **First Method** | **Second Method** | Aruna et al [16] |
|---|---:|---:|---:|
| example 1 [16] | **20.58 %** | **34.31 %** | 6.03 % |
| example of figure 5 [16] | **22.32 %** | **35.29 %** | 13.43% |

- in the case of 10 colors, the embedding capacity is 34.31 %;
- with 16 colors, the embedding capacity is 41.17 %;
- with 32 colors, the embedding capacity is 52.23 %;

### *5.2.3 Experimentation 2*

We apply our embedding algorithm and obtain the following results.

- in the case of 10 colors, the embedding capacity is 35.29 %;
- with 16 colors, the embedding capacity is 42.85 %;
- with 32 colors, the embedding capacity is 53.22 %;

## VI DISCUSSION

The aim of this paper is to propose a new scheme that improves the scheme proposed by Aruna et al [16]. The Table 7 recapitulates the embedding capacity of our schemes in comparison with the scheme of Aruna et al [16], in the case of 10 colors.

More Generally, the figure 4 gives the graphical representation of the performance of our scheme compared to other schemes.

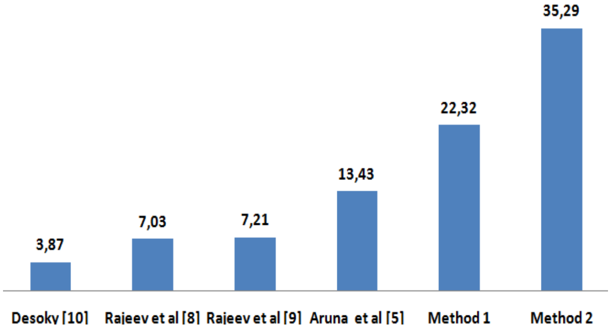


Figure 4: The graphical representation of the performance of our scheme compared to other schemes

In [16] the LZW algorithm is first apply to the secret message. The proposed scheme skips that step, thus our method decrease the computational complexity compared to the scheme of Aruna Malik et al.[16].

In order to increase the security of the model, the secret message can first be encrypted. On arrival, after extracting the secret data, the recipient will then have to decipher it to obtain the secret message.

It is true that coloring the text brings attention and then raises suspicion. We think that to solve this problem and provide a better undetectability, the cover text can be classified into the following fields: painting, poetry, art, humor, love. The color set is given by the initial permutation(the key). Before the communication, the two parties first share it by mail for instance.

## VII CONCLUSION

In this paper, two text steganographic schemes based on color coding have been proposed. The first based on permutation and the second based on numeration systems. Given a secret message and a cover text, the proposed schemes embed the secret message in the cover text by making it coloured. These two text steganographic schemes significantly improve the existing work of Aruna Malik et al.

## REFERENCES

### Publications

[1] D. Lehmer. "Teaching Combinatorial Tricks to a Computer". In: *Proceedings of Symposium in Applied Mathematics, Combinatorial Analysis, American Mathematical Society*. 1960.

[2] A. Nijenhuis and H. Wilf. *Combinatorial Algorithms: For Computers and Calculators*. Academic Press, 1978.

[3] C. Djamégni and M. Tchuente. "Cost-Optimal Pipeline Algorithm for Permutation Generation in Lexicographic Order". In: *Journal of Parallel Distribed Computing* 44.2 (1997), pages 153–159.

[4] W. Myrvold and F. Ruskey. "Ranking and Unranking Permutations in Linear Time". In: *Information Processing Letters* 79.6 (2001), pages 281–284.

[5] L. L. Leversha G, J. Pelikan, and K. Vesztergombi. *Discrete mathematics, elementary and beyond*. he Mathematical Gazette, 2004.

[6] B. Martin. *Codage, cryptologie et Applications*. Presses Polytechniques et Universitaires Romanes, 2004.

[7] A. Desoky and Listega. "list-based steganography methodology". In: *Journal of Information Security* 8 (2009), pages 247–261.

[8] B. Y. D. H. Kabetta and Suyoto. "Information hiding in CSS: a secure scheme text steganography using public key cryptosystem". In: *Int. Journal on Cryptography and Information Security* 1.13-22 (2011).

[9] J. Satir and H. Isik. "A compression-based text steganography method". In: *Journal of System and Software* (2012).

[10] H. Al-Bahadili. "A Secure Block Permutation Image Steganography Algorithm". In: *International Journal on Cryptography and Information Security* 3.3 (2013), pages 11–22.

[11] Z.-H. Wang, H.-R. Yang, T.-F. Cheng, and C.-C. Chang. "A high-performance reversible data-hiding scheme for LZW codes". In: *Journal of Systems and Software* (2013), pages 2771–2778.

[12] R. Tanwar and M. Bisla. "Audio steganography". In: *Proceedings of International Conference on Reliability Optimization and Information Technology (ICROIT)*. Faridabad, India, 2014.

[13] S. Ekodeck and R. Ndoundam. "PDF steganography based on Chinese Remainder Theorem". In: *Journal of Information Security and Applications* 1 (2016), pages 1–15.

[14] R. Kumar, S. Chand, and S. Singh. "A high capacity Email based text steganography scheme using Huffman compression". In: *Proceedings of the International Conference on Signal Processing and Integrated Networks*. 2016.

[15] R. Kumar, S. Chand, and S. Singh. “An Email based high capacity text steganography scheme using combinatorial compression”. In: *Proceedings of the 5th International Conference - Confluence The Next Generation Information Technology*. 2016, pages 336–339.

[16] A. Malik, G. Sikka, and H. K. Verma. “A high capacity text steganography scheme based on LZW compression and color coding”. In: *Engineering Science and Technology* 20.7127 (2017).

[17] M. MA, S. R, S. Z, and H. MK. “Review on Text Steganography Techniques”. In: *Mathematics* 9.21 (2021), page 2829.

[18] N. Subramanian, O. Elharrouss, S. Al-Maadeed, and A. Bouridane. “Image Steganography: A Review of the Recent Advances”. In: *IEEE Access* 9.1-4 (2021), pages 23409–23423.

[19] ayakanth Kunhoth, N. Subramanian, S. A.-m. adeed, and A. Bouridane. “Video steganography: recent advances and challenges”. In: *Multimedia Tools and Applications* 82.27 (2023), pages 1–43.

[20] J. K. Sadié, S. G. R. Ekodeck, and R. Ndoundam. “Binary Image Steganography Based on Permutation”. In: *Iran Journal of Computer Science* (2023).

[21] *Linux Compress Command Examples for Files and Directory*. https://linux.101hacks.com/unix/compress/.

## A ACKNOWLEDGEMENTS

This work was supported by *UMMISCO*, *LIA*, and the *University of Yaounde 1*. The authors are grateful for this support.