AUT Journal of Mathematics and Computing

AUT J. Math. Comput., 5(3) (2024) 233-244
https://doi.org/10.22060/AJMC.2023.21801.1111

Original Article

Text steganography by changing the black color

Bahman Khosravi*[a]

[a]*Department of Mathematics, Qom University of Technology, Qom, Iran*

ABSTRACT: Recently, a serious problem in communications is security. Hiding data is one of the most important security techniques. Steganography is the art and science of hiding information in a cover media. Texts are the most usual method of communication and so they are very suitable for cover objects. In this paper, we give a new technique for text steganography. There exist different models, such as RGB, HSL, HSV to determine a color. The main goal of the proposed algorithm is the fact that some different but very similar colors in RGB have the same code in HSL. We use this fact to hide data. For this purpose, first we find a color B' in RGB, which is very similar to black in such a way that they have the same code in HSL. Then, by changing the color of each character of the text to black or B', we conceal the information. We will show that the capacity of this method is better than some other methods of text steganography, and then we show that the invisibility of this algorithm is very high, which is the most prominent feature of the proposed technique.

Review History:
Received:23 September 2022
Revised:23 June 2023
Accepted:24 June 2023
Available Online:01 July 2024

Keywords:
Text steganography
Covert communication
Information security
Data hiding

MSC (2020):
94A99; 68P25; 68P27

1. Introduction

Steganography is the art and science of hiding communication of secret information, which is hidden in ordinary cover media such that the suspicion of the eavesdroppers is not arisen [28]. Among all of the known cover objects, the text is the most suitable, because of its great prevalence and its very common use in communications [38].

Text steganography is divided into three categories: linguistic, coverless, and structural text steganography. The third one is also divided into three categories. Feature-based text steganography is one of them, which hides the secret information by changing the feature of cover text such as RGB code (color), font, style, size, character spacing and others [36]. The four criteria for evaluating the efficiency of text steganography algorithm are security, invisibility, capacity, and robustness. Invisibility means that the hidden data has not caused noticeable or visible distortions that visual attacks can detect. Capacity refers to the amount of secret data hidden in the cover text. Robustness is the ability of the hidden data to withstand attacks, such as rotation, cropping, added noise, compression, and so on [36]. The purpose of this article is to present an algorithm for hiding information in a text. A steganography method is secure when it is resistant to attacks such as visual, structural, and statistical attacks. In this method, either the color of a character is not changed, or it is changed to a color which is very similar to black. So the difference between the cover and stego texts is very low, and they are almost identical, which seriously

*Corresponding author.
E-mail addresses: khosravi@qut.ac.ir

increases the invisibility. Therefore, visual attacks are not applicable to this method and suspicions of attackers are not arisen.

In Section 2, we point out a few related methods for text steganography. In Section 3, we give a brief description of the color system models. In Section 4, we present the hiding algorithm. In Section 5, the recovery method is presented. Experimental results and analysis of the security of the algorithm are presented in Section 6. Finally, in Section 7, we give the conclusion.

2. Related Works

Despite the difficulty of text steganography, many researchers have presented interesting methods, and especially they try to give some techniques in feature-based steganography for the improvement of the capacity and invisibility. Some of these methods use the embedding of special characters such as non-breaking space (A0) to conceal information [17]. In [8], this method was improved using the Chinese remainder theorem. In these papers, they embed A0 between words and characters to cover information. The capacity of this method is good, but some text editors show these kinds of characters. Also, by embedding special characters in the cover text, the size of the file is increased and by re-typing attack, the suspicion of the attacker has arisen. In another method for hiding data, the punctuation marks are used in wrong places to conceal the information ([30], [32]), but using them in wrong places raises the suspicions of attackers [7]. Also, the capacity for embedding of this algorithm is low. In [20] and [21], the secret message is embedded by a change tracking technique in Microsoft Word Documents, again the capacity of this scheme is low. In some methods, the secret message is embedded in a cover text, using the synonym of some special words in the cover text. Embedding the information in the animations of a powerpoint file is another method of steganography [40]. Another method for text steganography is SMS-Texing language which is a combination of abbreviated words used in SMS [31]. However, using the complete form of a word and its abbreviation simultaneously in a text can raise the suspicion of attackers. Another method is the use of white spaces of the cover- text, for example, spaces between words or enters at the end of lines. In [42] the authors worked on justified texts and for making more security they used the hash algorithm. In some methods, the space character is used to embed the secret message (for example see [27]). In [15], a stego-system was introduced which is based on justified text in a pdf file where the algorithm can be used on printed pages. Also, another method about justified text was presented in [14]. In [12], the secret message is embedded by changing the color of invisible characters.

Another algorithm is the use of forwarded mail platform and RGB code to hide data [16]. Later in [23], this algorithm was improved by using LZW code. Also, in [11], this algorithm was investigated and improved. The algorithm of changing color to conceal the information was investigated in [36] and [26], too.

3. Color Models and Color Difference

3.1. RGB and HSL Color Models

There are some color models, such as RGB, HSL, HSV and so on. RGB is an additive color model in which red, green, and blue, called primary colors, are added in many different ways to create a wide range of colors. Levels of R, G, and B can range from 0 to 255. This range is equivalent to the range of binary numbers from 00000000 to 11111111, or hexadecimal numbers from 00 to FF. Therefore, the total number of colors that can be created is $256 \times 256 \times 256$ or $16,777,216$. HSL is another model for representation of colors. The name HSL is derived from the first letter of the three words, Hue, Saturation, and Lightness. In HSL color model:

- H is the hue which is measured in degrees of the color circle. H can range from 0° to 360°. We note that 0° means red, 120° means green, and 240° means blue. Yellow, cyan, and magenta are also in 60°, 180°, and 300°, respectively.
- S is the saturation percent. S determines the color saturation percentage (%), starting from 0% which is a shade of gray to 100% full saturation.
- L is the lightness percent. L specifies the brightness of the color as a percentage (%), 0% means black and whatever goes to 100% becomes white. We note that 50% is normal.

By [10], the conversion formula from RGB to HSL is shown by the following equations:

$$R' = R/255, \quad G' = G/255, \quad B' = B/255,$$

$$C_{max} = MAX(R', G', B'),$$
$$C_{min} = MIN(R', G', B'),$$

$$\Delta = C_{max} - C_{min},$$

$$H = \begin{cases} 60^\circ \times \left(\frac{G'-B'}{\Delta} \bmod 6\right) & C_{max} = R' \\ 60^\circ \times \left(\frac{B'-R'}{\Delta} + 2\right) & C_{max} = G' \\ 60^\circ \times \left(\frac{R'-G'}{\Delta} + 4\right) & C_{max} = B' \end{cases}$$

$$S = \begin{cases} 0 & \Delta = 0 \\ \frac{\Delta}{1-|2L-1|} & \Delta \neq 0 \end{cases}$$

$$L = \big(C_{max} + C_{min}\big)/2$$

Using the above formulas, we conclude that $(1,1,1)$ in HSL is converted to $(0^\circ, 0, 1/255)$, and $(0,0,0)$ is converted to $(0^\circ, 0, 0)$. We note that $1/255 \equiv 0.4\%$. Since the third component of the code in the HSL model is in percentage, 0.4% is considered as 0% and so both $(0,0,0)$ and $(1,1,1)$ in RGB are converted to $(0^\circ, 0, 0)$ in HSL. We can check it by the sites for converting RGB to HSL, for example, see [25]. Also, we can check it in Microsoft Word. This is the main idea of the stego-system which is presented in this paper.

3.2. Color Difference and Delta-E

The separation between two colors, which is called "color difference" or "color distance", is a measure that quantitatively examines their difference. Euclidean distance, which is the distance between two colors in a color space, is a standard means of determining the distance between two colors, defined as follows:

$$\text{Distance} = \sqrt{(R_2 - R_1)^2 + (G_2 - G_1)^2 + (B_2 - B_1)^2}.$$

The international Commission on illumination (CIE) introduced a distance metric ΔE^* (or "Delta E"), where the term Δ is used to denote difference, commonly and "E" is the first letter of "Empfindung", a German word which means "sensation". After that CIE refined the metric, and it led to superior formulas called CIE 94 and CIE 2000. The range of Delta E values is between 0 to 100. If $\Delta E^* \leq 1$, the difference cannot be perceptible by human eyes. If $1 \leq \Delta E^* \leq 2$, the difference can be perceptible by close observation, and experienced can notice. If $2 \leq \Delta E^* \leq 3.5$, the difference can be perceptible even by inexperience. If $3.5 \leq \Delta E^* \leq 5$, the difference between two colors is clear. If $5 \leq \Delta E^*$, they are different.

4. Embedding and Hiding Algorithms

4.1. String of the Frequencies of the Characters

In our proposed method, first, we use Huffman coding to compress the message, and then the compressed data are hidden in a cover text. Since for Huffman decoding, the frequencies of characters are needed, we send them to the receiver in the cover text. For this purpose, the sender and the receiver agreed on an ordered table of characters which is called the agreed table. For sending the frequencies of the characters in the agreed table we proceed as follows.

We consider the following table for $-$ and the numbers:

Table 1:

Digits	0	1	2	3	4	5	6	7	8	9	-
Corresponding Code	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1111

Now we consider the string $n_1 - n_2 - n_3 - \cdots$ such that n_i is the frequency of the i-th character in the agreed table. Using the above table, change each digit of n_1 by the corresponding code, and we put them consecutively. Then we put 1111 to show that the digits of n_1 are finished. We continue this process for other n_i's, respectively to achieve string F. Note that if the frequency of n_i is 0, then we put only 1111 and do not put anything else. As an example, we assume that $257, 81, 0, \ldots$ are the frequencies of characters of the agreed table. It means that the frequency string is $257 - 81 - 0 - \cdots$. Using this method, we make F as follows:

$$F = \underbrace{0010}\,\underbrace{0101}\,\underbrace{0111}\,\underbrace{1111}\,\underbrace{1000}\,\underbrace{0001}\,\underbrace{1111}\,\underbrace{1111}\ldots$$

and continuously we add the frequencies of characters in the agreed table until F is obtained.

4.2. Embedding Algorithm

As we mentioned above the codes $(0, 0, 0)$ and $(1, 1, 1)$ in the RGB color model have the same code in HSL. Let S be a binary string and consider a cover text A. In our algorithm, we consider the characters of A, and we change their colors to embed the secret message S. We note that the default color of the text is $(0, 0, 0)$ in RGB. For hiding S in A, we proceed as follows. We change the color of the i-th character of the cover text from $(0, 0, 0)$ to $(0 + t_i, 0 + t_i, 0 + t_i)$, where t_i is the i-th bit in the string S. So the color of the i-th character of the cover text is $(0, 0, 0)$ if $t_i = 0$, and its color is $(1, 1, 1)$ if $t_i = 1$. Hence in this method, every bit is hidden in a character of the cover text.

4.3. Hiding Algorithm

In this section, we present the hiding algorithm which is based on the former steps.

(i) First, we use Huffman coding to compress the secret message, and get a binary string T.

(ii) To increase the security, we partition string T to blocks of length 8. Then we rewrite each of them in the reverse order, for example, sequence 10001110 is changed to 01110001. If the length of T is not divisible by 8, we do not change the last block of T. We denote the obtained string by S.

(iii) We embed F, the string of frequencies of the characters using the methods stated in Sections 4.1 and 4.2.

(iv) Finally, we embed string S in the sequel of the cover text.

5. Recovery Method

In this section, we present the recovery method. We note that the number of characters of the agreed table is known. Also the digits of the frequencies of the characters in the agreed table can be obtained according to the method stated in Section 4. We start from the beginning of the stego text and every 4 bits determine a digit until we receive 1111. Now the frequency of the first character of the agreed table is obtained. If the next 4 bits is 1111, then the frequency of the second character is 0, and otherwise, the frequency of the next character of the agreed table is obtained. We continue this process to find the frequency of the last character in the agreed table. Therefore the receiver can determine the Huffman codes of the characters in the agreed table. In the sequel of the text, the secret message is hidden. Since the frequency of each character and the length of the Huffman code of each character in the agreed table are known, the length of the string S is obtained. Therefore using the rest of the stego text, S is determined. Now we divide string S into blocks with length 8 and then we rewrite each block in the reverse order. Hence string T is obtained. Using Huffman decoding, we reach the secret message.

6. Experimental Results and Analysis of the Algorithm

To measure the performance of the text steganography algorithm, various parameters such as security, embedding capacity, invisibility (indistinguishably to the naked eye), and robustness against manipulation are examined [22]. To illustrate the algorithm, first an example is presented. Then hiding capacity, invisibility, security and analysis of the algorithm will be discussed. In this section, we implement the algorithm. For this purpose, we consider the secret message of the main example in [36]. Then we give the comparison of our method by some existing methods for text steganography. After that we give a complete analysis of the algorithm.

6.1. Experimental Results

Let the secret message be as follows:

> *behind using a cover text is to hide the presence of secret message s the presence of embedded messages in the resulting stego text cannot be easily discovered by anyone except the intended recipient*

If we substitute each character of the above message with 8 bits, the length of the corresponding binary string is 1592, but the length of its Huffman code is 783. Let the cover message be the following text which has 1081 characters.

> Steganography is the art and science of hiding communication of the secret information which is hidden in an ordinary cover media such that the suspicion of eavesdropper is not arisen. The word “Steganography” is derived from Greek words “stegos” meaning “cover” and “graphia” meaning “writing”. The use of steganography has a long history. The first documentation of using steganography was recorded by Herodotus, a Greek historian in the year 440 BC. In his book he quotes a story of a Greek ruler. When

this king was imprisoned by the king of Iran in Susa, he had to send a secret message to his son. To do this, he shaved off the hair of his servant and tattooed a message on his head. When the slave's hair was grown up enough, he sent the slave to his destination. Among all of the known cover objects, text is the most suitable cover object for steganography, because of its great prevalence and its very common use in communications. Despite the difficulty of text steganography, many researchers presented many interesting methods. In this section, we state some of them.

As the first step, we suppose that the agreed table is "abcdef. . . z, " where the last character is space.
The compressed version of the message by Huffman code is

$T =$ 1000110010000011010010101111010000101001101001010011110111111111010110111011111
0101101111100000111001100111011010101111100110111111000001100101001111100100000
1111011101011000101000100111010001111101101000111110100011010101100011001110110
1001010101001111010010011110101111100100000011110111010110001010001001110100011
1101101000111100011101100011000101010100010111101110100101010100111101001001010
1101101001111110010000001111011000101001000010111101100011010010100111110101100
0010011101111111000001110011001111101001111100110011101111001111000110011100011
1101001101011101000101110101011010101101011011101111100101100001011111000111000
1011101111100110001011011100100111000111001101000101110110011111001000000111011
0
1001110000100101010001011111011000110100110101110011000100110011100

Therefore

$S =$ 0011000111000001010100100010111110010100101001011011110011111111110110100111110
1
1011010000001110110011101101110111010111011001000011111010011001111100000000010
1
1110111100011010010001010010111010111110110001010001011110101011001100010111011
1
1010100111110010100100101101011100100111111100001010111001010001111001001110001
0
0101101100111100011011101000110000101010011110101001011100101010001011111010100
1
0101101101111110000000100110111110010100110100000001101101001011111110010001101
0
1110010011111111011100000110011001001011111001111110111001111001110011000111000
1
1100101101111010110100011101010101101010111011010011111000001101011111010001110
0
1101110101100111101101001001001111100011001011001011101011111001000010001101110
0
0011100110100100101000100110111100101100111010111000110000011100

The frequency of characters of the secret message is presented in the following table:

Table 2: Frequency Analysis

Character	space	a	b	c	d	e	f	g	h	i
Frequency	33	6	4	8	9	37	2	5	6	11
Code	111	01111	100011	11010	0101	00	010001	01001	10000	0110

Character	j	k	l	m	n	o	p	q	r	s
Frequency	0	0	2	3	13	8	4	0	7	15
Code			1011110	011101	1001	11011	101110		10110	1010

Character	t	u	v	w	x	y	z			
Frequency	16	2	2	0	3	3	0			
Code	1100	010000	1011111		011100	100010				

Hence the string of the frequencies is as follows:

$$F = \underbrace{0011}\underbrace{0011}\underbrace{1111}\underbrace{0110}\underbrace{1111}\underbrace{0100}\underbrace{1111}\underbrace{1000}\underbrace{1111}\underbrace{1001}\underbrace{1111}\underbrace{0011}\underbrace{0111}\underbrace{1111}\underbrace{0010}\underbrace{1111}\underbrace{0101}\underbrace{1111}\underbrace{0110}$$
$$\underbrace{1111}\underbrace{0001}\underbrace{0001}\underbrace{1111}\underbrace{1111}\underbrace{1111}\underbrace{0010}\underbrace{1111}\underbrace{0011}\underbrace{1111}\underbrace{0001}\underbrace{0011}\underbrace{1111}\underbrace{1000}\underbrace{1111}\underbrace{0100}\underbrace{1111}\underbrace{1111}\underbrace{0111}$$
$$\underbrace{1111}\underbrace{0001}\underbrace{0101}\underbrace{1111}\underbrace{0001}\underbrace{0110}\underbrace{1111}\underbrace{0010}\underbrace{1111}\underbrace{0010}\underbrace{1111}\underbrace{1111}\underbrace{0011}\underbrace{1111}\underbrace{0011}\underbrace{1111}\underbrace{1111}$$

Therefore, the length of the frequency string is 220. By noting the frequencies of each character and its corresponding Huffman code, we get the length of the string S which is 783. Now by the above algorithm, we hide the secret message. The result is presented in Figure 1.

Frequencies

Secret Message

Steganography is the art and science of hiding communication of the secret information which is hidden in an ordinary cover media such that the suspicion of eavesdropper is not aroused. The word "Steganography" is derived from Greek words "stegos" meaning "cover" and "graphia" meaning "writing". The use of steganography has a long history. The first documentation of using steganography was recorded by Herodotus, a Greek historian in the year 440 BC. In his book he quotes a story of a Greek ruler. When this king was imprisoned by the king of Iran in Susa, he had to send a secret message to his son. To do this, he shaved off the hair of his servant and tattooed a message on his head. When the slave's hair was grown up enough, he sent the slave to his destination. Among all of the known cover objects, text is the most suitable cover object for steganography, because of its great prevalence and its very common use in communications. Despite the difficulty of text steganography, many researchers presented many interesting methods. In this section, we state some of them.

Figure 1: Stego text

6.2. Capacity Ratio of the Algorithm

The embedding capacity of a method is defined as the number of letters/bits of secret message which can be hide in a cover text. We note that BPL means bit(s)-per-embeddable -locations. Therefore, if the embeddable locations of the cover message are the letters and $l_1, l_2, \ldots, l_n$ are the letters of the cover text, then the number of embeddable bits inside the cover message is equal to

$$E = \sum_{i=1}^{n} BPL_i.$$

Also, if the required embeddable bits for hiding a secret message inside a cover text is equal to RE, then the capacity ratio is defined as $CR = E/RE$ (see [35]). Therefore, the capacity ratio of our proposed method is equal to 100%, since we use each letter of the cover message. Of course, some authors for increasing the capacity ratio use Huffman coding or LZW. For example in [36], in pre-embedding stage, they compressed the message using Huffman coding, then the embedding stage was done. Similarly in [16], Huffman coding was used and considered in the capacity ratio. Later in [23], using LZW, they increase the capacity ratio of [16]. Since we also use Huffman coding in pre-embedding stage, and then we embed the message in the cover text, the capacity ratio is increased, but we do not consider it in comparison.

In [35], some comparisons between several algorithms of text steganography were presented. In Table 3, using them, we give the comparison with some methods.

Table 3: A comparison of capacity of some techniques

Method	Proposed method	CovertSYS [35]	AITSteg [33]	Homoglyphs [29]	Line-Chat [37]	FontCode [39]
Capacity (%)	100	100	100	18.5	10	100

6.3. Invisibility

As we mentioned above, Delta-E is a very strong measure to determine the difference between two codes in RGB. When Delta-E of two codes is less than 1, then they seem identical in human vision. In our algorithm, we consider

two colors in RGB such that they have the same code in HSL. Since Delta-E of them is 0.2742%, the change of colors in consecutive letters is invisible. We note that the main advantage of this method is its invisibility and, as we can see in Table 6, Delta-E of some existing algorithms is very high with respect to our method.

6.4. Robustness

Robustness in steganography is the ability of the hidden data to withstand attacks, such as rotation, cropping, added noise, and compression [41]. The ratio of the total number of embedding locations to the size of the cover text is called the losing probability (LP). Using the Losing probability (LP), the robustness of a text steganography technique is determined. The Distortion Robustness (DR) is determined by the following formula: $DR = 1 - LP$ (see [36], [34]).

6.5. Security

The widespread use of texts in communications makes them one of the best sources for hiding information. Since inexpert changes in a text arise the suspicion of attackers, text steganography is one of the most difficult types of steganography.

There are different steganalysis attacks to establish the security of a steganography algorithm. An adversary may perform many attacks on the cover text to know if the cover media contains some hidden data or not. There are two types of attacks, namely passive attack, and active attack. In a passive attack, the adversary observes the cover media and runs many tests. In an active attack, the adversary by making changes in cover text for example changing the font size checks whether the cover text is infected or not.

Generally, three types of attacks are used in a passive attack process, namely, visual, structural, and statistical attacks [22]. The proposed algorithm changes colors in RGB such that the main color and the changed color have the same code in HSL, so we get that the stego text and the cover text are similar enough. Also, the proposed algorithm does not add/change/modify any noise to the cover text. By noting that the algorithm does not add anything to the cover text, and does not change any visual properties else on the cover text, we conclude that this algorithm is resistant to visual attack. We note that this algorithm does not use abbreviated expressions instead of main words. Also, according to the rules of linguistics, this algorithm does not change any words or some parts of the cover text or replace them with anything else. Therefore, the structure of the cover text does not change and therefore it is resistant to structural attacks. Also by noting that this algorithm never changes the cover text and especially its statistical properties such as word frequency, we conclude that it is resistant to statistical attacks. Therefore, the proposed algorithm is resistant to passive attacks.

In active attacks, to determine whether the cover text is infected or not, the adversary performs some tests such as changing the color of the text, font size, white space manipulation and re-typing the text. Most of the methods for text steganography are not strong against active attacks, and by using active attacks, the secret message is destroyed in the cover text. Also by using active attacks on some of these algorithms, the existence of a secret message is revealed too, but some of these algorithms have the ability not to reveal the existence of secret message (see [17] and [8]). For example, by the re-typing attack, if the size of the main cover file and the new file is very different, it arises the suspicion of attackers. On the other hand, in some methods for text steganography, by using re-typing attack, although the secret message is destroyed, but the existence of the secret message is not detectable (see [20]). In the proposed method, similar to [20], by re-typing attack, the secret message is not detectable, but the suspicion of the attackers does not arise.

As mentioned former, security means that when hidden information is sent, an eavesdropper will not notice or suspect the existence of hidden information. To obtain more secure data, it is possible to use cryptography systems in the steganography system [13]. Optical and digital criteria are two measures that can be used in the field of steganography to check security criteria ([33], [3]).

6.5.1. Optical Standard

For Measuring the optical standard, there exist two methods. In the first method, the text is converted into a photo, and then the following formulas are used. In the first method, the Peak signal-to-noise ratio (on PSNR) is calculated which is the ratio m/n, where m is the maximum possible power of a signal, and n is the power of corrupting noise that shows the quality of its representation. we know many signals contain a wide dynamic range. So we usually use the logarithmic decible scale to define PSINR, and it is calculated by using the mean squared error (MSE). For every noise-free $m \times n$ monochrome image I, we consider its noisy approximation k. Then, by using stego system, MSE is calculated as follows:

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i,j) - K(i,j)]^2.$$

The PSNR (in dB) is defined as: $PSNR = 20log_{10}(MAX_I) - 10log_{10}(MSE)$, where MAX_I is the maximum possible pixel value of the image [3].

Table 4: Homogeneity ratio of some techniques

Methods	Proposed method	[3] (a)	[3] (b)	[7]	[5]	[6] (a)	[6] (b)	[9]	[2]
Excess visual characters	0	9	0	4	0	0	4	17	8
Homogeneity ratio	100%	91%	100%	96%	100%	100%	96%	83%	92%

6.5.2. Digital standard

"The other criterion is statistically measured by changing the size of the file according to the number of entries in each embedding algorithm. To apply this metric using the same example mentioned before, the original file size is 13.931 bytes include 30 characters. The statistics of the eight different methods are shown in Table 5. If the number of characters is increased more than the original text, this makes the security ratio less in that method. This is because the file size of the stego text becomes very different from the file size of the original text"[3].

$$\text{Discount value} = \frac{\text{Original Character} \times \text{Excess Character}}{100};$$

$$\text{Amount after discount} = \text{Original character} - \text{Discount value};$$

$$\text{Security ratio} = \frac{\text{Amount after discount}}{\text{Original character}} \times 100.$$

Using Table 5 of [3], we compare the proposed method with some techniques. For this purpose, we make a file with the size near to 13.9 bytes and with 30 characters. Then we hide a binary string by the method and results are presented in Table 5.

Table 5: The statistics of some techniques

Methods	Proposed method	[3] (a)	[3] (b)	[7]	[5]	[6] (a)	[6] (b)	[9]	[2]
File size (k bytes)	14	14.23	14.27	14.04	14.11	14.14	14.23	14.12	14.09
Total Characters	30%	54%	53%	34%	36%	42%	35%	47%	38%
Excess Characters	0	24	23	4	6	12	5	17	8

We use the results that presented in [3] and [4] to compare our method with them in Table 6. First, we note that

$$\text{Discount value} = \frac{\text{Original Character} \times \text{Excess Character}}{100} = \frac{\text{Original Character} \times 0}{100} = 0;$$

$$\text{Amount after discount} = \text{Original character} - \text{Discount value} = \text{Original character};$$

$$\text{Security ratio} = \frac{\text{Amount after discount}}{\text{Original character}} \times 100 = \frac{\text{Original character}}{\text{Original character}} \times 100 = 100\%.$$

Table 6: A comparison of security ratio of some techniques

Methods	Proposed method	[3] (a)	[3] (b)	[7]	[5]	[6] (a)	[6] (b)	[9]	[2]	[4]	[24]
Security ratio	100%	76%	77%	96%	94%	88%	95%	83%	92%	90.11%	100%

6.5.3. Side channel attack

Side channel steganalysis used to detect a steganographer in social websites using his /her behaviour analysis. The main tools in side channel attacks are computing methodologies, artificial intelligence and computer vision. According to the experimental results of side channel steganalysis, it is intuitively secure for the Steganographer to act identically to normal social users since he/she can avoid being detected side channel steganalysis [19]. For example, in [18] they consider the correlation between images sequence to detect the steganographer.

In our proposed method, since the cover text could be any text and each text can be considered as a cover and the difference between the cover text and the stego text is very low, the behavior of the steganographer is very similar to a normal user and so it has good resistance against side channel attacks.

6.6. Final Comparison

In Table 8 of [36], a comparison between some methods was presented. We use it to compare our method with them. In the Table 7, we compare our method with some methods about RGB. We note that our algorithm is compatible with any language which is called multilingual and it can be used in every language to hide the secret message, but some of algorithms are not multilingual which are determined in the table.

Table 7: A comparison with some techniques

Algorithms presented in	Capacity ratio	Average Delta-E in black color	Robustness	Multilingual
Our Method	100%	0.27%	4.3	✓
[36]	98.85%	4.7%	94.22	×
[16]	18.34%	113.2	14.64	✓
[23]	13.43%	103.3	14.64	✓
[11]	20.58%	100.6	10.28	✓
[26]	77.4%	3.3%	80.41	×
[1]	25.5%	2.4%	84.42	×

7. Conclusion

Text steganography is a powerful tool to send a secret message via a public channel safely. We conceal the secret message in a cover text such that one does not notice that the text contains a hidden message. There are many methods for text steganography and in this paper, we present a new method that has very high invisibility and a good capacity ratio. In fact, in our proposed method we compress the message by Huffman coding, and then by changing the color of each character we hide a bit of the secret message, which is resistant to many steganalysis attacks such as visual attacks, structural attacks and statistical attacks. The capacity ratio of this method is about 100% and the Delta-E of the cover text and the stego text is very low. This causes a significant improvement in the invisibility of this method.

Acknowledgment

The author would like to express his sincere gratitude to the referee for his/her useful and constructive comments and suggestions which improved the manuscript.

References

[1] A. F. Al-Azzawi1, *A multi-layer hybrid text steganography for secret communication using word tagging and rgb color coding*, Int. J. Netw. Secur. Appl., 10 (2018), pp. 1–12.

[2] S. Al-Nofaie, M. Fattani, and A. Gutub, *Merging two steganography techniques adjusted to improve Arabic text data security*, J. Comput. Sci. Comput. Math. (JCSCM), 6 (2016), pp. 59–65.

[3] S. Al-Nofaie, A. Gutub, and M. Al-Ghamdi, *Enhancing Arabic text steganography for personal usage utilizing pseudo-spaces*, J. King Saud Univ. Comput. Inf. Sci., 33 (2021), pp. 963–974.

[4] N. Alanazi, E. Khan, and A. Gutub, *Inclusion of unicode standard seamless characters to expand arabic text steganography for secure individual uses*, J. King Saud Univ. Comput. Inf. Sci., 34 (2022), pp. 1343–1356.

[5] R. A. Alotaibi and L. A. Elrefaei, *Utilizing word space with pointed and un-pointed letters for Arabic text watermarking*, in 2016 UKSim-AMSS 18th International Conference on Computer Modelling and Simulation (UKSim), 2016, pp. 111–116.

[6] R. A. Alotaibi and L. A. Elrefaei, *Improved capacity arabic text watermarking methods based on open word space*, J. King Saud Univ. Comput. Inf. Sci., 30 (2018), pp. 236–248.

[7] W. Bender, D. Gruhl, N. Morimoto, and A. Lu, *Techniques for data hiding*, IBM Systems Journal, 35 (1996), pp. 313–336.

[8] S. G. R. Ekodeck and R. Ndoundam, *PDF steganography based on Chinese Remainder Theorem*, J. Inf. Secur. Appl., 29 (2016), pp. 1–15.

[9] A. Gutub and A. Al-Nazer, *High capacity steganography tool for Arabic text using 'kashida'*, ISC Int. J. Inf. Secur, 2 (2010), pp. 107–118.

[10] N. Ibraheem, M. Hasan, R. Z. Khan, and P. Mishra, *Understanding color models: A review*, 2 (2012), pp. 265–275.

[11] J. Karnel Sadie, L. Moyou Metcheka, and R. Ndoundam, *Two high capacity text steganography schemes based on color coding*, Revue Africaine de Recherche en Informatique et Mathématiques Appliquées, (2024).

[12] M. Khairullah, *A novel text steganography system using font color of the invisible characters in Microsoft Word Documents*, in 2009 Second International Conference on Computer and Electrical Engineering, vol. 1, 2009, pp. 482–484.

[13] A. Khan, *Robust textual steganography*, Journal of Science, 4 (2015), pp. 426–434.

[14] B. Khosravi and B. Khosravi, *Text steganography based on text justifying methods*, Adv. Defence Sci. & Technol., 12 (2021), pp. 159–167.

[15] B. Khosravi, B. Khosravi, B. Khosravi, and K. Nazarkardeh, *A new method for pdf steganography in justified texts*, Journal of Information Security and Applications, 45 (2019), pp. 61–70.

[16] R. Kumar, A. Malik, S. Singh, and S. Chand, *A high capacity email based text steganography scheme using huffman compression*, in 2016 3rd International Conference on Signal Processing and Integrated Networks (SPIN), Noida, India, 2016, pp. 53–56.

[17] I.-S. Lee and W.-H. Tsai, *A new approach to covert communication via PDF files*, Signal Processing, 90 (2010), pp. 557–565.

[18] L. Li, W. Zhang, K. Chen, and N. Yu, *Steganographic security analysis from side channel steganalysis and its complementary attacks*, IEEE Transactions on Multimedia, 22 (2020), pp. 2526–2536.

[19] L. Li, W. Zhang, K. Chen, H. Zha, and N. Yu, *Side channel steganalysis: When behavior is considered in steganographer detection*, Multimed. Tools Appl., 78 (2019), pp. 8041—-8055.

[20] T.-Y. Liu and W.-H. Tsai, *A new steganographic method for data hiding in microsoft word documents by a change tracking technique*, IEEE Transactions on Information Forensics and Security, 2 (2007), pp. 24–30.

[21] S. Mahato, D. A. Khan, and D. K. Yadav, *A modified approach to data hiding in Microsoft Word documents by change-tracking technique*, J. King Saud Univ. Comput. Inf. Sci., 32 (2020), pp. 216–224.

[22] G. Maji and S. Mandal, *A forward email based high capacity text steganography technique using a randomized and indexed word dictionary*, Multimed. Tools Appl., 79 (2020), pp. 26549–26569.

[23] A. Malik, G. Sikka, and H. K. Verma, *A high capacity text steganography scheme based on lzw compression and color coding*, Eng. Sci. Technol. Int J., 20 (2017), pp. 72–79.

[24] A. A. Obeidat, *Arabic text steganography using unicode of non-joined to right side letters*, Journal of Computer Science, 13 (2017), pp. 184–191.

[25] Online Applet, *HSL to RGB/RGB to HSL/Hex Colour Converter*. `https://www.peko-step.com/en/tool/hslrgb_en.html`. Accessed: 2023.

[26] B. Osman, *Message hiding technique in text steganography using RGB colour approach and random location*, PhD thesis, Universiti Utara Malaysia, Changlun, Malaysia, 2020.

[27] L. Y. Por, K. Wong, and K. O. Chee, *Unispach: A text-based data hiding method using unicode space characters*, J. Syst. Softw., 85 (2012), pp. 1075–1082.

[28] N. Provos and P. Honeyman, *Hide and seek: an introduction to steganography*, IEEE Security & Privacy, 1 (2003), pp. 32–44.

[29] S. G. Rizzo, F. Bertini, and D. Montesi, *Fine-grain watermarking for intellectual property protection*, EURASIP J. Inf. Secur., 2019 (2019).

[30] M. Shirali-Shahreza and M. Shirali-Shahreza, *A new approach to Persian/Arabic text steganography*, in 5th IEEE/ACIS International Conference on Computer and Information Science and 1st IEEE/ACIS International Workshop on Component-Based Software Engineering,Software Architecture and Reuse (ICIS-COMSAR'06), 2006, pp. 310–315.

[31] M. Shirali-Shahreza and M. H. Shirali-Shahreza, *Text steganography in SMS*, in 2007 International Conference on Convergence Information Technology (ICCIT 2007), 2007, pp. 2260–2265.

[32] M. H. Shirali-Shahreza and M. Shirali-Shahreza, *A new synonym text steganography*, in 2008 International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Harbin, China, 2008, pp. 1524–1526.

[33] M. Taleby Ahvanooey, Q. Li, J. Hou, H. Dana Mazraeh, and J. Zhang, *AITSteg: An innovative text steganography technique for hidden transmission of text message via social media*, IEEE Access, 2018 (2018), pp. 65981–65995.

[34] M. Taleby Ahvanooey, Q. Li, H. J. Shim, and Y. Huang, *A comparative analysis of information hiding techniques for copyright protection of text documents*, Secur. Commun. Networks, 2018 (2018).

[35] M. Taleby Ahvanooey, M. X. Zhu, W. Mazurczyk, Q. Li, M. Kilger, K.-K. R. Choo, and M. Conti, *CovertSYS: A systematic covert communication approach for providing secure end-to-end conversation via social networks*, J. Inf. Secur. Appl., 71 (2022), 103368.

[36] R. Thabit, N. I. Udzir, S. M. Yasin, A. Asmawi, and A. A.-A. Gutub, *CSNTSteg: Color spacing normalization text steganography model to improve capacity and invisibility of hidden data*, IEEE Access, 10 (2022), pp. 65439–65458.

[37] D.-C. Wu and Y.-T. Hsu, *Authentication of LINE chat history files by information hiding*, ACM Trans. Multimedia Comput. Commun. Appl., 18 (2022), pp. 1–23.

[38] L. Xiang, W. Wu, X. Li, and C. Yang, *A linguistic steganography based on word indexing compression and candidate selection*, Multimed. Tools Appl., 77 (2018), pp. 28969–28989.

[39] C. Xiao, C. Zhang, and C. Zheng, *Fontcode: Embedding information in text documents using glyph perturbation*, ACM Trans. Graph., 37 (2018), pp. 1–16.

[40] W.-C. Yang and L.-H. Chen, *A steganographic method via various animations in PowerPoint files*, Multimed. Tools Appl., 74 (2015), pp. 1003–1019.

[41] M. Zamani, A. B. A. Manaf, R. B. Ahmad, F. Jaryani, H. Taherdoost, and A. M. Zeki, *A secure audio steganography approach*, in 2009 International Conference for Internet Technology and Secured Transactions, (ICITST), London, U.K., 2009, pp. 1–6.

[42] S. Zhong, X. Cheng, and T. Chen, *Data hiding in a kind of PDF texts for secret communication*, Int. J. Netw. Secur., 4 (2007), pp. 17–26.

Please cite this article using:

Bahman Khosravi, Text steganography by changing the black color, AUT J. Math. Comput., 5(3) (2024) 233-244
https://doi.org/10.22060/AJMC.2023.21801.1111