Engineering and Technology Journal
Journal homepage: engtechjournal.org

A New Proposed Coding Method for Steganography Purpose

Suhad M. Kadhem [a]*

[a] Computer Science Department, University of Technology, Baghdad, Iraq, 110102@uotechnology.edu.iq

* Corresponding author.

Submitted: 23/12/2019 Accepted: 09/09/2020 Published: 25/03/2021

K E Y W O R D S

Coding, Run Length Encoding, Move to Front, Non-printed characters, steganography

A B S T R A C T

Transferring data in a safe way is one of the things that have been of interest since ancient times. Data hiding or steganography is a method used to protect the data during its transmission. Coding is a method to represent the data in another shape, so a high level of security is achieved by using coding and hiding data together. The proposed method is a hybrid between coding and hiding, but this paper focuses on proposed a data coding part only, such that the cover text (that used for information hiding) will be used to extract private information between the sender and the receiver for coding process, and the output of the proposed coding method can be used for information hiding process.

Apply the proposed coding method will provide a high level of security and complexity and produce ASCII of non-printed characters which can be employed for steganography purposes to obtain complete similarity between secret text and cover text.

How to cite this article: S. M. Kadhem, "A New Proposed Coding Method for Steganography Purpose" Engineering and Technology Journal, Vol. 39, Part B, No. 01, pp. 243-251, 2021.
DOI: https://doi.org/10.30684/etj.v39i1B.1510

1. INTRODUCTION

Coding is a method to represent the data in another shape, with the same arrangement. There are many methods of encoding data, some of them are open space methods that encode through manipulation of white space, syntactic methods that utilize punctuation, and semantic methods that encode using manipulation of the words themselves [1].

Steganography is the art and science of hiding a message inside another message without drawing any suspicion to others so that the message can only be detected by its intended recipient. Data hiding (Steganography) is a method used for data security purposes and to protect the data during its transmission. Text steganography is used to hide the communication between two parties by embedding a secret text inside another cover text to produce stegotext [2].

Run length encoding (RLE) is one of a compression techniques that is used to compress the data to decrease the redundant size of the data. "The goal of the run-length algorithm is to determine the runs and record the length of each run and the symbol in the run" [3]. The data are encoded into two

bytes, the first byte will be the number of times to duplicate symbols and the other one will be the value of the symbols [4].

Move to front (MTF) is a method of data compression. The basic idea behind this technique is that there is a set A of the alphabet, the indexing of this alphabet is used to make the compression. In addition, when making a coding to each character, its corresponding character is moved to the front of the set A [5].

Non-printed characters are a set of characters like space, a special character to indicate the end of a line or the end of a paragraph and so on. The advantage of these characters in the steganography field is that they do not appear on the screen [6], so the cover text and stegotext are similar.

In this paper, an approach for coding data that can be used for steganography purpose is proposed because the higher level of complexity is achieved (that lead to gain a high level of security) when both techniques are used together. MTF, RLE and non-printed characters are used as tools for the proposed method.

The rest of the paper is organized as follows: The second section will present the related work. The third section discusses the proposed method. Experimentation and results are demonstrated in section fourth and the fifth section provides the conclusion.

2. RELATED WORK

The following are some of the related works:

- In 2016, S. M. Kadhem [6] proposed a new text steganography method that's based on a parser and the ASCII of non-printed characters with modification for the Run Length Encoding method (RLE).
- In 2016, suhad M., et al. [5], has proposed a coding technique method where the MTF coding method is applied five times to provide a high degree of complexity that can be used for security purposes.
- In 2016 S. Kadhem, et al. [7], Proposed new coding method that is suitable for compression such that its output will be contains a sequence of ones with fewer zeros. This proposed method modifies RLE and use stenography based on Unicode and non-printed characters to hide the secret information in an Arabic text.
- In 2017, Md. Khairullahet at. [8], presents a novel approach for information hiding or steganography in the Bengali digital text. The main idea of the proposed method is to exploit this special feature of the Bengali alphabet to hide secret information in the form of bits. One of these two forms can be used to represent the bit '0' and the other form can represent the bit '1' in a document without any risk of understanding by any intermediate user. The results show that the proposed method is very prominent to be a successful steganography technique.

3. PROPOSED METHOD DESCRIPTION

Design a proposed coding method used for steganography purpose is concerned with two main subjects: coding and information hiding. This paper will focus on designing the coding method, and about the information hiding method this paper uses the idea described in a previous paper.

The proposed coding method consists of two sides: The sender side (encoding) and the receiver side (decoding).

I. The Sender Side (Encoding)

This paper introduces a new approach for coding text that can be used for steganography purposes. The input will be an English secret text and an English cover text; the output will be a set of non-printed characters that can be embedded in the cover text. In this method MTF, RLE and non-printed characters are used as tools to obtain coding, lossless compression, high level of security, high level of complexity and complete similarity between cover text and stegotext (see figure1 and algorithm1).

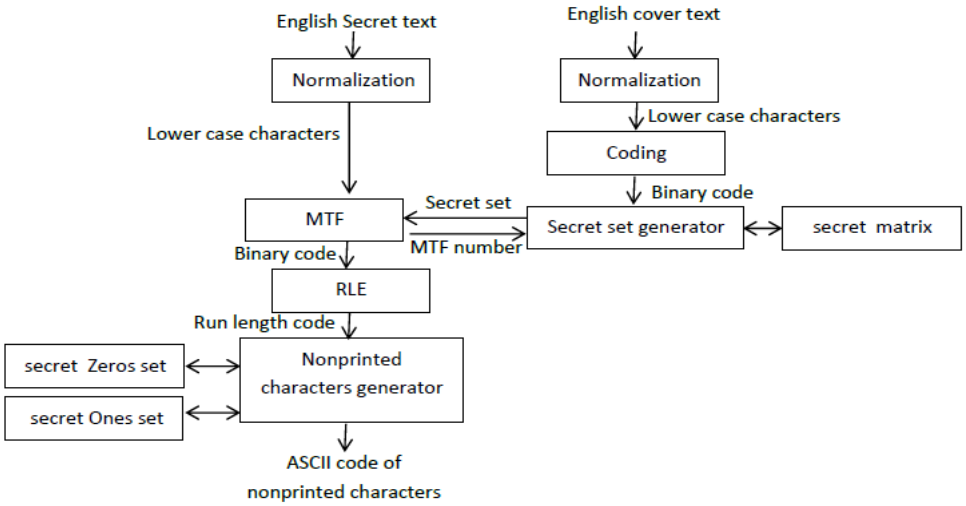

Figure 1: Block diagram of the proposed encoding method

Usually, the cove text is used for embedding the secret message only, but this paper uses the cover text not for embedding only but for extracting information that can be used for coding.

The original secret text can't be retrieved without private information between the sender and receiver that is unknown to the attacker, so in this paper, the sender and the receiver have the following private information:

- The number of applied MTF (n) will depend on the length of the first word in the cover text.
- Secret matrix (5*5) of the English alphabet that can be used to generate secret sets used by MTF (see Figure 2).
- Two secret sets of the ASCII code for non-printed characters (Figure 2 and Table I).

m	b	S	c	g
f	e	K	i	j
t	l	A	n	y
h	q	R	d	p
u	o	W	x	v

Figure 2: secret matrix

TABLE I: Swapping of Zero‘s Counters

Counter	ASCII of Non-Printed Character
0	18
1	15
2	14
3	20
4	25
5	17
6	19
7	16
8	12
9	21

Algorithm1: Encoding stage

Input: English Secret Text (**EST**), English Cover Text (**ECT**).

Output: ASCII code of non-printed characters (**NP**)

Process:

Step1: Normalize **EST** to a lower case form to be (**S**).

Step2: Normalize **ECT** to a lower case form.

Step3: Coding **ECT** into binary to be (**B**)./*using its ASCII code*/

Step4: n = length of first word in ECT

Step5: for (i=0 to n) do

5.1 Call algorithm2 that take (**B** and **i**) as input and return a secret set (**A**).

5.2 Call algorithm3 that takes (**S** and **A**) as input and returns set of integer values (**S2**) as output. /*apply move to front*/

5.3 Convert **S2** to characters to be **(S)** /* using ASCII code*/

Step6: Coding **S2** into binary to be **BC**.

Step7: Call algorithm4 that takes **BC** as input and returns (**C**). /*Apply run length encoding*/

Step8: Call algorithm5 that takes **C** as input and returns (**NP**).

Step9: Return (**NP**).

End.

TABLE II: Swapping of One's Counters

Counter	ASCII of Non-Printed Character
0	5
1	4
2	22
3	3
4	23
5	7
6	24
7	6
8	1
9	2

As illustrated in figure 1, after entering the secret English text the first process is the normalization process that converts the text into a lower case. After that MTF is applied to the secret English text for n times (where n depends on the length of the first word in the cover text) to increase the complexity and to make the algorithm not depend on static values. Now we must generate a secret random set each time we want to apply MTF, to do this we will use the cover English text and a secret matrix that contains the English alphabet as tools to generate such a secret set. The cover English text is normalized to a lower case also and converted to a binary code using its ASCII code. In order to generate a different random set each time we make a change in the secret matrix according to the MTF number such as if the MTF number is 3 then the third row in the secret matrix will be the first row. The generated binary code then will be used to get the indexes of the secret matrix such that we take each three bits and convert them to a decimal number to be the row and column numbers (if it is less than 5), these indexes will compose the secret set (see algorithm2).

Algorithm2: Generate secret set

Input: set of binary code (B), N: integer value, secret matrix of two dimensions that contains lower case English letters (M) of size (5*5).

Output: matrix (A) of size(26,2). /* generated secret set*/

Process:

Step1: if N >5 then N=N mod 5

Step2: Change M such that row N is moved to the front /*to get a new set each time we call this algorithm*/

Step3: k=0, i=0, j=0

Step4: While k<25 and i< length of B -3 do

4.1 let r be the decimal number that corresponds to the three binary bits: B[i], B[i+1] and B[i+2]

4.2 while (r > 4 and i< length of B -3) do

i=i+1

r = the decimal number that corresponds to the three binary bits: B[i], B[i+1] and B[i+2]

endwhile

4.3 j=i+1`

4.4 Let (c) be the decimal number that corresponds to the three binary bits: B[j+1], B[j+2] and B[j+3].

4.5 while (c> 4 and j< length of B -2) do

j=j+1

c = the decimal number that corresponds to the three binary bits: B[j], B[j+1] and B[j+2]

endwhile

4.6 temp = M[r,c]

4.7 If temp is not found in (A) then

A[k,0] =k, A[k,1]=temp, k=k+1 endif

4.8 i=i+1

endwhile

Step5: for ind='a' to 'z' do

if (ind) is not found in (A) then

A[k,0]=i, A[k,1]=ind, k=k+1

Step6: Return(A).

Using MTF (see algorithm3) will provide a compression ratio since its output is in a decimal form that takes five bits to be represented rather than eight bits and provide a high degree of complexity that depends on the number of how many MTF is applied.

Algorithm3: MTF

Input: set of characters (S), secret matrix (A).

Output: set of integers (S2)

Process:

Step1: for i=0 to length of S-1 do

Step2: j=0

Step3: *repeat*

If S[i]=A[j,1] then

S2[i]=A[j,0],

Change set A such that A[j,1] is moved to the front

Flag=true

Else

j=j+1

until flag

endfor

Step4: Return(S2).

End.

Now to generate the ASCII code of non-printed characters, RLE is applied after converting MTF output to a binary code (see algorithm4).

Algorithm4: run length encoding

Input: Binary code set (BC)

Output: matrix with two columns (RLE).

Process:

i=0,

while (i<= length of BC -1) do

Step1: C=0, j=i+1

Step2: while (BC[j]=BC[i]) do

C=C+1, j=j+1

Step3: Convert the counter C to a set of digits to be C1

Step4: for d=0 to length of C1 - 1 do

4.1RLE[d,0]=C1[d],

4.2 If BC[i]=0 then RLE[d,1]=0

Else RLE[d,1]=1

step5: i=j

Endwhile

Return (RLE).

End.

After that, two secret sets are used: S0 represents the mapping between the zeros counter and the ASCII code of non-printed characters (table I), and S1 represents the mapping between the ones counter and the ASCII code of non-printed characters (table II).

Since there are two types of bits (zero and one) so the bit's type will determine which set of the two specified sets will be utilized in swapping the counters, such as if the RLE output is four ones, this means, S1 will be used to make swapping the counter (four) with its corresponding ASCII of non-printed character (23). On the other side, when the receiver obtains ASCII 23, then he will understand that there are four ones by depending on S1 since 23 is appearing only in S1 that represents one's counter (see algorothim5), this ASCII code will be embedded in the cover text to generate the stegotext by using the method proposed by S. M. Kadhem, 2016 [5].

Algorithm5: producing non printed characters

Input: matrix with two columns (NBC), secret set of zeros(S0), secret set of ones (S1)

Output: set of ASCII code of non printed characters (NP)

Process:

For i=0 to length of BC-1 do

Step1: if BC[i,1]=0 then

Get the corresponding value (V) to BC[i,0] from set S0.

Change (S0) such that v will be moved to the front.

else

Get the corresponding value (V) to BC[i,0] from set S1.

Change (S1) such that v will be moved to the front.

endif

Step2: NP[i]=V

endfor

Return (NP).

End.

To increase the complexity of the proposed method there is a need to increase the probabilities of S0 and S2 (such that these two tables are dynamic), so in this method, the MTF concept is applied on S0 and S1 each time we get an index (moving the specified index to the front).

II. The Receiver Side (Decoding)

On this side, the same stages that used for encoding are applied, but in reverse order, such that when the receiver receives the stegotext, he extracts the ASCII code of non-printed characters and use it to retrieve the RLE code depends on the two secret sets S0 and S1 (table2 and table3), after that he will apply MTF for n times depends on the length of the first word in the cover text (since the cover text is not changed after embedding because of using non printed characters), the receiver also will generate the same secret sets for MTF because he has the cover text and the secret matrix (table1), the input to the MTF will be integer numbers and the output will be the secret text.

4. EXPERIMENTAL RESULTS

The experiment results showing in this part, lead to measure the performance of the proposed method. The compression ratio is computed using the following Eq. (1):

Compression ratio= size after compression / size before compression (1)

This part is used to calculate the change in the size of secret text due to the coding, compression and embedding process of the proposed method. First of all, we calculated the secret text size before

& after MTF as well as compute the compression ratio to determine the changes, the gain from using MTF to reduce the secret text sizes that will improve the hiding process by reducing cover capacity needed. The compression ratio is computed again after using RLE and its modification. (See Table III)
The results showing below that using MTF with modified RLE produce a good compression ratio; in fact, it is very useful with large secret texts.

TABLE III: Compression ratio of the proposed coding method

Example	Secret text length	Before comp.	After MTF Comp.	After modified RLE	Comp. ratio %
Text1	823	6584	4115	2181	33.126
Text2	454	3632	2270	1178	32.434
Text3	149	1192	745	371	31.124
Text4	156	1248	780	407	32.612

5. CONCLUSION

From this paper the following points can be concluded:

1) To gain a high level of security and to increase the complexity of the proposed method there is a need to increase the probabilities through:
   - Using both the coding method and information hiding method in the proposed system.
   - Making the proposed method depends on dynamic secret information such as making the stego key depends on the length of the first word in the cover text, and the secret sets depend on the cover text and MTF concept.
   - Applying the MTF method for n times (such that n depends on some information extracted from the cover text).
   - Applying MTF concepts on the secret sets (S0 and S1), such that each time an index is retrieved from these secret sets, it's rotated to the front.
2) The proposed method provides a good Compression ratio that can improve the hiding process by using the following:
   - MTF that convert eight bits to five bits.
   - RLE that provides a good compression ratio for identical data.
   - Modified RLE for un identical data to solve the RLE problem.
   - Non-printed characters that used to embed a block of bits each time rather than a single bit.
3) Complete similarity between cover text and stego text is achieved in the proposed system since we embed non-printed characters in the cover text only.
4) Using cover text to extract private information between sender and receiver since the cover text is not changed after embedding because of using non-printed characters.

References

[1] T. Gagie and G. Manzini, "Move-to-Front, Distance Coding, and Inversion Frequencies Revisited", Universit`a del Piemonte Orientale, 2010.

[2] M. Agarwal, " Text Steganographic Approaches: A Comparison", International Journal of Network Security & Its Applications (IJNSA), Vol.5, No.1, January 2013, pp.91-106.

[3] D. Salomon, "Data Compression- The Complete Reference", Third Edition, 2004.

[4] Mengyi Pu, "Fundamental Data Compression", Butterworth-Heinemann, ISBN:978-0-7506-6310-6 2005.

[5] S. M. Kadhem and S. Jassm "Proposed Data Coding Method," Eng. &Tech. Journal, Vol 34, Part (B), No. 2, 2016, pp. 194–203.

[6] S. M. Kadhem, "Text Steganography Method Based on Modified Run Length Encoding" Iraqi Journal of Science, Vol. 57, No.3C, 2016, pp:2338-2347.

[7] S. Kadhem and D. Wameedh "Proposed Arabic Text Steganography Method Based on New Coding Technique,"ISSN : 2248-9622, Vol. 6, Issue 9,(Part -1) September 2016, pp.38-46.

[8] Md. Khairullah and Md. Abu Shahriar Ratul, "Steganography in Bengali Unicode Text", SUST Journal of Science and Technology, Vol. 27, No. 1, 2017, P: 71-80.

[9] R. R. Baruah, et al., "Enhancing Dictionary Based Preprocessing For Better Text Compression", International Journal of Computer Trends and Technology (IJCTT) – volume 9 number 1– Mar 2014, pp:2338-2347.