2020 International Conferences on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on

A High Capacity Text Steganography Utilizing Unicode Zero-Width Characters

Hafsat Muhammad Bashir[1*], Qianmu Li[1,2*]
[1]*School of Computer Science and Enginering*
Nanjing University of Science and Technology
Nanjing, China
2Intelligent Manufacturing Department, Wuyi University,
Jiangmen, China
muhd.hafsat@njust.edu.cn; qianmu@njust.edu.cn

Jun Hou[3]
3 School of Social Science
Nanjing Institute of Industry Technology
Nanjing, China
houjunnjust@163.com

***Abstract*—Information hiding based on text has become the hottest and most discussed aspect of steganography in recent years, because it eases communication and consumes less memory compared to other types. The lack of redundant information in a text file makes it difficult to hide secret bits into a carrier message (CM) without being noticeable. Therefore, the total amount of hidden bits embeddable in a CM contributes immensely to the system efficiency. Today, individuals share information online with the fear that it could be destroyed by third parties. To achieve a structured text hiding system with high embedding capacity, imperceptibility, robustness and high security concurrently, securing sensitive data through encryption is not enough because attackers are able to suspect that a communication channel exist between the sender and the receiver. Therefore, this paper combines both steganography and cryptography to protect our shared information and to achieve a strong and highly secured non understandable text hiding system. We proposed a method which uses pairs of Zero Width Characters (ZWC) to represent each 4-bit binary classification which provides a high embedding capacity and reduces computational complexity than previous methods. The embedding is done by placing the invisible strings of the secret message (SM) into a single position of the open spaces in the CM which provides high invisibility compared to the reviewed methods. One Time Pad (OTP) and XOR bitwise encryption is used to encrypt the inverted message before the embedding process which provides end-to-end security to the system. The experiment shows that the proposed method provides a very high embedding capacity and high robustness than other literature reviewed.**

Keywords—*Information hiding, Text Steganography, Cryptography, Zero Width Characters (ZWC), One Time Pad (OTP), XOR bitwise, Carrier message (CM), Secret message (SM)*

I. INTRODUCTION

Internet today has made interaction easier and everyday large amount of sensitive data is communicated either inform of text, audio, video or images. This exposes sensitive data to various attacks and urges individuals to protect their valuable information against hackers and third parties and keep communication private [1]. In the years back, Individuals and organizations mainly protect their valuable information using cryptographic encryption methods, by encrypting the message it is only protected from individual hackers viewing that sensitive information which is not highly secured. Through this encrypted communication, third parties were able to suspect that information is being conveyed and will therefore attempt to destroy it. Hence, with the application of Stenographic methods, information can be hidden without any trace of suspicion from unauthorized users. Cryptography and Information hiding are the two techniques that secure information transmission and protect valuable information. Information hiding expands broadly into steganography and watermarking. While watermarking conceals secret message (SM) into another media where the identity of the owner is claimed, steganography embeds SM into a cover medium and hide the very existence of that communication [2, 3]. steganography prevents attackers from suspecting any means of communication taking place [4] and helps to achieve a high amount of security, robustness with high imperceptibility against visual attack [5]. Cryptography on the other hand hides the SM and makes it unreadable to unauthorized users. Combining both steganography and cryptography, rise into a strong and secure system with high invisibility and non-breakable against various attacks.

Steganography meaning "Cover writing" is an ancient term and comprises of text in its origin which now extends to other types [6]. It embeds SM into a selected cover message (CM) in an unnoticeable manner and hides its very existence. It hides information with the help of an optional stego key which is shared only between the sender and the receiver [7, 8]. Steganography is divided into different types based on cover medium as illustrated in Fig. 1.

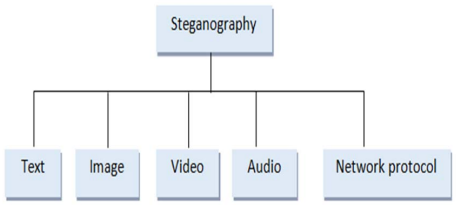

Fig. 1. Types of Steganography

Text steganography make use of a text as a cover medium to hide and transmit a piece of information. Text is light and eases communication; therefore to hide data into a text file we have to look into its characteristics and properties of its characters and the way they are organized. It is difficult compared to other types because of its lack of redundant information [9, 10] and any small modification or alteration is easily noticeable due to its structural and visual appearance [7].

This work was supported in part by the National Key R&D Program (2020YFB1804604) of, the 2020 Industrial Internet Innovation and Development Project from Ministry of Industry and Information Technology of China, the Fundamental Research Fund for the Central Universities (30918012204).

978-1-7281-7647-5/20/$31.00 ©2020 IEEE
DOI 10.1109/iThings-GreenCom-CPSCom-SmartData-Cybermatics50389.2020.00116

Unicode Standard has become the most common standard for processing digital text in computer system today. In this paper, UTF-8 is used to represent the binary bits of the SM and the Unicode Zero Width Character (ZWC) is used to represent each 4-bit length of the SM binary strings. Unicode standard is a universal character encoding system that supports the universal display and replacement of characters with different languages and technical disciplines.

Our research work has contributed immensely in security of data in the world of internet and can be applied in copyright protection and content authentication. Our proposed method is able to provide secret communication between parties without being noticeable or leaving a trace to be breached.

This paper is divided into five sections: the first section is introduction, the second section is definition of related terms, followed by related studies in the third section, proposed method and algorithms are explained in the fourth section and lastly we have our results and conclusion in the last section.

II. Related Reviews

Recently, a lot of reviews have been carried out related to text steganography but each review has its own level of accuracy and security, our proposed method look into these reviews and tries to cover up the lapses and improve both capacity and robustness of the system using a better approach than the previous methods. This section defines some studies relevant to the proposed scheme and looks into related works proposed by other authors.

1. Literature review

In years back, researchers mainly presented their work based on image, audio and video, but lesser attention is given in the field of text because of its lack of excess bits [11]. In [12], the author uses image as a cover medium to hide secret data using improved Least Significant Bits (LSB) algorithm of color image and RSA algorithm to encrypt the hidden message before embedding into the CM using a secret key. Another study is proposed in [13] which combine both cryptography and steganography to hide secret bits into the LSBs of the approximation coefficients of the Integer wavelet transform of gray scale images and component of the color images to form the cover image. Using the audio as a cover medium, [11] proposed a method base on Patch wise Code Formation techniques to hide video within another video which gives better robustness and secure video transmission.

Text is light and lack redundant information compared to audio, video and image which made it a very challenging area and researchers are now putting more efforts in exploring different methods and efficient algorithms to hide text into another text without a trace. Most studies proposed were able to hide data into the CM successfully but failed to achieve all the four text hiding criteria at a time which include invisibility, distortion robustness (DR) against various attacks, embedding capacity (EC) and security.

Most of today's works are concerned with the embedding capacity neglecting both security and robustness of the system. Kashida-based method is proposed in [14] which utilizes the Arabic extension character (Kashida) and the white space to embed the SM into the CM. In this method, the binary string of the SM is divided into 3-bit length, if the bits equal "100", the normal white space is put together with Thin space Unicode character, if the bits equals "010" the normal white space is put together with the Hair space Unicode character and if the bits equals "001" the normal white space is put together with the Six Per-Em space Unicode character. "000" takes no space and "111" represent combination of all the three Unicode spaces and the presence of Kashida itself hides one bit. This approach provides high EC, but low invisibility to human visual system. It is not applicable to embedding long SM into a short CM, and once the attacker tempers with the stego text either by deletion or changing the words, the SM is destroyed. Another method called the UniSpaCh is proposed in [15] which use the normal space together with the Unicode space characters to embed the SM between words, inter-sentence, end-of-lines and paragraph spaces. The method uses 2-bit classification (00, 01, 10, 11) which is replaced by the Unicode special spaces (Hair, Thin, Six-Per-Em, punctuation) to embed the secret bits into the CM. UniSpaCh provides high invisibility but fails where the CM is shorter than the SM and achieves low EC using 2-bit per space.

Another work is presented in [16] called the Bit Mapping approach which use the OTP encryption algorithm to encrypt the SM bits and then maps each bit of the SM to the first bit of each character in the CM respectively using a stego key. The method maps the bits in sequence and record the sequence number which serves as the key. This method provides high invisibility but low DR because once the stego key is exposed; the secret bits will be destroyed. It has low EC because each bit of the SM map only one bit of each character in the CM. [17] extended the UniSpaCh method by using 3-bit and 4-bit (3&4SpaCh) classification to hide SM into Microsoft word documents. The method uses the Unicode space characters to represent each bits classification based on a predefine pattern. 4-bit are embedded if the space is between words or inter-sentence and 3-bit for end-of-line and inter-paragraph spaces. The method provides high EC than UniSpaCh method, but it generates an extra space between the words due to the use of Unicode space characters. A text-based hiding technique is introduced in [18] which use both white space and extended line. This method adds extra white space to the normal spaces between words, end-of-lines and end of paragraph to embed the secret bits. It provides high invisibility but low EC and DR against visual attack. Two methods for Arabic texts were proposed in [19]. The first method is an improvement of capacity of the author's previous work while the second method (4SpaCh) make use of four Unicode characters (Pseudo-space, Thin space, Hair space and Zero-Width space) together with the normal space to represent each 4-bit length. The algorithm then embeds the Unicode characters between the word spaces of the CM. The method achieves high EC and high invisibility but low DR to visual attack and produces unconventional spaces between words.

An efficient Hybrid method is proposed in [20] which use both Unicode and ZWC to embed permuted and inverted secret bits. The algorithm uses sixteen 4-bit possible combination and divides it into four groups based on their left most 2-bit pairs that is the same for each four 4-bit classification. For example, the first group is based on "00", the second group is based on "01", the third group is based on "10" and the forth group is based on "11". The four groups are used as a key and the other right most 2-bit pairs in each combination are used for the embedding in inter-word, inter-sentence, inter-paragraph and end-of-line spaces based on a predefined pattern of each key. Each space hides in four bits by inserting both Unicode space character and ZWC at the same location. This approach provides high imperceptibility and high EC than UniSpaCh but cannot be used to embed long SM into a short CM. A text based hiding technique called AITSteg is presented in [21] which uses Gödel function to generate pair of numbers for all characters of the SM and obtain a hashed binary string according to those pair of numbers. The method uses 2-bit classification to represent one ZWC which is embedded into the CM. This approach provides high invisibility and high DR but low EC using only 2-bit classification.

With the above literature reviews, there is need to improve the embedding capacity, robustness and security of the system, therefore we propose a method which uses a combination of Unicode ZWCs to represent sixteen 4-bit classification ("0001", "0010", "0011", "0100", "0101", "0110", "0111", "1000", "1001", "1010", "1011", "1100", "1101", "1110", "1111", "0000"). The SM is first encrypted using OTP encryption, which is then converted into binary string. The binary string is inverted using ones complement which makes it more complex and secured. We use XOR encryption between the inverted pairs of 4-bit substring of each 8-bit character of the SM and then merge the first original 4-bit substring of that character to the XORed 4-bit length generated to form a new binary string. We replace the new strings with invisible Unicode characters which are embedded at the beginning before the characters of the CM. The proposed method provides high EC and DR with no traces of the SM in the stego text (ST), and can embed a long SM into a short CM without being noticeable.

2. *Text hiding Techniques*

Text hiding is divided into different techniques which are Structural or Formal, linguistic and Random and Statistical technique [22, 23]. Formal based technique hides the SM by considering the physical appearance of the CM without changing a word or the entire sentence. The algorithm involves modifying the layout features of the CM such as fonts, word/sentence spacing, text color, line spacing [24, 25]. Format based technique is further divided into line/word shift, white space, feature encoding and zero-width approach. Line shift involves shifting the lines either vertically up or down, while word shifting requires shifting the words horizontally to left or right to hide secret bits. It is not applicable because once an unauthorized user temper with the CM either by copying or moving it to a new location, the hidden message will be lost [24, 15, 26]. The whitespace method uses special Unicode spaces to hide the SM into the CM. The bits are embedded into spaces between words, sentence, and end of line or paragraph spaces. This method provides high invisibility against visual attacks [15, 27, 18]. Zero width approach uses the Unicode non width characters to embed the SM. These characters have no traces and provide high invisibility against visual attacks [24, 21].

The linguistic technique uses Natural Language Processing (NLP) algorithms to alter the syntax and semantics of a word in a CM such as nouns, synonyms substitution, abbreviations, the similarity between words, etc. to hide the secret bits [28, 29]. The Random and Statistical technique generate random sequence of letters as CM using a defined set of algorithms to hide the secret bits. It looks at the statistical features of the SM such as the way words or letters are repeated, word length etc. to generate a CM with similar properties as the SM. This method is slow compared to other techniques [30].

3. *Unicode Zero Width Characters*

The Unicode is a universal coding system for the processing and encoding of digital text which has been in existence since 1987 for different languages. Unicode standard is implemented in various programming languages and operating systems such as windows, IOS, android etc. for displaying digital text. It consists of three encoding formats which are UTF-8, UTF-16, and UTF-32.

Most researchers use the UTF-8 format which presents one byte for ASCII character and up to four bytes for other characters while UTF-16 uses 16-bits to encode a character [24]. Unicode standard has ZWCs which are special characters that have no width and written symbols and leave no traces when used to hide text into a cover medium [17]. ZWCs are used by many researchers today to hide data which provide high invisibility against visual attack. The proposed method uses a combination of eight ZWCs to represent sixteen 4-bit length classification of the SM binary, which are shown in Table 1.

TABLE I.

ZWCS PATTERN FOR THE 4-BIT BINARY CLASSIFICATION

4-bits classification	Hex code
0001	200C+202A
0010	202A+200C
0011	202C+202A
0100	202A+202C
0100	202D+202C
0101	202C+202D
0110	200C+202C
0111	202C+200C
1000	202A+202B
1001	202B+202A
1010	200E+202D
1011	202D+200E

1100	200E+202C
1101	202C+200E
1111	202D+200C
0000	200C+202D

III. The Proposed Method

In this paper, we proposed a method based on ZWCs and 4-bit binary classification to embed secret bits into a text file. This method provides high invisibility to observers and attackers who look up to the slightest opportunity to breach the information. The proposed method is divided into the embedding and extraction process with both sides of the sender and receiver in detail.

1. Embedding algorithm

The embedding process is divided into different steps as follows.

- **Step 1**: The message is first encrypted using OTP encryption with the help of a strong Key (K). This makes the secret message hard to break.
- **Step 2**: The encrypted message is then converted to binary string and inversion process is performed using ones complement that is, if the bit is "1", we invert it to "0" and if the bit is "0", we invert it to "1". For example, if the SM binary string is "01000001" then its ones compliment will be "10111110".
- **Step 3:** The inverted bits are then divided into 4-bit substring and XOR encryption is performed between the two pairs of substring obtained from each character. We therefore perform XOR operation between "1011" and "1110" to get another 4-bit string "0101". The XORed 4-bit string ("0101") is merged together with the first 4-bit substring of the original binary string of the character to get a new string "10110101".
- **Step 4:** The new string formed in step 3 is divided again into 4-bit lengths and replaced by pairs of zero width Unicode characters based on a predefined pattern as illustrated in Table 1. The invisible symbols generated are put together to form a string which is then embedded at the beginning before the characters of the CM.

The above steps are performed for all the characters of the secret message which result to a new encrypted, inverted and XORed binary string. These processes increase complexity, robustness and security of the system.

2. Extraction algorithm

The extraction algorithm first checks the existence of invisible strings within the ST. The algorithm extracts the invisible symbols in form of a string, and pairs of these symbols are checked to see which of the 4-bit substring is hidden in them. For example, if "202D+200E" is extracted, we replace it by "1011" else if "202C+202D" is extracted, we replace it by "0101" following a predefined pattern until all the invisible strings are replaced by 4-bit substring. After extracting all the bits, we perform XOR operation between each 4-bit substring and its subsequent one and then merge the result to the first 4-bit substring of each pair used for the XOR operation, to form a new decrypted XOR binary string. For example, we have "1110" when we XORed the two 4-bit substrings from our embedding example "1011" and "0101" and then merge it to the first 4-bit "10111110". We finally invert the bits and apply OTP encryption to extract the original SM.

Algorithm 1 Embedding algorithm
Input: Cover Message (CM), Secret Message (SM), encryption Key (K)
Output Stego Text (ST) which contains the hidden bits of the SM.

1. **SM** ⟵ Secret Message.
2. **CM** ⟵ Cover Message.
3. **K** ⟵ encryption Key.
4. The SM is encrypted by OTP encryption using a strong encryption key K to get a string Enc_M.
5. Convert Enc_M into binary string SM_binary using UTF-8 format.
6. Invert all binary bits in SM_binary using ones complement to get a string Inverted_SM_binary.
7. Separate Inverted_SM_binary into 4-bit blocks (e.g. '0011','0001').
8. Perform XOR operation on each 4-bit block and its subsequent one to get a new encrypted string XOR_Enc_binary.
9. Pick the first 4-bit block of each 8-bit classification from the Inverted_SM_binary and merge it to its corresponding 4-bit block obtained from computing the XOR between that that block and its pair to get a new string Hidden_bits.
10. Replace each 4-bit block in Hidden_bits by pairs of ZWCs based on Table 1 predefined pattern to form a new string Invisible_SM.
11. **ST** ⟵ Invisible_SM+CM (embed the invisible strings containing the SM bits before the CM).

Algorithm 2 Extraction algorithm
Input: Stego Text (ST), OTP key (K)
Output Secret Message (SM)

1. **ST** ⟵ Stego Text.
2. Read ST to check the existence of invisible characters.
3. Store each invisible character detected in a string Invisible_SM.
4. **for each** pairs of characters (p_i) in Invisible_SM **do**
5. **switch** Invisible_SM [p_i] {
6. **case** 'u200C+u202A':
7. Hidden_bits = Hidden_bits + "0001"; break;
8. **case** 'u202A+u200C':
9. Hidden_bits = Hidden_bits + "0010"; break;
10. **case** 'u202A+u200C':
11. Hidden_bits = Hidden_bits + "0010"; break;
12. **case** 'u202C+u202A':
13. Hidden_bits = Hidden_bits + "0011"; break;
14. **case** 'u202A+u202C':
15. Hidden_bits = Hidden_bits + "0100"; break;
16. **case** 'u202D+u202C':
17. Hidden_bits = Hidden_bits + "0100"; break;
18. **case** 'u202C+u202D':
19. Hidden_bits = Hidden_bits + "0101"; break;
20. **case** 'u200C+u202C':
21. Hidden_bits = Hidden_bits + "0110"; break;
22. **case** 'u200C+u202C':
23. Hidden_bits = Hidden_bits + "0111"; break;
24. **case** 'u202A+u202B':
25. Hidden_bits = Hidden_bits + "1000"; break;
26. **case** 'u202B+u202A':
27. Hidden_bits = Hidden_bits + "1001"; break;
28. **case** 'u200E+u202D':
29. Hidden_bits = Hidden_bits + "1010"; break;
30. **case** 'u202D+u200E':
31. Hidden_bits = Hidden_bits + "1011"; break;
32. **case** 'u200E+u202C':
33. Hidden_bits = Hidden_bits + "1100"; break;

```
34.      case 'u202C+u200E':
35.        Hidden_bits = Hidden_bits + "1101"; break;
36.      case 'u202D+u200C':
37.        Hidden_bits = Hidden_bits + "1111"; break;
38.      case 'u200C+u202D':
39.        Hidden_bits = Hidden_bits + "0000"; break; }
40. end for
41.   Divide the Hidden_bits into 4-bit blocks (e.g. '0001', '0110')
      and XOR two subsequent blocks at a time without repetition to
      get a string XOR_Enc_binary.
42. Select the first 4-bit block of each 8-bit classification in
      Hidden_bits and merge to each corresponding 4-bit XORed
      block obtained in XOR_Enc_binary to form a string
      Inverted_SM_bits.
43. Perform inversion process on Inverted_SM_bits to decrypt the
      bits to SM_binary.
44.  Decrypt SM_binary using a key K ⇐ Dec_M.
45. Dec_M to binary using UTF-8 format.
46. Return SM.
```

IV. Experimental Result And Comparison

In this section, we display our experimental result based on the efficiency of the proposed method. We evaluated and compare the result with other reviews against two most important text hiding criteria which are EC and DR. Suppose we want to send the SM "**Thank you dear.**" to the intended receiver, we first select our CM to be "**Good day!**" which is shorter than the message to be sent. Table 2 shows the details of the embedding process.

1. Embedding Capacity

Because of the use of ZWCs in our proposed method to successfully hide the SM into the CM, we are able to achieve high invisibility with no traces of the SM in the ST. ZWCs have no width or symbol and do not raise any suspicions to attackers and human visual systems thereby making the hidden bits difficult to be cracked. Our proposed method uses four ZWCs to hide one character (each 4-bit classification is represented by two ZWCs which provides high embedding capacity compared to the work in AITSteg [21] which uses six ZWCs to hide one character. AITSteg [21] uses 12-bit to encode each character; each 2-bit is replaced by one ZWC which makes a total of six ZWCs.

We have greatly improved the embedding capacity of AITsteg [21] using 4-bit classification and four ZWCs to encode each letter of the SM. We also compare our method with the work in 4SpaCh [19] which made use of ZWCs combined with Thin and Hair space Unicode characters to embed the hidden bits. This method can embed up to 4-bit per space location but the EC depends solely on the number of spaces between the words and does not support short CM. Our proposed method hides long SM into a short CM regardless of the number of spaces between the words or size of the CM therefore provides high EC than 4SpaCh [19].

$$\text{Embedding Capacity (EC)} = \text{Number of bits per location} \times \text{Number of embeddable location} \quad (1)$$

$$\text{Capacity ratio (CR)} = \text{Number of hidden bits (byte)} / \text{Total number of characters} \times 100 \quad (2)$$

We apply (1) and (2) on Table 3 to show the Comparison between AITSteg [21] and the proposed method using character limitation of some Social media sites we extracted from [21]. The result shows that our proposed method increases the capacity ratio of AITSteg [21] by 8%.

2. Robustness

The ST is exposed to multiple kinds of attacks which can destroy the hidden message. An attacker tries to temper with the ST either by deletion, insertion or changing the order of the characters of the ST. Either of these attacks, cannot destroy the hidden SM since our proposed method hides the SM bits at the first position before the CM and therefore is independent of the changes made to CM.

a) Deletion attack: Deletion attack occurs when an attacker has access to ST and will try to alter the file either by deleting some parts or all the characters. When either of these actions is performed, the hidden message will remain before the CM because the attacker only deletes the characters that are visible to the eye without tempering the invisible strings before it. This is because the attacker has no idea of the location of the hidden message in ST.

b) Insertion attack: Insertion attack occurs when an attacker tries to add some words or characters either at the beginning of the ST or at the end or at the middle which changes the ST and increases its size. Insertion can also be dispersed at random locations in the text. Either of these actions performed cannot destroy our hidden message because both the SM and the CM are not dependently joined together. Both are separated in different locations and therefore inserting new characters or words will only increase the size of the CM but will not affect the invisible string embedded at the beginning of the ST.

We have tested the DR of our proposed method on some text samples in Table 4 and we have successfully achieved a high DR against various attacks with low losing probability. Our experiment shows that no matter the amount of alteration done on the ST; the hidden bits will still remain hidden at the beginning of the ST. The robustness of ST (DR_{ST}) can be calculated using the equation as follows [24]:

$$DR_{ST} = [1 - LP] \quad (3)$$

Where LP = Number of embeddable location/Length of the CM

For example, using the ST from our example in Table 2, we can calculate the robustness using (3) as follows:

LP = 1/10 = 0.1
DR = [1 – LP] = [1 – 0.1] = 0.9 x 100 = 90%.

3. Proposed Method Capacity and Robustness comparison with other reviews

For comparison purposes, we evaluated the performance of the proposed method with other reviewed methods based on EC and DR. We selected three methods from the literature which shows that our proposed method is the only method that achieved high EC and DR at the same time compared to other methods that achieved either low capacity and high robustness or high capacity and low robustness. Table 5 and Table 6 show the comparison result using different text samples. We use short CM to embed short SM

(SoS), short SM on long CM (SoL), long SM on short CM (LoS) and then long SM on long SM (LoL) to make our comparison.

TABLE II. EMBEDDING PROCESS IN DETAIL

Elements	Values
SM	Thank you dear.
CM	Good day!
OTP_Key	YOHYFGXBKGIPJFJSJQZFOMMYUMCGZVQMBKABPJJSRVMTAJYURNXKWULOKBDNNPJR
Encrypted SM	')7-g!->g-5+4d
SM binary string	"00100111, 00101001, 00110111, 00101101, 01100111, 00100001, 00101101, 00111110, 01100111, 00101101, 00110101, 00101011, 00110100, 01100100"
Inverted_SM using ones complement.	"11011000, 11010110, 11001000, 11010010, 10011000, 11011110, 11010010, 11000001, 10011000, 11010010, 11001010, 11010100, 11001011, 10011011"
4-bit block classification	"1101, 1000, 1101, 0110, 1100, 1000, 1101, 0010, 1001, 1000, 1101, 1110, 1101, 0010, 1100, 0001, 1001, 1000, 1101, 0010, 1100, 1010, 1101, 0100, 1100, 1011, 1001, 1011"
Applying XOR operation between pairs of blocks.	"0101, 1011, 0100, 1111, 0001, 0011, 1111, 1101, 0001, 1111, 0110, 1001, 0111, 0010"
Merge each first 4-bit block in Inverted_SM with the corresponding XOR result.	"11010101, 11011011, 11000100, 11011111, 10010001, 11010011, 11011111, 11001101, 10010001, 11011111, 11000110, 11011001, 11000111, 10010010"
Take 4-bit substring and replace with its corresponding Unicode ZWCs based on Table 1.	"200E+202C+202D+202C"+"200E+202C+200E+202D"+"202D+200E+202A+202C"+"200E+202C+202D+200C"+"202A+202B+200C+202A"+"200E+202D+202C+202A"+"200E+202C+202D+200C"+"202D+200E+200E+202C"+"202A+202B+200C+202A"+"200E+202C+202D+200C"+"202D+200E+202C+202D"+"200E+202C+202A+202B"+"202D+200E+200C+202C"+"202A+202B+202A+200C"
Invisible strings (no width no symbol)	""
Stego Text, ST (Invisible strings+ CM)	Good day!

TABLE III. COMPARISON OF CAPACITY RATIO OF AITSTEG [21] AND PROPOSED METHOD USING CHARACTER LIMITATION OF SOME SOCIAL MEDIA SITES [21]

Social Media	Number of characters limits (UTF-8)	EC (AITSteg)	Capacity Ratio (%) (AITSteg)	EC (Proposed method)	Capacity Ratio (%) (Proposed method)
SMS	1024	170	16.60	256	25.00
Facebook	640	106	16.56	160	25.00
Whatsapp	30,000	5,000	16.67	7500	25.00
Wechat	16,207	2701	16.67	4051	25.00
Viber	7000	1166	16.66	1750	25.00
QQ	16,207	2701	16.67	4051	25.00
Skype	400	66	16.50	100	25.00
Yahoo	2048	341	16.65	512	25.00

TABLE IV. DR OF THE PROPOSED METHOD USING SOME TEXT SAMPLES

Text Samples	Number of characters	DR (%)
Sample 1	18	94.7
Sample 2	36	97.3
Sample 3	50	98.0
Sample 4	81	98.8
Sample 5	92	98.9
Sample 6	162	99.4
Sample 7	500	99.8

Because both Hybrid [20] and 3&4SpaCh [17] have the same embeddable locations, they are able to achieve the same DR but with different embedding capacity. Bit Mapping [16] has poor DR because the algorithm embeds the secret bits within bits of the characters of the CM which makes the SM to be totally destroyed either by deletion or changing the entire characters of ST. We have excluded the AITSteg [21] in the comparison because both the proposed method and AITSteg [21] embed the secret bits in only one location of the CM. But the proposed method achieves higher EC with smaller cover text size in memory as shown in Table 3 because it uses four ZWCs to embed single character compared to AITSteg [21] which uses six ZWCs to embed a single character.

Our research group has also done a lot of relevant adaptation experiments and research on security hardening and security defense methods in different application scenarios [31-42], which can be used in conjunction with the methods proposed in this paper.

TABLE V. COMPARISON OF EC OF PROPOSED METHOD AND OTHER EXISTING REVIEWS

Type	SM	CM	Total Length (SM)	Total Length (CM)	Embedding Capacity (Bits)				
					Proposed method	Hybrid [32]	4SpaCh [30]	3&4SpaCh [23]	Bit Mapping [25]
SoS	Hi.	Okay.	3	5	24	8	0	3	5
SoL	Hello.	How are you dear?	6	17	48	32	12	15	17
LoS	What is your name?	Good day.	18	9	144	16	4	7	9
LoL	The academic year is full of hard work and experience. We are grateful to all students who have participated in all the activities.	We have successfully award each scholar based on merit which shows that no matter how little it is, always appreciate the hard work in people.	131	142	1048	112	96	108	142

TABLE VI. COMPARISON OF DR OF PROPOSED METHOD AND OTHER EXISTING REVIEWS

Type	SM	CM	Total Length (SM)	Total Length (CM)	Robustness (%)				
					Proposed method	Hybrid [31]	4SpaCh [30]	3&4SpaCh[23]	Bit mapping [25]
SoS	One	Hi!	3	3	75	67	0	67	0
SoL	It is raining.	How is your day?	14	16	94	75	81	75	0
LoS	The title of the book is amazing.	Good job.	33	9	90	78	89	78	0
LoL	To achieve what you want, you need to work hard and focus on what you do. Education is the best Key.	It is always not good to take any wrong prescription without seeking the advice of a doctor. Health is wealth.	100	110	99	88	76	85	0

V. CONCLUSION

The proposed method uses zero-width format-based text steganography which utilizes the Unicode zero width and no symbol characters together with the open space before the cover text to embed secret bits. The proposed method uses 4-bit classification of the hidden bits which provides high embedding capacity and efficiency with less stego text size. We have achieved high DR by embedding the invisible Unicode characters in only one location of the CM and high security using inversion process, OTP and XOR encryption. We have applied our algorithm in different text samples and also compare our method with other existing techniques, which shows that the proposed method achieves high embedding capacity, robustness, invisibility and security.

ACKNOWLEDGMENT

This work was supported in part by The 4th project "Research on the Key Technology of Endogenous Security Switches" (2020YFB1804604) of the National Key R&D Program "New Network Equipment Based on Independent Programmable Chips" (2020YFB1804600), the 2020 Industrial Internet Innovation and Development Project from Ministry of Industry and Information Technology of China, 2018 Jiangsu Province Major Technical Research Project "Information Security Simulation System", the Fundamental Research Fund for the Central Universities (30918012204), Cooperative research project commissioned by Jiangsu Zhongtian Internet Co., Ltd..

REFERENCES

[1] A. M. Eskicioglu and L. Litwin, "Cryptography," IEEE *Potentials, vol*. 20, no. 1, pp. 36-38, 2001.

[2] I.J. Cox, M.L Miller, J.A Bloom, J. Fridrich, and T. Kalker, "Digital Watermarking and Steganography," Amsterdam/Boston: Morgan Kaufmann Publishers, 2008.

[3] .J. Cox, M.L Miller, J.A Bloom, J. Fridrich, and T. Kalker, "Digital Watermarking and Steganography," Amsterdam/Boston: Morgan Kaufmann Publishers, 2008.

[4] N.F. Johnson and S. Jajodia, "Exploring steganography: seeing the unseen," *IEEE Computer*, vol. 31, no. 2, pp. 26-34, 1998.

[5] Z. Wang, C. Chang, C. Lin, and Ming C., "A reversible information hiding scheme using left–right and up– down Chinese character representation," *J. Syst. Softw*., vol. 82, pp. 1362–1369, 2009.

[6]] R. Gupta, S. Gupta, and A. Singhal, "Importance and Techniques of Information Hiding: A Review," *Int. J. Comput. Trends Technol.*, vol. 9, no. 5, pp. 260–265, 2014.

[7] R. B Krishnan, P. K Thandra, M.S Baba, "An overview of Text Steganography" 4th International Conference on Signal Processing, Communications and Networking (ICSCN), pp 16 – 18, 2017.

[8] P. Johri, A. Mishra, and S. Das, "Survey on steganography methods," *2016 3rd Int. Conf. Comput. Sustain. Glob. Dev*., pp. 2906–2909, 2016.

[9] M. P. Uddin, M. Saha, S. J. Ferdousi, M. I. Afjal and M. A. Marjan, "Developing an Efficient Solution to Information Hiding through Text Steganography along with cryptography," in *the 9th International Forum on Strategic Technology (IFOST)*, Bangladesh, 2014.

[10] K. K. Mandal, A. Jana and V. Agarwal, "A New Approach of Text Steganography Based on Mathematical Model of Number System," in *International Conference on Circuit, Power and Computing Technologies [ICCPCT]*, 2014.

[11] S. Al-Nofaie, A. Gutub and M. Al-Ghamdi, "Enhancing Arabic text steganography for personal usage utilizing pseudo-spaces," *J. King Saud Univ.,* june 2019, 1319-1578.

[12] X. Zhou, W. Gong, W. Fu and L. Jin, "An Improved Method for LSB Based Color Image steganography Combined with Cryptography," ICIS 2016, Japan, June, 2016.

[13] E. Emad, A. Safey, A. Refaat, Z. Osama, E. Sayed and E. Mohamed, "A secure image steganography algorithm based on least significant bit and integer wavelet transform," *J. Syst. Eng. Electronics*, June 2018, vol. 29, pp. 639 – 649.

[14] A. Taha, et al. "A high capacity algorithm for information hiding in Arabic text," *J. King Saud Univ. Comput. Inf. Sci.* Egypt, July 2018, pp. 1319-1578.

[15] L. Y. Por, K. Wong, and K. O. Chee, ``UniSpaCh: A text-based data hiding method using Unicode space characters," *J. Syst. Softw.*, vol. 85, no. 5, pp. 1075-1082, 2012.

[16] A. Naharuddin, A. D. Wibawa and S. Sumpeno "A High Capacity and Imperceptible Text Steganography Using Binary Digit Mapping on ASCII Characters," International Seminar on Intelligent Technology and Its Applications, Indonesia, ISITIA 2018, pp. 287-292.

[17] R. Kumar, S. Chand and S. Singh, "An efficient text steganography scheme using Unicode Space Characters*. Int. J. Comput. Sci.* 2015, vol. 10, 8–14.

[18] H. J. Shiu, B.S. Lin, P. Y. Huang, C. H. Huang, C. L. Lei, "Data Hiding on Social Media Communications Using Text Steganography," In Proceedings of the International Conference on Risks and Security of Internet and Systems, Dinard, France, 19–21 September 2017; pp. 217–224.

[19] 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.* 2017, 30, 236–248.

[20] M. Aman, A. Khan, B. Ahmad and S. Kouser, "A hybrid text steganography approach utilizing Unicode space characters and zero-width character," *Int. J. Info. Tech. Secur.*, vol. 9, no. 1, pp. 85-100, 2017.

[21] M. T. Ahvanooey, Q. Li, J. Hou, H. D. Mazraeh and J. Zhang, "AITSteg: An Innovative Text Steganography Technique for Hidden Transmission of Text Message via Social Media," *IEEE Access* 2018, 6, 65981–65995.

[22] M. T. Ahvanooey, Q. Li, H. J. Shim and Y. Huang, "A Comparative Analysis of Information Hiding Techniques for Copyright Protection of Text Documents," *Secur. Commun. Netw.*, 2018, 5325040.

[23] M. H. Alkawaz, G. Sulong, T. Saba, A. S. Almazyad and A. Rehman, "Concise analysis of current text automation and watermarking approaches," *Secur. Commun. Netw.,* 2016, 9, pp. 6365–6378.

[24] M. T. Ahvanooey, Q. Li, J. Hou, R. A. Rajput and Y. Chen "Modern Text Hiding, Text Steganalysis, and Applications: A Comparative Analysis," *Entropy* 2019, 21, 355, pp. 11-14.

[25] L. Y. Por, T. F Ang and B. Delina, "Whitesteg: A new scheme in information hiding using text steganography," *Wseas Trans. Comput.* 2008, vol. 7, no. 6, pp. 735–745.

[26] Y. W. Kim, K. A. Moon and I. S. Oh, "A text watermarking algorithm based on word classification and inter-word space statistics," In Proceedings of the Seventh International Conference on Document Analysis and Recognition, Washington, DC, USA, August 2003; vol. 2, p. 775.

[27] S. G. Rizzo, F. Bertini, D. Montesi, C. Stomeo, "Text Watermarking in Social Media," In *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining (ASONAM),* Jul. 2017, pp. 208-211.

[28] R. Din, A. Samsudin and P. A Lertkrai, "Framework Components for Natural Language Steganalysis," *Int. J. Comput. Eng.* 2012, pp. 641–645.

[29] W. Bender, D. Gruhl, N. Morimoto and A. Lu, "Techniques for data hiding," *IBM Syst. J.* 1996, 35, pp. 313–336.

[30] N. Naqvi, A. T. Abbasi, R. Hussain, M. A. Khan and B. Ahmad, "Multilayer Partially Homomorphic Encryption Text Steganography (MLPHE-TS): A Zero Steganography Approach," *Wirel. Pers. Commun.* 2018, vol. 103, pp. 1563–1585.

[31] Qianmu Li , Yanjun Song, Jing Zhang, Victor S. Sheng. Multiclass imbalanced learning with one-versus-one decomposition and spectral clustering. Expert Systems With Applications, 147 (2020) 113152. doi:10.1016/j.eswa.2019.113152. 2020

[32] Qianmu Li , J. Hou, S. Meng, H. Long (2020). GLIDE: A Game Theory and Data-Driven Mimicking Linkage Intrusion Detection for Edge Computing Networks, Complexity, vol. 2020, Article ID 7136160, 18 pages, 2020. doi: 10.1155/2020/7136160. 2020

[33] Li, Q., Yin, X., Meng, S. et al. A security event description of intelligent applications in edge-cloud environment. Journal of Cloud Computing. 9, 23 (2020). https://doi.org/10.1186/s13677-020-00171-0

[34] Qianmu Li , Y. Tian, Q. Wu, Q. Cao, H. Shen and H. Long, "A Cloud-Fog-Edge Closed-Loop Feedback Security Risk Prediction Method," IEEE Access, vol. 8(1), pp. 29004-29020, 2020.

[35] Qianmu Li, Shunmei Meng, Sainan Zhang, Ming Wu, Jing Zhang, Milad Taleby Ahvanooey and Muhammad Shamrooz Aslam. Safety Risk Monitoring of Cyber-Physical Power Systems Based on Ensemble Learning Algorithm. IEEE Access, Vol.7, pp. 24788–24805. 2019

[36] Qianmu Li, Shunmei Meng, Shuo Wang , Jing Zhang and Jun Hou (2019) CAD : Command-level Anomaly Detection for Vehicle-Road Collaborative Charging Network. IEEE Access, Vo.7, pp. 34910–34924.

[37] Qianmu Li, Shunmei Meng, Sainan Zhang, Jun Hou, Lianyong Qi (2019) Complex Attack Linkage Decision-Making in Edge Computing Networks. IEEE Access, Vo. 7, pp. 12058 – 12072.

[38] Qianmu Li, Yinhai Wang, Ziyuan Pu, Shuo Wang, Weibin Zhang (2019) A Time Series Association State Analysis Method in Smart Internet of Electric Vehicle Charging Network Attack. Transportation Research Record, vol. 2673, pp. 217-228.

[39] Qianmu Li , Yaozong Liu, Shunmei Meng, Hanrui Zhang, Haiyuan Shen and Huaqiu Long. A dynamic taint tracking optimized fuzz testing method based on multi-modal sensor data fusion. EURASIP Journal on Wireless Communications and Networking(2020) 2020:110. https://doi.org/10.1186/s13638-020-01734-0.

[40] Xiaokang Wang, Laurence T. Yang, Liwen Song, Huihui Wang, Lei Ren, and M. Jamal Deen, "A Tensor-based Multi-Attributes Visual Feature Recognition for Industrial Intelligence," IEEE Transactions on Industrial Informatics, DOI: 10.1109/TII.2020.2999901, 2020.

[41] Lei Ren, Zihao Meng, Xiaokang Wang, Renquan Luan, and Laurence T. Yang, "A Wide-Deep-Sequence Model based Quality Prediction Method in Industrial Process Analysis," IEEE Transactions on Neural Networks and Learning Systems, vol. 31, no. 9, pp. 3721-3731, DOI: 10.1109/TNNLS.2020.3001602, Sep. 2020.

[42] Lei Ren, Zihao Meng, Xiaokang Wang, Lin Zhang and Laurence T. Yang, "A Data-driven Approach of Product Quality Prediction for Complex Production Systems," IEEE Transactions on Industrial Informatics, DOI: 10.1109/TII.2020.3001054, 2020.