IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 15, 2019, accepted October 6, 2019, date of publication October 10, 2019, date of current version October 22, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2946299

An Electronic Lattice Construction Method Based on Game Padding

CONGFEI SUN AND KAIXI WANG, (Member, IEEE)
College of Computer Science and Technology, Qingdao University, Qingdao 266071, China

Corresponding author: Kaixi Wang (kxwang@qdu.edu.cn)

This work was supported in part by the NSFC-General Technical Research Foundation Joint Fund of China under Grant U1536113, and in part by the CERNET Innovation Project under Grant NGII20180405.

ABSTRACT The concealment, the steganography success rate, and the steganographic capacity are key performance indicators to text steganography. The existing text steganography methods still have a low steganographic capacity, some syntactic or semantic ambiguity problems to some extent. We propose an electronic lattice construction method based on a padding game to hide a secret message. This method first computes the length of a secret message and a key is worked out based on the length; the walking rule, the constraints to a hidden path, the initial size of a grid and the starting cell are determined based on the key and a unique hidden path can be figured out for the given secret; then the secret message is stored in the cells along the unique path and all the remaining cells in the grid are filled according to the padding game rules, which helps to obscure that secret message; finally, elements in some cells are removed to get the crossword that will be shared with the receiver. To extract the message, the receiver fills the crossword according to the same game rules as the sender does, and calculates that unique hidden path based on the shared key. The experimental results and analyses show that its concealment is prominent, its steganography success rate is 100%, and it has a high steganographic capacity. It also has no readability issue as the steganography by generation usually does.

INDEX TERMS Information hiding, coverless text steganography, padding game, electronic lattice, hidden paths.

I. INTRODUCTION

With the rapid development of Internet technology, the dissemination of digital audio-visual products and other electronic publications become more and more convenient. Just as a coin has two sides, besides such facilitation, the new technologies also bring a lot of problems, such as information theft and disclosure, and other serious infringements. It has become a problem of great concern on how to make people not only enjoy the convenience but also effectively protect information security and digital rights. Information hiding technology [1] is one of potential approaches to these issues, one branch of which, i.e. steganography, is the practice of communicating a secret message by hiding it in a popular cover object. This approach can avoid the secret transmission to be noticed by an attacker and is attracting more and more attention. It plays an important role in protecting the information from uncovering, such as the secret communication.

Based on the cover types, steganography can be divided into five categories: image steganography [2], [3], video steganography [4], [5], audio steganography [6]–[8], text steganography [9] and network steganography [10]. As the most important and predominant media, a text is widely used in our daily life and work. This makes it difficult to arouse suspicion when a text is used as a cover, which conforms to the principal characteristics of a cover [11]. The text steganography was invented in ancient times. The most typical ones were the Tibetan head poems and the Tibetan tail poems [12]. Nowadays, texts are still the first media on the Internet, used in chats, emails, news, etc. Therefore the text steganography still has good application scenarios. However, due to the relative lack of redundancy in a text [13], [14], text steganography is difficult, its steganography capacity is low and has poor concealment to this day. Thus it still is a research difficulty and a hotspot at present.

The associate editor coordinating the review of this manuscript and approving it for publication was Xiaochun Cheng.

This paper proposes an electronic lattice construction method to implement text steganography based on a padding game. Different from the traditional Cardano lattice, this method designs a novel electronic lattice with a key. The secret is embedded in a unique hidden path, namely which is also called the lattice, and the most-filled grid is obtained by filling in the blank cells with elements according to the padding game rules. Then the partial cells are cleared to form a crossword. After getting the crossword, the receiver fills it to get a most-filled grid according to the same game rules as the sender does, and the unique hidden path is figured out based on the key shared with the sender, and then the secret message will be extracted.

The rest of the paper is organized as follows. Section II introduces the current existing steganography methods in three categories. After the notations are prescribed, Section III details the embedding process and the extraction process and demonstrates them. The performance indicators are analyzed in detail in Section IV, including the success rate of steganography, the hiding capacity, the concealment and security. Finally, the conclusion is drawn in Section V.

II. RELATED WORKS

From the perspective of the embedding manipulation, text steganography can be divided into three categories: steganography by cover modification, steganography by cover selection/search, and steganography by cover synthesis [14]. And text steganography by cover modification can be further subdivided into three types: steganography by format-based modification, syntactic modification, and semantic modification. All these classed are summarized and analyzed in the following.

Text steganography by format modification mainly operated on paragraph formats, character spelling, font characteristics, invisible characters, and file formats. The paragraph formats are modified [18], such as slightly by adjusting the line space or word space, to represent a binary digit bit. The character spelling or its font characteristics [15], [19], [20], such as the font type, size, and/or color, can be employed to denote a digital bit. The invisible characters can also be used for steganography by inserting spaces in lines or between paragraphs [21]. Steganography by modification based on file types completely depends on the specific features of file types [22], [23], such as case sensitiveness in a file.

Text steganography by syntactic modification [24] is to change the syntax of a sentence, including performing the active and passive transformation, moving the position of adjuncts, whether using an anticipatory subject or not, or adjusting the order of parallel structures in a sentence, while ensuring that its meaning is not changed.

Text steganography by semantic modification is mainly based on synonym substitution [25] and grammatical rules of functional words [28], [29]. Steganography by synonym substitution still might easily cause ambiguity sometimes although many new technologies have improved it, which will arouse the suspicion of attackers and the low concealment.

Text steganography by selection/search is a new approach to steganography, which takes full use of texts on the Internet as covers. Shi *et al.* [26] found suitable covers and the positions where the secret message is located. Chen *et al.* [27] directly generated a stego-vector from the hidden information at first, and then based on text big data, a normal text that includes the stego-vector will be retrieved. Wang and Gao [30] proposed a method based on the parity of the stroke number of Chinese characters. This method constructed a binary search tree based on texts from the Internet, and searched the texts corresponding to the secret based on the built search tree. Long *et al.* [39] proposed a coverless steganography by retrieving massive web texts on the Internet.

Text steganography by synthesis is to create a cover to convey the desired message. For example, both the NICETEXT algorithm [31] and TEXTO algorithm [32] employed sentence or text templates, and the secret is hidden in the filling words. The template for NICETEXT is more elaborate than for the TEXTO, so the generated text by NICETEXT looks more natural. Ren *et al.* [33] generated short texts and used the statistical features of these short texts to hide a secret. Luo and Huang [34] used a recurrent neural network to generate poetry. According to the rhythm and other characteristics of Ci-Poetry, a secret message is hidden in the generated new Ci-Poetry [35]. Yang *et al.* [36] proposed a linguistic steganography based on recurrent neural networks, which can automatically generate high-quality text covers on the basis of a secret bitstream. Different from the above techniques, our proposed methods will generate a crossword to implement steganography with good concealment and no ambiguity.

In summary, a secret message may be lost when the stego-text is re-typed in traditional format-based text steganography and its robustness is poor. Steganography by syntactic and synonym substitution may lead to bad readability and semantic ambiguity, which leads to poor concealment. Steganography by search might fail to implement steganography or its computation complexity might be tremendous. With the development of the natural language processing and deep learning technologies, more attentions are paid on steganography by synthesis, and the steganography by synthesis has been becoming a research hotspot. Our method, generating a crossword as a cover based on a padding game, will be discussed in detail in the following.

III. STEGANOGRAPHY ALGORITHM BASED ON AN ELECTRONIC LATTICE

There are many types of crossword puzzles, such as Crossword themes, Cryptic crosswords, Cipher crosswords, Diagramless crosswords, Fill-in crosswords and Acrostic puzzles. Each has its own clues to help solve it in a regular crossword game, such as shaded squares or bold bars. In the paper, a crossword is constructed for steganography based on the following conventions, and the embedding and extracting

processes. Its main idea is that a specific unique path can be prescribed in a crossword, and a secret is filled in the cells along that path, which is called a hidden path or a lattice.

A. NOTATIONS

- L_{msg}: the length of a secret message.
- P: a group of starting cells of a path. $P = \{p_i =< h, v > \mid p_i$ is a cell who at least has a L_{msg}-length path && h is the grid abscissa and v is the grid ordinate$\}$.
- R: a walking rule set. A walking rule defines how to locate the next cells. Once the starting cell p_i is determined, more than one hidden path will be identified based on a given walking rule. For example, the rules for performing a knight or elephant move in the Chinese Chess game can be defined as a walking rule; the clues to some crosswords might be regarded as a walking rule; an arbitrary curve can also be treated as a walking rule even if they cannot be described by a function; even more, a broken line can be treated as a walking rule though it is enumerated in a sequence. Obviously, for any rule, the next cell should be always never traversed before, that's is to say, there is no duplicate cell in a path. All these walking rules are denoted as $r_0, r_1, \ldots, r_{max_1}$, respectively, and construct a walking rule set $R = \{r_0, r_1, \ldots, r_{max_1}\}$.
- S: a constraint set. Just as mentioned above, there might be more than one hidden path to a given end cell when P, R is determined, so some further constraints are needed to figure out a unique hidden path. E.g., if the lexicographic order of the position's representation on the axis can be employed, the different orders are denoted as $s_{10}, s_{11}, \ldots, s_{1max_2}$ respectively, and all these constraints can be denoted as $S_1 = \{s_{10}, s_{11}, \ldots, s_{1max_2}\}$. Also a shortest, longest or other path between two cells can be selected, which are indicated by $s_{20}, s_{21}, \ldots, s_{2max_3}$, respectively, and all these constraints can be denoted as $S_2 = \{s_{20}, s_{21}, \ldots, s_{2max_3}\}$, so the whole constraint set can be denoted as: $S = S_1 \mathrm{x} S_2 = \{<s_{10}, s_{20}>, <s_{10}, s_{21}>, <s_{11}, s_{20}>, <s_{11}, s_{21}>, \ldots, <s_{1max_2}, s_{2max_3}>\}$.
- $PATH$: a list of hidden paths for a given secret. There are several hidden paths in the grid that can hide a L_{msg}-length message. The starting cells or the ending cells of these hidden paths may be different; moreover, for a given starting cell and an end cell, there might be more than one hidden path between them. Given P, R and S, if L_{msg} is calculated out, a unique hidden path can be determined. Therefore, the combination of P, R and S can define a list of hidden paths for a L_{msg}-length secret, which is denoted as: $PATH =<< p_i, r_j, s_k > \mid p_i \in P, r_j \in R, s_k \in S >$, where every element, i.e. $< p_i, r_j, s_k >$ is a unique hidden path for a L_{msg}-length secret.
- M: the number of elements in the $PATH$ set.

Rule 1: The mapping rules between a key and a unique hidden path in $PATH$. Many mapping functions can be employed to establish the mapping. Their general form can be expressed as follows:

$$\begin{cases} mapf_1 : key \sim n, n \in N, \\ mapf_2 : key \sim L_{msg}, \\ HiddenPath(key) = n \bmod M. \end{cases} \tag{1}$$

where N is integer.

To be convenient for illustration in this paper, the length of a secret message, i.e. L_{msg}, is taken as the key, i.e. $key = L_{msg}$. So the above mapping functions can be simplified as follows:

$$HiddenPath(L_{Msg}) = L_{msg} \bmod M. \tag{2}$$

where the hidden paths in the $PATH$ list are numbered and started from 0.

Rule 2: The padding rules of auxiliary information. To cloak the communication, the grid will be filled with some unrelated auxiliary information in the remaining cells according to the rules of a padding game. The auxiliary information might be letters, words, or numbers, etc, which depends on what language a secret message is in and what game is employed. One example is to fill idioms from Xinhua Idiom Dictionary (June 2009, Business Press Publication) in Chinese, or to fill words from The Oxford English Dictionary (The second edition) in English. If no information can be filled in a cell, just left it a blank. Thus, a most-filled grid is obtained when no more space can be padded.

Rule 3: The deletion rules in a most-filled grid. To get a crossword, some cells will be cleared according to the game rules. And specifically, a cell can be cleared in the most-filled grid when the following rules hold true:

1) If the content in the cell is a part of the secret message and it is the very element when the crossword is filled, the cell can be cleared; otherwise, it should be kept untouched. In other words, no other content except the deleted can be filled back.

2) If the content in the cell is not a part of the secret message, just delete it.

There are many elimination algorithms to remove letters from a most-filled grid. An example is given as follows.

(S1) Let the total number of to-be-deleted letters be denoted as T, which is an indicator of the game level. The larger is T, the more difficult is the game;

(S2) For each word in the most-filled grid, do:

Randomly delete one letter in it as long as the letter is not the right letter in the secret message.

The number of deletions is recorded as t_1.

(S3) If $t_1 < T$

For every letter that is from a secret message, do:

If the letter is the only one to be filled, it can be eliminated and do $t_1 + +$;

Else the letter is kept;

(S4) If $t_1 < T$

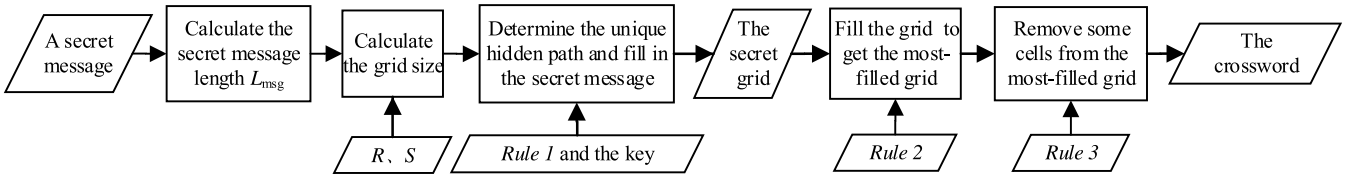


FIGURE 1. The embedding process.

For every letter not contained in the secret message and the letters in the same row and column as it are not a part of the secret message,
Delete it and do t_1++;

B. THE PROPOSED STEGANOGRAPHY METHOD

Based on the above notations and rules, this section will describe the processes of embedding and extracting in detail.

When embedding a secret message, the size of a grid must be firstly determined according to its length, i.e. L_{msg}. There might be several hidden paths that can hide the message in this grid. The unique hidden path is determined according to the key shared by the sender and receiver. The secret message is embedded in the cells along the unique hidden path. Then some auxiliary information is filled to construct a most-filled grid. Some cells are cleared in the most-filled grid to get a crossword. Generally, the extracting process is the reverse of the embedding process. The receiver fills the crossword according to the same game rules as the sender does, and calculate the lattice, i.e. the hidden path according to the key, then the secret message can be excerpted along the hidden path. The detail process is introduced as follows.

1) THE EMBEDDING PROCESS

In the embedding process, the important steps are to determine the size of the grid, to calculate the lattice and to fill the auxiliary information, as shown in Fig. 1. The whole procedure is described as follows:

1) Calculate the length L_{msg} of the secret message;

2) Determine the R, S;

3) Calculate the grid size. In order to hide a secret, the size of the grid should be no less than some size, and it should be determined before the lattice is constructed. The specific steps are as follows:

First, calculate the size of the initial grid $m * n$ under the following conditions:

(1) $m * n >= L_{msg}$;

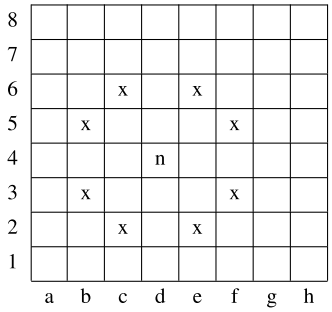
(2) the difference between m and n is the minimum;

Second, select any two cells as the starting cell and the ending cell respectively, then calculate the length L_{path} of the path between the two cells for every $(r, s) \in (R, S)$.

For $L_{path} < L_{msg}$, if $m > n$, let $n++$, otherwise do $m++$.

Repeat the above operations for computing m and n, until $L_{path} \geq L_{msg}$ holds true, and store the current grid size in M_{min} and N_{min}. Meanwhile, the start cell is also obtained, which indicates a hidden path that can hide a L_{msg}-length message.

TABLE 1. The walking rules.



4) Determine the unique hidden L_{path} lattice according to the key and *Rule* 1;

5) Put the secret message into the cells along on the unique hidden path;

6) Fill the auxiliary information according to *Rule* 2 and get the most-filled grid;

7) Clear some cells in the most-filled grid according to *Rule* 3 and get the crossword.

2) THE EXTRACTING PROCESS

The extracting process is illustrated in Fig. 2 and the detail is described as follows:

1) Fill the crossword according to *Rule* 2 to get the most-filled grid.

2) Construct the same mapping between a key and a unique hidden path as the sender does according to *Rule* 1.

3) Calculate the lattice according to the key and *Rule* 1.

4) Extract the secret message along the lattice.

C. DEMONSTRATION

In the following illustration, the string "network" is regarded as a secret message and its length L_{msg} is taken as the key to simplify the illustration. Other instances are given as follows:

- R: The knight move in the Chess [37] is defined as the walking rule, i.e. $R = \{r_0\}$. It is illustrated in an 8*8 grid as shown in Table 1. For the cell $(d, 4)$, one of eight cells around it can be reached in only one step, while no other cells can be touched directly. The eight cells are marked with 'x', namely, $f5$, $e6$, $c6$, $b5$, $b3$, $c2$, $e2$ and $f3$.
- S_1: Different constraints can be designed further to get a unique path. Only two constraints are listed for demonstration here.
  1) The cell is preferred if its label appears first in the alphabetical order among the optional cells, and if two

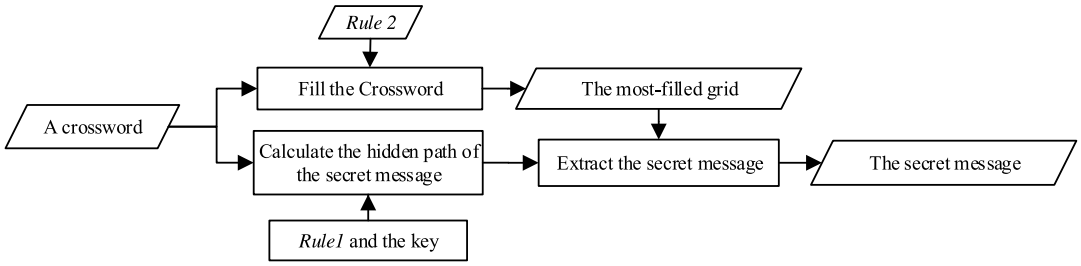


FIGURE 2. The extracting process.

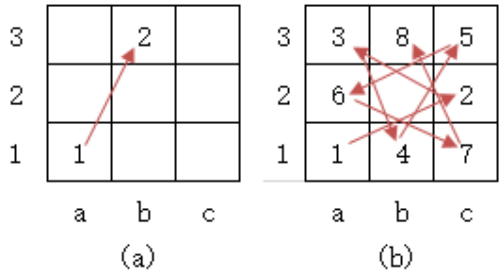


FIGURE 3. An example of the shortest path and the longest path.

cells are labelled with the same letter, the cell labelled with the smaller number is selected. The constraint is denoted by s_{10}.

2) The cell is the first choice if its label appears first in the reverse lexicographic order among the optional cells, and if two cells are labelled with the same letter, the cell labelled with the smaller number is selected. The constraint is denoted by s_{11}.

Then a simple set of constraints is $S_1 = \{s_{10}, s_{11}\}$;

- S_2: For any two cells, the shortest path constraint between them is denoted by s_{20}; the longest path constraint between them is denoted by s_{21}. For instance, the shortest path and the longest path from $a1$ to $b3$ are illustrated in Fig. 3 respectively. Just as prescribed in R, neither the shortest path nor the longest path has any repeated cell and the integer in the cell indicates the order along the path.

  Then a simple set of constraints is $S_2 = \{s_{20}, s_{21}\}$;
- $S = S_1 \times S_2 = \{<s_{10},s_{20}>,<s_{10},s_{21}>,<s_{11},s_{20}>,<s_{11},s_{21}>\}$.
- P: According to Section 3.2.1, the grid size is initialized to 3*3, and the minimum size is finally assigned to $M_{min} = 8, N_{min} = 8$ by calculation in case of $L_{msg} = 7$. Herein, the starting cell set is, $P = \{p_0 =< h, 1 >, p_1 =< h, 8 >, p_2 =< a, 1 >, p_3 =< a, 8 >\}$, which is shown in Table 2.
- $PATH$: Once P, R and S is given as above, for the secret, i.e. ''network'', all the hidden paths are determined as follows:

$PATH = \{<p_0, r_0,<s_{10}, s_{20}> >, <p_0,r_0,<s_{11},s_{20}> >,$
$<p_1,r_0,<s_{10}, s_{20}> >, <p_1,r_0,<s_{11},s_{20}> >,$
$<p_2,r_0,<s_{11}, s_{20}> >, <p_3,r_0,<s_{10},s_{20}> >,$
$<p_3,r_0,<s_{11}, s_{20}> >, <p_2,r_0,<s_{10},s_{20}> >,$
$<p_0,r_0,<s_{10}, s_{21}> >, <p_0,r_0,<s_{11},s_{21}> >,$
$<p_1,r_0,<s_{10}, s_{21}> >, <p_1,r_0,<s_{11},s_{21}> >,$
$<p_2,r_0,<s_{11}, s_{21}> >, <p_3,r_0,<s_{10},s_{21}> >,$
$<p_3,r_0,<s_{11}, s_{21}> >, <p_2,r_0,<s_{10},s_{21}> > \}$.

- $M = 16$.

TABLE 2. The starting cell.

8	a8							h8
7								
6								
5								
4								
3								
2								
1	a1							h1
	a	b	c	d	e	f	g	h

1) THE EXAMPLE OF EMBEDDING PROCESS

Based on the above instantiations, the detail embedding operations are introduced as follows:

(SE-1) Calculate the length of the secret message $L_{msg} = 7$, which is taken as the key only for illustration;

(SE-2) The mapping between a key and a unique hidden path simply employs the formula Eq. (2);

(SE-3) The minimal grid size is $M_{min} = 8$, $N_{min} = 8$ according to Step 3 in the embedding process in Section III.B;

(SE-4) The unique hidden path is obtained according to the mapping $Rule$ 1 and $key = L_{msg} = 7$:

$HiddenPath(L_{Msg}) = 7 \ mod \ 16 = 7$,

which corresponds to the element $<p_2,r_0,<s_{10},s_{20}> >$ in the above instantiation of $PATH$.

In this unique hidden path, the starting cell is $p_2 =< a, 1 >$, the walking rule is r_0, the constraint is $<s_{10},s_{20}>$. Thus, the unique hidden path is: $a1$->$b3$->$a5$->$b7$->$d6$->$f7$->$h8$, and its corresponding cells $a1, b3, a5, b7, d6, f7, h8$ will store 'n', 'e', 't', 'w', 'o', 'r', 'k' respectively. So far, the lattice is constructed successfully;

(SE-5) The remaining cells are filled in auxiliary information according to the $Rule$ 2. Herein, the English word padding game is employed. First, select a non-blank cell and find out how many blank cells, indicated as l_{word}, in the same row or column, then search a l_{word}-length word with the same letter as in the non-blank cell in the same position in 'The 2nd Oxford English Dictionary', if found, fill the blank cells with the word and select another non-blank cell and repeat the above padding; otherwise, find a short word to fill, or if

no more letter can be filled in a blank cell, just left it a blank. Finally, a most-filled grid is obtained as shown in Fig. 4 when there is no more blank cell can be filled or all the cells are filled;

e	a	r	n	w	a	1	k
y	w	a	t	e	r	u	n
e	e	1	o	X	e	n	i
t	x	i	c	e	s	c	f
h	p	e	a	r	p	h	e
g	e	n	e	r	o	u	s
i	r	X	a	X	n	g	a
n	t	i	r	e	d	e	d

FIGURE 4. The most-filled grid.

e	a		n		a	1	k
y	w		t	e			n
	e	1	o	X	e	n	
t	x	i		e	s	c	f
		e	a		p		e
g	e	n	e	r		u	s
i	r	X	a	X	n		a
n	t	i		e	d	e	

FIGURE 5. The crossword.

(SE-6) Erase some cells in the most-filled grid according to *Rule* 3 and get the crossword as shown in Fig. 5.

In this example, let $T = 16$ (T could be a different value, generally depends on the game difficulty.) and perform the elimination algorithm given in *Rule* 3 to get the crossword.

2) THE EXAMPLE OF EXTRACTING PROCESS

The extracting process is described in the following:

(SX-1) The receiver fills the crossword just as the sender does according to *Rule* 2 and acquires the most-filled grid, which might be different from Fig. 4. For example, Fig. 6 shows another case, in which the difference from Fig. 4 is marked in red;

(SX-2) The unique hidden path is computed according to the key, i.e. L_{msg} and the formula Eq. (2):

$HiddenPath(L_{Msg}) = 7 \ mod \ 16 = 7$.

Thus, $<p_2,r_0,<s_{10},s_{20}>>$ is selected as the hidden path;

(SX-3) The secret message is extracted from the cells along the unique hidden path $<p_2,r_0,<s_{10},s_{20}>>$.

IV. THE ANALYSES

The proposed method employs the crossword game to convey a secret and it is not subject to any modification or revision attack. Therefore, its robustness is better than steganography by modification. Next, its success rate of steganography, the hiding capacity (HC), the embedding rate (ER), the concealment and the security comparison are analyzed in detail.

e	a	r	n	t	a	1	k
y	w	a	t	e	r	u	n
e	e	1	o	X	e	n	i
t	x	i	c	e	s	c	f
h	p	e	a	r	p	h	e
g	e	n	e	r	o	u	s
i	r	X	a	X	n	g	a
n	t	i	r	e	d	e	y

FIGURE 6. Another different most-filled grid.

TABLE 3. The success rate of steganography.

Steganography Type	Modification			Non-modification			
	[21]	[24]	[25]	[16]	[26]	[33]	Our method
The success rate	100	100	100	100	<=100	100	100

A. THE SUCCESS RATE OF STEGANOGRAPHY

The success rate of steganography is an important indicator to the coverless text steganography. In this paper, it is defined as Eq. (3). This method is applicable to both Chinese and English. For Chinese, every character in the secret has been put into the grid, and even if there doesn't exist an idiom including that character not, a most-filled grid will be got because a blank cell is allowed. For English, any secret message can be expressed by 26 letters in the English alphabet, and every letter comprising the secret is stored in the grid and a most-filled grid can also be obtained because a cell can be empty. Therefore, our success rate of steganography is 100%.

$$SR = \frac{The\ number\ of\ success\ steganography}{The\ Total\ of\ steganography}. \quad (3)$$

B. THE HIDING CAPACITY

The HC is an also important performance indicator to steganography, which indicates how many characters/letters/bits can be hidden in a cover. It is usually represented by the ER that is a ratio of a hidden unit to the corresponding cover units. Two comparisons are carried out in different ways as follows.

- The HC comparison with other steganographic methods are performed in two different perspectives.

(1) The hiding capacity is calculated per block or cell. The analysis of hiding capacity is shown in Table 4. Obviously, our method has a higher HC per cell than [16] and [17],

TABLE 4. The hiding capacity per block/cell.

Method		HC(bit)
Literature [16]	Beginners	1.96
	Intermediate	1.89
	Expert	2.17
Literature [17]		0-3
Our method		8 or 16

TABLE 5. The hiding capacity per an article/a game.

Method	HC(bit)
Literature [39]	10.53
Literature [40]	9.04
Our method	$\geq$(8 or 16)

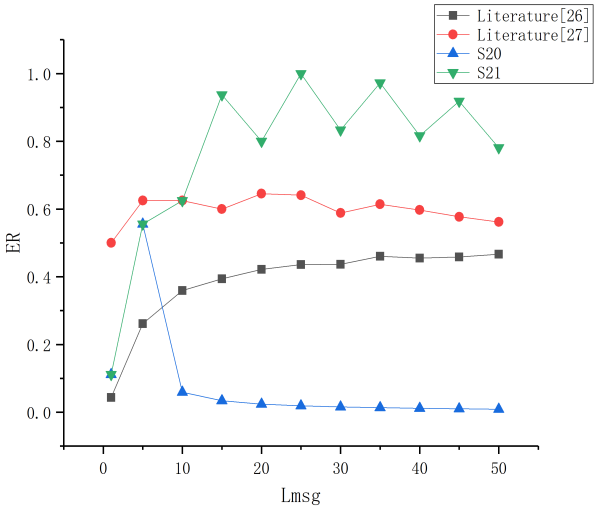


FIGURE 7. The embedding rate.

the key reason is that the character/letter is hidden directly, not per bit to be hidden. For a double-byte character, such as Chinese characters, it is 16, and for a English letter, it is 8.

(2) The hiding capacity is calculated per transmission, usually presented as per an article or a game. The comparison is shown in Table 5. For our proposed method, it can hide at least 8 bits per cell when the secret is in English and it employs a grid with at least 3*3 size, so the shorted hidden path can contain a cell. Therefore, the minimum capacity is 8 bits. Similarly, the minimum capacity is 16 bits for the language represented in double bytes, such as Chinese. For [39], [40], their average HCs are listed in Table 5. Thus, our method has better HC performance than other two methods because a shortest hidden path usually contain more than one cell.

- The ER comparison to other coverless text steganographic methods.

The ER is calculated as follows:

$$ER = \frac{The\ Length\ of\ secret\ message}{M_{min} * N_{min}} \tag{4}$$

The different path selections based on S_2 are compared with the literature [26] and [27] in the embedding rate. The comparative results are shown in Fig. 7. When the length of secret messages increases is more than 10, the method with the constraint s_{21} is better than other methods.

C. THE CONCEALMENT

We analyze the concealment from two perspectives: the perceptibility and the probability of successfully extracting the exact secret.

1) THE PERCEPTIBILITY

We are used to reading a phrase or a sentence whose words are in a line. But if the generated crossword is put into a Cartesian coordinate system and its down-left corner is attached to the ordinate origin, every word in a phrase or sentence will have a corresponding point and can be denoted as a pair of coordinates, e.g. (x_i, y_i) for the ith element. A straight line can be computed according to all these points via the least square method. Thus, the perceptibility can be measured according to these points' distribution in the crossword, i.e, the standard deviation of the distances of all these points to the straight line. Larger is the standard deviation, the stronger is the unperceptibility, i.e when the standard deviation is 0, the unperceptibility is the worst.

For example, in Fig. 5, the secret "network" occupies the following cells:

$$\begin{aligned} &m_1 = (x_1, y_1) = (1, 1); \quad m_2 = (x_2, y_2) = (2, 3);\\ &m_3 = (x_3, y_3) = (1, 5); \quad m_4 = (x_4,\ y_4) = (2, 7);\\ &m_5 = (x_5, y_5) = (4, 6); \quad m_6 = (x_6,\ y_6) = (6, 7);\\ &m_7 = (x_7, y_7) = (8, 8). \end{aligned}$$

We can assume that the following straight line is the best to fit all these points.

$$y = bx + a. \tag{5}$$

It can be solved as follows:

$$\begin{cases} \overline{x} = \dfrac{x_1 + x_2 + \ldots + x_n}{n}, \\ \overline{y} = \dfrac{y_1 + y_2 + \ldots + y_n}{n}, \\ \overline{xy} = \dfrac{x_1 * y_1 + x_2 * y_2 + \ldots + x_n * y_n}{n}, \\ \overline{x}\,\overline{y} = (\dfrac{x_1 + x_2 + \ldots + x_n}{n}) * (\dfrac{y_1 + y_2 + \ldots + y_n}{n}), \\ (\overline{x})^2 = (\dfrac{x_1 + x_2 + \ldots + x_n}{n}) * (\dfrac{x_1 + x_2 + \ldots + x_n}{n}), \\ \overline{x^2} = \dfrac{x_1^2 + x_2^2 + \ldots + x_n^2}{n}, \\ b = \dfrac{\sum_{i=1}^{n} x_i y_i - n\overline{x}\,\overline{y}}{\sum_{i=1}^{n} x_i^2 - n(\overline{x})^2} = \dfrac{\overline{x}\,\overline{y} - \overline{xy}}{\overline{x^2} - (\overline{x})^2}, \\ a = \overline{y} - b\overline{x}. \end{cases} \tag{6}$$

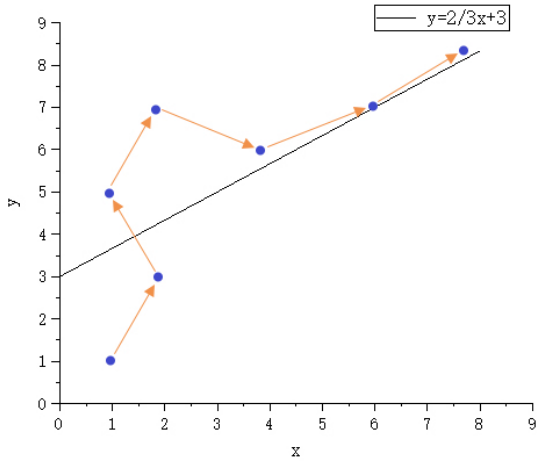


FIGURE 8. The secret points and its corresponding straight line.

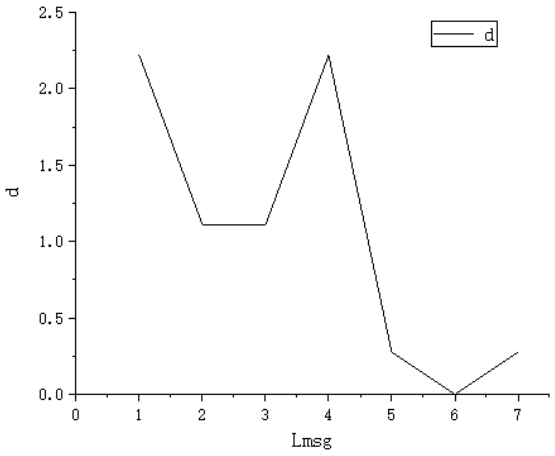


FIGURE 9. The distances of points to the line.

Therefore, for the cells $m_1 \sim m_7$, the following straight line will be acquired according to Eq. (6):

$$y = \frac{2}{3}x + 3. \tag{7}$$

The secret points and their corresponding line are shown in Fig. 8.

$$\begin{cases} d_i = |\dfrac{\frac{2}{3}x_i - y_i + 3}{\sqrt[2]{(A^2 + B^2)}}|, \\ M = \dfrac{1}{n}\sum\limits_{i=1}^{n} d_i, \\ S^2 = \dfrac{1}{n}\sum\limits_{i=1}^{n}(d_i - M)^2, \\ S = \sqrt{S^2}. \end{cases} \tag{8}$$

The standard deviation S is calculated according to Eq. (8) and it is approximately 0.849. The larger is the discreteness of these points that correspond the secret, the more difficult can the points form up in a line word, and the points fluctuate more greatly along a straight line, which will enhance the imperceptibility. The distance of points to the line is shown in Fig. 9.

2) THE PROBABILITY OF SUCCESSFULLY EXTRACTING

Even if a secret is perceived in a crossword, it is also very difficult to successfully extract the exact secret, which is analyzed in the following three cases. In such case, the concealment is mainly determined by the L_{msg} and the walking rule R. Other factors are directly or indirectly affected by L_{msg} and R, but aren't discussed here. The similar ciphertext attack [38] in cryptanalysis is used to analyze this method. In addition, the following assumption is further made that the attacker also knows any one of the two main factors, i.e. L_{msg} and R. In fact, the concealment is worse in this case than the actual situation. Following is the analysis.

First, we suppose that the attacker only knows L_{msg} and stego-text, i.e. the crossword. In this case, P_1, the probability of a successful extraction, is computed as follows:

$$\begin{aligned} P_1 &= \frac{1}{M_{min} * N_{min}} * \frac{1}{M_{min} * N_{min} - 1} * \ldots * \frac{1}{2} * 1 \\ &= \prod_{i=1}^{M_{min} * N_{min}} \frac{1}{i}. \end{aligned} \tag{9}$$

where $M_{min} * N_{min}$ is the size of the square crossword.

Second, we suppose that the attacker only knows $R = (r_1, r_2, \ldots, r_{max_1})$ and the stego-text. In this case, P_2, the probability of a successful extraction, is calculated as follows:

$$P_2 = \frac{1}{r_{max_1}} * \frac{1}{M_{min} * N_{min}} * \frac{1}{\sum_{k=1}^{M_{min} * N_{min}} K_k}. \tag{10}$$

where K_k is the number of all paths that satisfy R and its length is k.

For the above two cases, the concealment can be calculated as:

$$\begin{cases} C_1(msg) = 1 - P_1, \\ C_2(msg) = 1 - P_2, \\ C(msg) = 1 - P_1 - P_2. \end{cases} \tag{11}$$

where $C_1(msg), C_2(msg), C(msg)$ indicate the concealment respectively when L_{msg}, R or both of them are known. Obviously, the nearlier $C_1(msg), C_2(msg), C(msg)$ reach 100%, the better is the concealment.

As shown in Fig. 10, we can draw the conclusion that the concealment is nearly 100% when L_{msg} is greater than 5. When the same length secret is hidden, the concealment with s_{20} constraint is better than that with the constraint s_{21}.

D. THE SECURITY COMPARISON

In this paper, the security means the ability of anti-attack, includes the following: the anti-attack on the stego-text, such as format change, file format change, etc. In some cases, this is also called the robustness; the semantics analysis, such as whether it be ambiguity, and the syntactic analysis, such as

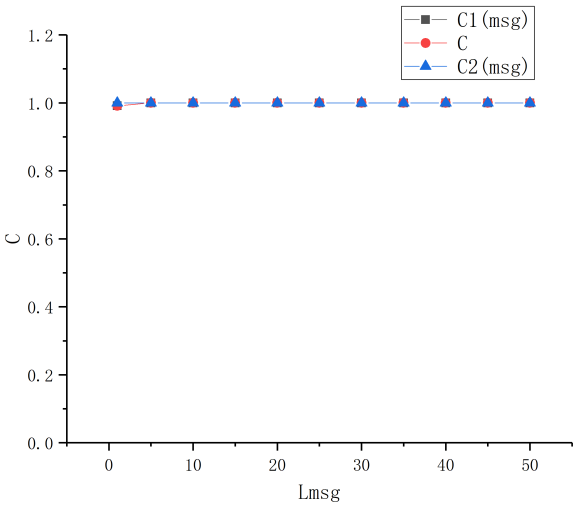


FIGURE 10. The concealment.

TABLE 6. The security comparison.

Steganography Type		Robustness	Syntactic	Semantic
Modification	[21]	N	Y	Y
	[24]	Y	N	Y
	[25]	Y	Y	N
Unmodification	[16]	Y	Y	Y
	[26]	Y	Y	Y
	[34]	Y	N	N
	Our method	Y	Y	Y

whether the syntax is smooth. Herein, if the secret message is lost or can be detected during the transmission, this method is not safe, which is indicated by 'N', otherwise by 'Y'.

The modification in [24] will cause secret messages to be lost, but not affect the original syntax and semantics, therefore its robustness is 'N', its syntactic and its semantic are identified as 'Y'. The method in [24] will lead to the syntax to be influent, so the syntactic is regarded as 'N'. Reference [25] brings semantic ambiguity, so the semantic is evaluated as 'N'. The method proposed in [34] is a kind of steganography by cover synthesis. The generated cover has problems in both syntax and semantics, so the syntax and semantics are estimated to be 'N'. Table 6 shows the results of the comparison to other steganographic methods.

Attackers usually analyze the text format, syntactic relation, semantic meaning, and statistical feature. Our method is to generate a crossword, and not to modify any text format. Therefore, there are no problems, such as influent syntax, semantic ambiguity and uneven statistical distribution feature.

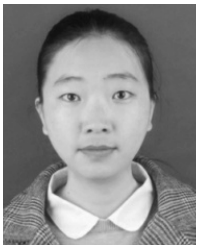
V. CONCLUSION

This paper proposes a novel method of coverless text steganography based on a generated crossword. It directly embeds a secret into a crossword. The sender and the receiver follow the same crossword game rules, a secret will be transferred based on a key. The crossword generated is not subject to the format modification and it also has no semantic or syntactic ambiguity problem. The method not only ensures the success of embedding but also strengthens the concealment, and improves the embedding rate when a suitable constraint is defined. Besides, the method is applied to other languages and we can also set different constraints to construct the lattice to meet the different demands of the hidden capacity and the concealment.

REFERENCES

[1] L. Zhenhua and Y. Ping, *Information Hiding Technology and its Application*. Wollerau, Switzerland: SciPress (in Chinese), 2002.
[2] B. Feng, W. Lu, and W. Sun, "Secure binary image steganography based on minimizing the distortion on the texture," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 2, pp. 243–255, Feb. 2015.
[3] X. Zhang, F. Peng, and M. Long, "Robust coverless image steganography based on DCT and LDA topic classification," *IEEE Trans. Multimedia*, vol. 20, no. 12, pp. 3223–3238, Dec. 2018.
[4] R. J. Mstafa and K. M. Elleithy, "Compressed and raw video steganography techniques: A comprehensive survey and analysis," *Multimed. Tools Appl.*, vol. 76, no. 20, pp. 21749–21786, Oct. 2017.
[5] T. Rabie and M. Baziyad, "The Pixogram: Addressing high payload demands for video steganography," *IEEE Access*, vol. 7, pp. 21948–21962, 2019.
[6] H. K. Qattous, "Hiding encrypted data into audio file," *Int. J. Comput. Sci. Netw. Secur.*, vol. 17, no. 6, pp. 162–170, Jun. 2017.
[7] Z. Wei, B. Zhao, L. Bo, J. Su, L. Xu, and E. Xu, "A novel steganography approach for voice over IP," *J. Ambient Intell. Humanized Comput.*, vol. 5, no. 4, pp. 601–610, Aug. 2014.
[8] T. Painter and A. Spanias, "Perceptual coding of digital audio," *Proc. IEEE*, vol. 88, no. 4, pp. 451–515, Apr. 2000.
[9] S. Mahato, D. K. Yadav, and D. A. Khan, "A novel information hiding scheme based on social networking site viewers' public comments," *J. Inf. Secur. Appl.*, vol. 47, pp. 275–283, Aug. 2019.
[10] W. Mazurczyk, S. Wendzel, and S. Zander, *Information Hiding in Communication Networks: Fundamentals, Mechanisms, Applications, and Countermeasures*. Hoboken, NJ, USA: Wiley, 2016.
[11] E. Zielińska, W. Mazurczyk, and K. Szczypiorski, "Trends in steganography," *Commun. ACM*, vol. 57, no. 3, pp. 86–95, Mar. 2014.
[12] C. Li and X. Meng, "Zestful comment on acrostic poem," (in Chinese), *Extracurricular Chin.*, no. 12, pp. 47–48, 2015.
[13] W. Bender, D. Gruhl, N. Morimoto, and A. Lu, "Techniques for data hiding," *IBM Syst. J.*, vol. 35, nos. 3–4, pp. 313–336, 1996.
[14] J. Fridrich, *Steganography in Digital Media: Principles, Algorithms, and Applications*. Cambridge, U.K.: Cambridge Univ. Press, 2010, p. 49, ch. 4.
[15] B. Feng, Z.-H. Wang, D. Wang, C.-Y. Chang, and M.-C. Li, "A novel, reversible, Chinese text information hiding scheme based on lookalike traditional and simplified Chinese characters," *KSII Trans. Internet Inf. Syst.*, vol. 8, no. 1, pp. 269–281, Jan. 2014.
[16] S. Mahato, D. K. Yadav, and D. A. Khan, "A minesweeper game-based steganography scheme," *J. Inf. Secur. Appl.*, vol. 32, pp. 1–14, Feb. 2017.
[17] W. Tao, G. Chen, and R. Li, "Security steganography in interactive tetris game," (in Chinese), *Appl. Electron. Techn.*, vol. 42, no. 4, pp. 120–123, 2016.
[18] N. Zhang *et al.*, "An overview of text information hiding methods based on text format," (in Chinese), *Informatization Res.*, vol. 43, no. 3, pp. 1–6, 2017.
[19] W. Bhaya, A. Rahma, and D. Al-nasrawi, "Text steganography based on font type in MS-Word documents," *J. Comput. Sci.*, vol. 9, no. 7, pp. 898–904, 2013.
[20] A. A. Ali and A.-H. S. Saad, "New text steganography technique by using mixed-case font," *Int. J. Comput. Appl.*, vol. 62, no. 3, pp. 6–9, 2013.
[21] S. Mahato, D. K. Yadav, and D. A. Khan, "A novel approach to text steganography using font size of invisible space characters in microsoft word document," in *Intelligent Computing, Networking, and Informatics*. New Delhi, India: Springer, 2014, pp. 1047–1054.
[22] Y. Deming and G. Sheng, "Data hiding method based on word document," (in Chinese), *Comput. Appl. Softw.*, vol. 5, pp. 314–318, May 2015.

[23] S. Imran, A. Khan, and B. Ahmad, "Text steganography utilizing xml, html and Xhtml markup languages," *Int. J. Inf. Technol. Secur.*, vol. 9, no. 3, pp. 99–116, Jan. 2017.
[24] H. M. Meral, E. Sevinc, and E. Ünkar, B. Sankur, A. S. Özsoy, and T. Güngör, "Syntactic tools for text watermarking," *Proc. SPIE*, vol. 6505, Mar. 2007, Art. no. 65050X.
[25] L. Xiang, J. Yu, C. Yang, D. Zeng, and X. Shen, "A word-embedding-based steganalysis method for linguistic steganography via synonym substitution," *IEEE Access*, vol. 6, pp. 64131–64141, 2018.
[26] S. Shi, Y. Qi, and Y. Huang, "An approach to text steganography based on search in internet," in *Proc. Int. Comput. Symp. (ICS)*, Dec. 2016, pp. 227–232.
[27] X. Chen, H. Sun, Y. Tobe, Z. Zhou, and X. Sun, "Coverless information hiding method based on the chinese mathematical expression," in *Proc. Int. Conf. Cloud Comput. Secur.*, 2015, pp. 133–143.
[28] X. Yinghui, Y. Yu, and N. Xinxin, "Text steganography based on semantics," (in Chinese), *Comput. Syst. Appl.*, vol. 15, no. 6, pp. 91–94, 2006.
[29] Z. Minzhi, S. Xingming, and X. Huazheng, "Research on the chinese text steganography based on the modification of the empty word," (in Chinese), *Comput. Eng. Appl.*, vol. 42, no. 3, pp. 158–160, Mar. 2006.
[30] K. Wang and Q. Gao, "A coverless plain text steganography based on character features," *IEEE Access*, vol. 7, pp. 95665–95676, 2019.
[31] M. Chapman and G. Davida, "Hiding the hidden: A software system for concealing ciphertext as innocuous text," in *Information and Communications Security*. Berlin, Germany: Springer, 1997, pp. 335–345.
[32] K.Masher. (1981). (TEXTO). Accessed: May 23, 2019. [Online]. Available:ftp://ftp.funet.fi/pub/crypt/ steganography/texto.tar.gz
[33] W. Ren, Y. Liu, and J. Zhao, "Provably secure information hiding via short text in social networking tools," *Tsinghua Sci. Technol.*, vol. 17, no. 3, pp. 225–231, Jun. 2012.
[34] Y. Luo and Y. Huang, "Text steganography with high embedding rate: Using recurrent neural networks to generate chinese classic poetry," in *Proc. 5th ACM Workshop Inf. Hiding Multimedia Secur.*, Jun. 2017, pp. 99–104.
[35] Z. S. Yu, L. S. Huang, Z. Chen, L. Li, W. Yang, and X. Zhao, "High embedding ratio text steganography by ci-poetry of the song dynasty," (in Chinese), *J. Chin. Inf. Process.*, vol. 23, no. 4, pp. 55–63, 2009.
[36] Z. Yang, X. Guo, Z. Chen, Y. Huang, and Y. Zhang, "RNN-stega: Linguistic steganography based on recurrent neural networks," *IEEE Trans. Inf. Forensics Secur.*, vol. 14, no. 5, pp. 1280–1295, May 2019.
[37] Y. Hui and Y. Pan, "Research on knight travel problem algorithm," (In Chinese), *Int. Electron. Elements*, vol. 19, no. 11, pp. 112–114, 2011.
[38] Y. Pan and Y. Deng, "A ciphertext-only attack against the Cai-Cusick lattice-based public-key cryptosystem," *IEEE Trans. Inf. Theory*, vol. 57, no. 3, pp. 1780–1785, Mar. 2011.
[39] Y. Long, Y. L. Liu, Y. Zhang, X. Ba, and J. Qin, "Coverless information hiding method based on Web text," *IEEE Access*, vol. 7, pp. 31926–31933, 2019.
[40] Z. Zhou, Y. Mu, N. Zhao, Q. M. J. Wu, and C.-N. Yang, "Coverless information hiding method based on multi-keywords," in *Proc. Int. Conf. Cloud Comput. Secur.*, Nanjing, China, Jul. 2016, pp. 39–47.



CONGFEI SUN is currently pursuing degree with the College of Computer Science and Technology, Qingdao University. Her research interests mainly include information security, big data technology, and coverless text steganography.

KAIXI WANG was born in Qingdao, China, in 1971. He received the Ph.D. degree in telecommunications from the Beijing University of Posts and Telecommunications, Beijing, China, in 2008. He is currently an Associate Professor with the College of Computer Science and Technology, Qingdao University. His research interests mainly include network security, content security, next-generation networks, telecommunication software, and distributed computing.