Journal of Information Hiding and Multimedia Signal Processing ©2014 ISSN 2073-4212
Ubiquitous International Volume **5**, Number **3**, July **2014**

A Novel Lossless Steganographic Scheme for Data Hiding in Traditional Chinese Text Files

Chuan Qin

School of Optical-Electrical and Computer Engineering,
University of Shanghai for Science and Technology,
Shanghai, 200093, China
qin@usst.edu.cn

Chin-Chen Chang and Szu-Ting Wang

Department of Information Engineering and Computer Science,
Feng Chia University,
Taichung, 40724, Taiwan,
alan3c@gmail.com; agnesb.sky@hotmail.com

Chin-Chen Chang

Department of Computer Science and Information Engineering,
Asia University,
Taichung 41354, Taiwan,
alan3c@gmail.com
Correspondence Author

Received April, 2013; revised January, 2014

Abstract. *In this paper, we propose a novel lossless steganographic scheme in traditional Chinese text files. Currently, traditional Chinese characters often are encoded by the Big5 standard, and some traditional characters, called coherent characters, have the same appearances as their corresponding simplified versions encoded by the GBK standard. The proposed scheme takes advantage of this to conduct data hiding by adjusting the encodings of the coherent characters, and the incoherent characters are also utilized to increase the embedding rate. During the embedding procedure, the characters in the cover text file are segmented adaptively into a number of sections, and secret bits are hidden according to specific rules and mapping relationships. After extracting the secret bits, the cover text file can be recovered easily by restoring all of the adjusted character encodings to Big5. The content and appearance of the stego text in our scheme are completely the same as the cover text, so semantic distortions are avoided and the imperceptibility is satisfactory. Experimental results show the effectiveness of the proposed scheme.*

Keywords: Text steganography, Data hiding, Chinese character, Embedding rate, Reversibility.

1. **Introduction.** In recent years, both academia and industry have made extensive efforts to study and develop the steganography, which is a kind of technique that embeds specific information into a cover medium for various purposes, e.g., covert communication, copyright protection, or illegal distribution. Data hiding, as one of the applications for steganography, is used to transmit secret information through an insecure channelf [1]. Texts, audios, images, and videos in digital form can be used as the cover medium for data hiding. Besides the high hiding capacity, the imperceptibility should also be achieved in

data hiding to avoid making adversaries suspicious and to circumvent steganalysis tools [2]. In this work, we focus mainly on the study of data hiding for text files.

Many research works have been reported for the data hiding in text files, which is also referred to as linguistic steganography [3, 4, 5, 6, 7, 8]. Brassil *et al.* proposed a format-based data hiding scheme for image-containing text, in which the lines, words, and characters were shifted by a small amount to represent the secret embedding operations [4]. The strategy of synonym substitution was adopted in the methods of [5, 6]. In these kinds of methods, a number of constantly used synonym pairs were constructed before embedding, and then, the secret bits were embedded by simply substituting some words with their corresponding synonyms. Although the synonym substitution strategy can maintain the meaning of the original text to a certain extent, a reader often can detect the changes due to the inconsistencies with common usage or with the authors writing style. Grothoff *et al.* proposed a translation-based text steganographic scheme to disguise the stego text as an imperfect version produced by machine translation [7]. The secret bits were embedded into the translation errors that can seem to be noise, and an adversary may not become suspicious about the stego text generated by this translation-based steganographic scheme. Similarly, a scheme for hiding data in text by using a change-tracking technique was proposed for Microsoft Word documents in [8]. In this scheme, the stego document was disguised as the product of a collaborative writing effort. By using Huffman coding, the secret bits were embedded according to the choices of the different degenerations mimicking an author with inferior writing skills.

Because of the vast Chinese population, the Chinese language is used extensively throughout the world, and Chinese text files exist in the Internet environment abundantly. Thus, a Chinese text file is a good choice of a cover medium for data hiding [9, 10, 11, 12, 13]. In 2004, Sun *et al.* proposed a component-based scheme for data hiding in Chinese text, called the L-R scheme, in which the Chinese characters constructed with the left and right components were selected for embedding the secret bits [10]. According to the embedding secret bit, it can be determined whether the space between the left and right components of the character should be changed or not. As an improvement to the method proposed in [10], in 2009, Wang *et al.* incorporated the up and down components of Chinese characters rather than only the left and right components, which enhanced the hiding capacity [11]. In addition, their method achieved the reversibility, which means that the original cover text can be recovered after the hidden secrets are extracted from the stego text.

In general, Chinese characters can be represented in two forms, i.e., the traditional Chinese characters and the simplified Chinese characters. These two representation forms have the same meaning, but they have different encodings. In this paper, we propose a lossless data hiding scheme for the traditional Chinese text file by using the relationship between the traditional Chinese characters and the simplified Chinese characters. To the best of our knowledge, such a scheme has not been published by earlier researchers. Some traditional Chinese characters have exactly the same appearance as the corresponding simplified Chinese characters, although their encoding standards are different. In the proposed scheme, such kind of traditional Chinese characters are called *coherent characters*. To hide secret bits in a traditional Chinese text file, we analyze the order in which all of the characters appear and determine adaptively how to segment the characters into the sections according to specific mapping rules. The secret bits are embedded by adjusting the encodings of the coherent characters. Because both the content and the appearance of the stego text file after embedding are exactly the same as those of the original text file, thus, no semantic distortions occur. Subsequently, after extracting the hidden secret bits, the original text file can be obtained easily by restoring the character encodings.

The remainder of the paper is organized as follows. Section 2 describes our proposed lossless steganographic scheme for traditional Chinese text files, including the procedures of embedding and extracting. Experimental results and analysis are given in Section 3, and Section 4 concludes the paper.

2. Proposed Scheme.

In current international practice, the traditional Chinese characters and the simplified Chinese characters are usually encoded by the Big5 and GBK standards, respectively. In our scheme, data hiding is conducted in a text file of traditional Chinese characters. The symbol **C** is used to denote the type of the traditional Chinese characters that have the same appearance as their corresponding simplified versions, i.e., coherent characters, and the symbol **T** denotes the type of the traditional Chinese characters that have different appearances from their corresponding simplified versions, i.e., incoherent characters. Figure 1 presents an example of these two situations. The first row in Figure 1 shows 18 traditional Chinese characters, and the second row shows the corresponding simplified versions of the characters in the first row. Their corresponding types of symbols are listed in the third row. The procedures of secret data embedding and extracting are described in detail below.

互	聯	網	系	指	網	絡	之	間	串	聯	成	的	龐	大	網	絡	群
互	联	网	系	指	网	络	之	间	串	联	成	的	庞	大	网	络	群
C	**T**	**T**	**C**	**C**	**T**	**T**	**C**	**T**	**C**	**T**	**C**	**C**	**T**	**C**	**T**	**T**	**C**

FIGURE 1. An example of traditional Chinese characters and the corresponding simplified versions

2.1. Embedding Procedure.

The original cover text consists of the traditional Chinese characters in Big5 encodings. Our embedding mechanism is based on the features that the traditional Chinese character categorized as type **C** can be encoded by two different standards, i.e., Big5 and GBK, with the same appearance, and the characters of type **T** can also be used to increase the embedding rate. During the embedding procedure, the characters in the traditional Chinese cover text file are read in order. For each character that is read, there are two cases, i.e., 1) the character belongs to type **C** or 2) the character belongs to type **T**. We utilize two rules for these two cases in order to embed secret bits. A flowchart of the embedding procedure is given in Figure 2.

In order to embed secret bits, all the characters in the cover text file are segmented adaptively into sections according to the embedding rules. If the current character X is a type **C** character, it is segmented as a section independently, and the embedding rule I is applied for this section to embed the secret bit, which simply adjusts the encoding of X to make it consistent with the relationship in Eq. (1).

$$\text{Encoding of } X = \begin{cases} \mathbf{B}, & \text{if } s = 1, \\ \mathbf{G}, & \text{if } s = 0, \end{cases} \tag{1}$$

where s is the current secret bit for embedding, **B** and **G** denote the encodings of Big5 and GBK, respectively. In other words, if the embedding secret bit is equal to 0, the character

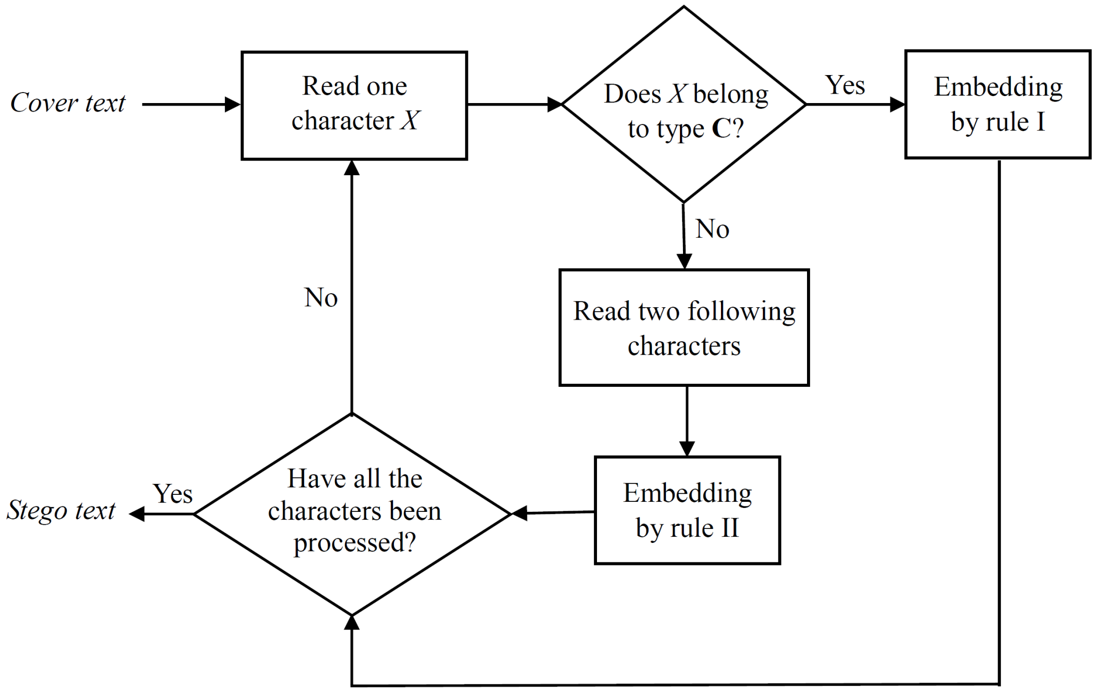

FIGURE 2. The flowchart of the embedding procedure

X is re-encoded using GBK, and if the embedding secret bit is 1, it is not necessary to change the encoding of X.

If the current character X belongs to type **T**, the following two characters are also read to form a section of three characters, and the embedding rule II described below is applied. Obviously, there are four different type representations for the section with these three characters, i.e., **TCC**, **TTC**, **TCT**, and **TTT**. If the type representation of the section is **TCC**, the first character with type **T** should be marked to denote that no secret bits are embedded, and the following two type **C** characters can both be embedded with one secret bit using Eq. (1). For the section with one of the three remaining type representations, i.e., **TTC**, **TCT**, or **TTT**, the mapping in Table 1 is used to embed the secret bits. Explicitly, there are three scenarios that can be used to embed two secret bits into the section directly, and the encodings of the section must be adjusted. The three scenarios are:

(1)If type representation is **TTC** and the current two secret bits to be embedded are 10 or 11, the encodings of the three characters in this section are adjusted to **BBG** or **BBB** accordingly.

(2) If type representation is **TCT** and the current two secret bits to be embedded are 01 or 00, the encodings of the three characters are adjusted to **BGB** or **BBB** accordingly.

(3) If type representation is **TTT** and the current two secret bits to be embedded are 01, the encodings of the three characters are kept unchanged as **BBB**.

If the type representations of the section and the two secret bits to be embedded are not consistent with any one of these three scenarios, i.e., the mapping relationships in Table 1, the first character of type **T** in the section should be marked to denote that no secret bits are embedded. Then, for the inconsistent section with type representations of **TTC** or **TTT**, the first marked character of type **T** is skipped and one character following the remaining two characters is read to form a new section of three characters. The newly formed section, i.e., one of **TCC**, **TCT**, **TTC**, **TTT**, can be conducted to embed secret bits recursively by the above rule. For the old inconsistent section with type representation **TCT**, after the first character of type **T** is marked and skipped, the middle character of type **C** can be embedded with one secret bit using Eq. (1), and two characters following the third character of type **T** are read to form a new section of three characters. Similarly, the above rule can be applied recursively on the newly formed section to conduct data hiding.

The embedding procedure is not complete until all the characters in the text file are processed using the above two rules. Based on the above analysis, we can see that, although secret bits are hidden, the text file after embedding, i.e., the stego text, has exactly the same appearance and content as the cover text, and only the encodings of some type **C** characters are changed, which does not influence the semantics of the text file. Therefore, the imperceptibility of the proposed steganographic scheme is satisfactory.

TABLE 1. The mapping for three type representations

Type Representations	Mapped Encodings	Secret Bits for Embedding
TTC	**BBG**	10
	BBB	11
TCT	**BGB**	01
	BBB	00
TTT	**BBB**	01

2.2. **Extracting Procedure.** In addition to the stego text file, the extra information that denotes the characters of type **T** with no secret bits embedded should also be transmitted to the receiver side for data extraction. First, all of the characters in the stego file are identified to obtain their types, i.e., **C** or **T**, and their encodings, i.e., **B** or **G**. Then, the two rules used in the embedding procedure assist in the extraction of the secret bits. A flowchart of the extracting procedure is shown in Figure 3.

During the extracting procedure, the characters in the stego text file also are read in order. For each current character X that is read, if its type is **C**, the rule I is adopted. The character is segmented as an independent section, and Eq. (2) is used to extract one embedded secret bit s directly.

$$s = \begin{cases} 1, & \text{if Encoding of } X \text{ is } \mathbf{B}, \\ 0, & \text{if Encoding of } X \text{ is } \mathbf{G}, \end{cases} \tag{2}$$

If the current character X is a type **T** character, the rule II is adopted to extract the secret bits. First, according to the extra information, the current character X of type **T** is checked to determine whether it is marked or not. If X is marked, that means no secret bits are embedded in this character, and it should be skipped. Thus, a new following character is read recursively. However, if X is not marked, the two following characters are read to form a section of three characters, and the type representation of this section

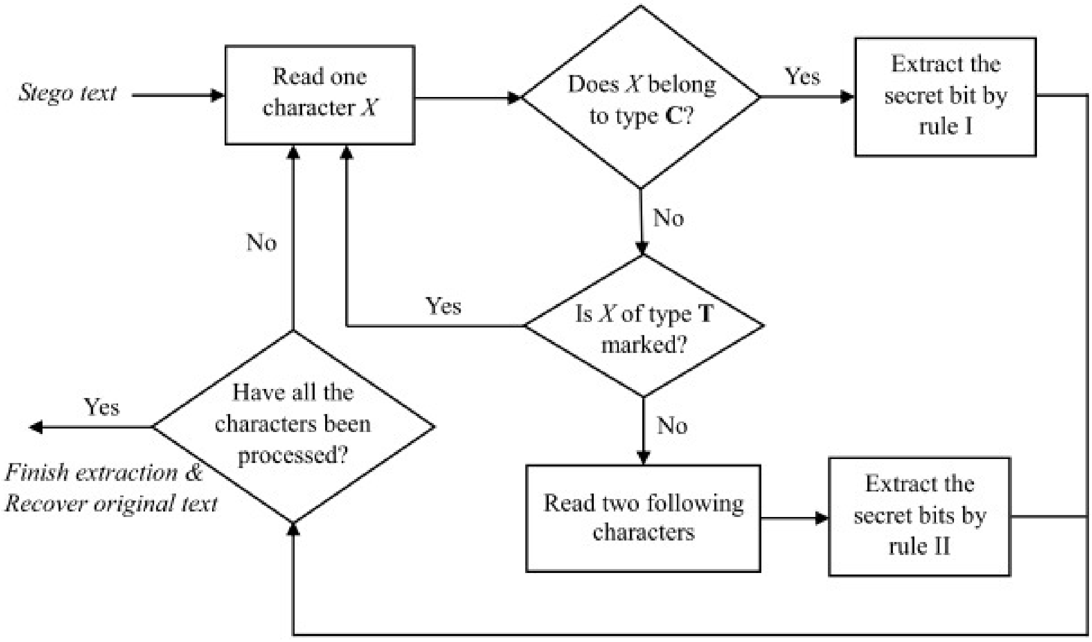

FIGURE 3. The flowchart of the extracting procedure

must be one of **TTC**, **TCT**, **TTT**. According to the rule II in the embedding procedure, we can know that two secret bits are embedded in this section. Therefore, according to the type representation and the encodings of this section, the two embedded secret bits can be extracted easily by looking up the mapping relationships in Table 1. Then, the following characters continue to be processed in the same way by rule I or rule II.

After all the characters in the stego text file are processed, the extracting procedure is finished, and all of the embedded secret bits are extracted. Then, the encodings of all adjusted characters are restored to **B**, i.e., Big5, which is the standard encoding of traditional Chinese characters. Consequently, the stego text file is reversed to its original version losslessly.

3. **Experimental Results and Analysis.** Experiments were conducted on a large number of traditional Chinese text files. Secret bits for embedding were generated pseudorandomly. In order to show the effectiveness of the proposed scheme, three aspects of the experimental results are presented: (1) An illustration of secret data embedding and extracting; (2) Comparisons of embedding results between the usage of both types **C** and **T** and the usage of only type **C**; and (3) Performances of the proposed scheme with different portions and distributions of the character types.

3.1. **An Illustration of Data Embedding and Extracting.** In this subsection, an illustration of data embedding and extracting for our scheme is presented. Assume that the eighteen characters in the cover traditional Chinese text file are shown in the first row of Figure 1 and that all of them are initially encoded by **B**, i.e., Big5. The types of all the characters are **CTTCCTTCTCTCCTCTTC**, and the secret bits for embedding are given as "00101011110".

Since the type of the 1^{st} character is **C** and the first secret bit for embedding is 0, the encoding of this character is adjusted as GBK, i.e., **G**, according to Eq. (1). The type of the 2^{nd} character is **T**, so its two following characters, i.e., the 3^{rd} and 4^{th} characters, are also read to form a section of the 2^{nd}, 3^{rd}, and 4^{th} characters. Because the two secret bits "01" that are to be embedded do not match the mapping secret bits of the type representation listed in Table 1 for this section, i.e., **TTC**, the 2^{nd} character should be marked and skipped. The 5th character following this section is read to form a new section of the 3^{rd}, 4^{th}, and 5^{th} characters with type representation **TCC**. According to rule II, the 3^{rd} character of type **T** is marked and skipped, and then, the 4^{th} and 5^{th} characters of type **C** are embedded individually with the secret bits 0 and 1 by adjusting the encoding of the 4^{th} character into **G**. Then, the section of the 6^{th}, 7^{th}, and 8^{th} characters with type representation **TTC** are read, but the current to-be-embedded secret bits "01" do not match the mapping relationship of the section in Table 1. Thus, the 6^{th} character is marked and skipped, and the following 9^{th} character is read to form a new section of the 7^{th}, 8^{th}, and 9^{th} characters with type representation **TCT**. The secret bits "01" are embedded into this section by adjusting the encoding of the 8^{th} character into **G**. The 10^{th} character of type **C** is embedded with the secret bit 0 by changing its encoding into **G** directly. The following section of the 11^{th}, 12^{th}, and 13^{th} characters with type representation **TCC** is read. In the same way, the 11^{th} character is marked and skipped, and the secret bits "11" are embedded into the 12^{th} and 13^{th} characters without changing the encodings. The type representation **TCT** of the next section for the 14^{th}, 15^{th}, and 16^{th} characters is inconsistent with the mapping of the two current embedding secret bits "11", thus, the 14^{th} character is marked and skipped, and the 15^{th} character of type **C** is embedded with secret bit 1. The last two secret bits "10" are embedded into the final section of the 16^{th}, 17^{th}, and 18^{th} characters with type representation **TTC** by adjusting the encoding of the 18^{th} character into **G**. Therefore, after finishing the embedding procedure, 11 secret bits are hidden, and the encodings of all 18 characters are **GBBGBBBGBGBBBBBBBG**. Only the encodings of the 1^{st}, 4^{th}, 8^{th}, 10^{th}, and 18^{th} characters are changed to **G**, and the 2^{nd}, 3^{rd}, 6^{th}, 11^{th}, 14^{th} characters of type **T** are marked to denote that no secret bits are embedded.

On the receiver side, the 18 characters in the stego text file have the same appearance as the original version, and their encodings can also be obtained. First, the receiver identifies the types of all of the characters and skips five marked characters according to the received extra information. The types of the remaining 13 characters are **CCCTCTCCCCTTC**, and the corresponding encodings are **GGBBGBGBBBBBG**. By rules I and II, we can segment these characters into sections, i.e., **C** | **C** | **C** | **TCT** | **C** | **C** | **C** | **C** | **TTC**, and their section encodings are **G** | **G** | **B** | **BGB** | **G** | **B** | **B** | **B** | **BBG**, where "|" denotes the segmentation symbol. Therefore, by using Eq. (2) and checking Table 1, the embedded secret bits, 00101011110, can be extracted easily.

Finally, all the changed encodings for these 18 characters are restored to **B**, so the original text file of traditional Chinese characters is recovered losslessly.

3.2. **Comparison of the Usage of Character Types.** A special situation of the proposed scheme is that only the type **C** characters in the cover text are used for embedding by Eq. (1), and none of the type T characters is used [12]. Obviously, in this special situation, no extra information is required. We compared the performances of our scheme when only type **C** characters were used and when both type **C**and type **T** characters were used.

Note that, in order to reduce redundancy, the extra information in the proposed scheme is compressed by arithmetic coding before transmission. Hence, the *pure* embedding rate R of our scheme is calculated as follows:

$$R_s = \frac{C_s - C_e}{L} \text{ (bpc)}, \quad (3)$$

where C_s and C_e denote the number of all embedded bits and the number of the compressed extra information bits, respectively, L is the number of total characters in the traditional Chinese text file, and bpc denotes bits per character.

Figures 4 and 5 show two examples, and each of them has 48 traditional Chinese characters, i.e., $L = 48$. The encodings of the characters in these two examples are **B** initially. Assume that the embedding secret bits for the characters in Figures 4 and 5 are "01101010101001010101001101010" and "010111000110110010111110011010101101", respectively. The first three rows of Figures 6 and 7 show the types of the characters in Figures 4 and 5. The middle three rows of Figures 6 and 7 are the adjusted encodings of the characters in Figures 4 and 5 after embedding secret bits by only using the type **C** characters. The last three rows of Figures 6 and 7 are the adjusted encodings obtained by using both type **C** and type **T** characters. The results of the performance comparison for these two methods are presented in Table 2. We can find that, although the method in which only type **C** characters are used does not require the extra information, the method in which both type **C** and type **T** characters are used has a higher pure embedding rate R.

3.3. **Performances with Different Percentages and Distributions of Character Types.** We also evaluated the performance of the proposed scheme for traditional Chinese text files, in which the percentages and distributions of the types of characters are different. Note that both type **C** and type **T** characters were used for embedding in this evaluation.

Suppose that there are two groups, both of which consist of five paragraphs of characters, and the five paragraphs in each group all have 100 characters with the percentages of type **T** characters ranging from 30% to 70%, individually, as shown in (a)-(e) of Figures 8 and 9. The distribution of the two types, i.e., **C** and **T**, in each character paragraph of Figure 8 is relatively uniform, whereas the distribution of the two character types in Figure 9 is more concentrated. Table 3 presents the performance results after embedding the secret bits in these 10 character paragraphs of Figures 8 and 9. Because the extra information in the proposed scheme is easier to compress for the character paragraph with the concentrated distribution than for the uniform distribution, we can see from Table 3 that the pure embedding rates R of the character paragraphs in Figure 9 are greater than those of the character paragraphs in Figure 8.

齒 為 鈣 化 組 織 齒 的 組 織 構 成 複 雜 少 數 動 物 齒 裏
有 著 複 雜 神 經 僅 具 觸 覺 潔 齒 劑 通 常 為 膠 狀 鹽 與
小 蘇 打 能 潔 淨 口 氣

FIGURE 4. Example 1 of traditional Chinese characters

牙 齒 類 型 是 門 齒 犬 齒 和 臼 齒 門 齒 用 來 切 斷 犬 齒
用 來 撕 裂 臼 齒 則 是 用 來 磨 碎 外 層 構 成 組 織 為 象
牙 質 屬 於 主 軀 體 幹

Figure 5. Example 2 of traditional Chinese characters

Table 2. Performance comparison between the usage of only **C** and the usage of both **C** & **T**

	No. of changed encodings		All embedded bits C_s		Extra info. bits C_e		Pure embedding rate R	
Methods	only **C**	**C** & **T**	only **C**	**C** & **T**	only **C**	**C** & **T**	only **C**	**C** & **T**
Example 1	7	11	14	29	0	12	0.2917	0.3542
Example 2	9	9	22	36	0	13	0.4583	0.4792

T T T C T T T C T T T C T T C T T C T T
C T T T C T T C T T T T T C C T T T T T
C T C C T T C T

B B B G B B B B B B B B B B G B B B B B
G B B B B B B G B B B B B B G B B B B B
B B G G B B B B

B B B B B B B G B B B G B B G B B G B B
G B B B G B B G B B B B B G G B B B B B
B B G B B B G B

Figure 6. The character types and the adjusted encodings after embedding secret bits for Example 1 in Figure 4. The first three rows are the character types in Figure 4, and the middle three rows and the last three rows are the adjusted encodings for the usage of only type **C** and the usage of both type **C** and type **T**, respectively.

C T T C C T T C T C C T T T C T C T C T
C T C C C T T C C T C C C T T C T T T C
C T T T C T T T

G B B B G B B B B B B B B B G B G B G B
B B B G B B B B G B G B G B B B B B B B
B B B B B B B B

G B B G B B B B B G G B B B B B G B B B
B B B G B B B B B B B G G B B B B B B G
B B B B B B B B

FIGURE 7. The character types and the adjusted encodings after embedding secret bits for Example 2 in Figure 5. The first three rows are the character types in Figure 5, and the middle three rows and the last three rows are the adjusted encodings for the usage of only type **C** and the usage of both type **C** and type **T**, respectively.

C C T C C T T C C C
C C T T C C C T C C
C T C T T C C C C C
C C T C T C C T C C
C C C T T C C C C T
C T T C C T C C C C
C C C C T T C C T C
C T C C C C C T T C
C C C T C C T T C C
T C T C C C T C C C

(a)

C T T C C T T C C C
C C T T C C C T T C
C T C T T C T C C C
C C T C T T C T C C
C T C T T C C C C T
C T T C C T C T C C
C C C T T T C C T C
C T T C C C C T T C
C C T T C C T T C C
T C T C C T T C C C

(b)

C T T C C T T T C C
C C T T T C C T T C
C T T T T C T C C C
C T T C T C C T T C
C T C T T C T C C T
C T T C T T C T C C
C C T T T T C C T C
C T T C C C T T T C
C C T T C C T T T C
T C T T C T T C C C

(c)

C T T C C T T T T C
C C T T T C T T T C
C T T T T C T C T C
C T T C T T C T T C
C T C T T C T T C T
C T T C T T C T T C
C C T T T T C T T C
C T T C C T T T T C
C C T T T C T T T C
T C T T C T T C T C

(d)

C T T C T T T T T C
T C T T T C T T T C
C T T T T T T C T C
C T T C T T T T T C
C T C T T T T T C T
C T T C T T T T T C
C C T T T T T T T C
C T T C T T T T T C
C C T T T C T T T T
T C T T C T T C T T

(e)

FIGURE 8. Relatively uniform distributions of character types with different percentages of **T**. The percentages of type **T** characters of the five character paragraphs in (a)-(e) are 30%, 40%, 50%, 60%, and 70%, respectively.

```
C C C C C C C C C C
C C C T C T C T T C
T C T C T C T T C T
C T T C T T T C T T
C T T T C T T C T T
T T T C T T C T C C
C C C C C C C C C C
C C C C C C C C C C
C C C C C C C C C C
C C C C C C C C C C
```

(a)

```
C C C C C C C C C C
C C T T C T C T T C
T C T C T C T T C T
C T T C T T T C T T
C T T T C T T C T T
T T T C T T C T T T
T C T T T C T T T C
C C C C C C C C C C
C C C C C C C C C C
C C C C C C C C C C
```

(b)

```
C C T C T T C T T T
C T T T C T C T T C
T C T C T C T T C T
C T T C T T T C T T
C T T T C T T C T T
T T T C T T C T T T
T C T T T C T T T C
T T C T C C C C C C
C C C C C C C C C C
C C C C C C C C C C
```

(c)

```
C C C C C T T T T T
C T T T C T T T T C
T C T C T T T T C T
C T T C T T T C T T
C T T T C T T C T T
T T T C T T T T T T
T C T T T C T T T C
T T C T T C T T T T
T T T C C C C C C C
C C C C C C C C C C
```

(d)

```
C T T T T T T T T T
C T T T C T T T T C
T C T C T T T T C T
C T T T T T T C T T
C T T T C T T C T T
T T T C T T C T T T
T C T T T C T T T T
T T C T T C T T T T
C T T T C T T T T C
T C C C C C C C C C
```

(e)

FIGURE 9. Relatively concentrated distributions of character types with different percentages of **T**. The percentages of type **T** character of the five character paragraphs in (a)-(e) are 30%, 40%, 50%, 60%, and 70%, respectively.

TABLE 3. Results with different percentages and distributions of character types

Percentage of **T**	Distributions in Fig. 8			Distributions in Fig. 9		
	C_s	C_e	R	C_s	C_e	R
30%	76	25	0.51	81	14	0.67
40%	70	18	0.52	75	15	0.60
50%	67	17	0.50	69	16	0.53
60%	57	19	0.38	61	18	0.43
70%	54	19	0.35	55	19	0.36

4. **Conclusions.** A reversible steganographic scheme for data hiding in traditional Chinese text files is presented in this paper. The type of each traditional Chinese character in the cover text is first judged by identifying whether it has the same appearance as its simplified version or not. Then, all the characters are segmented adaptively into a number of sections with one character or three characters. According to their types and the mapping rules, secret bits are embedded into the segmented sections of characters by adjusting the encodings of some coherent characters from Big5 to GBK. Although the encodings of some characters are changed, all the characters in stego text file have exactly

the same appearance as those of the cover text. The original cover text file can be recovered completely by simply restoring all the changed character encodings back to Big5. The experimental results demonstrate that the proposed scheme has high embedding rate and satisfactory imperceptibility.

Acknowledgment. This work was supported by the Natural Science Foundation of China (61303203), the Natural Science Foundation of Shanghai, China (13ZR1428400), and the Innovation Program of Shanghai Municipal Education Commission (14YZ087).

REFERENCES

[1] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn, Information hiding – a survey, *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1062-1078, 1999.

[2] H. Wang, and S. Wang, Cyber warfare: steganography vs. steganalysis, *Communications of the ACM*, vol. 47, no. 10, pp. 76-82, 2004.

[3] M. J. Atallah, V. Raskin, M. Crogan, C. Hempelmann, F. Kerschbaum, D. Mohamed, and S. Naik, Natural language watermarking: design, analysis, and a proof-of-concept implementation, *Proc. of the 4th International Workshop on Information Hiding*, pp. 185-200, 2001.

[4] J. T. Brassil, and N. F. Maxemchuk, Copyright protection for the electronic distribution of text documents, *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1181–1196, 1999.

[5] K. Bennett, Linguistic steganography: survey, analysis, and robustness concerns for hiding information in text, *CERIAS Technical Report 2004-13*, Purdue University, West Lafayette, IN, 2004.

[6] Y. L. Chiang, L. P. Chang, W. T. Hsieh, and W. C. Chen, Natural language watermarking using semantic substitution for chinese text, *Proc. of International Workshop on Digital Watermarking*, LNCS 2939, pp. 129-140, 2003..

[7] C. Grothoff, K. Grothoff, L. Alkhutova, R. Stutsman, and M. Atallah, Translation-based steganography, *Journal of Computer Security*, vol. 17, no. 3, pp. 269-303, 2009.

[8] T. Y. Liu, and W. H. Tsai, A new steganographic method for data hiding in microsoft word documents by a change tracking technique, *IEEE Trans. Information Forensics and Security*, vol. 2, no. 1, pp. 24-30, 2007.

[9] X. M. Sun, H. W. Chen, L. H. Yang, and Y. Y. Tang, Mathematical representation of a chinese character and its applications, *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 16, no. 8, pp. 735-747, 2002.

[10] X. M. Sun, G. Luo, and H. J. Huang, Component-based digital watermarking of chinese texts, *Proc. of the 3rd International Conference on Information Security*, pp. 76-81, 2004.

[11] Z. H. Wang, C. C. Chang, C. C. Lin, and M. C. Li, A reversible information hiding scheme using left-right and up-down chinese character representation, *Journal of Systems and Software*, vol. 82, no. 8, pp. 1362-1369, 2009.

[12] 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 Transactions on Internet and Information Systems*, vol. 8, no. 1, pp. 269-281, 2014.

[13] C. C. Lin, L. C. Yang, and Y. H. Chen, Data hiding scheme based on features of chinese text, *Proc. of the 7th International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pp. 161-164, 2011.