About using Unicode to hide information in a text document

N.R. Zaynalov,[a] U.Kh. Narzullaev,[b] A.N. Muhamadiev,[c] and D. Qilichev[d]
Samarkand branch of the Tashkent University of Information Technologies named after Muhammad al-Khwarizmi, Uzbekistan

(Dated: 5 September 2020)

Abstract. Steganography develops tools and methods for hiding the fact of message transmission. The first traces of steganographic methods are lost in ancient times. From detective works, various methods of secret writing between the lines of ordinary text are well known: from milk to complex chemical reagents with subsequent processing. Digital steganography is based on hiding or embedding additional information in digital objects while causing some distortion of these objects. In this case, text, images, audio, video, network packets, and so on can be used as objects or containers. To embed a secret message, steganographic methods rely on redundant container information or properties that the human perception system cannot distinguish. Recently, there has been a lot of progress in hiding information in a text container, since text documents are used in many organizations. Based on this, here the MS Word document is considered as a data carrier, which has various parameters, changing these parameters can achieve data integration. In the same article, we present steganography using invisible Unicode characters of the Space type, but with a different encoding.

INTRODUCTION

The development of information and communication technologies has led to the emergence of modern steganography, which deals with information in electronic form, rather than with physical objects and texts. This is mainly because the process of hiding and retrieving a secret message can be automated. This allows you to effectively conduct experiments using computer technology and appropriate software applications.

Steganography is the science of hiding data inside a coverage object to preserve an invisible secret message without compromising the integrity of the coverage object. A common feature of these methods and algorithms is that the hidden message is embedded in some harmless, non-attracting object that is passed to the recipient openly [1]. When using cryptography, the presence of an encrypted message itself attracts the attention of an attacker; in the case of steganography, the presence of hidden information remains unnoticeable. The plain text where information will be hidden by the steganographic algorithm is called a container.

Volume, safety, and reliability, which are the three main factors affecting steganography, are in principle factors that contradict each other. Volume is the relative number of bits of secret information that can be hidden in a container. Security is the ability to find out hidden information from the enemy. Reliability refers to the number of modifications that the stegosred can withstand before the enemy destroys the hidden information [2]. An appropriate balance should be sought between the three aspects by specific requirements.

From the review given in [3, 4], we can conclude that most text steganography is based on the formats TXT, MS Word, PDF, PPT, and so on. However, here we try to improve the method of invisible characters between words with additional spaces for embedding data in an MS Word document. This article also discusses the existing algorithmic approaches to steganography in MS Word documents to hide additional information in it.

Analysis of text steganography methods indicates that the variety of methods has not yet led to a qualitative method of text steganography, which is stable and capacious. Text steganography is relatively backward compared to the main concealment methods that use images, audio, and video as covering data, due to the lack of redundancy in the text [5, 6]. Despite this, storing text files requires less memory, and it‘s easier compilation and exchange makes it preferable over other types of steganographic methods.

This paper presents a method for hiding data using non-visible character attributes from a Unicode table in MS Word. This article introduces a new approach to text steganography by hiding a message in a set of Space characters of various Unicode codes, which we will denote as UniSpace. This method works with the ASCII character value, not bits.

[a]Corresponding author: nodirz@mail.ru
[b]Electronic mail: ulug1956_56@mail.ru
[c]Electronic mail: nabi8888@bk.ru
[d]Electronic mail: sinfdosh1990@mail.ru

Unicode Standard. Unicode is a universal character encoding standard that is used to support non-ASCII characters. Unicode provides support for all the world's languages and their unique character sets. Unicode can support more than 1 million characters. The reason is that Unicode can use more position bits to represent a character, which are units of information in computers. ASCII characters only require 7 bits, while Unicode can use 16 bits. This is necessary because some languages, such as Chinese and Arabic, require more position bits. At the same time, the Unicode table for characters in a language such as Arabic includes languages such as Persian, Urdu, Pashto, Sindhi, and Kurdish. The standard provides detailed explanations of implementation methods, including the letter join method, right-to-left text insertion, and much more [7].

For our research, we will rely on the work [8], where we are interested in Unicode codes for spaces, which will be used in the following sections, namely (see table 1):

TABLE I. UNISPACE space code notation in Unicode.

Code	Name	Code	Name
U+0020	Space	U+2005	Four-Per-Em Space
U+00A0	No-Break Space	U+2006	Six-Per-Em Space
U+1680	Ogham Space Mark	U+2007	Figure Space
U+180E	Mongolian Vowel Separator	U+2008	Punctuation Space
U+2000	En Quad	U+2009	Thin Space
U+2001	Em Quad	U+200A	Hair Space
U+2002	En Space	U+202F	Narrow No-Break Space
U+2003	Em Space	U+205F	Medium Mathematical Space
U+2004	Three-Per-Em Space	U+3000	Ideographic Space

EXISTING APPROACH

In this section, we present some of the well-known approaches to text steganography in MS Word documents. At the same time, the methods of text steganography considered are based on invisible characters or based on Unicode encoding, the implementation of which in various ways allows you to create sequences of bits of a secret message. The study of scientific literature on this topic allows you to create new directions in methods of hiding information.

One well-known method is White Steg, which uses the standard Space character to hide a secret message. At the same time, bit encoding is performed reasonably. For example, one space after a word represents bit 0, and two spaces after a word represent bit 1 [9].

The wbStego4open method also uses a space character, together with a null space, which has the code 0x00. At the same time, the space between sentences and between words is used for embedding the payload. To embed a secret message, the space character is replaced with the code value 0x00 for embedding bit 1 or the code value 0x20 for embedding bit 0 [10].

A modification of this method is proposed in [11]. In the proposed algorithm, additional null space will be added if the embedded bit is equal to 1, otherwise, the null space will remain unchanged.

A unique application of Unicode encoding is given in [12-14]. These papers propose a method based on a Unicode table where the composite form of some characters (i.e. a sign consists of two or more Unicode codes) is used in Unicode to hide the secret code bits. These characters defined in Unicode have both a single form and a composite form. By alternating these forms of writing letters, you can represent a single bit of information. The use of this approach to hide secret data can be observed in Chinese, Bengali, Arabic, and Persian texts.

Certain modifications of these algorithms can be observed in other works. For example, [15] uses features of Arabic writing and presents a steganographic algorithm also based on Unicode encoding. Which is based on processing only related letters, while the size and shape of the text remain unchanged?

The following articles provide an overview of various steganographic methods for Arabic text, where Arabic letters have many forms following the Unicode standard [16]. This method uses different possible Unicode values of the same letter to hide the bits, as explained in [17-19]. In this paper [16] we propose a steganographic algorithm based on the features of the Arabic text, taking into account the Unicode encoding. In this case, the main idea is to process isolated Arabic letters, which use individual letters as hiding data in Arabic texts written in Unicode format. And to

simplify the complexity of the algorithm, it is proposed to consider only individual letters at the beginning and end of words, and not all isolated letters in words.

In [8], a method called UniSpaCh is proposed. This method is an improved version of the White Steg method discussed above. Here, additional characters of the Space type, from the Unicode encoding, are inserted between the words suggested. For example, characters such as Punctuation, Thin, En Quad, Em Quad, Hair in sentences between words. The advantage of these spaces over a normal space is that the width of these characters is too small. As an alternative to the text container in [20], a study is conducted to hide bits in an MS Excel document. This paper also proposes a steganographic method for effectively hiding information using the Unicode character encoding system. In this case, a unique fact is used, namely, seven numbers (9, 8, 7, 3, 2, 1, 0) in the Unicode standard, they have the same form, but different codes in Arabic and Persian. As a result, by alternating these codes, you can hide information in an MS Excel document.

The method called SEFT technique in [21] is useful for our research. This study proposes a new method of text steganography that takes font types into account. This method depends on the similarity of font types in English. It works by replacing the font with more similar fonts. The secret message was encoded and embedded in similar fonts in the capital letters of the accompanying document, combining different fonts, which are designated as F1, F2, F3. by Combining these fonts, you can encode 27 characters, which is enough for English text.

A brief overview of scientific research in the field of steganography in MS Word documents and the formation of these methods are given in many works [4]. In this paper, it is proposed to hide information between words by the additional implementation of several invisible codes. And instead of the standard Space code, the combination of these invisible UniSpace codes will mean one letter of the Latin alphabet, following the proposed encoding.

PROPOSED APPROACH

As was correctly noted in [19], Unicode-based steganography methods have common disadvantages, which can be characterized as follows:

- Some Unicode-based steganography methods provide high performance, but this requires radically changing the content of the carrier text, while the main idea in steganography is that the method should be statistically undetectable.

But it should be noted that the essence of all Unicode-based steganography methods automatically implies changes to characters in the container text, based on its analog from the Unicode code table. This will cause data to be hidden in each letter in the target word. However, the grammatical form of a word or sentence changes, so you need an algorithm that does not spoil the form of words.

The word-spacing method allows you to embed a message in the text that has a binary format by placing one or two spaces after each word in sentences. However, these methods have a small amount of embedding. Based on this, it is suggested to embed ASCII characters instead of binary data. This technology is implemented using the following codes, which will be the basis for this approach (see table 2):

TABLE II. Basic space codes in the algorithm.

Space	THIN SPACE	HAIR SPACE	ZERO WIDTH SPACE
Unicode	2009	200A	200B

Thus, this study proposes a new method using characters that have a single character within the Unicode encoding system (i.e. similar characters with different codes in the Unicode table) for embedding a secret message in an MS Word document. In the proposed version, you can hide a secret message in a Word document using various variants consisting of three basic space codes from table 2.

The comparison of one-to-one correspondence of letters from the Latin alphabet will be carried out according to the following scheme (to save space, skip the word SPACE in this table, and enter the abbreviation ZERO WIDTH = ZW, see table 3).

The last combination of the triple ZERO WIDTH can be used as the beginning and end of the hidden text. To digitize this data, we apply a ternary number system to the data in table 3, namely (we denote THIN-0, HAIR-1, ZERO WIDTH-2) (see table 4).

For the convenience of defining a set of spaces by the character (and then by its code), we will create an array for 3 types of myspace spaces (3), where the elements of the MySpace(i) array can take one of the values: THIN, HAIR, ZERO WIDTH.

TABLE III. Encoding of the Latin alphabet.

Combination of spaces			Symbol	Combination of spaces			Symbol
THIN	THIN	THIN	A	HAIR	HAIR	ZW	O
THIN	THIN	HAIR	B	HAIR	ZW	THIN	P
THIN	THIN	ZW	C	HAIR	ZW	HAIR	Q
THIN	HAIR	THIN	D	HAIR	ZW	ZW	R
THIN	HAIR	HAIR	E	ZW	THIN	THIN	S
THIN	HAIR	ZW	F	ZW	THIN	HAIR	T
THIN	ZW	THIN	G	ZW	THIN	ZW	U
THIN	ZW	HAIR	H	ZW	HAIR	THIN	V
THIN	ZW	ZW	I	ZW	HAIR	HAIR	W
HAIR	THIN	THIN	J	ZW	HAIR	ZW	X
HAIR	THIN	HAIR	K	ZW	ZW	THIN	Y
HAIR	THIN	ZW	L	ZW	ZW	HAIR	Z
HAIR	HAIR	THIN	M	ZW	ZW	ZW	
HAIR	HAIR	HAIR	N				

TABLE IV. Numeric encoding of the Latin alphabet.

Position			The numeric value of the code	Symbol	Position			The numeric value of the code	Symbol
1	2	3			1	2	3		
0	0	0	0	A	1	1	2	14	O
0	0	1	1	B	1	2	0	15	P
0	0	2	2	C	1	2	1	16	Q
0	1	0	3	D	1	2	2	17	R
0	1	1	4	E	2	0	0	18	S
0	1	2	5	F	2	0	1	19	T
0	2	0	6	G	2	0	2	20	U
0	2	1	7	H	2	1	0	21	V
0	2	2	8	I	2	1	1	22	W
1	0	0	9	J	2	1	2	23	X
1	0	1	10	K	2	2	0	24	Y
1	0	2	11	L	2	2	1	25	Z
1	1	0	12	M	2	2	2	26	
1	1	1	13	N					

The proposed concealment algorithm consists of 6 stages. In the first stage, an empty .docx text container is opened. In the second stage, a hidden text consisting of a sequence of Latin letters only is requested. In the third stage, the document takes the starting point for embedding data and marks it with code 26. At the fourth stage, the container capacity is checked by the length of the embedded message. At the fifth stage, we consistently change the standard space characters based on the numeric encoding of the letter with UniSpace characters. And the last sixth stage puts a label with the code 26 at the end of the secret message in the Word document, and the file is saved and the process ends. To implement this idea, a software application in the VBA programming language has been developed.

To extract data, this process is repeated. Namely, to begin with, we find the numeric code 26 between the words, and then each space is analyzed by the value of the sequence of space codes from the Unicode table (see table 4). the Process will stop if the numeric code 26 is encountered.

EXISTING APPROACH

The proposed method was implemented using software developed by the authors. At the same time, various documents from the Microsoft Word series were used as a container. The built-in VBA programming language was chosen as the programming language. We will demonstrate the program using the following example (see figure 1) [22].

If you hide the word “Nazokat” in this text, for example, we will get the following result after executing the program

FIGURE 1. Source text, empty container.

(see figure 2).

FIGURE 2. Stegotext with the secret word "Nazokat".

To understand how the program works, in figure 2, after the word “has”, three UNISPACE characters are additionally shown alternating. Comparing figures 1 and 2, we can conclude that these two texts are quite difficult to distinguish visually. In principle, this text is very difficult to distinguish from the original. Visually, the text of the stegacontainer does not differ from the original, i.e. an untrained reader will most likely not be able to detect the presence of hidden information in the text being read. When we reverse read the secret message from this text, we get the word "NAZOKAT". Please note that the response contains only uppercase letters, although the input was both uppercase and lowercase. This is because the table 4 letters of the Latin alphabet are encoded only as uppercase. In General, this method does not have a limit on the volume of a secret message being implemented. However, this algorithm, as well as many text steganography algorithms, has a weakness for changing the text format, which can make the text useless.

CONCLUSION

Modern steganography deals with information in electronic form, not with physical objects. And so, due to the rapid development of digital technologies, steganography has received a strong impetus for development. The reason for this situation is the following: Embedding and extracting can be automated since computers can process data efficiently. In General, secret information can be hidden almost anywhere, and some container objects are more suitable for hiding information than others.

Here is a steganography scheme in an MS Word document based on embedding invisible Space characters from a set of Unicode codes. Since the Space symbol has the highest frequency in the text, we can conclude that the amount of embedded information is limited only by the number of this symbol in the text. The proposed steganography algorithm includes both the embedding and extraction process. In this case, each character of embedded data is hidden in the cover file without any noticeable degradation of the cover file itself. Also, this approach works with ASCII character values, not with their binary value. In General, an MS Word document has a diverse set of attributes that can be used in steganography. This includes the attributes of the text itself, which are successfully used in MS Word and for which many scientists have studied the possibility of hiding data [4].

Thus, digital steganographic methods that use the features of information representation in computer files are a promising direction of applied science. These methods can be applied in applications such as copyright protection, prevention of forgery of electronic documents, the transmission of secret messages, etc.