ISSN: 0067-2904

Text-based Steganography using Huffman Compression and AES Encryption Algorithm

Rawaa Hamza Ali*[,2] , Jamal Mohamed Kadhim[1]
1Computer Department, College of Science, Al-Nahrain University, Baghdad, Iraq
2Department of Biology, Collage of Science, University of Misan, Maysan, Iraq

Received:2/11/2020 Accepted: 9/1/2021

Abstract
In every system of security, to keep important data confidential, we need a high degree of protection. Steganography can be defined as a way of sending confidential texts through a secure medium of communications as well as protecting the information during the process of transmission. Steganography is a technology that is used to protect users' security and privacy. Communication is majorly achieved using a network through SMS, e-mail, and so on. The presented work suggested a technology of text hiding for protecting secret texts with Unicode characters. The similarities of glyphs provided invisibility and increased the hiding capacity. In conclusion, the proposed method succeeded in securing confidential data and achieving high payload capacity by using the Huffman compression algorithm, which was implemented on an unlimited text length. In addition, this approach has the ability to hide a single bit in every digit or letter in the cover file. Also, the approach meets the cognitive transparency and does not make the modifications apparent on the original data. The method suggested in this work increases the security level through coding a secret message before embedding it within the cover text, with the use of the Advanced Encryption Standard (AES) algorithm.

Keywords: Text Steganography, Unicode Characters, AES, Huffman compression algorithms.

إخفاء المعلومات المستندة إلى النص باستخدام ضغط هوفمان وخوارزمية تشفير AES

رواء حمزه علي*[2] ، جمال محمد كاظم[1]
قسم الحاسبات، كلية العلوم، جامعة النهرين ، بغداد، العراق
2 قسم علوم الحياة , كلية العلوم , جامعة ميسان ,ميسان, العراق

الخلاصه
اي كل نظام أمان، من أجل الحفاظ على سرية البيانات المهمة ، نحتاج إلى درجة عالية من الحماية. يمكن تعريف Steganography كطريقة لإرسال نصوص سرية من خلال وسيلة اتصالات آمنة وكذلك حماية المعلومات أثناء عملية الإرسال. إخفاء المعلومات هي تقنية تُستخدم لحماية أمن وخصوصية المستخدمين. يتم الاتصال على نطاق واسع باستخدام الشبكة عبر البريد الإلكتروني والرسائل القصيرة وما إلى ذلك ، في هذه المقالة، تم اقتراح تقنية إخفاء النص لحماية النص السري بأحرف .Unicode أعطت أوجه التشابه بين الحروف الرسومية * الاختفاء وزيادة قدرة الاختباء. الاستنتاجات نجحت الطريقة المقترحة في تأمين البيانات السرية وتحقيق سعة حمولة عالية باستخدام خوارزميات ضغط هوفمان ، وطول النص غير

* Email: rawaaha@uomisan.edu.iq

محدود. بالإضافة إلى ذلك ، يمكن لهذه الطريقة إخفاء بت واحد في كل حرف أو رقم في ملف الغلاف ، وهي تلبي الشفافية المعرفية ولا تجعل أي تغييرات واضحة على الأصل. في طريقتنا ، نقوم بزيادة مستوى الأمان عن طريق تشفير الرسالة السرية قبل تضمينها في نص الغلاف باستخدام خوارزمية معيار التشفير المتقدم القياسية (AES).

1. Introduction

The importance of information hiding for securing communications has been increasingly growing because of the developments in communications, web, and information technology. Although confidential data is encrypted, it must be hidden with the use of some techniques. The goal is securing the confidential information which might be related to commercial or official documents related to government or military messages, which should be connected to the Internet with high security, even with the existence of hackers. The Internet is the most applicable and relevant way to communicate between individuals, but confidential information are prone to several threats. The Internet is not guaranteed from start to finish while communicating over the network [1]. Secret messages are covered in unclear media with the use of steganography. The major aim of hiding information is preventing individuals from realizing the existence of hidden information. Generally, there are various types of cover media, such as video, image, text, and sound. The term steganography is derived from two words in the Greek language, the first word is steganós (concealed or covered), while the second word is graphia (writing). Generally, there are 2 major processes to hide information. Initially, secret data might be hidden in cover media via a process of embedding. Secondly, secret bits might be recovered from stego text via extractions [2].

The naked eye won't be able to notice the secret data that is hidden within the cover media. Therefore, the existence of a secrete message might not be spotted via unsuspicious users. The major aim of steganography is protecting secret data from unauthorized use, with no disturbance to cover media. It includes the two components of secret information and cover text, as shown in equation (1) [1].

Stego object= cover object + secret data …….(1)

The rest of the paper is organized as follows: section 2 presents a brief explanation about the type of steganography, section 3 presents Related Work, section 4 presents an explanation Unicode stander, section 5 presents Advanced Encryption Standard (AES) , section 6 presents a brief explanation about Huffman Encoding algorithm, Section 7 presents the proposed steganography method, section 7.1 describe the Unicode table method, section 7.2 describes the embedding phase, section 7.3 presents the proposed algorithm for embedding method , section 7.4 presents the extraction phase and algorithm for extraction , section 8 discusses the result of the proposed method, and finally section 9 states the main conclusions.

2.Steganography Types

On the basis of the cover media utilized to hide secret texts, steganography has many types. The steganography type is decided via the many utilized file formats [3]. Steganography types include Network or Protocol Steganography, Image Steganography, Audio Steganography, Text Steganography, and Video Steganography.

-Image Steganography

Hiding secret messages in image files is indicated as hiding information for pictures as well as hiding data via taking a cover object. Also, hiding image has been referred to as hiding information in a image. However, such type has some drawbacks, like the inability of including a lot of data in the image, since it might cause distortion and raise suspicions that is containing data.[4] .The majorly utilized techniques for encoding are the Most significant bit insertion (MSB) and the Least significant bit insertion (LSB) [4].

-Audio Steganography

Masking audio information is the approach used to hide secret information in a sound. Also, it is very robust, yet with limitations on the amount of data that can be hidden. MP3, WAV, AU are the audio files used to hide data in this approach. There are many approaches to hide the sound, such as Phase Coding, Encoding, Low Bit, and Spread Spectrum [4].

-Video Steganography

In the condition of hiding confidential information in a video, the video (set of images) is utilized as a carrier for hiding the data. There are two types of methods used for masking the information in spatial and frequency domain technology [5]. These methods are the Transform Domain-Based Method and the Spatial Domain-Based Method.

-Protocol or Network Steganography

Hide To hide network information , a modified single network protocol is used. It includes hiding information via applying one of the network protocols, such as IP, UDP, TCP, ICMP, PDU, and so on, as a shell object. It is considered as one of the robust and safe approaches [6].

-Text file Steganography

The concealment of information inside text is the most significant among the other steganography methods. Confidential data is hidden in a text file. In such an approach, the secret data is hidden within each letter in each word in a text message. Less memory is required by hiding text information, as it has the ability to store text files only [7, 8]. The mainly utilized approaches in such a scheme include the Format-based steganography (Word-shift code, Feature code, Line-shift code), Linguistic steganography (syntactic, semantic), and Random and Statistics Techniques [9, 10].

3.Related work

In this section we will investigate some of steganograpgy methods that are related to our work.

Shakir and Abdulameer [11] suggested that a new font with special properties be developed for data hiding purposes. Furthermore, for the embedding process, the set of high-frequency letters called SHFL in the English language was chosen based on making the same glyphs for the multiple codes. The hiding method replaces another code with the English symbol code that has exactly the same glyph. Two bits are hidden at once using 00, 01, 10, and 11. This technique suffers from increasing stego-text since there are also several requirements for embedding the Unicode characters with 2 bytes to replace the original glyphs with new ones.

Aman, *et al.*[12] introduced a new hybrid system incorporating the methods of Unispach and ZeroWidth Characters based on format-based open-spaces. The proposed solution outweighs the disadvantages of the current approaches. For example, it has a high hiding ability by using the loss-less compression algorithm and by using any variant of MS Word file as a stego carrier to hide 4 bits per room. The robustness is highly improved by adding multi-layers of security and SHA-1 algorithm. They ensureed that their novel method is best suitable for large messages.

Sivabalan et al. [13] developed an emoticon-based text steganography scheme called EM_ST. It generates a random text as a cover message consisting of some words. In addition, it transforms all the characters of the hidden message into emoticons based on a basic pattern and thus embeds the emoticons via the cover text between words. In reality, this method has high capacity and clear transparency (low invisibility), but suffers from low robustness against attacks.

Kumar *et al*. [14] described a text steganography scheme called 4&3SpaCh, which expanded the UniSpaCh by using the Unicode characters effectively. By considering the embeddable positions, including inter-sentence, inter-letter, end-of-line and inter-paragraph spaces, this scheme masks the SMbits in the MS Word file. The researchers used two separate patterns in the cover text to mark the hidden message bits. However, as opposed to the UniSpaCh, this

device provides high imperceptibility and greater embedding ability. However, through the stego text, it generates some unconventional gaps between terms, causing increased visual attacks.

Odeh *et al*. [15] used ZWCs(Zero width character) to conceal hidden message bits within an MS Word format. They suggested a new text-masking algorithm called ZW-4B. Four ZWCs are used by this algorithm to differentiate four hidden message bits between characters in the cover message format. For example, after a letter through the cover text, the algorithm inserts all four ZWCs, then represents the secret code '0001'; if it contains three ZWCs, it determines '0001', and so on. This technology promises high latency and higher embedding capacity in realistic terms and can be implemented in multilingual texts. However, because the embeddable position is only between characters, it suffers from reduced durability. Moreover, the embedded bits can be preserved against structural attacks by this approach.

Por *et al*. [16]presented a technique (UniSpaCh) for hiding text-based data. Every 2 bits (11, 00, 10, 01) are replaced with a special space. It includes spaces that have been created in special locations within the cover text. This technique provides a high degree of invisibility, but it also has low embedding capacity and cannot be applied to include long -secret bits in the cover text.

Stojanov *et al*.[17] described the format-based methods of text steganography in Ms-Word files that belong to a proposed system for hiding data that is called Property coding. Any text formatting that is invisible to the human eye is used. Property coding takes advantage of the properties of various document objects, such as the scale or underlines of characters and the border of paragraphs for embedding details. Higher capacity and visibility perfect the strategy. However, it has low robustness and, according to experimental results, it also imposes a slight overhead of around 1% on the file size.

4.UNICODE Standard

Before Unicode, computers majorly only handling with numbers. They are storing characters and letters via assigning numbers to them. Prior to the introduction of the Unicode standard, there have been various systems, referred to as character encoding, used to assign such numbers. Earlier character encodings have limitations and they are not covering the characters in all languages. Even for one language such as English, there is no single encoding which is covering all punctuation marks, letters, and the mainly used technical symbols [18].

The Unicode standard was introduced for addressing such problems. Also, it was developed on a code which is adequately large for supporting the writing systems utilized by all the languages in the world. The many languages seen on the web recently are due to the Unicode character support, that allows computers to support the majority of languages in the world. It allows programmers and users to develop content in the native language. Also, it is capable of maintaining a million characters and it might be conducted by many numbers, symbols, and characters to be encoded. The most utilized encodings are UTF8 and UTF16. In addition, UTF8 is considered to be similar to the ASCII coding, which applies the same 8bit code. UCS2 or UTF16 uses 2 bytes for each one of the characters. The Unicode characters have been notable through code points, that were represented predictably by the letter U succeeded by 4 hexadecimal digits [10]

5.Advanced Encryption Standard (AES)

Rijndael algorithm is defined by this standard, which is considered as one of the symmetric block ciphers handling data blocks of 128bit, with the use of cipher keys which are 128, 192, and 256 bits long. Rijndael has been developed for handling additional block sizes as well as key lengths; yet, it wasn’t used in such standard. With regard to the remaining sections of this work, the algorithm is going to be indicated as "AES Algorithm". Such algorithm might be utilized with 3 distinctive key lengths mentioned earlier; thus, this high number of "flavors" might be indicated as "AES128", "AES192" and "AES256". Yet, AES indicates that that the

algorithm might just accept a 128-bit block size In addition, it is represented by Nb = 4, which reflects a 32bit word count (number of columns) in a case, while the length of the key for AES is 128, 192, or 256bits. Furthermore, the length of the key might be specified by Nk = 4, 6, or 8, reflecting 32bit word count (number of columns) in encryption key. Also, the number of rounds taken throughout the AES algorithm is based on the size of the key. The number of rounds is denoted as Nr, in which Nr = 10 when Nk = 4, Nr = 12 when Nk = 6, and Nr = 14 when Nk = 8. The only groups met for such parameters were given key block groups.[19] . AES structure is provided in Figure 1 [20].

Table 1-Combinations Key-Block-Round [21].

Key size (Nk words)	**Plaintext Block Size** (Nb word)	**Number of round**
128 bit (4)	128 bit	10
192 bit (6)	128bit	12
256 bit (8)	128 bit	14

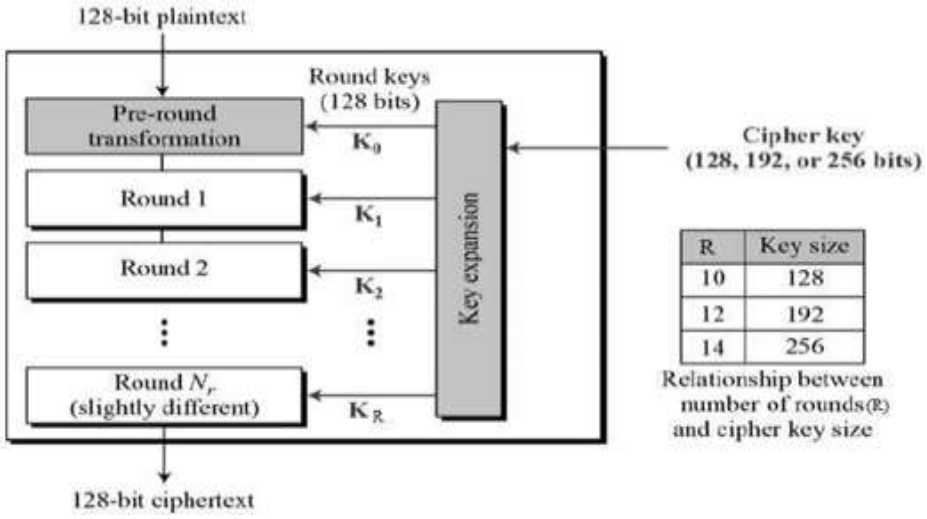

Figure 1-AES Structure.[20]

6.Huffman Encoding algorithm

This algorithm is very commonly utilized for data compression in computer science area.
This algorithm has been developed by David Huffman and utilized for the reduction of the frequency of coding with no loss in the quality of the data. The utilization of the frequency of the data is the fundamental Huffman encryption concept. The approach in the algorithm assigns symbols from the alphabet while changing the coded words according to their frequency. Higher frequency codes are indicated with shorter codes for a higher compression result [22].

7.The Proposed Method

In the present paper, an improved method was presented for text steganography in English scripts using Unicode characters. The aim is to ensure the secrecy of a message that has been sent over an insecure channel. The method is divided into two stages, namely the inclusion and extraction stage. The whole process of the proposed model consists of two main

successive steps; the first step is the coding stage, where a standard encryption algorithm is used to convert the secret message into a naturally encrypted file in the form of binary data for maximum security. The proposed method uses the AES encryption algorithm to encrypt the secret message which will be sent over an insecure channel. The second step is the use of a Unicode character for securing the secret message transmission. Glyph was considered as one of the hidden forms which might have letters in the English alphabet that should be matching the original letters The suggested method uses all uppercase, lowercase, and special characters. The secret message is hidden by the sender through utilizing the cover text for securing data, while the output that is related to the embedding algorithm is referred to as the stego-text. UTF8 or ASCII encoded text indicates each one of the characters as bytes' string. The information should be serialized; thus, it might be included slowly in the cover. Furthermore, the private message shouldn't be longer than the cover. Data that includes a text message encoded with ASCII values for characters may be included. Then, finally, after the embedding of the cover text, it will be compressed with the use of lossless data compression algorithms to the point where a large amount of the file has been included using the Huffman algorithm. Figure 4 represents the flow chart of the proposed method.

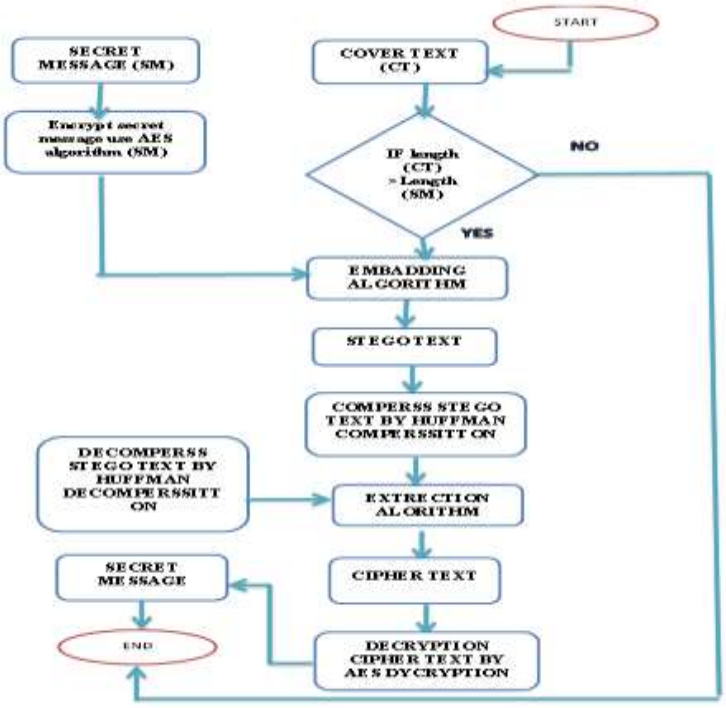

Figure 2-Flow chart of the proposed method for steganography

7.1Unicode table

Unicode is a modern text standard that describes each symbol and letter utilized in digital and printed media today. Also, Unicode has grown to the highest level of character selection in the text in almost any language; the range starts from U + 0000 to U + 1F9FF and involves all types of supported languages. However, these algorithms use certain values to encode and decrypt confidential data. The ranges which have been utilized in this procedure are U + 0041 to U + FF21[23].

7.2Embedding Phase

After encrypting the secret message using the AES 128-bit key, and in the case where the secret message was hidden within the cover text, the secret message will be converted to binary values and every one of the letters includes 8-bits. Such binary value was replaced in

the cover text utilizing approaches for hiding the text. Each bit in each letter in the cover text was set. Besides, the progress of Unicode occurs at this point. Unicode is considered as a 2 way process; U + 0041 or U + FF21, if the binary value "1" the Unicode character of the cover text was changed to the value U + FF21. In the case when binary value is "0" there is no change in cover text. It includes a list of the Unicode tables for all letters – lowercase and uppercase, numbers, and special characters. It is a predefined value with regard to the computer system. U + FF21 and U + 0041 were comparable to the original English alphabet. After completing the embedding process, the secret message within the cover text using the suggested modification method we compress the cover text (stego text)using one Huffman compression algorithms.

7.3Proposed algorithm for embedding the method

Input: a cover message (CM), a secret message (SM), a secret key (K)

Output: a carrier text message or stego- which consists of cover message (CM) and hidden message (HM).

1-SM← Secret Message

2. CM← Cover Message

3. K← Secret Key

4- Encrypted _SM← Encrypts the SMbits based on K using AES;

for each $C_i \in SM = \{C_1, C_2, ..., C_n\}$ do

Reading a character of SM

count←count the character of SM.

5- Converting the secret message to ASCII bits (8 bits) [b] equivalent. $b = (n \times 8) <= l$.

6- Reading a message bit and a letter from the cover.

7- For every bit in the secret message

if bit = 0, then no change in the cover text (ASCII code),

else bit =1 replace with Unicode which is equivalent from the Unicode (Table 2) with the use of glyphs.

8- Repeat **step5**, to the point where all binary bits have been embedded.

9- Use Huffman compression algorithm on the file after embedding.

10- **Return** the stego-text.

11- **End.**

7.4Extraction phase

After completing the process of hiding the secret message inside the cover text, the next step is to apply the extraction algorithm via the message recipient. It reverses the steps of the embedding algorithm.

Proposed algorithm For Extraction

Input: carrier message, a secret key (K).

Output: a secret message (SM).

1- Apply Huffman algorithm to decompress stego text.

2- HS← Discovers the existing hidden marks/symbols from CM open stego text.

3- For each $C_i \in H S = \{C_1, C_2, ..., C_n\}$ do

Reading a character of words (n) from stego file.

4- l = the word length,

For every one of the bits in stego –text

$b = (n \times 8) <= l$,

5- Check the selected characters' code.

If the code is lowercase or uppercase English Latin letters

then the secret message = 0

else the secret message =1 based upon Table 2.

6- Repeat step5, to the point where all the binary bit values have been obtained.
7- SM← performs the decryption of the Encrypted _SM based on K with the use of the corresponding decryption function (AES);
8- **Return** the SM.
9- **End.**

Table 2-Unicode Table for steganography process

Original Alphabet	Unicode	Original Alphabet	Unicode
A	U+FF21	a	U+FF41
B	U+FF22	b	U+FF42
C	U+FF23	c	U+FF43
D	U+FF24	d	U+FF44
E	U+FF25	e	U+FF45
F	U+FF26	f	U+FF46
G	U+FF27	g	U+FF47
H	U+FF28	h	U+FF48
I	U+FF29	i	U+FF49
J	U+FF2A	j	U+FF4A
K	U+FF2B	k	U+FF4B
L	U+FF2C	l	U+FF4C
M	U+FF2D	m	U+FF4D
N	U+FF2E	n	U+FF4E
O	U+FF2F	o	U+FF4F
P	U+FF30	p	U+FF50
Q	U+FF31	q	U+FF51
R	U+FF32	r	U+FF52
S	U+FF33	s	U+FF53
T	U+FF34	t	U+FF54
U	U+FF35	u	U+FF55
V	U+FF36	v	U+FF56
W	U+FF37	w	U+FF57
X	U+FF38	x	U+FF58
Y	U+FF39	y	U+FF59
Z	U+FF3A	z	U+FF5A

8.Results of the proposed method

The method of steganography includes two pieces of data, namely the cover text and the confidential data which should be hidden. The method to embed information is different in

purpose and complexity. In our method, the information in the text documents can be hidden using Unicode characters. This method will include one bit in each English letter or number in the cover text after encrypting the secret message using the AES algorithm and compressing the result (Stego text) using the Huffman algorithm. When comparing our method with the previous methods, see Table 4, we notice that our method uses all letters and numbers in the embedding process. Therefore the memory capacity used is very high, so that each symbol takes 8 bits of memory, while Huffman's algorithm is used to reduce the embedding capacity. Our method uses all English letters in the embedding process, and by using the Huffman compression algorithm to reduce the hiding capacity, we obtained a high compression ratio, as shown in Table 3. The capacitance ratio is calculated using equation (2).

Capacity ratio = (Stego text size) / (Stego text size after compression) ……..(2)

The proposed method for text-based steganography achieved important characteristics. For example, to reduce the embedding capacity, Huffman compression was used. As a result of using Unicode characters, the stego text is similar to the original text. Also, it does not change the appearance of the text and does not require a certain font. Hence, the robustness is high because it is difficult to be detected by attackers, unless they know the type of method is used to hide the message or the type of Unicode block used. Our method is also characterized by a high degree of security, which was increased by encryption, and then the message was included with the confidential message in the cover text.

Table 3-Compression ratio for stego text

Stego text	Stego text after compression	Compression ratio
836	59	92.94%
2508	61	97.57000000000001%
1670	60	96.41%
2904	63	97.83%

Table 4-Comparison among steganography algorithms

Algorithm	Capacity	Invisibility	Robustness
Proposed method	High	Invisible	High
SHFL [11]	Low	Invisible	High
Unispach and ZeroWidth [12]	High	Invisible	High
EM_ST [13]	High	Visible	Low
4&3SpaCh [14]	High	Invisible	Medium
ZW-4B [15]	Low	Invisible	Medium
UniSpaCh [16]	Low	Invisible	Medium
Stojanov [17]	High	Invisible	Low

Example of the proposed method

Cover text:- (You can fool all of the people some of the time, and some of the people all of the time, but you can't fool all of the people all of the time.)

Secret message:-(hello world)

The secret message was converted to a binary bit for embedding, as follows:

(01101000 01100101 01101100 01101100 01101111 00100000 01110111 01101111 01110010 01101100 01100100).

Every binary bit of the secret message was embedded in every character of the cover text. So that, if the value of the bit is" 0 ", there is no change in cover text, and if the value of the bit is

"1", we change the UNICODE character for the cover text and thus include the secret message in the cover text, as follows (See Table 5):

Y	O	u	c	A	n	F	o	o	l	.	.	t	i	m	e
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓			↓	↓	↓	↓
0	1	1	0	1	0	0	0	0	1			0	1	0	0

Finally, the result is the stego text, as follows:

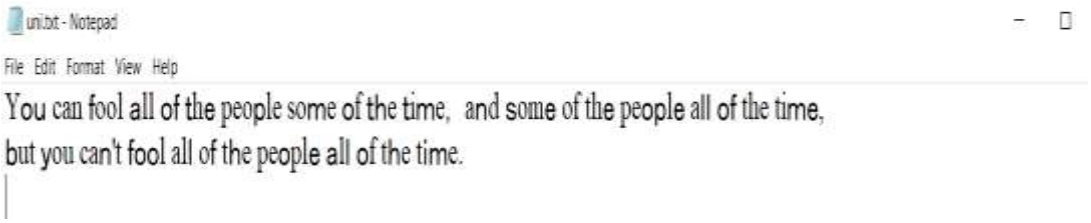

Table4-Example of the proposed text steganography method **.**

Binary Value	Unicode	Original Alphabet	Substituted Alphabet
0	U+0059	Y	Y
1	U+ff4f	o	ｏ
1	U+ff55	u	ｕ
0	U+0063	c	C
1	U+ff41	a	ａ
0	U+006e	n	N
0	U+0066	f	F

9. Conclusions

Steganography can be defined as a process that is used to secure a secret by hiding, thus its presence might not be noticed. In the suggested approach, the main aims of text hiding are security, availability, confidentiality, and integrity. To test the results of the Unicode process. This technology helps to hide the maximum number of the characters, increase performance, and achieve high payload capacity. Thus, it might not be simply extracted by an unauthorized party. In this method, data is hidden in the text documents through including letters or numbers using Unicode characters. The main goal in designing this method is the cognitive transparency. This method is characterized by excellent cognitive transparency, since the stego-text which is seen by the user is considered to be the same as the original. Thus, the capability of hiding achieved by this suggested approach was extremely high. This method is powerful in digitizing the text, indicating that copying and pasting text between the computer programs keeps information hidden. There is no increase in the size of the stego text in this method, due to the use of Huffman compression algorithm to reduce the size. Therefore, the proposed method provides an effective means for masking texts with Unicode and an excellent way to obtain a secure transfer of information.

References

[1] Ditta, A., et al., "Information hiding: Arabic text steganography by using Unicode characters to hide secret data". *International Journal of Electronic Security and Digital Forensics*, vol. 10, no. 1, pp. 61-78, 2018.

[2] Ahvanooey, M.T., et al., "AITSteg: An innovative text steganography technique for hidden transmission of text message via social media". *IEEE Access*, vol. 6, pp. 65981-65995, 2018.

[3] [3] Krishnan, R.B., P.K. Thandra, and M.S. Baba. An overview of text steganography. in 2017 Fourth International Conference on Signal Processing, Communication and Networking (ICSCN). 2017. IEEE.

[4] Arya, A. and S. Soni, "A literature review on various recent steganography techniques". *International Journal on Future Revolution in Computer Science & Communication Engineering*, vol. 4, no. 1, pp. 143-149, 2018.

[5] Gupta, H., et al., "Review On Various Techniques Of Video Steganography". *Journal Of Scientific And Technical Advancements*, vol. 4, no. 1, pp. 161-164, 2018.

[6] Bedi, P. and A. Dua, "Network Steganography using the Overflow Field of Timestamp Option in an IPv4 Packet". *Procedia Computer Science*, vol. 171, pp. 1810-1818, 2020.

[7] Sharma, S., et al. Analysis of different text steganography techniques: a survey. in 2016 Second International Conference on Computational Intelligence & Communication Technology (CICT). 2016. IEEE.

[8] Malalla, S. and F.R. Shareef, "Improving Hiding Security of Arabic Text Steganography by Hybrid AES Cryptography and Text Steganography". *Journal of Engineering Research and Application*, vol. 6, no. 6, pp. 60-69, 2016.

[9] Al-Nofaie, S.M.A. and A.A.-A. Gutub, "Utilizing pseudo-spaces to improve Arabic text steganography for multimedia data communications". Multimedia Tools and Applications, vol. 79, no. 1-2, pp. 19-67, 2020.

[10] Taleby Ahvanooey, M., et al., "Modern text hiding, text steganalysis, and applications: a comparative analysis". *Entropy*, vol. 21, no. 4, pp. 355, 2019.

[11] Baawi, S.S. and D.A. "Nasrawi. Improvement of "Text Steganography Based on Unicode of Characters in Multilingual" by Custom Font with Special Properties". in I*OP Conference Series: Materials Science and Engineering*. 2020. IOP Publishing.

[12] Aman, M., et al., "A hybrid text steganography approach utilizing Unicode space characters and zero-width character". *International Journal on Information Technologies and Security*, vol. 9, no. 1, pp. 85-100, 2017.

[13] Patiburn, S.A., V. Iranmanesh, and P.L. Teh, "Text steganography using daily emotions monitoring". *International Journal of Education and Management Engineering*, vol. 7, no. 3, pp. 1, 2017.

[14] Kumar, R., S. Chand, and S. Singh, "An efficient text steganography scheme using Unicode Space Characters". *Int. J. Forensic Comput. Sci*, vol. 10, no. 1, pp. 8-14, 2015.

[15] Odeh, A., K. Elleithy, and M. Faezipour. Steganography in text by using MS word symbols. in Proceedings of the 2014 Zone 1 Conference of the American Society for Engineering Education. 2014. IEEE.

[16] Por, L.Y., K. Wong, and K.O. Chee, "UniSpaCh: A text-based data hiding method using Unicode space characters". *Journal of Systems and Software*, vol. 85, no. 5, pp. 1075-1082, 2012.

[17] Stojanov, I., A. Mileva, and I. Stojanovic, "A new property coding in text steganography of Microsoft Word documents". 2014.

[18] Moran, S. and M. Cysouw, *The Unicode cookbook for linguists. 2018: Language Science Press.*

[19] Al-Mamun, A., et al., "Security analysis of AES and enhancing its security by modifying S-box with an additional byte". *International Journal of Computer Networks & Communications* (IJCNC), 2017. vol. 9, no. 2, 2017.

[20] Abdelrahman, A.A., M.M. Fouad, and H. Dahshan, "Analysis on the AES implementation with various granularities on different GPU architectures". *Advances in Electrical and Electronic Engineering*, vol. 15, no. 3, pp. 526-535, 2017.

[21] AL-Mozani, A.S.S. and W.A.J. Awadh, "A new text steganography method by using non-printing unicode characters and unicode system characteristics in English/Arabic documents". *Journal of Thi-Qar Science*, vol. 3, no. 3, pp. 192-200, 2012.

[22] Erdal, E. and A. Ergüzen, "An Efficient Encoding Algorithm Using Local Path on Huffman Encoding Algorithm for Compression". *Applied Sciences*, vol. 9, no. 4, pp. 782, 2019.

[23] https://unicode-table.com/en/, Unicod*e Character Table, 2012–2021*. 2012–2021.