ISSN: 0067-2904

Proposed Hiding Text in Text Based On RNA for Encoding Secret Information

Suhad M. Kadhem, Dhurgham W. Mohammed Ali*
Department of Computer Science, University of Technology, Baghdad, Iraq.

Abstract

In any security system, we need a high level of security, to maintain the secrecy of important data. Steganography is one of the security systems that are hiding secret information within a certain cover (video, image, sound, text), so that the adversary does not suspect the existence of such confidential information.

In our proposed work will hide secret messages (Arabic or English) text in the Arabic cover text, we employed the RNA as a tool for encoding the secret information and used non-printed characters to hide these codes. Each character (English or Arabic) is represented by using only six bits based on secret tables this operation has provided a good compression since each Arabic character needs 16 bits and each English character needs 8 bits in a conventional method, using RNA with secret table has provided a good degree of security, moreover using non-printed characters, these characters do not appear on screen so have provided a complete similarity between the Arabic cover text and stegotext.

Keywords: Security, Steganography, RNA, Codon, Coding.

نظام مقترح لأخفاء نص في نص بالأعتماد على أر أن أي لترميز المعلومات السرية

سهاد مال الله كاظم، ضرغام وميض محمد علي*
قسم علوم الحاسوب، الجامعة التكنولوجية، بغداد، العراق.

الخلاصة

في اي نظام امني نحتاج الى مستوى عالي من الامنية وذالك للحفاظ على سرية البيانات المهمة. اخفاء البيانات هوفرع من فروع الانظمة الامنية التي تقوم باخفاء المعلومات السرية داخل غطاء معين (فيديو، صورة، صوت، نص) بحيث ان الخصم لايشك في وجود هذه المعلومات السرية.

في عملنا المقترح سوف نخفي الرسالة السرية (عربي او انكليزي) في نص الغطاء العربي، وأضفنا ال **أر أن أي** كأداة لترميز المعلومات السرية واستخدمنا الاحرف غير المطبوعة لاخفاء هذه الرموز.كل حرف (عربي او انكليزي) مثلا باستخدام 6 بت فقط بالاعتماد على جداول سرية للاحرف وهذه العملية زودتنا بضغط جيد لان كل حرف عربي يحتاج 16 بت وكل حرف انكليزي يحتاج 8 بت في الطرق التقليدية، وباستخدمنا ال **أر أن أي** مع الجدول السري زودتنا بدرجة عالية من الامنية علاوة على ذالك استخدمنا الاحرف غير المطبوعة وهذه الاحرف لاتظهر على الشاشة لذلك وفرنا تشابه بشكل كامل بين نص الغطاء العربي ونص الاخفاء.

Introduction

Steganography is a technique of concealing original text in such a way that no one else except the sender and the intended recipient can understand it from attack eyes. Cryptography and Steganography

*Email: dhurgham_w@yahoo.com

are techniqies to secure information transfer over the Internet. While Cryptography technique is used to protect the contents of the message, Steganography is used to conceal the existence of a message using cover text [1]. Data hiding is a significant side of communication and data security technology. Information hiding is an important method of steganographic communication. In other words, the important data to be sent should be embedded in the carrier, so that it cannot be easily found secret data. There are many carriers applied information hiding, such as text, image, audio and video [2]. Invisible ink is an example for Steganography where an original message is transferred between source and destination. Any attack in the middle can read the message without knowing about the hidden data, while, authorized persons can read it depending on substances features [3].

In this paper, has been proposed a steganography method, it uses RNA to add more security in coding step, it uses Unicode and non-printed characters in hiding step because most of these characters do not appear on the screen when written.

This rest of paper is organized as follows: section 2 presents a brief explanation about text steganography, section 3 presents an explanation about RNA Codon, section 4 presents an explanation about non printed characters, section 5 presents Unicode, section 6 presents Related Work, Section 7 presents the proposed steganography method, section 7.1 describe the sending stage of the proposed method, section 7.2 describe the recieving stage of the proposed method, section 7.3 presents implementation of the proposed method, section 7.4 presents discussion about our method section 7.5 presents the performance analysis and finally section 7.6 state the main conclusions.

Text Steganography

The text is one of the oldest media used in steganography. Secret messages were hidden in the texts of Letters, books and telegrams in earlier times, i.e. before the electronic age comes. Text steganography refers to the hiding of information within text, i.e. character-based messages. There are three basic categories of text steganography: format-based methods, random and statistical generation methods and linguistic methods [4].

Text steganography is the process of hiding text inside the text. Some of the main components in the system involved are shown in Figure1- [5].

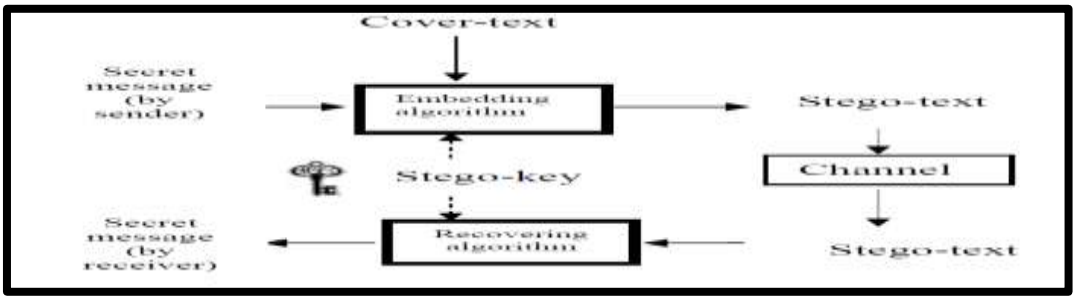

Figure 1-The mechanism of the text steganography

RNA Codons Generation [6]

For encoding a generated bits stream to RNA code, we have to encode each two bits in processing step of RNA code (A, U, G and C) by using Table-1 to get RNA strand. The stream of genetic- codes is splitted into three genetic –codes to be a (codon). Figure-2 illustrates splitting of the stream genetic code to triple codon.

Table 1- The RNA Genetic Code.

2 bits	The RNA genetic code
00	A
01	C
10	G
11	U

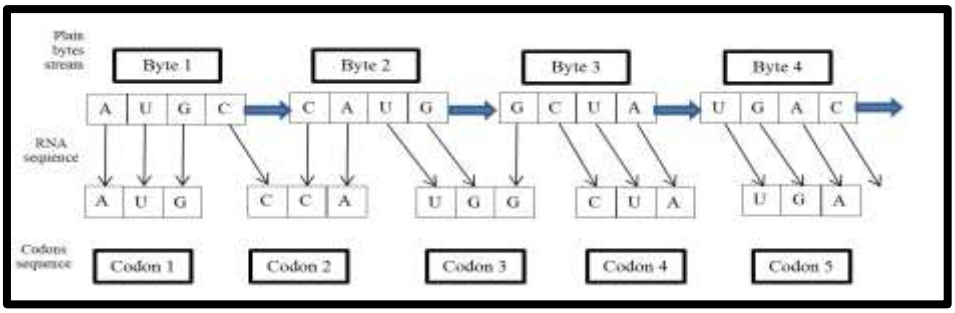

Figure 2-Splitting of stream genetic code to triple codon.

Random RNA Codon Table Generation [6]

The genetic code of RNA is made up of three elements called codons created from arrangement of three nucleotides (e.g. ACU, CAG…) since there are four bases (A, C, G, U) in three letter combinations so there are 64 possible codons (4^3 combinations),which give all amino acid. Table-2 explains all possible amino acids.

Table 2- RNA amino acid

AAA	AAC	AAG	….	UUC	UUG	UUU
000000	000001	000010	….	111101	111110	111111

Non Printed Characters

We have several letters which are not normally displayed on the screen. For example, there is a special character to indicate the end of a line or the end of a paragraph or null, and so on [7]. Table-3 illustrates these characters.

Table 3- Non Printed Characters

Dec	Hex	Character(Code)	Dec	Hex	Character(Code)
0	0	NULL	16	10	DATA LINK ESCAPE (DLE)
1	1	START OF HEADING (SOH)	17	11	DEVICE CONTROL 1 (DC1)
2	2	START OF TEXT (STX)	18	12	DEVICE CONTROL 2 (DC2)
3	3	END OF TEXT (ETX)	19	13	DEVICE CONTROL 3 (DC3)
4	4	END OF TRANSMISSION (EOT)	20	14	DEVICE CONTROL 4 (DC4)
5	5	END OF QUERY (ENQ)	21	15	NEGATIVE ACKNOWLEDGEMENT (NAK)
6	6	ACKNOWLEDGE (ACK)	22	16	SYNCHRONIZE (SYN)
7	7	BEEP (BEL)	23	17	END OF TRANSMISSION BLOCK (ETB)
8	8	BACKSPACE (BS)	24	18	CANCEL (CAN)
9	9	HORIZONTAL TAB (HT)	25	19	END OF MEDIUM (EM)
10	A	LINE FEED (LF)	26	1A	SUBSTITUTE (SUB)
11	B	VERTICAL TAB (VT)	27	1B	ESCAPE (ESC)
12	C	FF (FORM FEED)	28	1C	FILE SEPARATOR (FS) RIGHT ARROW
13	D	CR (CARRIAGE RETURN)	29	1D	GROUP SEPARATOR (GS) LEFT ARROW
14	E	SO (SHIFT OUT)	30	1E	RECORD SEPARATOR (RS) UP ARROW
15	F	SI (SHIFT IN)	31	1F	UNIT SEPARATOR (US) DOWN

Unicode System Standard

The Unicode Standard is a character coding system designed to support the worldwide interchange, processing, and display of the written texts of diverse languages and technical disciplines of the modern world. In addition, it supports classical and historical texts of many written languages. [8].

In our proposed method, two characters of the Unicode system are used to embed the secret information. These two characters are utilized in Arabic language where one of them, which has (157) ASCII, its job is to convert the Arabic letters from connected shape to isolated shape and the other character which has (158) ASCII, converts Arabic letters from isolated shape to connected shapes. These two Unicode did not make any change to the Arabic cover text but it gives an indication to the receiver. So if use the **157** ASCII will isolate the characters and **158** will connect the characters.

Related Works

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

a. Design Of Proposed Arabic Text Steganography Approach Using Non-Printed Character, 2016 [9].

This method used specific Arabic Unicode characters with non-printed characters to hide the secret information. Using specified Arabic Unicode characters with non-printed characters to hide secret information provided complete similarity between cover text and stego text since these characters don't appear when written. This complete similarity gave the ability to use the Arabic language features to provide information used as indications to determine secret keys between the sender and the receiver. Also B+ tree was used for dealing with the proposed database in order to provide speed and efficient access to the desired database contents.

b. Utilization of Two Diacritics for Arabic Text Steganography to Enhance Performance", 2015, [10].

An improved Arabic text steganography method was proposed. In this work it hides secret data into text cover media in two Diacritics that have been chosen based on highest availability percentage among all Diacritics, which are eight in Arabic language. This utilization of Diacritics- or Harakat - for security purposes is benefiting from the natural existence of diacritics as historical characteristics of Arabic language, originated to just represent vowel sounds. The paper exploited the possibility of hiding data in two Diacritics, i.e. Fathah and Kasrah. Adjusting the previously presented single (Fathah only) diacritic hiding scheme. The two Diacritics stego-work proposed in this work featured higher capacity and security showing interesting promising results.

c. Information Hiding in Arabic Text Using Natural Language processing Techniques", 2014 [11].

In this method, the natural language processing (NLP) for Arabic text techniques was utilized as a tool in order to increment the efficiency of the steganography. Each sentence in the cover text was parsed to be indications to the hiding method, so more than one hiding method was used in one text, and therefore the system complexity was increased. This method depended on the grammar of the Arabic language to choose the method of hiding; The B+ Tree was utilized to index the grammar in the lexicon.

d. A New Text Steganography Method By Using Non-Printing Unicode Characters", 2010 [12].

This is an approach for text steganography by using Unicode standard characters, (which have the non-printing properties) to encode the letters of English language and embeded the secret message letter by letter into the cover-text. This approach had high hiding capacity, it could embed (K+1) letters in a text with K characters and it did not make any apparent changes in the original text. So it satisfied perceptual transparency**.**

e. Proposed Arabic Text Steganography Method Based on New Coding Technique", 2016, [13]

In this paper we will use a new coding method such that its output will be contains sequence of ones with few zeros, so modified RLE that we proposed in this paper will be suitable for compression, finally we employ the modified RLE output for stenography purpose that based on Unicode and non-printed characters to hide the secret information in an Arabic text.

The Proposed Method

The proposed hiding information system consists of two stages: sending and receiving stages. Sending stage will take from the user as input two texts (the English or Arabic secret text and the Arabic text as the cover) and the output of this stage will be an Arabic stego text that will be used by the receiving stage to extract the original English or Arabic secret text.

Sending Stage

This stage consists of main steps as in Figure-3 and illustrates in algorithm **1**.

a. Algorithm (1):
Sending Stage
Input: Secret English or Arabic text (**T**), Arabic cover text (**T1**).
Output: Stego text (**S**).
Begin
Step1: Normalize **T** by converting all English characters
in to lower case.
/*normalization doesn't work if **T** is Arabic text*/
Step2: Call algorithm (**2**) that takes **T** as input and returns binary codes (**B**) and check bit **(CH).**
Step3: Call algorithm (**3**) that takes **(B)** and **(CH)** as input and returns binary code (**F**).
Step4: Call algorithm (**4**) that takes (**F)** and (**T1)** as input and returns stego text (**S**).
Step5: Return (**S**).
step6: End.

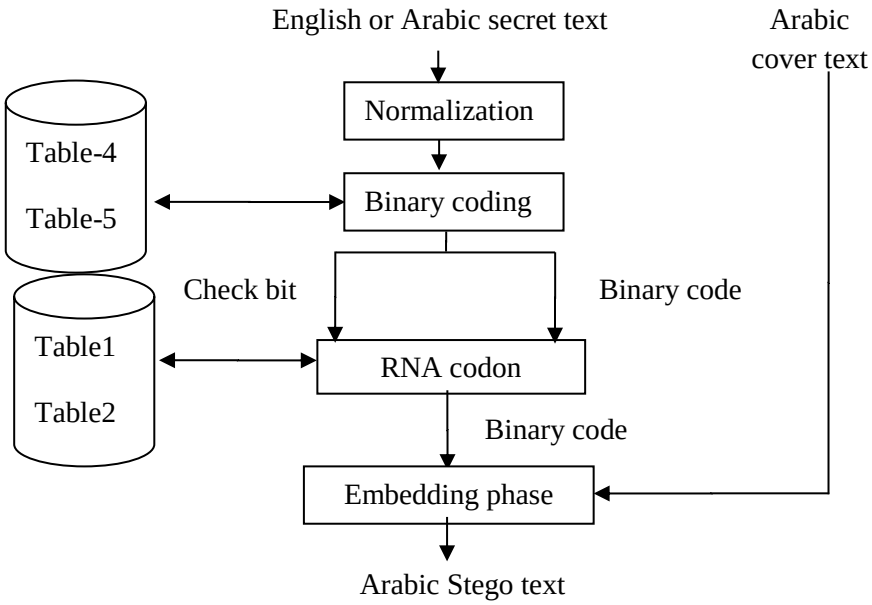

Figure 3- Illustrate sending stage

The normalization step converts the secret English text into lower case, the normalization doesn't work with Arabic character and special character like (@#$%). The output from normalization will be the input to the binary representation step that converts each character into binary codes, as shown in the reference [13].
In a binary coding, there are two databases, Table-4 for the Arabic characters and Table-5 for the English characters. If the secret message is an Arabic character, it will take one as a check bit and get the corresponding character on the secret message from Table-4 otherwise it will take zero and get the corresponding character in a secret message from Table-5. Algorithm 2 Illustrates the main steps in binary coding stage.

Table 4- DB for Arabic Characters

Arabic character	'ا'	'ب'	…	'،'	'ً'	….	'٩'
6 bits	000000	000001	….	101000	101001	….	111111

Table 5- DB for English Characters

English character	'a'	'b'	…	':'	'@'	….	'9'
6 bits	000000	000001	…	101000	101001	…..	111111

b. Algorithm (2):
Binary Coding
Input: English or Arabic text (**T**)**.**
Output: Binary code (**B**), check bit (**CH**)
Begin
Step1: B="".
Step2: If **T** is an Arabic text then
CH=1
Else
CH=0
Step3: while **T! ="" Do**
Begin
Cut character **(C)** from **T**
Get the corresponding code for **(C)** from **Table-4** or **Table-5** according to the **(CH).**
/*CH=1 Use **Table-4**
CH=0 Use **table5*/**
Add this code to **B.**
End //while
Step4: Return (**B** and **CH**).
Step5: End.

After that we apply RNA codon to the binary code resulted from the binary coding step to get a new complex code and a good security level by using Table**1** and Table**2**, (see algorithm **3**).

c. Algorithm (3):
RNA Coding
Input: Binary code (**B**), Check bit (**CH**)
/*binary code result from secret table of characters Table- (4 or 5) */
Output: Stream of binary (**F**) /*binary code result from RNA probability table 3*/
Begin
Step 1: **F**="".
Step 2: If (**B**) ≠ "".
Step 3: Use Table- **1** to convert each two bit to its corresponding RNA.
Step4: Use Table- **2** for mapping from RNA Codons (**three symbols**) strand to its corresponding binary to get (**CO**)**.**
Step5: Put **CH** in front of **CO** to get **F.**
Step 6: Return stream of binary (**F**).
Step 7: End**.**

The last stage is the embedding phase. If the binary code is one we will embed the non-printed character that has ASCII **2** with special character, and we will embed thenon-printed character that has ASCII **1** with diacritics. We will embed the non-printed character that has ASCII **3** with unpoint letter (without dot letter), and we will embed the non-printed character that has ASCII **5** with spaces between words, and we embed Unicode character with dot (dotted character) according to the character type (Unicode that has **158** ASCII for connected character and Unicode that has **157** ASCII for isolated character of Arabic cover text). And if the code is zero, we leave the cover character without any embedding. As an indication of the end binary codes we will embed the non-printed character that have ASCII **6**. Algorithm **4** illustrates these steps.
If we reach to end of the cover and we still have a binary code then we will add all the remaining non-printed characters to the end of stego text, also for each one of binary code we will add the non-printed character that have **4** ASCII code, and for each zero of binary code we will add the non-printed character that have **9** ASCII code.

d. Algorithm (4):

Embedding phase

Input: Binary code (**F**), Cover text **(T1).**

Output: Arabic stegotext Text **(S).**

Begin

Step1: initialization step:

S="".

D=length of **F**.

O=length of **T1.**

j=**k**=1.

Step2: while ((**j**<=**O)** and (**k** <= **D**)) do

/* While we do not reach the end of cover text and we still have binary code do */

Begin

If **k<=D** then /* if we still have a binary code */

Add **T1[j]** to **S**

If **F[k] =1** then

If **T1[j]** is unpoint letter then

Add the non-printed character that has **3** ASCII to **S.**

Else if **T1[j]** is diacritics (Haracat) then

Add the non-printed character that has **1** ASCII to **S**.

Else if **T1[j]** is special character then

Add the non-printed character that has **2** ASCII to **S.**

Else if **T1[j]** is a space between words then

Add the non-printed character that has **5** ASCII to **S.**

Else If **T[j]** is point letter then

Add Unicode character according to the letter type

If **T[j]** is connected character then

Add the Unicode character that has **157** or **158** ASCII to **S** According to the character type

/* Unicode **158** with connected character

Unicode **157** with isolated character***/**

k=k+1

j=j+1

End//while

End if

Step3: If (**j**>**O)** and (**K**<**D**) then

/*if we reach to end of cover text and we stil binary code */

While **k<=D** do

Begin

If **F[k] =1** then

Add the non-printed character that has **4** ASCII to **S**

Else

Add the non-printed character that has **9** ASCII to **S**

k=k+1

End

Else /* we don't reach to the cover text yet and the binary code is finished */

Begin

Add the non-printed character that has **6** ASCII to **S** /* the stop mark of binary code*/

While **j**<=**O** do

Begin

Add **T1[j]** to **S** /* add the remaining cover text to the stego text */

j=j+1

End//while

End

End//while

End
Step4: Return (**S**).
End.
Receiving Stage

This stage consists of main steps as in Figure- **4**. The input to this stage will be the Arabic stego text and the output will be the extracted original secret text. Algorithm **5** illustrates these steps.

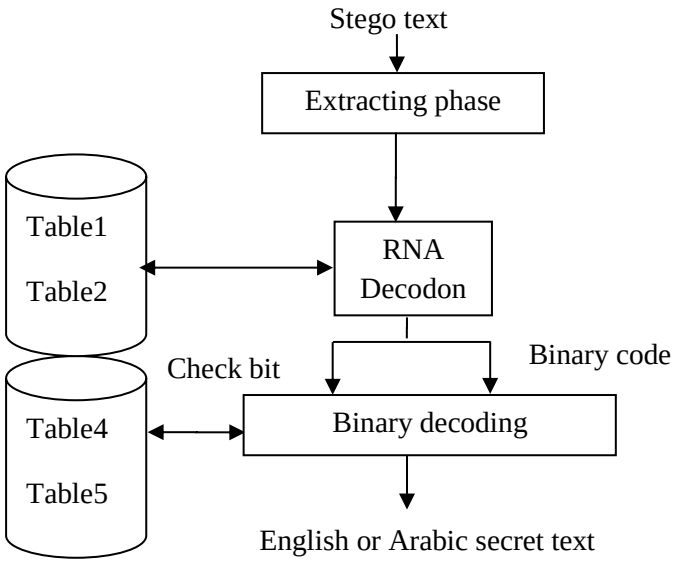

Figure 4-Illustrate receiving stage

a. Algorithm (5):
Receiving Stage
Input: Stego text (**S**).
Output: Secret English or Arabic text (**T**)
Begin
Step1: Call algorithm (**6**) that takes stegotext (**S**) as input and returns **F.**
Step2: Call algorithm (**7**) that takes Binary code **F** as input and returns stream of binary code **B** and a check bit **CH**.
Step3: Call algorithm (**8**) that takes binary codes (**B**) as input and returns **T** according to **CH**.
Step4: Return (**T**).
End.

The first stage is the extracting phase that extracts the binary code is illustrated in (algorithm **6**).

b. Algorithm (6):
Extracting phase
Input: Stego Text **(S).**
Output: Binary code (**F**)
Begin
Step1: initialization step
F=""
N=length of **S.**
i=1, **X**=1
Flag=false
Step2: while **i**<=**N** do
Begin
If **S[i]** =**6** then
Flag=true
Else
If **S[i]** =**4** then

F[X] = 1
Else
If **S[i]** =**9** then
F[X] = 0
Else
If **S[i]** is a member in **[1, 2, 3, 5]**
F[X] = 1
Else
If **S[i]** =158 or **S[i]** =157 then
F[X] = 1
If not flag then
Begin
F[X] = 0
Go to step 3
End
X=**X**+1
i=**i**+1
End //while
Step3: Convert (**F)** to string
Step4: Return (**F**).
End.

After that we will extract the check bits and decode RNA by applying (algorithm **7**). The input to this algorithm will be the binary code resulted from the previous extracting phase.

c. Algorithm (7):
RNA Decoding
Input: stream of binary (**F**)
Output: stream of binary (**B**), Check bit (**CH**).
Begin
Step 1: B= ""
Step 2: Cut one bit from the front of **F** to get **CH**, **CO**.
Step 3: Use **Table- 2** for mapping from **CO** to RNA codon (**three symbols**) to get RNA strand (**NN**).
Step 4: Use **Table- 1** for mapping from (**NN**) to its corresponding binary to get (**B**).
Step 5: Return (**B**, **CH**).
End.

The last step will be the binary decoding to retrieve the English or Arabic secret message (see algorithm **8**); the input to this algorithm will be the binary code that results from the previous step.

d. Algorithm (8):
Binary Decoding
Input: Binary code (**B),** Check bit (**CH**).
Output: English or Arabic text (**T**)
Begin
Step1: initialization
T = ""
Step2: While **B**≠ "" do
Begin
2.1: Cut 6-bits **C** from **B**.
2.2 If **CH** =1 then
2.2.1 Get the corresponding character (**B1**) to **C** from **Table- 4**
Else
2.2.2 Get the corresponding character (**B1**) to **C** from **Table- 5**
2.3 Add (**B1**) to (**T**).
End //while
Step3: Return (**T**).
End.

Implementation of the proposed method

a) Example (1)

1) Cover text قال الله تعالى في كتابه الحكيم

{يَا أَيُّهَا الَّذِينَ آمَنُوا إِنْ جَاءَكُمْ فَاسِقٌ بِنَبَإٍ فَتَبَيَّنُوا أَنْ تُصِيبُوا قَوْماً بِجَهَالَةٍ فَتُصْبِحُوا عَلَى مَا فَعَلْتُمْ نَادِمِينَ}

سورة الحجرات ايه رقم (6) **صدق الله العظيم**

2) Secret message

Data compression is compressed by reducing its redundancy, but this also makes the data less reliable, more prone to errors.

1. In this step convert each character to stream of 6-bits **B**=000100000001010100000001000000000011001111001101010000010010000101010011010011001001001111001110000000001001010011 ……

2. After applying the RNA coding the output will be

RNA=ACAAACCCAAACAAAAAUAUUAUCCAACAGACCCAUCAUAGCAUUAUGAAAAG …..

3. After converting each 3 symbols from RNA strand to binary this process called codon.

F=110100011100011111011100011000011110110001001011101000000001101011000111000110001010111000101000101110000001010001111

4. After applying the embedding algorithm the output will be

{يَا أَيُّهَا الَّذِينَ آمَنُوا إِنْ جَاءَكُمْ فَاسِقٌ بِنَبَإٍ فَتَبَيَّنُوا أَنْ تُصِيبُوا قَوْماً بِجَهَالَةٍ فَتُصْبِحُوا عَلَى مَا فَعَلْتُمْ نَادِمِينَ}

As we described there is a complete similarity between stegotext and cover text.

b) Example (2)

The secret message is

Information plays a vital role in today's generation and this information is very extensive and huge from the storage perspective. The computing methodology to get decisive result, DNA sequences must be used.

The cover text is

قال الإمام الرضا (عليه السلام) : ليست العبادة كثرة الصيام والصلاة وإنما العبادة كثرة التفكر في أمر الله

c) Example (3)

The secret message is

عن الامام السجاد (ع) اياك ومصاحبة الكذاب فانه بمنزلة السراب يقرب لك البعيد ويبعد لك القريب واياك ومصاحبة الفاسق فانه بايعك باكلة او اقل من ذلك.

The cover text is

عن الامام علي (ع) أكرم ضيفك وان كان حقيراً وقم على مجلسك لأبيك ومعلمك وان كنت اميراً.

Discussion

In this section we will illustrate some of the specifications of our proposed method.

1. The Embedding Capacity

The result shows that the capacity of the proposed method is very high compared with that of other text steganography methods. This high capacity is due to exploiting the entire cover text of the character, the spaces between words and diacritics, also in coding step there is a compressed data, this compressed data will increase the percentage of capacity, moreover, it stores the binary code after the stego text if the cover text has finished, all these things provide a high capacity for the proposed method.

2. Camouflage (Transparency)

The results of the proposed method show complete similarity between the Arabic cover text and the stego text compared with other approaches, the main reasons behind the best result are using Unicode and non-printed characters, the proposed method uses the Unicode and non-printed characters to embed secret information inside cover text, these characters do not appear when they merge with the cover text to embed data. These characters were put as indications to the receiving stage to retrieve secret data.

3. Security

Generally, in the steganography systems, there are two basic levels (coding and embedding). In the proposed method there is a high security in both levels.

Coding level uses secret information tables, Table-4 , Table -5 for mapping between each character with corresponding binary code so that security is provided.

In the RNAC uses secret tables, Table-1 for mapping between RNA element with two bits, and Table- (4and 5) for mapping between three elements RNA codon with binary code so that's provided a complexity, secrecy and give us a new code will be ready for embedding.
In embedding step it's hard to know where the information has been hidden, because of the complete similarity between the cover and stego text and using more than one method for embedding.

Performance analysis

We have computed the evaluation similarity of the proposed method by using Jaro-Winkler Distance and the comparison was made with other related approaches (see Table- **6**). Also, we ave computed the capacity ratio of the proposed hiding method for different cover text file sizes (see Table-**7**), and the results were compared with other methods (see Table **8**) by depending on equation (1).

Capacity ratio = (amount of hidden bytes) / (size of the cover text bytes)……...… (1)

Table 6- Jaro similarity score for the proposed and others approaches

Approach	Jaro similarity score	Jaro Similarity score %
AbdulRaheem approach [11]	0.9373	93.73%
Khan approach [14]	0.443	44.3%
Monika Agarwal approach [15]	0.95	95%
Sabrin Approach [9]	1.00	100%
Proposed approach	1.00	100%

This table illustrate that the proposed method and Sabrin approach is complete similarity between the cover text and stegotext.

Table 7- Capacity Ratio of the Proposed Approach

Information Hiding Method	Secret Message Sizes(byte)	Real Used Sizes of Cover(byte)	Percentage of Hiding Capacity Ratio (byte/byte)
Example(1)	922	2416	38%
Example (2)	1664	1648	100%
Example (3)	2272	1360	167%

This table illustrates capacity ratio for different cover text and different secret messages and the results is better than other approaches**.**

Table 8-Capacity ratio of the other approaches

Approach	Average of capacity (byte/byte)%
Kashidaa approach [16]	10.25%
Utilization of Two Diacritics approach [10]	71.45%
Sabrin Approach [9]	89%
Shirali-Shaherza [17]	74.32%

This table shows the percentage of capacity for other approaches, the result explain the proposed approach is better than the others.

Conclusion

In this section we will state the main conclusions:

1. Complete similarity between cover text and stegotext has been achieved by using non-printed characters in hiding method.

2. Using more than one hiding methods in the same cover text has provided a complexity that is suitable for security purpose.
3. We have satisfied flexibility to deal with Arabic and English languages in coding step which provides compression; by using Table-4, Table-5 we represent the characters of Arabic and English with special characters and Arabic diacritics by six bits.

References

1. Kataria, S., Singh, B., Kumar, T. and Shekhawat, H. S. **2013**. An Efficient Text Steganography using Digit Arithmetic. Proc. of Int. Conf. on Advances in Computer Science. AETACS, Elsevier, pp: 156-163.

2. Liu, Z., Jiang, Y. and Qian, P. **2013**. A Data-Hiding Method Based on TCP/IP Checksum. College of Computer Sci. Jiangsu Univ. of Sci. & Tech. Zhenjiang,*China Springer-Verlag Berlin Heidelberg*, pp: 1039-1044.

3. Odeh, A. and Elleithy, K. **2012**. Steganography in Arabic Text Using Full Diacritics Text. Presented at the 25th International Conference on Computers and Their Applications in Industry and Engineering (CAINE-2012), New Orleans, Louisiana, USA, 2012.

4. Banerjee, I., Bhattacharyya, S. and Sanyal, G. **2012**. *Text Steganography through Quantum Approach*. Wireless Networks and Computational Intelligence, Springer-Verlag Berlin Heidelberg, pp: 632-643.

5. Hosmani, S., Rama Bhat, H. G. and Chandrasekaran, K. **2015**. *Dual Stage Text Steganography Using Unicode Homoglyphs*, National Institute of Technology, Mangalore, Karnataka, India, Springer International Publishing Switzerland.

6. Alia, K. and Huda, N. **2015**. A proposed data encryption method based on RNA and Logistic map. M.Sc. Thesis, Department of Computer Science, The University of Technology.

7. Allen Wyatt. **2015**. Displaying Non Printing Characters, Available at: http://wordribbon.tips.net/t008879-DispalayingNonPrintingcharacters.html.

8. http://unicode.org/standard/standard.html 1991-2016 Unicode Inc. All rights reserved. 18/4/2016

9. Kadhem, S. M. and Bany, S. J. **2016**. Design of proposed Arabic text Steganography approach using non printed character. M.Sc. Thesis, Department of Computer Science, The University of Technology.

10. Ahmadoh, E. M. and Gutub, A. A. **2015**. Utilization of Two Diacritics for Arabic Text Steganography to Enhance Performance. *Lecture Notes on Information Theory*, **3**(1), pp: 42-47.

11. AbdulRaheem, A. A. **2014**. Information Hiding in Arabic Text Using Natural Language Processing Techniques, M.Sc. Thesis, Department of Computer Science, The University of Technology.

12. Akbas, E. A. **2010**. A New Text Steganography Method By Using Non Printing Unicode Characters. *Eng & Tech Journal*, **28**(1): 72-83.

13. Kadhem, S. M. and Wameedh, D. **2016**. Proposed Arabic Text Steganography Method Based on New Coding Techniqes. *International journal of Engneering research and Application*, **6**(9): 38-46.

14. Khan, F. R. and Sher, M. **2013**. On the Limits of Perfect Security for Steganographic System *International Journal of Computer Science Issues* (*IJCSI*), **10**, 4(1): 121-126.

15. Monika, A. **2013**. Text Steganography Approach: A Comparison, *International Journal of Network Security & Its Applications* (*IJNSA*), **5**(1): 91-106.

16. Gutub, A. A., Al-Alwan, W. and Bin Mahfoodh, A. **2010**. Improved Method of Arabic Text Steganography Using Extension Kashida Character, *Bahrain University Journal of information & communication Technology*, **3**(1): 68-72.

17. Shahreza, M. H. S. and Shahreza, M. S. **2006**. A New Approach to Persian/Arabic Text Steganography, In Proceedings of 5th IEEE/ACIS Int. Conf. on Computer and Information Science and 1st IEEE/ACIS Int. Workshop on Component-Based Software Engineering, Software Architecture and Reuse.