Journal of Theoretical and Applied Information Technology
15th April 2017. Vol.95. No 7

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

PROPOSED HYBRID METHOD TO HIDE INFORMATION IN ARABIC TEXT

[1] SUHAD M. KADHEM , [2] DHURGHAM W. MOHAMMED ALI

[1] University of Technology,Department of Computer Science, Iraq

[2] University of Technology,Department of Computer Science, Iraq

E-mail: [1]suhad_mallala@yahoo.com, [2] dhurgham_w@yahoo.com

ABSTRACT

In this method a new proposed approach to hide English texts (Secret data) in the Arabic text (Covers media). The secret text will be passed through several steps, Eventually it will be embedded in the cover text. In the proposed coding step each English character is converted to the binary code through secret tables that exist on the two sides (sending and receiving), which give us compression data. The output from the proposed coding will be two parts and these parts will be input to the next steps.

The next step is Modified RNA Codon (MRNAC) which takes the first part that result from binary code and returns a stream of binary to be ready for embedding in embedding step.

After that Modified Run Length Encoding (MRLE) that takes the second part that results from the proposed coding method and this result always contains a sequence of ones with fewer zeros, and apply RLE to this result.

The last step is the embedding step using specific Arabic Unicode characters and non-printed characters to embed the secret information and provide complete similarity between cover text and stego text since these characters don't appear when written.

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

1. INTRODUCTION

Steganography is the science and art of hiding information in which no one knows from the recipient parties that there is a secret message. Steganography hides a message inside another message and displays as a normal way that does not raise suspicions. The objective of steganography is to conceal, deliver messages, taking into consideration the exchange of information [1]. The Arabic language has some features that make it distinct from other languages. Shape of Arabic characters changes depending on the character position inside word, Besides, there are lots of Arabic characters that have points (dots), Some of them have one point and the others have two or three. The locations of these points are on top or the bottom of the letter. There are special forms of the Arabic characters located on top or bottom character, named "Diacritics"or "harakat"[2].

In the proposed approach the cover text is Arabic text, the coding step has a complicated coding and compression method on secret message. A new hybrid method of text steganography is presented based on modified RNA (MRNA) and modified run length encoding (MRLE) in which to overcome the drawback of classic RLE and functionality of MRNA to increase the level of security and complexity to get a good method used in security purpose, after that the result is hidden in sharp edges characters, characters, spaces between words and special characters, diacritics, using non printed characters and special Unicode character that have ASCII 157,158.

2. TEXT STEGANOGRAPHY

Steganography consists of four classes audio, video, image and text. Steganography is based on cover media that is used to conceal secret information. Text steganography can consist of anything from editing the formatting of an existing text, like replacing word within the text, to generating random character sequences or using context-free grammars to generate

Journal of Theoretical and Applied Information Technology
15th April 2017. Vol.95. No 7

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

readable texts [3]. Figure (1) illustrates the text steganography idea. Firstly, a secret message will be hidden in a cover-text by applying an embedding algorithm to produce a stego-text. After that the stego text will be transfered by communication channel [4]. There are several types of stego, that depend on the cover text, whether it is Arabic or English, in our method we have Arabic cove text, so we have Arabic text steganography.

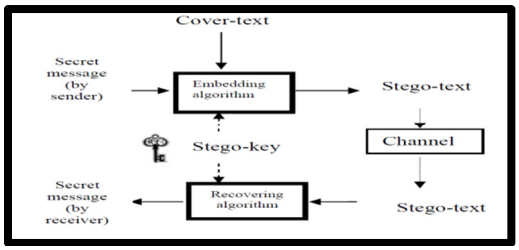

Figure 1: The mechanism of text

Text steganography is the preferable type to another type of communication and transfer because when store stegotext not require much memory and it is faster and effortless, as shown in Figure (2) [5].

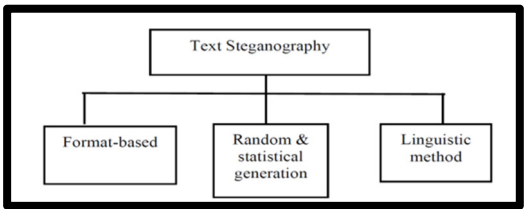

Figure 2: Three basic categories of text steganography

3. RUN LENGTH ENCODING METHOD (RLE)

It is a compression technique which is used for a given file that contains many redundant data. The input file or message is called run, which is encoded into two bytes. The first byte contains the number of times for a given character appears in the run. The second byte represents the value of the character [6]. If we have a sequence of an identical data, then the drawback of this method will appear, because the data will be expanded rather than compressed, for example, if we have a stream of data like 1010 the output of RLE will be **11101110,** so it is expanded rather than compressed, but if we have the a sequence of data like **1111111111111110**, then the output will be **16110**, so the data will be compressed [7].

In our proposed method the data are prepared in order to be a sequence of identical data to benefit from the RLE compression method, also will modify RLE in order to overcome its drawback.

4. RNA CODONS GENERATION [8]

The RNA codon is one of coding methods, to encode a generated bits stream to RNA code, we have to encode each two bits in processing step to RNA code (A, U, G and C) by using table (1) to get RNA strand. The stream of genetic-codes is cut into three genetic –codes to be a (cordon). Figure (3) illustrates the splitting of 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

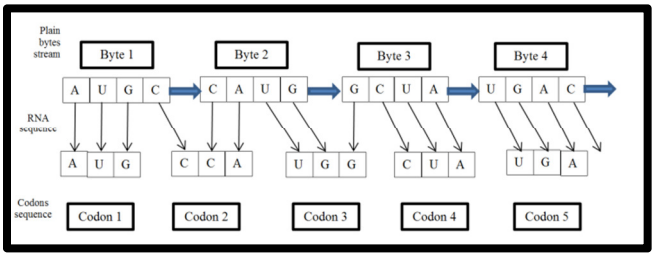

Figure 3: The splitting of stream genetic code to triple codon.

5. NON PRINTED CHARACTERS

We have several characters 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, and so on [9]. Table (2) illustrates these characters.

Table 2: Non Printing Characters

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Non printing characters					
Des	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 ARROW

6. UNICODE SYSTEM STANDARD

Unicode is a standard to encode all of the word's languages correctly on computers .It is an international standard. Its objective is to overcome ambiguities that traditionally arise when displaying complex scripts like Japanese, Arabian or Chinese on computer systems. Traditional character sets (like the American National Standards Institute ANSI alphabet) are depending on (8 bit) letters named a byte. A single byte can represent up to (256) different values and thus letters. This is well enough to represent western scripts like that being used in English, French or German language. However, if it gets to more complex languages like Japanese or Korean, (256) different letters are simply not enough [10].

7. LITERATURE SURVEY

These latest studies concerned with hiding secret information in Arabic language texts will be explained.

1. Ahmed C.S., GU X., Jia M., "Chinese Language Steganography Using the Arabic Diacritics as a Covered Media", 2010 [11].

In this study the procedure in steganography is developed by using the diacritics-Harakat- of Arabic language as a cover medium to hide the Chinese stroke text. So that in Arabic language, the diacritics-Harakat- which are used to represent vowel sounds are not useful when writing and sending the documents because the receiver can clearly understand the text without needing the Harakat. This approach uses eight different diacritical symbols in Arabic to hide binary bits in the original cover media. The embedded data are then extracted by reading diacritics from the document and translating them back to binary. Two diacritics are used to hide one Unicode character for strengthening the power of the security. The dictionaries of English-Chinese and Chinese-English are stored on both sides. Finally the diacritics are saved in the covered media so that the receiver can use them for getting the original message.

2. Adnan G., et., "Utilizing Diacritic Marks for Arabic Text Steganography", 2010 [12].

They utilize efficient carriers to hide information in plain text using Arabic diacritic marks. The ability of diacritics to be invisible superimposed on each other when typed multiple times consecutively makes them suitable for robust data-hiding applications. In this method, they propose two different algorithms to map secret messages into repeated diacritics in a non-wasteful fashion, where the number of extra diacritics is defined in fixed and variable size fashions. Therefore, the size of the outputted text is decided by the encoding flexibility. Both steganographic algorithms are characterized by several advantages over their existing counterparts. Finally, they provide a detailed performance analysis of both algorithms in terms of embedding capacity, robustness and file-size measures.

3. Nuur A.R., Ramlan M., Nuur I.U., "Sharp-Edges Method In Arabic Text Steganography", 2011 [13].

This method focuses on Arabic text steganograph they propose a sharp-edge method to encounter this issue. This new method will hide the secret bits in the sharp-edges of each character in the Arabic text document. The main processes involved are identifying the sharp-edges of the cover-text, secret message preparation to be hidden as a binary string and lastly, the bits hiding process. The experiments were increased up to 37.8% in resolving the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

capacity issue. The stage-text for this method has high invisibility, and therefore it is possible to be published publicly. They introduce keys to determine the position of the secret bit randomly. This method utilizes the advantages of the Arabic text for steganography and can also be implemented in the same text scripting of languages such as Jawi, Persian or Urdu.

4. Estabraq A.K., "Improvement of Information Hiding Using Artificial Intelligent Techniques", 2014 [14]

The main goal of Estabraq method is to mix the Elliptic Curve (EC) arithmetic and metaheuristic algorithms with steganography techniques in order to increment the capabilities of steganography in the diacritical Arabic text. Estabraq method consists of three phases. Firstly compression phase is to compress the secret messages into small codes based on B+ indexing method, secondly cryptography phase for adding the security by producing new algorithm to generate mask key based on EC algebra operations and metaheuristic algorithms, and thirdly is the steganography phase by embedding the encrypted secret message in diacritics of Arabic text based steganography algorithm, in this steganography algorithm DNA coding for Arabic diacritics and Arabic grammar rule.

5. Abdualraheem A. A., et., "Information Hiding in Arabic Text Using Natural Language processing Techniques", 2014 [15].

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

6. Ashraf. A. M, "An Improved Algorithm for Information Hiding based on Features of Arabic Text: A Unicode approach", 2014 [16].

A. A Mohamed presents text steganography method for Arabic text or any languages written in Arabic letters like Urdu, Persian, Pashto based on the features of Arabic text which offers high capacity, more security, and good robustness (depending just on isolated letters). The method embeds the information after applying Run length encoding and then hides the information in isolated letters only.

7. Esraa M.A, Adnan A.G., Utilization of Two Diacritics for Arabic Text Steganography to Enhance Performance", 2015, [17].

This method improved Arabic text steganography method, It hides secret data into text cover media in two Diacritics 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. This method exploits the possibility of hiding data in two diacritics, i.e. Fathah and Kasrah, adjusting the previously presented single (Fathah only) diacritic hiding scheme. This method of two diacritics stage-work features higher capacity and security showing interesting promising results.

8. Ammar O., Khaled E.," Robust Text Steganography Algorithms for Secure Data Communications, 2015 [18].

Ammar Odeh and Khaled Elleithy investigated different Steganography algorithms and presented novel algorithms employing text file as a carrier file. The proposed model hides secret data in the text file by applying various properties into file font format by inserting special symbols in the text file. In addition, the suggested model can be applied in both Unicode and ASCII code languages, regardless of the text file format. This system achieves a high degree of the main Steganography attributes like hidden ratio, robustness, and transparency. In addition, this method provides guidance for other researchers in text Steganography algorithms.

8. THE PROPOSED METHOD

In general the proposed stego method contains two stages: the sending stage, which forces the user to enter English secret text and the Arabic cover text to get the stego text. The other stage is the receiving stage, which takes the Arabic stego text in order to extract the secret text.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

8.1. The Sending Stage

This stage consists of main steps as in figure**4,** and algorithm**1** illustrate these steps

Algorithm (1): Sending Stage process

Input: Secret English text (**T**), Arabic cover text (**T1**).

Output: Arabic Stego text (**S1**).

Begin

Step 1: Normalize **T** by converting all its characters into lower case.

Step 2: Call algorithm **(2)** that takes **T** as input and returns binary codes **(B)** and list of check bits (**List1**).

Step 3: Call algorithm **(3)** that takes **B** as input and returns a stream of binary **F1** and check bit **C**.

Step 4: Call algorithm (**4**) that takes **List1** and **C** as input and returns List of non- printed characters **F2**

Step 5: Call algorithm (**5**) that takes **F1**, **F2** and **T1** as input and returns the Arabic stego text (**S1**).

Step 6: Return (**S1**).

End.

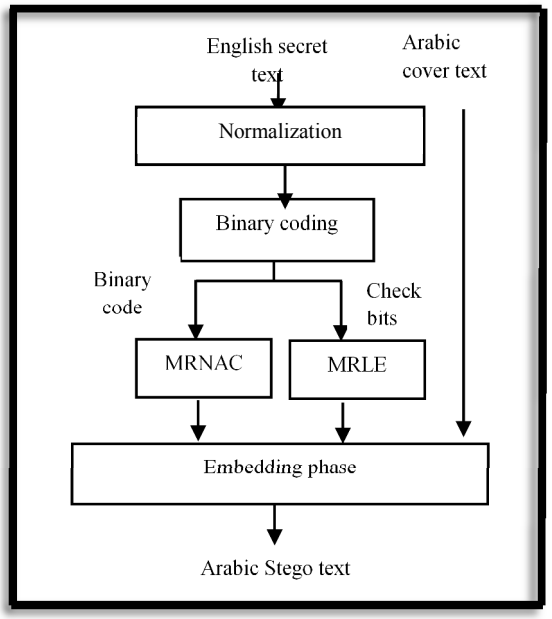

Figure 4: The structure of sending stage process

In this stage after entering the English text the first process is the normalization to convert the secret English text into the lower case. After that there are several steps: the proposed binary coding, the proposed Modified RNA Codon (MRNAC), the proposed Modified Run Length Encoding (MRLE) and the proposed embedding phase.

8.1.1. The proposed binary coding

This step converts each character of secret English text into binary code by using two databases (secret tables) and a list of check bits, such that any characters which will be found in table (3) will take one as a check bit otherwise found in table (4) it will take zero as a check bit, see algorithm (2).

Table 3: Characters that have one as acheck bit

special character	'!'	'?'	' @ '	…	'7'	'8'	'9'
5 bits	00000	00001	00010	…	11101	11110	11111

Table 4: Characters that have zero as a check bit

character	' '	'a'	'b'	….	'"' '	'.'	';'
5 bits	00000	00001	00010	….	11101	11110	11111

Algorithm (2): Binary Coding

Input: English text (**T**), **table (3), table (4)**

Output: Binary Code (**B**), List of check bits (**List 1**).

Begin

Step1: initialization

1.1 List1= [],**B**=""

Step2: While (**T**) ≠ "" do

Begin

2.1 Get the front character **C** from **T**

2.2 If **C** found in **table (3.1)** then

2.2.1 Get the corresponding binary code (**BC**) to **C** from **table (3)**

2.2.2 Add (**1**) to (**List1**).

Else

2.2.3 Get the corresponding binary code (**BC**) to **see** from **the table (4)**

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

2.2.4 Add (**0**) to (**List1**).
2.3 add (**BC**) to (**B**).
End//while
Step3: Return (**B** and **List1**).
End.

8.1.2 the proposed modified rna codons (mrnac)

The genetic code of proposed RNA Codons consists of **2** elements of nucleotides like (AC, CC, AU,….) called codons, since there are four bases (A,C,G,U) in two letter combinations so there are (16) possible codons (4^2 combination) which give all possibility of amino acid, All probability of RNA codon are generated randomly which represent amino acids, Figure (5) shows the splitting of stream genetic code to two codons, and table (5) illustrates the proposed amino acid probability.

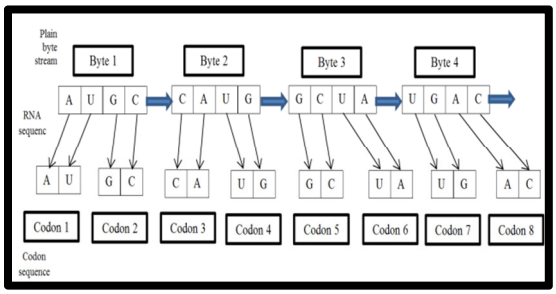

Figure 5: The splitting of stream genetic – code to two codons

Table 5: The proposed RNA amino acid probability

AA	AC	AG	….	UC	UG	UU
0000	0001	0010	….	1101	1110	1111

Algorithm (3) Modified RNA Codons MRNAC
Input: stream of binary **5** bits (**B**), **Table (1)** and **Table (5).**
Output: binary code (**F1**), Check bit (**C**).
Begin
Step 1: **F1**="".
Step 2: If (**B**) ≠ "".
Step 3: Use **table (1)** to convert each two bits to their corresponding RNA to get (**List RNA**).
Step 4: Compute length of (**List RNA**) to get (**NN**).
4.1: If (**NN**) is odd then.
4.2: Insert (**A**) at the front of (**List RNA**).
4.3: C=1.
Step 5: Else **C=0.**
Step6: Use **table (5)** for mapping from Modified RNA Codons (**two symbols**) strand to their corresponding binary to get stream for **4** bits (**F1**).
Step 7: Return Stream of binary **4** bits (**F1**), Check bit (**C**).
End.

8.1.3. The proposed modify run length encoding (mrle)

The proposed modified run length encoding (MRLE) will apply to the check bits list that results from the binary coding step, and this list always contains a consequence of ones with fewer zeros since each alphabetical letter (that appears frequently) found in table (3) takes one and the other special characters (that appear few times) will take zero, therefore (RLE) will be suitable for compression (see algorithm (4)). MRLE will depend on two databases (secret tables), one of zero's counters as in table (6) and the other one of ones counters as in table (7), such that each counter will be replaced by the ASCII code of a non-printed character that will be suitable for the steganography purpose in the next step.

Table 6: Database for swapping zeros counters of RLE with non-printed characters

Table 7: Database for swapping ones counters of RLE with non-printed characters

Counters of RLE for zeros	Non-printed character	Counters of RLE for ones	Non-printed character
0	30	0	31
1	11	1	21
2	12	2	22
3	14	3	23
4	15	4	24
5	16	5	25
6	17	6	26
7	18	7	27
8	19	8	28
9	20	9	29

Algorithm (4): Modified Run Length Encoding (MRLE)
Input: List of check bits (**List1**), check bit (**C**), **Table (6)** and **Table (7)**.
Output: List of ASCII of non-printed characters (**F2**).

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Begin
Step1: Initialization: - **F2**= [], **i**=1, **C**=1 or **C**=0
Step2: Put **C** in front of **List1** /* merge the check bit **C** with the list of check bit */
Step3: Compute length of **List1** to be **N**
Step4: while **i <= N** do
Begin
4.1 C=1
4.2 j=i+1
4.3 While **list1 [i]** =**list[j]** do
4.3.1 C=C+1
4.3.2 j=j+1
End//while
4.4 If **list1 [i]** =0 then
4.4.1 For each digit (**D**) of **C** do
A. Use Table (**6**) to get the corresponding non-printed character (**NP**) of **D**.
B. Add **NP** to **F2.**
Else
4.4.1 For each digit (**D**) of **C** do
A. Use Table (**7**) to get the corresponding non-printed character (**NP**) of **D**.
B. Add **NP** to **F2.**
End//If
4.5 i=j
End//while
Step5: Return (**F2**).
End.

8.1.4. THE PROPOSED EMBEDDING PHASE

The last step in sending side is the embedding phase that consists of two algorithms, the first algorithm takes the output from MRNAC and the Arabic cover text as input, the output from this algorithm is the Arabic stego text, the second algorithm takes the Arabic stego text that results from the first algorithm and the list of non-printed characters that results from MRLE as input and discarding the final Arabic stego text. So in the embedding phase the two different algorithms will be applied to produce the Arabic stego text, so it's hard to know where the secret information is embedded in the Arabic cover, moreover more security is added to our method. See algorithm (5) which illustrates the embedding phase.

Algorithm (5): The Embedding Process

Input: Binary code (**F1**), list of ASCII of non-printed characters **(F2),** the Arabic Cover text **(T1).**
Output: The Arabic stego text **(S1).**
Begin
Step1: S1="".
Step2: Call algorithm **(6)** that takes **F1** and **T1** as input and returns the Arabic stegotext **(S).**
Step3: Call algorithm **(7)** that takes **F2** and **S** as input and returns the Arabic stegotext (**S1).**
Step4: Return **(S1)**
End.

8.1.4.1. THE FIRST EMBEDDING ALGORITHM

The first algorithm in the embedding process method takes the binary codes (that result from the MRNAC) and the Arabic cover text as input and returns an Arabic stego text. Algorithm (6) illustrates that.

Algorithm (6): The first embedding process

Input: Binary code (**F1**), Arabic cover text **(T1).**
Output: Stego Text **(S).**
Begin
Step1: initialization step:
S="".
M=length of **F1**.
L=length of **T1.**
j=k=1.
Step2: while ((**j**<=**L**) and (**k** <= **M**)) do
/*While we don't reach the end of cover text and we still have a binary code */
Begin
If **k<=M** then /* if we still have a binary code */
Add **T1[j]** to **S**
If **F1[k] =1** then
Begin
If **T1[j]** is diacritics (Haracat) then
Add the Unicode character that has **157** or **158** ASCII to **S** according to the pervious character
/* Unicode **158** for connected character Unicode **157** for isolated character*/
Else if **T1[j]** is a special character or spaces between words then
Add the Unicode character that has **157** ASCII to **S**
Else
k=k+1
j=j+1
Else//If **F1[k]=0 then**
k=k+1.
End//while

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

End if
Step3: if (**j**>**L**) and (**k**<**M**) then
/* if we reach to end of the cover and we still have a binary code */
While **k<=M** do
Begin
If **F1[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 the end of the cover text yet and the binary code is finished */
Begin
Add two Unicode characters that have ASCII **157** or **158** to **S** according to the character type /* the stop mark of binary code*/
While **j**<=**L** do
Begin
Add **T1[j]** to **S** /* add the remaining cover text to the stego text */
j=j+1
End // while
End
Step4: Return (**S**).
End**.**

8.1.4.2.the second embedding algorithm

The second algorithm in the embedding process method takes a list of ASCII codes of non-printed characters (that result from the MRLE), and the Arabic stego text that results from the first embedding algorithm as input and returns Arabic stego text. This text represents the final result in our proposed system, see algorithm (7) which shows the main steps of the second embedding algorithm that use sharp Edges of Arabic characters that come after the spaces between words in order to embed the ASCII code of non-printed characters.

Algorithm (7): The second embedding process
Input: List of ASCII of non-printed characters **(F2)**, Arabic stegotext **(S).**
Output: Arabic stegotext **(S1).**
Begin

Step1: initialization step:

S1="".
N=length of **F2.**
L=length of **S.**
i=**j**=1.
Flag= false

Step2: while ((**j**<=**L)** and (**i** <= **N**)) do
/* While we don't reach to the end of cover text and we still have a non-printed characters do */
Begin
If **S[j]**=" " then
Flag=true
Else
Add **S[j]** to **S1**
If flag then
Begin
M= length of **F2[i]**
If **M=2** then
If **S[j]** is a character that has one or three sharp edges then
Begin
Add the non-printed character to ASCII found in **F2[i]** to **S1**
i= i+1
End
Else
If **M=4** then
If **S[j]** is a character that has four or five sharp edges then
Begin
Add the non-printed character to ASCII found in **F2[i]** to **S1**
i= i+1
End
Else
If **M=6** then
If **S[j]** is a character that has two sharp edges then
Begin
Add the non-printed character to ASCII found in **F2[i]** to **S1**
i= i+1
End
Flag= false
End if
j=j+1
End // while
Step 3: If **(j> L)** and **(i<N)** then
/* if we reach to end of the cover and we still have non-printed character*/
While **i<=N** do
Begin
If **F2[i] = One non-printed character** then
Add the non-printed character that has **3** ASCII to **S1**
Else If **F2[i] = Two non-printed character** then
Add the non-printed character that has **5** ASCII to **S1**

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Else If **F2[i] = Three non-printed character** then
Add the non-printed character that has **7** ASCII to **S1**
i=i+1
End
Else /* we don't reach the end of cover text yet and the non-printed character is finish */
Begin
While **j<=L** do
Begin
Add **S[j]** to **S1**
j=j+1
End // while

Step4: Return (**S1**)
End.

8.2. The Receiving Stage

This is the second stage of the proposed method. This stage consists of main steps as in figure (6). The input to this stage will be the Arabic stego text and the output will be the extracted original secret text, this phase will reverse the steps of the sending stage, in this stage will extract the secret message from the Arabic text by reverse the sending side steps, where (MRNAD) is Modifyied RNA Decoding and (MRLD) is Modified Run length Decoding.

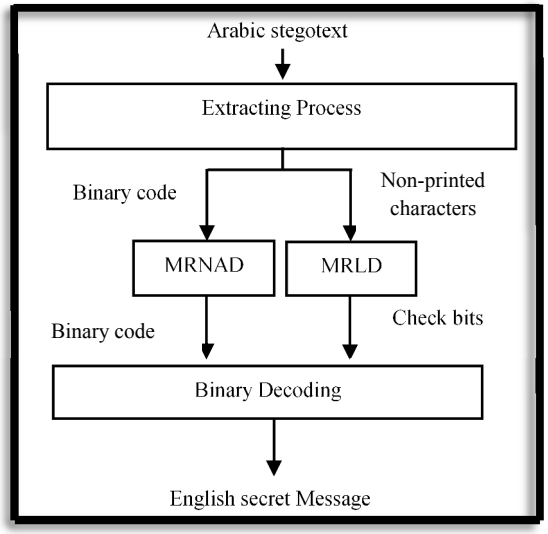

Figure 6: The proposed receiving stage

9. IMPLEMENTATION OF THE PROPOSED METHOD

Example1

English secret message:
Choose your neighbor before you choose your house.
Arabic cover text:
بِسْمِ اللَّهِ الرَّحْمَٰنِ الرَّحِيمِ

(اللهُ نُوْرُ السَمَوَاتِ وَالارضْ)
صدق الله العظيم
Algorithm (2) binary coding will take the secret message.
This algorithm will compute the number of characters after that will cut each character in the secret message, so the output from this step is a list of characters.
The number of characters is **50**
["c','h','o','o','s','e',' ','y','o','u','r',' ','n','e','i','g','h','b','o','r',' ','b','e','f','o','r','e',' ','y','o','u',' ','c','h','o','o','s','e',' ','y','o','u','r',' ','h','o','u','s','e','.']

The next step will check each character if found in table (3) then get its corresponding binary and discarding one as a check bit.
Else// if
Will check each character if it is found in table (4) then get its corresponding binary and discarding zero as a check bit.
So the output will be a stream of (5 bits) with check bits as follows.
[t("00011",0),t("00011",1),t("01000",1)…]
The next step will separate the stream of binary code (5 bits) from identical check bits.
List of stream (5 bits)
["00011","00011","01000,…]

The list of identical check bits that represents **List1** in the algorithm
[0,1,1,1,1,1,1,1,1,1,1,……1,1]

The last step in **algorithm (2)** is convert the list of binary (5 bit) to stream of binary that represents **B** in the algorithm.
000110001101000……
….0101101

Algorithm (3) will take **B** and convert each two bits to RNA to produce a list of an RNA strand using **table 1**

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

["A","C","G","A"U","U","U","U","A","U","A",
……,"U","U","C","C","G
","C","G","C","C"]

The next step in this algorithm computes the length of RNA after converting it to a string.
AGUUGGUAGAAA……..……
….AUCGGUAGAAAGACUUCCGCGC

The total number of RNA is **125**.
So the algorithm doesn't add **A** at the front of RNA strand
The check bit (**C**) will be **one (1).**

The last step in this algorithm modifies RNA codon through a mapping from each two RNA element cordons with a stream of 4 bit binary by using **table (5)**
["0000","0110","1111","0011","0010",……
……..,"0111","1101","0110",
"0110","0101"]

Convert this list to string in order to get **F1,** and this result is ready for embedding.
000001100011010000111101111101011O……
……..11010110011001
01

Algorithm (4) will take **list1** and the check bit **(C)** and will add the check bit in front of **List1** after converting the **List1** to string.
1111111111111111111111111111111111111111
111111111111

After that, RLE is applied
[p(51,1)]

The next step is MRLE in this step will swap the zero counter of RLE with non-printed character using **table (6)**, and swap the one counter of RLE with non-printed character using **table (7),** so the result will be list of non-printed characters **[2521]** that represent **F2** in algorithm.

The first step in **Algorithm (6)** takes the cover text **(T1)** and binary code **(F1)** to embed **(F1)** in the Arabic cover text
The output of the **algorithm (6)** is
(الله نُوْرُ السَمَوَاتِ وَالارضْ)
Algorithm (7) will take the output that results from **algorithm (6)** and **F2** that results from **algorithm (4) and the output is**
The output of the **algorithm (7)** is
(الله نُوْرُ السَمَوَاتِ وَالارضْ)

This is the last result on sending side, as we see our method give as complete similarity between the cover text and stego text.

Example2:
Secret text is:
Linguistic steganography, a type of steganography is defined as a collection of techniques and methods that allows the hiding of any digital information within texts based on some linguistic knowledge

The Arabic cover text is:
مقطع خطبة الإمام علي (عليه السلام) المعروفة بالطالوتية

(الحمد لله الذي لا اله الا هو ، كان حيا بلا كيف ، ولم يكن له كان ولا كان لكانه كيف ، ولا كان له اين ولا كان في شئ ، ولا كان على شئ)

Example3
English Secret Text:
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 results, DNA sequences must be used.
Arabic cover text:
قال الإمام الرضا (عليه السلام): ليست العبادة كثرة الصيام والصلاة وإنما العبادة كثرة التفكر في أمر الله

10. PERFORMANCE ANALYSIS

This section will compute the similarity of the proposed method by using Jaro-Winkler Distance and compared with other related approaches (see table8). Also, we will compute the capacity ratio of the proposed hiding method for different cover text file sizes (see table 9), and compared it with other methods (see table 10) by depending on equation (1).

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

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 8: Jaro Similarity Score for proposed method and Others Approaches

Test Number	Jaro Similarity scores	Jaro Similarity score %
Example1,2,3	1.00	100%
WMM approach [19]	0.8771	87.71%
Khan approach [20]	0.443	44.3%
Monika Agarwal approach [20]	0.95	95%

Table 9: Capacity Ratio of the Proposed Approach

Information Hiding Method	Secret Message Sizes(byte)	Real Used Sizes of Cover(byte)	Hiding Capacity Ratio (byte/byte) %
Example 1	400	560	71.4%
Example 2	1600	2096	76%
Example 3	1664	1664	100%

Table 10: Capacity Ratio of Other Approaches

Approach	Average of capacity (byte/byte)%
Kashidaa approach [22]	10.25%
Utilization of Two Diacritics approach [10]	71.45%
Shirali-Shaherza [23]	74.32%
Sabrin Approach [24]	89%

11. CONCLUSIONS

1. Do not depend on a dictionary in a specific field for a secret message will make the system accept any English secret text. Also, do not depend on a big database which will speed the system execution.

2. Achieving a complete similarity between a cover text and stego text will give a power to the system, so the attacker doesn't raise doubt because of the existence of hidden message.

3. Using the proposed MRNA and MRLE will increase the complexity and therefore the security will be increased.

4. Using more than one embedding method will increase the security system.

5. Using secret information tables, table (3), table (4) for mapping between each character with corresponding binary code, and table (1) for mapping between RNA element with two bits, and table (5) for mapping between two elements of RNA codon with binary code, and table (6), table (7) for mapping between the counters of RLE with ASCII of non printed characters will increase the security level.

6. Using a Unicode character that has 158 ASCII Arabic with connected characters and Unicode character that has 157 ASCII with Arabic isolated character, (in this case this Unicode doesn't work) will be useful for stego purpose.

7. The proposed method takes all the cover texts (characters, diacritics, spaces between word and special character) and utilizes the spaces after the cover text, thus will increase capacity ratio.

8. There is a new compression method to compress the English text in coding step and this method provides security.

9. Theoretical implications are built hiding system Which functions following
   - Compress data before hiding throughout MRLE
   - Increase the security throughout MRLE, MRNA and secret tables of them

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

- Using Hybrid method for hiding that increase complexity.

REFERENCE

[1] M. K. Kaleem., *" An overview of various forms of linguistic steganography and their applications in protecting data",* Journal of Global Research in Computer Science, Volume 3, No. 5, May 2012.

[2] Fahd Al-Haidari.,et..,*"Improving Security and Capacity for Arabic Text Steganography Using 'Kashida' Extensions",* AICCSA 2009 - The 7th ACS/IEEE International Conference on Computer Systems and Applications , Rabat, Morocco, 10-13 May 2009, PP. 396-399.

[3] Monika Agarwal., *"Text Steganographic Approaches: a Comparison"*, International Journal of Network Security & Its Applications (IJNSA), Vol.5, No.1, January 2013.

[4] Fabien A. P. Petitcolas, Ross J. Anderson and Markus G. Kuhn. *"Information Hiding – A Survey"***,** Proceedings of the IEEE, special issue on protection of multimedia content, July 1999, pp. 1062 – 1078.

[5] Por L. Y., Delina B., *"Information Hiding: A New Approach in Text Steganography",* 7th WSEAS int. Conf. On Applied Computer & Applied Computational Science (ACACOS '08), Hangzhou, China, April 6-8, 2008.

[6] David Salomon**,** *"Data Compression The Complete Reference"***,** Fourth Edition, Computer Science Department, California State University, springer, 2007.

[7] By Steven W. Smith, Ph.D http://www.dspguide.com/ch27/2.htm

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

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

[10] Micha, Kosmulski, 2004, Introduction to Unicode, available at: http://www.Linux.com/archive/articles/39911

[11] Ahmed C.S., Gu X., Jia M., "*Chinese Language Steganography using the Arabic Diacritics as a Covered Media***",** International Journal of Computer Applications (0975 – 8887) Volume 11– No.1, December 2010**.**

[12] Adnan A Gutub ., Lahouari M Ghouti, Yousef S Elarian,Sameh M Awaideh2, Aleem K. Alvi2," *Utilizing Diacritic Marks for Arabic Text Steganography***",** Kuwait Journal of Science & Engineering (KJSE), Vol. 37, No. 1, June 2010

[13] Nuur A.R., Ramlan M., Nur I.U., **"***Sharp-Edges Method in Arabic Text Steganography***"** Journal of Theoretical and Applied Information Technology, 2011.

[14] Estabraq A.R.K., Hala B.A.*," Improvement of Information Hiding Using Artificial Intelligent Techniques"***,** M.Sc. Thesis, Department of Computer Science of University of Technology, 2014.

[15] Abdulraheem A.A.*," Information Hiding in Arabic Text Using Natural Language Processing Techniques"*, M.Sc. Thesis, Department of Computer Science ,The University of Technology, 2014.

[16] Ashraf A.M.," *An Improved Algorithm For Information Hiding Based On Features Of Arabic Text: A Unicode Approach*", Egyptian Information Journal, 2014.

[17] Esraa Mohammad Ahmadoh, Adnan Abdul-Aziz Gutub., *"Utilization of Two Diacritics for Arabic Text Steganography to Enhance Performance",* Collage of Computer & Information Systems, Umm Al-Qura University, Makkah, Saudi Arabia, Lecture Notes on Information Theory Vol. 3, No. 1, June 2015.

[18] Ammar O., Khaled E.*," Robust text steganography algorithms for secure data communications",* School of Engineering, University of Bridgeport Connecticut, May, 2015.

[19] Souvik Bhattacharyya, Indradip Banerjee and Gautam, *"A Novel Approach of Secure Text Based Steganography Module Using Word Mapping Method (WMM)",* International Journal of Computer and Information Engineering, 2010

[20] Khan F.R., M. Sher, "*On the Limits of Perfect Security for Steganographic System***"**, Department of Computer Science, International Islamic University Islamabad, October 2013

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[21] Monika Agarwal., "Text Steganographic Approaches: a Comparison", International Journal of Network Security & Its Applications (IJNSA), Vol.5, No.1, January 2013.

[22] Adnan A.G., Wael A., and Abdulelah M., "*Improved Method of Arabic Text Steganography Using Extension Kashida Character*", Bahrain University Journal of information & communication Technology, December, 2010

[23] Shahreza M. H. S., and Shahreza M. S., "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, 2006.

[24] Sabrin J.B., Suhad M.K.," Design of proposed Arabic text Steganography approach using non printed character", M.Sc. thesis, Department of Computer Science, University of Technology, 2016.