International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 11, No.1 (Jan-2022)

http://dx.doi.org/10.12785/ijcds/110155

Enhancing Arabic Text Steganography Based on Unicode Features

Adeel Alshamsi[1], Salem Albaloushi[1], Mohammed Alkhoori[1], Hamed Almheiri[1] and Nedal Ababneh[1]

Department of Information Security Engineering Technology (ISET), Abu Dhabi Polytechnic, P.O. Box 111499, Abu Dhabi, UAE

Received 15 Jun. 2021, Revised 18 Sep. 2021, Accepted 18 Nov. 2021, Published 31 Jan. 2022

Abstract: Steganography is a technique used for hiding secret information or data in a text, image, audio, or video to make it harder for the attackers to detect the hidden information. Many techniques were developed that target the English language but only few Arabic text steganography techniques exist. In this paper, we explore steganography and its types, presented multiple techniques for Arabic steganography, discuss advantages and disadvantages of each technique and described the main functions in our implemented code, after that, we evaluate each technique and suggested improvements that can benefit each technique to enhance their capacity and minimize any problems that can face them to reach our main goal, which is to improve this field to be able to implement its without worrying about any disadvantages that the techniques may have.

Keywords: Steganography, Cover Text, Stego-text, Non-printable letters, La.

1. INTRODUCTION

In these days, the rise of technology has tremendously increased, and it became one of the most important aspect in life of most people, people use technology in their day-to-day life and it became unavoidable to use technology, some of the uses of day-to-day technology includes but not limited to activities like online shopping, online banking services, and much more . This is why security becomes important more than ever. one of the main topics in security is data protection. Figure 1 presents three of most important types of data protection is Steganography, Cryptography, and Watermarking, these types of data protection all have the same goal of preventing unauthorized users to view or modify the data that are private in their different ways, firstly Cryptography is a technique used for displaying a plain text in an unclearway that make any unauthorized user unable to read or see the message, watermarking is hiding the data in a way that conveys some information about cover mediums such as copyright and ownership, steganography is a technique used to send hidden information secretly by using the characteristics of digital media as a cover, and there are different types of covers including text, video, audio, and image. And in this research, we would discuss steganography in general and its types and techniques. Meanwhile we would focus more about Arabic steganography techniques in addition to how we would improve the techniques so they would be unpredicted and would be more effective to be used in.

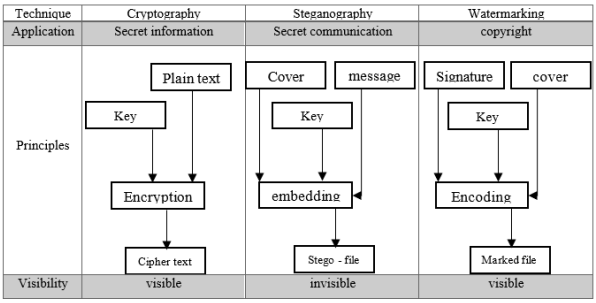

Figure 1. A comparison between data protection types

Steganography is a way to provide a communication secretly between two persons by embedding data in a digital file that could be an image, video, audio, or text as cover using a key to produce a stego-file that will make it very difficult or nearly impossible for the person in the middle to detect the stego-file and only the sender and the

E-mail address: Adeel.Alshamsi@gmail.com, A16000030@adpoly.ac.ae, A00037324@adpoly.ac.ae, A00036754@adpoly.ac.ae, Nedal.Ababneh@adpoly.ac.ae

receiver are able to detect and understand the file [16]. Embedding the data to make a stego-file is not a simple task and there are things to consider like size of the cover which should not be increased as much, and there are multiple features that we should consider when applying stenographic procedures: Firstly, the embedding capacity, which is how much can we change and alter the size of embedded data without affecting or changing the original quality. This means what is the maximum size of the data hiding in the cover file. Secondly how un-detectable the embedded data is, this means that the hidden data should be very hard or nearly impossible to detect without using a special tool to uncover it and a normal person cannot detect the hidden data [23]. Thirdly, robustness, which means the capacity of the embedding algorithm before and after extract the hidden data, in other words, use a complex algorithm so the intruder or an attacker cannot easily break it and get the secret message. And finally, tamper resistance, which means that if the unauthorized person gets hold of the hidden data and was able to detect it, it will be very hard for them to alter or destroy the secret data or message [8].

A. TEXT STEGANOGRAPHY

Text steganography is generally alienated into trio modules, the first technique is the Format Based Technique in which text data is embedded in the cover text by changing the formatting of the cover text [26]; this could be done by resizing the font size, injecting spaces among words, non-displayed typescripts [22]. The second technique is the Linguistic Method, Linguistic analysis is done in this practice. Finally, the third technique would be Random and Statistical generation method where comparison is not done with the known plain text and most stenographers generates their own cover texts [18].

B. IMAGE STEGANOGRAPHY

The art of how to hide sensitive or secret data in an image, to prevent any unauthorized person to assume that the image includes any sensitive data [9]. In other words, it embeds data or information in a digital image as a cover by keeping these data hidden and infeasible from anyone who can view and see the image [7]. The main factors in this type of steganography are data volume, the stability of embedded data without losing, modification, and third-party removal [1]. To achieve the goals of transmitting a secure message using image steganography requires knowledge about basic terminologies, classifications, and techniques [11].

C. VIDEO STEGANOGRAPHY

Video steganography is way to hide secret data or massages in a video so the chances of unauthorized users to access the data is reduced [14], and in the recent years the use of videos through the internet has increased drastically so it's natural to develop a steganography embedded in videos, steganography in videos has higher potential in hiding secrets due to the nature of videos which has numerous redundant bit [4], although multiple videos steganography techniques has been used, no ideal technique is ultimately ideal [24], it all depends on the needs and the circumstances, the main idea of video steganography is to combine a video file with a secret message or data through an steganographic algorithm and in return a stego file will be the result [2][13].

D. AUDIO STEGANOGRAPHY

Audio steganography is used to send a secret message through an audio file that appears as a normal file that if ran, will output normal sounds like any other audio file, but if it was decrypted through a certain procedure will give a secret message that unlike other encryption techniques [21], it won't be an obvious encrypted message but a clear one that uses the audio file as a cover, it can be a text, image, or an audio that is covered by the cover audio, the main technique is that the sender will use a key known only to the receiver [25], then they will use the audio file to cover the secret message which will create the stego audio file. There are three main domains in the audio steganography field that differentiate the techniques and techniques based on what domain they use were each similar technique are grouped together [10].

2. ARABIC TEXT STEGANOGRAPHY TECHNIQUES

A. MULTIPOINT ARABIC LETTER

Multipoint Arabic letter allows the user to hide more than 2bits per letter. And to do that, the user must increase the size of the file by using vertical shift point algorithm, this way the carrier's size will be increased without alteration or modification, after adding the secret information inside, a process of converting the file into a picture in order to avoid and prevent any re-typing issues [12]. Also, not being able to re-type can be a disadvantage that can cause problems because the hidden information is dependent on the format of the file, this makes the attacker suspicious about these different formats [3].

B. LA STEGANOGRAPHY TECHNIQUE

La stands for the combination of the letters Lam and Alef in order to conceal information, this technique hides information by embedding an Arabic extension character between the letters Lam and Alef, for example if the users want to hide a 0 bit they will have to use the normal form of the Lam and Alef, but if the users want to hide 1 bit they will have to use a special word La, La technique can be used printed documents and as well as electronic documents [17].

C. POINtED LETTER

This technique is made possible by Shirali-Shahreza [6], this technique is performed by hiding the secret information in text form into the dots of the letters of the Arabic language as seen in figure 2, and to do that the process start with compressing the size of the secret text, then examining the cover medium content from each line or character. The location the dots may be affected by the hidden information bit in case that the pointed letter is uncovered. Pointed letter has a big advantage that is it provides good secrecy and a big capacity of the secret text [15].

Figure 2. Pointed letter

D. ARABIC DIACRITICS

In the Arabic language, there are eight different diacritics, each of them pronounces in a different way as shown in Figure 3, in this technique they use a Stego-file that contain a diacritical text by hiding a secret message in the Diacritic [5]. So, any diacritic letter held “1” and a non-diacritic letter held “0”. This technique has high ability but it does not have enough invisibility so a reader might note that the text is not normal [28].

Haraka	Letter with Haraka	Pronunciation
Dama	دُ	Do
Kasra	دِ	De
Fatha	دَ	Da

Figure 3. Arabic language diacritics pronounce.

E. KASHIDA TECHNIQUE

Another technique that may have a better effect than the previous techniques is the Kashida technique that uses extensions. These extensions will happen either before or after a pointed letter that are considered "ones", the only exception would be if the pointed letter came the last in the word for example "حوت" since the last letter cannot have an extension after it, the better solution would be to ignore it and use the next pointed letter to hide the secret bit or use extension before the pointed letter. The binary value for the letter " أ " is 11011000 10100011, by using the extension technique we can hide those bits into the cover text. As show in the below example, we used an extension before pointed letters that hide a "1" bit and added an extension before non-pointed letters that hide a "0" bit. These techniques have their advantages and disadvantages.

F. REVERSE FATHA

The technique used in this technique is customizing the Fatha on a single alphabet by reversing it [19], the Fatha line is always displayed from left to right, changing the state of the Fatha line from left to right is stated by reverse Fatha as seen in Figure 4.

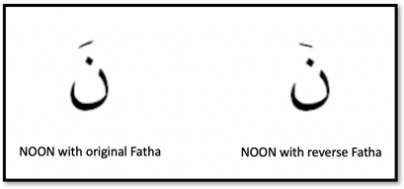

Figure 4. reverse Fatha

As seen in Figures 5-7 the alphabets that contains the secret text is selected by implementing the reverse Fatha on the alphabets which contains the hidden message where each alphabet that has a reverse Fatha contains a hidden text.

مُجْتَبَىٰ وَهُّوْ مَحَاضَر

Figure 5. cover text with original Fatha

مُجْتَبَىٰ وَهُّوْ مَحَاضَر

Figure 6. cover text with reversed Fatha

توم حض

Figure 7. Hidden text

The algorithm premeditated to hide the secret message by detecting each letter which has a Fatha so it could be used to reverse the Fatha, breaking separately each word of the secret message while the reversing the Fatha of the following alphabets would help processing the new Font family in the word software.

The algorithm designed to find the secret message after receiving the text starts by marking the alphabets that contains reverse Fatha, extract the alphabets starting from the top of the article to the bottom and in the last step starts merging the extracted alphabets to retrieve the secret message [19].

G. UNICODE AND NON-PRINTABLE CHARACTERS

Unicode allows users to encode all kinds of language in the world, each character in each language have their own special Unicode that makes them different than the rest of characters, the Unicode standard is user internationally to print and present texts in computers, it have 65536 spaces for

character from all language thank to its 16-bit encoding, the Arabic language letters are represented in Unicode in a range of values (0600-06FF) which represents all form of each Arabic character [27], for example, as shown in Figure 8, the letter (ب) is represented in the value of FF8F when it is isolated and not connected to any other character, it is represented in the value of FE90 when it appears in the end of the word, its represented in the value of FE92 when it appears in the middle of the sentence and is connected to two characters, one before it and one after, it is also represented in the value of FE91 when it appears at the start of the word were it is not connected to any other letter [5].

Figure 1. Example of Arabic letters in Unicode

For Non-Printable characters, As seen in the table [1], these characters, even though they have a hexa value that represents them, they do not have any shape that represents them which can be used in steganography to hide secret bits in these characters, were they will not be noticed since they do not get printed, and the Stego-text will have the same appearance as the cover text without any changes [5].

Table 1. Non-printable characters

Unicode code point	character name
U+2000	EN QUAD
U+2001	EM QUAD
U+2002	EN SPACE
U+2003	EM SPACE
U+2004	THREE-PER-EM SPACE
U+2005	FOUR-PER-EM SPACE
U+2006	SIX-PER-EM SPACE
U+2007	FIGURE SPACE
U+2008	PUNCTUATION SPACE
U+2009	THIN SPACE
U+200A	HAIR SPACE
U+200B	ZERO WIDTH SPACE
U+200C	ZERO WIDTH NON-JOINER
U+200D	ZERO WIDTH JOINER
U+200E	LEFT-TO-RIGHT MARK
U+200F	RIGHT-TO-LEFT MARK

3. IMPLEMENTATION METHODOLOGY

The technique that we implemented and chose was the non-printable characters, since it is the best technique to hide secret messages without changing the shape or content of the cover text, which perfectly serves the main purpose of steganography, this technique, if done properly, will not have as much limitations as the other techniques in terms of capacity or having a specific cover text, which is why we chose to implement it to dive more into it and explore it more than the other techniques.

A. CODE

The code we are using has been created by Mosaed BaOmar [28], the code allows the user to hide secret messages in an Arabic language cover text, the secret message can be either Arabic or English, the user will need to input a suitable amount of text that will be enough to hide the secret message without taking in mind the value of the cover text in terms of having to use specific letters or not, we will start displaying the functions of the code and what each one of them do.

B. IMPLEMENTATION

The program asks the user to input an Arabic cover text which will be used to hide the secret message within it, after that the user will input the secret message, if the cover text's size is not enough to hide the secret message, the program will send an alert to the user telling them that they need a larger cover text, after the user presses "get stego text" button, the program will convert the input to binary and create a binary bit stream of ones and zeros where each hidden letter is separated from the next hidden letter by an extra bit that act as a separator, after that, the program will add the zero-width joiner and zero-width non joiner letters which are the non-printable letter that will represent the secret bits, after adding them to the cover text, the stego text will be created. In the decoding part, the program will ask the user to input the stego text and press "retrieve secret message" button, after pressing the button, the program will remove the non-printable letters from the stego text which will retrieve the secret message without any changes, Figure 9 presents three different scenarios that we used when using the program, first, using an Arabic secret message, after that, using an English secret message, after that, using a mix different languages as secret message to prove that the program will accept any secret message input type as long as it has a Unicode value.

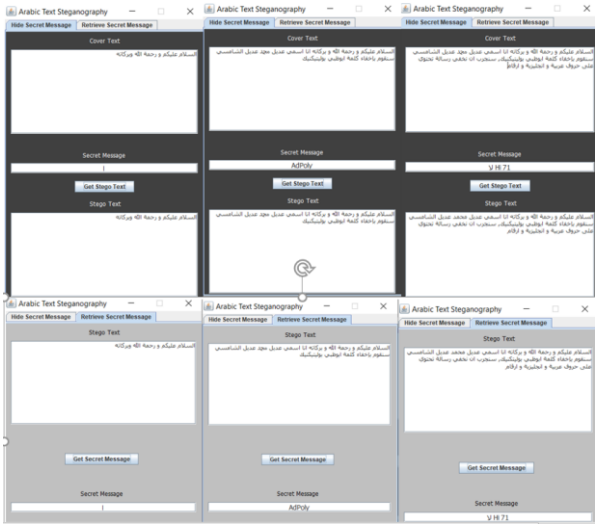

Figure 9. program Implementation

4. RESULTS AND DISCUSSION

A. NON-PRINTABLE LETTERS

As seen in the "Implementation" section, the program was able to embed Arabic letters, English letters and numbers and successfully retrieve them, based on our evaluation of the Arabic steganography techniques we decided to choose the non-printable letters to be implemented since compared to the other techniques it is the best one in terms of the ability to implement it without having to have a specific cover text. To prove that the program will use any non-printable letters and not only zero width-joiner and zero width non-joiner, but we also used two other non-printable letters which gave the same results as can be seen in Figure 10 and in Figure 11 the output gave the secret message without any changes.

```
// Add ZWNJ -> 'u200F' after ignored letters
stegoText.append('\u200F');
stegoText.append(coverTextChar[j]);
f (isConnectedToNextLetter(coverTextChar[j - 1])
if (Character.isWhitespace(coverTextChar[j]))
        stegoText.append('\u200F');
else
        // Add ZWN -> 'u200E' after ignored lett
        stegoText.append('\u200E');
stegoText.append(coverTextChar[j]);
```

Figure 10. Changing Non-printable letters

Figure 11. New Non-printable letters retrieve

After reviewing the program, we decided to change the values that are appointed to each letter to random values, each letter in the program was assigned a 0 or 1 value based on its appearance, the pointed letters were assigned a 0 value and non-pointed letters were assigned a 1 value, table 2 shows the letters that were assigned a "1" value and table [3] shows the letters that were assigned a "0" value, the values were changed to a random sequence to eliminate any sequences in the values of the letters.

Table 2. One Values

Letter	ل	م	ه	ر	ع	أ	د	ك	ح	ى	ت	إ	خ	ط	ء	ض	غ	ظ
Value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Table 3. Zero Values

Letter	ن	ي	ب	ف	ق	س	ج	ذ	ث	ص	ش	ز	ا	ئ	آ	ؤ
Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Each secret letter in the non-printable letters technique needs 9 bits since 8 bits will be assigned to one secret letter and an extra bit will be used to separate each secret letter from the next secret letter, so to get the number if secret letters that can be hidden in each text, we can use this equation, Hidden letters = number of letters in a text/9, as demonstrated in Figure 12, the capacity of this technique is high compared to the other techniques that had far more problems and obstacles compared to the non-printable letters technique.

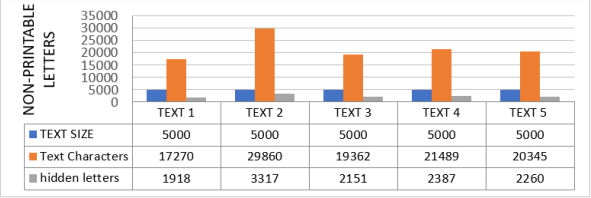

Figure 12. NON-PRINTABLE LETTERS

B. LA TECHNIQUE

We started implementing the LA technique to five different random texts that contains an average of 5,000 words in order to make an estimation of using the LA in any text that's given by the user where in our research we found out that there is an average of 1,834 LA in a text of 25,000 words while that equals a percentage of 0.0736% found in a combination of 5 different texts with a total of 25,000 words as each text have a different ratio of LA in the text. Figure 13 demonstrates number of LA found in each text of five different random texts.

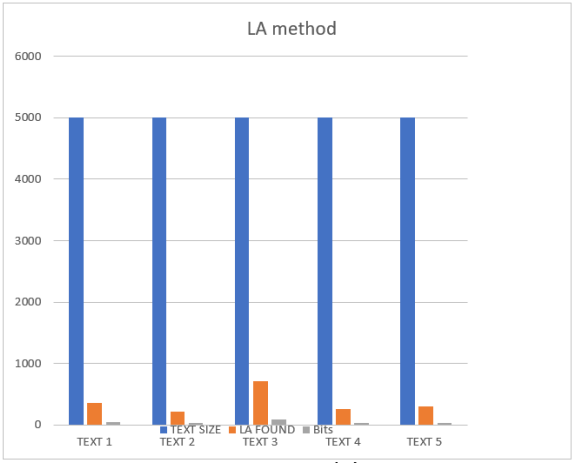

Figure 13. LA Technique

As shown in the figure, different random texts could have different number of LA found, as we found 385 bits of LA in the text 1 which equals 44.75 byes on the other hand we found 211 LA in text 2 which is equal to 26.375 bytes while in text 3 we found 708 LA that is corresponding to 88.5 bytes which is the highest number we found between the five random texts as text 4 contains 261 LA that equals 32.625 bytes and finally text 5 has 296 LA which is exactly 37 bytes. If we combine all five texts together, we will gain a number of 1,834 LA which is equal to 229.25 bits which sounds to be good, but the size of the text is extremely Large.

C. AL TECHNIQUE

The LA technique could be used but the size of the text has to be enormous in order to find many LA in the text meanwhile we found out a solution to be used as instead of using LA for the purpose of steganography we would use AL "ال" as it could generate more bits than using the LA technique as tested in Figure 14. As we used AL instead of LA we could see a notable difference on the charts as in text 1 we found 1758 AL which equals to 219.75 bits, while in text 2 we found 709 Al which equals to 88.625 bits on the other hand the highest is text 3 where we found 2254 Al that corresponds to 281.75 bits whereas on text 4 we found 1014 AL that equals 126.75 however on text 5 we originated 1103 Al that is 137.875 bits. Overall a total of 6,838 AL/854.75 bits where found in the same random texts used in the LA technique so we concluded using AL instead of LA would be much better as smaller texts would generate more bits to be used in steganography than using the LA technique which needs an enormous size of text in order to hide few bits, Figure 14 presents the evaluation of the AL technique.

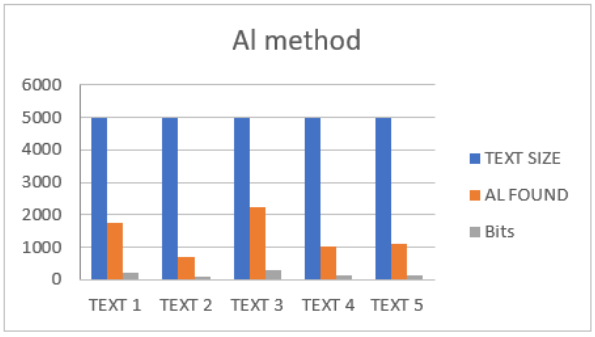

Figure 14. AL Technique

D. POINTED LETTERS

As we know there are 28 letters in the Arabic language 15 are pointed letters and 13 non-pointed letters. Which is more than half is pointed letters, as shown in Figure 15.

1	ب	Ba
2	ت	Ta
3	ث	Tha
4	ج	Jeem
5	خ	Kha
6	ذ	Thal
7	ز	Zay
8	ش	Sheen
9	ض	Daad
10	ظ	Thoh
11	غ	Ghayn
12	ف	Fa
13	ق	Qaf
14	ن	Noon
15	ي	Ya

pointed letters

1	أ	**Alif**
2	ح	Hha
3	د	Dal
4	ر	Ra
5	س	Seen
6	ص	Saad
7	ط	Toh
8	ع	Ayn
9	ك	Kaf
10	ل	Lam
11	م	Meem
12	ه	Ha
13	و	Wow

Non-pointed letters

Figure 15. Pointer letters & non-pointed letters

We started implementing the pointed letter technique to the same five texts was used in the previous technique. We found out that there are 33,849 pointed letters in a text of 25,000 words. So, we determine that mostly each word contains one or more pointed letters. As shown in Figure 16, we can see how many pointed letters are found in each text and calculate how many secret letters we can insert in this text. We found 7020 pointed letters in text 1 that means 877.5 bits so we can insert 109 secret letters. In the second text, we found 5609 pointed letters which are 701.125 bits so we can insert 87 secret letters in text 2, Also, in-text 3 we found 8394 pointed letters it's equal 1049.25 bits so we can insert 131 secret letters, in the fourth text we found 5750 pointed letters which are 718.75 bits so we can insert 89 secret letters, and finally, in-text 5 we found 7076 pointed letters which are 884.5 bits so we can insert 110 secret letters.

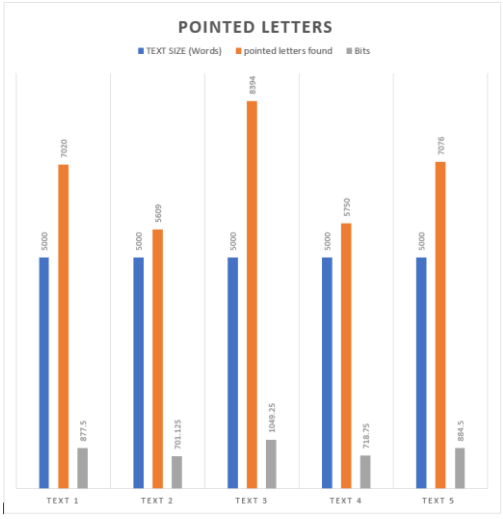

Figure 16. pointed letters in five texts

E. MULTIPOINT ARABIC LETTER

There are five letters in the Arabic language that contains more than one point, listed in Figure 17.

1	ت	Ta
2	ث	Tha
3	ش	Sheen
4	ق	Qaf
5	ي	Ya

Figure 17. multipoint Arabic letters

We started implementing the pointed letter technique to the same five texts was used in the previous techniques. We found out that there 16,898 multipoint letters in a text of 25,000 words.

As shown in Figure 18, we can see how many multipoint letters are found in each text and calculate how many secret letters can we insert in each text. We found 3605 multipoint letters in text 1 which are 450.62 bits so we can insert 56 secret letters, in the second text we found 2777 multipoint letters which are 347.12 bits so we can insert 43 secret letters, in the third text we found 4647 multipoint letters which are 580.87 bits so we can insert 72 secret letters, in the fourth text we found 2481 multipoint letters which are 310.12 bits so we can insert 38 secret letters, and finally, in-text five we found 3388 multipoint letters which are 423.5 so we can insert 52 secret letters.

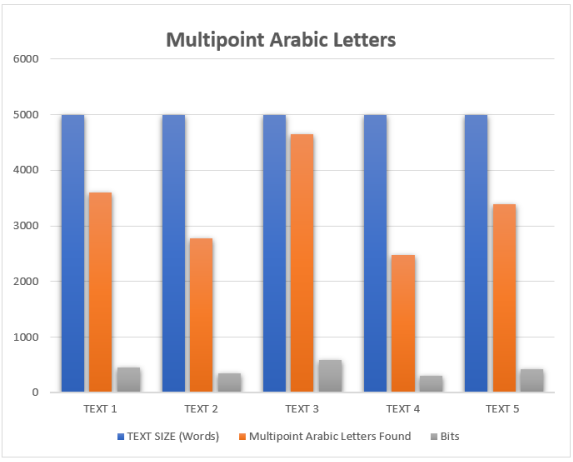

Figure 18. multipoint letter in five texts

F. KASHIDA

The Kashida technique uses the Kashida (-) which extends the Arabic word without changing its meaning, it is mostly used in letters or titles which makes it strange to see a text full of extended words [20], the main disadvantage of the Kashida technique is that it uses the Arabic pointed letters as a "1" value and the non-pointed letters as zeros which makes it easy for any reader to extract the hidden bits by just looking if the letter after the Kashida is a pointed letter or not, thus, break the main purpose of the steganography technique, a good solution for this problem would be to redesign the technique were the pointed letter and non-pointed letters would not be considered when doing the steganography, the letters should be put in a pseudorandom number generator that will choose between 28 letters 14 letters that would represent "1" value and 14 letters that would represent a "0" value, this way the technique will skew any sequences between the letters, make it impossible to guess the values of letter based on their shape and reach steganography's ultimate goal which is to hide the secret message without allowing any one to be able to extract the hidden information without knowing details of the technique that was used to hide the secret message.

G. REVERSE FATHA

The ‘’Fatha’’ is used in most Arabic words which makes it have a high capacity compared to La technique, but Reverse Fatha shares same problems with La technique in modifying the shape of the Fatha will make it obvious that something is not right about the text. And as shown in Figure 19, we have implemented the Reverse Fatha technique to five random text files containing Arabic words and sentences to study how many times ‘’Fatha’’ has been used and determine on average how many bytes in total does the five text has. we estimated to use of ‘’Fatha’’ used in the texts which on average has about 5000 words and 25000 characters, and we found out that 30% of the characters in the words contains ‘’Fatha’’ within them, which is around 1 500 times used in a 5000 words text. As shown in Figure 19 below, different random texts could have different numbers of Fatha found in the texts, Firstly, text 1 has 924.65 bits of Fatha when we convert it to bytes, we will get 115.5 bytes, Secondly, text 2 has about 647.5 bits Fatha and when converted to bytes will result in 80.9 bytes, Thirdly in text 3

we found about 1119.75 bits Fatha and when converted to bytes will result in 139.9 bytes, Fourthly, in text 4 726.075 bits Fatha that equals 90.7 bytes, and finally on text 5 we found out about 805.8 bit which equals to approximately 100.725 bytes.

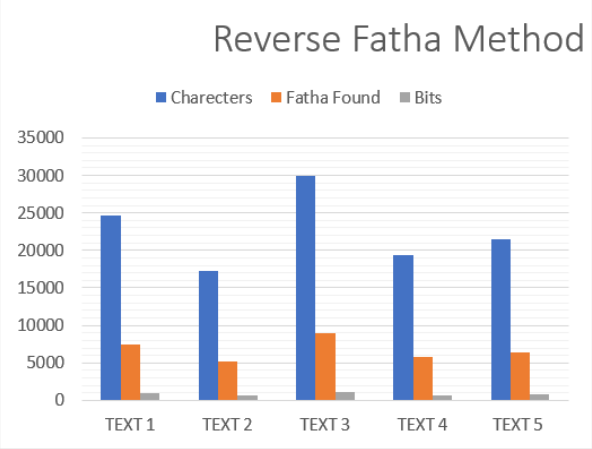

Figure 19. Reverse Fatha Technique

In Total, if we combine all the bytes found in the different 5 text files which contains about 250000 words, we found out that they contain around 527.72 bytes in total, which makes the Reverse Fatha technique a very good option for Arabic Steganography, because ''Fatha'' can be used in almost any Arabic text, and because Fatha is a very popular ''Haraka'' and used widely in Arabic language, and most Arabic words in sentences contains ''Fatha'' within them, unlike ''La technique'' which need a big text in order for it to work, Reverse Fatha can be used on smaller or larger texts because of that. But on the other hand, the problem of modifying the ''Fatha'' and changing its look can be suspicious to the trained eyes.

5. CONCLUSION

In this research, we thoroughly studied and evaluated six Arabic text steganography techniques and identified their differences. We also, suggested improvements to overcome these flaws and fill the gap. The improved techniques were evaluated against the plain ones and proved to outperform their performance in terms of capacity. The non-printable letters technique showed the best performance compared to other techniques in terms of capacity and obstacles, it did not change the appearance of the cover text or rely on the Arabic language grammar to hide data, we suggest adding cryptography to this technique, which adds extra layer of security to it.

REFERENCES

[1] A. Cheddad, J. Condell, K. Curran and P. Mc Kevitt, "Biometric Inspired Digital Image Steganography," in Engineering of Computer-Based Systems, IEEE International Conference on the, null, 2008 pp. 159-168. doi: 10.1109/ECBS.2008.11

[2] A. Munasinghe, A. Dharmaratne and K. De Zoysa, "Video steganography," 2013 International Conference on Advances in ICT for Emerging Regions (ICTer), Colombo, 2013, pp. 56-59, doi: 10.1109/ICTer.2013.6761155.

[3] A. Odeh, A. Alzubi, Q. B. Hani and K. Elleithy, "Steganography by multipoint Arabic letters," 2012 IEEE Long Island Systems, Applications and Technology Conference (LISAT), Farmingdale, NY, 2012, pp. 1-7, doi: 10.1109/LISAT.2012.6223209.

[4] A. T. Bhole and R. Patel, "Steganography over video file using Random Byte Hiding and LSB technique," 2012 IEEE International Conference on Computational Intelligence and Computing Research, Coimbatore, 2012, pp. 1-6, doi: 10.1109/ICCIC.2012.6510230.

[5] Al-Nofaie, Safia & Gutub, Adnan & Al Ghamdi, Manal. (2019). Enhancing Arabic Text Steganography for Personal Usage Utilizing Pseudo-Spaces. Journal of King Saud University - Computer and Information Sciences. 10.1016/j.jksuci.2019.06.010.

[6] Alshahrani, H., & Weir, G. (2017). Hybrid Arabic text steganography. IEEE CIT 2017.

[7] Altaay, S. Sahib and M. Zamani, "An Introduction to Image Steganography Techniques," in 2012 International Conference on Advanced Computer Science Applications and Technologies (ACSAT), Kuala Lumpur, 2012 pp. 122-126.

[8] C. Biswas, U. D. Gupta and M. M. Haque, "An Efficient Algorithm for Confidentiality, Integrity and Authentication Using Hybrid Cryptography and Steganography," 2019

[9] C.-H. Yang, C.-Y. Weng, S.-J. Wang, Member, IEEE and H.-M. Sun, "Adaptive Data Hiding in Edge Areas of Images with Spatial LSB Domain Systems", IEEE Transactions on Information Forensics and Security, vol. 3, no. (2008) September 3, pp. 488-497. doi: 10.1109/ACSAT.2012.25 doi: 10.1109/TDSC.2019.2933621

[10] F. Djebbar, B. Ayad, H. Hamam and K. Abed-Meraim, "A view on latest audio steganography techniques," 2011 International Conference on Innovations in Information Technology, Abu Dhabi, 2011, pp.409-414, doi: 10.1109/INNOVATIONS.2011.5893859.

[11] G. L. Smitha and E. Baburaj, "A survey on image steganography based on block-based edge adaptive based on Least Significant Bit Matched Revisited (LSBMR) algorithm," 2016 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), Kumaracoil, 2016, pp. 132-139, doi: 10.1109/ICCICCT.2016.7987931.International Conference on Electrical, Computer and Communication Engineering (ECCE), Cox'sBazar, Bangladesh, 2019, pp. 1-5, doi: 10.1109/ECACE.2019.8679136.

[12] Jusoh, Suhaibah & Mustapha, Aida & Ismail, Azizan & Din, Roshidi. (2020). A review of arabic text steganography: past and present. Indonesian Journal of Electrical Engineering and Computer Science. 17. 1040. 10.11591/ijeecs.v17.i2.pp1040-1046.

[13] K. J. Velmurugan and S. Hemavathi, "Video Steganography by Neural Networks Using Hash Function," 2019 Fifth International Conference on Science Technology Engineering and Mathematics (ICONSTEM), Chennai, India, 2019, pp. 55-58, doi: 10.1109/ICONSTEM.2019.8918877.

[14] M. Dixit, N. Bhide, S. Khankhoje and R. Ukarande, "Video Steganography," 2015 International Conference on Pervasive Computing (ICPC), Pune, 2015, pp. 1-4, doi: 10.1109/PERVASIVE.2015.7087159.

[15] M. H. Shirali-Shahreza and M. Shirali-Shahreza, "A New Approach to Persian/Arabic Text Steganography," 5th IEEE/ACIS International Conference on Computer and Information Science and 1st IEEE/ACIS International Workshop on Component-Based Software Engineering,Software Architecture and Reuse (ICIS-COMSAR'06), Honolulu, HI, 2006, pp. 310-315, doi: 10.1109/ICIS-COMSAR.2006.10.

[16] M. S. Abbas, S. S. Mahdi and S. A. Hussien, "Security Improvement of Cloud Data Using Hybrid Cryptography and Steganography," 2020 International Conference on Computer Science and Software Engineering (CSASE), Duhok, Iraq, 2020, pp. 123-127, doi: 10.1109/CSASE48920.2020.9142072.

[17] M. Shirali-Shahreza and M. H. Shirali-Shahreza, "An Improved Version of Persian/Arabic Text Steganography Using "La" Word," 2008 6th National Conference on Telecommunication Technologies and 2008 2nd Malaysia Conference on Photonics, Putrajaya, 2008, pp. 372-376, doi: 10.1109/NCTT.2008.4814305.

[18] M. Shirali-Shahreza, "Text Steganography by Changing Words Spelling," 2008 10th International Conference on Advanced Communication Technology, Gangwon-Do, 2008, pp. 1912-1913, doi: 10.1109/ICACT.2008.4494159.

[19] Memon, Mujtaba.S & Shah, Dr. (2015). A Novel Text Steganography Technique to Arabic Language Using Reverse Fat5Th5Ta. Pakistan Journal of Engineering, Technology & Science. 1. 10.22555/pjets.v1i2.167.

[20] Odeh, Ammar & Elleithy, Khaled. (2013). STEGANOGRAPHY IN ARABIC TEXT USING ZERO WIDTH AND KASHIDHA LETTERS.

[21] P. P. Balgurgi and S. K. Jagtap, "Intelligent processing: An approach of audio steganography," 2012 International Conference on Communication, Information & Computing Technology (ICCICT), Mumbai, 2012, pp. 1-6, doi: 10.1109/ICCICT.2012.6398182.

[22] Patiburn, Sivabalan & Iranmanesh, Vahab & Teh, Phoey. (2017). Text Steganography using Daily Emotions Monitoring. International Journal of Education and Management Engineering. 7. 1-14. 10.5815/ijeme.2017.03.01.

[23] Paul, Aryya. "Steganography Tutorial-A Complete Guide For Beginners." Medium, Edureka, 9 Sept. 2020, medium.com/edureka/steganography-tutorial-1a3c5214a00f.

[24] R. Balaji and G. Naveen, "Secure data transmission using video Steganography," 2011 IEEE INTERNATIONAL CONFERENCE ON ELECTRO/INFORMATION TECHNOLOGY, Mankato, MN, 2011, pp. 1-5, doi: 10.1109/EIT.2011.5978601.

[25] R. Tanwar and M. Bisla, "Audio steganography," 2014 International Conference on Reliability Optimization and Information Technology (ICROIT), Faridabad, 2014, pp. 322-325, doi: 10.1109/ICROIT.2014.6798347.

[26] S. Sharma, A. Gupta, M. C. Trivedi and V. K. Yadav, "Analysis of Different Text Steganography Techniques: A Survey," 2016 Second International Conference on Computational Intelligence & Communication Technology (CICT), Ghaziabad, 2016, pp. 130-133, doi: 10.1109/CICT.2016.34.

[27] Sabir, Aliea & Akeel, Wid. (2012). A New Text Steganography Method by Using Non-Printing Unicode Characters and Unicode System Characteristics in English/Arabic documents. 3.

[28] [Source code] Github (2017) Arabic Text Steganography https://github.com/mosaedb/arabic-text-steganography

Adeel Alshamsi Applied Bachelor: Information Security Networks and Cyber Security technology, Abu Dhabi Polytechnic, United Arab Emirates, 2021.

Mohammed Alkhoori Applied Bachelor: Information Security Networks and Cyber Security technology, Abu Dhabi Polytechnic, United Arab Emirates, 2021.

Salem Albaloushi Applied Bachelor: Information Security Networks and Cyber Security technology, Abu Dhabi Polytechnic, United Arab Emirates, 2021.

Hamed Almheiri Applied Bachelor: Information Security Networks and Cyber Security technology, Abu Dhabi Polytechnic, United Arab Emirates, 2021.

Nedal Ababneh is the head of Information Security Engineering Technology Department (ISET) at Abu Dhabi Polytechnic. He received his PhD in 2009 from The University of Sydney, Australia and Master Degree in 2004 from The University of New South Wales, Australia, both in Computer Science and Engineering. From 2017 to 2019 he served as Senior Lecturer and Program Chair of Undergraduate Program of IT (Network Engineering, and Web and Mobile Application Development) at Victoria University, Australia. His main research interests include Internet of Things, Wireless Sensor Networks, Wireless Body Area Networks, Blockchain, Network and Information Security, and Steganography. Dr. Ababneh has published a number of journal and conference papers in top international venues. Dr. Ababneh is a member of IEEE, IEEE Communication Society (ComSoc), and Australian Computer Society (ACS).