Republic of Yemen

Ministry of Higher Education
& Scientific Research

Al-Rayan University

Faculty of Higher Studies

Data Hiding Enhancement in Arabic Text

Thesis Submitted to Al-Rayan University to Complete the
Requirements of Obtaining a Master's Degree in Information Technology

By
Abdulaziz Seleh Abdullah Al-Dini

Supervised by

Dr. Khalid Kayid Shafel

Dr. Naziha Mohammed Al-Aidroos

1442 / 2021

Republic of Yemen

Ministry of Higher Education
& Scientific Research

Al-Rayan University

Faculty of Higher Studies

جـــامـــعـــة الـــريـــــان
AL-RAYAN UNIVERSITY

Data Hiding Enhancement in Arabic Text

Thesis Submitted to Al-Rayan University to Complete the Requirements of Obtaining a Master's Degree in Information Technology

By
Abdulaziz Seleh Abdullah Al-Dini

Supervised by

Dr. Khalid Kayid Shafel **Dr. Naziha Mohammed Al-Aidroos**

1442 / 2021هـ

Approval of the Scientific Supervisor

I certify that this master's dissertation titled ,

(Data Hiding Enhancement in Arabic Language)

submitted by the student, **Abdulaziz Saleh Abdullah Al-Dini**

has been completed in all its stages under my supervision and so I nominate it for discussion.

Supervisor : Dr. Naziha Mohammed Al-Aidroos

Signature : .....

Date : 27/ 1 / 2021

Approval of the Proofreader

I certify that the master's dissertation titled ,

(Data Hiding Enhancement in Arabic Language)

submitted by the student, **Abdulaziz Saleh Abdullah Al-Dini**

has been linguistically reviewed under my supervision and has become in scientific style and clear from linguistic errors and for that I sign.

ProofreaderA : Abdullah Amer Al-kathiri

Academic Title : Assistant Professor

University : Hadhramout University

Signature : .....

Date : 5 / 2 /2021

The Discussion Committee Decision

Based on the decision of the President of the University No. (2) in the year 2021 regarding the nomination of the committee for discussing the master's thesis entitled (**Data Hiding Enhancement in Arabic Text**) for the researcher **Abdulaziz Saleh Abdullah Al-Dini**. We, the head of the discussion committee and its members, acknowledge that we have seen the aforementioned scientific thesis and we have discussed the student in its contents and what related to it.

Chairman of the Committee

Associate Professor

Dr. Saeed Mohammed Baneamoon

Signature: .....

Committee member

Associate Professor

Dr. Naziha Mohammed Al-Aidroos

Signature: .....

Committee member

Assistant Professor

Dr. Mohammed Abdullah Bamatraf

Signature: .....

قال الله تعالى: -

أَمَّنْ هُوَ قَانِتٌ آنَاءَ اللَّيْلِ سَاجِدًا وَقَائِمًا يَحْذَرُ الْآخِرَةَ وَيَرْجُو رَحْمَةَ رَبِّهِۗ قُلْ هَلْ يَسْتَوِي الَّذِينَ يَعْلَمُونَ وَالَّذِينَ لَا يَعْلَمُونَۗ إِنَّمَا يَتَذَكَّرُ أُولُو الْأَلْبَابِ ۝

[سورة الزمر، 9]

Dedication

I dedicate this research for

The soul of my father

My beloved mother

My beloved wife

My heart sisters

My heart daughters

My friends

And to all the people who helped me do this research.

Acknowledgment

First of all, I thank God for giving me the strength and the ability to complete this study.

Second, I would like to express my deep gratitude to my thesis supervisors:

Dr. Khalid Kayid Shafel whose expertise, understanding, and patience added considerably to my graduate experience.

Dr. Naziha Mohammed Al-Aidroos for her continuous support, encouragement and her help throughout my studies and continuous advice to reach the end of this research.

Also I would like to thank ***CARE ASSOCIATION OF SCIENCE STUDENT WARFARE*** and ***SELAH FOUNDATION FOR DEVELOPMENT*** for them materialist and moral support and them help throughout my studies.

In addition, I would like to thank my parents, and my wife for their support and encouragement throughout the entire academic life.

I would like to thank the Information Technology Faculty members, my colleagues at the department of Information Technology.

Abdulaziz Saleh Al-Dini

Abstract

Recently, information security has become a very important topic for researchers as well as military and government officials. For secure communication, it is necessary to develop novel ways to hide information, for this purpose, steganography is usually used to send secret information to their destination using different techniques. The aim of this thesis, is to provide a new text steganography method. Hidden information in text files is difficult to discover as text data has low redundancy in comparison to other mediums of steganography. Hence, we use an Arabic text to hide secret information using a combination of Unicode character's zero-width-character (ZWC), zero-width-joiner (ZWJ) and pseudo-space (PS) in the proposed algorithm. The experimental results show increasing in hidden data capacity per word in comparison to the recently proposed algorithms. The major advantage of our proposed algorithm over previous researches is the high visual similarity in both cover and stegotext which can reduce the attention of intruders.

تحسين البيانات المخفية في النصوص العربية

الملخص

في الآونة الأخيرة، أصبح أمن المعلومات موضوعاً بالغ الأهمية بالنسبة للباحثين فضلا عن المسؤولين العسكريين والحكوميين. وللتواصل الآمن من الضروري استحداث طرق جديدة لإخفاء المعلومات، ولهذا الغرض، عادة ما يستخدم علم إخفاء البيانات لإرسال معلومات سرية إلى مقصدها باستخدام تقنيات مختلفة.

والهدف من هذه الأطروحة هو توفير طريقة جديدة لعلم اخفاء البيانات بالتقنية النصية. من الصعب اكتشاف المعلومات المخفية في الملفات النصية، حيث أن البيانات النصية ذات إسهاب اقل بالمقارنة مع تقنيات أخرى من علم إخفاء البيانات. ومن هنا فإننا نستخدم نصا عربيا لإخفاء المعلومات السرية باستخدام مزيج من الحرف ذو العرض الصفري، والانضمام ذو العرض الصفري، والفضاء الزائف في الخوارزمية المقترحة. وتظهر النتائج التجريبية زيادة في سعة البيانات المخفية لكل كلمة مقارنة بالخوارزميات المقترحة مؤخرا. الميزة الرئيسية لخوارزميتنا المقترحة على البحوث السابقة هو التشابه البصري العالي في كل من الغلاف ونص الاخفاء الذي يمكن أن يقلل من انتباه الدخلاء.

Table of Contents

	Page
Dedication	A
Acknowledgments	B
Abstract	C
Abstract (in the second language)	D
Contents	E
List of Tables	G
List of Figures	H
List of Abbreviations	I
CHAPTER ONE: INTRODUCTION	1
1.1 Introduction	2
1.2 Motivations and Research Problem	3
1.3 Objective	3
1.4 Scope and Limitations	3
1.5 Thesis Organization	4
CHAPTER TWO: BACKGROUND	5
2.1 Overview of Steganography	6
2.2 Types of Steganography	7
2.2.1 Cover Type Classification	7
2.2.2 Method of Hiding Data Classification	8
2.3 Text steganography	9
2.3.1 Introduction	9
2.3.2 Existing Approaches	12
2.4 Arabic Text Steganography	15
2.4.1 Arabic language characteristics	15
2.5 Related Works	16

CHAPTER THREE: RESEARCH METHODOLOGY 24

3.1 Methodology 25

3.2 Inserting Start-End Bits into Cover Text 26

3.3 How To Embed Secret Bits Into Cover Text 26

3.4 The Proposed Scheme 27

3.4.1 Embedding Phase 27

3.4.2 Extracting Phase 28

3.5 Procedure of Hiding Secret Data into Cover Text 29

3.6 Extracting Algorithm To Retrieve Secret Message From Stego-Text 32

CHAPTER FOUR: RESULTS AND DISCUSIONS 35

4.1 Results and Discussions 36

CHAPTER FIVE: CONCLUSION AND FUTURE WORK 42

5.1 Conclusion 43

5.2 Future Work 43

References 44

List of Publications 51

List of Tables

Table No.	Title	Page
2.1	Representing relation between shifting and distance to letter format	19
2.2	Letter shapes according to its position in the word	22
2.3	Summarize of related works	23
4.1	Results of data hiding capacity on different cover text using our algorithm and comparison with other used approaches	37
4.2	Results and comparison of data hiding capacity on one selected cover text with our algorithm and other approaches	41

List of Figures

Figure No.	Caption	Page
2.1	Types of Steganography	8
2.2	The Mechanism of Text Steganography	10
2.3	Three basic categories of text steganography	12
2.4	Use of normal and reverse Fatah	17
2.5	Example of hiding data in diacritics	18
2.6	Example of Alotaibi's method	19
2.7	Example of the method presented in (Alhusban et al., 2017)	20
2.8	Data hiding with pointed and un-pointed letters by inserting Arabic extension	21
2.9	Data hiding with pointed and un-pointed letters by inserting Arabic extension	23
3.1	Procedural flow chart for embedding secret bits into Arabic cover text file	30
3.2	Reverse algorithm flow chart for extracting secret message from stego-text file	33
4.1	Comparison of data hiding capacity with other steganographic methods	40
4.2	Hidden data capacity against other steganographic methods with same cover text	41

List of Abbreviations

Symbols	Nomenclatures
ZWC	Zero-Width-Character
ZWJ	Zero-Width-Joiner
PS	Pseudo-Space
ISP	Internet Service Provider
HVS	Human Visual System
CFG	Context-Free Grammars
GNF	Greibach Normal Form
OCR	Optical Character Recognition
SSCE	Secret Steganographic Code for Embedding
CSS	Cascading Style Sheet
ZWNJ	Zero-Width Non-Joiner

Chapter 1
Introduction

Introduction

1.1 Introduction

In the current usage of data, like civilization, education, and military, these data must be secure. One of the data securing mechanisms is steganography; steganography is a mechanism of hiding the secret message in a non-secret cover media (message). It has many methodologies based on the nature of secret message and covering messages (Al-Alwani et al., 2007).

Steganography is the science of hiding information. It is often confused with cryptography as both are used to protect confidential information. The difference between the two is in the appearance of the processed output; the output of steganography operation is not obviously visible but in cryptography the output is twisted so that it can draw attention. If a nefarious government or Internet service provider (ISP) is looking for encrypted messages, they can easily find them.Whereas the goal of cryptography is to make data unreadable by a third party, the goal of steganography is to hide the data from a third party.

Steganalysis is the process to detect the presence of steganography. The key terminologies in the context of steganography are: plaintext which is the original secret message that needs to be communicated; cover media is the larger and harmless looking data which is used as container for the plaintext; and the stego media which is the data generated after embedding the plaintext into the cover media.

Steganography mainly employs redundant data in the carrier media to hide the secret message. In most cases, text steganography is more challenging due to the lack of redundant data in text files (Shirali-Shahreza and Shirali-Shahreza, 2008). In addition, the structure of text documents is almost identical to their looks so any change may be visible. In general, text steganography depends mainly on the language characteristics. It is known that the grammatical and orthographic characteristics differ from one language to another.

1.2 Motivation & Research Problem

Recently, information security has become a very important topic for researchers as well as military and government officials. For secure communication, it is necessary to develop novel ways to hide information, for this purpose, steganography is usually used to send secret information to their destination using different techniques.

Steganography uses some safe carrier called stego media in which the message is embedded. It may use image, audio, video, or text to hide the existence of the message.

Of all carrier media, images are the most widely used carrier to hide data. However, the text has some advantages that make it comparable to images (Gutub and Fattani, 2007):

- It has less memory usage.
- It is faster than other methods.
- It is easier in communication.

Recently, there have been several successful attempts to design text steganography schemes for many languages including English, Chinese, Hindi, Bengali, Urdu, and Arabic.

The richness of the Arabic language by its characteristics makes it a strong candidate for hiding information.

Many methods and algorithms were set to hide secret data into cover media. Some of those algorithms provide a small capacity to hiding information and not keeping the quality of text, and some provide that. So, it is necessary to develop more new algorithms with more capacity and quality.

1.3 Objective

The main objective is to develop an inventor approach of hiding secret data inside an Arabic text with high security, preservation of the text quality and extra capacity compared to current algorithms.

1.4 Scope and Limitation

This research will focus only on steganography in Arabic text.

We will hide bits in all types of letters in Arabic text except last letter in each word.

1.5 Thesis Organization

The thesis is organized as follows:

Chapter One gives an overall introduction about the steganography, research problem, and the thesis objective.

Chapter Two introduces a general background of steganography as a method of covert communication. In addition, it describes different types of steganography and cover types. It describes more types of text steganography approaches and categories. Additionally, it discusses some of the most related works and gives a comparison between them depending on some approved evaluation aspects.

Chapter Three In this chapter, we explain our proposed Data Hiding Enhancement in Arabic Text, and approach which we followed in this research.

Chapter Four In this section, we described our experiments on Arabic text for data hiding process and their results.

Chapter Five this chapter summarizes the main points of the thesis and lists the main conclusions of the work. Finally, it talks about future work.

Chapter 2

Background

Background

This chapter is organized into five sections. Section 2.1, introduces a general background of steganography as a method of covert communication. Also it describes different types of steganography and cover types. Section 2.2 we provide an overview of text steganography approaches and categories. In section 2.3, talks about text steganography. Section 2.4, shows a review about Arabic Text Steganography. Section 2.5, reviews some related work presented for Arabic text steganography.

2.1 An Overview of Steganography

Security of information is a significant issue of information technology and communication issues. It locates the privacy of its existence and/or the privacy of how to decode it. Cryptography, watermarking and Steganography can be used in information security.

The cryptography techniques hide secret information by encrypting it using encryption key (s). The output of encryption is chipper text or the secret information in unreadable format. This may draw the attention of attackers to the existence of confidential information.

Digital watermarking is the process of embedding information into digital multimedia content so that the information (watermark) can be extracted or detected for different purposes including copy prevention and control (Hussein, 2015).

Steganography is the art of hiding information by different ways which avoid the discovery of hidden messages. Steganography, derived from Greek, literally means "covered writing" (Greek words "stegos" meaning "cover" and "gratia" meaning "writing")(Das & Tuithung, 2012). The thesis area is about steganography.

A steganographic system involves a cover media which the secret information is embedded inside it. The embedding process produces medium stego replacing information with data from hidden message. To hide the information, steganography gives a great opportunity in such a way that no one can know the existence of a hidden message (M. Karim, 2011).

The main objective of steganography is to avoid the attention to the transmission of hidden information. If uncertainty occurred, then hackers will be noted that there is a

change in the sent message and then they will try to extract of the hidden information (Wu et al., 2010).

2.2 Types of Steganography

Steganography ensures the confidentiality of data objects within the digital carriers such as images, audio and video so that they can not be easily detected by a human visual system (HVS).

There are two ways for the general classification of steganographic systems. The first is based on the type of cover file, while the second approach is based on a method of hiding data (Abo Mousa, 2017).

2.2.1 Cover Type Classification:

There are five types of steganography according to the object which is used for embedding secret data. These types are briefly described as given in Figure2.1 (Muhammad et al., 2015).

1. **Text steganography**: It is one of the most common methods of steganography. It hides a secret message into a text file. The appearance of the Internet and different types of digital file formats has a little importance.
2. **Image steganography**: Images are used as a popular cover object for steganography. The message was embedded in a digital image using an algorithm, using a secret key. It is sent resulting stego image to the receiver, to retrieve the hidden message by the extraction algorithm.
3. **Audio steganography:** It is concerned with embedding information in an innocuous cover speech in a secure and robust manner. Communication, transmission security and robustness are essential for the transfer of vital information for the intended sources while denying access by unauthorized persons. Therefore, an audible sound can be inaudible in the presence of another louder audible sound. This feature allows selecting the channel to hide the information. Audio steganography software can embed messages in WAV and MP3 sound files.
4. **Video steganography**: It is a technique to hide any type of files in any extension into a carrrying Video file.

5. **Protocol steganography**: This is used for embedding information within network protocols such as TCP/IP. Information will be hidden in the header of a TCP/IP packet in fields that can be either optional or never used (Devi, 2013).

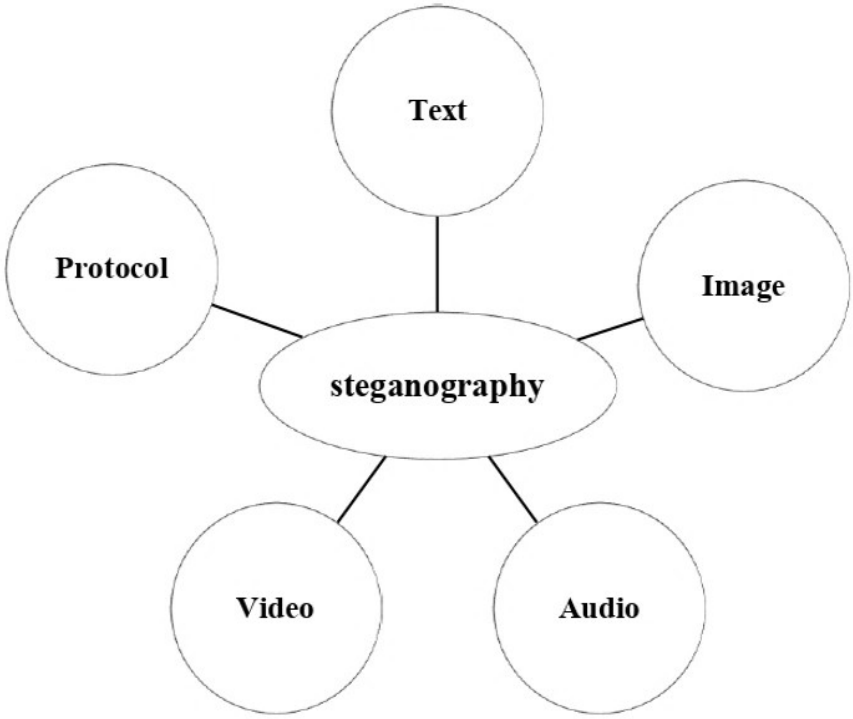

Figure (2.1): Types of Steganography

2.2.2 Method of Hiding Data Classification:

Hiding information can be classified according to the method used to hide secret data. Moreover, this approach of classification in steganography is the most preferred in the research community approach for hiding the information, there are three different ways to hide secret data in a cover files: insertion-based, substitution-based and generation-based method.

1. Insertion-Based Method

Insertion-based method hide data in sections of a file that have been ignored by the processing application and not to modify bits that define that the contents are relevant to the end user file(Weiss, 2012). This method inserts secret data within the cover file, therefore the resulted stego file size will be larger than the cover file size. The main advantage of this method is that the contents of the cover file will not change after the embedding process

because this method depends on the accumulation or to add the secret data to the cover file (Abo Mousa, 2017).

2. Substitution-Based Method

In a Substitution-based algorithm, the secrete information is replaced by the most insignificant bit of the cover file over information that identifies the original content of the file with the new data in a way that causes the least amount of distortion. However, the file size does not change during the implementation of the algorithm, and the amount of data that can be hidden includes unlimited amounts of insignificant bits in the file (Abo Mousa, 2017).

3. Generation-Based Method

This method does not need a cover file like insertion and substitution methods, it uses secret data to generate a suitable stego files. This steganography detection technique is based on comparing cover files with stego files. One advantage of this method is to prevent such kind of detection. So the major limitation of this method is in limited stego files that can be generated. Moreover, the generated stego files might be unrealistic files for end users (e.g. an image contains different shapes and colours without any sense or a text without any meaning) (Abo Mousa, 2017).

2.3 Text steganography

2.3.1 Introduction:

Steganography can be classified into image, text, audio and video steganography depending on the cover media used to embed secret data. Text steganography can involve anything (Benett, 2014):

- Changing the formatting of an existing text.
- Changing words within a text.
- Generating random character sequences.
- Using context-free grammars to generate readable texts.

Text steganography is believed to be the trickiest due to deficiency of redundant information which is present in image, audio or a video file. The structure of text documents is identical with what we observe, while in other types of documents such as in picture, the structure of

document is different from what we observe. Therefore, in such documents, we can hide information by introducing changes in the structure of the document without making a notable change in the concerned output (Shirali-Shahreza & Shirali-Shahreza, 2007). Unperceivable changes can be made to an image or an audio file, but, in text files, even an additional letter or punctuation can be marked by a casual reader (Bender et al., 2007). Storing text file require less memory and its faster as well as easier communication makes it preferable to other types of steganographic methods (Shirali-Shahreza & Shirali-Shahreza, 2006).

Figure (2.2) shows the basic text steganography mechanism. Firstly, a secret message (or an embedded data) will be concealed in a cover-text by applying an embedding algorithm to produce a stego-text. The stego-text will then be transmitted via communication channel, e.g. Internet or mobile device to a receiver. To recover the secret message sent by the sender, the receiver needs to use a recovering algorithm that is parameterised by a stego-key to extract the secret message. A stegokey is used to control the hiding process as well as to restrict detection and/or recovery of the embedded data to parties who know it (Petitcolas et al., 1999).

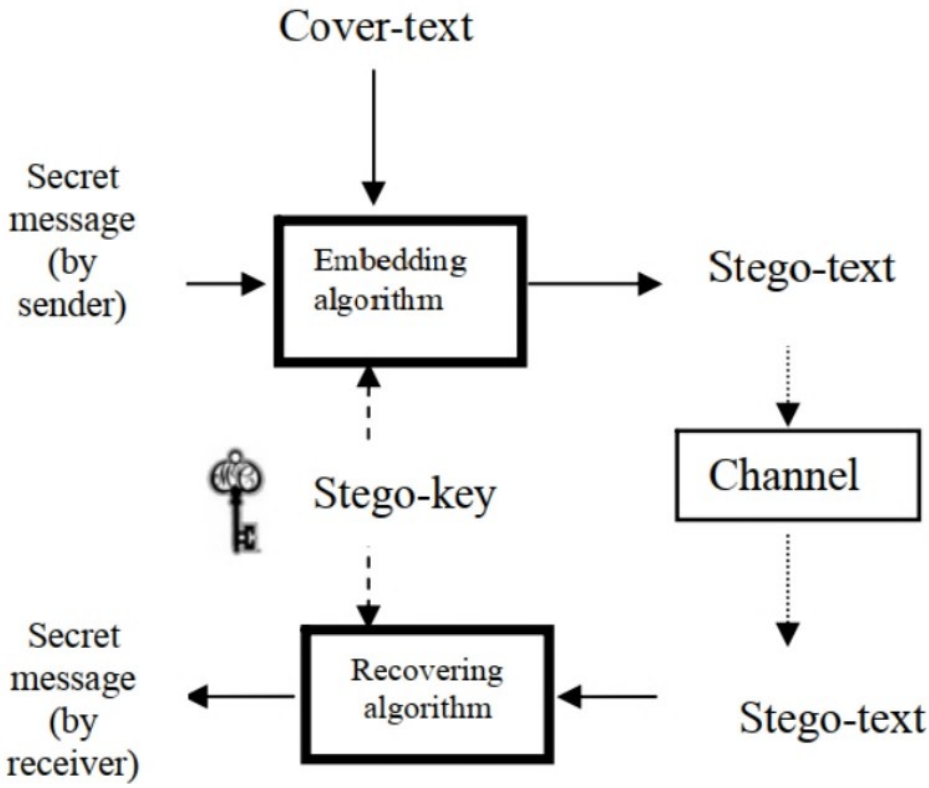

Figure (2.2): The Mechanism of Text Steganography

Text steganography can be classified into three categories - format-based, random and statistical generation and linguistic method (Benett, 2014). As shows in figure (2.3).

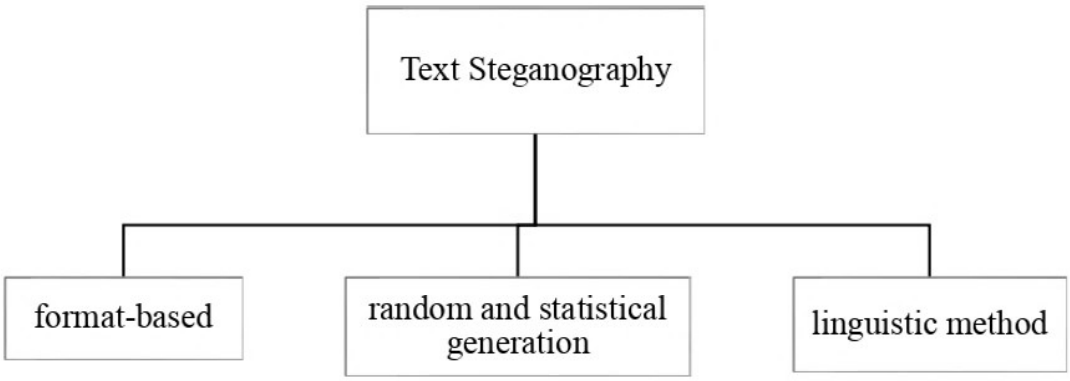

Figure (2.3): Three basic categories of text steganography

1. **Format Based Methods**

Format based methods involve altering physically the format of text to conceal the information. This method has certain flaws. If the stego file is opened with a word processor, misspellings and extra white spaces will get detected. Changed fonts sizes can arouse suspicion to a human reader. Additionally, if the original plaintext is available, comparing this plaintext with the suspected steganographic text will make manipulated parts of the text quite visible (Benett, 2014).

2. **Random and Statistical Generation**

In order to avoid comparison with a known plaintext, steganographers often resort to generating their own cover texts (Benett, 2014). One method is concealing information in random looking sequence of characters. In another method, the statistical properties of word length and letter frequencies are used in order to create words which will appear to have same statistical properties as actual words in the given language (Por & Delina, 2008; Por et al., 2008).

3. Linguistic Steganography

Linguistic steganography specifically considers the linguistic properties of generated and modified text, and in many cases, uses linguistic structure as the space in which messages are hidden (Benett, 2014). Context-Free Grammars (CFG) create tree structure which can be used for concealing the bits where left represents '0' and right branch corresponds to '1'. A grammar in Greibach Normal Form (GNF) can also be used where the first choice in a production represents bit 0 and the second choice represents bit 1. This method has some drawbacks. First, a small grammar will lead to lot of text repetition. Secondly, although the text is syntactically flawless, but there is a lack of semantic structure. The result is a string of sentences which have no relation to one another (Benett, 2014).

2.3.2 Existing Approaches

In this sub-section, we will present some of the popular approaches of text steganography.

A. Line Shift

In this method, secret message is hidden by vertically shifting the text lines to some degree (Shirali-Shahreza & Shirali-Shahreza, 2008; Shirali-Shahreza & Shirali-Shahreza, 2006). A line marked has two unmarked control lines one on either side of it for detecting the direction of movement of the marked line (Low et al., 1995). To hide bit 0, a line is shifted up and to hide bit 1, the line is shifted down (Cummins et al., 2004). Determination of whether the line has been shifted up or down is done by measuring the distance of the centroid of marked line and its control lines (Low et al., 1995). If the text is retyped or if a character recognition program (OCR) is used, the hidden information would get destroyed. Also, the distances can be observed by using special instruments of distance assessment (Shirali-Shahreza & Shirali-Shahreza, 2006).

B. Word Shift

In this method, secret message is hidden by shifting the words horizontally, i.e. left or right to represent bit 0 or 1 respectively (Cummins et al., 2004). Words shift are detected using correlation method that treats a profile as a waveform and decides whether it originated from a waveform which middle block has been shifted left or right (Low et al., 1995). This method can be identified less, because change of distance between words to fill a line is quite

common (Shirali-Shahreza & Shirali-Shahreza, 2008; Shirali-Shahreza & Shirali-Shahreza, 2006). But if someone knows the algorithm of distances, he can compare the stego text with the algorithm and obtain the hidden content by using the difference. Also, retyping or using OCR programs destroys the hidden information (Shirali-Shahreza & Shirali-Shahreza, 2008; Shirali-Shahreza & Shirali-Shahreza, 2006).

C. Syntactic Method

This technique uses punctuation marks such as full stop (.), comma (,), etc. to hide bits 0 and 1. But the problem with this method is that it requires identification of correct places to insert punctuation marks (Shirali-Shahreza & Shirali-Shahreza, 2008; Shirali-Shahreza & Shirali-Shahreza, 2006). Therefore, care should be taken in using this method as readers can notice improper use of the punctuations (Bender et al., 2007).

D. White Steg

This technique uses white spaces for hiding a secret message. There are three methods of hiding data using white spaces (Bender et al., 2007):

- In Inter Sentence Spacing, we place single space to hide bit 0 and two spaces to hide bit 1 at the end of each terminating character.
- In End of Line Spaces, fixed number of spaces is inserted at the end of each line. For example, two spaces to encode one bit per line, four spaces to encode two bits and so on.
- In Inter Word Spacing technique, one space after a word represents bit 0 and two spaces after a word represents bit1. But, inconsistent use of white space is not transparent.

E. Spam Text

HTML and XML files can also be used to hide bits. If there are different starting and closing tags, bit 0 is interpreted and if single tag is used for starting and closing it, then bit 1 is interpreted (Cummins et al., 2004). In another technique, bit 0 is represented by a lack of space in a tag and bit 1 is represented by placing a space inside a tag (Cummins et al., 2004).

F. SMS-Texting

SMS-Texting language is a combination of abbreviated words used in SMS (Shirali-Shahreza & Shirali-Shahreza, 2007). We can hide binary data by using full form of word or its abbreviated form. A codebook is made which contains words and their corresponding abbreviated forms. To hide bit 0, full form of the word is used and to hide bit 1, abbreviated form of word is used (Shirali-Shahreza & Shirali-Shahreza, 2007).

G. Feature Coding

In feature coding, secret message is hidden by altering one or more features of the text. A parser examines a document and picks out all the features that it can use to hide the information (Cummins et al., 2004). For example, point in letters i and j can be displaced, length of strike in letters f and t can be changed, or by extending or shortening height of letters b, d, h, etc. (Rabah, 2004; Brassil et al., 1995). A flaw of this method is that if an OCR program is used or if re-typing is done, the hidden content would get destroyed.

H. Secret Steganographic Code for Embedding (SSCE):

This technique first encrypts a message using SSCE table and then embeds the cipher text in a cover file by inserting articles a or an with the non specific nouns in English language using a certain mapping technique (Banerjee et al., 2011). The embedding positions are encrypted using the same SSCE table and saved in another file which is transmitted to the receiver securely along with the stego file.

I. Word Mapping

This technique encrypts a secret message using genetic operator crossover and then embeds the resulting cipher text, taking two bits at a time, in a cover file by inserting blank spaces between words of even or odd length using a certain mapping technique (Bhattacharyya et al., 2010). The embedding positions are saved in another file and transmitted to the receiver along with the stego object.

J. MS Word Document

In this technique, text segments in a document are degenerated, mimicking to be the work of an author with inferior writing skills, with secret message being embedded in the choice of degenerations which are then revised with changes being tracked (Liu & Tsai, 2007). Data embedding is disguised such that the stego document appears to be the product of collaborative writing (Liu & Tsai, 2007).

K. Cricket Match Scorecard

In this method, data is hidden in a cricket match scorecard by pre-appending a meaningless zero before a number to represent bit 1 and leaving the number as it is to represent bit 0 (Khairullah, 2011).

L. Cascading Style Sheet (CSS)

This technique encrypts a message using RSA public key cryptosystem and cipher text is then embedded in a Cascading Style Sheet (CSS) by using End of Line on each CSS style properties, exactly after a semicolon. A space after a semicolon embeds bit 0 and a tab after a semicolon embeds bit 1 (Kabetta et al., 2011).

2.4 Arabic Text Steganography:

The richness of the Arabic language by its characteristics makes it a strong candidate for hiding information.

Recently, a lot of researches has been done using Arabic text, which used selected as cover media for hiding secret information in different methods like follows:

- Point shifting steganography.
- Extension and pointed letters steganography.
- Using pseudo-space and pseudo connection steganography.
- Using Arabic diacritic steganography.
- Unicode approach steganography.
- Using Kashida and Pseudo-Space.

2.4.1 Arabic language characteristics

Arabic language is the language of about 1.7 billion Muslims. It is spoken by approximately 422 million speakers (native and non-native) in the Arab world. Thus, Arabic language becomes the fifth spoken language in the world.

Arabic language has 28 different letters written from right to left, in a cursive style analogous to Urdu and Farsi scripts.

Every Arabic letter takes various shapes according to its location in the word. Furthermore, Arabic language is characterized by many pointed (doted) letters, with a point, two points, or three points placed above or below some letters. Every Arabic word includes more than one character joined with each other. This connectivity feature is profitable to hide data.

In addition, Arabic has fifteen pointed letters, five of which are multipoint letters unlike English which has no multipoint letters.

Moreover, every word is together with some special signs called ‘‘Harakaat” (i.e. text with diacritics) added to form the vowel sounds only.

There are eight Arabic text diacritics (Harakat):

- Fathah (◌َ).
- Kasrah (◌ِ).
- Damah (◌ُ).
- Sukun (◌ْ).
- Tanwin Fathah (◌ً).
- Tanwin Kasrah (◌ٍ).
- Tanwin Damah (◌ٌ).
- Shaddah (◌ّ).

These diacritics are represented digitally inside the computer as separate (zero location) characters.

Note that the use of diacritics in Arabic is optional in practice and in modern standard Arabic writing; however, it is essential for the holy Quran and most religious and historical scripts.

2.5 Related Works

Text steganography is the most challenging kind of steganography techniques, due to the relative lack of redundant information in text files as compared to images or audio. Recently, there have been several successful attempts to design text steganography schemes for many languages including English (Malik et al., 2017; Aman et al., 2017; Kumar et al., 2016; Chaudhary and Dave, 2016), Chinese (Qin et al., 2014), Hindi (Nagarhalli et al., 2016), Malayalam (Vidhya and Paulb, 2014), Bengali (Khairullah, 2018) and Urdu (Abbasi et al., 2015). However, Arabic is not as fortunate. In Arabic script, several researches have been done, Most of these researches focus on increasing the capacity for hidden bits.

The first proposal in the Arabic text steganography was done in (Shirali-Shahreza & Shirali-Shahreza, 2006). Their schema is dependent on concealing binary values within Persian or Arabic scripts by using a characteristic of coding strategy. This approach is dependent upon the points inherited within the Persian, Urdu and Arabic letters. The points' location on the letters conceal certain data. The hidden data length (i.e. the Secret Object) is considered as a binary using the first several bits. The middle text, (i.e. the Cover Object) is scanned in such a way when a pointed letter is detected. The position of the point is shifted up slightly when the hidden value is 1 otherwise, the location remains unchanged. The advantage of this method is the large amount information will be concealed in text due to the large number of points in letters for both Persian and Arabic.

Authors in (Mohammed et al., 2007) proposed a new steganography approach to conceal hidden secret data within Arabic text cover media. The suggested method employs Arabic language diacritics that are utilized for vowel sounds and it is located in several religious documents. About 8 of these symbols are found in Arabic. They discovered that the "Fatha" symbol has been used in Arabic text more than the other 7 symbols. Thus, they used the "Fatha" symbol to present 1 and the other symbols to present 0. The benefit of this method is the big capacity due to each Arabic character is relevant for a diacritic. The downside is that concealing some diacritics can get the reader's attention.

(Memon et al., 2008) use the reverse "Fatha" was used to hide the secret data within cover text rather than the normal ''Fatah". An inverse direction ''Fatah" is used on the Arabic text cover to hide bit 0 and the original direction ''Fatah" to hide bit 1 as shown in Figure (2.4)

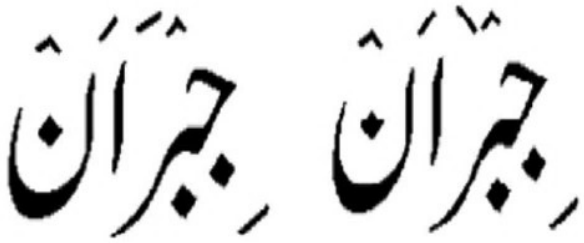

Figure (2.4) Use of normal and reverse Fatah

The researchers in (Ahmadoh and Gutub, 2015) improve the method of (Memon et al., 2008) by using two diacritics to hide secret data into Arabic text instead of using one diacritic. They focus on hiding data in ‘‘Fatah” and ‘‘Kasrah” together instead of using only ‘‘Fatah”.

Moreover, Bensaad and Yagoubi (Lahcen, 2011) present a technique for hiding secret data in diacritics. They first sure that all possible diacritics are present in the cover text, and then match every diacritic to a bit from the secret bits. After that, if the secret bit is 1, it keeps the diacritics as it is, but if the secret bit is 0, it removes the diacritic as shown in Figure (2.5). The method provides high capacity, but it may attract attention where some diacritics are present and the others are not.

Cver text:	مُـسْـتَـفْـعِـلٌ
Secrete bits:	0 1 1 0 0 1
Stego-text:	مُـسـتـفْـعِـل

Figure (2.5) Example of hiding data in diacritics

All diacritics-based methods suffer from two inherent problems; first, they need a fully diacritical text, and most of the Arabic texts nowadays have no diacritics. Second, they have low invisibility and can raise suspicious of the readers.

In (Odeh et al., 2012) Odeh et al. present a method for embedding data in multipoint Arabic/Persian letters like (ث, tha). In Arabic language, there are five multipoint letters. Each character can be used to hide two bits to determine the shifting and distance between letter points, as shown in Table (2.1). However, both the sender and receiver must have a specific font to show the text and preserve the hidden message. This means that the output text is

fixed for the use of only one font. Hence, the main weakness for pointsbased methods is the lack of robustness. Moreover, the hidden information is lost in any retyping or scanning.

Table (2.1) Representing relation between shifting and distance to letter format

Point shifting	Distance	Code	Letter effects
0	0	00	No change
0	1	01	Only distance between points increase
1	0	10	Only little upper shifting
1	1	11	Upper shifting and distance between points increase

The authors in (Odeh and Elleithy, 2013) hide information in text without any change in file format. They use a Zero-Width Character (ZWC) that does not occupy any space. They add ZWC before and after space character. Also, Odeh et al. (Odeh, 2013) use Kashida in Arabic text to hide data inside cover media. In addition, they apply four scenarios randomly to improve data privacy.

Malalla and shareef (Malalla and shareef, 2016) improve Arabic text steganography by presenting a hybrid between steganography with cryptography. They encrypt the message then embed it in Arabic text using a Kashida based technique. Also, Ahmadoh and Gutub (Ahmadoh and Gutub, 2015) improve the use of Kashida to maximize the capacity by utilizing all possible locations for Kashida in Arabic letters and putting Kashida where it is applicable.

The authors in (Nofaie and Gutub, 2016) enhance the capacity of using Kashida by utilizing the whitespaces between words of Arabic language for hiding. If the secret bit equals 1, Kashida is added between the suitable Arabic letters as can be accepted, until all possible Kashida additions are handled, then, before moving to the next word, whitespaces are used, if the secret bit is 1, two consecutive whitespaces are added between words, otherwise normal whitespace is kept as it is.

In general, the main disadvantage of Kashida-based methods is that it draws the attention to the reader, increases the file size and changes the text appearance.

Alotaibi and Elrefai (Alotaibi, 2016) present a watermark technique to hide a secret data in Arabic text. They utilize what is called a pseudo-space, it is a small space employed to break the same word into two words. The bits are hidden by checking each space between word letters before it. If the letter is pointed and the bit is 1, the pseudo-space is added. Else if the bit is 0, no pseudospace is added. If the letter is un-pointed and the bit is 1, pseudo-space is added. Else if bit is 1, no pseudo-space is added as shown in Figure (2.6).

Watermarking bits	1001011
Original text	البر حسن الخلق والإثم ما حاك في نفسك
Output text	البر حسن الخلق والإثم ما حاك في نفسك 1 1 0 1 0 0 1

Figure (2.6) Example of Alotaibi's method (Alotaibi, 2016)

Recently, Alhusban and Alnihoud (Alhusban et al., 2017) present a method in which each existing kashida can hide two bits instead of only one bit. Their method depends mainly on the nature of letters; whether they are pointed or un-pointed letters. The cover text is divided into two equal blocks. Each one is being dealt with in different way. Figure (2.8) shows an example of this method. However, in some cases, the sequences of the secret bits, and suitable appearances of the targeted Kashida may affect the capacity ratio of this method. This might be considered as a drawback and may lead to undesirable results of such cases.

(Cover text) بلغت الاستثمارات السعودية في الأردن نحو عشرة مليارات دولار	
بلغت الاستثمارات السعودية في **(Block no.1)**	الأردن نحو عشرة مليارات دولار **(Block no.2)**

Figure (2.7) Example of the method presented in (Alhusban et al., 2017)

Moreover in (Anes et al., 2017), Shaker et al., hide the text in Arabic language using sun and moon letters. They apply four scenarios. In the first scenario, a Kashida is inserted

after a sun letter to hide the secret bits 00. The second scenario is that two Kashida is inserted after a sun letter to hide the secret bits 11. In the third scenario, a Kashida is inserted after a moon letter to hide the secret bits 01. The fourth scenario is that two Kashida is inserted to hide the secret bits 10. The method try to increase the probability of hiding the secret bits in any letter by employing the moon and sun letters concept. Nonetheless, it is also very important to maintain the imperceptibility aspect while improving the capacity.

Third-order Arabic language has 28 letters. Authors categorized these letters into two groups. Pointed like 'ا' Alif; 'ر' Raa; 'و' Waaw, and un-pointed letters like 'ت' taa; 'ذ' Thaal; 'ز' Zaa respectively. Out of 28 letters, 14 are pointed letters. Authors used these pointed letters to hide secret data. By shifting points upwards to hide secret bit 1 and keep it normal distance to hide bit zero (Shirali-Shahreza, 2006). Drawback of this method is hidden information can be lost by using optical character recognition (OCR) software or retyping these documents.

Out of 28 letters, 22 Arabic letters can be connected to next or previous letter of the word by Arabic extension. Gutub and Fattani (2007) used this extension with pointed letters to hide secret bit 1 and un-pointed letters to hide bit 0 as shown in Figure 2. Authors (Al-Nofaie et al., 2016) improved extension/Kashida approach by utilizing 'Kashida' with whitespaces between the words to enhance the hidden data capacity of cover text. After using extension in some words, length and size of cover text file increased Figure (2.9) which becomes suspicious and gets attention of intruders.

Secret bit	110010
Original text	من حسن اسلام المرء تركه مالا يعنيه
Output text	مـن حسن اسلام المرء تـركـه مـالا يـعنـيه 0 1 0 0 1 1

Figure (2.8) Data hiding with pointed and un-pointed letters by inserting Arabic extension

Shirali et al (2008) proposed a new approach of Arabic Unicode texts to hide data in Persian and Arabic languages by using pseudo-space and pseudo-connection also known as zero-width non-joiner (ZWNJ) and zero-width-joiner (ZWJ), respectively. This approach sorted out weather the letter of word is connected to next letter or not. If the letter is connected

to the next letter, it hided bit one by inserting ZWJ letter between two letters. However, if the letter is not connected to the next letter, then it inserted ZWNJ between letters and do not insert any character to hide bit zero. This proposed method is only useful for non-diacritics Arabic text.

Arabic language mainly used eight diacritics like Fatha, Kasrah, Dhammah, etc. Diacritics are used on Arabic text to perform correct pronunciation of Arabic language. To embed secret information (Aabed et al., 2007) used fully diacritised Arabic text as a cover media. To hide bit 1 they found first Fatha in the Arabic text and hide on that letter. But they removed all other diacritics like Kasrah, Dhammah, etc., from cover text if appeared before first Fatha. However, bit 0 embedded in all non-Fatha diacritics. To increase the data hiding capacity (Ahmadoh and Gutub, 2015) improve the work of (Aabed et al., 2007) and proposed a new data hiding approach by using two diacritics 'Fateh' and 'Kasrah' on the basis of even and odd bits of the secret message. A lot of efforts have been done to hide secret data by using more than one Arabic diacritics like Fatha, reverse Fatha or diacritics shift (vertically upward) (Odeh and Elleithy, 2012). However, recently these methods are facing different challenges. Firstly, according to recent trends Arabic text is used without diacritics. Secondly, after removing some diacritics or its shift, file size is changed and it becomes suspicious.

In Arabic language, 22 letters can be connected to next or previous letter of the word. Therefore, one letter can be written in four different shapes as shown in Table (2.2). All of these connected shapes of the letter can be first, middle, last or an isolated format in the word. Each specific shape of letter has its unique Unicode. Shirali et al (2008) used these kinds of Arabic Unicode to hide secret information into Arabic cover text. This approach is suitable if the text appears in non-diacritics form. Unicode approach has also been used (Mohamed, 2014) to embed secret data 0 and 1 in run length encoding format. However, this approach is restricted to hide data using only in isolated letters.

Table (2.2) Letter shapes according to its position in the word

At the end of the word	In the middle of the word	At initial form of the word	Isolated form in the word
ـس	ـسـ	سـ	س

Kashida and pseudo-space (PS), extends (Al-Alwani, et al., 2007), which used only Kashida, and in (Al-Nofaie et al. 2016), which used Kashida with normal spaces to hide secret bits. Authors in (Al-Nofaie et al. 2019) used Kashida-PS, This approach check the bit, and letter of word is connected to next letter or not, if the bit is 1 and the letter is connected to the next letter, it hided bit 1 by inserting two kashida between two letters. However, if letter is not connected to the next letter, then it inserted two PS between letters. Otherwise, if the bit is 0 and the letter is connected to the next letter, it hided bit 0 by inserting one kashida between two letters. However, if the letter is not connected to the next letter, then it inserted one PS between letters, length and size of cover text file increased Figure (2.10) which becomes suspicious and gets attention of intruders.

Secret bit	100101100101110011100011 1
Original text	احرص على ماينفعك واستعن بالله ولاتعجز
Output text	احرص علـى ماينفعـك واسـتـعـن بـالله ولاتعجز
	100101100101110011100011 1

Figure (2.9) Data hiding with pointed and un-pointed letters by inserting Arabic extension

Finally we show summarize of some related works by display author name, year, methods and weakness of that works as shown in table (2.3).

Table (2.3) Summarize related works

Author(s), Year	Method	Weakness
Shirali et al (2008)	Zero-Width Non-Joiner (ZWNJ) and Zero-Width-Joiner (ZWJ)	- Provide a small capacity. - Is only useful for non-diacritics
Odeh and Elleithy (2013)	Zero-Width Character (ZWC)	- Provide a small capacity.
Ahmadoh and Gutub (2015)	Diacritics 'Fateh' and 'Kasrah'	- File size is changed and it becomes suspicious.
Al-Nofaie et al. (2016)	Extension/Kashida Approach	- Length and size of cover text file increased. - Raise suspicious of the readers.

Alotaibi and Elrefai (2016)	Pseudo-Space (PS)	- Provide a small capacity.
Al-Nofaie et al. (2019)	Kashida and Pseudo-Space (PS)	- Raise suspicious of the readers.

Chapter 3

Research Methodology

Research Methodology

In this chapter, we explain our proposed Data Hiding Enhancement in Arabic Text, approach which we followed in this research. This chapter is organized into six sections. Section 3.1, presents general view of our proposed approach. Section 3.2, will give a description of how can we keep more secure our hiden data in cover text from the attacker. Section 3.3, give a description of how to embed secret bits into cover text? Section 3.4 show proposed schema is comprised of two phases: Embedding Phase Algorithm and Extracting Phase Algorithm. Section 3.5, presents procedure of hiding secret data into cover text. Section 3.6, presents procedure of retrieve secret message from stego-text.

3.1 Methodology:

All private letters out of 28 Arabic letters are classified as special letters. These special letters are 'ا' Alif, 'د' Daal, 'ذ' Thaal, 'ر' Raa, 'ز' Zaa, and 'و' Waaw. These special letters are be private in Arabic text in the following cases: if show at start of word or at the end of the word. These special letters cannot be connected with letter next to it if appeared in the center of the word. This special feature of these letters can be of benefit as word divider. For example, in 'المرسلات' we can see that Arabic word divides into parts because:

'ا' Alif appear at start

'ر' Raa belongs to six defined letters so we can see that it is not combined to the next letter.

'ت' Taa appears as an isolated letter as before this 'ا' Alif belongs to six defined letters and it is second last letter of the word so the last letter will be isolated letter (Mohamed, 2014).

We used this performance of these special letters in our algorithm. Since, our algorithm based on the combinations of zero-width-character (ZWC), ZWJ and PS, therefore it consists of three different scenarios as follows:

- If the letters belong to special letters we insert combination of ZWC.
- If letters are combined to next letter or isolated letter, then we insert combination of ZWJ character between the letters to hide pair of secret message bits.
- If there are spaces between words, then we insert PS character between the words to hide one bit of secret message.

While pseudo-space (PS) is a character that does not appear in printing.

These combinations of ZWC, ZWJ and PSs increase the capacity of hidden information. This is because every character (letters and spaces) is exploited to hide a secret. Unlike the previous methods mentioned above. These combinations are made by deep analysis of Arabic text. As a result, after embedding secret message bits, shape and visibility of cover-text and stego-text remain same with higher capacity of hidden data. The other distinction of our proposed algorithm is that we used it on both diacritic and non-diacritics Arabic text.

Generated binary bits of secret text are calculated as:

$$Bsm = Tc \times 8 \quad (1)$$

Where Bsm = total No. of bits to secret message, Tc refers to total No. of characters and 8 corresponds to every character's length.

3.2 Inserting Start-End Bits into Cover Text

How can we keep our hidden data more secure in cover text from the attacker? For this, we embed some bits to keep track of start and end point. If we find start-end bits in text, it ensures the existence of secret message in text file. To make stego-text file more secure we further embed start-end point at different places more than one time in cover text file with the combination of ZWC and ZWJ with PS. The combination of ZWC and ZWJ with PS increase the security and confidentiality because it is hard to judge for hackers about start-end points due to similar combinations of ZWC and ZWJ with PS for hiding secret message. It also improves the data security.

3.3 How To Embed Secret Bits Into Cover Text?

To embed secret bits into cover text, firstly the algorithm read pair of bits of Bsm and then read the next letter of cover text. If the next letter is from special letters, then it applies zero width character's (ZWC) combination after special letter. Otherwise, it applies ZWJ combination before the next letter, if there is space between words will insert pseudo-space (PS). No action is taken if the next letter is the last letter of the word, which indicates no data is hidden in the last letter of each word. Hided data's capacity in each word is calculated by using this equation $[(NL_w) - 1] \times 2$ (2)

Where NL represents number of letters in each word of cover text and w is word count range from 1 to n. For example, in 'المرسلات', this word contains eight letters, so its capacity will be: [8 – 1] × 2 = 14. It means an eight-letter word can hide 14 bits of secret message. Therefore, for n number of words:

Hided capacity of bits $= \left(\sum_{w=1}^{n}[(NL_w - 1) * 2]\right) + NS$ (3)

Where NS represents number of spaces in the cover text.

3.4 The Proposed Scheme

The proposed scheme is comprised of two phases, as follows:

3.4.1 The Embedding Phase:

The embedding phase is based on checking the cover text capacity, if it has capacity larger than number of bits, it will insert start bits then read next character if it is space or not, if it is space it will check next bit and insert PS. If the character is not a space it will check if the special letter will take next bit pair and apply ZWC, and if next letter is not special letter it will take next bit pair and apply ZWJ. Then it will check the all bits has hidden or else return previous steps until hidden all bits. The embedding phase algorithm is shown in **Algorithm 1**.

Algorithm 1. The Embedding Phase Algorithm

Input : Secret Message bits (M), Cover Text File (C)

Output : Stego_text(St)

1. Calculate the size of C according to equation (3).
2. Set i = 0 // Counter to hold the current secret bit being processed
3. Set j = 0 // Counter to hold the current character in the cover text file being processed
4. Set k = 0 // Counter to hold the current character in the Stego_text
5. Insert start bits.
6. For each bit (b_i) in the secret bit sequence

If the current character in the cover file (C_j) is space

then

If $b_i = 1$

Add two pseudo-space to Stego_text

else // $b_i = 0$

Add one pseudo-space to Stego_text

i++

j++

k++.

else if the current character in the cover file (C_j) is special letter

generate ZWC(b_i , b_{i+1})

i = i+2

j++

k++

else // current character in the cover file (C_j) is not special letter

generate ZWJ(b_i , b_{i+1})

i = i+2

j++

k++

7. Repeat the step 6 until all M's bits are embedded.
8. Insert end bits for stop hidden secret message process
9. Generate the Stego_text St

3.4.2 The Extracting Phase:

The extraction algorithm of the proposed method is shows in **Algorithm 2.** The algorithm takes stego-text as an input and returns the secret message as an output.

The stego-text is sequentially processed as follows: for each character in a word, if it is space, it will check is their one pseudo-space will return bit 0, or bit 1 if their two pseudo-space, then check the next character if it special letter will return bit pair of ZWC, or bit pair of ZWJ if the letter is not from special letters.

Algorithm 2. The Extracting Phase Algorithm

Input : Stego_text(St)

Output : Secret Message bits (M)

1. Set k = 0 // Counter to hold the current character in the Stego_text
2. Insert start bits.
3. For each character (C_k) in the Stego_text

If (C_k) is space

search for the number of pseudo-space

and extract the corresponding secret bit

add extract bit to (M)

k++

else if (C_k) is special letter

use map-zwc combination

and extract the corresponding secret bits

add extract bits to (M)

k++

else // (C_k) is not special letter

use map-zwc combination

and extract the corresponding secret bits

add extract bits to (M)

k++

4. Return Secret Message bits (M)

3.5 Procedure of Hiding Secret Data into Cover Text

To hide secret data, we made an algorithm that embed secret data into cover text and generate stego-text file by using proposed combinations of ZWC, ZWJ and PS as shown in Figure (3.1).

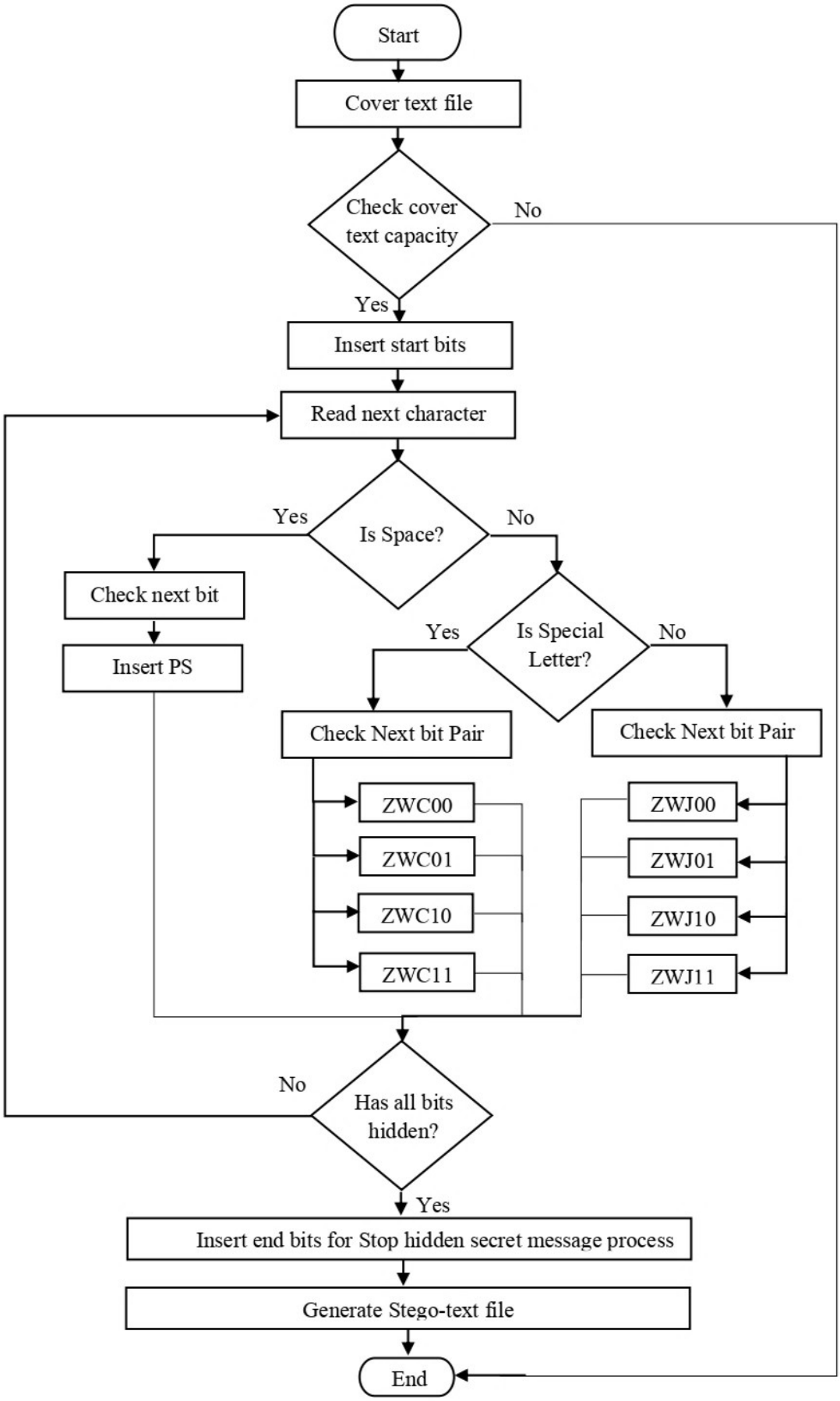

Figure (3.1): procedural flow chart for embedding secret bits into Arabic cover text file.

As an example to understand the proposed algorithm, we selected this Arabic sentence 'انا عند ظن عبدي بي' as cover text and '100110111' (9 bits) data bits to hide. Following steps have been taken to hide above mentioned data bits into cover text:

Step 1 At first, algorithm verifies cover text capacity. It will only hide data if capacity is more than bits count. As we can see, our selected Arabic sentence consists of five words. Each word has 3, 3, 2, 4, and 2 letters and 4 spaces, respectively. Therefore, according to equation (3), we can hide up to 22 bits in this selected Arabic verse as shown below:

[(3 - 1) * 2 + (3 - 1) * 2 + (2 - 1) * 2 + (4 - 1) * 2 + (2 - 1) * 2] + 4 = 22

It shows, our selected sentence/string has enough capacity to hide data. As our selected data ('100110111') which is being hided has only 9 bits, so it can be easily hided in the selected string. For this, we inserted some starting bits at the start of cover text and proceeded.

Step 2 At the next stage, algorithm read next letter of selected string. If next letter of selected string is among special defined letters then it inserts ZWC combination after received letter, otherwise it will insert ZWJ combination before received letter according to bit pair. Since it received 'ا' as next letter which is from the special defined six letters and the first bit pair is '10'. Therefore, our algorithm will add ZWC10 + 'ا'.

Step 3 Next letter of selected string is 'ن' and '01' is the next hidden bit pair. Which is 'ن' does not belong to the special defined letters. Therefore, our algorithm inserts ZWJ01 before the letter 'ن' as ZWJ01 + 'ن'.

Step 4 Next is 'ا' which is last letter of the word. As we defined in Section 3.3, our algorithm will skip it and move to next character.

Step 5 Next character is space. Our algorithm will insert two PSs, because the next bit is 1. Next hidden bit is '1'. Therefore, it adds two PSs in selected string.

Step 6 Next letter is 'ع' which is not from special letter. As defined in Section 3.3, our algorithm will insert combination of ZWJ. Next hidden bit pair is '01'. Therefore, it adds 'ع' + ZWJ01 in selected string.

Step 7 Next one is non special letter 'ن'. Therefore, cover text will be ZWJ11 + 'ن'.

Step 8 Next, algorithm did not find any bit pair in the selected data, which confirm whole data is embedded successfully in the cover text. After this, algorithm will insert end bits in the cover text to indicate hidden process is completed. In the end, it will add random ZWC and ZWJ combinations into the remaining cover text to create confusing situation for intruders in steg-analysis and message regeneration.

Step 9 At the end, we have a stego-text file, which contains completely hidden data. In stego-text file, visual appearance of text is exactly the same as cover text file which is the key strength of our proposed algorithm.

3.6 Extracting Algorithm To Retrieve Secret Message From Stego-Text

After creating stego-text file, it is important to know how to extract hidden message. For this, we adopt reverse procedure of embedding algorithm as shown in **Figure (3.2).** The extracting Algorithm is described in the following steps:

Step 1 First, algorithm will verify the existence of start/end bits. If start and end bits were found, then it starts regenerating secret message other it stops. From start bit it read next letter of the selected string.

Step 2 Next letter of stego-text is 'ا'and it is among the special defined. In this case, there will be a combination of ZWC after letter 'ا'as stated in Section 3.3. As a result, our algorithm will extract ZWC10 information and add '10' bit pair in the regenerated message string, i.e., 10.

Step 3 Next one is letter 'ن' which is connected to next letter. For this, it will extract a combination of ZWJ01 and add '01' bit pair in the regenerated message string, i.e., 1001.

Step 4 Next one is 'ا'and it is last letter of the word. For this, no action will be taken as stated earlier.

Step 5 Next character is space. For this algorithm will extract 1 bit and add 1 in regenerating string, i.e., 10011.

Step 6 Next word starts with letter 'ع'which is non-special letter. For this algorithm will extract ZWJ01 and add 01 in regenerating string, i.e., 1001101.

Step 7 Next one is a non-special letter 'ن'. For this, it will extract ZWJ11 and add 11 with regenerating string, i.e., 100110111.

Step 8 For the next letter'د' we find end bits instead of data bits which confirm the extraction of hided data. After this algorithm will stop further extraction process.

Step 9 Finally, we converted regenerated binary string into humane readable format. That is our secret message.

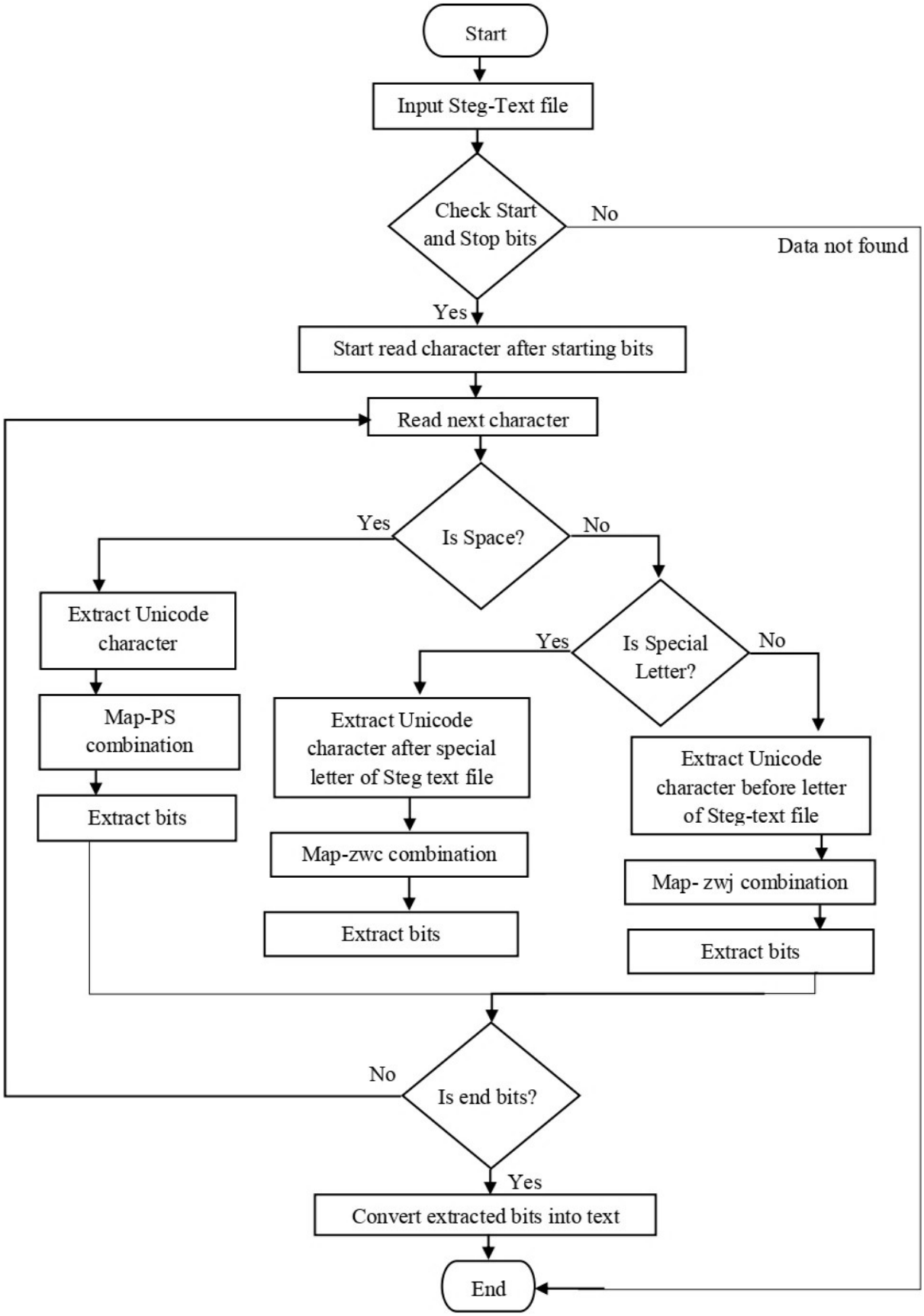

Figure (3.2): Reverse algorithm flow chart for extracting secret message from stego-text file

Chapter 4

Results and Discussions

4.1 Results and Discussions

In this section, we described our experiments on Arabic text for data hiding process and their results. We applied our algorithm not only on diacritic but also on non-diacritic cover text strings for hiding secret data as shown in Table (4.1). As we can see clearly from Figure (4.1), data hidden capacity is higher than any other used approaches like inserting Kashida after pointed letters, Kashida variation algorithm, Unicode approach to hide information using isolated letters, Kashida-PS, etc., (Aabed et al., 2007; Gutub and Fattani, 2007; Mohamed, 2014; Odeh et al., 2013; Al-Nofaie et al., 2019). In all these mentioned proposed techniques, authors used minimum letter of each Arabic word to hide secret bit 1 or 0. Therefore, hidden data capacity per word very low as compared to our proposed approach. We used all letters except the last letter of the word to embed pair of bits in each letter of the cover text file. From experimental results of Table (4.1), we can see that against all recently data hiding proposed approaches, visual similarity of cover text and stego-text is compromised due to inserting Kashida before/after pointed letters or removing diacritics from the letter to hide secret data. While our proposed algorithm gives better results and achieved same visual appearance of cover-text and stego-text file with significantly increased hidden data capacity per word.

In order to measure the performance of the proposed algorithm, comparisons are made on the basis of the capacity ratio. In text steganography, the hiding capacity is a key metric to analyze the performance. That is how to maximize the length of the secret message that can be embedded in the cover file without affecting it is quality as much as possible. Hiding capacity is defined as the size of hidden message relative to the size of the cover (Odeh et al., 2013; Ahmadoh and Gutub, 2015; Alotaibi and Elrefai, 2017). Accordingly, the capacity ratio is generally defined as follows (Odeh et al., 2013; Ahmadoh and Gutub, 2015; Alotaibi and Elrefai 2017; Alhusban et al., 2017; Anes et al., 2017):

$$\text{Capacity ratio} = \frac{\text{amount of hidden bytes}}{\text{size of the cover text}} \tag{4}$$

Table (4.1) Results of data hiding capacity on different cover text using our algorithm and comparison with other used approaches

Input/output	*Selected string as a cover object and stego-object*	*No. of hidden bits capacity*	*Approaches*	*References*
Cover object	من حسن السلام المرء تركه مالا يعنيه	6 bits	Kashida approach	Gutub and Fattani (2007)
Secret information	110010			
Stego-object	مـن حسن السلام المرء تـركـه مـالا يـعـنـيه			
Cover object	من حسن السلام المرء تركه مالا يعنيه	50 bits	Combinations of ZWJ, ZWC and PS	**Proposed scheme**
Secret information	01001000011001010110110110001101100011011110010100			
Stego-object	من حسن السلام المرء تركه مالا يعنيه			
Cover object	حَدَّثنا سُفيَانُ عَنْ يَحْيَى	8 bits	Diacritics approach	Aabed et al. (2007)
Secret information	11100111			
Stego-object	حَدَثنا سُفيَان عَن يَحْيى			
Cover object	حَدَّثنا سُفيَانُ عَنْ يَحْيَى	27 bits	Combinations of ZWJ, ZWC and PS	**Proposed scheme**
Secret information	0111100101100 10101110011010			
Stego-object	حَدَّثنا سُفيَانُ عَنْ يَحْيَى			

Cover object	اتق شر من احسنت اليه	9 bits	Kashida variation algorithm	Odeh et al. (2013)
Secret information	001011100			
Stego-object	اتـق شـر من احـسنت الـيـه			
Cover object	اتق شر من احسنت اليه	26 bits	Combinations of ZWJ, ZWC and PS	**Proposed scheme**
Secret information	01111001011001010111001011			
Stego-object	اتق شر من احسنت اليه			
Cover object	صلاح أمرك للأخلاق مرجعه فقوم النفس بالأخلاق تستق	5 bits	Unicode approach to hide data in isolated letters	Mohamed (2014)
Secret information	11010			
Stego-object	صلاح أمرك للأخلاق مرجعه فقوم النفس بالأخلاق تستق			
Cover object	صلاح أمرك للأخلاق مرجعه فقوم النفس بالأخلاق تستق	bits75	Combinations of ZWJ, ZWC and PS	**Proposed scheme**
Secret information	011010010111010000100111 10110111001100 100000011011110110101101 0011000010111			

Stego-object	صلاح أمرك للأخلاق مرجعه فقوم النفس بالأخلاق تستق			
Cover object	البر حسن الخلق والإثم ما حاك في نفسك	7 bits	Pseudo Space method	Alotaibi and Elrefai, 2016
Secret information	1001011			
Stego-object	البر حسن الخلق والإثم ما حاك في نفسك			
Cover object	البر حسن الخلق والإثم ما حاك في نفسك	46 bits	Combinations of ZWJ, ZWC and PS	**Proposed scheme**
Secret information	010010000110100110110110001101100110111001010100			
Stego-object	البر حسن الخلق والإثم ما حاك في نفسك			
Cover object	احرص على ماينفعك واستعن بالله ولاتعجز	25 bits	Combinations of Kashida and PS	Al-Nofaie et al (2019) b
Secret information	1001011001011100111000011			
Stego-object	احـرص عـلـى ماينفـعـك واسـتـعـن بـالله ولاتـعجز			
Cover object	احرص على ماينفعك واستعن بالله ولاتعجز	57 bits	Combinations of ZWJ, ZWC and PS	**Proposed scheme**
Secret information	11 010101001000011001010110110001010001101100011011110010101			
Stego-object	احرص على ماينفعك واستعن بالله ولاتعجز			

In figure (4.1) show hidden secret bits capacity in the selected cover text as shown table (4.1)

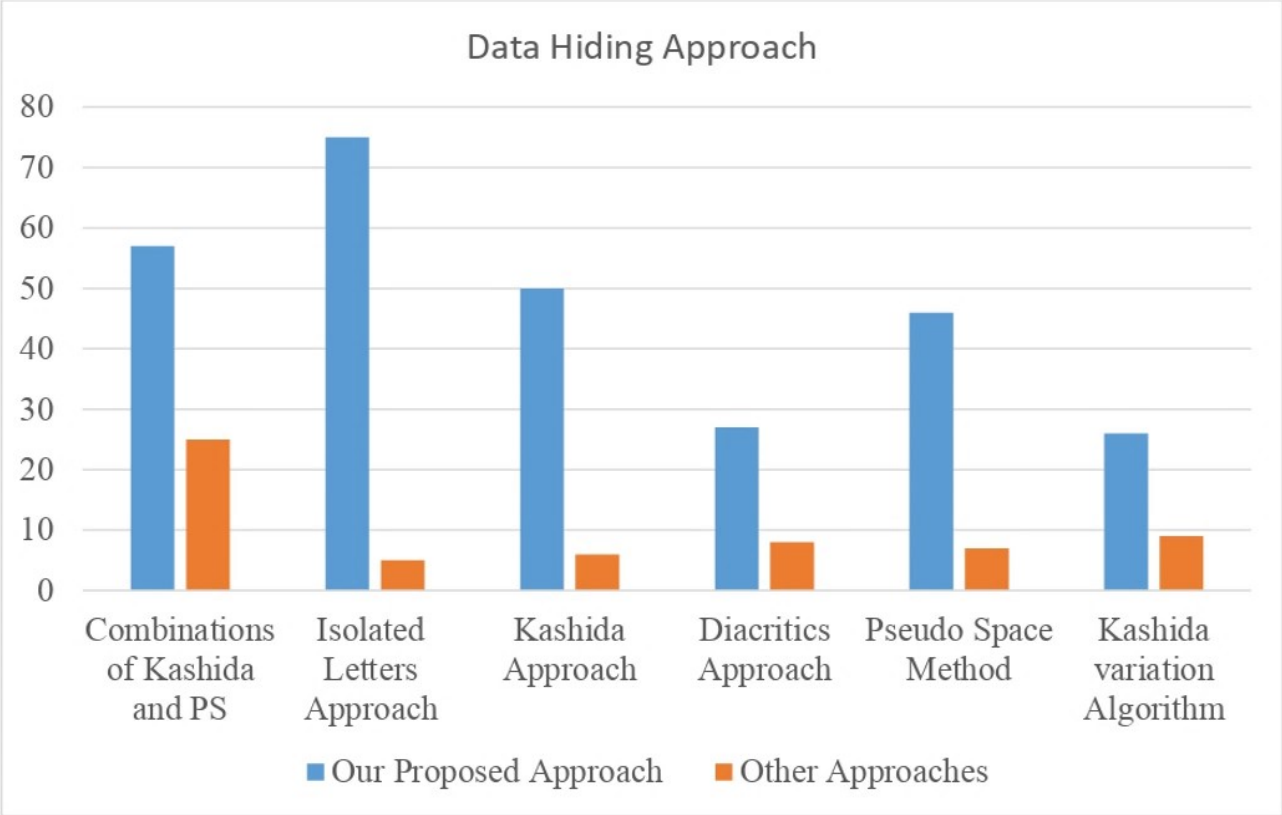

Figure (4.1) Comparison of data hiding capacity with other steganographic methods.

To verify the strength of hiding higher capacity of secret data with our algorithm, we selected one cover string which is given below.

"من حسن السلام المرء تركه مالا يعنيه"

First, we applied (Odeh and Elleithy, 2012) approach on above cover string who used ZWC and space character between the words and hided 2 bits per available space, as a result we could hide 12 bits in this cover text. Then we used (Shirali-Shahreza and Shirali-Shahreza, 2008) approach to hide data on the same cover text that used ZWJ and ZWNJ for every character, we obtained 29 bits in total. Then we used (Alotaibi and Elrefai, 2017) approach who used watermark method to use space character between the words and hided 4 bits per available space, we obtained 20 bits in total. At the end, we applied our proposed approach (hiding one bit per space character and pair of bits per character except last letter of each word) and the hided data capacity was 50 bits which is much higher than above mentioned approaches. Details of all approaches are listed in Table (4.2). Our

experimental results validate that hidden data capacity is significantly increased with actual visual appearance of cover-text and stego-text file as shown in Figure (4.2) and Table (4.2).

Table (4.2) Results and comparison of data hiding capacity on one selected cover text with our algorithm and other approaches

Selected cover string	*Per character or space*	*Bit capacity*	*References*
من حسن السلام المرء تركه مالا يعنيه	Number of spaces in string = 6 per space 2 bits hiding capacity	Hidden bit capacity = 6 * 2 = 12 bits	Odeh and Elleithy (2012b)
من حسن السلام المرء تركه مالا يعنيه	Hiding one bit per character	Number of total string character * 1 = 29 bits	Shirali-Shahreza and Shirali-Shahreza (2008)
من حسن السلام المرء تركه مالا يعنيه	Hiding four bits per space	Number of total spaces * 4 = 20 bits	Alotaibi and Elrefai, 2017
من حسن السلام المرء تركه مالا يعنيه	Hiding one bit per space and pair of bits per character except last letter of each word	$([(NLW) - 1] \times 2)+6=$ $((29–7) * 2) + 6 = 50$ bits	Our work

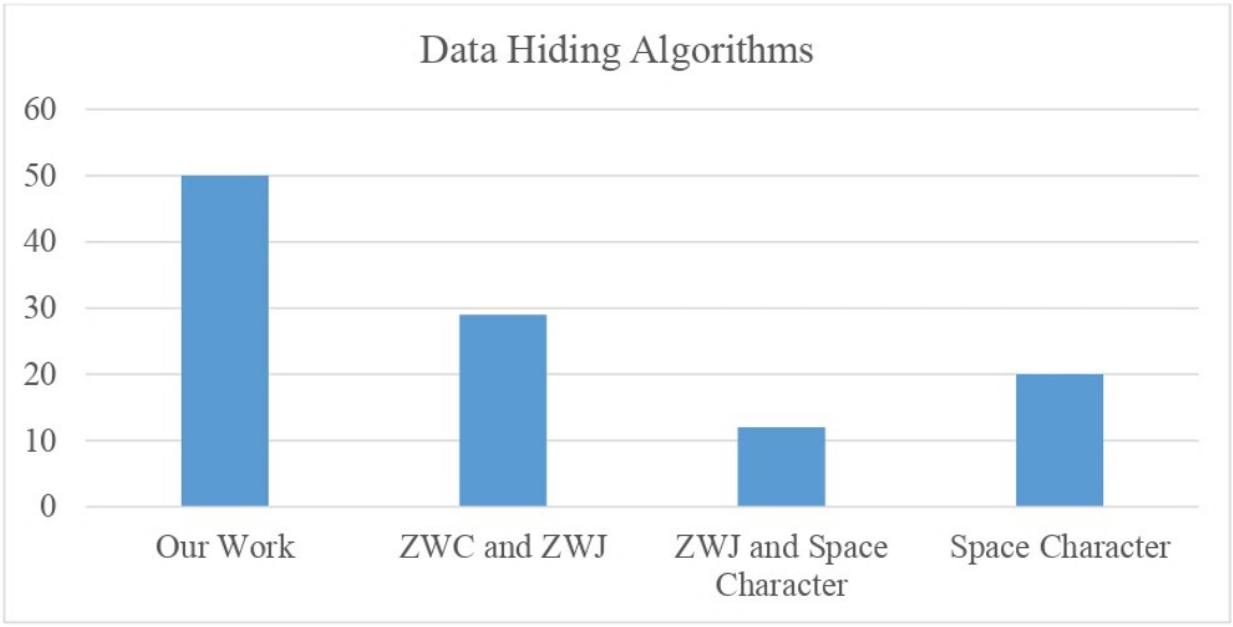

Figure4.2 Hidden data capacity against other steganographic methods with same cover text

Chapter 5

Conclusion and Future Work

This chapter summarizes the research outlined in this thesis. Section 5.1 summarizes the main points of the thesis and list the main conclusions of the work. Future work is outlined for future investigations in Section 5.2.

5.1 Conclusion

We used Arabic text as a cover media to hide secret information by using combinations of ZWC and ZWJ with PS in our proposed algorithm. Combinations of ZWC and ZWJ with PS were made by deep analysis of Arabic text. To calculate data hiding capacity for each word, we formulated this equation $[(NL_w) - 1] \times 2$ Addition to number of spaces (NS). Our experimental results show that with the help of our algorithm we can hide higher capacity of data into the cover text than any other reported algorithms. We improved the security by placing start bits, end bits into the cover text to create confusing situation for intruders. The proposed method provides the same visual appearance of cover-text and stego-text file that will reduce the attention of hackers and intruders.

5.2 Future Work

The future work to improve the proposed methods can be summarized as follows:

1. Increase the data capacity ratio by using all letters of the word.
2. Use other features of Arabic language to enhance the hiding capacity.
3. Improve more security phase by using some compression tools on data to reduce the size of text file.

References:

Aabed, M. A., Awaideh, S., Elshafei, M., & Gutub, A. A. (2007). Arabic diacritics based steganography. *IEEE International Conference on Signal Processing and Communications, 2007, ICSPC 2007, IEEE*.

Abo Mousa, S. M. (2017). *LSBs Steganography Based on R-Indicator*. Gaza: The Islamic University–Gaza.

Abbasi, Tabassum A., Naqvi, Nuzhat, Khan, Aihab, Ahmad, & Basheer (2015). Urdu text steganography: utilizing isolated letters. *In: Proceedings of the 13th Australian Information Security Management Conference.* Edith Cowan University Joondalup Campus, Perth, Western Australia, pp. 37–46.

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

Al-Alwani, W., Bin-Mahfooz, A., & Gutub, A. (2007). A novel arabic text steganography method using extensions. *World Acad. Sci. Eng. Technol*, pp. 83-86.

Al-Nofaie, S., Fattani, M., & Gutub, A. (2016). Capacity improved Arabic text steganography technique utilizing 'Kashida' with whitespaces. *In: 3rd International Conference on Mathematical Sciences and Computer Engineering (ICMSCE),* p 38–44.

Al-Nofaie, S., Gutub, A., & Al-Ghamdi, M. (2019). Enhancing Arabic text steganography for personal usage utilizing pseudo-spaces. *Journal of King Saud University – Computer and Information Sciences*, p. 13.

Alhusban, Ala'a M., Odeh, Jehad Q., & Alnihoud (2017). A meliorated Kashida-based approach for Arabic text steganography. *International Journal of Comput. Sci. Inf. Technol. (IJCSIT),* p 9.

Alotaibi, R., & Elrefai, L. (2016). Utilizing Word Space with Pointed and Unpointed Letters for Arabic Text Watermarking. *IEEE 18th International Conference on Computer Modelling and Simulation (UKSimAMSS)*. Cambridge, UK.

Alotaibi, R., & Elrefai, L. (2017). Improved capacity Arabic text watermarking method based on open word space. *ScienceDirect* , pp. 50-62.

Aman, Muhammad, Khan, Aihab, Ahmad, Basheer, Kouser, & Saeeda (2017). A hybrid text steganography approach utilizing unicode space characters and zero-width character. *International Journal of Inf. Technol. Secur.*, p 85–100.

Anes, A., Shaker, Ridzuan, Farida, Pitchay, & Ali, S. (2017). Text steganography using extension Kashida based on the moon and sun letters concept. *International Journal of Adv. Comput. Sci. Appl.,* p 8.

Banerjee, I., Bhattacharyya, S., & Sanyal, G. (2011). Novel text steganography through special code generation. *International Conference on Systemics, Cybernetics and Informatics*, 2011, pp. 298-303.

Bender, W., Gruhl, D., Morimoto, N., & Lu, A. (1996, 1). Techniques for Data Hiding. *Ibm Systems Journal*, pp. 313-336.

Bennett, K. (2004). *Linguistic Steganography: Survey, Analysis, and Robustness Concerns for Hiding Information in Text.* Purdue University: CERIAS Tech. Report.

Bensaad, M. L., & Yagoubi, M. B. (2011). High Capacity Diacritics-based Method for Information Hiding in Arabic Text. *IEEE International Conference on Innovations in Information Technology (IIT).* Abu Dhabi, UAE.

Bhattacharyya, S., Banerjee, I., & Sanyal, G. (2010). A novel approach of secure text based steganography model using word mapping method. *International Journal of Computer and Information Engineering*, vol.4, pp. 96-103.

Brassil, J. T., Low, S. H., Maxemchuk, N. F., & Gorman, L. O. (1955). Electronic marking and identification techniques to discourage document copying. *IEEE Journal on Selected Areas in Communication,* vol.1, pp. 1495-1504.

Chaudhary, S., Dave, M., Sanghi, A., & Manocha, J. (2016). An Elucidation on Steganography and Cryptography. *Second International ACM Conference on Information and Communication Technology for Competitive Strategies (ICTCS '16)*, (p. Article No. 43). Udaipur, India.

Cummins, J., Diskin P., Lau S., & Parlett R. (2004). Steganography and digital watermarking. *School of Computer Science*, pp.1-24.

Das, R., & Tuithung, T. (2012). A Novel Steganography Method for Image Based on. *Emerging Trends and Applications in Computer Science (NCETACS) 3rd National Conference*, (p. 5).

Devi, K. J. (2013). *A Sesure Image Steganography Using LSB Technique and Pseudo Random Encoding Technique*. National Institute of Technology-Rourkela .

Gutub, A. A.-A., & Fattani, M. M. (2007). A Novel Arabic Text Steganography Method using Letter points and Extension. *WASTET International Conference on Computer, Information and System Science and Engineering (ICCISSE)*, (pp. 28–31). Vienna, Austria.

Hamdan, A. M. (2018, February 8). AH4S: an algorithm of text in text steganography using. *Wiley Online Library*, p. 13.

Hussein, H. A. (2015). *Multi-Level Image Steganography by Using Pixel Intensity.* Sudan University of Science and Technology.

Kabetta, H., Dwiandiyanta, B. Y., & Suyoto (2011). Information hiding in CSS: a secure scheme text steganography using public key cryptosystem. *International Journal on Cryptography and Information Security,* vol.1, pp. 13-22.

Karim, S. M., Rahman, M. S., & Hossain, M. I. (2011). A new approach for LSB based image steganography using secret key. *14th International Conference on Computer and Information Technology (ICCIT 2011)*, (p. 6). Dhaka, Bangladesh .

Khairullah, M. (2011). A novel text steganography system in cricket match scorecard. *International Journal of Computer Applications*, vol.21, pp. 43-47, 2011.

Khairullah, & Md (2018). A novel steganography method using transliteration of bengali text. *Journal of King Saud University – Computer and Information Sciences*.

Kumar, R., Malik, A., Singh, S., & Chand, S. (2016). A high Capacity Email Based Text Steganography Scheme Using Huffman Compression. *In: Proceedings of IEEE Conference on Signal Processing and Integrated Networks (SPIN),* Noida, India, p 1–7.

Liu, T. Y., & Tsai, W. H. (2007). A new steganographic method for data hiding in Microsoft word documents by a change tracking technique. *IEEE Transactions on Information Forensics and Security*, vol.2, no.1, pp. 24-30.

Low, S. H., Maxemchuk, N. F., Brassil, J. T., & Gorman, L. O. (1995). Document marking and identification using both line and word shifting. *INFOCOM'95 Proceedings of the Fourteenth Annual Joint Conf. of the IEEE Computer and Communication Societies*, pp. 853-860.

Malalla, Suhad, shareef, & Farah R. (2016). Improving hiding security of Arabic text steganography by hybrid AES cryptography and text steganography. *International Journal of Eng. Res. Appl.*, p 60–69.

Malik, Aruna, Sikka, Geeta, Verma, Harsh K. (2017). A high capacity text steganography scheme based on LZW compression and color coding. *In the International Journal Eng. Sci. Technol*.

Memon, Ahmed, G., Kowhai, Karman, Kazi, & Hameedullah. (2008, April). Evaluation of steganography for Urdu/Arabic text. *J. Theor. Appl. Inf. Technol. (JATIT)*, p. 4.

Mohamed, A. (2014). An improved algorithm for information hiding based on features of Arabic text: a Unicode approach. *Egyptian Informatics Journal*, pp. 79–87.

Mohammed, A., Sameh, A., Abdul Rahman, E., & Adnan, G. (2007). Arabic Diacritics Based Steganography. *IEEE International Conference on Signal Processing and Communications (ICSPC'07),* pp. 756–759, Dubai, UAE.

Muhammad, K., Ahmad, J., Farman, H., & Zubair , M. (2014, 5 22). A Novel Image Steganographic Approach for Hiding. *Middle-East Journal of Scientific Research*, pp. 647-654.

Nagarhalli, Tatwadarshi P., Bakal, J.W., Jain, & Neha (2016). A survey of hindi text steganography. *In the International Journal of Sci. Eng. Res.*

Nofaie, Meteb, S., Gutub, & Adnan. (2016). Merging two steganography techniques. *J. Comput. Sci. Comput. Math.(JCSCM)*, p. 6.

Odeh, A., & Elleithy, K. (2012). Steganography in Arabic text using zero width and Kashidha letters. *International Journal of Computer Science & Information Technology*, p. 1.

Odeh, A., & Elleithy, K. (2013). Steganography in Text by Merge ZWC and Space Character. *Proceedings of the 28th International Conference on Computers and Their Applications (CATA-2013).* Honolulu, Hawaii, USA.

Odeh, A., Elleithy, K., & Faezipour, M. (2014). A Reliable and Fast Real-Time Hardware Engine for Text Steganography. *Proceedings of the IEEE Long Island Systems, Applications and Technology Conference (LISAT 2014).* Long Island, New York, USA.

Odeh, A., Elleithy, K., & Faezipour, M. (2013). Steganography in Arabic Text Using Kashida Variation Algorithm (KVA). *Proceedings of the IEEE Conference on Systems, Applications and Technology (LISAT).* Long Island, USA.

Petitcolas, F. A., Anderson, R., & Kuhn, M. G. (1999). Information Hiding - A Survey. *Proceedings of the IEEE, special issue on protection of multimedia content*, (pp. 1062-1078).

POR, L., & Delina, B. (2008). *Information hiding- a new approach in text steganography.* Hangzhou, China: 7th WSEAS Int. Conf. on APPLIED COMPUTER & APPLIED COMPUTATIONAL SCIENCE (ACACOS '08).

Qin, Chuan, Change, Chin-Chen, Wang, & Szu-Ting (2014). A novel lossless steganographic scheme for data hiding in traditional chinese text files. *In the Journal of Inf. Hiding Multimedia Signal Process.*

Rabah, K. (2004). Steganography-the art of hiding data. *Information Technology Journal*, vol.3, pp.245-269.

Shirali-Shahreza, M. H., & Shirali-Shahreza, M. (2006). A New Approach to Persian/Arabic Text Steganography. *Proceedings of the 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*, (pp. 310-315).

Shirali-Shahreza, M. H., & Shirali-Shahreza, M. (2008). A New Synonym Text Steganography. *International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, (pp. 1524-1526). Harbin, China.

Shirali-Shahreza, M., & Shirali-Shahreza, M. H. (2007). Text Steganography in SMS. *Convergence Information Technology, International Conference*, (pp. 2260-2265).

Shirali-Shahreza, Mohammad, Shirali-Shahreza, & Sajad. (2008). Steganography in TeX Document. *3rd IEEE International Conference on Intelligent System and Knowledge Engineering (ISKE)*, (pp. 1363–1366). University Convention Center Xiamen, China.

Taha, A., Hammad, A. S., & Selim, M. M. (2018, July 22). A high capacity algorithm for information hiding in Arabic text. *Journal of King Saud University – Computer and Information Sciences*, p. 8.

Vidhya, P.M., Paulb, & Varghese (2014). A method for text steganography using malayalam text. *In the International Conference of Inf. Commun. Technol. (ICICT),* p 524–531.

Weiss, M. (2012). *Principles of steganography*.

Wu, H.-C., Wang, H.-C., Tsai, C.-S., & Wang, C.-M. (2010). Reversible image steganographic scheme via predictive coding., (pp. 35-43).

List of Publications

Abdulaziz Saleh Al-Dini, Naziha Mohammed Al-Aidroos, Khalid Kayid Shafel, "Data Hiding Enhancement in Arabic Text" **International Journal of Engineering and Artificial Intelligence (IJEAI),** Vol 2 No 1 (2021) 32–42.

الجمهورية اليمنية
وزارة التعليم العالي والبحث العلمي
جامعة الريان
كلية الدراسات العليا

جامعة الريان
AL-RAYAN UNIVERSITY

تحسين إخفاء البيانات في النصوص العربية

رسالة مقدمة إلى جامعة الريان؛ لاستكمال متطلبات نيل درجة الماجستير، تخصص تقنية المعلومات

إعداد
عبدالعزيز صالح عبدالله الديني

إشراف
د. خالد قائد شعفل
د. نزيهة محمد العيدروس

2021م / 1442هـ

الجمهورية اليمنية
وزارة التعليم العالي والبحث العلمي
جامعة الريان
كلية الدراسات العليا

تحسين إخفاء البيانات في النصوص العربية

رسالة مقدمة إلى جامعة الريان؛ لاستكمال متطلبات نيل درجة الماجستير، تخصص تقنية المعلومات

إعداد
عبدالعزيز صالح عبدالله الديني

إشراف
د. خالد قائد شعفل
د. نزيهة محمد العيدروس

2021م / 1442هـ