Enhancing Open Space Method in Data Hiding Technique – Text under Text

Yaman F. Abdullah, Hebah H. O. Nasereddin
MEU—Middle East University
Amman, Jordan
{yamanfakhry, hebah66}@hotmail.com

ABSTRACT: *We are involved in this research study methodology for hide data within different texts. That's where the data masking process is one of the areas of information security to hide and encrypt data and other branches that represent information security. The process of hiding data within the text are completely different from the rest of methods of concealment, it will be through a change in the empty spaces in the text or change the text itself or change in some of the properties other than text and empty spaces. These changes are a way by which to hide data within the text. This research proposes an enhancement of white space method for hiding data, which is processed by changing the secret text through extracting the indexes of the characters from the cover text or ASCII code, then converting these numbers from the decimal numeral system into the octal numeral system in order to use the number 8 and 9 as indicators against the remaining numbers. Then, merge these outcomes with the white spaces between the words in the cover text by changing the font size of these numbers to 1pt, and changing the font color to match background color of the cover text.*

Keywords: Steganography, Host Signal, Signal Encoder, White Space, syntactic, semantic, Microsoft Word, UniSpaCh, Justified Format, Secret Text, Cover Text

Received: 26 May 2013, Revised 23 July 2013, Accepted 29 July 2013

1. Introduction

Information security has two branches; data encryption and data hiding (DH). Data encryption is masking the data to become meaningless, while data hiding is concerned with concealing the data to become unreadable.

Data hiding is: "*Data hiding represents a class of processes used to embed data, such as copyright information, into various forms of media such as image, audio, or text with a minimum amount of perceivable degradation to the "host" signal* ". [1]

"*Famous examples of steganography go back to antiquity. According to a story from Herodotus, a slave's head was shaved by his master, Histiæus, and tattooed with a secret message around 440 B.C. After growing the hair back, the message disappeared and then the slave journeyed to carry the message. When shaved his head upon arriving, message was revealed* ". [2] "*In* 1860, *the major problems had been solved to make a small image by "Darjun", who is a French photographer worked in the war. Frank and France in* 1870- 1861 *when Paris was besieged, by writing messages on photographic films which were sent by the carrier pigeon. The purpose of this was to invoke disobedience against his antagonist Persinas. Steganography is the ability of hiding data in redundant bits of any cover media. Its target is to keep the secret information unreadable without damaging the cover media environment.*". [3] "*A security issue must be used in each step. Information hiding can be used in different* *applications include military, E- commerce, confidential communication, copyright protection, copy control, authentication, digital elections. In these fields, hiding information is better than ciphering. Because in the former, nobody can notice that there is a message hiding behind an image*". [4]

2. Related Works

The technique is based on two texts; secret text and cover text. By taking advantage of the unused white spaces in the cover text, this paper proposed to change the format of the secret text by changing the secret text font size and font color to the font size 1pt and secret text font color white, then to separate the secret text into many parts "*characters*", then to hide these parts within the cover text's unused white spaces. The outlined that the proposed data hiding technique as Image, audio, and text are used for data hiding. Data hiding in text is to embed text within another text to be unreadable. Open space methods used for data hiding in text and white space method is one of these methods; the paper takes advantages of the unused white space from the text "*Cover Text*" to hide the data "*Secret Data*" on the cover text. Changing the format of the secret by setting the text size to 1px, setting the font color to white as the back color of cover text, and then merging the secret text with the cover using white space method to generate the result text hiding the secret message within it. Figure 1 below shows a brief description of the proposed technique. [5]

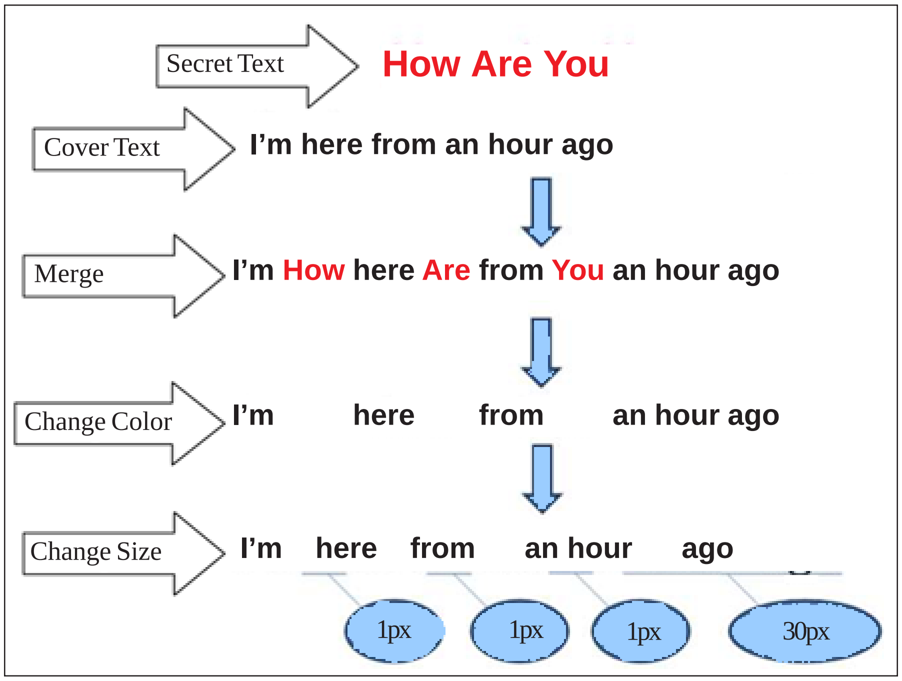

Secret Text
How Are You
Cover Text
I'm here from an hour ago
Merge
I'm How here Are from You an hour ago
Change Color
I'm here from an hour ago
Change Size
I'm here from an hour ago
1px
1px
1px
30px

Figure 1. Proposed data hiding technique – Text under Text

Based on Microsoft Word Document to hide date within text by using DASH to indicate the location of the hidden text, and by using UniSpaCh technique to counter DASH AND analyze it, the Unicode space characters are inserted between words, lines and end of lines. UniSpaCh also used to retrieve the embedded information and reconstruct the original Microsoft Word document. [6]

Proposed a technique to take advantage of physical file format to store files in the system, the proposed technique used the unused blocks in Microsoft Compound Document File Format (MCDFF). The proposed system embeds text in the file structure (Binary File Format) of document file which is a file of Microsoft Word Document file. [7]

Proposed a new theoretical framework for data hiding which is concerned with text format feature manipulation to hide text within text by quantizing the color or luminance intensity for each character in a way the human cannot distinguish the modification in the original characters. [8]

3. Methodologies

3.1 Character Indexes

Character Indexes used in the proposed solution to get the index of secret characters from the cover text to be able to retrieve the secret text in the stage of the text show.

Please be informed that the character indexes begin count from 0.

Example, represent character indexes in the text. Suppose the cover text is: **Information hiding techniques**

The characters index for above sentence is appearing in the table 1:

Indexes	Characters set	Indexes	Characters set
0	I	15	i
1	n	16	n
2	f	17	g
3	o	18	
4	r	19	t
5	m	20	e
6	a	21	c
7	t	22	h
8	i	23	n
9	o	24	i
10	n	25	q
11		26	u
12	h	27	e
13	i	28	s
14	d		

Table 1. Character indexes for cover text

So, from above example, suppose the secret text is: hi man

Table 2 shows the index of the secret text characters:

Indexes	Characters set
12	h
0	i
11	(space)
5	m
6	a
1	n

Table 2. Character indexes for secret text

The "*h*" character is repeated twice in the cover text; in this case, the index of the first one is enough to know the intended character.

3.2 ASCII Code Characters

The characters' indexes are not enough to indicate the intended character from the cover text, so, ASCII code characters give a unique code for each character in the secret text "*with sensitive case*". Any character exists in the secret text; not exists in the cover text; will take the character code from the ASCII code characters.

3.3 Octal Numeral System

The octal numeral system used to convert the decimal codes as octal numbers for:

1. Security reasons, in case anyone finds out the hidden data, the data will be meaningless and unreadable. The secret text will be formatted then merged within the cover text.

2. Identify character, after getting the character index for the secret text as octal numbers; these numbers will be separated by inserting the numbers eight and nine between the octal numbers to dedicate each character number.

4. Proposed solution

4.1 Format Secret Text

Formatting secret text by extracting the index of the secret text characters from cover text and convert the index numbers from decimal numeral system into octal numeral system as the following:

Step One: Index of the secret text characters:

As mentioned above; there are two texts used in the proposed solution, secret text and cover text, the proposed solution will hide the secret text within the cover text.

This section will show in detail how to extract the index of the secret text character from the cover text by using the following steps:

First. Taking each character from the secret text.

Second. Looking for character in the cover text.

Third. Taking the index of the character from the cover text.

Fourth. Compilation of the indexes taken from the cover text in the array.

For more clarification; below is an example to get index of the secret text character from the cover text:

Suppose the cover text is: **Information hiding techniques**

And the secret text is: **hi man**

The table 3 shows the index for each character in the cover text while table 4 show the index of the secret text characters in the cover text and it's shown in the highlighted columns.

I	n	f	o	r	m	a	t	i	o	n		h	i	d	i	n	g		t	e	c	h	n	i	q	u	e	s
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28

Table 3. Characters index for cover text

I	n	f	o	r	m	a	t	i	o	n		h	i	d	i	n	g		t	e	c	h	n	i	q	u	e	s
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28

Table 4. Characters index for secret text in the cover text

The steps of take these indexes will be like the following:

I. Create an empty array for secret characters: "***hi man***" is 6 characters so the array length is 6.

II. Take the first character from the secret text "*h*".

III. Look for "*h*" character in the cover text; what is needed is the first "*h*" character of the cover text.

IV. Add the index of "*h*" character in the array:

V. Take the second character from the secret text "*i* ".

VI. Look for "*i* " character in the cover text; what is needed is the first "*i* " character of the cover text.

VII. Add the index of "*i*" character in the array:

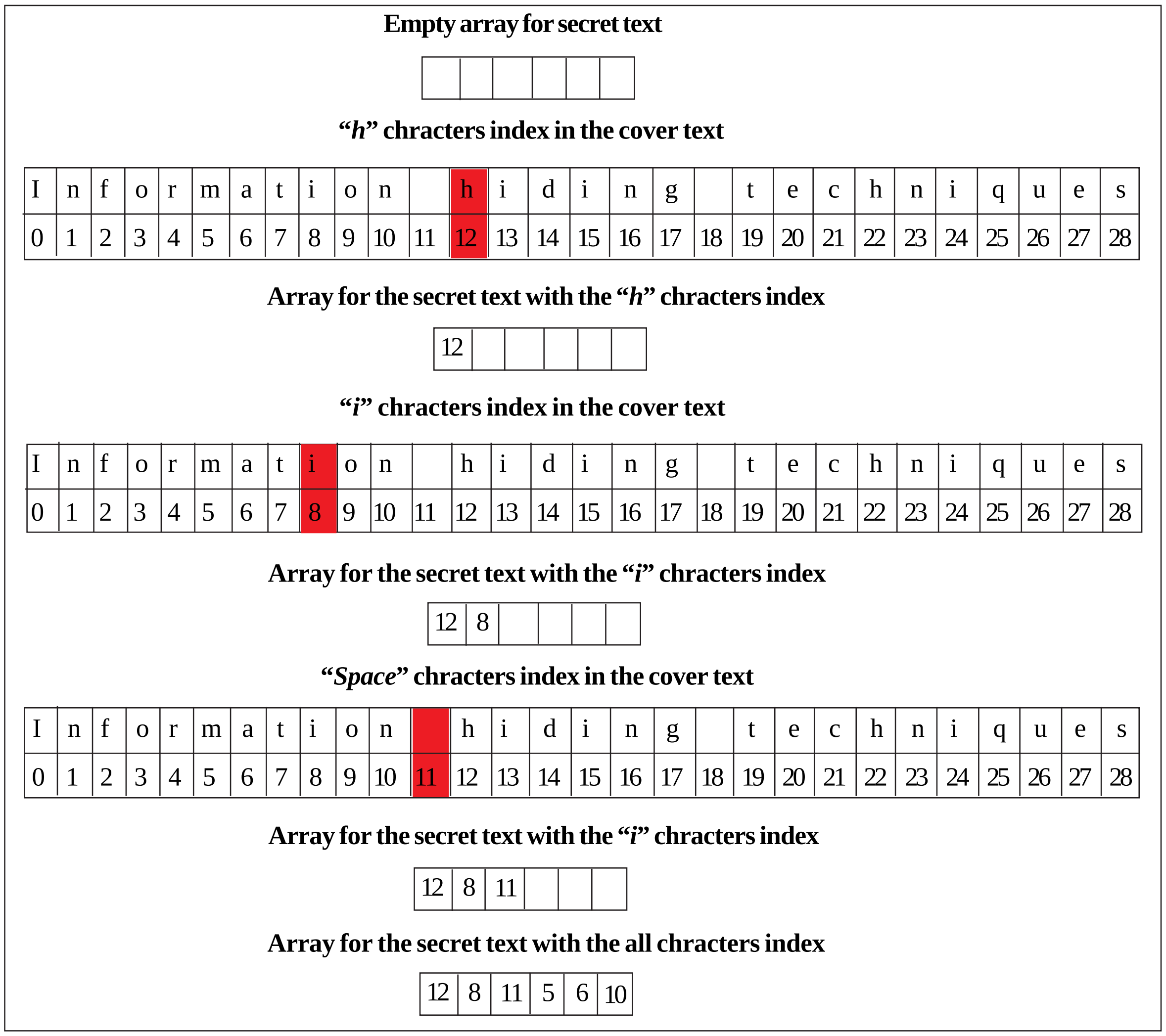
VIII. Take the third character from the secret text " "; its space.

IX. Look for "*space*" character in the cover text; what is needed is the first "*space*" character of the cover text.

X. Add the index of "*space*" character in the array.

XI. Do the above steps for all secret characters until fill the index array of secret text

After these steps, the cover text contains all the characters in the secret text and the indexes is taken from the cover text and saved on the array, Figure 2 will show the steps of taking the indexes.



Empty array for secret text

"*h*" chracters index in the cover text

I	n	f	o	r	m	a	t	i	o	n		**h**	i	d	i	n	g		t	e	c	h	n	i	q	u	e	s
0	1	2	3	4	5	6	7	8	9	10	11	**12**	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28

Array for the secret text with the "*h*" chracters index

12					

"*i*" chracters index in the cover text

I	n	f	o	r	m	a	t	**i**	o	n		h	i	d	i	n	g		t	e	c	h	n	i	q	u	e	s
0	1	2	3	4	5	6	7	**8**	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28

Array for the secret text with the "*i*" chracters index

12	8				

"*Space*" chracters index in the cover text

I	n	f	o	r	m	a	t	i	o	n		h	i	d	i	n	g		t	e	c	h	n	i	q	u	e	s
0	1	2	3	4	5	6	7	8	9	10	**11**	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28

Array for the secret text with the "*i*" chracters index

12	8	11			

Array for the secret text with the all chracters index

12	8	11	5	6	10

Figure 2. Steps of taking the indexes

Step Two: ASCII Code Characters of the secret text characters:
Get the character code of the secret text characters from the ASCII code characters in case the character in the secret text does not exist in the cover text characters or it does not exist in the cover text characters with the same capitalization.

In these cases, follow the below proposed solution:

First. Take each character from the secret text.

Second. Look for the character in the cover text.

Third. If the character does not exist in the cover text or exists with different capitalization; get the character code from the

ASCII code characters as it's listed in Table 3-3.

Fourth. Compile the indexes taken from the cover text in the array.

For more clarification; the below is example to get index of the secret text character from the cover text:

Suppose the cover text is: **Information hiding techniques**

Suppose the secret text is: **Hi man**

The table 5 show the index for each character in the cover text while table 6showthe index of the secret text characters in the cover text and it's shown in the highlighted columns but "*H*" character does not exists in the cover text with the same capitalization.

I	n	f	o	r	m	a	t	i	o	n		h	i	d	i	n	g		t	e	c	h	n	i	q	u	e	s
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28

Table 5. Characters index for cover text

I	n	f	o	r	m	a	t	i	o	n		h	i	d	i	n	g		t	e	c	h	n	i	q	u	e	s
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28

Table 6. Characters index for secret text in the cover text

In this case; the solution is to take the character code for "*H* " character from the ASCII code characters.

The steps to take these indexes will be similar to the steps in step one, but the "*H* " character will take its code from ASCII code characters. And the result array of the secret text characters will be as it appears in the Table 7:

72	8	11	5	6	10

Table 7. Array for secret text with the all characters index

The first column "72" is the ASCII code for uppercase "*H* " character, but the other columns in the index of the remaining secret text characters are taken from the cover text characters.

Once reaching the phase to retrieve the hidden data from the cover text, indexes should be distinguished from the ASCII codes to be able determine the source of the character of the given number "*know the source of the number either it's index or ASCII code*"; so; what if there is number 72 in the cover text for another character? The problem is solved in step three.

Step Three: Octal numeral system:
After collecting the index of the secret character from the cover text and ASCII code for the characters of the secret text that do not exist in the cover text characters with the same capitalization; convert the result array of numbers from decimal numeral system into octal numeral system.

The conversion shall follow the below procedures:

1- Every character of the secret text exists in the cover text characters will be taken from the index of these characters. Then, convert this index from decimal numeral system into octal numeral system.

2- Every character of the secret text that does not exist in the cover text characters will be taken from the ASCII codes, then follow the below steps:

a. Convert the ASCII code from decimal numeral system into octal numeral system.

b. Add number "8" begin of the converted octal number; this step to distinguish the number is from ASCII code characters.

3- Compile the indexes and the ASCII codes for secret text characters in the same order in the text by inserting the number "9" between these numbers to separate the characters indexes and ASCII codes from each other.

For more information; the octal number doesn't contain the 8 and 9 numbers. This feature enables us to use those numbers to:

1- Separate between characters indexes and ASCII codes.

2- Distinguish that the number is coded from ASCII code characters or index from secret text characters index in the cover text characters.

4.2 Hide the secret text:
The used way to hide the secret text is to merge the result array number of secret text characters index and the ASCII code within the cover text in the unused white spaces between cover text words.

The data will be hidden with font size 1pt and color as cover text back color, by follows these steps:

Step One: Detect the number of digits:
To merge the result array number of secret text characters index and ASCII code within the cover text in the unused white spaces between cover words; you must specify how many digits should be inserted between each word in the unused white spaces.

The number of digits determined depends on the font size; for each font size there is a space size between words, these spaces will have the secret data and the size of this data should be the same of the empty space size in the cover text.

The font size scale is the point; "*In typography, a point is the smallest unit of measure, being a subdivision of the larger pica. It is commonly abbreviated as pt. The point has long been the usual unit for measuring font size and leading and other minute items on a printed page. The original printer point, from the era of foundry metal typesetting and letter press printing, varied between* 0.18 *and* 0.4 *mm depending on various definitions of the foot. By the end of the* 19th *Century, it had settled to around* 0.35 *to* 0.38 *mm, depending on one's geographical location.*

In the late 1980*s to the* 1990*s, the traditional point was supplanted by the desktop publishing point (also called the PostScript point), which was defined as* 72 *points to the inch (*1 *point =* 1*D* 72 *inches =* 25.4*D* 72 *mm =* 0.3527 *mm). In either system, there are* 12 *points to the pica. In metal type, the point size of the font described the size (height) of the metal body on which the typeface's characters were cast. In digital type, the body is now an imaginary design space, but is used as the basis from which the type is scaled.*

*A measurement in picas is usually represented by placing a lower case p after the number, such as "*10*p" meaning "*10 *picas." Points are represented by placing the number of points after the p, such as* 0*p*5 *for "*5 *points,"* 6*p*2 *for "*6 *picas and* 2 *points," or* 1*p*1 *for "*13 *points" which is converted to a mixed fraction of* 1 *pica and* 1 *point. (An alternate nomenclature is described in the pica article*.)". [9]

The space between words will be different depending on the font size. "*Sentence spacing is the horizontal space between sentences in typeset text. It is a matter of typographical convention. Since the introduction of movable-type printing in Europe, various sentence spacing conventions have been used in languages with a Latin-derived alphabet. These include a normal word space (as between the words in a sentence), a single enlarged space, two full spaces, and, most recently in digital media, no space. Although modern digital fonts can automatically adjust a single word space to create visually pleasing and consistent spacing following terminal punctuation, most debate is about whether to strike a keyboard's spacebar once or twice between sentences. Traditionally, two spaces could distinguish from a mid-sentence abbreviation or initials, as in, "He was faster than I. P. Jones was next.*"" [10]

According to results shown in chapter four; the best number of digits with size 1pt to hide within the cover text is as appearing in table 8.

Step Two: Merge the index and ASCII code numbers in the cover text:
After detecting the number of digits according to the cover text font size; merge these digits within the cover text in the white spaces, this operation will be done in iterations.

Before that, the indexes array must be done and ready to work on it and hide it, so; suppose the following: The secret text is:

"*Become important in $ a number of application areas*"

The cover text is shown in figure 3.

Cover text font size	Number of hidden digits
10	1
12	2
14	3
16	4

Table 8. The best number of digits with size 1pt to hide within the cover text

ABSTRACT

Information hiding techniques have recently become important in a nimber of application areas. there are many techniques to achieve hididng data, and hiding text inside images is one field of them. The paper gives short example of these techniques and proposes a technique to guide text inside digital image by the concept of the visual representation of the text within image.

Keywords : ***Hiding data, Steganography, Visual representation of text, cover image, Stego image. 1.***

INTRODUCTION

In the world data hiding science is entirely separated from the science of data encryption cryptograpy. It is called Steganograpy. " Famous examples of steganography go back to antiquity. According to a story from Herodotus, a salve's head eas shaved by his master, Histiaeus, ans tattooed with a secret message around 440 B.C.

Figure 3. The cover text

ABSTRACT

Information 29 hiding 36 techniques 93 have 79 recently 15 become 91 important 79 in 36 a 92 number 59 of 22 application 91 areas, 79 there 77 are 91 many 59 techniques 16 to 92 achieve 19 hiding 20 data, 91 and 39 hiding 21 text 92 inside 59 image 22 is 9 lone 39 field 25 of 84 them. 49 The 25 paper 92 gives 09 short 25 example 91 of 39 these 44 techniques 91 nad 79 proposes 66 a 93 new 69 technique 16 of 92 hide 59 text 15 inside 91 digital 49 image 25 by 92 the 09 concept 77 of 97 the 79 visual 63 representation 92 of 29 the 37 text 92 within 09 image.

Keywords: 21 Hiding 92 data, 29 Steganograpgy, 15 Visual 91 representation 39 of 25 text, 92 cover 09 image, 16 stego 93 image. 69 1. 20

INTRODUCTION

In 94 the 6 world Data hiding science is entirely sepaparated from the science of data encryption cryptography. It is called Steanography. " Famous examples of steganography go back to antiquity. According to a story from Herodotus, a salve's head eas shaved by his master, Histiaeus, ans tattooed with a secret message around 440 B.C.

Figure 4. Insert digits in the white space of the cover text

The font size of the cover text is 12pt, so; the best number of digits to hide in each white space is 2 digits "*as explained in step one*".

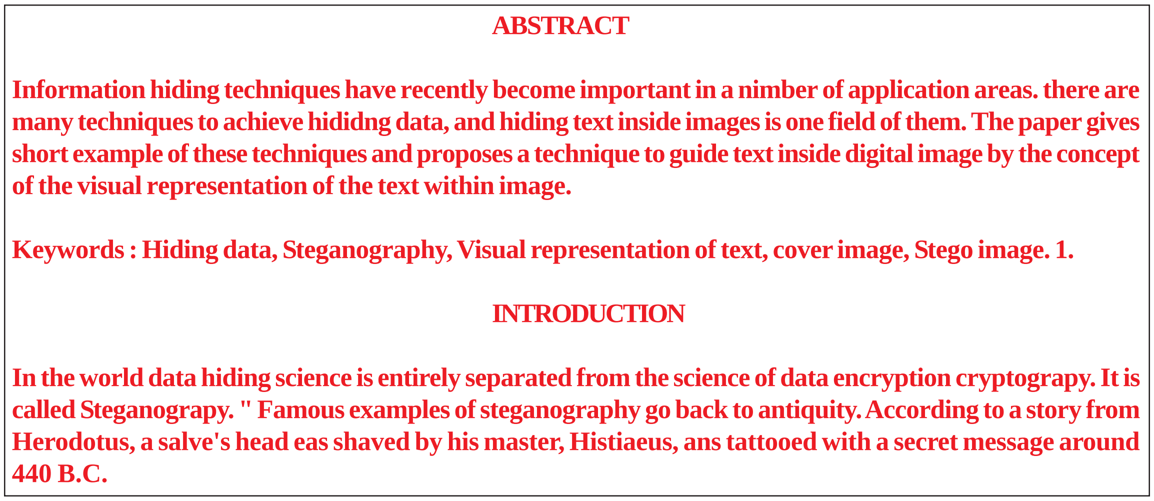
The result array of the indexes and ASCII codes for the secret text characters in the cover text in octal numeral system is:
“29369379159179369259229179779159169219209139219259229139258449259209259139449179669369169259159149259209779779
6392293792092192291591392592091693692094 6”

This array will be divided into two digits together and then merge it like the following:
“29, 36, 93, 79, 15, 91, 79, 36, 92, 59, 22, 91, 79, 77, 91, 59, 16, 92, 19, 20, 91, 39, 21, 92, 59, 22, 91, 39, 25, 84, 49, 25, 92, 09, 25, 91, 39, 44, 91, 79, 66, 93, 69, 16, 92, 59, 15, 91, 49, 25, 92, 09, 77, 97, 79, 63, 92, 29, 37, 92, 09, 21, 92, 29, 15, 91, 39, 25, 92, 09, 16, 93, 69, 20, 94, 6”

Every two digits will be insert in white spaces of cover text is shown in Figure 4.

Step Three: Change font size and color for the numbers:
All inserted digits will be format as font size 1pt and color as cover text back color “*white color*”.

The result of change font size to 1pt is shown in figure 5.



ABSTRACT

Information hiding techniques have recently become important in a nimber of application areas. there are many techniques to achieve hididng data, and hiding text inside images is one field of them. The paper gives short example of these techniques and proposes a technique to guide text inside digital image by the concept of the visual representation of the text within image.

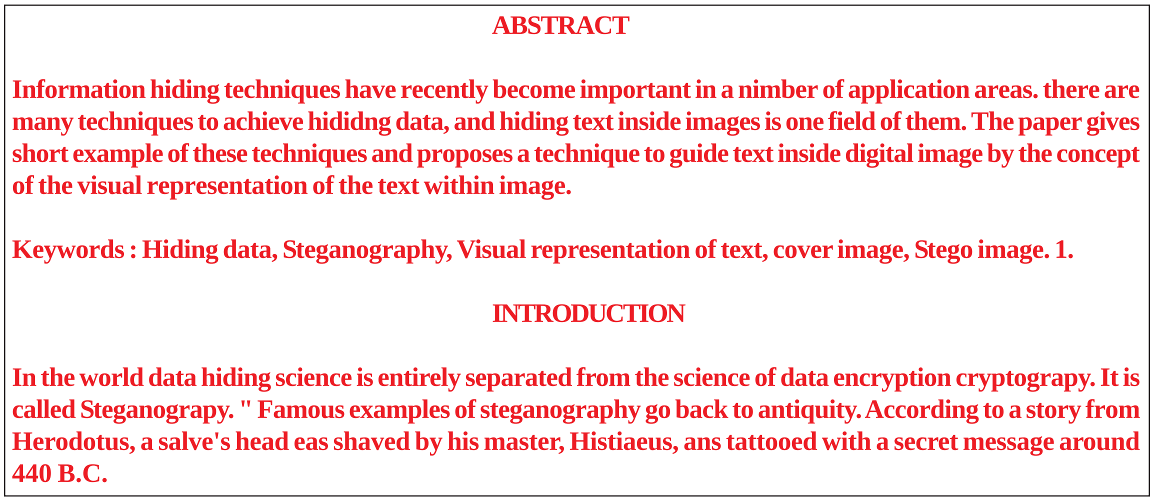
Keywords : Hiding data, Steganography, Visual representation of text, cover image, Stego image. 1.

INTRODUCTION

In the world data hiding science is entirely separated from the science of data encryption cryptograpy. It is called Steganograpy. " Famous examples of steganography go back to antiquity. According to a story from Herodotus, a salve's head eas shaved by his master, Histiaeus, ans tattooed with a secret message around 440 B.C.

Figure 5. The result with hidden data font size 1pt

Then will change the color of the digits as the back color of the cover text “*white color*”.
The final result of hidden secret text in the cover text is shown in figure 6:



ABSTRACT

Information hiding techniques have recently become important in a nimber of application areas. there are many techniques to achieve hididng data, and hiding text inside images is one field of them. The paper gives short example of these techniques and proposes a technique to guide text inside digital image by the concept of the visual representation of the text within image.

Keywords : Hiding data, Steganography, Visual representation of text, cover image, Stego image. 1.

INTRODUCTION

In the world data hiding science is entirely separated from the science of data encryption cryptograpy. It is called Steganograpy. " Famous examples of steganography go back to antiquity. According to a story from Herodotus, a salve's head eas shaved by his master, Histiaeus, ans tattooed with a secret message around 440 B.C.

Figure 6. The result with hidden data font size 1pt and white color

In case the cover text back color is not white; the proposed solution will change the secret text color as cover text back color.

Original file Text Size	Hidden Process Speed in Seconds
10pt	48.8557944
12pt	59.0033748
14pt	61.5805222
16pt	63.3666244

Table 9. Jebran implementation speed results of hiding secret text

Original file Text Size	Number of hidden digits in white space	Hidden Process Speed in Seconds
10pt	1	0.8620493
10pt	2	0.5160296
10pt	3	0.283016 2
10pt	4	0.2090119
10pt	5	0.2120121
12pt	1	0.9030516
12pt	2	0.4840276
12pt	3	0.2530145
12pt	4	0.2070118
12pt	5	0.1660095
14pt	1	0.8940511
14pt	2	0.4170238
14pt	3	0.3230185
14pt	4	0.1990114
14pt	5	0.1700097
16pt	1	0.908052
16pt	2	0.4210241
16pt	3	0.2740157
16pt	4	0.2290131
16pt	5	0.1590091

Table 10. The proposedimplementationspeed results of hiding secret text

Original file Text Size	Original file Size in K.B	Result file Size in K.B
10pt	4	6
12pt	4	4
14pt	4	4
16pt	4	6

Table 11. Jebran implementation size of the result text for hiding the secret text

Original file Text Size	Number of hidden digits in white space	Original file Size in K.B	Result file Size in K.B
10pt	1	4	13
10pt	2	4	9
10pt	3	4	8
10pt	4	4	7
10pt	5	4	7
12pt	1	4	13
12pt	2	4	9
12pt	3	4	8
12pt	4	4	7
12pt	5	4	7
14pt	1	4	13
14pt	2	4	9
14pt	3	4	8
14pt	4	4	5
14pt	5	4	5
16pt	1	4	7
16pt	2	4	6
16pt	3	4	5
16pt	4	4	5
16pt	5	4	5

Table 12. The proposed implementation size of the result text for hiding the secret text

5. Results

5.1 Comparing the Proposed Solution with White Space Method

White space is part of data hiding text into text. Some of white space problems are.
To hide two words like "*Top Secret*" requires text size cover of more than 80 words; because each character size is 8 bit "1 *byte*" and each bit requires one space. That means "*T + o + p + + S + e + c + r + e + t*" equal 10 characters, 10 characters multiply by 8 bits equal 80 bits.

To be able to hide a large secret message; the result will be a very large message.

In a properly justified format of text, not all spaces are available to be used to hide the required data.

By using the proposed solution; above problems are solved, and the result is:

To hide two words like "*Top Secret*" in text size 12pt and by using 2 digits; requires text size cover not more than 20 words in case the result of each character index or ASCII code is 4 digits; that means "*T + o + p + +S + e + c + r + e + t*" equal 10 characters, each character index 4 digits multiply by 10 characters equal 40 digits, 40 digits divided into two digits equal 20 parts, each part disappears in one white space.

To be able to hide a large secret message; the result will be close to the size of the original text.

In a properly justified format of text, all spaces are available to be used to hide the required data.

5.2 Evaluating the Speed of the Data Hiding Process

The speed process is different from case to another, the cases of evaluating the speed process are: hiding secret text under cover text using cover text size 10pt, hiding secret text under cover text using cover text size 12pt, hiding secret text under cover text using cover text size 14pt and hiding secret text under cover text using cover text size 16pt.

The comparison will be between the proposed implementation and the other implementation is done by [11].

Original file Text Size	Result Text is Match Original Text?
10pt	FALSE
12pt	FALSE
14pt	FALSE
16pt	FALSE

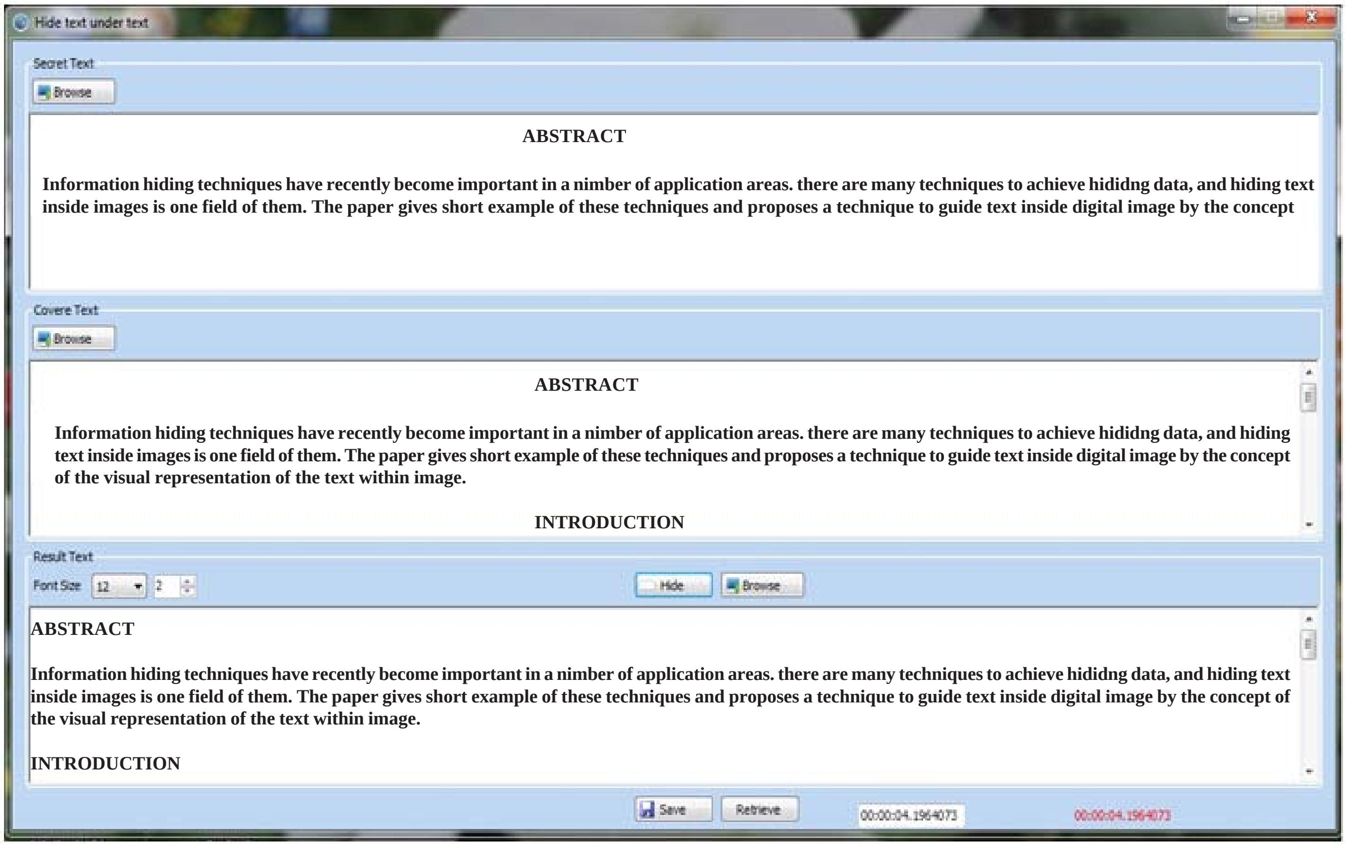
Table 13. Jebran implementation speed results of hiding secret text

Original file Text Size	Number of hidden digits in white space	Hidden Process Speed in Seconds
10pt	1	TRUE
10pt	2	FALSE
10pt	3	FALSE
10pt	4	FALSE
10pt	5	FALSE
12pt	1	FALSE
12pt	2	TRUE
12pt	3	FALSE
12pt	4	FALSE
12pt	5	FALSE
14pt	1	FALSE
14pt	2	FALSE
14pt	3	TRUE
14pt	4	FALSE
14pt	5	FALSE
16pt	1	FALSE
16pt	2	FALSE
16pt	3	FALSE
16pt	4	TRUE
16pt	5	FALSE

Table 14. The proposed implementation speed results of hiding secret text

5.5 Hide Data

The result of hide data by using the proposed Data Hiding Technique – Text under Text in figure 7:



ABSTRACT

Information hiding techniques have recently become important in a nimber of application areas. there are many techniques to achieve hididng data, and hiding text inside images is one field of them. The paper gives short example of these techniques and proposes a technique to guide text inside digital image by the concept

ABSTRACT

Information hiding techniques have recently become important in a nimber of application areas. there are many techniques to achieve hididng data, and hiding text inside images is one field of them. The paper gives short example of these techniques and proposes a technique to guide text inside digital image by the concept of the visual representation of the text within image.

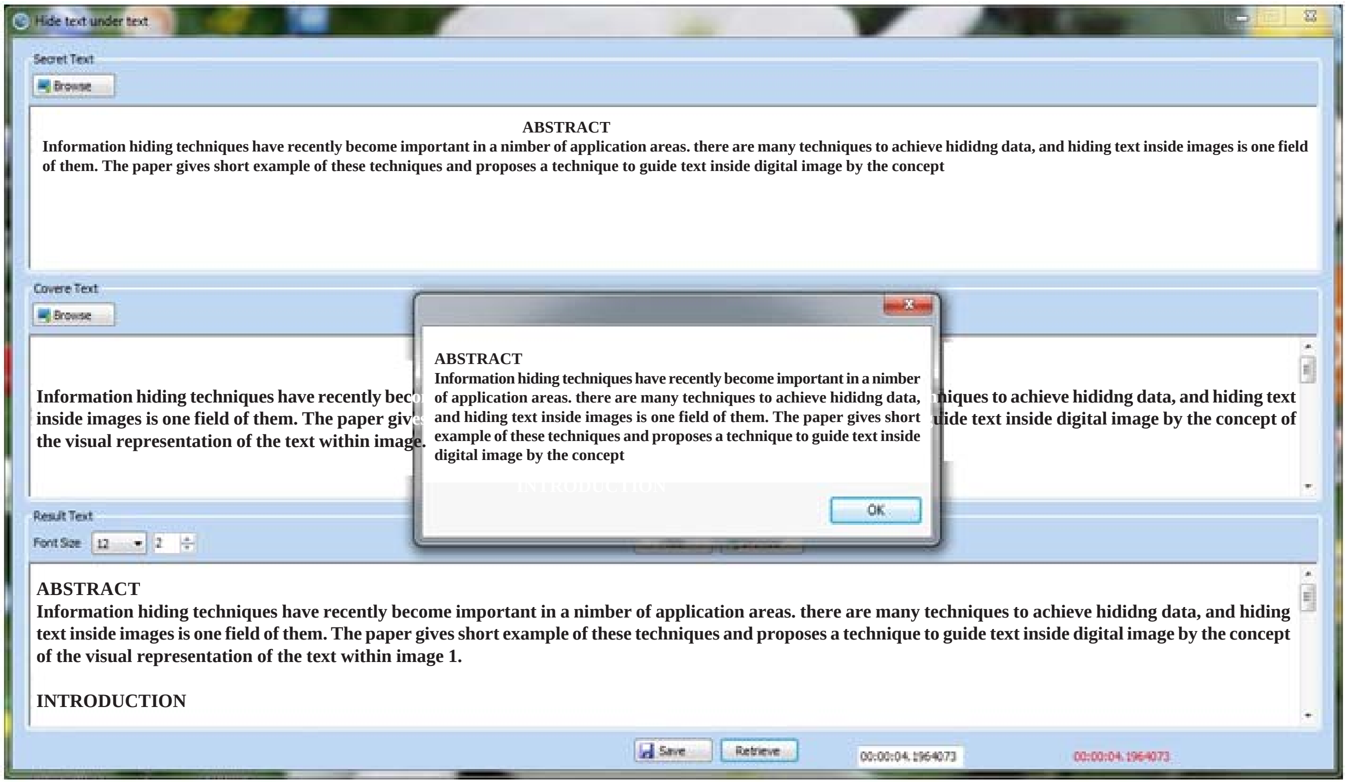
INTRODUCTION

ABSTRACT

Information hiding techniques have recently become important in a nimber of application areas. there are many techniques to achieve hididng data, and hiding text inside images is one field of them. The paper gives short example of these techniques and proposes a technique to guide text inside digital image by the concept of the visual representation of the text within image.

INTRODUCTION

Figure 7. The result of hide data by using the proposed Data Hiding Technique – Text under Text



ABSTRACT
Information hiding techniques have recently become important in a nimber of application areas. there are many techniques to achieve hididng data, and hiding text inside images is one field of them. The paper gives short example of these techniques and proposes a technique to guide text inside digital image by the concept

Information hiding techniques have recently beco… …niques to achieve hididng data, and hiding text inside images is one field of them. The paper giv… …uide text inside digital image by the concept of the visual representation of the text within image.

ABSTRACT
Information hiding techniques have recently become important in a nimber of application areas. there are many techniques to achieve hididng data, and hiding text inside images is one field of them. The paper gives short example of these techniques and proposes a technique to guide text inside digital image by the concept

ABSTRACT
Information hiding techniques have recently become important in a nimber of application areas. there are many techniques to achieve hididng data, and hiding text inside images is one field of them. The paper gives short example of these techniques and proposes a technique to guide text inside digital image by the concept of the visual representation of the text within image 1.

INTRODUCTION

Figure 8. Retrieve the hidden data using the proposed Data Hiding Technique – Text under Text

5.6 Retrieve Data

The retrieval of the hidden data using the proposed Data Hiding Technique – Text under Text is given in the figure 8.

6. Conclusions

Information security has two branches; data encryption and data hiding. Image, audio, and text are used for data hiding. Data hiding in text is to embed text within another text to be invisible. Digital media has become more prevalent and expanding.

There are three major methods for data hiding text under text; open space methods that encode through manipulation of white space (unused space on the printed page), syntactic methods that utilize punctuation, and semantic methods that encode using manipulation of the words themselves.

There are two reasons why the manipulation of white space in particular yields useful results in open space method. First, changing the number of trailing spaces has little chance of changing the meaning of a phrase or sentence. Second, a casual reader is unlikely to take notice of slight modifications to white space.

There are three methods of using white space to encode data. The methods exploit inter-sentence spacing by placing either one or two spaces after each terminating character, end-of-line spaces by insert spaces at the end of lines, the data are encoded allowing for a predetermined number of spaces at the end of each line, and inter-word spacing in justified text by controlling where the extra spaces are placed, one space between words is interpreted as a "0" two spaces are interpreted as a "1".

The character indexes; used in the proposed solution to get the index of secret characters from the cover text to be able to retrieve the secret text in the stage of the text show. The characters' indexes are not enough to indicate the intended character from the cover text, so; ASCII code characters give a unique code for each character in the secret text "*with sensitive case*".

The octal numeral system is the base-8 number system, and uses the digits 0 to 7. Octal numerals can be made from binary numerals by grouping consecutive binary digits into groups of three (starting from the right).

The proposed solution takes advantage of the unused white space from the text "*Cover Text*" to hide the data "*Secret Data*" on the cover text. Changing the format of the secret text by setting the text size to 1px, setting the font color to white as the back color of cover text, then extract the index of the secret text characters from cover text, the remaining secret text characters that do not exist in the cover text will generate a unique code for each none existing characters from ASCII code characters, then convert the result of indexes and ASCII codes from decimal numeral system into octal numeral system by separating the characters with the number "9" and identifying the ASCII code "*not index*" with the number "8", then merging the secret text with the cover text using white space method to generate the result text which is hiding the secret message within it.

References

[1] Bender, W., Gruhl, D., Morimoto, N., Lu, A. (1996). Techniques for data hiding. *IBM Systems Journal*, p. 313-336.

[2] YILMAZ, A. (2003). Robust VIDEO Transmission using Data Hiding. Master of Science, the graduate school of natural and applied sciences of the Middle East technical university, p. 20-30.

[3] Nasereddin, H. H., Al Farzaeai, M. S. (2010). Proposed Data Hiding Technique Text Image Inside Image (TIII). *International Journal of Research and Reviews in Applied Sciences*, p. 183- 193.

[4] Ibrahim, A., Zabian, A. (2009). Algorithm for Text Hiding in Digital Image for Information Security. *International Journal of Computer Science and Network Security*, 262-268.

[5] Abdullah, Y. F., Nasereddin, H. H. (2013). Proposed Data Hiding Technique – Text under Text. *American Academic & Scholarly Research Journal* (AASRJ), 243-248.

[6] Por, L. Y., Wong, K., Chee, K. O. (2012). A text-based data hiding method using Unicode space characters. *The Journal of Systems and Software*, 1075 – 1082.

[7] Rahma, A. S., AbdulWahab, H. B., Al-Noori, A. Y. (2011). Proposed Steganographic Method for Data Hiding in Microsoft Word Documents Structure. *Al-Mansour Journal*, p. 1 - 29.

[8] Vill´an, R., Voloshynovskiy, S., Koval, O., Vila, J., Topak, E., Deguillaume, F., et al. (2006). Text Data-Hiding for Digital and Printed Documents: Theoretical and Practical Considerations. *Stochastic Information Processing* (*SIP*), p. 15-26.

[9] Wikipedia. (2013, March 08). Point (typography). Retrieved May 02, from Wikipedia, the free encyclopedia: http://en.wikipedia.org/wiki/Point_(typography)

[10] Wikipedia. (2013, April 30). Sentence spacing. Retrieved May 02, from Wikipedia, the free encyclopedia: http://en.wikipedia.org/wiki/Sentence_spacing

[11] Jebran, A. (2007, June 19). Text 2Text Steganography - Part 2. Retrieved January 6, from Code Project: http://www.codeproject.com/Articles/19260/Text-2Text-Steganography-Part-2.